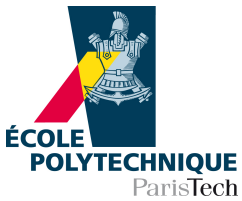
Link-State Routing Optimization for Compound Autonomous Systems in the Internet

Juan Antonio Cordero



Link-State Routing Optimization for Compound Autonomous Systems in the Internet

A dissertation presented

by

Juan Antonio Cordero

to

The Doctoral School of École Polytechnique (EDX)

in partial fulfillment of the requirements

for the degree of

Docteur de l'École Polytechnique

in the subject of

Mathematics & Computer Science

École Polytechnique

France

June 2011

Optimisation du Routage à État de Liens dans les Systèmes Autonomes Hybrides sur Internet

Thèse presentée

par

Juan Antonio Cordero

à

l'École Doctorale de l'École Polytechnique (EDX)

pour obtenir le grade de

Docteur de l'École Polytechnique

en

Mathématiques et Informatique

École Polytechnique

France

Juin 2011

Thesis defended on September 15th, 2011, before a Ph.D jury formed by:

Thèse soutenue le 15 septembre 2011, devant un jury formé par:

- **Philippe ROBERT**, Ph.D, HDR, INRIA Paris–Rocquencourt / École Polytechnique

  Examiner and president of the jury — *Examinateur et président du jury*

- **Christophe GUETTIER**, Ph.D, SAGEM Défense et Sécurité

  Examiner — *Examinateur*

- **Thomas R. HENDERSON**, Ph.D, The Boeing Company / University of Washington (UW)

  Examiner — *Examinateur*

- **Mark TOWNSLEY**, Cisco Inc.

  Examiner — *Examinateur*

- **Mukul GOYAL**, Ph.D, University of Wisconsin–Milwaukee (UWM)

  Reviewer — *Rapporteur*

- **David SIMPLOT-RYL**, Ph.D, HDR, INRIA Nord / Université de Lille 1

  Reviewer — *Rapporteur*

- **André-Luc BEYLOT**, Ph.D, HDR, ENSEEIHT / Université de Toulouse

  Reviewer — *Rapporteur*

- **Philippe JACQUET**, Ph.D, HDR, INRIA Paris–Rocquencourt / École Polytechnique

  Ph.D advisor — *Directeur de thèse*

- **Emmanuel BACCELLI**, Ph.D, INRIA Paris–Rocquencourt

  Ph.D co-advisor — *Co-encadrant*

Awarded with mention

Qualifiée avec mention

Très honorable

Thesis advisor(s) Author

Philippe Jacquet & Emmanuel Baccelli **Juan Antonio Cordero**

Abstract

This manuscript addresses the coexistence of planned and spontaneous interconnected networks in the Internet core. In this realm, the focus is on routing within a specific type of Autonomous System (AS) called compound AS, which contains both wireless ad hoc networks and wired fixed networks. The approach studied in this manuscript is to enhance existing Interior Gateway Protocols (IGPs), typically based on the link-state algorithm, in order to enable them to operate both in ad hoc networks and in wired networks.

The manuscript thus analyzes the use of link-state routing in ad hoc networks. Based on this analysis, different techniques are proposed and theoretically evaluated, aiming at optimizing the performance of link state routing in a compound AS.

The manuscript then investigates the impact of these techniques when applied to OSPF, one of the main IGPs used in the Internet. The performance of OSPF extensions on MANETs using the studied techniques are compared via simulations. Finally, OSPF operation over compound internetworks is evaluated via experiments on a testbed.

Directeur(s) de thèse

Philippe Jacquet & Emmanuel Baccelli

Doctorant

Juan Antonio Cordero

Résumé

Ce manuscrit étudie la coexistence de réseaux fixes et de réseaux spontanés dans le coeur d'Internet. Plus particulièrement, on étudie le problème du routage dans un certain type de système autonome (AS) appelé AS hybrides, qui contiennent à la fois des réseaux ad hoc sans fil et des réseaux filaires. L'approche proposée dans ce manuscrit est d'adapter des protocoles actuellement utilisés dans les AS au coeur d'Internet, typiquement basés sur l'algorithme à état des liens, pour leur permettre d'opérer dans les réseaux ad hoc (MANETs) comme dans les réseaux filaires.

Le manuscrit analyse donc l'utilisation du routage à état de liens dans les réseaux ad hoc. Différentes techniques sont ensuite proposées et évaluées théoriquement, dans le but d'optimiser la performance des protocoles à état de liens dans les AS hybrides.

Le manuscrit étudie alors l'impact de ces techniques lorsqu'elles sont appliquées à OSPF, l'un des principaux protocoles actuellement utilisés dans les AS. Les performances d'OSPF dans les MANETs utilisant les différentes techniques étudiées sont ensuite analysées au moyen de simulations. Pour finir, le fonctionnement du protocole OSPF utilisant certaines des techniques étudiées est évalué au moyen d'expériences sur un réseau test réel.

"Mi voz buscaba el viento para tocar su oído."
(Pablo Neruda, Veinte poemas de amor
y una canción desesperada)

"— Ce qui embellit le désert, dit le petit prince,
c'est qu'il cache un puits quelque part..."
(Antoine de Saint-Exupéry, Le petit prince)

"— How am I to get in?, asked Alice again, in a louder tone.
— Are you to get in at all?, said the Footman. That's the first question, you know.
It was, no doubt: only Alice did not like to be told so."
(Lewis Carroll, Alice's adventures in Wonderland)

A mis padres, por todo.

A mi hermana, que aunque aún no lo sepa,
ya ha empezado a ganar el pulso con la ciencia
que todo científico debe dar antes o después.

Y a mi primo Luis Antonio,
que es otro Cordero expatriado.

Acknowledgments

First of all, I want to thank the three professors and researchers from the HIPERCOM team that have guided my research during my stay in Paris: Philippe Jacquet, Emmanuel Baccelli and Thomas H. Clausen. Their scientific advise has been essential for the successful completion of this Ph.D. Without their support and their help, the work presented in this manuscript would have been less complete and way more painful to do.

I want to thank all those who accepted to review the manuscript, and in particular the members of the jury, reviewers and examiners, for accepting to examine my work. Their comments have been useful and constructive, and the manuscript has significantly profitted from their experience and feedback.

I would also like to thank my current and former colleagues in the HIPERCOM team and LIX for their friendship and the nice times shared in the lab; they made the office a very pleasant place to work. I greatly enjoyed working, but also discussing and having fun, with (in approximate order of appearance) Ulrich Herberg, Georgios Rodolakis, Song-Yean Cho, Nestor Mariyasagayam, Veronika Bauer, Matthias Philipp, Georg Wittenburg, Alberto Camacho, Axel Colin de Verdière, Jiazi Zi and Ming-Chih Chien. The same goes to other colleagues from LIX, in particular Vivek Nigam and Carlos Olarte.

I want to thank the assistants that have worked in HIPERCOM during my stay in Paris for their kindness and help in any administrative matter. Without the support of Lydie Fontaine, Marie-Jeanne Gaffard, Cindy Vingadassamy and Valérie Lecompte, I would have been lost too many times within the complex administrative procedures of French administration.

Outside Polytechnique, I want to mention the friends I have met at the *Colegio de España* in Paris, mostly (but not only) Ph.D students as me. They reminded me that there was something else in life than research, and made smoother the last months of hard work before the defense.

Beyond the Pyrenées, I am of course very grateful to my parents and my sister Elena. This work would simply not exist without them, their tireless encouragement and their patience, anytime, anywhere.

I thank also Clàudia for her support in these years.

Table of Contents

List of Figures

List of Tables

Introduction

Since the first computer networks appeared in the nineteen-sixties, two trends have been present in the evolution of computer networking. The first trend is related to the increase of the number of users that can exchange information or access to contents by way of computer networks – that is, *which* and *how many* computers are involved in communication. The second trend is set towards broadening the range of situations in which communication can be established among a set of user devices – that is, *when*, *where* and *how* communication is enabled over a computer network.

Spread and Growth of Computer Networks: Internetworking and the Internet

The first trend has led to the spread and growth of computer networks, on one hand, and the development of internetworking, on the other. Internetworking consists of interconnecting existing computer networks in such a way that users attached to any of these networks can interact with users from any other. In particular, communication between users is possible even when they are attached to networks based on different technologies. The main example of internetworking is the Internet itself, a world-wide collection of interconnected networks that enables communication among hundreds of millions of computers and users[1]. Figure 0.1 shows a simplified representation of the way that networks are connected to each other through the Internet[2]. Each point in the

[1] According to the *Internet Domain Survey Count* (July 2010), `http://www.isc.org/solutions/survey`, the Internet is estimated to integrate more than 750 million hosts connected through different networks.

[2] Image from The Opte Project, `http://opte.org`. The figure traces the path through the Internet followed by packets sent from a single computer towards every Class C networking block – that is, within the range of IPv4 addresses between `1.0.0.0/24` and `255.255.255.0/24`. Such paths are monitored by way of the `traceroute` utility. The Internet architecture and the IPv4 addressing model are described in chapter 1 of the manuscript.

picture represents a network able to contain a maximum of 254 computers. The picture provides a simplified view of the Internet topology, as networks represented as points may be divided, in turn, in several subnetworks.

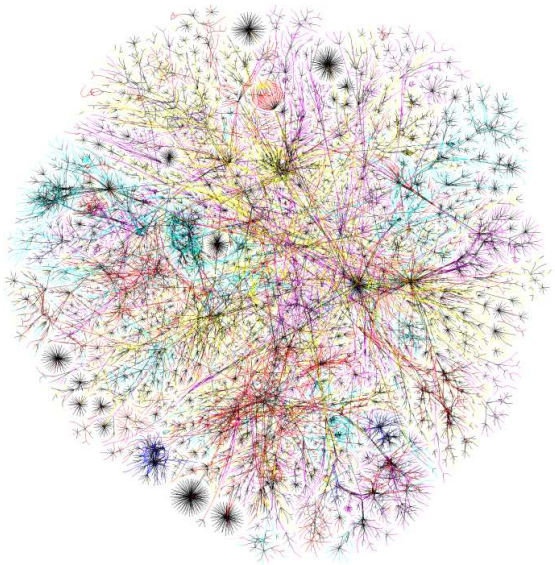


Figure 0.1: Visual map of the Internet recreated by The Opte Project (data from November 2003).

The exchange of information between distant users through the Internet is performed through a complex networking infrastructure, that involves the following:

- A large number of inter-network high-capacity connections, sometimes referred as the *Internet backbone*.
- The Internet core protocols which are a set of common rules for information transmission and forwarding.
- The activity of a number of global entities (such as ICANN-IANA[3], IETF[4] and others) that provide global management, interoperability, administration and standardization services for the Internet.

[3]ICANN: Internet Corporation for Assigned Names and Numbers; IANA: Internet Authority for Assigned Numbers.

[4]The Internet Engineering Task Force.

Unlike other world-wide network infrastructures (such as telegraph or analogue telephone network), the Internet infrastructure enables users to send and receive natively (*i.e.*, without modems) any kind of digital information – not only voice or alphanumerical characters.

More Flexible Computer Networks: Ad hoc Networking

The second trend in computer networking focuses on requirements for setting up a computer network. The first computer networks were based on three main assumptions: (i) computers were mostly connected through wires; (ii) the topology was static, meaning that the way that computers were connected to each other was not supposed to change, and (iii) this topology was known in advance. Under these assumptions, interaction between a computer and the rest of the network was performed through a predictable and stable set of neighbors with which the computer could communicate directly. In case of topology change, the intervention of a central authority (either human or automatic) was required to restore or establish connectivity. As the Internet was developed in parallel with these first computer networks, this type of interaction between computer and network was also assumed in the Internet.

These three assumptions were relaxed as computer networks became bigger and more complex. The growth of the Internet and the decentralization of its architecture implied that topology was not known and could not be longer handled in a centralized manner – instead, distributed routing approaches were implemented in the Internet during the 1980s and 1990s [117, 127]. Moreover, the use of wireless communications in computer networks started to spread in the 1980s, when unlicensed use of wireless spectrum bands – the Industrial, Scientific and Medical bands – was allowed by the US Federal Communications Commission (FCC). Computer networks based on wireless communication present more dynamic topologies, and this dynamism increases significantly if computers in the network are allowed to move. While computer networks became more popular, wireless communication became more widespread and computer mobility more common (*e.g.*, in the context of embedded networking devices in smartphones or vehicles). Thus, the need of more flexible models for computer networking became unavoidable [92]. In the 1990s, the concept of Mobile Ad hoc Net-

working was introduced to address network dynamism – this revealed useful for computer networks in which the previously stated assumptions (i) to (iii) cannot be assumed.

The concept of Mobile Ad hoc Network (MANET) provides an abstract model for networking with the highest degree of flexibility with respect to such characteristics: MANETs are wireless networks, designed to operate when:

(i) the topology is not known in advance;

(ii) the topology may change in an unpredictable manner, at any time and at any rate (for instance because elements of the network are mobile relatively to one another); and

(iii) no network infrastructure (physical connections between computers, networking hierarchy or central authority) can be assumed to be available.

Computers in a MANET thus cannot count on a predictable and stable set of neighbors through which they can interact with the network, nor on a central authority to advertise topology changes. Instead, the fact that topology in ad hoc networks is dynamic implies that computers have to be able to interact with the network as a whole, by way of the sets of neighbors that are rechable at each particular time. For that, they need to rely on the cooperation of neighboring computers that are able to forward information over the network, that is, neighboring routers. Such cooperative interaction is necessary both for keeping track of topology changes, and for enabling communication even when the set of available neighbors cannot be accurately determined.

Ever since the IETF formally defined MANETs in 1997 [89], envisioned applications of such networks have ranged from wireless sensor networks to vehicular networks, also including emergency and military deployments. Routers of a wireless sensor network [27, 61], for instance, are usually spread arbitrarily and thus produce static multi-hop topologies that cannot be predicted *a priori*. In Vehicular Ad Hoc Networks (VANETs) [68], topology changes rapidly due to high relative speed between devices installed in moving vehicles. In cases of recovery deployments for catastrophes or natural disasters (earthquakes, flooding, etc.) or military deployments, topology may also be dynamic and networking devices cannot rely on existing communication infrastructure because such

infrastructure may be damaged, destroyed or insecure. In all these cases, establishing communication presents challenges and issues.

Routing in Internetworks

The use of internetworking and ad hoc networking permits achieving two goals. Internetworking enables communication among an increasing number of users that are connected by way of a world-wide networking infrastructure, the Internet. Ad hoc networking, in turn, improves the capacity to establish network communication through computers that are deployed in a dynamic and non-predictable fashion.

Both quantitative and qualitative improvement of networking communication capabilities, can be achieved simultaneously by combining Mobile Ad hoc Networking and the Internet, that is, integrating ad hoc networks into the Internet architecture. This is the problem explored in this manuscript. Internetworks that result from such combination are those that support ad hoc properties in (parts of) their topology while being capable of communicating through the Internet infrastructure. As these internetworks present the same flexibility properties as MANETs in at least parts of their topology, they can be used for the very same purposes, *e.g.* vehicular communications, decentralized sensor deployments, etc. The fact that these internetworks are connected or embedded into the Internet by way of fixed networks implies that they can also be used for additional purposes – user Internet access, social networking, geographic services and such.

This manuscript restricts to the problem of routing within such internetworks: building and maintaining routes through which data can be sent from and towards computers in the internetwork. More precisely, the manuscript addresses the setting-up of mechanisms for enabling communication and information exchange (i) between computers from within one of the networks part of the internetwork, and (ii) between computers from one network and the rest of the internetwork. Such mechanisms are needed to ensure that information is routed successfully within the internetwork.

Figure 0.2 illustrates the two approaches possible for such internetworks. As routing properties of ad hoc networks and fixed networks differ significantly, a natural approach consists of

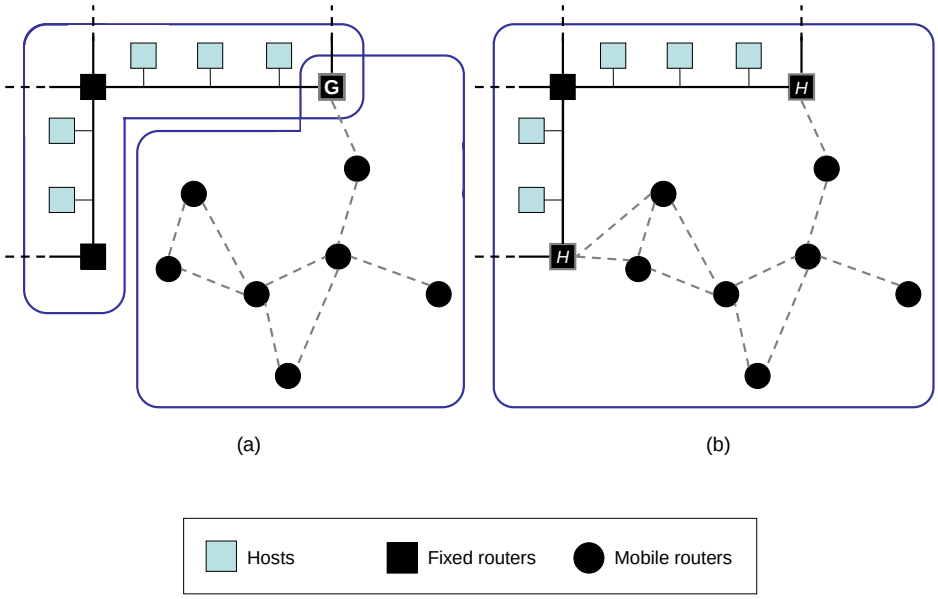


Figure 0.2: Two approaches for routing in an internetwork containing ad hoc networks and fixed networks connected to the Internet: **(a)** two routing domains, one for the fixed network and another for the ad hoc network, connected through a gateway G, and **(b)** one single routing domain that contains the ad hoc and the fixed networks of the internetwork, which are connected through two H routers. *While it is possible to use more gateways in (a) in order to improve connectivity between domains, each additional gateway G is costly due to the specific hardware and routing configuration required for these gateways, together with the additional complexity introduced in the internetwork. This is not the case in (b), as H routers do not need capabilities other than those from the rest of routers.*

treating ad hoc and fixed networks as separate routing domains, with each routing domain being a part of the internetwork in which routers use the same instance of a routing protocol (Figure 0.2.a). The fixed networks that provide access to the Internet may use one of the Internet routing protocols, as OSPF[5] or IS-IS[6], while the attached MANET(s) may use instances of a specific protocol optimized for ad hoc operation, such as OLSR[7] or AODV[8]. The use of different routing protocols in the same internetwork makes necessary the presence of *gateways*, denoted G in Figure 0.2.a. Gateways are specific routers that ensure the exchange of routing information between the different routing

[5]Open Shortest Path First protocol [107].

[6]Intermediate System to Intermediate System protocol [122].

[7]Optimized Link-State Routing protocol [71].

[8]Ad hoc On-demand Distance Vector protocol [75].

domains in the internetwork, and therefore participate in both the ad hoc and the fixed networks, and they provide support for the different involved routing protocols.

This approach has three main drawbacks. First, the use of different protocols in the same internetwork is more difficult to handle than the use of a single protocol, and thus also more expensive in terms of hardware/software requirements, network maintenance and configuration. Second, gateways cause an additional level of complexity in terms of management and routing of the whole internetwork. This additional level of complexity comes from the fact that gateways need to be able to distribute the necessary routing information among different networks, in order to ensure that computers in any part of the internetwork can communicate. As these tasks typically involve specific hardware and software for gateways, such complexity also implies higher costs. Third, inter-network routes are not necessarily optimal, even if the involved routing protocols are designed to provide optimal paths in their respective domains. In the case of several routing domains, routes traversing gateways consist of the juxtaposition of several (at least two) "locally" shortest paths (optimal in each routing domain traversed by the route), which does not necessarily lead to a "globally" shortest path (in the whole internetwork). Moreover, these drawbacks cannot be simultaneously minimized, as they are closely intertwined: reducing the number of gateways, while alleviating the additional complexity and costs, may damage significantly the quality of the performed routes (suboptimality).

Instead of separate routing domains, this manuscript explores the second approach, illustrated in Figure 0.2.b. This approach seeks to address these drawbacks by developing a single routing domain in the internetwork that contains both ad hoc networks and fixed networks, and is thus handled by a single routing protocol in a single routing domain. The use of a single protocol in the internetwork implies that gateways are no longer necessary, and that route computation is performed over the whole internetwork, therefore improving the quality of the selected routes. With this approach, the role of gateways is fulfilled by simple routers, which have interfaces both to ad hoc and fixed routers, and use the same routing protocol as any other router in the routing domain.

Link State Routing in Compound Internetworks

Internetworks that combine ad hoc and Internet fixed networks are denominated *compound* internetworks throughout this manuscript, which explores a single routing protocol for such internetworks. In this context, routing can be performed by way of different techniques. The main protocols used for routing within Internet fixed networks are, however, all based on the *link-state* technique [100]. This manuscript explores and analyzes the use of this link-state technique for routing in compound internetworks, not only for the fixed networks but also for the (mobile) ad hoc networks of the internetwork.

Link-state algorithms are based on the assumption that routers acquire and maintain information about the topology of the network in which they are used – this information forms the *Link State Database* (LSDB) of the network. This information is disseminated over the network through a local-to-global distributed procedure: routers describe their *local* topology and flood these descriptions to the *whole* network. By receiving topology descriptions and updates from every other router in the network, any router is able to maintain a complete description of network topology. Based on this description, routers compute the best (shortest) paths to every possible destination in the network – Dijkstra's algorithm [135] is used to determine such shortest paths.

OSPF and IS-IS protocols are the main examples of link-state routing protocols for networks in the Internet. The two protocols are similar in several aspects: both have a modular architecture, meaning that they are able to support different extensions for specific networking properties, and different extensions may coexist in the same routing domain while using the same core mechanisms. Also, both have been designed for wired networks with static topologies and therefore are not adapted to the challenges and restrictions of wireless ad hoc networking. For instance, control traffic generated in standard OSPF and IS-IS operation, while manageable in the context of wired and fixed networks, becomes excessive in wireless ad hoc networks in which bandwidth is severely limited. In order to be applicable in ad hoc networks, these link-state protocols need therefore to be adapted in their operation to accommodate the new restrictions and features that are present in such networks.

This is the approach that is developed throughout this manuscript. Taking advantage of modular architecture, the extension of already existing Internet link-state routing protocols for operation in MANETs is explored. The objective of such an extension is two-fold. First, to minimize changes in the routing infrastructure of fixed networks already in use inside a compound internetwork. Second, to obtain an extended protocol that can be used as single routing protocol for all networks (fixed and ad hoc) of a compound internetwork. The extended protocol should be therefore able to accommodate the properties and issues of ad hoc networking in the Internet without requiring substantial changes in the routing mechanisms already used for Internet networks.

Network Overlays in Link State Routing

In order to ensure accuracy and consistency of topology information maintained by routers running a link-state protocol, different operations need to be performed over the network. Such operations are related to the advertisement of topology changes to all the routers in the network: description, flooding and synchronization of LSDB. In ad hoc networks, these operations are performed in a distributed fashion, meaning that routers autonomously take the decisions required to execute each of such operations. The way to perform these operations needs to take into account the properties and limitations that prevail in MANETs: in this manuscript, link-state operations are treated separately due to significant differences between such operations, in terms of goals, scope, involved routers and impact in the network. The manuscript introduces the concept of a *network overlay*, to be associated with each link-state operation, and proposes an analysis of the link-state routing technique and each of their related operations in terms of such overlays.

A network overlay is a network built on top of an existing computer network. In literature, a network overlay usually denotes an abstraction layer in which an underlying networking infrastructure (one or more computer networks already existing and enabling communication between any pair of attached computers) is used to provide specific communication services between computers of the network [21]. In such cases, the topology of the network overlay may be independent from the topology of the underlying network: any topology is possible as far as the involved computers

are connected through the underlying network. The Internet itself can be understood as an overlay network, and other well-known examples include peer-to-peer (P2P) networks for file exchange [90], content distribution [95] or multicast video-conference services [96].

In this manuscript, however, the term of *overlay* is used in a slightly different sense. Rather than an arbitrary topology built on top of an existing networking infrastructure, a link-state overlay over a MANET includes some of the computers attached to the network and uses some of the available links between such computers to perform one of the above-mentioned link-state operations. For each of these operations, the manuscript explores requirements and recommended properties that the associated overlay should satisfy. Based on this exploration, the underlying trade-offs for different operations are identified, and several distributed techniques for building and maintaining link-state overlays are examined and compared.

Identification of link-state operations and separate analysis of the corresponding link-state overlays permit independent optimization of the performance of each of the associated link-state operations. Such optimizations apply to MANET extensions of modular link-state protocols. An extended protocol that uses one of such extensions can then be used for routing in compound internetworks. While this manuscript focuses on the particular case of OSPF, the performed analysis and the presented arguments can be generalized to other Internet link-state routing protocols, such as IS-IS.

Structure and Overview

This manuscript is organized in three Parts. The main concepts and elements of networking are presented in Part I. Chapter 1 introduces basic concepts related to computer networks (interface, link, network, routing) and presents a brief overview of the notion of internetworking and the Internet addressing and routing architecture. Chapter 2 concentrates on the specific case of wireless networks, pointing out the impact that the use of radio channel has in terms of network communication. Chapter 3 analyses the issues and challenges that arise in the context of wireless multi-hop ad hoc networks, a particular class of wireless networks. This chapter also presents and discusses the

implications of the notion of compound Autonomous Systems, as the result of embedding ad hoc networking into the traditional Internet networking framework.

Part II studies the implementation of link-state routing mechanisms for (mobile) ad hoc networks. Chapter 4 describes the characteristics and operations related to link-state routing, first, and identifies the most relevant issues that need to be addressed for performing link-state routing, second. Chapter 5 elaborates on the problem of packet collisions due to simultaneous retransmissions during flooding in wireless networks, and analyzes (both theoretically and through simulations) the impact of jittering. This technique consists of distributing, over time, wireless retransmissions of the same packet, in order to avoid collisions. Chapter 6 introduces the concept of a *link-state overlay* associated to a link-state operation, and identifies the required properties for each link-state overlay based on the characteristics of its associated operation. The analysis in this chapter provides the criteria to examine, evaluate and compare the different link-state overlay techniques proposed in following chapters. Chapter 7 proposes the Synchronized Link Overlay (SLO) technique and presents a theoretical analysis of the properties of its associated network overlay, focusing on its density and the stability of their links. Most results presented in this chapter are published in [14] and in [10]. Chapter 8 focuses on the Multi-Point Relaying (MPR) technique [88]. Although MPR is primarily used for flooding purposes, the chapter explores the applicability of MPR and MPR-based techniques for other link-state operations, LSDB synchronization and topology selection. The discussion and analysis of techniques based on MPR for topology selection purposes is published in [12]. Finally, chapter 9 studies the Smart Peering technique and discusses its applicability as a synchronization technique, some of the presented results being included in [4]. A summary of the main results presented in this Part was published also in [3].

Finally, Part III applies the previously presented techniques to OSPF, one of the main Internet link-state routing protocols. Chapters in this Part evaluate the performance of these techniques as extensions of OSPF for ad hoc networks, and studies the extended OSPF protocol as a candidate for link-state routing in compound internetworks, based on network simulations and a real testbed. Chapter 10 describes the operation and architecture of OSPF, as well as some significant

aspects of IS-IS, in order to identify similarities between both protocols. Chapter 11 examines some existing extensions of OSPF for MANET operation, and proposes some additional improvements based on the analysis deployed in Part II. Presented extensions include those standardized by the IETF: RFC 5449 [24], RFC 5614 [22] and RFC 5820 [19]. Proposed additional extensions include MPR+SP, based on the combination of RFC 5449 and RFC 5820, presented and evaluated in [4]; a variation of RFC 5449 that uses the SLOT technique for synchronization (SLOT-OSPF, evaluated in [10]); and some additional variations of RFC 5449 that explore use of link persistency in different link-state overlays. Chapter 12 performs an analysis of the main aspects that are required for a MANET extension of OSPF based on comparison via simulation of the presented extensions. Results and experiments described in this chapter have been published in different papers, in particular [20] and [13] for the comparison between RFC 5449 and RFC 5820, and [11] for the impact of MPR link change rate and different persistent strategies in RFC 5449. Chapter 13 completes these analysis by describing set-up, operation and experiments of a testbed, composed of a wired and a wireless network, in which routing is performed by way of OSPF extended with the MPR-OSPF extension for wireless interfaces; results from these experiments have been documented in [1].

The final chapter concludes this manuscript by presenting and summarizing final results, their implications and perspectives for future work.

Part I

NETWORKING FUNDAMENTALS

Chapter 1

Computer Networks

In 1962, L. Kleinrock introduced networking based on packet switching [134]. Before that, communication between two points (nodes) was only possible by establishing a persistent electrical circuit between them, through which data could be sent. That was the principle of the Public Switched Telephone Network (PSTN), where a set of terminals or endpoints (typically, but not only, telephones) were connected through a set of wires and telephone switches. These switches were responsible for establishing a persistent circuit between the calling terminal and the called terminal. Once the circuit was established, its use was exclusively reserved to the two connected endpoints. Such circuit (telephone call) was maintained until the end of the communication (*e.g.*, voice conversation), after which the connection was closed. Figure 1.1.a illustrates the main characteristics of PSTN calls: during the call between terminals A and C, no other terminal is able to establish communication with either A or C, as the circuit between A and C is persistent and exclusive.

Packet switching is based on a different approach (see Figure 1.1.b). Rather than communicating by establishing persistent circuits between endpoints, the use of data packets permits using the same channel (*e.g.*, a wire) to provide support for simultaneous communications between many different pairs of endpoints. Data to be sent from a source to a (set of) destination(s) is encapsulated in data units, called *packets*, each of which can be treated autonomously and separately. These packets may need to be forwarded by one or more intermediate nodes before reaching their

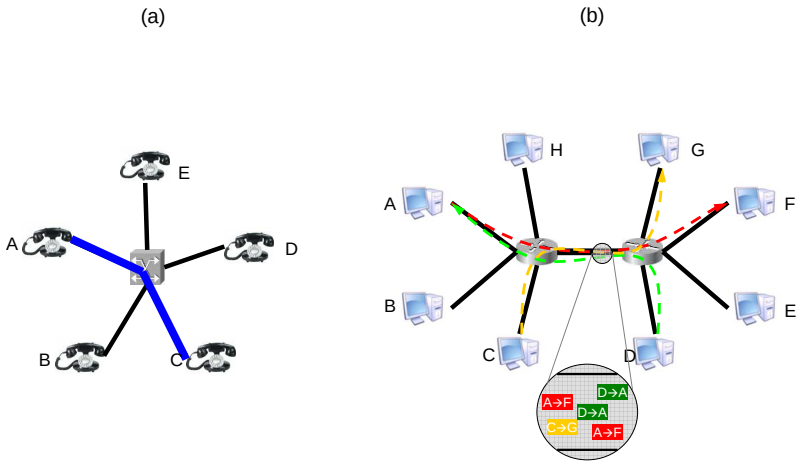


Figure 1.1: Examples of **(a)** communication through circuit switching in PSTN, and **(b)** communications through packet switching networking.

final destination(s).

This approach enables more flexible communication between nodes within a network than the circuit-switching approach, as it enables any endpoint to maintain several communications concurrently. By not dedicating the channel to a particular pair of endpoints, it also allows a more efficient use of the channel. This is at the expense of lowering reliability of communication: packets in a packet-switching network may be lost or delivered out of order. Characteristics of circuit switching are appropriate for requirements and properties of voice transport (reliable communication, delivery of data in the same order in which it was sent, balanced amount of data in both directions); packet switching, in turn, has become the basis of computer networking, and in particular the Internet.

1.1 Outline

This chapter presents the main elements of computer networks and the Internet. Section 1.2 presents the basic terms and concepts of computer networking – network, interface, link, routing and routing protocol. While many terms are in common use in networking research, they are defined formally in this section in order to avoid ambiguity and clarify the precise meaning and the sense in which they are employed throughout this manuscript. Section 1.4 addresses the interconnection of

existing networks (internetworks), presents the concept of internetworking and provides an architecture overview of the most prominent case of internetwork – the Internet. In particular, the section describes the IP addressing model and the Internet routing hierarchy. Finally, section 1.5 concludes the chapter.

1.2 Networking and Routing Concepts

This section presents and discusses the basic elements of computer networking. Section 1.2.1 defines the concepts of packet, computer network, interface and link. Section 1.2.2 presents the graph representation of a network and discusses its interest as analysis tool. Based on these definitions, section 1.3 elaborates on the conditions that need to be fulfilled in a computer network so as to ensure that information can be exchanged between computers.

1.2.1 Networks and Links

A computer network is defined as follows:

Definition 1.1 (**Packet computer network**)**.** A *computer network* is a set of two or more computers that are connected in such a way that every pair of computers can exchange information. A *packet computer network* or *packet-switching computer network* is a computer network in which information is exchanged by means of **packets**, *i.e.*, data units that contain sufficient information about their source and destination(s) to be routed and delivered separately through the network. Unless otherwise specified, all references to networks relate to packet computer networks.

Computers are connected to other computers in a network through *links*.

Definition 1.2 (**Link between computers**)**.** There is a *link* between two computers A and B, denoted by $A \longrightarrow B$, if and only if A is able to transmit data to B and B is able to receive such data, without intervention of any other computer.

Definition 1.3 (**Symmetric link between computers**)**.** A link between two computers A and B is said to be *symmetric* (or *bidirectional*), and denoted by $A \longleftrightarrow B$, if and only if there are links

$A \longrightarrow B$ and $B \longrightarrow A$, *i.e.*, data can be transmitted from A and received by B and vice versa, without intervention of any other computer.

A computer participates in a link by way of a **network interface**:

Definition 1.4 (**Network interface**)**.** A *network interface* of a computer is a device that provides access from that computer to a link through an underlying physical communication channel.

In this sense, link definitions 1.2 and 1.3 can be rephrased as follows, in terms of interfaces:

Definition 1.5 (**Link between interfaces**)**.** There is a *link* between two network interfaces a and b, denoted by $a \longrightarrow b$, if and only a is able to transmit data (bits) to b and b is able to receive such data, without the intervention of any other interface.

Definition 1.6 (**Symmetric link between interfaces**)**.** A link between two network interfaces a and b is said to be *symmetric* (or *bidirectional*), and denoted by $a \longleftrightarrow b$, if and only if there are links $a \longrightarrow b$ and $b \longrightarrow a$, *i.e.*, data can be transmitted from a and received by b and vice versa, without requiring the intervention of any other interface.

The existence of a link between two computers implies the existence of (at least) one link between two network interfaces of these computers. Let A and B be two computers, and let $I(A)$ and $I(B)$ be the set of network interfaces of A and B, respectively; then,

$$A \longrightarrow B \Longrightarrow \exists a \in I(A), b \in I(B) : a \longrightarrow b$$

Reciprocally, the existence of a link between two network interfaces implies the existence of one link between the computers to which the interfaces are attached. In this manuscript, the term *link* denotes a link between network interfaces, unless otherwise specified.

Unless stated otherwise, the term *link* in this manuscript denotes a symmetric link. Non-symmetric links are explicitly called *asymmetric links*.

Depending on the number of interfaces in a link, different types of links can be distinguished. Figure 1.2 illustrates three different types of links and networks: broadcast links, point-to-point links and wireless links. The first two are defined in definitions 1.7 and 1.8; wireless links are described in chapter 2.

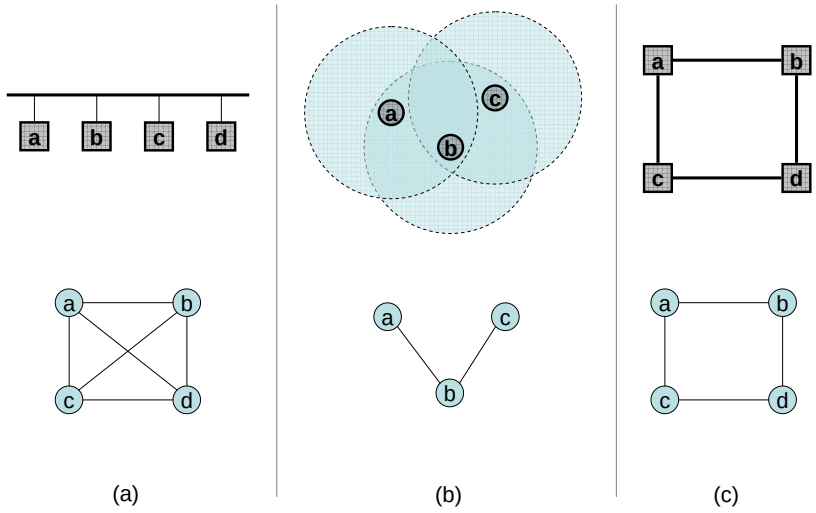


Figure 1.2: Examples of computer networks and links, with their network graph representations: **(a)** Broadcast network based on a single multiple-access link, **(b)** Wireless multi-hop network with several links, **(c)** Distributed network based on several point-to-point links. *The existence of an edge between two vertices in a network graph implies that there is a link between the interfaces represented by such vertices.*

Definition 1.7 (**Point-to-point link**)**.** A link l between two network interfaces a and b is a *point-to-point link* if and only if data can be transmitted from a to b (and/or vice versa) by way of l and no other interfaces x and y ($x \neq a, b; y \neq a, b$) can exchange information through the same link l.

Definition 1.8 (**Broadcast link**)**.** A link l is a *broadcast link* for a set of network interfaces $\{x_i\}_{i \leq k}$ if and only if data can be transmitted from x_i to x_j for any value of $i, j \leq k$, and a packet transmitted by any interface x_i is received by every other interface in the network x_j, $j \neq i$.

- Defs. 1.5 and 1.8 imply that links between different interfaces (*e.g.*, $a \longrightarrow b$ and $c \longrightarrow d$ in Figure 1.2.a) may correspond to the same broadcast link. For a criterion to identify equivalent links, see the link equivalence relation presented in Appendix A.

- Broadcast links are always symmetric: for any interfaces a and b attached to such a link, data can be transmitted either from a to b or from b to a.

Definitions of broadcast and point-to-point links illustrate particular cases of the concept

of *link*: both allow communication from one network interface to another through a physical communication channel – in the case of the broadcast link, in particular, information can be exchanged between *any* pair of attached network interfaces. Examples of point-to-point links include PPP[1] (see Figure 1.2.c), while the most prominent examples for broadcast networks include Ethernet and Token-Ring technologies (the architecture of a broadcast network is displayed in Figure 1.2.a).

Broadcast and point-to-point categories do not cover all possible cases of link. Communication between wireless network interfaces, in particular, cannot be modeled in general by any of these two definitions: in the example of Figure 1.2.b, the wireless link between b and c is not a point-to-point link (as packets sent from b to c are also received by a) and neither is a broadcast link (in particular, a cannot receive packets sent by c). Properties and challenges of wireless links and networks are discussed in detail in chapter 2.

1.2.2 Graph and Hypergraph Representation

The topology of a computer network at a particular point of time can be represented as a graph $G = (V, E)$, in which the set of vertices V corresponds to the set of attached computers and the set of edges V indicates the presence of links between computers. Such graph G is called **network graph** throughout this manuscript, and is assumed to be connected – otherwise, G denotes the network corresponding to a connected component of the graph instead. Given two vertices x and y of V, the edge $\overline{xy}$ is included in E if and only if there is a link between computers represented by x and y. Asymmetric links are represented by directed edges, while symmetric links correspond to undirected edges.

The graph representation of a network is useful for a number of purposes, and is used throughout this manuscript to analyze properties of networking and routing algorithms from a theoretical perspective. For instance, the path that a packet follows from a source computer, x, to a destination computer, y, can be represented as a path through the network graph, p_{xy}.

Definition 1.9 (**Network path**). A *network path* between two vertices $x, y \in V$ in a network

[1]Point to Point Protocol, basic specification in RFC 1661 [119].

graph $G = (V, E)$ is a collection of edges of E, $p_{xy} = \{\overline{xm_1}, \overline{m_1 m_2}, ..., \overline{m_{k-1} y}\}$ such that every pair of contiguous edges have one vertex of V in common. Given p_{xy}, $|p_{xy}| = k$ denotes the **length** of the path, that is, the number of hops of the path.

However, the graph abstraction has some significant limitations that need to be taken into account. The most relevant is that different edges in a network graph do not necessarily indicate different links in the network: the same link may be represented by several (at least one) edges. Figure 1.3 illustrates some implications of this fact: networks with different architectures may present equivalent graphs.

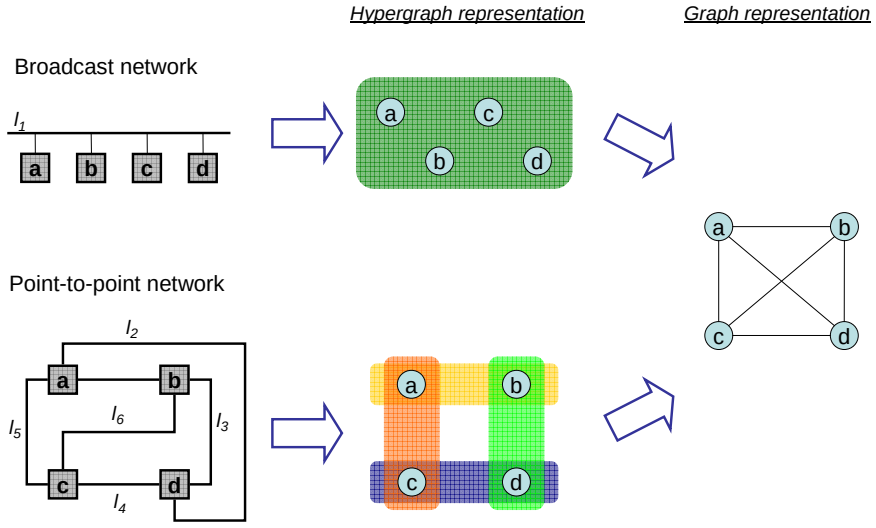


Figure 1.3: Two networks with different architectures and different number of links may have the same graph.

For networks in which the number of interfaces (computers) participating in a link can be higher than two, such as broadcast or wireless networks, links do not necessarily correspond to edges and therefore, the graph representation cannot be used for analyzing aspects such as collisions or available bandwidth in a shared medium. The properties of wireless and mobile communication, in particular, impose additional constraints to the validity of network graphs, that are discussed in chapters 2 and 3.

For a more accurate representation in terms of collisions and link reachability, the notion of

hypergraph may be useful, in particular for wireless ad hoc networks [23]. A **network hypergraph** is a pair $H = (X, \tilde{E})$ where X denotes the set of vertices and $\tilde{E}$ denotes the set of hyperedges. Vertices from X correspond, as for network graphs, to computers attached to the network; an hyperedge $e_x \in \tilde{E}$, where $x \in X$, contains all the vertices corresponding to computers that receive a transmission from computer x, x itself included – in this sense, it generalizes the notion of edge, which is a particular case of hyperedge that only contains two vertices. Formally, an hyperedge e_x is a subset of the hypergraph vertices ($e_x \subseteq X$). Given that the number of vertices included that an hyperedge may contain is not restricted to two, hypergraphs are able to capture more accurately than graphs the connectivity and collision issues in networks where links may involve more than two interfaces.

1.3 Addresses, Direct and Indirect Communication

Communication between computers connected through links and networks may require that the interfaces involved in communication can be identified without ambiguity. These identifiers are called **addresses**.

For links that connect two and only two interfaces (point-to-point links), sender and receiver of a particular packet can be identified by the receiving interface even in the absence of addresses: there is no other possible receiver than itself, and there is no other possible sender than the other interface in the link. For links involving more than two interfaces, however, an interface identity is required. This identity, sometimes called *physical address*, has to be unique within the link in order to enable unambiguous communication with the rest of interfaces in the link.

The transmission of packets from one interface to another in a network requires that:

(i) interfaces have a unique address in the network (*network layer address*), so that source and destination(s) of packets can be unambiguously identified by including such addresses in the packets[2],

[2]Not to be confused with the physical address of an interface, expected to be unique across the link.

(ii) interfaces agree in the formats and procedures to communicate (network technology).

These two conditions are sufficient for enabling communication between network interfaces in the same link: packets are then delivered in a *single hop*, *i.e.*, in the same link that they were transmitted. When two interfaces do not participate in the same link, packets between them need to be **routed** across the network by intermediate computers, that is, sent from the link in which they were first transmitted to a link in which they are received by their destination.

Computers able to perform such forwarding operation between different links are called **routers** (or **intermediate systems**), and those that process information as senders or final receivers are called **hosts** (or **end systems**). Computers can behave simultaneously as hosts and routers as far as they have interfaces attached to (at least) two links and are able to make forwarding decisions [116].

Therefore, communication between interfaces that do not participate in the same link (indirect communication) requires the following additional conditions:

(iii) in case they have multiple interfaces, hosts must be able to determine to which interface (and thus, to which router) packets need to be sent.

(iv) routers must be able to forward packets to their final destination, if there is a link to it, or to a router that is *closer*[3] to the final destination.

Equivalently, hosts and routers in a network must be able, for any packet, to deliver it to a link to which its destination is attached, or to determine the next hop towards its destination. The maps between possible destinations and next hops are called **routing tables**. In case of hosts, the routing table indicates as next hops routers that are reachable through each of the available interfaces. Routing tables from hosts and routers also contain information about the links to which they are able to deliver (and forward, in case of routers) packets. Information collected in the set of routing tables enables thus the communication between computers with interfaces not attached

[3]According to a certain metric.

to a common link; such information is maintained and updated in the routers by way of a *routing protocol*.

For stable networks that contain a small number of hosts and routers, routing tables can be filled and maintained manually, with human operation (static routing). As the network grows, and changes in the topology are more frequent (for instance, due to router failures), routing tasks become more complex and dynamic routing protocols are needed.

Definition 1.10 (**Routing protocol**). A *routing protocol* is a set of procedures performed over the network in order to collect routes and maintain the routing tables of the routers in the network, so that they enable computers to transmit and successfully deliver packets to every possible destination in the network.

There are two main approaches to dynamic routing:

- *Proactive routing.* Routers collect topology information from the network and maintain proactively (*i.e.*, regardless on whether they are used) routes towards all destinations. This way, routers are able to forward packets at any time to any destination in the network. Depending on how the information for such forwarding decisions is acquired, three approaches can be distinguished:
  - *Link state routing.* Routers advertise the status of their links (link-state) to the whole network. This way, every router in the network receives the link-state of other routers in the network, maintains information about the whole network topology and is therefore able to locally compute network-wide shortest paths, usually by way of Dijkstra's algorithm [135]. Examples of this approach are the Open Shortest Path First (OSPF, RFCs 2328 and 5340 [107, 28]) and the Intermediate System to Intermediate System (IS-IS, RFC 1142 [122]) protocols, as well as the Optimized Link State Routing protocol (OLSR, RFC 3626 [71]). OSPF and IS-IS are described in more detail in chapter 10.
  - *Distance-vector routing.* A router shares its routing table only with its neighbors, indicating its *distance* and the next hop towards any reachable destination. Neighbor distance is

defined according to the current link metric, which assigns a scalar cost to any available link in the network. By receiving the routing tables of all its neighbors, which in turn have been shared with the neighbors of the neighbors, a router is able to identify, for each advertised destination, the neighbor that provides shortest distance and select it as next hop. Distance-vector protocols mostly use the distributed Bellman-Ford algorithm [136, 133] to identify network-wide shortest paths. The Routing Information Protocol (RIP, RFCs 1058 [124], 2080 [110] and 2453 [102]) is a prominent example of this family.

  - *Path-vector routing.* Based on the same principle as distance-vector routing, a router advertises to its neighbors the paths to all reachable destinations. Each path is described by indicating the routers that are traversed. This way, local distribution of locally maintained paths enables all routers in the network to build routes to all possible destinations. The most prominent example of this family of protocols is the Border Gateway Protocol (BGP, RFC 1771 [117]).

- *Reactive routing.* A router calculates routes to a destination only when it receives information addressed to that destination and it is not known (*i.e.*, the routing table does not provide a next hop). Dynamic Source Routing (DSR, RFC 4728 [38]) or Ad hoc On-Demand Distance Vector (AODV, RFC 3561 [75]) are examples of reactive routing protocols.

The main advantage of proactive algorithms when compared to reactive algorithms is that all routes are immediately available for proactive routers when the network has converged, which reduces the delay for data traffic with respect to reactive routing protocols. Such immediate availability of routes requires, however, that topology information is flooded periodically over the network and independently from the data traffic load.

Among proactive algorithms, distance-vector and link-state are the main types of algorithms [100] – path-vector algorithms being a variation of distance-vector. Distance-vector protocols were used in the early stages of computer networking, but were replaced gradually by link-state protocols in the Internet. The reasons for this replacement were the existence of problems in distance-vector algorithms, in particular the well-known count-to-infinity problem [70] (which does

not appear on path-vector protocols), as well as the poor scalability and slow convergence properties of distance-vector with respect to link-state algorithms [98, 100].

Convergence time differences between distance-vector and link-state can be observed by looking at the way to advertise the failure of a link over the network. In distance-vector algorithms, once a router detects such a failure, it updates the cost of its route towards the lost neighbor and sends its new distance-vector to its neighbors. Neighbors receive this update and recompute the cost of the affected route, and then transmit in turn their new distance-vectors. Propagation of topology changes is thus slower than in link-state algorithms, in which a router detecting the failure of the link towards one of its neighbors floods an updated topology description which is directly forwarded over the network, without delays caused by route re-computation in intermediate routers [100, 127].

1.4 Connecting Networks

Condition (ii) of section 1.3 states that the information exchange within a computer network requires that the involved interfaces of the corresponding computers agree on the formats and procedures to communicate. Given the existence of different network technologies[4] – and, therefore, different sets of formats and procedures for communication within networks, the question arises on how to connect different networks (that may use different families of communication protocols) and how to enable communication between computers (interfaces) not in the same network.

The Internet Protocol (IP, RFC 791 [128]) provides such ability to exchange information between interfaces belonging to interconnected networks. These interconnected networks are called *internetworks*.

Definition 1.11 (**Internetwork**). An *internetwork* is a computer network (in the sense of def. 1.1) that results from connecting already existing computer networks. Such computer networks may be based on different network technologies.

IP enables communication in internetworks mainly by way of two elements: (a) a common

[4]Some examples: Asynchronous Transfer Mode (ATM), Frame Relay, X.25, Ethernet (IEEE 802.3), Token Ring (IEEE 802.5), Wi-Fi (IEEE 802.11)...

addressing model for interfaces, the IP addressing model, and (b) an additional abstraction layer that permits treating links from different network technologies as IP links. Both concepts (IP addressing model and IP link) are presented in section 1.4.1.

IP is the main protocol for internetworking and one of the base protocols of the architecture of the biggest internetwork in the world, the Internet. The architecture of internetworks and, in particular, the Internet, is discussed in section 1.4.2, and the Internet routing architecture is detailed in section 1.4.3. Due to its popularity, IP has become a standard protocol not only within internetworks, but also for networks based on a single network technology.

1.4.1 IP Addressing and IP Links

IP employs a common addressing model for all interfaces that belong to the internetwork. An identifier assigned to a network interface is called an *IP address* of the interface, and contains information about:

(i) the identity of the interface in the internetwork, by means of the *host identifier*, and

(ii) its location within the internetwork, more precisely the network to which the interface is connected, by means of the *network identifier* or *network prefix*.

Both entities (host identifier and network prefix) are distinguished with the network mask, as illustrated in Figure 1.4. The network mask is a sequence of bits with the length of the address such that, with an IP address IPA and a network mask NM with length $[NM]$:

$$\begin{aligned} IPA \otimes (\neg NM) &= \text{host identifier} \\ (IPA \otimes NM) \gg [NM] &= \text{network prefix} \end{aligned}$$

Where $\otimes$ denotes the AND bitwise operator, $\neg$ the NOT bitwise operator (complement) and $\gg$ denotes the bit right shift.

It is worth to observe that interfaces of two different hosts may have the same host identifier as long as they do not belong to the same network – and thus the network prefixes are different.

An IP address is a *network layer address*, and thus different from the previously mentioned *physical address*: the physical address is only used in the link in which the interface participates, while the IP address identifies the interface within an internetwork. The structure of IPv4 and IPv6 addresses is presented in Figure 1.4. IPv4 addresses have a length of 32 bits, and are commonly represented as a set of four decimal values between 0 and 255 separated by dots (192.168.0.1, in Figure 1.4.a), followed by a slash and the length of the network prefix (24 in Figure 1.4.a). IPv6 addresses are 128 bits long, and are commonly represented as a set of 8 hexadecimal values, each of them between 0 (`0x0000`) and 65535 (`0xFFFF`) separated by colons, and followed by a slash and the prefix length (length of the network mask, 64 in the example of Figure 1.4.b). In case of zero values, the representation may be compressed to ignore them, as long as no ambiguity is introduced [112]: in Figure 1.4.b, 4th to 7th hexadecimal values of the displayed address are zero and thus its IPv6 representation is compressed to ::.

a) IPv4 addressing example: 192.168.0.1/24

$$/24: \quad \underbrace{11111111.11111111.11111111}_{\text{netmask (24 bits)}}.00000000$$

$$192.168.0.1: \quad \underbrace{11000000.10101000.00000000}_{\text{network prefix}}.\underbrace{00000001}_{\text{host identifier}}$$

b) IPv6 addressing example: $2001:0DB8:02DE::0E13/64$

$$/64: \quad \underbrace{FFFF:FFFF:FFFF:FFFF}_{\text{netmask (64 bits)}}:0000:0000:0000:0E13$$

$$2001:0DB8:02DE::0E13: \quad \underbrace{2001:0DB8:02DE:0000}_{\text{network prefix}}:\underbrace{0000:0000:0000:0E13}_{\text{host identifier}}$$

Figure 1.4: IP address structure, for IPv4 and IPv6.

IP addresses are used to identify the source and the destination of packets transmitted in an internetwork. Any interface participating in an IP internetwork has at least one IP address, with the only exception of *unnumbered interfaces*[5].

[5]These are interfaces that participate in point-to-point links, and are allowed to borrow an IP address from other running interface of the same router [48, 116]. In these cases, a packet sent to a shared IP address is delivered to all

In order to prevent confusion with the destination of a packet, the IP address of a given interface, a, needs to be unique among the interfaces that are reachable from the routers that can send packets to interface a. This implies that interfaces reachable through the whole internetwork need IP addresses that are unique in the internetwork – these are called *public IP addresses*. For communication within a single network, interfaces only need IP addresses that are unique (unambiguous) in such network but may be reused by interfaces within other networks – these are called *private IP addresses* in IPv4 [114] and *Unique Local Addresses* (ULA) in IPv6 [49]. The address shown in Figure 1.4.a is an example of a private IPv4 address.

IP addresses play a central role in the transmission of data packets across an internetwork. Routers make forwarding decisions based on such IP addresses, which are included, with some additional information, in the IP header of every packet.

Successful transmission of a packet from the source to the destination may require that several routers forward it. The number of routers involved in the transmission of a packet corresponds to the number of *IP hops* traversed from the source to the destination. The number of hops traversed by a packet within the internetwork is stored in the TTL field (*Time To Live*, called *hop limit* in IPv6) of the IP header, which is decreased every time a router forwards the packet. In order to prevent undeliverable packets to remain indefinitely in the network, a packet is discarded when it has traversed a maximum number of hops without reaching its destination. In IPv4 and IPv6, the maximum TTL value is 255. When a packet can be delivered without being forwarded by any router, the TTL is not decreased and the number of traversed hops is one. In this case, source and destination belong to the same *IP link* (see Figure 1.5).

Definition 1.12 (**IP link**)**.** Two network interfaces are connected to the same *IP link* when they can exchange packets without requiring that any router forwards them, that is, when packets sent from one interface are received in the other with the same TTL value. Then, communication is performed in a single IP hop.

---

the interfaces that use such address – all from the same router.

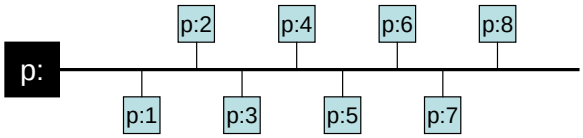


Figure 1.5: An IP link $p:$ with network prefix p. *IP addresses of computers in this IP link have the structure* $p:i/[p]$, *for* $0 < i < 2^{[p]}$.

In terms of IP addressing, interfaces that share a link (def. 1.5) and have IP addresses with the same IP network prefix p belong to the same IP link (def. 1.12), then denoted $p:$. In this case, illustrated in Figure 1.5, an IP link can be unambiguously identified by the set of network interfaces that share the corresponding IP network prefix.

Let $\sim_{IP}$ denote the IP link relationship by which $x \sim_{IP} y$ if and only if network interfaces x and y belong to the same IP link, and let a, b and c be network interfaces. Definition (1.12) implies the following properties of IP links:

- *Symmetry*: $a \sim_{IP} b \iff b \sim_{IP} a$.
- *Transitivity*: $a \sim_{IP} b, b \sim_{IP} c \implies a \sim_{IP} c$.

It also induces a partial order $\subseteq_{IP}$ in the addressing space:

Definition 1.13 (**IP partial order**)**.** Given two IP addresses IPA_1/m_1 and IPA_2/m_2 (m_i being the prefix length of IP address IPA_i), $IPA_1 \subseteq_{IP} IPA_2$ if and only if:

(i) $IPA_1 \otimes NM_{\max\{m_1,m_2\}} = IPA_2 \otimes NM_{\max\{m_1,m_2\}}$

(ii) $m_1 \geq m_2$

where NM_k is the netmask of k bits and $\otimes$ denotes the bitwise AND operation.

- The relationship $\subseteq_{IP}$ satisfies trivially the axioms of partial order:
  - *Reflexivity*: $IP_a \subseteq_{IP} IP_a$.
  - *Antisymmetry*: $IP_a \subseteq_{IP} IP_b, IP_b \subseteq_{IP} IP_a \implies IP_a \sim_{IP} IP_b$, that is, IP_a and IP_b are in the same IP link.

– *Transitivity*: $IP_a \subseteq_{IP} IP_b, IP_b \subseteq_{IP} IP_c \Longrightarrow IP_a \subseteq_{IP} IP_c$.

For routing of packets for which the IP link of the destination is not the same as the IP link of the source, the IP addressing model provides a simple rule for making forwarding decisions. Given the IP address of the packet's destination, a router should forward the packet through the interface providing connection to the IP network closest to the destination, where the notion of *closeness* is as follows:

Definition 1.14 (**IP closeness**). Given an IP address IPA_d/m_d and two IP addresses IPA_1/m_1 and IPA_2/m_2, IPA_1/m_1 is *IP-closer* to IPA_d/m_d than to IPA_2/m_2 if:

- $IPA_d \subseteq_{IP} IPA_1$ and $IPA_d \not\subseteq_{IP} IPA_2$, or
- $IPA_d \subseteq_{IP} IPA_1 \subseteq_{IP} IPA_2$ (which is equivalent to $|m_1| \geq |m_2|$, for the case $IPA_d \subseteq_{IP} IPA_{1,2}$).

According to this decision criterion, routers select send a given IP packet to the interface whose IP address has the *longest prefix match* with respect to the IP packet destination. Such longest prefix match is guaranteed to exist if and only if the router has a *default route* (0.0.0.0/0 in IPv4, :: /0 in IPv6). In case that the router has no routes with a common prefix with the one of the received IP packet, the default route is closer to its destination than any other route.

1.4.2 Network Reference Models

A network model provides a hierarchy of the operations that need to be performed in an internetwork in order to enable communication between interfaces. This hierarchy is based on the level of abstraction of such operations with respect to the transmission/reception of physical signals over a communication channel.

In 1984, ISO[6] proposed Open Systems Interconnections (OSI) as a reference model for internetworks. OSI was based on already implemented network models, one of which was TCP/IP[7], a

[6]International Organization for Standards.

[7]TCP, Transport Control Protocol; IP, Internet Protocol.

network model designed and implemented in the 1970s for the Internet. These two, OSI and TCP/IP, are the two most prominent computer network models, the latter thus being an implementation of the former. Both propose stacks consisting of different layers (seven for OSI and four for TCP/IP [123]), each of them corresponding to a specific type of network operations and data processing. Protocols running in an internetwork are typically placed in one of these layers. Table 1.1 indicates the relationship between layers from both models.

<table>
<tr><th colspan="3">OSI</th><th colspan="2">TCP/IP</th></tr>
<tr><td>7</td><td colspan="2">Application</td><td rowspan="3">Application</td><td rowspan="3">5</td></tr>
<tr><td>6</td><td colspan="2">Presentation</td></tr>
<tr><td>5</td><td colspan="2">Session</td></tr>
<tr><td>4</td><td colspan="2">Transport</td><td>Transport</td><td>4</td></tr>
<tr><td>3</td><td colspan="2">Network</td><td>Internetwork</td><td>3</td></tr>
<tr><td rowspan="2">2</td><td>Data Link</td><td>MAC</td><td rowspan="2">Link</td><td rowspan="2">2</td></tr>
<tr><td></td><td>LLC</td></tr>
<tr><td>1</td><td colspan="2">Physical</td><td colspan="2"></td></tr>
</table>

Table 1.1: The OSI and the TCP/IP network reference models.

The main features of each of these layers can be summarized as follows. For a more detailed layer description, see [70].

- The TCP/IP *Link layer (2)* includes all aspects related to layers 1 and 2 from the OSI model.
  - OSI's *physical layer (1)* concerns the transmission of information (bits) over the physical medium (copper wire, fiber, radio, etc.) with physical signals (such as electromagnetic or acoustic), management of the available bandwidth and modulation/demodulation processes in the transmitter and receiver units of network interfaces.
  - OSI's *data link layer (2)* handles transmission and reception of packets within a link. OSI distinguishes two sublayers: Logical Link Control (LLC), responsible for managing packet formats, and Medium Access Control (MAC), which handles communication rules, channel sensing and access to the medium of network interfaces that share the same physical channel. Protocols at this layer define the network technology, some examples

being IEEE[8] 802.3 (Ethernet), IEEE 802.5 (Token Ring), IEEE 802.11 (Wi-Fi) for local-area networks (LAN), or the Point-to-Point Protocol (PPP), Frame Relay, X.25 and ATM for wide-area networking (WAN).

- The *network* or *Internetwork layer (3)* provides support for packet transmission across an internetwork, regardless of whether network interfaces of sender and destination are on the same or different links, belong to the same network or not. This support includes packet delivery to its final destination, when possible, and packet forwarding by intermediate interfaces. The main example of network protocol is the Internet Protocol (IPv4 and IPv6). Routing protocols for internetworks are also placed in this layer.

- The *Transport layer (4)* is related to end-to-end (or host-to-host) communication features such as reliability, reordering or multiplexing (ports). This involves only the endpoints of a network communication – that is, the sender and the final destination of the corresponding packets. Two main protocols exist: the Transport Control Protocol (TCP), for reliable connection-oriented communication, and the User Data Protocol (UDP), for connectionless and unreliable communication.

- The *Application layer (5)* from TCP/IP (corresponding to layers 5, 6 and 7 of OSI reference model, session, presentation and application, respectively) includes handling the format, semantics and final processing of the exchanged data, which depends on the application that generates and receives it at the endpoints of communication. Protocols at this layer thus handle the interaction between processes of the same application running at different hosts. Examples of applications include Telnet for remote terminal connection, the File Transfer Protocol (FTP), the Domain Name Service (DNS) for mapping domain names to IP addresses, or the Hyper-Text Transfer Protocol (HTTP) for remote access to Web resources.

[8]The Institute for Electrical and Electronics Engineers, `http://www.ieee.org/`.

1.4.3 Routing in the Internet

In the early days of ARPANET[9], the antecessor to the Internet between 1969 and 1990, routing was performed by way of a distance-vector distributed algorithm called the Gateway-to-Gateway Protocol (GGP) [91, 109]. As the size of the network grew large, however, the control traffic required to keep updated routing tables of all computers became excessive and the distance-vector algorithm proved not to scale [127].

In 1982, GGP was abandoned and replaced by a hierarchical routing infrastructure that splits the Internet into different regions, called *Autonomous Systems*. Separation between routing inside and outside an Autonomous System allowed reducing the amount of routing control traffic and to contain the size of routing tables [98], as computers from the same Autonomous System could be treated as a single destination for computers outside the AS. With the partition of the Internet into a set of Autonomous Systems, the concepts of *core* and *edge* of the Internet may be defined in terms of ASes: an AS belongs to the *core* of the Internet if it is able to relay traffic from other ASes to other ASes; otherwise it belongs to the *edge* of the Internet and only handles Internet traffic for which either of the source or the destination is inside that AS.

The definition of Autonomous System has slightly changed over time. In RFC 975, an AS was defined in technical terms, as a set of routers using the same interior routing protocol:

Definition 1.15 (**Autonomous System**, RFC 975, 1986)**.** "An *Autonomous System (AS)* consists of a set of gateways, each of which can reach any other gateway in the same system using paths via gateways only in that system. The gateways of a system cooperatively maintain a routing data base using an interior gateway protocol (IGP)..." [126].

Later, the presence of a single routing protocol was removed as a necessary condition and the concept of Autonomous System was reformulated as follows:

Definition 1.16 (**Autonomous System**, RFC 1930, 1996 & RFC 1812, 1995)**.** "An *Autonomous*

[9]The Advanced Research Projects Agency NETwork. The Advanced Research Projects Agency (ARPA) is an agency of the United States Department of Defense (DoD). Founded in 1958, it was responsible of the ARPANET project that led to the Internet.

System (AS) is a connected group of one or more IP prefixes [internetwork] run by one or more network operators which has a SINGLE and CLEARLY DEFINED routing policy" [111], the term "routing policy" denoting the way that routing information is exchanged between (but not within) Autonomous Systems. In the interior of an AS, "routers may use one or more interior routing protocols, and sometimes several sets of metrics" [116].

Under this latter definition, several routing protocols may coexist in the same Autonomous System as far as, according to RFC 1771 [117], the AS "appears to other ASes to have a single coherent interior routing plan and presents a consistent picture of what networks are reachable through it".

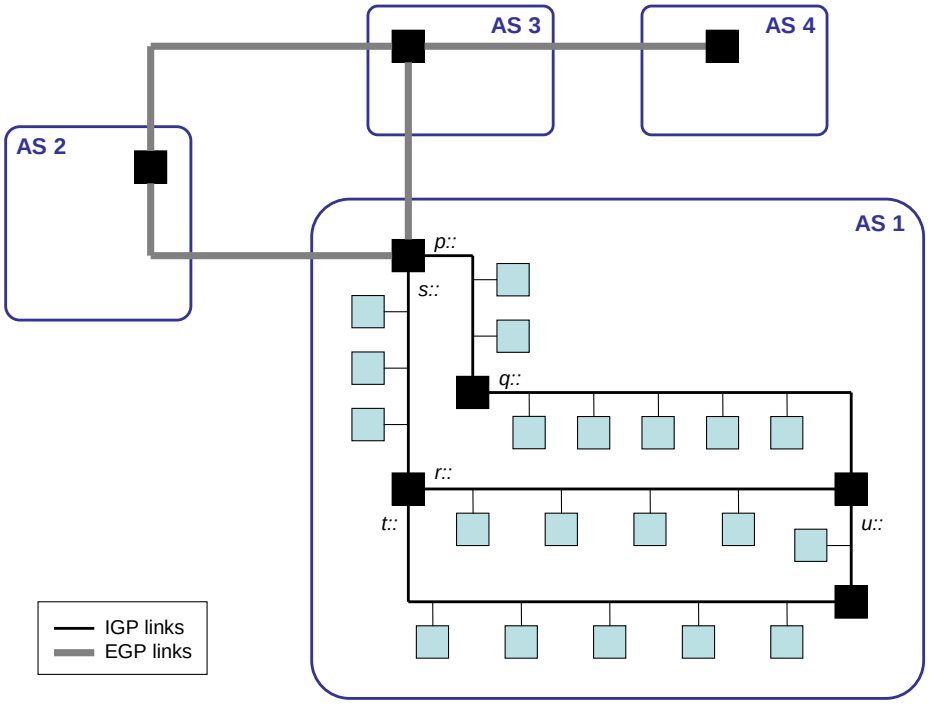


Figure 1.6: Connection of different Autonomous Systems.

Therefore, an AS is an aggregation of computer networks that share a routing policy and behaves itself as a network, in the sense of def. 1.1. Control traffic necessary for route computation within an AS is not flooded outside the corresponding Autonomous System, and neither is the data

traffic sent to a destination in the AS. Links between Autonomous Systems are used for exchanging routing information for computation of inter-AS routes and data traffic for which source and destination belong to different ASes.

The distinction between routing inside an Autonomous System (intra-AS routing) and routing between different ASes (inter-AS routing) leads to two different types of routing protocols:

(i) Interior Gateway Protocols (IGPs), for route discovery and maintenance within an Autonomous System; and

(ii) Exterior Gateway Protocols (EGPs), for route acquisition and information exchange between different Autonomous Systems.

Figure 1.6 illustrates the domain of operation for each of these routing protocol types. The main examples of IGPs are OSPF and IS-IS, both link-state routing protocols; and RIP as a distance-vector protocol. Link-state protocols have displaced distance-vector protocols for routing inside ASes due their better convergence and scalability properties, as mentioned in section 1.3. For inter-AS routing, BGP is the current standard [98].

1.5 Conclusion

The ability of two network interfaces to exchange information through a network depends on the capability of such network to route successfully packets from any of these interfaces to the other. When the source and the destination of a packet are not attached to the same link, packet routing requires that intermediate routers are able to forward packets through the network in a way such that the packet can be delivered to their intended destination. Enabling routers to take such routing decisions is therefore a basic task in a computer network – this task is performed by routing protocols.

In the Internet, interfaces able to communicate directly, without a router's intervention, are part of the same IP link. For communication between different IP links, Internet routing is performed in two hierarchical levels, for scalability reasons. The Internet is split in Autonomous

Systems that may contain several interconnected networks (internetworks). Routing within each AS (intra-AS routing) is performed separately from routing between different ASes (inter-AS routing). The main protocols used for intra-AS routing are link-state protocols, due to the better properties in terms of coverage and scalability of this family of protocols with respect to other available families. The rest of this manuscript explores the use and optimization of existing link-state approaches for routing in the interior of specific types of Autonomous Systems, as it is detailed in further chapters.

Chapter 2

Wireless Computer Networking

The term *wireless communication* refers to communication performed by network interfaces that exchange information by transmitting and receiving electromagnetic signals through the air, rather than through a wire. In this case, the properties of links differ significantly from properties of wired links.

These differences are mainly related to the transmission of signals by propagation in the air of electromagnetic waves, and the physical phenomena (distortion, interference, absorption, reflection) that may affect transmitted packets in the way from the source to the destination(s). Although these physical phenomena are also present in wired networks communication, their impact is not significant in electromagnetic wave propagation through guided media and can therefore be ignored – this is not the case in wireless networks.

2.1 Outline

This chapter elaborates on the physical aspects that affect the quality of wireless communication, that is, the probability that transmitted packets are successfully received by their intended destinations through a wireless network. The focus is on upper layers of communication – in particular, the network layer. Section 2.2 explores the impact of these aspects in properties of links and

networks communicating through wireless media. Section 2.3 describes how issues of wireless communication are addressed by network technologies that provide support for IP networking, paying attention to the particular case of the Wi-Fi technology (IEEE 802.11 family of standards), as it is the most popular wireless network technology used at link layer [46]. Section 2.4 concludes the chapter.

2.2 Wireless Communication

This section provides an overview of the main physical properties of wireless transmission and elaborates on their impact on communication in wireless networks. Section 2.2.1 introduces the frequency and wavelength of electromagnetic signals used for wireless communication. Section 2.2.2 presents the concepts of interface coverage and interference between interfaces. Section 2.2.3 describes the most relevant properties of wireless links. Section 2.2.4 explores communication performed among a set of wireless interfaces, and introduces the concept of semibroadcast communication as a generalization of broadcast to extract the main issues that may arise in wireless networks.

2.2.1 Frequency of Wireless Signals

Wireless signals are electromagnetic microwaves. Their frequency is in the order of GHz, within the UHF/SHF[1] bands. From the relation:

$$c = \lambda f \tag{2.1}$$

where c is Einstein's constant and f is the signal frequency, the wavelength (λ) of wireless signals is in the order of centimeters[2]. The frequency and wavelength of wireless signals determine the propagation properties of such signals. The *Friis' Transmission Equation* models the fraction

[1] *Ultra High* and *Super High Frequencies*, as defined by the International Telecommunication Union (ITU).

[2] $f \sim O(10^9 Hz) \Longrightarrow \lambda \sim O(10^{-1} m)$.

of power that is received by an interface from another interface, depending on the signal wavelength and the distance between interfaces, when transmission occurs in free space:

$$\frac{P_r}{P_t} = G_t G_r \left(\frac{\lambda}{4\pi d}\right)^2 \tag{2.2}$$

where λ is the wavelength, d is the distance between transmitter and receiver, $P_{r/t}$ is the power at the input (or output) of the transmitting (or receiving) antenna, and $G_{r/t}$ is the gain of the transmitting (or receiving) interface antenna, assumed isotropic. The signal wavelength also determines the impact that external conditions may have on signal propagation, as well as the type of obstacles that may cause reflections to signals. Such obstacles are those for which size has a higher or equal order of magnitude than the signal wavelength.

2.2.2 Coverage and Interference in Wireless Interfaces

The region in which interfaces can successfully decode a signal, transmitted by another interface, is the *coverage area* of that interface.

Definition 2.1 (**Coverage area**). Given a wireless interface A, the *coverage area* of A is the geographical region in which packets transmitted by A can be received by other interfaces on the same wireless medium as A, when no other transmission is ongoing. The coverage area of A is denoted by $Cov(A)$.

The coverage area of an interface, and the quality of the signal that may be received by other interfaces within such area, depend on several factors, some of them being:

(i) The physical properties of the transmitting and receiving antennas and of the transmission itself: modulation scheme, transmission power, antenna directivities.

(ii) The physical topology of the coverage area: fading caused by obstacles, reflection and absorption causing multi-path interference and signal loss.

(iii) The characteristics of the wireless medium: signal frequency band, weather conditions or interferences from other interfaces.

Due to the variability of factors having impact on wireless communication, the coverage area of an interface is time-variant. Even within the coverage area at a particular time, when communication is possible, a wireless channel is inherently unreliable and prone to transmission errors and packet losses [47], for instance due to interferences from other interfaces in the network or external sources transmitting in the same frequency band.

Definition 2.2 (**Interference area**). Given a wireless interface A, the *interference area* of A is the geographical region in which interfaces connected to the same wireless medium as a may be unable to receive other packets when there is an ongoing transmission from A. The interference area of A is denoted by $Intf(A)$.

Given a set of wireless interfaces S, the coverage area of an interface is always contained in the interference area of such interface, *i.e.*:

$$Cov(A) \subseteq Intf(A), \forall A \in S$$

This is due to the fact that an interface within the coverage area of another interface a is unable to receive packets from other sources when there is an ongoing transmission from a. The interference area of a may be bigger than its coverage area – that is, some interfaces may be interfered by a's transmissions even when they are not able to receive successfully packets from a [39, 94]. Figure 2.1 illustrates the coverage and interference areas for interface a: interfaces d and f may be unable to decode other transmissions (*e.g.*, d from e) while a is transmitting a packet.

Proposition 2.1 defines the coverage and interference areas under the conditions of the Friis' Transmission Equation (2.2), and shows in particular that the latter is bigger than the former.

Proposition 2.1. *Let a be a wireless interface in a wireless network, in which information propagates under free space conditions. Let P be the power at which all interfaces transmit in the network, and N the noise power, assuming an AWGN[3] model. Let $T > 1$ be the minimum signal-to-interference-and-noise ratio (SINR) for a transmission to be successfully decoded by a. Then, the coverage area of a is a circle centered in*

[3]Additive White Gaussian Noise.

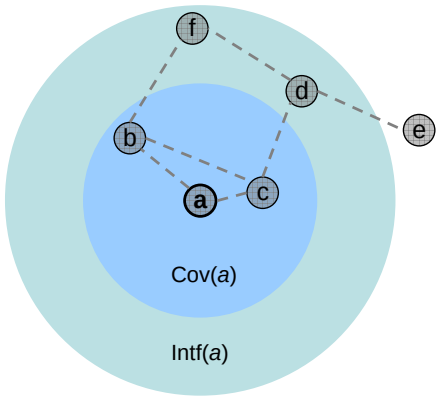


Figure 2.1: Coverage and interference areas of an interface a.

a with radius $r = 4\pi\sqrt{\frac{P}{NT}}$, and the interference area of a is a circle centered in a with radius $r_i = 4\pi\sqrt{\frac{P}{N}}$. As $T > 1$, $r_i > r$.

Proof. The coverage area of a is the geographical region in which the SINR of the received signal is higher than T, in absence of other transmissions (def. 2.1). If not other transmissions occur, there is no interference ($I = 0$), and the SINR for an interface b, at distance d of a becomes the signal-to-noise ratio, SNR of b.

$$SINR|_b = \left.\frac{S}{N+I}\right]_b = [I = 0] = \left.\frac{S}{N}\right]_b = SNR|_b$$

Applying the Friis Transmission Equation (2.2) and assuming unitary gains $G_r, G_t = 1$,

$$\begin{aligned} SNR|_b = \left.\frac{S}{N}\right]_b &> T \\ \frac{P\left(\frac{\lambda}{4\pi d}\right)^2}{N} &> T \\ d < 4\pi\sqrt{\frac{P}{NT}} &= r \end{aligned}$$

When $d < r$, an interface at distance d from a is able to receive packets transmitted from a. For the interference area (def. 2.2), consider the case when an interface b at distance d receives signals from a and from another (neighboring) interface, c, at distance d_o from b. Transmission from a causes interference with a transmission from c, in b, if the SINR at b is lower than T. Considering that the impact of the noise is

negligible with respect to interference ($N \ll I$):

$$SINR_b \approx SIR_b = \frac{S}{I}\Big]_b > T$$
$$\frac{P\left(\frac{\lambda}{4\pi d}\right)^2}{P\left(\frac{\lambda}{4\pi d_o}\right)^2} > T$$
$$d < d_o\sqrt{T}$$

Consider the worst case: distance between b and the main transmitter c is maximum, *i.e.*, $r = 4\pi\sqrt{\frac{P}{NT}}$. Then:

$$d < \frac{\lambda}{4\pi}\sqrt{\frac{P}{N}} = r_i$$

where r_i is the radius of the interference area of a. □

In practice, wireless signals do not propagate in the free space conditions of Friis Transmission Equation [76]. In real conditions, coverage and interference areas are not circular and their evolution cannot be accurately predicted. In consequence, the characteristics of the medium are simplified in approximated models for analysis and simulations. Throughout this manuscript, two models for wireless propagation are used: the Unit Disk Graph (UDG) for theoretical analysis, and the Two-Ray Propagation model, for simulation purposes. Both are presented in Appendix B.

2.2.3 Wireless Links

Wireless links between interfaces are links (in the sense of def. 1.5) that may present the following specific characteristics [23]:

- *Short lifetime* and *time-variant link quality.* The existence of a shared medium in which wireless interfaces may interfere each other, and the variations on the wireless environment (obstacles, reflection and absorption issues, weather conditions), imply that wireless links are likely to have short lifetimes and, even when they are available, that the quality of communication they provide can vary significantly with time.

- *Asymmetry.* A wireless link between two interfaces s and t may be able to handle packet transmissions in one direction (*e.g.*, from s to t, $s \longrightarrow t$) but not in the other (from t to

s). This may be due to different capacities of the two involved interfaces' antennas (different radio coverage may cause that t is included in the coverage area of s but not the inverse, as Figure 2.2.a indicates), different environmental conditions in such two interfaces or the fact that different additional interfaces interfere in the two involved interfaces, as many studies have pointed out [64, 76]. Some of these factors, such as antenna capabilities, cause permanent link asymmetries; others, such as interferences or environmental conditions, may be transient and cause temporary link asymmetries.

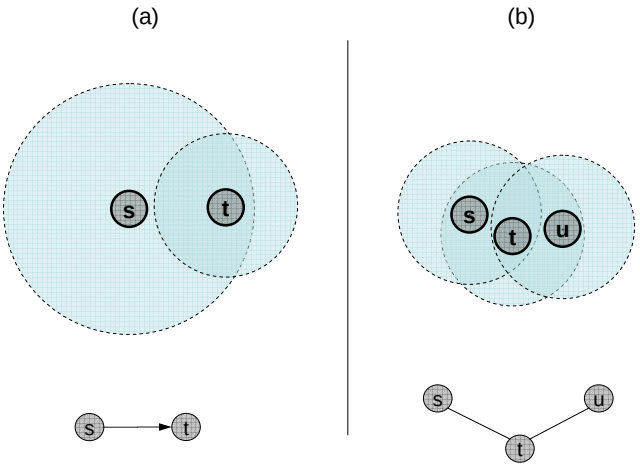


Figure 2.2: Hypergraph (top) and graph (down) representations: **(a)** Asymmetric link between wireless interfaces s and t ($s \longrightarrow t$), **(b)** Non-transitivity of wireless links: existence of a link $s \longleftrightarrow t$ and $u \longleftrightarrow u$ does not imply that s and u can communicate directly.

- *Non-transitivity.* A wireless interface t can exchange packets with another interface if both interfaces belong to the coverage area of each other, *i.e.*, if both are located within the intersection of their coverage areas. Since such intersection is different for each interface to which t can establish bidirectional communication, the fact that t is able to exchange packets with two different interfaces, say s and u, does not imply that such two interfaces s and u receive packets transmitted from each other. Figure 2.2.b illustrates an example of non transitivity: there is no link from s to u (or vice versa) although links $s \longleftrightarrow t$ and $t \longleftrightarrow u$ exist.

For multi-hop wireless networks, non-transitivity of wireless links may cause interfaces on a wireless link to not agree on the neighbors reachable over the link they share. In Figure 2.2.b,

for instance, only node s notices t as an interface participating in its link, while t would consider a link enabling communication to s and t. Given an hypergraph $H = (X, \tilde{E})$ (see section 1.2.2), link conflicts corresponds to the situation in which $e_x, e_y \in \tilde{E}$, $x \in e_y, y \in e_x$, *i.e.* x and y are neighbors, but $e_x \neq e_y$, *i.e.* they have different sets of neighbors, for $x, y \in X$, $e_x, e_y \in \tilde{E}$, $x \in e_y, y \in e_x$.

2.2.4 Semibroadcast Properties of Wireless Communication

Communication in a wireless computer network can be described through the concept of **semibroadcast communication**. This concept generalizes the notion of broadcast communication, which can be described as a particular case of semibroadcast.

Broadcast communication among a set S of network interfaces, is based on the existence of a shared channel, a broadcast link (see def. 1.8) through which all the interfaces in S can transmit and receive packets from/to all other interfaces on the link. In particular, this implies the following properties:

- All pairs of interfaces can communicate directly and bidirectionally, *i.e.*, there exists a symmetric link $i \longleftrightarrow j \ \forall i, j \in S$.
- When an interface in S transmits a packet (i) every other interface in S receive the transmitted packet, and (ii) no other transmission can occur between interfaces in S without interfering with such packet and causing a packet collision.

In order to prevent concurrent packet transmissions over the same channel, interfaces on a broadcast link may implement a **channel sensing** mechanism. With such mechanism, an interface only transmits a packet after sensing the channel and concluding that it is available – no other transmissions are being performed.

Semibroadcast communication describes properties of the communication performed in a wireless computer network among a set of wireless interfaces, and these can be presented by relaxing the characteristics of broadcast communication. Interfaces in semibroadcast communication, or *semibroadcast interfaces*, communicate through a shared medium. As such shared medium does not

need to be the same for all pairs or interfaces, it cannot be assumed that every wireless interface can directly communicate with all other interfaces over the same link [17]. Moreover, as mentioned in section 2.2.3, a wireless link between two interfaces a and b may be asymmetric if a is contained in the coverage area of b but b does not belong to the coverage area of a, or vice versa.

The fact that semibroadcast communication is performed through shared media has two main implications in terms of packet reception and interference. When a wireless interface $i \in S$ transmits a packet, this packet is received by every other wireless interface in S within the coverage area of i. No other packet can be received by these interfaces during the transmission from i. Moreover, interfaces within the interference area of i are unable to receive any packet during the transmission of i, even when they are not able to receive successfully the packet transmitted by i.

It is worth observing that semibroadcast communication among a set of interfaces W becomes a case of broadcast communication when all wireless interfaces belong to the coverage and interference area of any other wireless interface, *i.e.*:

$$j \in C(i) \cap I(i), \forall i, j \in W$$

Packet collisions may occur in a wireless network as a consequence of the described properties of semibroadcast communication. Part of these collisions can be avoided with a channel sensing mechanism. Such a mechanism enables interfaces not to transmit when a neighbor is already transmitting, but do not prevent collisions when they are caused by non-neighboring interfaces. This is the case of *hidden interfaces*. Figure 2.3.c illustrates a case of hidden node problem: nodes s and u are not neighbors (they are hidden to each other), but when they transmit a packet at the same time towards t, there is a collision at t.

Definition 2.3 (**Hidden interface**). A wireless interface i is *hidden* for k when packet transmissions by k are not received and do not interfere at i, but concurrent packet transmissions by i and k interfere with each other and cannot be received by (at least) one common neighbor of i and k, j. In terms of coverage and interference areas, i is *hidden* for k if and only if k does not belong neither to the coverage area nor to the interference area of i, but the intersection of the coverage areas of i

and k contains (at least) one common neighboring interface j.

During the transmission of a packet by an interface, the channel sensing mechanism prevents all neighboring interfaces to transmit concurrently, as additional transmissions would cause packet collisions. Interfaces prevented to transmit are called *exposed interfaces*.

Definition 2.4 (**Exposed interface**). A wireless interface i is *exposed* to another interface j if the fact that j transmits a packet implies that i, after sensing the carrier, decides not to transmit concurrently in order to not interfere with the ongoing transmission of j. In terms of coverage areas, i is exposed to j if i belongs to the coverage area of j and uses a carrier sense mechanism before transmitting packets.

In a semibroadcast communication context, not all the prevented transmissions from exposed interfaces would cause collisions – in particular, if the destinations do not receive several packets at the same time. Figure 2.3.b illustrates the exposed node problem: node u is not able to transmit packets to v when a packet transmission from s to t is ongoing – even when transmission from u to v would not interfere with the one from s to t.

These issues do not appear in broadcast communication: on a broadcast link, all interfaces are directly reachable to each other and therefore there are no hidden interfaces. Moreover, while all interfaces are exposed (in the sense of def. 2.4) to any other interface in the link, there are no prevented transmissions that could be performed without causing a packet collision in the link: the channel sensing mechanism does not produce, in this case, any false positive.

2.3 Wireless Networks under the IP Model

The properties of wireless communications, described in this chapter, show that wireless links cannot be directly identified with IP links, as they were described in def. 1.12. Wireless links cannot be assumed to be transitive nor symmetric. The semibroadcast nature of wireless communication does not correspond with the broadcast assumptions that underlie the definition of an IP link.

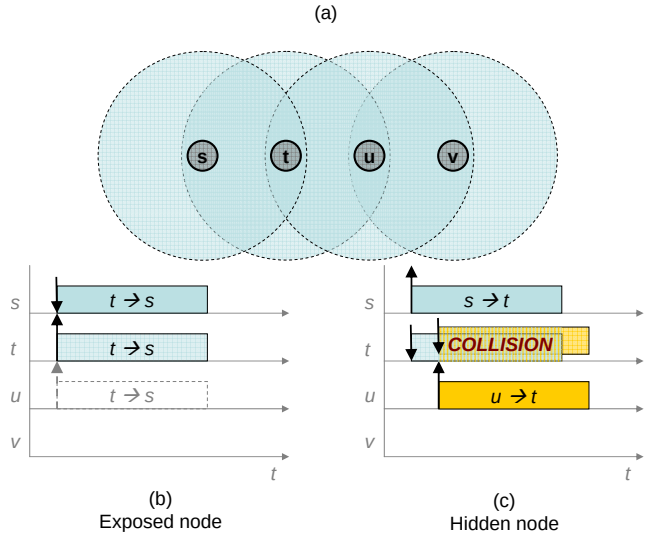


Figure 2.3: **(a)** Multi-hop wireless network with 4 nodes s, t, u and v. **(b)** u is an *exposed node* with respect to the communication from s to t, it would renounce to start a transmission, for instance, to v, even if such transmission would be successful. **(c)** Hidden node problem: t hears s and u, but s is not heard (*hidden*) by u and vice versa, which leads to a collision when both s and t try to transmit packets to t.

Multi-hop wireless communication can support the IP model, under certain conditions. The most obvious way to address such restrictions is to ensure that the shared medium is common to all the interfaces participating in the network. In this case, communication between two interfaces in the network is always performed in a single hop and the wireless channel provides in practice support for a broadcast link, as defined in def. 1.8.

When there are pairs of wireless interfaces that cannot communicate directly, the properties of an IP link can be emulated by introducing a central entity in the network. Such central entity has to enable interfaces in the network to send packets to destination that are not directly reachable. Symmetry and transitivity of communication between wireless interfaces is therefore provided by the central entity.

The way that wireless properties are adapted to the IP model depends on the network technology. The IEEE has specified three families of networking standards, each of them addressed to a different network scope, that support IP networking on wireless deployments:

- 802.11 for Wireless Local Area Networks (WLAN), commercially known as *Wi-Fi*;

- 802.15 for Wireless Personal Area Networks (WPAN), based on the *Bluetooth* and *ZigBee* technologies; and
- 802.16 for Wireless Metropolitan Area Networks (WMAN), also known as *Worldwide Interoperability for Microwave Access* (WiMAX).

WLAN standards are good examples of the two strategies (broadcast communication and IP link emulation through a central entity) that can be employed for adapting wireless communication to the requirements of IP networking. Section 2.3.1 examines the mechanisms specified in IEEE 802.11 link layer standards for establishing IP communication in such networks.

2.3.1 IEEE 802.11

The IEEE 802.11 family of standards provides specifications for physical and link layers of Wireless Local Area Networks (WLAN). Such networks are expected to provide wireless communication among computers located within a reduced (local) coverage area of a few hundreds of meters of radio, typically in indoor scenarios such as an office or a household. They use signals with frequency in the order of GHz, within the unlicensed *Industrial - Scientific - Medical* (ISM) band, which implies that WLANs can be freely deployed without special administrative permissions.

The 802.11 family consists of several physical layer standards. Their main properties and differences are summarized in Appendix C. Beyond the physical layer, however, IEEE 802.11 provides a unified specification for the link layer of a Wireless LAN (WLAN). Such a WLAN is organized in one or more *Basic Service Sets* (BSSes).

Definition 2.5 (**Basic Service Set, BSS**). In the IEEE 802.11 family of protocols, a *Basic Service Set*, *BSS*, is a set of devices that have established a logical association to each other, in order to be able to communicate with all other devices through a wireless medium by means of an IEEE 802.11 protocol. The fact that a device is member of a BSS does not imply, however, that it can establish communication with all other members [37, 46].

IEEE 802.11 supports two modes of BSS operation, illustrated in Figure 2.4. These two

modes use two different ways to perform IP networking over a set of wireless interfaces and overcome the differences between wireless links and IP links.

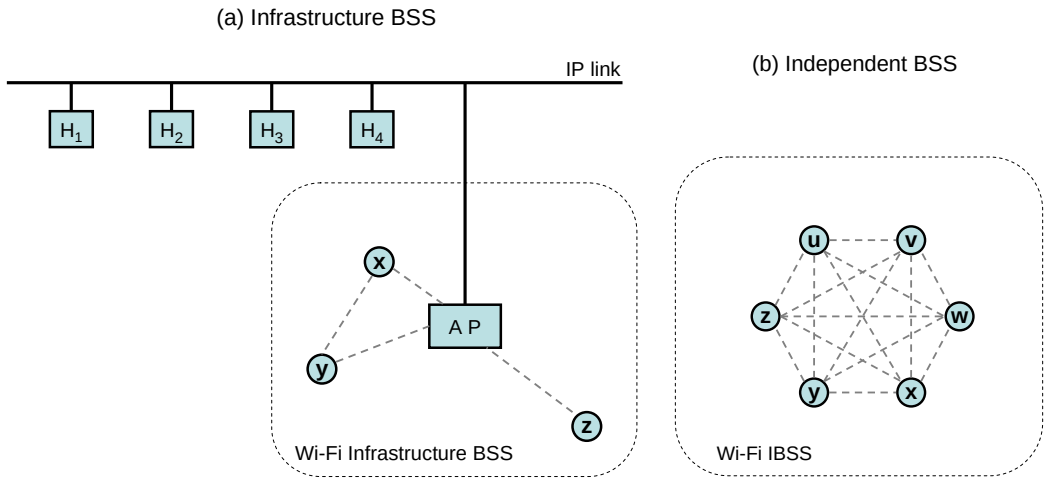


Figure 2.4: **(a)** Infrastructure BSS, **(b)** Independent BSS (IBSS).

- *Infrastructure mode.* Communication among wireless interfaces is managed by a central entity, called an *Access Point* (AP), that needs to be able to directly communicate to all the interfaces participating in the BSS. Such a BSS is called an *infrastructure BSS*, and can be part to a bigger network, as shown in Figure 2.4.

  Communication between interfaces in an *infrastructure BSS* is always performed through the AP. The AP then performs two main tasks: (i) it regulates the access to the channel in the BSS, by allowing and advertising packet transmissions from the interfaces, (ii) it relays packets sent by interfaces in the BSS towards their destination and, in case that the BSS is connected or part of other networks, it relays packets from/to the BSS. This way, the AP avoids semibroadcast issues (hidden node problem, etc.) that are related to the fact that interfaces do not have complete information about the interfaces attached to the wireless network.

  AP operation as a bridge ensures that communication within an *infrastructure BSS* can be configured as (part of) an IP link (def. 1.12); interfaces from the BSS can communicate with each other symmetrically, through the AP, and such ability is transitive. Consequently,

interfaces in an infrastructure BSS are able to acquire their IP addresses through stateful mechanisms such as DHCP[4], from the router responsible for the corresponding IP link.

- *Ad hoc mode.* Different wireless interfaces may establish direct communication on their own, forming an *independent BSS* (IBSS). No central entity is present for coordinating communication or handling IP addresses. Link-local IP addresses are chosen by the interfaces themselves and without negotiation, following the IPv4 or IPv6 autoconfiguration mechanisms[5].

  Since these autoconfiguration mechanisms assume that the interfaces share the same IP prefix, successful operation on this mode is only possible when all the participating interfaces can receive packets from each other.

The coverage area provided by a single BSS can be extended and increased by using several mechanisms (coordination of multiple BSS, bridges, etc.). For a more detailed description of these mechanisms, see [46].

2.4 Conclusion

The use of wireless communication has made possible that computer networks are deployed and provide computer communication facilities in environments in which wired networking was not available, not possible or too expensive to be taken into consideration.

However, communication between wireless interfaces yields some issues that need to be addressed in the framework of the Internet. Wireless links are unreliable and prone to errors, their quality is time-variant, they may be asymmetric and are not necessarily transitive. In these conditions, communication among wireless interfaces presents a set of characteristics –often described as *semibroadcast* characteristics– that can be seen as a generalization (in the sense of loosening) of the broadcast properties. Wireless interfaces communicate through a shared medium (the air) that

[4]Dynamic Host Configuration Protocol, specified in RFC 2131 for IPv4 [108] and RFC 3315 for IPv6 [77].

[5]For IPv4, link-local addresses are selected within the prefix `169.254.0.0/16` (RFC 3927 [57]). For IPv6, the Stateless Address Autoconfiguration (SLAAC) mechanism provides each interface with a link-local address belonging to the prefix `FE80::00/10` (RFC 4862 [35]).

can be common to other interfaces, but such medium is not necessarily the same for every pair of communicating interfaces in a wireless network, nor for every neighbor of a particular wireless interface.

Therefore, a wireless network cannot be always configured as a single IP link, as the two main properties of IP links (see def. 1.12), symmetry and transitivity in communication, cannot be ensured for interfaces participating in a wireless network. IP networking is however possible over those wireless networks in which any interface can directly communicate with every other, in a single hop – that is, when semibroadcast communication becomes broadcast communication. For networks not satisfying this property, symmetry and transitivity can be emulated by adding a central entity such that (i) any wireless interface can communicate with it, and (ii) communication between interfaces in the network is always performed through such central entity. Operation of such central entity enables the network to be configured as a single IP link.

Each of these alternatives are used in standard IP networking mechanisms for wireless networks. When none of them are available –because broadcast conditions are not fulfilled and a central entity cannot be used–, these mechanisms are not sufficient and additional elaboration is required. The following chapters present and explore these cases, that correspond to ad hoc multi-hop wireless networks.

Chapter 3

Communication in Ad hoc Networks and Compound ASes

Ad hoc networks are wireless networks. As such, they present the characteristics that were described in chapter 2 – semibroadcast communication, shared medium, unreliable wireless channel. Ad hoc networking implies some additional restrictions and issues, in particular related to the absence of available networking infrastructure and the dynamism of network topology. These conditions exclude the solutions presented in chapter 2 (full connectivity and the presence of a central authority), as such solutions rely on assumptions that are explicitly discarded in ad hoc networks. Communication within a multi-hop wireless ad hoc network, in particular communication between non-neighboring computers in such networks, needs thus to be performed by way of alternative mechanisms, in particular routing.

3.1 Outline

This chapter addresses the needs of communication in multi-hop networks in which topology is dynamic and there is no available infrastructure (connecting wires, central control), by defining and exploring the concepts of Mobile Ad hoc Networks and compound Autonomous Systems. It

describes the assumptions that underlie ad hoc networking and explores the implications of such assumptions in the architecture of ad hoc networks and internetworks that contain ad hoc networks. Section 3.2 presents the notion of ad hoc networks and generalizes this to the broader notion of compound Autonomous System, in which ad hoc and fixed networks may coexist. This section also presents some applications of ad hoc networks and compound ASes. Section 3.4 examines the basic mechanisms that can be used for enabling communication in such network – neighbor sensing for direct communication and routing for indirect communication. Section 3.3 describes the most significant properties of routers and links that form an ad hoc network. Section 3.5 concludes the chapter.

3.2 Ad hoc Networks and Compound ASes

Ad hoc networking has to accommodate the fact that the typical assumptions on which computer communication relies in traditional (wired) networks cannot be taken for granted. In that sense, more than describing a well-defined set of properties or features, the concept of ad hoc network provides an abstract definition that holds for a wide range of network types, all sharing a certain degree of flexibility and ability to operate without relying on an established infrastructure. This section presents the main use cases of ad hoc networks and discusses integration of ad hoc networking in the Internet architecture, by way of the notion of compound Autonomous Systems.

Section 3.2.1 describes the main constraints of ad hoc networking, the implications that such constraints has in the operation of ad hoc networks and some examples of use of ad hoc networking. Section 3.2.2 introduces the concept of compound Autonomous System for addressing the coexistence of ad hoc networks and fixed networks in the same internetwork.

3.2.1 Ad hoc Networks and Applications

The MANET working group of the IETF has defined a (mobile) ad hoc network as follows:

Definition 3.1 (**(Mobile) Ad hoc network**). A *(mobile) ad hoc network* is “an autonomous

system of routers (and associated hosts) connected by wireless links, either mobile or static, the union of which form an arbitrary graph", and in which "routers are free to move randomly and organize themselves arbitrarily". In such a network, routers "form a dynamic topology which may change unpredictably and rapidly", and are connected via wireless "links" – presenting characteristics uncommon to IP networks [89].

Perkins [92] identifies the main characteristics of ad hoc networks and the requirements they impose for establishing communication within an ad hoc network. That topology in an ad hoc network is arbitrary and may change unpredictably implies that the communication cannot be based on network or user configuration prior to network operation. Rather, nodes are expected to dynamically learn their neighborhood and detect changes in the topology. As direct communication cannot be assumed between every pair of nodes (that is, in a single hop), ad hoc networking mechanisms need to provide support for multi-hop communication. Since communication in ad hoc networks does not rely on any planned infrastructure, establishment and maintenance of communication within the network is achieved through dynamic self-organization and cooperation between ad hoc nodes.

From def. 3.1, nodes in an ad hoc network communicate through wireless links, and therefore ad hoc networking is a particular case of wireless computer networking. Links between ad hoc nodes have the same basic properties, with the additional considerations further described in section 3.3, as wireless links presented in chapter 2. The mechanisms detailed in such chapter, however, cannot be used to establish or maintain IP communication in multi-hop ad hoc networks. These mechanisms were based on the assumptions that (i) direct communication was possible between any pair of nodes, or (ii) there was a centralized access point able to emulate in layer 2 the characteristics required for IP networking in layer 3. As neither of these assumptions hold for multi-hop ad hoc networks, additional mechanisms are required for enabling communication in such networks.

The properties and requirements of ad hoc networking can be found to some extent in a number of different applications. Some of the most relevant are networks for military or emergency recovery purposes, wireless sensor networks (WSNs) or Vehicular Ad hoc Networks (VANET), each

with specific requirements.

Military and recovery ad hoc networks

Communication needs for military units (vehicles and human units) when deployed in the battlefield is the classic example of Mobile Ad hoc Networks (MANETs) [93]: infrastructure is often not available (either because it has been destroyed, because it is controlled by the enemy or because it cannot be assumed to be reliable) and so military units must rely only on themselves to establish ad hoc communication.

A similar situation, in terms of unavailability of local communication infrastructure, is in disaster scenarios, such as those affected by terrorist attacks, earthquakes or other natural catastrophes. In these cases, rescue operations may benefit from Mobile Ad hoc Networks rapidly deployed in the affected area. In both military and recovery situations, networking devices are not limited by energy or computational restrictions, and the network does not need to cope with high relative speed between nodes. The main target of ad hoc networks in these deployments is to be able to establish communication, without significant set-up delays nor human intervention.

Wireless Sensor Networks (WSN)

WSNs are collections of sensors intended to measure one or several properties of the environment in which they are deployed. Communication facilities required by such networks need to include, at least, the transmission of collected information from the sensors to a gateway or central server that stores and eventually process it, and the transmission of information (*e.g.*, configuration instructions) from the server to one or more sensors.

There is a broad range of information that may be collected and exchanged through WSNs, some examples including climate studies, bird observation, power monitoring in buildings or tracking of patients' health parameters with body sensors. Properties of a WSN may vary depending on the purposes of the sensor deployment. [61] presents a detailed overview of WSN applications and characteristics.

Despite such variability, there are some properties that are typically related to networks of this kind: sensor deployments form ad hoc networks, in which topology cannot be predicted *a priori*, even when sensors are not supposed to move. The communication pattern, in contrast, is somewhat more predictable: as mentioned, it usually involves transmission from the sensors to the server or from the server to the sensors. That implies that sensor-to-sensor communication is required for multi-hop WSNs. Moreover, sensors are often battery driven, thus one boundary of the lifetime of a sensor is its battery lifetime. Protocols for enabling communication within WSNs must therefore be designed with energy consumption [16, 26] and energy-efficiency in mind [63].

Vehicular Ad Hoc Networks (VANET)

Vehicular Ad Hoc Networks are those networks designed to enable communication from and towards vehicles (cars) while they are moving, for example, for distributing information along the highway about traffic-related events – *e.g.*, jams or accidents [5]. Communication between vehicles and fixed stations placed along the road might be used for distributing information about weather conditions, highway restrictions (speed, works, etc.) or services available in the area (oil stations, hostels, hospitals and such), but also for medical or police assistance calls from vehicles.

The speed of vehicles in highways is expected[1] to be below $130\frac{km}{h}$. Relative speed of a vehicle with another vehicle varies from values close to zero, for vehicles in the same lane or direction, to values up to double of the maximum speed limit, for relative speed between vehicles moving in opposite lanes. Vehicular networks need thus to be able to cope with high mobility scenarios. Significant delays are not acceptable while establishing communication, as the topology may change and reachability between intended source and destination may be affected in the meanwhile.

Devices participating in vehicular networks (either inside of vehicles or in roadside equipment units) have neither significant energy constraints nor severe computational limitations. However, the private character of nodes (vehicles) in a vehicular network, which correspond to independent and unrelated users, reduces their willingness to cooperate on supporting or enabling commu-

[1] In United States and Europe. For US, maximum speed limits are below $75mph = 120.7\frac{km}{h}$ [2]. For countries from the European Union, maximum speed limits are below or equal to $130\frac{km}{h}$ [7].

nication between other nodes. Protocols for enabling communication within VANETs are therefore oriented towards (i) minimization of own resources dedicated to others' communication, and (ii) immediate availability of communication with other vehicles or with roadside equipment units.

3.2.2 Compound Autonomous Systems

An ad hoc network enables communication among its attached computers. For enabling information exchange with computers beyond such ad hoc network, it needs to be part of a larger internetwork – an Autonomous System, in case interaction with the Internet is targeted. This section addresses the cases of Autonomous Systems that combine ad hoc and fixed networks, and enable communication from and towards computers inside by way of a single routing protocol.

Definition 3.2 (**Compound Autonomous System**). A *compound Autonomous System* is an AS in which ad hoc networks coexist with fixed networks. Routers that are able to participate both in ad hoc and fixed networks are denominated *hybrid routers*.

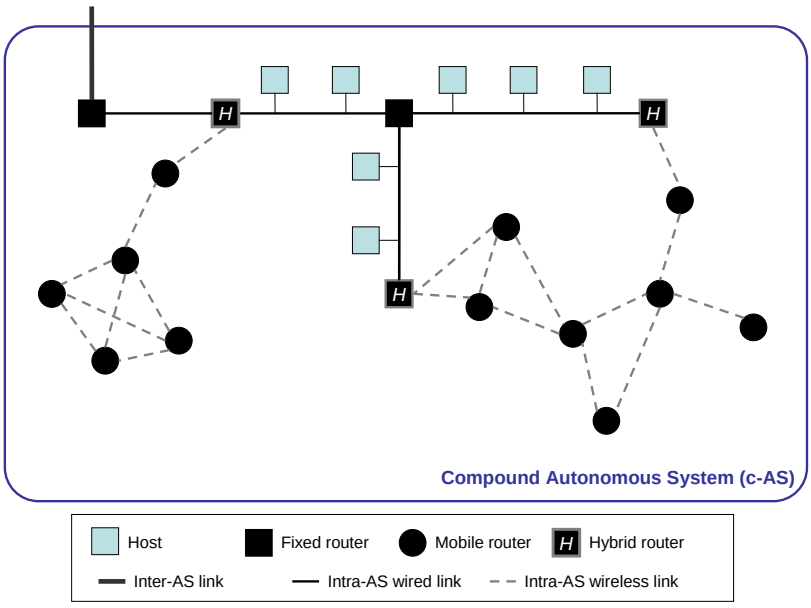


Figure 3.1: Compound Autonomous System.

Figure 3.1 shows an example of a compound AS. This definition allows the presence of

fixed networks and ad hoc networks in the same AS. The network is therefore composed of a set of heterogeneous links, with different stability and reliability patterns. This manuscript concentrates on compound ASes in which a single protocol is used for routing inside. Interconnection of both types of networks (ad hoc and fixed) is provided by *hybrid* routers, each of them maintaining interfaces attached to fixed and ad hoc networks of the AS.

Access to the Internet appears as a desirable feature in some of the most significant applications of (mobile) ad hoc networking: *e.g.*, sensor networks connected through a common networking infrastructure able to process the data of the different testbeds, and possibly compare them or make them available through the Internet; or vehicles interacting with the fixed roadside equipment units, which in turn are able to relay information from remote networks. In these cases, compound ASes can be useful for addressing not only communication within the ad hoc network, but also information exchange between separate ad hoc deployments (see wireless networks belonging to the compound AS of Figure 3.1) and interaction with Internet resources. Moreover, the development of pervasive computing and the increasing role of wireless ad hoc and sensor networks in such pervasive deployments open new scenarios in which fixed and ad hoc networks may need to be treated as single networking entities. For such new scenarios, the concept of compound AS may also be appropriate.

3.3 Nodes, Links and Addresses in Ad hoc Networks

The characteristics of ad hoc networking impose some conditions on nodes that participate. As the topology is dynamic, and as no central entity can be assumed to be available for providing routes, all nodes need to be able to act as routers and thus cooperate in forwarding others' traffic over the network. Throughout this manuscript, the term *router* will be used as an equivalent to *node* of an ad hoc network, given that hosts cannot participate directly as such in ad hoc networking. Indeed, hosts are connected to a router (*e.g.*, through an IP link) that acquires route information from the network and enables thus interaction with the rest of nodes of the network. Figure 3.2 illustrates such model for MANET nodes.

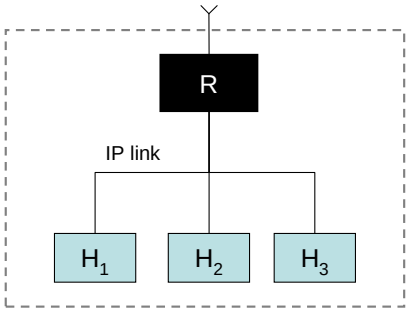


Figure 3.2: Model for a MANET node.

Links in ad hoc networks are wireless links, and therefore show the main characteristics described in section 2.2. Due to relative mobility between routers, links in a MANET may be even less stable than links in a wireless non-mobile ad hoc network, in which links only vary as a result of time-variant wireless channel conditions.

Configuration of ad hoc links and networks in accordance to the IP model is not straightforward. The existence of a link in an ad hoc network between two interfaces (see definition 1.5) does not imply that there is an IP link (see definition 1.12) between these interfaces, in particular because (i) IP links are transitive whereas links between wireless interfaces are not in general, as stated in chapter 2, and (ii) IP links are stable during the lifetime of the involved interfaces, whereas wireless links between interfaces in an ad hoc network may appear and disappear dynamically several times in the lifetime of the involved interfaces.

As communication between interfaces of a (mobile) ad hoc network cannot be assumed to be stable or transitive, no IP links should be set between routers in such networks. In particular, IP addressing in an ad hoc network should not make assumptions about IP connectivity between wireless interfaces, even when interfaces can communicate directly (that is, there is a link between them) at a particular time [17].

From def. 1.12, there is a IP link between two interfaces when there is a link between them and both interfaces have IP addresses with the same network prefix. Therefore, in order to prevent assumptions about IP links in an ad hoc network, wireless interfaces should be configured in a way

such that their IP addresses do not share network prefixes. Moreover, as links and neighborhood relationships cannot be predicted and may vary during the network lifetime, the network layer address that an interface uses for interacting with interfaces in its coverage area must be unique in the whole internetwork. Absent this uniqueness, address collisions may happen – *i.e.*, two interfaces with the same IP address might find themselves in the same link at some point.

The IETF has proposed an IP addressing model for ad hoc networks [9] that addresses these issues and tries to avoid the connectivity implications of IP addresses by recommending the use of maximum-length prefixes (/32 for IPv4, /128 for IPv6) and discouraging the use of link-local addresses for autoconfiguration purposes, as such addresses cannot guarantee uniqueness beyond the link in which they are generated. The use of maximum-length unique prefixes also prevents the formation of IP links in ad hoc networks.

Properties of IP links (stability, transitivity) do not correspond to those of links between ad hoc routers, but IP links can be configured between hosts and the routers through which they interact with the ad hoc network [39]. In Figure 3.2, the link between hosts H_1, H_2, H_3 and router R, inside the MANET node, can be configured as an IP link.

3.4 Single and Multi-Hop Communication

In a multi-hop ad hoc network, a router is able to directly communicate with a subset of the other routers in the network – these are the *neighbors* of the router. For enabling communication with other routers in the network, a routing mechanism is needed. Discovery and maintenance of the neighbors of a router, although not always required for performing routing[2], is often used to perform routing.

[2]Reactive protocols such as DSR (Dynamic Source Routing protocol, specified in RFC 4728 [38]) are able to obtain routes on-demand only relying on broadcast mechanisms.

3.4.1 Neighbor Sensing

A router communicates with the rest of an ad hoc network through its neighbors. Since the set of neighbors of a node is not necessarily stable, cannot be predicted, and may change dynamically, a router needs to be able to dynamically detect its neighbors and identify those with which a bidirectional communication can be established. These tasks are denominated *neighbor sensing*.

The most widespread and basic mechanism for neighbor sensing consists of periodic transmission of Hello packets by every router in the network. Hellos enable the routers that receive them to identify those other routers in the network that have a link towards itself. If the Hello contains information not only about its source, but also about the routers from which the source has received Hellos, the exchange of Hello packets enables routers in a network to identify bidirectional neighbors – that is, routers with which communication is possible in both senses. Figure 3.3 illustrates the process through which two routers (A and B) learn their ability to exchange information, if each router advertises its neighbors in Hello packets. Hello exchange for neighbor sensing purposes was first described as part of the routing protocol OSPFv2 [107].

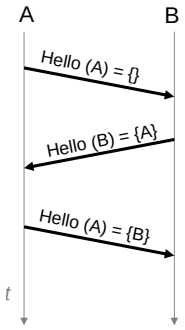


Figure 3.3: Establishment of bidirectional communication in 3 steps, through Hello exchange.

Periodic Hello exchange also enables routers to detect whether a neighboring router is no longer a neighbor. After having established bidirectional communication through the process displayed in Figure 3.3, a router detects that such bidirectional communication is not available when Hello packets stops being received from the former neighbor. In such cases, the first router

declares the second to be *dead*.

Definition 3.3 (**Dead neighbor**). A router declares a neighbor to be *dead*, and removes it from the list of neighbors, when no Hello packets are received during a certain period of time. This implies that bidirectional communication with such router is no longer possible. Typically, this period is configured as a multiple of the interval between periodic Hello transmissions.

As packets related to neighbor sensing are not forwarded, Hello traffic is not generally significant with respect to the overall traffic, that is, the sum of user data traffic and network-wide control traffic required for delivering it. The role of the Hello protocol is however essential; not only because it enables routers to identify their neighbors, but also because Hello exchange may be useful for acquiring additional information about such neighbors (geographic position, remaining battery power, willingness to accept responsibilities in communication), the links to them (link quality measures) or the neighbors of such neighbors (2-hop neighborhood acquisition).

Analysis, improvements and optimizations of periodic Hello protocol have been performed and discussed for ad hoc networks in [33]; from a mostly theoretical perspective in [56]; for a simulation-based approach and evaluation in [80], that focuses on the optimal Hello interval in OSPF; and in [86], which analyzes the impact of the interval between periodic Hello transmissions in AODV on the quality of communication with described neighbors. [80] highlights the importance of the expected network congestion in the choice of an optimal Hello interval. [86], in turn, concluded that Hello packets should be as similar as possible (in terms of size and processing) to the packets forming user data traffic intended to be exchanged, in order to optimize the quality of the links towards the set of maintained neighbors.

3.4.2 Routing in Ad hoc Networks and Compound ASes

Several routing protocols have been proposed for ad hoc operation, some examples being DSR [38], the Topology Dissemination Based on Reverse-Path Forwarding protocol (TBRPF) [62] or AntSens [6]. As mentioned in section 1.3, there are two main approaches for routing: table-driven or proactive protocols, and on-demand or reactive protocols. The two most prominent protocols

for routing in mobile ad hoc networks, standardized by the IETF, are the Optimized Link State Routing protocol (OLSR, first version specified in RFC 3626 [71], OLSRv2 core operation specified in [18]), proactive; and the Ad-hoc On-demand Distance Vector protocol (AODV, specified in RFC 3561 [75]), reactive. This section overviews the basics of the operation of these two protocols.

Optimized Link State Routing – OLSR

OLSR is a link state protocol that uses Multi-Point Relays (MPR) for distributing topology information in the network. A router in OLSR selects a set of MPRs from among its neighbors, in a way such that every 2-hop neighbor is covered by at least one MPR[3]. The selection relies on the neighborhood information acquired by way of Hello packets exchange.

Routers that have been selected as MPRs over the network advertize the links they maintain to their MPR selectors, and periodically broadcast this information over the network in Topology Control (TC) packets. Such TCs are forwarded by the MPRs of the source, and then iteratively by the MPRs of the forwarders until they reach every router in the network. The set of TCs received from every other router in the network enables the receiving router to acquire and maintain information about the network topology, and to compute shortest paths based on this information.

More details on the architecture of link-state routing protocols can be found in Part II.

Ad-hoc On-demand Distance Vector – AODV

As a reactive protocol, AODV enables routers to acquire routes to a destination when they need to forward packets towards that destination and when there are no routes locally stored. In such case, the router broadcasts a request (RREQ). When receiving a RREQ, a router may (i) reply to the request by sending a unicast reply (RREP) back to the request source, if it is the requested destination or it maintains a route towards it; or (ii) otherwise forward the request.

Routers that forward requests store the neighboring router from which they received the request, in order to be able to send back a reply, in case that such reply is received. The reply to

[3]See chapter 8 for further details on MPR.

a request advertises the distance (in hops) to the destination from the replying router, and such distance is updated in every intermediate router in the way back towards the request source. This way, the source is able to identify the next hop and the total distance towards the destination.

Considerations on Routing in Compound ASes

Literature abounds with analysis and performance evaluation of the different routing approaches for MANETs [36, 87, 103, 104]. In such networks, proactive link-state protocols such as OLSR show better routing quality (in terms of data delivery ratio and packet delay) than reactive protocols [36, 103], at the expense of requiring a constant amount of control traffic. In proactive protocols, such control traffic does not depend on network mobility or data traffic patterns, as is the case in reactive routing, and in AODV in particular [87].

Routing in Autonomous Systems that include ad hoc and fixed networks yields issues other than those that arise for routing in isolated MANETs, in particular related to the establishment and maintenance of routes between ad hoc and fixed networks. One solution for routing in such case is to split the Autonomous System in different *routing domains*, in a way such that networks inside a single routing domain are either all fixed or ad hoc, but there is no coexistence between ad hoc and fixed networks within a routing domain.

Definition 3.4 (**Routing domain**)**.** In an Autonomous System, a *routing domain* is a set of interconnected networks, or internetwork, in which routers use the same routing protocol instance.

By splitting an AS in multiple routing domains, different routing protocols, maybe several instances of each, run independently in the AS. For instance, OSPF [107] may be used in fixed networks while OLSR [71] is used in ad hoc networks. Figure 3.4 illustrates, over the AS of Figure 3.1, a configuration of three routing domains, A, B and C. A and C are ad hoc networks, and may use different instances of OLSR; and B is a fixed internetwork that may use a single instance of OSPF. Different routing domains interact through specific routers denominated *gateways* (denoted by G in Figure 3.4).

Definition 3.5 (**Gateway**)**.** Throughout this manuscript, a *gateway* in an internetwork (in par-

ticular, in an Autonomous System) with several routing domains denotes a router able to run simultaneously different instances of different routing protocols, and thus enables the exchange of routing information between different routing domains in the internetwork.

The fact that such gateways are able to exchange routing information from different protocols and between different domains, enables them to ensure communication between any pair of computers in the AS.

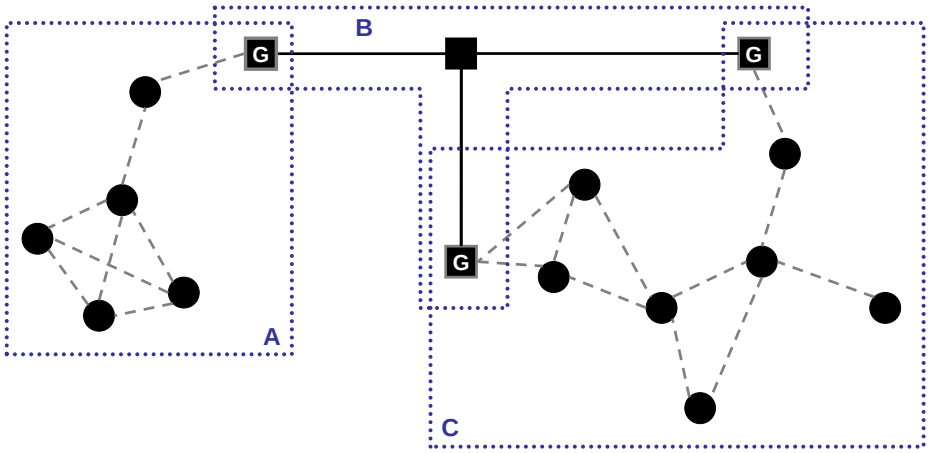


Figure 3.4: An Autonomous System composed of different routing domains: domains A and C correspond to ad hoc networks, and B corresponds to a fixed network.

The use of different protocols is however suboptimal in several ways: it may lead to suboptimal paths between different networks of the AS, through a single gateway – and this even in cases where more diverse connectivity might be leveraged, and the network may benefit from traffic engineering. Figure 3.5 illustrates a simple case in which communication between two computers is performed through a suboptimal path due to the fact that they are in different routing domains. When host H_1 sends a packet to the ad hoc router r_5, router R_2 forwards it towards its default gateway for external destinations – which is R_1. The packet then may follow the locally optimal path in the ad hoc network $\{R_1, r_1, r_3, r_5\}$. From the perspective of the whole AS, however, path $\{R_2, R_3, R_4, r_5\}$ is shorter (in terms of hops) than $\{R_2, R_1, r_1, r_3, r_5\}$.

Moreover, familiarity with a single protocol is an advantage – training engineers to operate and maintain an additional routing protocol is costly both from an economic and a time perspec-

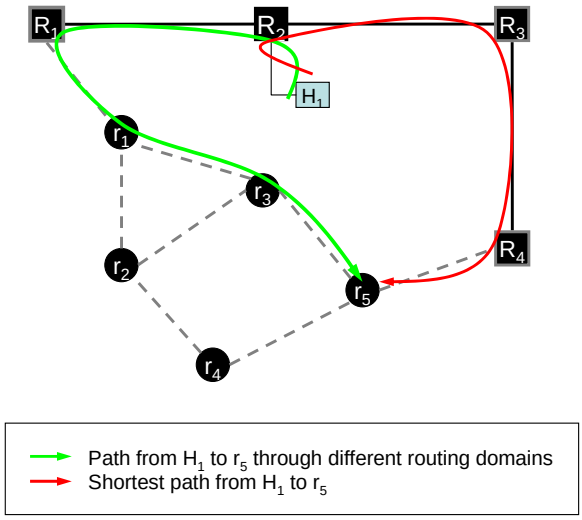


Figure 3.5: Path suboptimality due to the presence of several routing domains in the same AS.

tive. As gateways require a more specialized hardware and software than the rest of routers, the coexistence of different routing protocols in the same AS becomes also more expensive than the use of a single protocol, for which no gateways are needed.

3.5 Conclusion

Multi-hop wireless ad hoc networking is useful for a growing number of networking applications. The main issue with such networks is that their dynamic, non-planned characteristics, as well as the lack of central control, cannot be addressed within the IP networking model with the techniques used in ordinary networks. Mechanisms described in chapter 2 for enabling wireless communication by configuring or emulating IP links, in particular, cannot be applied to multi-hop ad hoc networks.

Topology dynamism has significant implications for the nodes and the links of ad hoc networks. In the absence of any pre-planned routing infrastructure or central entity, nodes have to be able to assume router and host roles simultaneously. The interaction of a router with its neighbors can be handled through a dynamic neighbor sensing via Hello message exchange. Such

neighbors may change frequently during the network lifetime, and therefore IP links should not be configured in these networks. For enabling such router to take valid forwarding decisions, different distributed routing protocols could be implemented in MANETs: the most prominent ones are OLSR, a proactive link-state routing protocol, and AODV, a reactive protocol.

Compound ASes generalize the notion of ad hoc networking in an Autonomous System in which ad hoc networks may coexist with fixed infrastructure routers. Such situation correspond to interesting applications, mostly related to ad hoc networks in which routers are expected not only to communicate among themselves, but also to exchange information with devices outside the ad hoc deployment, for instance reachable through the Internet. To provide communication and perform routing in such compound scenarios additional issues arise besides those specific of ad hoc properties. While ad hoc and fixed networks in a compound AS may be in principle handled through instances of different routing protocols, this solution has severe drawbacks related to the suboptimality of the routes and the computational cost of the inter-protocols routing information exchange in those nodes participating in both protocols. Instead, this manuscript explores the extension of existing and well-known link-state protocols, already used for routing in Autonomous Systems, for operation in wireless (mobile) ad hoc networks. Such extension has the major advantage of enabling compound ASes to run a single routing protocol able to deal efficiently both with their attached fixed and ad hoc networks, without requiring the use in the AS of specialized hardware (*gateways*) or software (MANET-specific routing protocols).

Part II

LINK-STATE ROUTING IN AD HOC NETWORKS

Chapter 4

Elements of Link State Routing

This manuscript investigates the use of a single link-state protocol for routing inside compound Autonomous Systems (ASes). As mentioned in chapter 1, link-state routing assumes that routers collect information from the network about the network topology, and base their forwarding decisions on such information. This chapter analyzes link-state routing, describes different mechanisms for performing link-state routing in ad hoc networks and discusses challenges that arise in such networks.

4.1 Outline

Section 4.2 describes how link-state routers construct and maintain routing tables based on the information they have about the network topology. Section 4.3 presents the mechanisms that enable such routers to acquire and update this topology information. Section 4.4 presents some of the most significant issues that are present for link-state routing in ad hoc networks, and identifies techniques to address these issues or minimize their impact in the routing performance. Finally, section 4.5 concludes the chapter.

4.2 The Link State Database

Routers using a link-state protocol are able to forward packets to any possible destination in a network at any time, and rely on the information they have about the network topology for taking forwarding decisions. Topology information is stored in the *Link-State Database* (LSDB).

Definition 4.1 (**Link State Database**). The *link-state database* (LSDB) of a network is a database that describes the network topology by way of the following elements:

(i) the set of routers in the network,

(ii) a set of links between routers in the network, and

(iii) the cost of links, according to the metric in use.

These elements enable the router to reconstruct the network graph. Every link-state router maintains a local instance of the distributed LSDB.

Routers compute paths from themselves to every other router in the network by executing Dijkstra's shortest-path algorithm [135] over the network graph based on the topology information from the LSDB. The output of Dijkstra's algorithm is a Shortest Path Tree (SPT) of the computing router. Based on the Shortest Path Tree, a router builds its routing table and is thus able to forward packets to its next hop in the shortest path towards their final destination. Construction of routing tables based on the Link-State Database is illustrated in Figure 4.1.

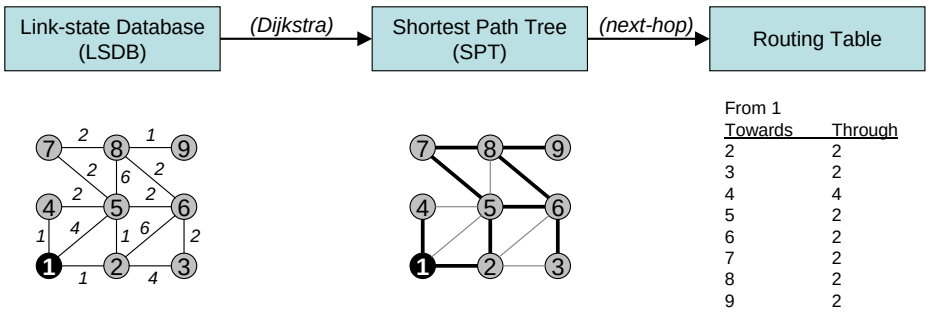


Figure 4.1: Construction of the routing table from the network graph indicated in the LSDB, with a network example.

4.3 Topology Acquisition

In order to ensure that routers in the network acquire topology information describing the network and update accordingly their instances of the LSDB, each router creates link-state advertisements (LSAs). Each LSA describes links local to the originating router, and is flooded through the network. The local instance of the LSDB maintained by a router, therefore, is the aggregation of link-state advertisements received by that router from the rest of the network. Link-state protocols ensure that such advertisements are received by all routers in the network; this way, local instances of LSDBs of different routers are consistent to each other.

The process through which link-state advertisements are disseminated to all the routers in the network, is denominated **flooding**. Routers can also update their local instance of LSDB by *synchronizing* it with the local instance of a neighboring router.

4.3.1 Flooding

Local instances of LSDB need to be updated in the network every time that topology changes. Hence, a router floods new advertisements when changes are detected in the set of links maintained by the router, in order to enable any other router to modify its local instance of LSDB accordingly and, if necessary, recalculate paths. In ideal conditions[1], this mechanism would be sufficient for keeping identical LSDB instances in every router in the network. As transmission errors, packet losses and disconnections may occur in wireless, mobile or ad hoc networks, additional mechanisms may be used to reduce the impact of failures.

- **Periodic flooding of advertisements.** Even if no changes are noticed in the router set of links, the router floods periodically its link-state advertisement over the network. Periodicity in flooding brings to routers in the network an additional means of detecting the disappearance of a particular router, when no advertisement is received for more than the time interval between two consecutive floods.

[1]Ideal conditions imply static and always-connected networks with error-free links, for which all routers are reachable for any topology change advertisement.

- **Reliable flooding of topology messages.** Reception of packets containing link-state advertisements is acknowledged by every receiver, or retransmitted by the sender/forwarder in absence of such acknowledgment. Reliable flooding is used by the main routing protocols for wired networks (OSPF and IS-IS); however, it is not used in MANET-specific link-state protocols such as OLSR.

Periodic and reliable flooding address different issues concerning topology flooding. Requiring that advertisements are acknowledged by the receiving routers (reliable flooding) enables senders and forwarders to overcome channel failures by retransmitting the missing packet until an acknowledgement is received from the corresponding neighbor. Reliable flooding, however, does not guarantee that routers receive flooding descriptions. Figure 4.2 illustrates a case in which reliable flooding is useless due to router mobility: when router x moves, it stops being reachable from router f_1 and is not yet known by its new neighbor f_2. If a link-state advertisement has been received by f_1 and f_2 during the flooding procedure, the advertisement is forwarded by f_1 in $t_0 < t < t_1$, and it is not received by x. Retransmissions of f_1 in absence of acknowledgement are not received by x in $t > t_0$. f_2, in turn, may not expect acknowledgement from (or may not flood the advertisement towards) x as x has not yet been discovered as a neighbor by f_2.

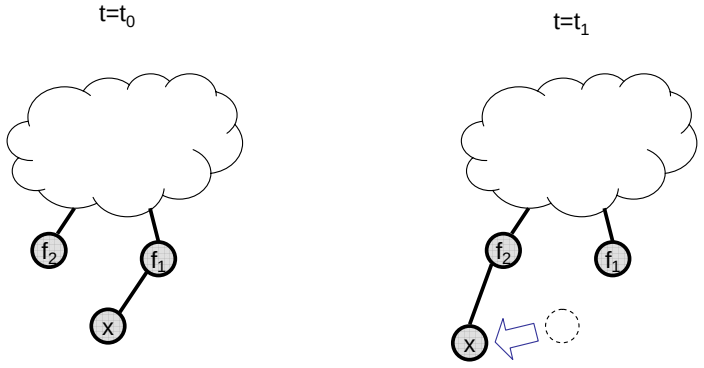


Figure 4.2: Mobility and neighborhood change in an ad hoc network.

Moreover, acknowledgements may also be lost due to wireless channel failures – the loss of a link-state acknowledgement implying additional, and unnecessary, retransmissions of the acknowl-

edged advertisement.

Periodic flooding enables routers that have missed a link-state advertisement, to acquire the missing topology information in following floods. This way, the time that a router has stale information about the set of links of a particular router due to the loss of its link-state advertisement is bounded by the time interval between consecutive floods in the network. The optimal length for this time interval depends on the characteristics and purpose of the network. Such length needs to accommodate factors such as:

a) the bandwidth available for flooding traffic, as shorter intervals cause higher flooding overhead, and

b) the network tolerance to topology information staleness, as longer intervals imply longer average periods in which routers may keep obsolete information after the loss of a flooded packet.

4.3.2 LSDB Synchronization

The synchronization of local instances of LSDB of two neighboring routers consists of (i) the exchange of the contents (advertisements) of local instances of LSDB of both routers, and (ii) the installation of the most updated topology information from both routers in each of both local instances of LSDB. After an LSDB synchronization process, each of the participating routers has the most recent topology information that was present in any of the routers before the synchronization.

LSDB synchronization does not replace flooding, as it does not guarantee on its own the consistency of LSDB local instances across the network. The fact that all routers have synchronized their local instances of the LSDB with all their neighbors does not imply that such local instances will continue to contain the same information about the network topology without additional mechanisms[2]. When a pair of neighboring routers have synchronized (exchanged and updated with the

[2]This is different, for instance, in proactive distance-vector routing, in which the network is expected to converge (meaning that routing tables of all routers are consistent with the network topology and provide network-wide shortest paths) through *repeated* database synchronization processes. In the considered link-state context, synchronization occurs, at most, *once* in time that a link is up, which is not sufficient for assuring that all LSDB local instances contain the same information when topology changes.

most recent advertisements) their LSDB local instances, the link between them is denoted a *synchronized link* – this term is used throughout the manuscript. A network path composed of only synchronized links is denoted a *synchronized path*.

Definition 4.2 (**Synchronized path**). A network path between routers x and y, p_{xy}, is a *synchronized path* if all the edges that are part of such path, correspond to synchronized links in the network.

The use of LSDB synchronization in a network reduces the impact of flooding packet losses and disconnection, as it replaces the obsolete link-state advertisements of the local instances of LSDB with the most recent advertisements of both synchronizing routers. In particular, it permits routers that just joined the network to acquire the topology information that has been previously flooded through the network, at once, by synchronizing their local instances of LSDB with one of its neighbors.

This mechanism is implemented in the main link-state routing protocols for wired networks (OSPF, IS-IS), but the conditions in which such synchronization is performed are not completely adapted to mobile ad hoc operation. Therefore, the mechanism "as-is" is not considered in MANET-specific protocols such as OLSR, and its use is limited, for instance, in the different OSPF MANET extensions, as it is described in Part III.

4.4 Issues in Ad hoc Networks and Compound ASes

The use of a link-state routing protocol in ad hoc networks or compound ASes gives rise to a set of issues, which are related to the dynamic, unpredictable topology of these networks and the implications of these properties in communication. This section identifies three main aspects: constraints imposed by bandwidth scarcity in wireless ad hoc networks (section 4.4.1), the performance of flooding operations in wireless environments (section 4.4.2) and the interest of LSDB synchronization in the context of compound ASes, in which fixed and ad hoc networks coexist in the same internetwork (section 4.4.3).

4.4.1 General Bandwidth Constraints

In ad hoc networks, the scarcity of bandwidth and the unreliable nature of links impose additional constraints to operation of link state routing protocols. Advertising link changes to all routers in the network may produce an excessive amount of control traffic when these changes are frequent, as it may be the case in mobile ad hoc networks. Control traffic dedicated to the update of local instances of the LSDB over the network depends on three factors:

(i) the topology update rate, which should be at least the link change rate and may get higher in case of periodic flooding,

(ii) the size of the packets carrying link-state advertisements, and

(iii) the number of times that an advertisement is retransmitted over the network in order to reach all routers.

The topology update rate cannot be reduced below the link change rate without affecting network convergence and thus correctness of topology information and optimality of computed paths. The other two, retransmissions per flooded link-state advertisement and size of such advertisements, can be optimized in order to reduce the resulting overhead without compromising the quality of the selected routes. These optimizations, however, require more complex link-state operations. In particular, routers in an ad hoc network need to modify their behavior in the following senses:

1. Instead of describing all the links that are maintained in a link-state advertisement, routers select a subset of such links to be advertised to the network.

2. Instead of forwarding all link-state advertisements that are received (pure flooding), routers participate in flooding of a limited part of the link-state advertisements sent over the network.

While both modifications reduce the overhead caused by link-state flooding, they need to be compatible with the main objective of such operation – the update of all local instances of LSDB in the network in a way such that shortest paths can be computed by all routers. Following chapters

(from chapter 6 to 9) elaborate on how the trade-off between flooding cost and performance can be addressed.

4.4.2 Flooding over Wireless Interfaces

The specific properties of wireless media and the presence of a (partially) shared bandwidth, described in chapter 2, impacts the way that link state routing is performed in ad hoc networks. In particular, the flooding procedure needs to accommodate the following characteristics:

- **Semibroadcast properties of wireless communication**. As mentioned in chapter 2, a wireless interface can communicate directly and simultaneously with all its wireless neighbors – not necessarily all wireless interfaces in the network. Operations that require that information is received by all such neighbors (such as flooding or Hello packets exchange) are performed via multicast. Moreover, as the sets of neighbors of two wireless interfaces that can communicate directly may be different, flooding may require that a packet is transmitted through the same wireless interface that it has been sent. In case that reliable flooding is used and acknowledgements are expected for link-state advertisements, such transmissions over the same wireless interface implicitly acknowledge the successful reception of the corresponding advertisement by that interface.

- **Wireless collisions**. The fact that packets may be forwarded simultaneously by wireless interfaces having received them in the same shared medium is likely to cause packet collisions during the flooding procedure. This effect is more significant as the wireless network is denser and the amount of flooding traffic increases, but it might be alleviated by distributing retransmissions along a time interval after its reception by an intermediate wireless interface. This technique can be implemented by delaying every received packet with a random delay (jitter) before forwarding it over the wireless interface in which it was received.

4.4.3 LSDB Synchronization in Compound ASes

The time that interfaces need to acquire the topology information contained in lost link-state advertisements (called *re-hooking time* in this section), can be bounded by way of two mechanisms presented previously in this chapter: periodic flooding and LSDB synchronization. Periodic flooding provides a maximum interval between two consecutive updates from the same router, and LSDB synchronization enables routers to update their topology information by exchanging their local instance of LSDB with those of (some of) their neighbors.

The existence of two such mechanisms for addressing the same issue may appear redundant. In particular, there is no agreement about the role of LSDB synchronization in link-state protocols that already use periodic flooding: OSPF uses both mechanisms, whilst other protocols do not implement synchronization (OLSR) or use it only for specific types of networks (IS-IS, see chapter 10 for details).

For some types of internetworks, and in particular for compound ASes, LSDB synchronization offers some advantages for containing the impact of topology update losses, that cannot be provided with periodic flooding. Such advantages can be observed in internetworks in which there is a coexistence between networks with opposite profiles in terms of available bandwidth and link dynamism, as it is the case for compound ASes.

Reduction of re-hooking time through periodic flooding is performed by increasing the rate of consecutive floods, which has an impact on the flooding overhead over the whole internetwork. High periodic flooding rates cause excessive overhead in ad hoc networks, given the scarcity of bandwidth on wireless media. Moreover, link-state advertisements coming from routers of a fixed network may be in part redundant if flooded at a high rate, as fixed links are stable in average and therefore the set of links of a fixed router is not likely to change.

LSDB synchronization enables interfaces, and in particular those belonging to ad hoc networks, to reduce their *re-hooking time* by exchanging and updating their local instances of LSDB with (some of) their neighbors. Rather than affecting the whole internetwork, overhead generated by the increase of the number of synchronization processes of an ad hoc router has only a local

impact – that is, in the neighborhood of synchronizing routers. The re-hooking time of interfaces of ad hoc routers with respect to fixed routers' advertisements can be optimized independently from the flooding configuration of fixed routers.

Consider the example of Figure 4.3, in which routers 1 and 2 are fixed and maintain only wired links; 3 and 4 are hybrid fixed routers able to maintain both wired and wireless links and routers 5, 6 and 7 are ad hoc routers that maintain wireless links and may move freely through the network. Fixed routers 1 and 2 can handle changes in their wired links by transmitting topology updates at low rate. Mobile ad hoc routers (5, 6 and 7) and, more general, routers maintaining wireless links (also the hybrid routers 3 and 4) should use significantly lower time intervals. If, for any reason, a mobile router did not receive a topology update from a fixed router (*e.g.* router 1), it will be unable to update its local instance of LSDB until the next update from the fixed router, failing at computing valid routes that involve that router in the meanwhile. By synchronizing their local instance of LSDB with a neighbor, ad hoc routers are able to acquire the topology information lost due to their mobility without depending on the rest of routers, in particular those with lower flooding rates.

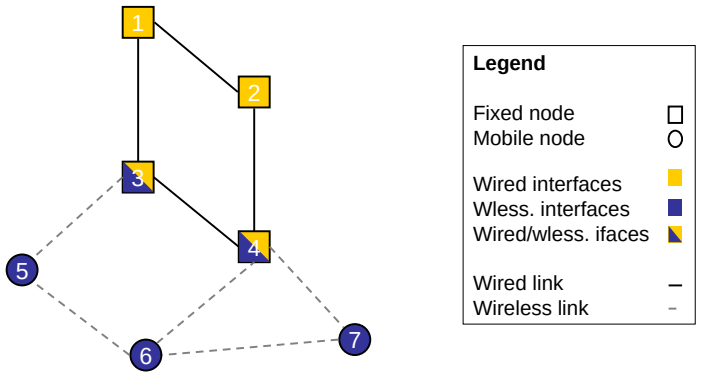


Figure 4.3: Example of compound (wired/wireless) network.

In the context of compound Autonomous Systems, the use of LSDB synchronization enables independent optimization of interfaces' *re-hooking time*. In particular, the re-hooking time of interfaces prone to LSA losses, due to mobility or wireless failures – *i.e.*, wireless interfaces of ad

hoc and fixed routers. LSDB synchronization becomes a complement to other flooding mechanisms –reliable and periodic flooding– that may be used together.

4.5 Conclusion

In networks that use link-state routing, the network topology is stored in the Link State Database (LSDB). This LSDB is distributed among the routers in the network. Link state routing over a network requires that routers maintain consistent and updated information in their respective local instances of the LSDB. Based on this information, they select the shortest paths to every possible destination. Updates of the topology information maintained by each router is therefore an important issue for link-state routing protocols.

Three operations are performed over the network to ensure consistency and accuracy of local instances of the LSDB: routers describe their links, the resulting topology updates are flooded over the networks and neighboring routers may synchronize their local instances of LSDB. The way that these operations are performed determines the characteristics of a link-state routing protocol.

Link-state routing protocols in multi-hop wireless ad hoc networks need to accommodate several issues and challenges that arise from the characteristics of wireless communication, topology dynamism and absence of networking infrastructure. In particular, they need to address the scarcity of bandwidth in wireless networks and the semibroadcast characteristics of communications among wireless interfaces. Bandwidth scarcity needs to be taken into consideration in the three link-state operations. Routers may select only a subset of links to be advertised (topology selection). Flooding needs to be optimized (i) to take advantage of semibroadcast capabilities of the medium, (ii) to prevent packet collisions on the shared media, and (iii) to minimize the resulting overhead by restricting the number of routers and links involved in flooding.

The number of links that are *synchronized* in a network, that is, the number of LSDB synchronization processes that are performed between neighboring routers, may also be limited in order to minimize overhead. In case of LSDB synchronization, the very presence of this operation in link-state protocols may be controversial for networks with bandwidth limitations, as the role

may appear redundant with flooding. However, the update of local instances of LSDB through the exchange with neighboring routers, without additional floods, is useful in internetworks in which some routers need topology updates from the rest of the internetwork at a higher rate than the rate at which topology information is flooded. This is the case for compound Autonomous Systems, in which the coexistence of fixed and ad hoc networks implies different needs of flooding and topology update rates. Consequently, the routing extensions presented and analyzed in Part III of this manuscript implement both mechanisms, and LSDB synchronization in particular, for operation in ad hoc networks inside compound ASes.

The remainder of Part II of this manuscript details the different issues presented in this chapter. Chapter 5 analyzes the use of random delays (jitter) before forwarding topology updates, in order to minimize the probability of wireless collisions. Chapter 6 introduces the concept of network overlays for analyzing each link-state operation separately in ad hoc networks, and chapters 7, 8 and 9 propose and evaluate different techniques for minimizing the overhead required by each link-state operation without affecting their performance.

Chapter 5

Packet Jittering for Wireless Dissemination

In ad hoc networks, and in general in wireless networks, simultaneous packet transmissions by neighboring routers may lead to packet collisions, as explained in chapter 2. In order to prevent such collisions, RFC 5148 [29] proposes that routers randomly delay their packet transmissions by a small amount, in order to attempt to distribute transmissions over time. This mechanism of random distribution of packet transmissions is herein called *packet jittering*.

As some link-state operations (*e.g.*, flooding or neighbor sensing) are prone to cause collisions in wireless ad hoc networks, jittering may be employed to improve the performance of link-state routing in ad hoc networks. This chapter describes the application of jitter techniques to link-state mechanisms and, in particular, explores the use of jitter in topology flooding.

5.1 Outline

This chapter provides an analysis of the impact of jittering, based on a statistical model of wireless flooding at a particular router using a link-state protocol. Section 5.2 describes packet jittering in detail, and discusses the cases in which it may be advantageous to use jitter for link-

state routing. That section details the use of jittering for preventing packet collisions in flooding. Section 5.3 presents an analytical model of flooding in a router using a link-state protocol. The impact of random delays in packet forwarding is studied in this analytical framework. Section 5.4 validates the results obtained in the previous section through simulations. Finally, section 5.5 concludes the chapter.

5.1.1 Terminology

Throughout this chapter, the following terminology is used:

- Given a real valued random variable X, its probability density function (PDF) is denoted by $f_X(x)$, and its cumulative distribution function (CDF) is denoted by $F_X(x)$, satisfying that $F_X(x) = P(X < x) = \int_{\infty}^{x} f_X(s)ds$. The mean of the random variable X is denoted by $\mathbb{E}\{X\}$, defined as follows:

$$\mathbb{E}\{X\} = \int_{-\infty}^{\infty} x f_X(x)dx$$

- $\delta(x)$ denotes the Dirac's delta generalized function, defined canonically as satisfying the following two conditions:

$$\int_{-\infty}^{\infty} \delta(x)dx = 1 \quad ; \quad \delta(x) = 0, \forall x \neq 0$$

- $H(x)$ denotes Heaviside's step function, which is defined canonically as follows:

$$H(x) = \begin{cases} 1 & , x \geq 0 \\ 0 & , x < 0 \end{cases}$$

5.2 The Jitter Mechanism

Wireless collisions occur when two neighboring wireless interfaces or two wireless interfaces with one common neighbor, transmit a packet at the same time. When transmissions causing a

packet collision are not based on fully autonomous decisions from the corresponding interfaces, *i.e.*, when they are determined or conditioned by a common input or configuration, the probability of a collision may be reduced significantly by randomly distributing such transmissions over a time interval.

5.2.1 Common Input and Common Configuration

Figure 5.1 illustrates the case of common input: routers B and C react to the transmission from A by sending packets immediately after receiving A's packet. This results in a collision if B and C are neighbors of each other.

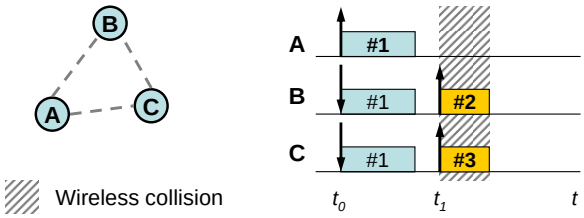


Figure 5.1: Wireless collision caused by reaction to a common input. *Transmission of packet #1 by router A implies that routers B and C react by transmitting packets #2 and #3 immediately after receiving #1, and thus packets #2 and #3 cause a collision.*

A common configuration may also cause wireless collisions, as shown in Figure 5.2. The fact that A and B transmit packets periodically, with the same time interval, may lead to consecutive packet collisions if A and B transmissions start at the same time or are separated a multiple of the time interval. Whilst the probability that two neighboring interfaces start periodic transmissions in times satisfying this condition is low in ad hoc networks, this situation is taken into consideration because time synchronization (*i.e.*,) in these cases has severe implications. Interfaces affected by these issues are unable to perform any successful transmission without modifying the interval between consecutive transmissions.

Periodic packet transmissions from A and B cause collisions if transmissions

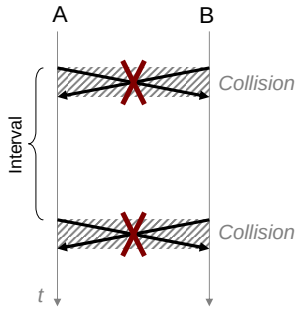


Figure 5.2: Wireless collision caused by synchronization in periodic packet transmissions.

5.2.2 Wireless Collisions and Jitter in Link-State Routing

Both cases illustrated in Figures 5.1 and 5.2 may occur while performing link-state routing over ad hoc networks. Periodic transmission of Hello packets with a uniform Hello interval (see section 3.4.1) may, *e.g.*, be affected by synchronization. This might also be the case of topology flooding, when topology descriptions are generated periodically (see section 4.3.1).

Wireless packet collisions caused by reaction to a common input may happen in two tasks related to topology flooding: packet forwarding and packet acknowledgement, in case of reliable flooding. When an interface forwards a packet, several neighbors of this interface may forward in turn a topology update immediately after having received it, thus causing a packet collision if they hear each other. When neighbors of an interface acknowledge a packet transmitted by the interface, they may send explicit acknowledgements immediately after the reception of the packet, with the same result.

RFC 5148 [29] specifies techniques for minimizing the probability of packet collisions in cases of reaction to common inputs and common configuration (periodic transmissions). Jitter values (denoted as $Jitter$) are selected randomly through a uniform distribution within $[0, MAXJITTER]$, and are used in the following two cases:

- **Periodic transmissions.** Given an interval $MESSAGE\_INTERVAL$, the time lapse between two consecutive transmissions is

$$\Delta t = MESSAGE\_INTERVAL - Jitter \tag{5.1}$$

This corresponds to the interval between consecutive messages in absence of jitter ($MESSAGE\_INTERVAL$), decreased by a random amount ($Jitter$) computed independently for each transmission. Jitter values need to satisfy therefore the following condition:

$$MAXJITTER < MESSAGE\_INTERVAL \tag{5.2}$$

- **Reaction to common input.** A transmission caused by an external input (a topology description that has to be acknowledged, needs to be forwarded or generated for flooding due to a topology change) is delayed a $Jitter$ interval, instead of being trasmitted immediately after receiving the input. In case that these reactions cannot be performed before a minimum time interval $MESSAGE\_MIN\_INTERVAL > 0$ from the last reaction, such minimum interval is reduced to $MESSAGE\_MIN\_INTERVAL - Jitter$. Such a non-zero $MESSAGE\_MIN\_INTERVAL$ parameter may exist to prevent too frequent flooding and forwarding decisions – *e.g.*, consecutive floods in OSPF [107], which are not allowed within intervals shorter than $MinLSInterval$. This parameter is reduced by the jitter value in order to prevent that packet jittering leads to slowing-down the flooding processes across the network. That implies that, when $MESSAGE\_MIN\_INTERVAL > 0$, jitter values need to satisfy:

$$MAXJITTER < MESSAGE\_MIN\_INTERVAL \tag{5.3}$$

RFC 5148 [29] provides additional restrictions for the value of $MAXJITTER$, in order to improve jittering performance and minimize side effects on the corresponding protocols.

5.2.3 Forwarding Flooding Packets with Jitter

This chapter explores the use of jittering for forwarding topology description messages in the framework of a link-state routing protocol. In this context, wireless collisions may occur due when neighboring interfaces react (forward a packet) to a common input (reception of a flooded packet). The motivation for using jittering in this case is therefore two-fold: to minimize wireless collisions by distributing transmission events, and to reduce the number of performed transmissions

by aggregating several messages in a single packet. The chapter thus focuses on the use of jitter for the "reaction to a common input" case, presented in section 5.2.1.

Topology description messages are flooded over the network in multi-message packets. A wireless interface that receives such a topology packet may decide to forward some of the messages contained in the packet. The interface thus assigns a jitter value to those messages in the packet that will be forwarded – the same value for all messages belonging to the same packet, and schedule their transmission after the expiration of such value. Together with forwarding messages from other interfaces, a wireless interface may flood self-generated messages describing its own topology. When a topology description message is self-generated this way, it is scheduled for immediate transmission. This is equivalent to assign such self-generated messages a jitter of zero. When a transmission is scheduled, all topology messages –either received from other interfaces or self-generated– that have been scheduled and not yet transmitted are sent in a single topology packet, as summarized in Figure 5.3.

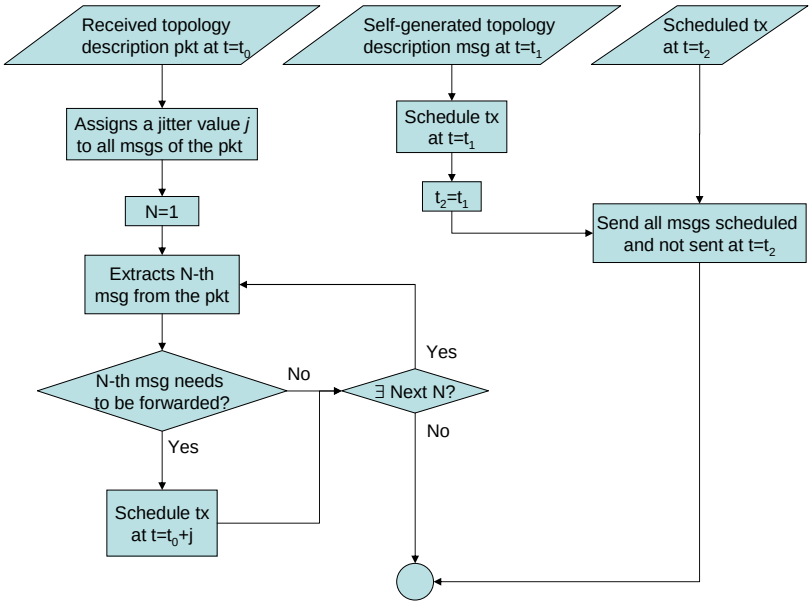


Figure 5.3: Forwarding algorithm with jitter.

At least three aspects can be highlighted in this procedure:

- **Effective and scheduled time to transmission.** Topology messages are forwarded with a delay shorter than or equal to their scheduled time, given the fact that all pending transmissions are performed together when the jitter of any pending message expires. The difference between scheduled delay and effective delay depends on the arrival rate of packets with messages to be forwarded.

- **Immediate flooding of self-generated messages.** The fact that self-generated topology description messages are sent immediately also contributes to the difference between scheduled and effective delays. Message self-generation rate, packet reception rate and jitter value bounds (maximum value of jitter, $MAXJITTER$) are therefore factors that impact the effective delay of forwarded messages. If message self-generation rate increases significantly, it may dominate the effect of the other two factors and make changes in jitter range irrelevant.

- **Impact on packet rate.** Since forwarded packets may contain messages from one or more received packets, the use of jittering leads to a reduction in the rate of flooded packets, for sufficiently high jitter values. A wireless interface sends packets at a lower rate than it receives packets to be forwarded, if jitter values are bigger than the inter-arrival time of in-packets. This is, however, at the expense of increasing the length of the forwarded packets, as they contain, under these conditions, a growing number of messages.

The analysis presented in this chapter permits evaluating the impact of these three elements by way of a probabilistic theoretical model.

5.3 Analytical Model

This section presents a statistical analytical model of the traffic received and forwarded by a wireless router (denoted throughout this section as a *node*) that uses jittering for avoiding wireless collisions. This analytical model is used to describe two aspects of forwarding operation that are affected by the jitter mechanism: given a node, the rate at which such node forwards packets and the effective delay of packets when they are forwarded by such node, depending on the jitter range.

The model described in this section thus focuses on the use of jitter in a particular node. Section 5.3.1 presents the main assumptions of the model, the elements and variables to describe forwarding traffic (message and packet rates) sent/received by a particular node. Section 5.3.2 studies the relationship that the jitter mechanism establishes between the rate of forwarded packets and the rates of received and self-generated packets.

The second aspect in which the impact of jitter is evaluated concerns the effective delay of forwarded packets caused by jitter. The time between reception and retransmission of the contents of a flooding packet p depends on the arrival of packets that be in one of the following three cases (see also Figure 5.4):

(i) packets received after p, but prior to the scheduled time for p,

(ii) packets received before p, scheduled to be sent after p has been received (and before p is scheduled to be sent), and

(iii) self-originated messages that are generated after p is received, and before p is scheduled to be sent.

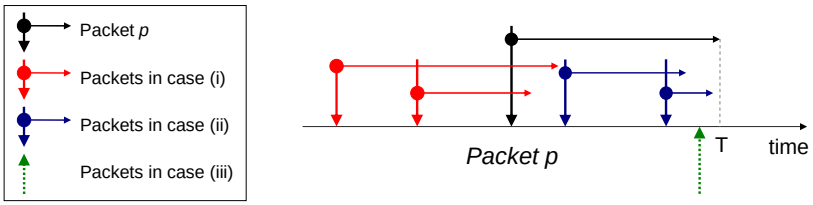


Figure 5.4: Illustration of packet cases, for jitter analysis.

Section 5.3.3 defines and characterizes random variables (in terms of PDF and CDF) for describing the scheduled time of transmission of packets that before or after p and may impact the effective delay for the retransmission of p. These random variables are used in section 5.3.4 to define the time interval between when a packet p is received in a node and until it is forwarded. That section thus provides an upper and a lower bound for the average of such time-to-transmission.

	Packets	Messages
Received to fwd	λ_{in}	γ_{in}
Self-originated	λ_g	γ_g
Sent	λ_{out}	γ_{out}

Table 5.1: Traffic model variables.

Finally, section 5.3.5 summarizes the most prominent results achieved by way of this model, as well as discusses the limitations and possible extensions of such model.

5.3.1 Traffic Model and Assumptions

This section examines a node which participates in flooding of messages (topology descriptions) from other nodes in a network, which also generates its own messages to be flooded over the network. These messages are sent in packets, each packet containing one or more messages.

Four types of traffic are distinguished: traffic received by the node to be forwarded (in-traffic); traffic generated by the node (self-traffic); traffic sent by the node (out-traffic) and traffic received by the node, but not forwarded. For the purposes of this chapter, this received non-forwarded traffic is not relevant, and is thus not considered: in this chapter, all packets received are to be forwarded. Table 5.1 displays the variables used for describing the traffic rates in terms of messages per second (γ) and packets per second (λ), and Figure 5.5 illustrates the role of each variable in the operation of a node.

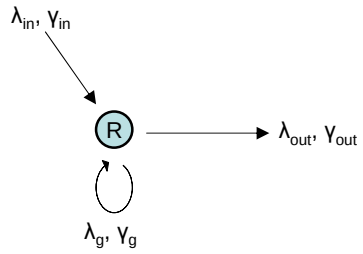


Figure 5.5: Node model.

Packet arrivals to the node (either self-generated or received from other nodes) are modeled

as punctual homogeneous Poisson processes. Function $f \equiv f(k; \lambda, \Delta t)$ denotes the probability that k packets arrive at a rate λ in a time interval Δt, that is:

$$f(k; \lambda, \Delta t) = e^{-\lambda \Delta t} \frac{(\lambda \Delta t)^k}{k!} \tag{5.4}$$

5.3.2 Message and Packet Rates

This section describes the relationship between message and packet rates received and sent by a node. Every message that a node sends to the network (out-message) has been either received to be forwarded (in-message), or created by the node itself to describe its own topology (self-message). Therefore, message rates satisfy the following relationship:

$$\gamma_{out} = \gamma_{in} + \gamma_g \tag{5.5}$$

Packets contain one or more messages. For consistency, it is assumed that a self-generated packet contains one and only one self-generated message, that is:

$$\lambda_g = \gamma_g \tag{5.6}$$

The relationship between packet rates (λ_{out}, λ_{in}, λ_g) is less straightforward. In-messages may be forwarded by way of:

(1) out-packets that contain only other in-messages, or

(2) out-packets that contain one (and only one) self-generated message.

The rate of out-packets in (2) is then exactly λ_g. Out-packets in (1) contain (only) in-messages for which no self-traffic is generated while they were waiting for retransmission. As out-packets in (1) contain the messages from all the in-packets received, but not yet forwarded, the rate of out-packets in (1) is equal to, at worst, or lower than the in-packet rate. *Theorem 5.1* describes a lower bound for the out-packet rate as a function of in-packet and self-packet rates.

Theorem 5.1. *Let λ_g be the rate of self-generated packets and λ_{in} the rate of in-packets. Let T^* be the random variable of the time interval between the arrival of the first in-packet after a transmission and the time in which messages of such in-packet are forwarded (not considering the impact of packet self-generation). Assume that T^* is independent from the arrival of in-packets after the first.*

Then, the rate of out-packets is:

$$\lambda_{out} = \frac{\lambda_{in} + \lambda_g}{1 + \frac{\lambda_{in}}{\lambda_g}(1 - t(\lambda_g))} \tag{5.7}$$

where $t(\theta) = \mathbb{E}\{e^{-\theta T^*}\} = L\{T^*\}(\theta)$, and where L denotes the Laplace transform.

Proof. Packet transmission corresponds to a renewal process. The renewal process starts with the waiting time before the arrival of a packet I (received to be forwarded) or a packet G (self-generated to be flooded). This period is of average length $WT = \frac{1}{\lambda_{in}+\lambda_g}$. Depending on the type of the first packet that arrives, two cases are considered:

- If it is a packet G (with probability $\frac{\lambda_g}{\lambda_{in}+\lambda_g}$) then the renewal phase ends here.
- If it is a packet I (with probability $\frac{\lambda_{in}}{\lambda_{in}+\lambda_g}$), then there is an additional phase that ends with the arrival of a packet G if it occurs before time T^*, or by the interval of length T^* otherwise. As T^* is independent from I arrivals, no other cases are possible. T^* denotes the random variable for the interval between:
  - the time of the first I packet arrival after after a transmission (*i.e.*, when no other packets I are waiting to be forwarded), and
  - the time in which messages from this I packet are forwarded (possibly together with other messages), absent self-generated packets.

  Given a value x of T^*, the probability density function that a G packet appears at time x is $\lambda_g e^{-\lambda_g x}$ (exponential distribution of Poisson arrivals). Then, the average contribution of the phase when G arrives before T^* is equal to $\int_0^{T^*} x e^{-\lambda_g x}\lambda_g dx$. The average contribution of the phase when G arrives after T^* is equal to $T^* e^{-\lambda_g T^*}$. The sum $\int_0^{T^*} x e^{-\lambda_g x}\lambda_g dx + T^* e^{-\lambda_g T^*}$ is equal to $\frac{1}{\lambda_g}(1 - e^{-\lambda_g T^*})$. Averaging over all values of T^*, it is equal to $\frac{1}{\lambda_g}(1 - t(\lambda_g))$.

Therefore the average renewal phase duration is equal to:

$$\frac{1}{\lambda_{in} + \lambda_g} + \frac{\lambda_{in}}{\lambda_{in} + \lambda_g}\frac{1}{\lambda_g}(1 - t(\lambda_g))$$

And the output packet rate is the inverse of the renewal phase average (exactly one output packet per renewal phase):

$$\lambda_{out} = \frac{\lambda_{in} + \lambda_g}{1 + \frac{\lambda_{in}}{\lambda_g}(1 - t(\lambda_g))}$$

□

- **Asymptotic behavior.** Notice that when $\lambda_g \to \infty$, $t(\lambda_g) = 0$, *i.e.*, when packet G arrives before T^* with probability 1, then $\lambda_{out} = \lambda_g$. Conversely, when $\lambda_g \to 0$: $1 - t(\lambda_g) = \mathbb{E}(T^*)\lambda_g + O(\lambda_g^2)$ (from the Taylor decomposition of $t(\theta)$), *i.e.*, packet G arrives after T^* with probability 1 then $\lambda_{out} = \frac{\lambda_{in}}{1+\lambda_{in}\mathbb{E}(T^*)} + O(\lambda_g)$. If no jitter is implemented, $T^* = 0$, $t(\lambda_g) = 1$ and, therefore, $\lambda_{out} = \lambda_{in} + \lambda_g$. If, on the contrary, jitter is selected within an interval $[0, 2T]$ arbitrarily long, *i.e.*, with $T \longrightarrow \infty$, then $T^* \longrightarrow \infty$ and the out-packet rate approaches $\lambda_{out} \longrightarrow \frac{\lambda_{in}+\lambda_g}{1+\frac{\lambda_{in}}{\lambda_g}} = \lambda_g$. In this case, out-packets are transmitted when a new message is self-generated – and immediately forwarded via an out-packet, together with all in-messages not yet forwarded.

Theorem 5.1 assumes the independence between T^* and posterior in-packet arrivals (and jitter scheduling). As in practice the interval between the arrival of a first in-packet and the retransmission of its messages may be affected (shortened) by the scheduled retransmission time of packets arriving after such first in-packet (see section 5.2.3), equation (5.7) corresponds to a lower bound for the out-packet rate that can be achieved with a given in-packet rate and jitter range.

5.3.3 Statistical Description of Traffic to be Forwarded

Let p be an in-packet received at time $t = 0$. The arrival at the node of other in-packets after and before p is modeled as a collection of random variables $\{T_t(i)\}_{i\in\mathbb{Z}^*}$ with a punctual homogeneous Poisson distribution with rate λ_{in}, where i indicates the order of arrival with respect to p ($i > 0$ for in-packets received after p, $i < 0$ for in-packets received before p)[1]. $T_t(i)$ is thus the random variable that indicates the arrival time of the i-th packet received after p (before, if $i < 0$).

[1]Observe that the case $i = 0$ corresponds to the reception of packet p, which is deterministically received in $t = 0$, so it is excluded from the random process.

When the i-th in-packet is received, the messages contained in such packet are scheduled to be forwarded after a random delay (jitter). According to [29], all messages in the same i-th packet are assigned the same jitter value $T_j(i)$. The random variable corresponding to such jitter value is denominated $T_j(i)$ and is uniformly distributed within the interval $[0, 2T]$, where $2T < MAXJITTER$.

Figure 5.6 shows the role of random variables $T_t(i)$ and $T_j(i)$ within the considered traffic model, for a particular node.

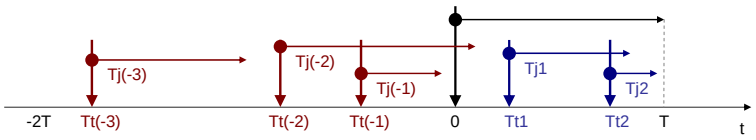


Figure 5.6: Illustration of the traffic model for packets containing messages to be forwarded.

The scheduled time for retransmission of the messages contained in the i-th received packet is therefore a random variable defined as follows:

$$X(i) = T_t(i) + T_j(i) \tag{5.8}$$

Theorem 5.2 describes statistically the set of random variables $X(i)$ associated with packets received after p $(i > 0)$. Figure 5.7 shows the PDF (Figure 5.7.a) and CDF (Figure 5.7.b) of $X(i)$ for different values of i, with $T = 0.1sec$.

Theorem 5.2. *Random variables $X(i)$, for $i > 0$, are defined by the following probability density function (PDF):*

$$f_{X(i)}(x) = \frac{1}{2T}\lambda_{in}^{i}\left[-e^{-\lambda_{in}x}\sum_{n=1}^{i}\frac{x^{i-n}}{\lambda_{in}^{n}(i-n)!}\right]_{g(x)}^{x} \tag{5.9}$$

where $g(x)$ has the following expression:

$$g(x) = \begin{cases} 0 & , x < 2T \\ x - 2T & , x \geq 2T \end{cases} \tag{5.10}$$

Proof. From the definition of $X(i)$,

$$X(i) = T_t(i) + T_j(i) \iff f_{X(i)}(x) = (f_{T_t(i)} * f_{T_j(i)})(x)$$

where $*$ denotes convolution. Operating on such expression,

$$\begin{aligned} f_{X(i)}(x) &= (f_{T_t(i)} * f_{T_j(i)})(x) = \int_{-\infty}^{\infty} f_{T_t(i)}(\tau) f_{T_j(i)}(x-\tau) d\tau = \int_{-\infty}^{\infty} \frac{1}{2T} H(\tau) f_{T_t(i)}(\tau) d\tau = \\ &= \int_{-\infty}^{\infty} \tau^{i-1} \frac{\lambda_{in}^i e^{-\lambda\tau}}{\Gamma(i)} H(\tau) \frac{1}{2T} \left(H(x-\tau) - H(x-\tau-2T)\right) d\tau = \\ &= \frac{1}{2T} \frac{\lambda_{in}^i}{\Gamma(i)} \int_{g(x)}^{x} \tau^{i-1} e^{-\lambda_{in}\tau} d\tau \end{aligned} \tag{5.11}$$

where $g(x)$ is defined in equation (5.10).

Let I_0 denote the primitive function for the integral in (5.11). Then, by integrating by parts iteratively, i times, I_0 becomes:

$$I_0 = -e^{-\lambda_{in}x}(i-1)! \sum_{n=1}^{i} \frac{x^{i-n}}{\lambda_{in}^n (i-n)!} \tag{5.12}$$

Applying (5.12) to expression (5.11), the CDF of $X(i)$ becomes:

$$\begin{aligned} f_{X(i)}(x) &= \frac{1}{2T} \frac{\lambda_{in}^i}{\Gamma(i)} \left[I_0(x) - I_0(g(x))\right] = \\ &= \frac{1}{2T} \frac{\lambda_{in}^i}{\Gamma(i)} \left[-e^{-\lambda_{in}x}(i-1)! \sum_{n=1}^{i} \frac{x^{i-n}}{\lambda_{in}^n (i-n)!} \right]_{g(x)}^{x} \end{aligned} \tag{5.13}$$

As $i \in \mathbb{N}^+$, $\Gamma(i) = (i-1)!$ and therefore:

$$f_{X(i)}(x) = \frac{1}{2T} \lambda_{in}^i \left[-e^{-\lambda_{in}x} \sum_{n=1}^{i} \frac{x^{i-n}}{\lambda_{in}^n (i-n)!} \right]_{g(x)}^{x} \tag{5.14}$$

□

From properties of homogeneous Poisson processes, the statistical description of $X(i)$ variables simplifies considerably if a fixed number of packets (say k) is assumed to arrive within a fixed interval. As jitter values are within the interval $[0, 2T]$, and assuming that the in-packet p arrives at $t = 0$ and is assigned a jitter T, the in-packets prior and subsequent to p that may condition the time of retransmission of messages contained on p are:

(a) Subsequent in-packets ($i > 0$) arrived within $(0, T]$, that is, i-th in-packets such that $T_t(i) < T$.

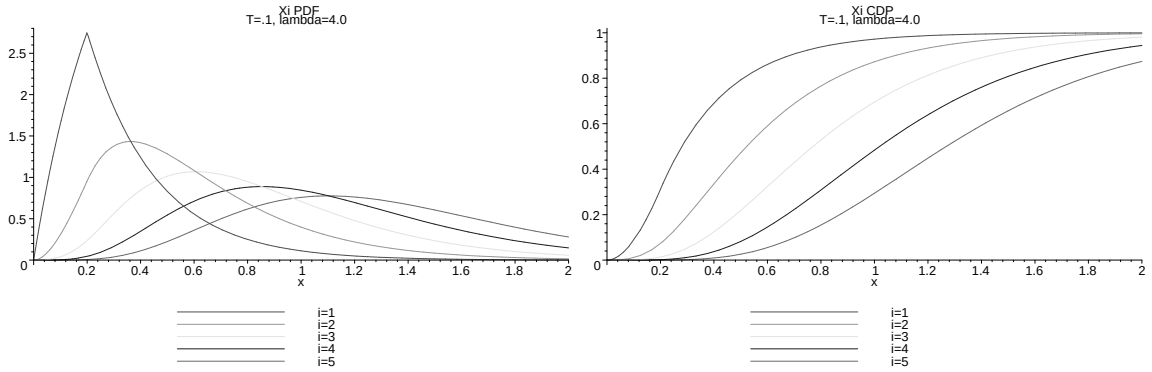


Figure 5.7: **(a)** Probability density function (PDF) for $X(i)$, for $i = 1, 2, 3, 4, 5$, $T = 0.1sec$; **(b)** Cumulative distribution function (CDF) for $X(i)$, for $i = 1, 2, 3, 4, 5$, $T = 0.1sec$.

(b) Prior in-packets ($i < 0$) arrived within $[-2T, 0)$, that is, i-th in-packets such that $-2T < T_t(i) < 0$; and scheduled to be sent after $t = 0$, that is, $0 < X(i) < T$.

Conditions (a) and (b) correspond to conditions (i) and (ii) presented in the beginning of section 5.3.

The following subsections explore the statistical definition of $X(i)$ for conditions (a) and (b). That is, when such k packets that may impact the transmission time of packet p arrive in $t > 0$ (that is, within the interval $(0, T]$) or in $t < 0$ (that is, within the interval $[-2T, 0)$) – this last case being more general than (b). For completeness, arrival at $t = 0$ and scheduled time to transmission of packet p are also defined statistically, as a deterministic random variable X_0.

Packets received within $(0, T]$

When k packets arrive within $(0, T]$, packet arrival time $T_t(i) \equiv T_t$ is distributed uniformly between 0 and T and therefore, variables $X(i)|(0 < T_t(i) \leq T) \equiv \overline{X(i)}$ have the characteristics presented in *Theorem 5.3*. Figure 5.8 illustrates the PDF and CDF of $\overline{X(i)}$ for different values of i ($T = 0.1sec$).

Theorem 5.3. *The random variable,* $\overline{X} \equiv \overline{X(i)}$*, has the following probability density function (PDF):*

$$f_{\overline{X(i)}}(x) = \begin{cases} \frac{1}{2T^2}x & ,0 < x \leq T \\ \frac{1}{2T^2}T & ,T < x \leq 2T \\ \frac{1}{2T^2}(3T - x) & ,2T < x \leq 3T \\ 0 & , \ otherwise \end{cases}$$

The cumulative distribution function (CDF) then is as follows:

$$F_{\overline{X(i)}}(x) = \begin{cases} 0 & ,x < 0 \\ \frac{1}{4T^2}x^2 & ,0 < x \leq T \\ \frac{1}{2T}x - \frac{1}{4} & ,T < x \leq 2T \\ -\frac{1}{4T^2}x^2 + \frac{3}{2T}x - \frac{5}{4} & ,2T < x \leq 3T \\ 1 & , \ otherwise \end{cases}$$

Proof. Direct from the convolution of two random variables uniformly distributed within $(0, T]$ (for $\overline{T_t(i)} \equiv \overline{T_t}$) and $[0, 2T]$ (for $T_j(i)$), respectively. □

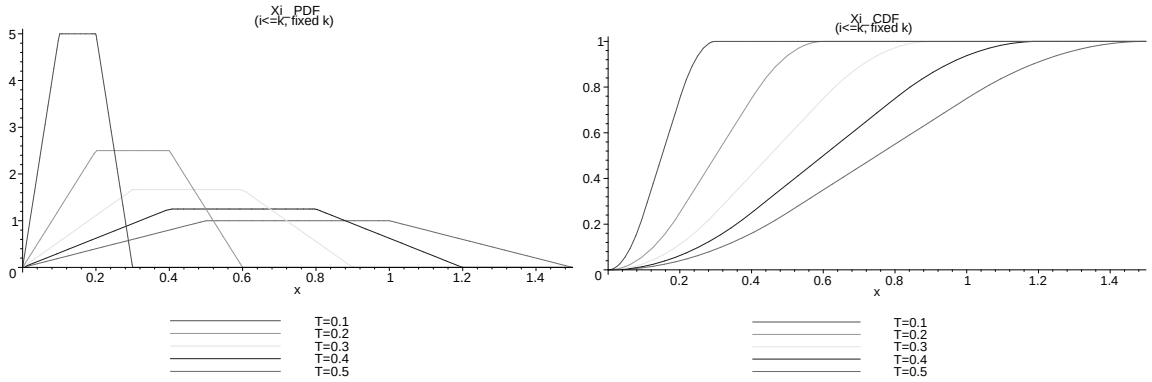


Figure 5.8: **(a)** Probability density function (PDF) for $\overline{X(i)} = (X(i)|\overline{T_t(i)} < T), i \leq k$, for $T = 0.1, 0.2, 0.3, 0.4, 0.5sec$; **(b)** Cumulative distribution function (CDF) for $\overline{X(i)}$.

Packets received within $[-2T, 0)$

When k packet arrivals within the interval $[-2T, 0)$, the distribution of such arrivals $T_{t(-i)}$ is equivalent to the distribution of i.i.d.[2] uniform variables within $[-2T, 0)$. Random variables $X(-i)$,

[2]Independent and identically distributed.

associated to packets received within $[-2T, 0)$ and scheduled within $(0, T]$, are thus statistically described in *Theorem 5.4*. Figure 5.9 shows the PDF of scheduled time $X(-i)$ of packets arriving within $[-2T, 0)$, first, and of packets arrived within $[-2T, 0)$ and scheduled at $t > 0$, second. The corresponding CDFs are shown in Figures 5.10.a and 5.10.b.

Theorem 5.4. *Assuming that k packets arrive within the interval $[-2T, 0)$, the random variable $\overline{X(-i)} \equiv X(-i)|(T_{t(-i)} \in [-2T, 0), X(-i) > 0)$ has the following probability density function (PDF):*

$$f_{\overline{X}(-i)}(x) = \begin{cases} \frac{1}{T} - \frac{1}{2T^2}x & , 0 < x \leq 2T \\ 0 & , \ otherwise \end{cases}$$

The cumulative distribution function (CDF) then is as follows:

$$F_{\overline{X}(-i)}(x) = \begin{cases} 0 & , x < 0 \\ \frac{1}{T}x - \frac{1}{4T^2}x^2 & , 0 \leq x < 2T \\ 1 & , x \leq 2T \end{cases}$$

Proof. Consider the random variable $X(i)|(T_t(i) \in [-2T, 0))$. In the conditions of the theorem, $T_t(i) \sim U[-2T, 0)$. The PDF for the conditioned $X(i)$ corresponds to the density of a triangular distribution,

$$f_{X(-i)}(x) = (f_{T_{t(-i)}} * f_{T_{j(-i)}})(x) = \begin{cases} \frac{1}{4T^2}(\frac{1}{2}x + T) & , -2T < x \leq 0 \\ \frac{1}{4T^2}(T - \frac{1}{2}x) & , 0 < x \leq 2T \\ 0 & , \ \text{otherwise} \end{cases} \tag{5.15}$$

and the cumulative distribution function is

$$F_{X(-i)}(x) = \begin{cases} 0 & , x < -2T \\ \frac{1}{2T}x + \frac{1}{8T^2}x^2 + \frac{1}{2} & , -2T \leq x < 0 \\ \frac{1}{2T}x - \frac{1}{8T^2}x^2 + \frac{1}{2} & , 0 \leq x < 2T \\ 1 & , x \leq 2T \end{cases} \tag{5.16}$$

For each packet arrival, there is a probability $q = P(X(-i) > 0) = \frac{1}{2}$ that the packet is scheduled to be send at $t > 0$ (see Figure 5.10.a) – and only in this case it should be taken into account for determining the time to transmission of a packet arrived in $t = 0$.

The Poisson process corresponding to those arrivals for which $X(-i) \geq 0$, has a rate $q\lambda = \frac{1}{2}\lambda_{in}$. We will denote by $\overline{X}(-i)$ the random variable of the scheduled time of transmission of a message arrived within $[-2T, 0)$, when such scheduled time is > 0.

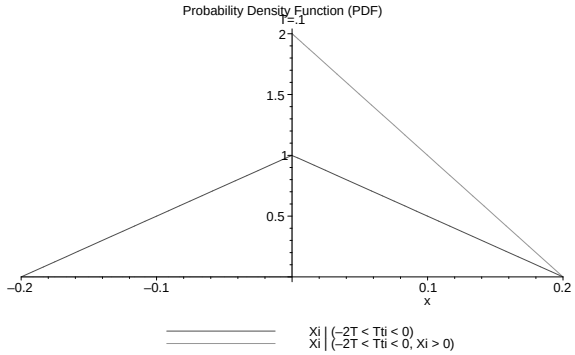


Figure 5.9: PDF of $X(-i)|(-2T \leq T_{t(-i)} < 0)$ and $X(-i)|(-2T \leq T_{t(-i)} < 0, X(-i) > 0)$.

$$\overline{X}(-i) = X(-i)|(T_{t(-i)} \in [-2T, 0), X(-i) > 0)$$

The PDF and CDF of $\overline{X}(-i) \equiv \overline{X}$ are then immediately obtained by conditioning over expressions (5.15) and (5.16).

□

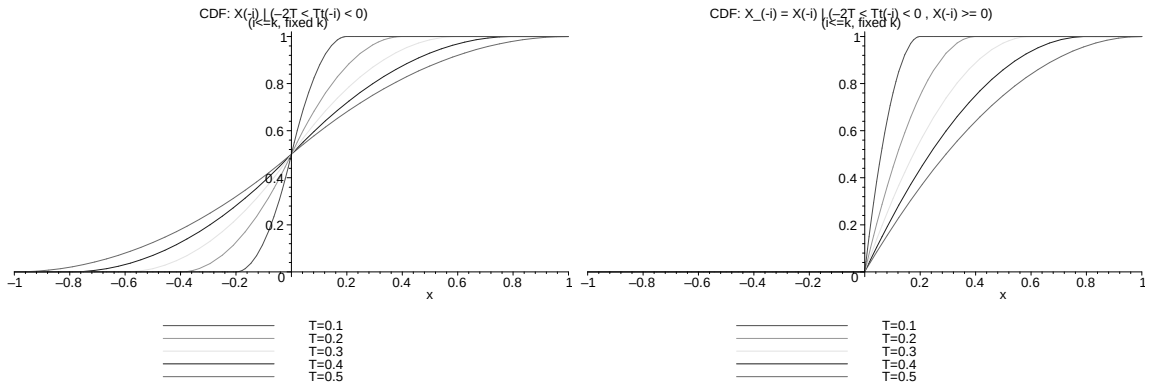


Figure 5.10: CDF of **(a)** $X(-i)|(-2T \leq T_{t(-i)} < 0)$, **(b)** $\overline{X}(-i)$.

Packet received at $t = 0$

Packet p is received deterministically in $t = 0$, and it is assigned a deterministic jitter T. For compatibility with the family of presented variables $\{X(i) : X(i) = T_{t_t(i)} + T_{t_j(i)}\}_{i \in \mathbb{Z}^*}$, this is

modeled as the random variable $X_0 = T$, with the following statistic description:

$$\begin{cases} f_{X_0}(x) &= \delta(x-T) \\ F_{X_0}(x) &= H(x-T) \end{cases} \tag{5.17}$$

5.3.4 Time to Transmission for a Received Message

This section addresses the average time to transmission for a message contained in a packet p that arrives to a router at time $t = 0$, assuming that the message is assigned a jitter value $T = E\{T_j\}$ ($T_j \sim Unif[0, 2T]$). The goal is to examine the impact of the jitter range $[0, 2T]$ in the average time to transmission for messages, when such messages have a jitter corresponding to the average jitter value (assuming a uniform distribution for the jitter value within the range).

Two cases are considered:

1. Upper bound: packet p arrives when no prior messages are waiting to be forwarded.

2. Lower bound: all the prior packets having arrived at $t < 0$ and scheduled to be sent after $t = 0$ are waiting in the router's queue.

In practice, message forwarding is between both extremes: it is possible that some of the messages received before $t = 0$ have not been sent at $t = 0$ (and thus their presence might reduce the time to transmission of the considered message below the upper bound), but it is also possible that some of such messages received before $t = 0$ have been forwarded in a prior transmission (for instance, due to the transmission of a self-generated message, or to the expiration of the jitter of another received in-message) – the time to transmission might be therefore higher than the lower bound.

In order to address the cases of self-generated messages, the following random variable is introduced to indicate the time (from $t = 0$) until the generation of the next self-message:

$$T_O \sim Uniform\left[0, \frac{1}{\lambda_g}\right] \tag{5.18}$$

PDF and CDF of T_O are described in (5.19).

$$\left\{ \begin{array}{lcl} f_{T_O}(x) & = & \left\{ \begin{array}{ll} \lambda_g & ,0 \leq x \leq \frac{1}{\lambda_g} \\ 0 & , \text{ otherwise} \end{array} \right. \\ F_{T_O}(x) & = & \left\{ \begin{array}{ll} 0 & ,x < 0 \\ x\lambda_g & ,0 \leq x < \frac{1}{\lambda_g} \\ 1 & ,x \geq \frac{1}{\lambda_g} \end{array} \right\} = x\lambda_g \left(H(x) - H\left(x - \frac{1}{\lambda_g}\right)\right) + H\left(x - \frac{1}{\lambda_g}\right) \end{array} \right. \tag{5.19}$$

The analysis for the lower and the upper bound is performed in two steps: in the first one, it is described the behavior of the time to transmission with respect to the number of packets arrived under the studied situation. In the second, the average time to transmission is computed considering all possible number of arrival events.

Upper bound

The upper bound corresponds to the situation in which the time to transmission for a message contained in packet p (received at $t = 0$) is only influenced by messages received in the router in the cases (i) and (iii) of the beginning of section 5.3, as illustrated in Figure 5.11.

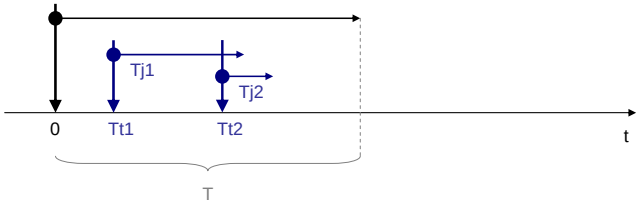


Figure 5.11: Illustration of the traffic model for packets containing messages to be forwarded, for the upper bound of time to transmission.

The impact of packet arrivals corresponding to (i) (see Figure 5.4) can be modeled by way of the following random variable:

$$M_k = min\{X(i)\}_{1 \leq i \leq k} \tag{5.20}$$

This random variable represents the minimum of the scheduled times for transmission of

packets arrived in $(0, T]$, assuming that k packets have been received. *Proposition 5.5* describes the CDF of M_k, and Figure 5.12.a illustrates the trace of the CDF, for $T = 0.1sec$ and different values of k.

Proposition 5.5. *The cumulative distribution function (CDF) of $M_k(T)$ is as follows:*

$$F_{M_k(T)}(x) = 1 - (1 - F_{\overline{X}}(x))^k \tag{5.21}$$

Proof. From the definition,

$$\begin{aligned} F_{M_k(T)}(x) &= P(M_k < x)|_T = 1 - \prod_{i=1}^{k} P(\overline{X(i)} > x) = 1 - \prod_{i=1}^{k} \Big(1 - P(\overline{X(i)} < x)\Big) = \\ &= 1 - \prod_{i=1}^{k} \Big(1 - F_{\overline{X(i)}}(x)\Big) = 1 - (1 - F_{\overline{X}}(x))^k \end{aligned}$$

□

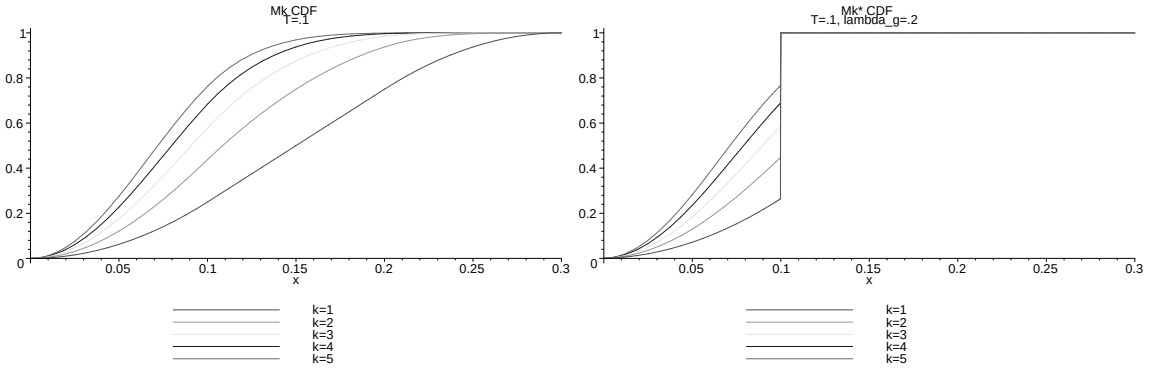


Figure 5.12: Cumulative distribution function (CDF) for **(a)** $M_k(T)$, for $T = 0.1$ and different values of k, and **(b)** $M_k^*(T)$, for $T = 0.1$, $\lambda_g = 0.2$ and different values of k.

The impact of case (iii) in the retransmission of messages contained in in-packet p (see Figure 5.4) is studied by considering the possibility that the router self-generates and transmits a message within the interval $(0, T]$. In case that no prior transmission is scheduled, messages contained in p are sent deterministically at $t = T$. The generalized random variable $M_k^*(T)$ takes these additional phenomena into consideration.

$$M_k^*(T) = min\{T_O, X_0, \{X(i)\}_{1 \le i \le k}\} = min\{T_O, \{X(i)\}_{0 \le i \le k}\} \tag{5.22}$$

where T_O is the time that the next self-originated message is generated, as defined in (5.19).

Theorem 5.6 describes the CDF of random variable $M_k^*(T)$. The trace of the CDF is displayed in Figure 5.12.b, for different values of k.

Theorem 5.6. *The cumulative distribution function (CDF) of M_k^* is as follows:*

$$F_{M_k^*(T)}(x) = 1 - (1 - H(x-T))\,(1 - \lambda_g x)\,(1 - F_{\overline{X}}(x))^k \tag{5.23}$$

Proof. From the definition,

$$\begin{aligned} F_{M_k^*(T)}(x) &= 1 - (1 - H(x-T))P(T_O > x)\prod_{i=1}^{k} P(\overline{X(i)} > x) = \\ &= 1 - (1 - H(x-T))\,(1 - \lambda_g x)\prod_{i=1}^{k}\left(1 - P(\overline{X(i)} < x)\right) = \\ &= 1 - (1 - H(x-T))\,(1 - \lambda_g x)\prod_{i=1}^{k}\left(1 - F_{\overline{X(i)}}(x)\right) = \\ &= 1 - (1 - H(x-T))\,(1 - \lambda_g x)\,(1 - F_{\overline{X}}(x))^k \end{aligned}$$

□

The upper bound for the time to transmission of a message contained in in-packet p (received in $t = 0$), scheduled to be sent in $t = T$, can be therefore modeled as follows:

$$T_{tx}(T)^{\text{upper}} = \sum_{k=0}^{\infty} f(k;\lambda_{in}) M_k^*(T) \tag{5.24}$$

Proposition 5.7 describes the CDF of the time to transmission $T_{tx}(T)^{\text{upper}}$. Figure 5.13 illustrates the CDF of $T_{tx}(T)^{\text{upper}}$, for $T = 0.1sec$, $\lambda_{in} = 4\frac{pkt}{sec}$ and different values of λ_g.

Proposition 5.7. *The cumulative distribution function (CDF) of $T_{tx}(T)^{upper}$ is as follows:*

$$\begin{aligned} F_{T_{tx}(T)^u}(x) &= e^{-\lambda_{in}T}\left[(1 - (1 - H(x-T))\,(1 - \lambda_g x)) + \right. \\ &\quad \left. + \sum_{k=1}^{\infty} \frac{(\lambda_{in}T)^k}{k!}\left(1 - (1 - H(x-T))\,(1 - \lambda_g x)\,(1 - F_{\overline{X}}(x))^k\right)\right] \end{aligned} \tag{5.25}$$

And the mean of $T_{tx}(T)^{upper}$:

$$\begin{aligned} \mathbb{E}\{T_{tx}(T)^{upper}\} &= \sum_{k=0}^{\infty} f(k;\lambda_{in},T)\mathbb{E}\{M_k^*(T)\} = \\ &= f(0;\lambda_{in},T)\mathbb{E}\{M_0^*(T)\} + \sum_{k=1}^{\infty} f(k;\lambda_{in},T)\mathbb{E}\{M_k^*(T)\} \end{aligned} \tag{5.26}$$

where

$$\begin{cases} \mathbb{E}\{M_0^*(T)\} &= T - \frac{1}{2}\lambda_g T^2 \\ \mathbb{E}\{M_k^*(T)\}_{k>0} &= T(1-\lambda_g T)\left(1-F_{\overline{X}}(T)\right)^k + \int_0^T x\lambda_g \left(1-F_{\overline{X}}(x)\right)^k dx + \\ & \quad + \int_0^T xk f_{\overline{X}}(x)\left(1-\lambda_g x\right)\left(1-F_{\overline{X}}(x)\right)^{k-1} dx \end{cases}$$

Proof. From *Proposition 5.6*, the cumulative distribution function (CDF) of $M_k^*(T)$ is:

$$F_{M_k^*(T)}(x) = 1 - (1-H(x-T))\left(1-\lambda_g x\right)\left(1-F_{\overline{X}}(x)\right)^k$$

Therefore, the expression of the CDF of $T_{tx}(T)^{\text{upper}}$ is as follows:

$$\begin{aligned} F_{T_{tx}(T)^{\text{u}}}(x) &= \sum_{k=0}^{\infty} f(k;\lambda_{in},T) F_{M_k^*(T)}(x) = f(0;\lambda_{in},T)\left(1-(1-H(x-T))\left(1-\lambda_g x\right)\right) + \\ &\quad + \sum_{k=1}^{\infty} f(k;\lambda_{in},T)\left(1-(1-H(x-T))\left(1-\lambda_g x\right)\left(1-F_{\overline{X}}(x)\right)^k\right) \end{aligned}$$

Applying (5.4),

$$\begin{aligned} F_{T_{tx}(T)^{\text{u}}}(x) &= e^{-\lambda_{in}T}\left[\left(1-(1-H(x-T))\left(1-\lambda_g x\right)\right) + \right. \\ &\quad \left. + \sum_{k=1}^{\infty} \frac{(\lambda_{in}T)^k}{k!}\left(1-(1-H(x-T))\left(1-\lambda_g x\right)\left(1-F_{\overline{X}}(x)\right)^k\right)\right] \end{aligned}$$

Expression (5.26) is direct from (5.24). From *Theorem 5.6*, the cumulative distribution function (CDF) of $M_k^*(T)$ is:

$$F_{M_k^*(T)}(x) = 1 - (1-H(x-T))\left(1-\lambda_g x\right)\left(1-F_{\overline{X}}(x)\right)^k$$

Consequently, the probability density function (PDF) is, for $k > 0$:

$$\begin{aligned} f_{M_k^*(T)}(x) &= \frac{d}{dx}F_{M_k^*}(x) = \frac{d}{dx}\left(1-(1-H(x-T))\left(1-\lambda_g x\right)\left(1-F_{\overline{X}}(x)\right)^k\right) = \\ &= -(-\delta(x-T))\left(1-\lambda_g x\right)\left(1-F_{\overline{X}}(x)\right)^k - (1-H(x-T))(-\lambda_g)\left(1-F_{\overline{X}}(x)\right)^k - \\ &\quad -(1-H(x-T))\left(1-\lambda_g x\right)k\left(1-F_{\overline{X}}(x)\right)^{k-1}\left(-f_{\overline{X}}(x)\right) = \\ &= \delta(x-T)\left(1-\lambda_g x\right)\left(1-F_{\overline{X}}(x)\right)^k + \lambda_g(1-H(x-T))\left(1-F_{\overline{X}}(x)\right)^k + \\ &\quad k f_{\overline{X}}(x)(1-H(x-T))\left(1-\lambda_g x\right)\left(1-F_{\overline{X}}(x)\right)^{k-1} \end{aligned}$$

and for $k = 0$:

$$\begin{aligned} f_{M_0^*}(x) &= \frac{d}{dx} F_{M_0^*}(x) = \frac{d}{dx} \left(1 - (1 - H(x-T))\,(1 - \lambda_g x)\right) = \\ &= \delta(x-T)\,(1-\lambda_g x) + \lambda_g (1 - H(x-T)) \end{aligned}$$

The mean for variables $M_k^*(T)$ can be computed as follows:

$$\begin{aligned} \mathbb{E}\{M_0^*(T)\} &= \int_{-\infty}^{\infty} x \left(\delta(x-T)\,(1-\lambda_g x) + \lambda_g (1 - H(x-T))\right) dx = \\ &= \frac{1}{2}\lambda_g T^2 - \lambda_g T = T - \frac{1}{2}\lambda_g T^2 \end{aligned}$$

$$\begin{aligned} \mathbb{E}\{M_k^*(T)\}_{k>0} &= \int_{-\infty}^{\infty} x \Big(\delta(x-T)\,(1-\lambda_g x)\;\left(1 - F_{\overline{X}}(x)\right)^k + \lambda_g (1 - H(x-T))\,(1 - F_{\overline{X}}(x))^k + \\ &\quad + k f_{\overline{X}}(x)(1 - H(x-T))\,(1-\lambda_g x)\;\left(1 - F_{\overline{X}}(x)\right)^{k-1}\Big)\, dx = \\ &= T(1-\lambda_g T)\,(1 - F_{\overline{X}}(T))^k + \int_0^T x\lambda_g \left(1 - F_{\overline{X}}(x)\right)^k dx + \\ &\quad + \int_0^T x k f_{\overline{X}}(x)\,(1-\lambda_g x)\;\left(1 - F_{\overline{X}}(x)\right)^{k-1} dx \end{aligned}$$

□

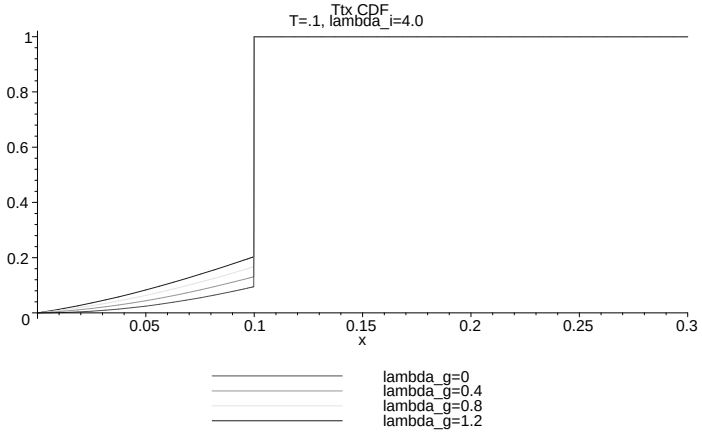


Figure 5.13: Cumulative distribution function (CDF) for the upper bound of $T_{tx}(T)$, for different values of λ_g.

The case described for computing the upper bound of the time to transmission of a packet p with jitter value T ($T = \mathbb{E}\{T_j\}$, $T_j \sim Uniform[0, 2T]$), can be generalized to study the average

duration of the interval between the first in-packet arrival after a retransmission and the following retransmission – which corresponds to the average length of the phase in which the node accumulates self- and in-messages before forwarding them in out-packets. *Theorem 5.8* describes the average duration of this accumulating phase, denoted by D, depending on the jitter value t of the first arrived in-packet and the maximum jitter value, previously denominated as $MAXJITTER$ and denoted in this section as J_m for simplicity reasons. Jitter values are thus selected randomly within $[0, J_m]$. Relationship between the accumulating phase $D(t)$ and random variable $T_{tx}^{upper}(T)$ is then presented.

Theorem 5.8. *Let $D(t)$ be the average duration of the accumulating phase (*i.e.*, the interval between the arrival of the first in-packet after a retransmission, and the next retransmission), with $t \in [0, J_m]$ being the scheduled time of retransmission of such first in-packet and J_m being the maximum jitter value. Let λ_{in} be the Poisson arrival rate of in-packets, and λ_g the Poisson generation rate of self-generated packets. Then, if the jitter is selected following an uniform distribution $T_j \sim Uniform[0, J_m]$, the expression of $D(t)$ is as follows:*

$$D(t) = \sqrt{\frac{J_m}{2\lambda_i}\pi}\left[Erf\left(\sqrt{\frac{J_m}{2\lambda_{in}}}\left(\lambda_g + \frac{\lambda_{in}t}{J_m}\right)\right) - Erf\left(\sqrt{\frac{J_m}{2\lambda_{in}}}\lambda_g\right)\right] e^{-\frac{\lambda_g^2}{2\lambda_i}J_m} \tag{5.27}$$

Where $Erf(.)$ denotes the error function, defined as follows:

$$Erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-s^2} ds \tag{5.28}$$

Proof. Given a scheduled jitter value t for the first in-packet, the effect of events happening in dt in the average duration D is examined. For sufficiently small values of dt, only one Poisson event (an in-packet arrival, with rate λ_{in}; or a self-generated packet, with rate λ_g) may occur. An in-packet arrival at dt (with probability $\lambda_{in}dt$) may modify the duration $D(t)$ if the scheduled jitter T_j of the arrived packet is lower than the scheduled time of retransmission t; a self-generated packet arrival within at dt (with probability $\lambda_g dt$) implies that the duration $D(t)$ becomes equivalent to the duration of the phase for a scheduled time dt. When no in- or self-packets arrive at dt, duration $D(t)$ is equivalent to the duration obtained by waiting a dt interval and then scheduling retransmission after an interval $t - dt$. This is described formally in the following transition equation:

$$\begin{aligned} D(t) &= \lambda_{in} dt \left(P(T_j > t) D(t) + \int_0^t f_{T_j}(x) D(x) dx \right) + \\ &+ \lambda_g dt D(dt) + (1 - (\lambda_{in} + \lambda_g) dt)(D(t - dt) + dt) \end{aligned}$$

Then,

$$\begin{aligned} D(t) - D(t - dt) &= \lambda_{in} dt \left(P(T_j > t) D(t) + \int_0^t f_{T_j}(x) D(x) dx \right) + \\ &+ \lambda_g dt D(dt) + dt - (\lambda_{in} + \lambda_g) dt (D(t - dt) + dt) \end{aligned}$$

And dividing over dt,

$$\begin{aligned} \frac{D(t) - D(t - dt)}{dt} &= \lambda_{in} \left(P(T_j > t) D(t) + \int_0^t f_{T_j}(x) D(x) dx \right) + \\ &+ \lambda_g D(dt) + 1 - (\lambda_{in} + \lambda_g)(D(t - dt) + dt) \end{aligned}$$

For $dt \longrightarrow 0$, and taking into account that $D(dt \longrightarrow 0) \longrightarrow 0$ by definition of D, the following differential equation arises:

$$\begin{aligned} D'(t) &= \lambda_{in} \left(P(T_j > t) D(t) + \int_0^t f_{T_j}(x) D(x) dx \right) - (\lambda_{in} + \lambda_g) D(t) + 1 = \\ &= \lambda_{in} \left((P(T_j > t) - 1) D(t) + \int_0^t f_{T_j}(x) D(x) dx \right) - \lambda_g D(t) + 1 = \\ &= \lambda_{in} \left(-F_{T_j}(t) D(t) + \int_0^t f_{T_j}(x) D(x) dx \right) - \lambda_g D(t) + 1 \end{aligned}$$

As $F_X(t) = \int_0^t f_X(x) dx$,

$$D'(t) = \lambda_{in} \int_0^t f_{T_j}(x)(D(x) - D(t)) dx - \lambda_g D(t) + 1$$

Differentiating this expression over t:

$$D"(t) = \frac{d}{dt} \left[\lambda_{in} \int_0^t f_{T_j}(x)(D(x) - D(t)) dx \right] - \lambda_g D'(t) \tag{5.29}$$

Where the derivative in brackets, denoted I_1, can be calculated as follows:

$$
\begin{aligned}
I_1 &= \frac{d}{dt}\left[\lambda_{in}\int_0^t f_{T_j}(x)(D(x)-D(t))dx\right] = \\
&= \frac{d}{dt}\left[\lambda_{in}\left(\int_0^t f_{T_j}(x)D(x)dx - \int_0^t f_{T_j}(x)D(t))dx\right)\right] = \\
&= \lambda_{in}\frac{d}{dt}\left[\int_0^t f_{T_j}(x)D(x)dx\right] - \lambda_{in}\frac{d}{dt}\left[D(t)\int_0^t f_{T_j}(x)dx\right] = \\
&= \lambda_{in}f_{T_j}(t)D(t) - \lambda_{in}\frac{d}{dt}\left[D(t)F_{T_j}(t)\right] = \\
&= \lambda_{in}f_{T_j}(t)D(t) - \lambda_{in}D'(t)F_{T_j}(t) - \lambda_{in}D(t)f_{T_j}(t) = \\
&= -\lambda_{in}D'(t)F_{T_j}(t)
\end{aligned}
$$

Then, replacing I_1 in equation (5.29), an ordinary differential equation (ODE) of order 2:

$$D"(t) = -(\lambda_{in}F_{T_j}(t) + \lambda_g)D'(t)$$

Imposing initial conditions $D(0) = 0$, $D'(0) = 1$, and assuming an uniform distribution for jitter values ($f_{T_j}(t) = \frac{1}{J_m}$ for $t \in [0, J_m]$), this ODE has the following solution:

$$D(t) = \sqrt{\frac{J_m}{2\lambda_i}}\pi\left[Erf\left(\sqrt{\frac{J_m}{2\lambda_{in}}}\left(\lambda_g + \frac{\lambda_{in}t}{J_m}\right)\right) - Erf\left(\sqrt{\frac{J_m}{2\lambda_{in}}}\lambda_g\right)\right]e^{-\frac{\lambda_g^2}{2\lambda_i}J_m}$$

Where $Erf(.)$ denotes the error function of equation (5.28).

□

It is worth to observe that the closed expression (5.27) provides a generalization of the mean of the upper bound of the time of transmission for an in-packet p, as described in *Proposition 5.7*. Assuming $t = T$ and $J_m = 2T$, equation (5.27) becomes:

$$D(T) = \sqrt{\frac{T}{\lambda_i}}\pi\left[Erf\left(\sqrt{\frac{T}{\lambda_{in}}}\left(\lambda_g + \frac{\lambda_{in}}{2}\right)\right) - Erf\left(\sqrt{\frac{T}{\lambda_{in}}}\lambda_g\right)\right]e^{-\frac{\lambda_g^2}{\lambda_i}T} \tag{5.30}$$

Which is a closed expression equivalent to equation (5.26) from *Proposition 5.7*.

Lower bound

The lower bound corresponds to the situation in which the time to transmission for a message contained in in-packet p (received at $t = 0$) is influenced not only by messages in cases (i)

and (iii), but also (ii), *i.e.*, by all messages received at $t < 0$ and scheduled to be sent at $t > 0$. In order to study such situation, the analysis considers not only the arrivals of packets after $t = 0$ (with scheduled times $X(i)$, for $i > 0$), but also those received before $t = 0$ but scheduled for $t > 0$ (with scheduled times $X(-i)$, for $i > 0$). The random variables $M_k(T)$ and $M_k^*(T)$ are generalized as follows:

$$M_{k,l}(T) = min\{X(i)\}_{-k \leq i \leq l, i \neq 0} \tag{5.31}$$

The random variable $M_{k,l}^*(T)$ extends naturally from $M_{k,l}(T)$:

$$M_{k,l}^*(T) = min\{T_O, X_0, \{X(i)\}_{-k \leq i \leq l, i \neq 0}\} = min\{T_O, \{X(i)\}_{-k \leq i \leq l}\} \tag{5.32}$$

Proposition 5.9 indicates the expression for the CDF of this random variable. Figure 5.14 displays the trace of the CDF of $M_{k,l}^*$, for pairs (k, l) with $0 \leq k \leq 4$, $0 \leq l \leq 4$.

Proposition 5.9. *The cumulative distribution function (CDF) of $M_{k,l}^*(T)$ is as follows:*

$$F_{M_{k,l}^*(T)}(x) = 1 - (1 - \lambda_g x)(1 - F_{\overline{X}(-i)}(x))^k (1 - H(x - T))(1 - F_{X_i}(x))^l \tag{5.33}$$

Proof.

$$\begin{aligned} F_{M_{k,l}^*(T)}(x) &= 1 - (1 - F_{T_O}(x))(1 - F_{\overline{X}(-i)}(x))^k (1 - F_{X_0}(x))(1 - F_{X_i}(x))^l = \\ &= 1 - (1 - \lambda_g x)(1 - F_{\overline{X}(-i)}(x))^k (1 - H(x - T))(1 - F_{X_i}(x))^l \end{aligned}$$

□

Theorem 5.10 defines the random variable that corresponds to the lower bound for the time of transmission of a message contained in a packet received in $t = 0$, scheduled to be sent in $t = T$; the theorem also describes its CDF and its mean. Figure 5.15 illustrates the CDF of $T_{tx}(T)^{lower}$, with $T = 0.1sec$, $\lambda_{in} = 4\frac{pkt}{sec}$, for different values of λ_g.

Theorem 5.10. *The random variable for the lower bound of the time to transmission, $T_{tx}(T)^{lower}$, is as follows:*

$$T_{tx}(T)^{lower} = e^{-2\lambda_{in}T} \left(min\{T_O, T\} + \sum_{m=1}^{\infty} \frac{(\lambda_{in}T)^m}{m!} (M_{0,m}^* + M_{m,0}^*) + \sum_{k=1}^{\infty} \sum_{l=1}^{\infty} \frac{(\lambda_{in}T)^{k+l}}{k!l!} M_{k,l}^* \right) \tag{5.34}$$

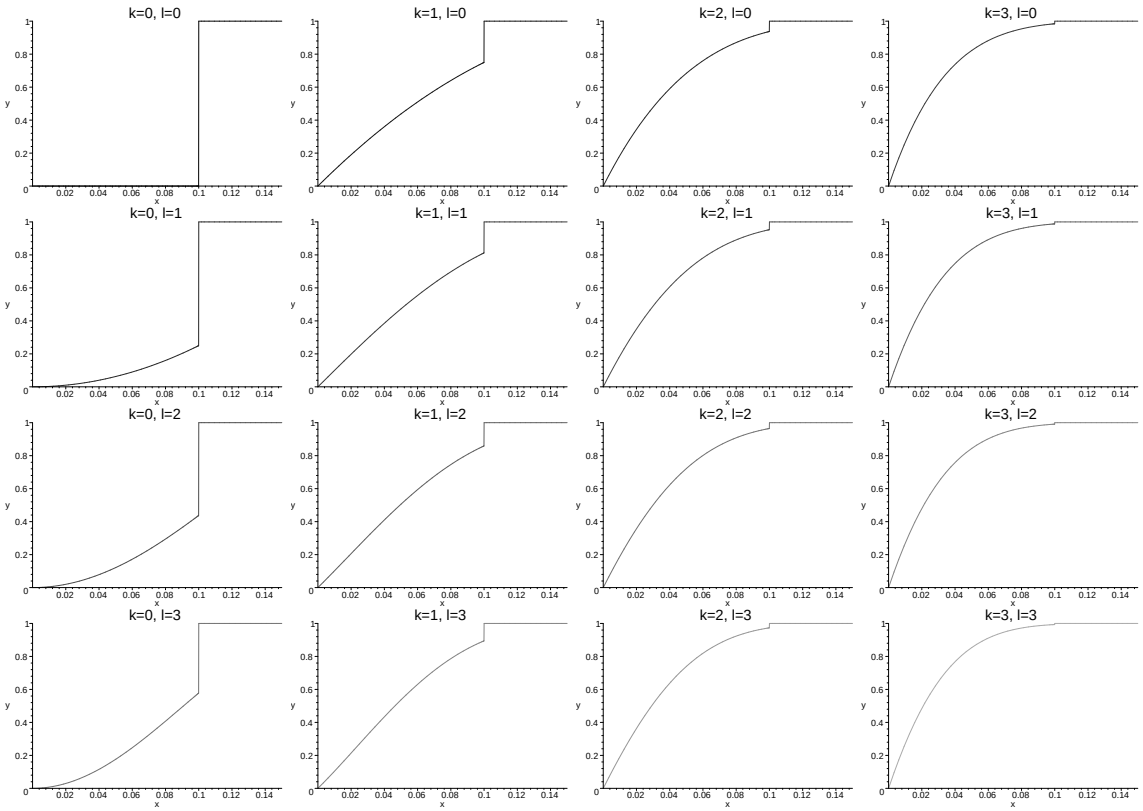


Figure 5.14: CDF of $M^*_{k,l}$, for different pairs (k,l) $(0 \leq k \leq 3,\ 0 \leq l \leq 3)$, for $T = 0.1sec$, $\lambda_{in} = 4\frac{pkt}{sec}$, $\lambda_g = 0.2\frac{pkt}{sec}$.

The cumulative distribution function (CDF) for $T_{tx}(T)^{lower}$ is as follows:

$$\begin{aligned} F_{T_{tx}(T)^l}(x) \quad = \quad & e^{-2\lambda_{in}T}\left(1-(1-\lambda_g x)(1-H(x-T))+\right. \\ & + \sum_{m=1}^{\infty}\frac{(\lambda_{in}T)^m}{m!}(2-(1-\lambda_g x)(1-H(x-T))((1-F_{X(i)}(x))^m+(1-F_{\overline{X}(-i)}(x))^m))+ \\ & \left.+\sum_{k=1}^{\infty}\sum_{l=1}^{\infty}\frac{(\lambda_{in}T)^{k+l}}{k!l!}(1-(1-\lambda_g x)(1-F_{\overline{X}(-i)}(x))^k(1-H(x-T))(1-F_{X(i)}(x))^l)\right) \end{aligned} \tag{5.35}$$

The mean of the random variable $T_{tx}(T)^{lower}$ has the following expression:

$$\begin{aligned} \mathbb{E}\{T_{tx}(T)^{lower}\} \quad = \quad & e^{-2\lambda_{in}T}\left[\mathbb{E}\{min\{T_O,T\}\}+\sum_{m=1}^{\infty}\frac{(\lambda_{in}T)^m}{m!}\left(\mathbb{E}\{M^*_{0,m}(T)\}+\mathbb{E}\{M^*_{m,0}(T)\}\right)+\right. \\ & \left.\sum_{k=1}^{\infty}\sum_{l=1}^{\infty}\frac{(\lambda_{in}T)^k+l}{k!l!}\mathbb{E}\{M^*_{k,l}(T)\}\right] \end{aligned} \tag{5.36}$$

where

$$\begin{cases} \mathbb{E}\{min\{T_O,T\}\} &= T-\frac{1}{2}T^2\lambda_g \\ \mathbb{E}\{M^*_{0,m}(T)\}_{m>0} &= \lambda_g\int_0^T x\left(1-\frac{x}{T}+\frac{x^2}{4T^2}\right)^m dx+\int_0^T\frac{x^2}{2T^2}\left(1-\frac{x}{T}+\frac{x^2}{4T^2}\right)^{m-1}(1-\lambda_g x)dx+ \\ & \quad +T(1-\lambda_g T)\left(\frac{3}{4}\right)^m \\ \mathbb{E}\{M^*_{m,0}(T)\}_{m>0} &= \lambda_g\int_0^T x\left(1-\frac{x}{T}+\frac{x^2}{4T^2}\right)^m dx+m\int_0^T(1-\lambda_g x)\left(1-\frac{x}{T}+\frac{x^2}{4T^2}\right)^{m-1}\left(\frac{1}{T}-\frac{x}{2T^2}\right)dx+ \\ & \quad +T(1-\lambda_g T)\left(\frac{1}{4}\right)^m \\ \mathbb{E}\{M^*_{k,l}(T)\}_{k,l>0} &= \lambda_g\int_0^T x\left(1-\frac{x}{T}+\frac{x^2}{4T^2}\right)^k\left(1-\frac{x^2}{4T^2}\right)^l dx+ \\ & \quad +k\int_0^T x\left(\frac{1}{T}-\frac{x}{2T^2}\right)(1-\lambda_g x)\left(1-\frac{x}{T}+\frac{x^2}{4T^2}\right)^{k-1}\left(1-\frac{x^2}{4T^2}\right)^l dx+ \\ & \quad +T(1-\lambda_g T)\left(\frac{1}{4}\right)^k\left(\frac{3}{4}\right)^l+l\int_0^T\frac{x^2}{2T^2}(1-\lambda_g x)\left(1-\frac{x}{T}+\frac{x^2}{4T^2}\right)^k\left(1-\frac{x^2}{4T^2}\right)^{l-1}dx \end{cases}$$

Proof. Considering all possible cases for k and l,

$$\begin{aligned} T_{tx}(T)^{\text{lower}} &= \sum_{k=0}^{\infty}\sum_{l=0}^{\infty} f_-\left(k;\frac{\lambda_{in}}{2},2T\right) f_+(l;\lambda_{in},T)M^*_{k,l}(T) = \\ &= f_-\left(0;\frac{\lambda_{in}}{2},2T\right) f_+(0;\lambda_{in},T)M^*_{0,0}(T)+f_-\left(0;\frac{\lambda_{in}}{2}\right)\sum_{l=1}^{\infty} f_+(l;\lambda_{in},T)M^*_{0,l}(T)+ \\ &\quad +f_+(0;\lambda_{in},T)\sum_{k=1}^{\infty} f_-\left(k;\frac{\lambda_{in}}{2},2T\right)M^*_{k,0}(T)+\sum_{k=1}^{\infty}\sum_{l=1}^{\infty} f_-\left(k;\frac{\lambda_{in}}{2},2T\right) f_+(l;\lambda_{in},T)M^*_{k,l}(T) \end{aligned}$$

where f_- and f_+ correspond to the function described in (5.4). Subindexes $+$ and $-$ are used to distinguish between the Poisson processes for arrivals after $t=0$ (f_+) and before $t=0$ (f_-). Note that $M^*_{0,0}(T)=min\{T_O,X_0\}=min\{T_O,T\}$. Therefore,

$$\begin{aligned} T_{tx}(T)^{\text{lower}} &= e^{-\frac{\lambda_{in}}{2}2T}e^{-\lambda_{in}T}min\{T_O,T\}+e^{-\frac{\lambda_{in}}{2}2T}\sum_{l=1}^{\infty}e^{-\lambda_{in}T}\frac{(\lambda_{in}T)^l}{l!}M^*_{0,l}+ \\ &\quad +e^{-\lambda_{in}T}\sum_{k=1}^{\infty}e^{-\frac{\lambda_{in}}{2}2T}\frac{((\lambda_{in}/2)2T)^k}{k!}M^*_{k,0}+\sum_{k=1}^{\infty}\sum_{l=1}^{\infty}e^{-\frac{\lambda_{in}}{2}2T}\frac{((\lambda_{in}/2)2T)^k}{k!}e^{-\lambda_{in}T}\frac{(\lambda_{in}T)^l}{l!}M^*_{k,l}= \\ &= e^{-2\lambda_{in}T}\left(min\{T_O,T\}+\sum_{l=1}^{\infty}\frac{(\lambda_{in}T)^l}{l!}M^*_{0,l}+\sum_{k=1}^{\infty}\frac{(\lambda_{in}T)^k}{k!}M^*_{k,0}+\sum_{k=1}^{\infty}\sum_{l=1}^{\infty}\frac{(\lambda_{in}T)^{k+l}}{k!l!}M^*_{k,l}\right)= \\ &= e^{-2\lambda_{in}T}\left(min\{T_O,T\}+\sum_{m=1}^{\infty}\frac{(\lambda_{in}T)^m}{m!}(M^*_{0,m}+M^*_{m,0})+\sum_{k=1}^{\infty}\sum_{l=1}^{\infty}\frac{(\lambda_{in}T)^{k+l}}{k!l!}M^*_{k,l}\right) \end{aligned}$$

The CDF of $T_{tx}(T)^{\text{lower}}$, $F_{T_{tx}(T)^{\text{l}}}(x)$, is computed by applying expression (5.33), that describes the CDF of $M^*_{k,l}$ for any combination (k,l), over equation (5.34). The average of random variable $T_{tx}(T)^{\text{lower}}$ is computed by using standard algebra.

□

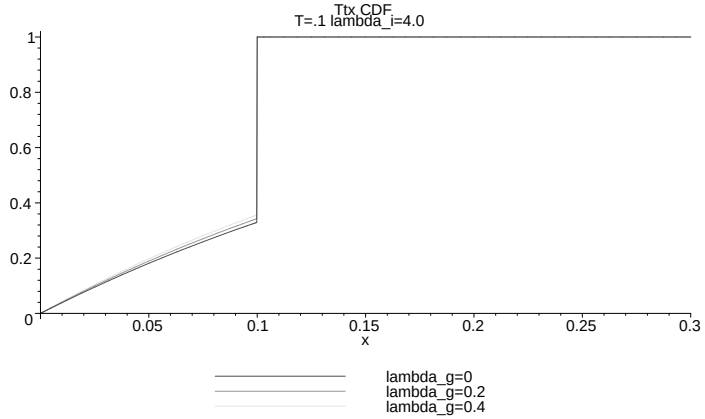


Figure 5.15: Cumulative distribution function (CDF) for the lower bound of $T_{tx}(T)$, for different values of λ_g.

5.3.5 Discussion of Results and Model Limitations

The analytical model presented in this section has presented two main results:

- **Impact of jittering in the rate of transmitted packets by a single interface.** This has been modeled by studying the out-packet rate λ_{out} and its relationship with variables λ_{in} (in-packet rate), λ_g (self-generated packet rate) and T^*. This last variable is a random variable describing, for a forwarded packet (out-packet), the time between the arrival of the first in-packet included in such out-packet, and the time at which the out-packet is forwarded. The expression of $\lambda_{out} \equiv \lambda_{out}(\lambda_{in}, \lambda_g, T^*)$ is detailed in equation (5.7) and proved in *Theorem 5.1.*

- **Delay introduced by jittering in interface forwarding.** The random variable T_{tx} of an in-message m describes the time interval between the arrival of such message in an in-packet and the time at which such message is transmitted through the following out-packet, assuming that the in-message is assigned a jitter value T. The model gives upper and lower bounds of T_{tx}, and presents closed forms for their means in *Proposition 5.7*, *Theorem 5.8* and *Theorem 5.10.* The upper bound models the case in which the in-message m arrives when no previous in-messages are waiting to be forwarded. The lower bound models the case in which all in-messages received before m, and scheduled to be forwarded after the arrival of m, have not been sent when m arrives. Means of both T_{tx} bounds depend on variables λ_{in}, λ_g and the

average jitter value T (jitter values assumed uniformly distributed in $[0, 2T]$).

It is worth noting that variable T^*, used for determining λ_{out}, corresponds to the average of the upper bound of T_{tx} when $\lambda_g = 0$, that is, in the absence of self-generated packets, for all possible values of the jitter within $[0, 2T]$.

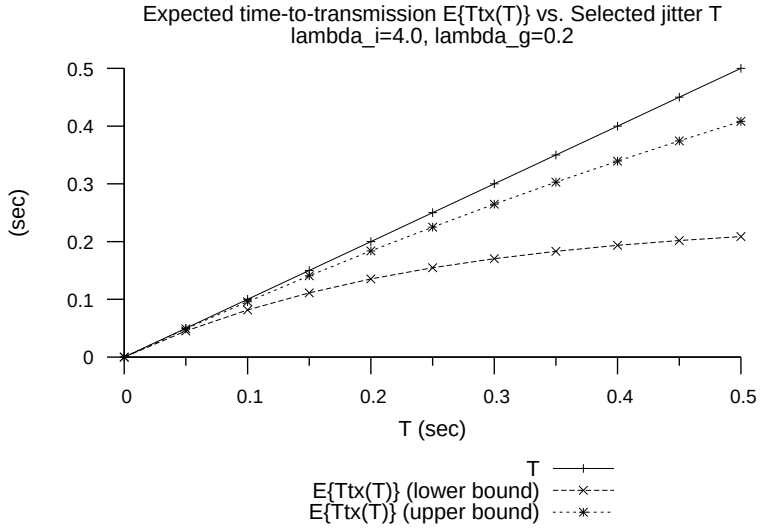


Figure 5.16: Lower and upper bounds for $\mathbb{E}\{T_{tx}(T)\}$.

Figure 5.16 displays the average of upper and lower bounds of T_{tx} for an interface with in-packet traffic rate $\lambda_{in} = 4\frac{pkts}{sec}$ and self-packet traffic rate $\lambda_g = 0.2\frac{pkts}{sec}$, when jitter values are selected within $[0, 2T]$, for different values of T.

These results are valid under the assumptions stated in section 5.3.1. The model that results from these assumptions is therefore limited in the following aspects:

- Packet arrival is modeled as a Poisson punctual homogeneous process. In particular, this implies that received in-packets do not cause collisions, as arrivals occur at different times and packet transmissions do not overlap. In practice, packet transmissions have a non-zero duration and the reception of such packets over an interface may be impossible if they overlap in time.

- The use of jitter enables an interface to achieve an out-packet rate lower than the in-packet rate,

by way of sending in-messages of several in-packets in a single out-packet. This is, however, at the expense of increasing the length of out-packets (which is considered to be negligible in the model): the less packets, the more messages per packet, given that the relationship between in-message rate γ_{in}, out-message rate γ_{out}, and self-message rate γ_g, is subject to equation (5.5). The longer an out-packet is, the more likely is that its transmission causes a collision with another packet in the network – reduction of wireless collisions is one of the main objectives of jittering, as stated in section 5.2. Whereas this aspect is not considered in the presented model, it needs to be taken into consideration, together with the forwarding delay (T_{tx}) and the packet rate reduction (λ_{out} vs. λ_{in}), in the design and evaluation of jittering.

This theoretical analysis may be extended and completed in some additional ways. An accurate (non based on upper and lower bounds) description of random variable T_{tx} could be explored, not only depending on $T = \mathbb{E}\{T_j\}$, with $T_j \sim [0, 2T]$, but also depending on an arbitrary $t \in [0, 2T]$. Efforts in this direction would follow the differential approach used for computing the average duration of the interval between the first in-packet arrived after a retransmission and the next retransmission of a node (see *Theorem 5.8*).

The interface-centric model described in this chapter should also be generalized to a network-based dynamic model, able to track the interaction between interfaces using the same jittering configuration and to evaluate the impact of jitter in the properties of the overall flooding traffic.

5.4 Simulations

This section provides supporting evidence, obtained by way of simulations, to the model presented in section 5.3. These simulations focus on the two main results of the model: the delay introduced by jitter in packet forwarding, and the relationship between out-packet traffic rate, self-packet traffic rate and in-packet traffic rate when jittering is used. In-packet arrivals and self-packet generation are modeled as Poisson processes, according to the traffic model, out-packet transmis-

sions are scheduled according to the forwarding algorithm with jitter (Figure 5.3) and jitter impact is measured by way of a discrete-event simulator implemented in Maple. Packet receptions and transmissions are assumed to be punctual events. Presented results are averaged over 30 iterations per value.

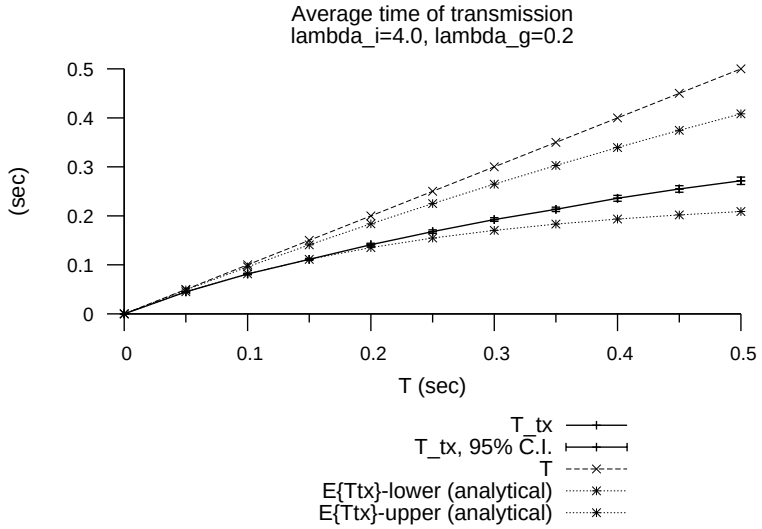


Figure 5.17: Average time between in-message arrival and forwarding (time to transmission, T_{tx}) *vs.* average jitter value (T), for $\lambda_{in} = 4\frac{pkt}{s}, \lambda_g = 0.2\frac{pkt}{s}$ (simulations and analytical results)..

Figure 5.17 presents the theoretical upper and lower bounds for the mean of T_{tx}, together with the averaged results from simulations. As expected, the average time between in-message arrivals and forwarding (T_{tx}) is always smaller than the average jitter value (T, included in Figure 5.17 for comparison), and the difference between both values grows bigger as T increases. Results from simulations fit in the interval defined by the two theoretical bounds ($\mathbb{E}\{T_{tx}^{upper}$ and $\mathbb{E}\{T_{tx}^{lower}$); they are significantly closer to the lower bound than to the upper bound. This suggests that the transmission time of in-messages is frequently determined by the jitter assigned to in-messages previously arrived, and the event that an in-packet arrival follows an out-packet transmission is rare. The probability of such an event may increase when in-packet traffic rates decrease, thus approaching the values of T_{tx} obtained by simulation to the upper theoretical bound of T_{tx}.

Figure 5.18 displays the in-packet and out-packet rates obtained in simulations for different values of T, with a nominal in-packet rate of $\lambda_{in} = 4\frac{pkts}{sec}$ and self-packet rate of $\lambda_g = 0.2\frac{pkts}{sec}$.

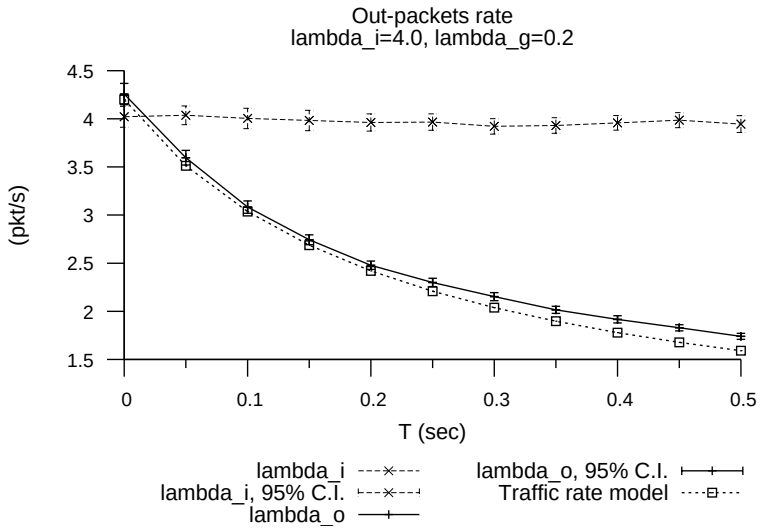


Figure 5.18: Out-packet (λ_{out}) and in-packet (λ_{in}) rates, for different values of T and a theoretical in-packet rate $\lambda_{in} = 4\frac{pkt}{s}$ (simulations).

Simulations are compared with the out-packet rate provided by the theoretical model via expression (5.7), assuming that T^* has deterministically the value of $\mathbb{E}\{T_{tx}^{upper}(T)|_{\lambda_g=0}\}$. As predicted, values from the model are slightly lower than those observed in the simulations, in particular due to the independence assumption on *Theorem 5.1*, and the difference tends to increase with higher values of T. It can be observed that the out-packet rate for $T = 0$ corresponds to $\lambda_{in} + \lambda_g = 4 + 0.2\frac{pkts}{sec} = 4.2\frac{pkts}{sec}$. For non-zero average values of jitter, the out-packet rate decreases significantly as T grows. The slope of this decrease becomes lower (in absolute terms) as T value is higher. Although the range of simulated T is not long enough, the observed evolution is consistent with the horizontal asymptote at $\lambda_{out} = \lambda_g = 0.2\frac{pkts}{sec}$, mentioned in section 5.3.1.

5.5 Conclusion

Some of the mechanisms used in link-state routing, such as topology flooding and neighbor sensing based on Hello exchange, may lead to wireless collisions when performed over MANETs. In topology flooding, packet collisions may occur when two neighboring interfaces receive a packet from a common neighbor and forward it immediately after the reception, or generate and flood a

packet at the same time in reaction to a common event. In neighbor sensing, periodic Hellos from two neighboring interfaces may cause collisions in every Hello transmission if both interfaces were switched on at the same time and use the same time interval between consecutive Hello packets.

Jittering addresses these issues by enabling each interface to distribute randomly over a time interval packet transmission events that are either periodic or in reaction to an external input (*e.g.*, a link failure or the arrival of a packet to be forwarded). Instead of sending such packets immediately, transmissions are delayed a random amount to time, denominated *jitter*, in order to reduce the probability of wireless collisions.

This chapter has focused in the use of jitter in packet forwarding, in the context of link-state flooding, as specified in [29]. Interfaces using jitter for flooding assign a random delay to each packet to be forwarded, and send all pending packets together when any of such delays expire. In case the interface generates a message to be flooded, it is sent immediately, together with all packets waiting to be forwarded. This jittering mechanism causes three effects over flooding traffic: (i) it delays the link-state flooding operation over the network, as any packet needs more time to reach all interfaces; (ii) reduces the packet rate, as several packets may be sent together in a single transmission, and (iii) increases the size of such packets, due to the same reason.

This chapter has provided an analytical model for the study of effects (i) and (ii) in a single interface. Two main results are obtained through this model: upper and lower bounds of the average delay before forwarding a received packet, and the reduction on the packet rate caused by the use of jitter. Both results are validated via simulations.

The analysis of the performance and effects of jittering needs to be developed beyond the results presented in this chapter. In order to explore (iii), the model should consider non-instantaneous packet transmissions. The three effects should be also studied in the whole network, thus extending the interface-based analysis presented in this chapter.

Chapter 6

Overlays in Link State Routing

Link-state routing in a network requires that three operations are performed over the network. These operations have been identified in chapter 4: selection of links to be advertised network-wide by way of topology advertisements, flooding of these advertisements over the network and LSDB synchronization between (a subset of) neighboring routers.

These operations can be treated and optimized separately. Separate analysis and optimization is useful for routing in ad hoc networks. Independent analysis permits handling efficiently the issues of wireless multi-hop communication, semibroadcast and dynamic topology, given that the implications of such aspects are different in each link-state operation.

6.1 Outline

This chapter introduces the concept of a *link-state overlay* as a tool for analysis of link-state routing characteristics in ad hoc networks, each link-state overlay being associated with a link-state operation. The chapter examines the impact of ad hoc networking issues on the characteristics and requirements for such overlays. Section 6.2 identifies the theoretical properties that overlays, associated with these operations, need to fulfill, and explores optimization objectives for each operation. The concept of overlay is used in the following chapters, which analyze different optimization

techniques for link-state routing in ad hoc networks. For comparison, section 6.3 presents an initial analytical evaluation of the performance of using all routers and all links in the network (full network overlay) in each of the studied operations. This corresponds to the usage of classic link-state routing mechanisms within wireless scenarios. Finally, section 6.4 concludes the chapter.

6.2 LS Routing in terms of Overlays

The three main operations of link-state routing in ad hoc networks can be reduced to overlay definition problems. Link-state overlays are defined as follows:

Definition 6.1 (**Link-state overlay**). A *link-state overlay* in a network is a set of routers and links of the network, used to perform a specific link-state operation to which the overlay is associated. Therefore, properties of link-state overlays are determined by the requirements of their associated link-state operations. A link-state overlay can be represented as a subgraph of the network graph, $S \subseteq G$, that contain a subset of vertices $V(S) \subseteq V(G)$ and a subset of links $E(S) \subseteq E(G)$.

In an ad hoc network, link-state routing operations are performed locally (independently by each router in the network) and thus, the associated overlays are built in a distributed fashion and may change dynamically during the network lifetime. This chapter examines the requirements that each link-state operation imposes on its associated overlay. Some of the studied properties are those that follow:

Definition 6.2 (**Asymptotic connection**). A link-state overlay defined over an ad hoc network is *asymptotically connected* if it is represented by a connected subgraph $S \subseteq G$, *i.e.*, if for each pair of vertices $x, y \in V(S)$ there exists a path p_{xy} within S.

Definition 6.3 (**Asymptotic dominance**). A link-state overlay defined over an ad hoc network is *asymptotically dominant* in the network if and only if its representation as a subgraph S is dominant in the network graph G, *i.e.*, if every vertex in G is either included in S or has a link to (at least) one vertex of G, *i.e.*, $V(S) = V(G)$.

Definition 6.4 (**Asymptotic spanning property**). A link-state overlay defined over an ad hoc network is a *asymptotically spanning* overlay in the network if its representation as a subgraph S includes all vertices in the network graph G.

The term *asymptotic* in these definitions means that the corresponding properties are attained in ideal conditions, in which packet losses $\longrightarrow 0$, transmission delays $\longrightarrow 0$ and probability of packet collision $\longrightarrow 0$. The fact that the subgraph representation of an overlay is connected, dominating or spanning in the network graph does not imply that, in practice, the overlay itself is connected, dominating or spanning at all times. Due to router mobility, delays in the exchange of information, loss of packets and such, asymptotically connected overlays may suffer disconnections and routers within an asymptotically dominant overlay may not be able to reach all routers in the network. If the asymptotic property is satisfied, however, these phenomena are temporary and to be corrected as topology information is updated.

Table 6.1 summarizes the requirements imposed by each link-state operation to the associated overlay subgraph and the minimization objectives that have to be addressed for each overlay. These requirements and design objectives are detailed in sections 6.2.1, for flooding; 6.2.2, for LSDB synchronization and 6.2.3, for topology selection.

	Graph / Overlay	Topology requirements	Minimization targets
Full Network	$G = (V, E)$	Connected	-
Flooding	$G_F = (V_F \subseteq V, E_F \subseteq E)$	Connected and dominating (CDS)	Number of retransm. Flooding latency
Link-State DB Synchronization	$G_S = (V, E_S \subseteq E)$	Connected and spanning	Number of synchr. processes
Advertised Links (topology selection)	$G_R = (V, E_R \subseteq E)$	Connected and spanning Includes sh.-paths of G	Number of links and updates

Table 6.1: Summary of overlay requirements.

6.2.1 Topology Update Flooding

Flooding of packets from a source, s, is performed through a source-dependent overlay composed of the directional links between routers transmitting the updates and routers forwarding

them. Source dependency implies that the overlay may change (although it is not required) depending on the router that transmits first. Figure 6.1 illustrates the flooding procedure and the flooding overlays for two packets sent from two different routers in a network: routers are part of the flooding overlay for a packet when they forward the packet after first reception – and they forward a packet when they have neighbors that have not yet received the packet.

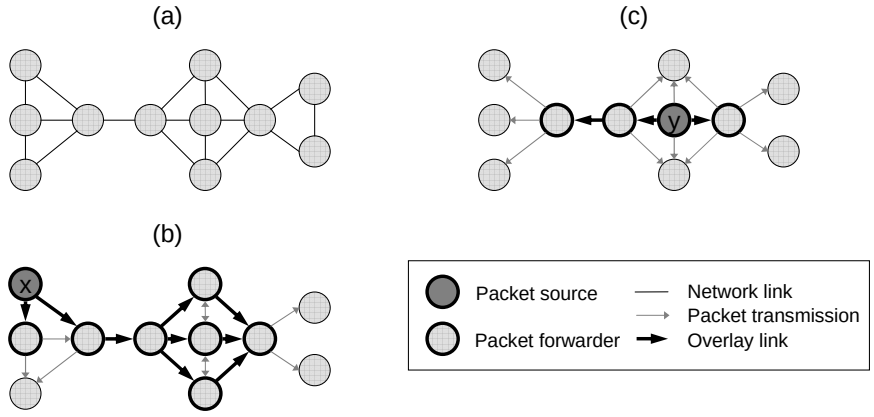


Figure 6.1: Flooding example: **(a)** Network graph, **(b)** Overlay flooding for a packet sent from router x, **(c)** Overlay flooding from a packet sent from router y.

Given a flooded packet, this overlay has to ensure that, for every router in the network, regardless of whether it participates in the packet flooding or not, gets (at least) one copy of the packet. This requires that flooding overlays are *connected* and *dominate* the network graph. The use of Connected Dominating Sets (CDS) for broadcast/multicast flooding in ad hoc networks has been widely studied in the literature (see [66] for reference). In order to avoid collisions and wireless channel saturation, caused by simultaneous packet retransmissions, the link density of the overlay should be reduced. As excessively sparse overlays may lead to increasing the time for a flooded packet to reach all routers, and flooding latency is also a minimization objective, the trade-off between overlay density and latency should be taken into account.

6.2.2 Point-to-point Synchronization

A synchronized overlay contains links between the routers, which have exchanged their LSDBs and which keep their local instances of LSDB synchronized. Due to the symmetric nature

of LSDB synchronization, the graph resulting from the union of synchronized links is not directed.

Formally, such an overlay needs to form a spanning connected subgraph within the network graph[1], in order to disseminate the LSDB over the whole network. Given that a LSDB synchronization process is performed once in the lifetime of a synchronized link, the number of synchronization processes performed in a network depends on (i) the synchronized overlay density, that is, the number of links included in the synchronized overlay; and (ii) the stability of links in the synchronized overlay, that is, the time that links stay within the synchronized overlay before disappearing or being excluded from the overlay. Minimizing the overhead associated with LSDB synchronization necessitates an overlay which has:

1. low overlay link density (*i.e.*, few number of overlay links per router), and
2. low overlay link change rates (*i.e.*, stable links).

6.2.3 Topology Selection

In link-state routing, topology selection has as objective, together with flooding, to provide routers with sufficient information about the network topology to independently compute shortest paths to all destinations. Global topology information enables routers to compute shortest paths over the network, while local topology information enables a router to compute local shortest paths within its neighborhood. Throughout this manuscript, the following terms are used to distinguish between these types of shortest paths:

Definition 6.5 (**Network-wide shortest path**). A path between two vertices $x, y \in V(G)$, p_{xy}, is a *network-wide shortest path* between x and y if there is no other path p'_{xy} between x and y such that $cost(p'_{xy}) < cost(p_{xy})$.

Definition 6.6 (**Local (k-hop) shortest path**). A path between two vertices $x, y \in V(G)$, p_{xy}, is a *local (k-hop) shortest path* between x and y if $|p_{xy}| \leq k$ and there is no other path p'_{xy} between x and y such that $|p'_{xy}| \leq k$ and $cost(p'_{xy}) < cost(p_{xy})$.

[1] *I.e.*, has to include every vertex (router) in the network.

The optimality notion of defs. 6.5 and 6.6 depends on the *cost* function for links and paths. This function can be defined in different ways depending on the characteristics of the network or the feature (or set of features) to be optimized in routing. For a given *cost* function, however, optimal (shortest, with respect to the *cost*) paths are preferable to sub-optimal (non-shortest) paths – otherwise the *cost* function may be redefined to identify the most preferable paths.

Link-state routing protocols typically advertise all links in the network to ensure that all routers have an identical and complete views of the network topology. In practice, the set of links that routers advertise to the network can be reduced as far as it does not prevent the receiving routers from selecting network-wide optimal routes. This permits reducing the amount of control traffic spent on disseminating the advertisements and updates of unnecessary links, *i.e.*, links that are not required in order to form network-wide shortest paths. The fact of receiving information about a non-complete subset of network links via flooding implies also that routers' views of the network topology are not completely consistent, as neighborhood information about the local topology is complete while flooding information about global topology is partial. Different network topology views are, however, acceptable if the shortest paths computed by different routers are consistent, *i.e.*, they do not cause permanent routing loops. Hence, selection of advertised links provides a trade-off between the size of the topology update messages and the accuracy of the topological view of the network in all routers.

A topology selection overlay must be connected and spanning in the network, in order to enable route computation towards any destination in the network. For the computation to be asymptotically optimal[2], the set of edges included in the overlay must contain network-wide shortest paths from the computing router to all destinations.

[2]In real conditions, the computation may be suboptimal due to stale topology information, transmission failures and such. *Asymptotic optimality* implies that in ideal conditions (message transmission delay $\longrightarrow 0$, collision probability $\longrightarrow 0$, channel failure probability $\longrightarrow 0$) the computation provides shortest (optimal) paths.

6.3 Full Network Overlay

The overhead of a link-state operation (flooding, topology selection and LSDB synchronization) depends on the size (number of involved routers and links) of the overlay in which such operation is performed: bigger overlays lead to more overhead, for performing the same operation. Each link-state operation incurs in a different amount of overhead. For comparison, this section describes the cost, in terms of needed traffic, of performing each operation in a single overlay – the overlay that includes all routers and all links in a network. Such an overlay is denominated **full network overlay**.

Analysis in this section assumes a Unit Disk Graph (UDG) network model with uniform router density, with the variables described in Table 6.2.

n	Number of routers in the network
ν	Network router density, assumed uniform
m	Average number of neigbhbors per router, $m = \pi\nu$
p	Probability that a packet transmission is successful. $0 < p \leq 1$ ($p = 1$ for error-free channels)

Table 6.2: Variables of the analysis.

Section 6.3.1 computes the overhead caused by flooding through the full network overlay, in messages and number of advertised links per second. Section 6.3.2 provides a lower bound for the message rate caused by LSDB synchronization processes between every pair of neighboring routers in the network. Based on these computations, section 6.3.3 evaluates the order of magnitude of control traffic of a link-state routing protocols that uses full network overlay for all the link-state operations.

6.3.1 Full Network Topology Flooding

Flooding of a single topology update message over the network over the full network overlay requires n transmissions of the message. Since all routers are included in the overlay, each router is allowed to retransmit the message exactly once.

Let t be the average link lifetime, then the average rate f (for frequency) of link changes

for a router with m neighbors is:

$$f = \frac{m}{t} \tag{6.1}$$

Assuming that every topology change in the neighborhood of a router causes flooding of a new topology update message, the control traffic (in number of messages per second) for disseminating topology updates of a single router (not including periodic flooding) is:

$$F_1 = fn = \frac{m}{t}n = \frac{nm}{t} \quad , \quad \left(\text{in } \frac{msg}{s}\right) \tag{6.2}$$

The control traffic caused by topology advertisements generated and flooded by every router in the network can be computed as:

$$F_n = nF_1 = n\frac{nm}{t} = \frac{n^2m}{t} \quad , \quad \left(\text{in } \frac{msg}{s}\right) \tag{6.3}$$

Expressions (6.2) and (6.5) assume an ideal, error-free channel ($p = 1$). For more realistic channel model ($p < 1$), the average number of transmissions that is needed to transmit successfully (without errors) a packet is:

$$\sum_{k=1}^{\infty} k(1-p)^{k-1}p = \frac{1}{p} \tag{6.4}$$

The packet transmission rate, caused by network-wide flooding can be expressed in function of p:

$$F_n^{msg}(p) = \frac{n^2m}{pt} \quad , \quad \left(\text{in } \frac{msg}{s}\right) \tag{6.5}$$

Using a full network overlay, topology update messages advertise all the links to all neighbors maintained by the router that creates the topology update (m in average). Therefore, the number of links advertised per second by router is:

$$F_n^{lnk}(p) = \frac{(nm)^2}{pt} \quad , \quad \left(\text{in } \frac{lnk}{s}\right) \tag{6.6}$$

6.3.2 Full Network Synchronization

This section evaluates the cost, in terms of packet transmissions, of performing LSDB synchronization over a full network overlay. Synchronization of a link between two routers (synchronization endpoints) includes exchange, and update, of their respective local instances of the LSDB in a master/slave manner. This exchange usually[3] consists of two phases, executed by each endpoint: (i) announcement of the topology advertisements that are part of the LSDB, and (ii) transmission of a subset of these, as reply to a request by the other endpoint.

The number of transmissions in phase (ii) depends on the differences between the local instances of LSDB maintained by each of the synchronizing routers. Phase (i) is deterministic, the number of transmissions is a function of the LSDB size and the announcement method only. Therefore, the number of packets per second transmitted by a router for completing phase (i) is $\lceil \frac{n}{k} \rceil$, where n is the number of routers and k is the number of topology advertisements announced in a single transmission. Assuming that a router synchronizes its LSDB with LSDBs from all its neighbors (full network overlay), that leads to the following transmission rate for a router:

$$S_1^{(i)} = \frac{m}{t} \lceil \frac{n}{k} \rceil \quad , \quad \left(\text{in } \frac{msg}{s}\right) \tag{6.7}$$

The packet transmission rate for phase (ii) in the whole network then being:

$$S_n^{(i)} = nS_1^{(i)} = n\frac{m}{t} \lceil \frac{n}{k} \rceil \tag{6.8}$$

For channels with a non-negligible packet error rate $(1 - p)$, (6.8) yields:

$$S_n^{(i)}(p) = \frac{nm}{pt} \lceil \frac{n}{k} \rceil \tag{6.9}$$

[3] *E.g.*, OSPF and IS-IS.

6.3.3 Overall Control Traffic

The control traffic incurred by topology distribution (not considering neighbor sensing) can be estimated as the sum of the topology update packets that are flooded over the network (6.5) and the packets exchanged during LSDB synchronization processes (6.9). The resulting packet transmission rate is as follows:

$$\begin{aligned} C_n(p) &= F_n^{msg}(p) + S_n(p) \leq F_n^{msg}(p) + S_n^{(i)}(p) = \frac{n^2 m}{pt} + \frac{nm}{pt}\lceil\frac{n}{k}\rceil = \\ &= O(n^2) \end{aligned} \tag{6.10}$$

When network density grows with the number of routers, n (for a fixed network grid A, $\nu = \frac{n}{A}$), (6.10) becomes:

$$\begin{aligned} C_n(p) &\leq \frac{n^2 m}{pt} + \frac{nm}{pt}\lceil\frac{n}{k}\rceil = \left[m = \pi\nu = \pi\frac{n}{A}\right] = \frac{n^3\pi/A}{pt} + \frac{n^2/A}{pt}\lceil\frac{n}{k}\rceil = \\ &= \frac{1}{Apt}\left(n^3\pi + n^2\lceil\frac{n}{k}\rceil\right) = O(n^3) \end{aligned} \tag{6.11}$$

Expressions (6.10) and (6.11) give lower bounds, as they do not include the traffic generated by phase (ii) of link synchronization. Even without considering errors or packet losses in wireless links ($p = 1$), (6.10) indicates clearly that the full network overlay does not scale for large ad hoc networks. This has been analytically [69] and empirically confirmed for OSPF [72, 113], showing that this protocol requires a high portion of the available bandwidth for link-state diffusion and update in ad hoc networks, being unable to perform successfully routing even in small networks – with more than 20 routers.

6.4 Conclusion

The use of link-state overlays facilitates the analysis of the properties and features that are needed for performing link-state routing in ad hoc networks. Each link-state overlay is associated with a specific link-state operation: topology selection, flooding and LSDB synchronization.

One of the main limitations in ad hoc networking is the available bandwidth. If all routers and all links participate in the three link-state overlays, the performance of their three associated link-state operations cause a control traffic overhead that does not scale. Minimization of its impact becomes a necessity, and a target in the design of link-state protocols for ad hoc networks.

From the separate analysis of the three link-state overlays, it can be concluded that different operations yield different, and not always compatible, optimization requirements. A natural way to accommodate these different, sometimes competing requirementss is to design independently the overlays corresponding to different link-state operations. The flooding overlay of a router needs to be connected and dominating, and optimization efforts should focus on reducing the number of involved links. For LSDB synchronization, the synchronized overlay has to include all routers in the network, the optimization should target both minimization of the number of links and selection of the most stable links, in order to minimize the number of database exchanges. Finally, topology selection overlays generated by the addition of links listed in link-state advertisements must provide to every router with enough topology information from the network so that it can compute optimal routes to all possible destinations – that is, it must contain network-wide shortest paths.

The following chapters in Part II propose different techniques for generating link-state overlays, compare them to each other and to the full network overlay, and discuss their use for the different link-state operations based on the characteristics required for each of them, according to this chapter.

Chapter 7

The Synchronized Link Overlay Triangular – SLOT

The Synchronized Link Overlay Triangular (SLOT) technique defines an overlay that can be constructed in a distributed fashion by routers in an ad hoc network. Routers only need information about their 1-hop neighbors for selecting and updating links in the overlay. This chapter motivates the use and interest of this overlay for link-state routing, relates it to other graphs, and explores the applicability of this overlay for the main link-state operations, mainly by evaluating analytically the properties of SLOT.

Two variations of SLOT are presented and analyzed in the chapter, each using a different link metric: a variation of hop-count metrics, denoted as SLOT-U; and a variation of distance-based link metrics, denoted as SLOT-D. The use of different metrics causes significant changes in some of the described properties of the overlay.

7.1 Outline

Section 7.2 describes the relationship between SLOT and other well-known overlays, and some properties of the SLOT overlay are deduced from this relationship. Sections 7.3 and 7.4 elab-

orate on the performance of SLOT and its variations. The focus in these sections is overlay density and overlay link change rate, identified in chapter 6 as essential parameters in synchronization and flooding overlays. Section 7.3 studies these two parameters (overlay density and overlay link change rate) analytically for SLOT variations in 2-dimensional mobile networks, and validates the results by way of simulations, while section 7.4 extends the analysis to 1-dimensional and 3-dimensional networks. Section 7.5 examines, probabilistically, the length of links selected by both variations of SLOT. Finally, section 7.6 concludes the chapter.

7.2 Definition, Related Overlays and Variations

The Synchronized Link Overlay Triangular (SLOT) is an overlay, defined over a network graph $G = (V, E)$. SLOT is a particular case of the more general Synchronized Link Overlay (SLO), and is also inspired by the Relative Neighborhood Graph (RNG) defined over a set of points in $\mathbb{R}^n$ [131]. This latter graph is, in turn, a subgraph of the Gabriel Graph (GG) [132]. These relations are illustrated in Figure 7.1, and are detailed throughout this section.

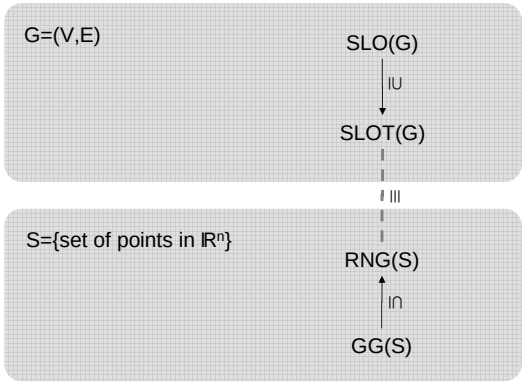


Figure 7.1: Relations between the Gabriel Graph, the Relative Neighbor Graph, the Synchronized Link Overlay and SLO Triangular.

Section 7.2.1 defines the Gabriel Graph and the Relative Neighbor Graph of a set of points $S \subseteq \mathbb{R}^n$, and proves that the latter is a subgraph of the former. Section 7.2.2 defines the Synchronized Link Overlay (SLO) of a network graph $G = (V, E)$ and describes the SLOT overlay as a particular

case of SLO. This section also illustrates the relationship between SLOT and RNG. Finally, section 7.2.3 defines formally the two variations of SLOT, SLOT-U and SLOT-D.

7.2.1 Gabriel Graphs and Relative Neighborhood Graphs

The Gabriel Graph was introduced by K. R. Gabriel, jointly with R. R. Sokal [132]. Given a set of points $S \subseteq \mathbb{R}^n$, the edge between two points u and v in S is included in this graph if the ball[1] centered in the midpoint between u and v contains no other points in S (see Figure 7.2.a). More formally, the Gabriel Graph (GG) of a set S is defined as follows:

$$\begin{aligned} &u, v \in S, c_{u,v} = \tfrac{u+v}{2} \in \mathbb{R}^n \\ &\overline{uv} \in GG(S) \iff \nexists w \in S : w \in B_{\frac{1}{2}d(u,v)}(c_{u,v}) \end{aligned} \tag{7.1}$$

The Relative Neighborhood Graph (RNG) [131] of a set of points S, $RNG(S)$, is the graph that results from considering edges between points u and v such that there is no other point that is *closer*[2] to u and v that they are to each other. Selected links connect pairs of routers $\{u, v\}$ for which the intersection of circles centered on u and v, with radius $d(u, v)$ (the distance from u to v), contains no other routers (see Figure 7.2.b, the intersection corresponds to the dotted region). More formally, the relative neighbor subgraph of S is defined as follows:

$$RNG(S) = \{\overline{uv}, u, v \in S : (\nexists w \in S : d(u, w), d(w, v) < d(u, v))\} \tag{7.2}$$

As *Lemma 7.1* proves, the Relative Neighbor Graph is a subgraph of the Gabriel Graph, since every link included in the former is automatically included in the latter.

Lemma 7.1. *Let $S = \{p : p \in \mathbb{R}^n\}$ a set of points in $\mathbb{R}^n$. Then,*

$$RNG(S) \subseteq GG(S) \tag{7.3}$$

[1] A *ball* in $\mathbb{R}^n$, with radius r and center c, is the set of $\mathbb{R}^n$-points at distance $\leq r$ of the point c, for the Euclidian notion of distance in $\mathbb{R}^n$.

[2] *Closer* in the sense of the Euclidean distance of $\mathbb{R}^n$.

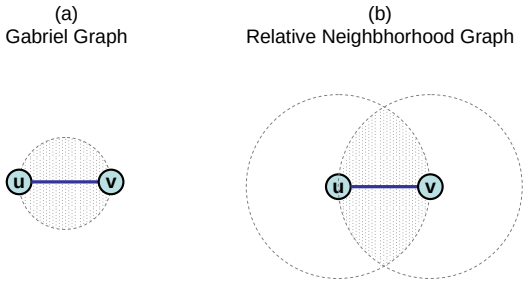


Figure 7.2: The link $\overline{uv}$ belongs to **(a)** the Gabriel Graph and **(b)** the Relative Neighbor Graph, if the corresponding dotted region does not contain any other vertex (node).

Proof: Let e be an edge of $RNG(S)$. Then, from the definition of Routable Neighbor Graph, e connects two vertices u and v such that there is no other vertex $w \in S$ for which $d(u,w) < d(u,v)$ and $d(w,v) < d(u,v)$. Let $c_{u,v} = \frac{u+v}{2}$ be the midpoint between u and v, and consider the ball $B_{\frac{1}{2}d(u,v)}(c_{u,v})$ which is contained in the region $Q = \{q \in \mathbb{R}^n : d(q,u) < d(u,v), d(q,v) < d(u,v)\} \subseteq \mathbb{R}^n$. Therefore,

$$(\nexists s \in S : s \in Q) \Longrightarrow (\nexists s \in S : s \in B_{\frac{1}{2}d(u,v)}(c_{u,v}))$$

and e belongs to $GG(S)$. □

Both graphs (GG and RNG) are instances of Delaunay's triangulation [137]. Also, both definitions are dimension-agnostic, so they can be used for any dimension D, in particular for the cases of linear networks ($D = 1$), planar networks ($D = 2$) and cubic networks ($D = 3$), S corresponding in all cases to the set of router positions. For a further analysis and discussion of the properties of Gabriel and Relative Neighborhood Graphs, see [118, 131] for RNG and [130, 132] for GG. The Relative Neighor Graph has been proposed and experimentally evaluated as a broadcasting principle for ad hoc networks [78] and, more in particular, energy-constrained wireless ad hoc networks [59, 81].

7.2.2 The Synchronized Link Overlay and SLOT

The Synchronized Link Overlay of a network graph $G = (V, E)$ is an overlay, composed of those links $\overline{xy} \in E$ $(x, y \in V)$ for which one (and only one) of the two following conditions is

satisfied:

(i) There are no common neighbors between x and y, that is, $N(x) \cap N(y) = \emptyset$.

(ii) For each chain $\{c_1, c_2, ..., c_n\}$ of common neighbors of x and y, the cost of the direct link between x and y, $m(\overline{xy})$, is smaller than the maximum cost of the links in the chain $m(\overline{c_i c_{i+1}})$, with $0 \leq i \leq n$, $x \equiv c_0$ and $y \equiv c_{n+1}$.

Links included in the synchronized overlay are also denoted *synchronized links*. Equivalently, the overlay discards a link between routers x and y when there is a set of common neighbors of x and y $\{c_i : c_i \in N(x) \cap N(y)\}_{1 \leq i \leq k}$ such that the cost of each link $\overline{xc_1}, \overline{c_1 c_2}, ..., \overline{c_k y}$ is lower (with respect to the metric) than the cost of the link between x and y. Formally:

$$\overline{xy} \notin SLO \iff \exists c_1, c_2, c_3, ..., c_n : \begin{cases} \forall i \leq n, c_i \in N(x) \cap N(y) \\ m(x,y) > max\{m(x,c_1), m(c_1,c_2), ..., m(c_n,y)\} \end{cases} \tag{7.4}$$

It can be observed that, with this definition, SLO links are included in RNG while RNG links are not necessarily included in SLO, as Figure 7.3 indicates. Assuming a link cost based on distance, the link between u and v is included in RNG (given that there is no other router in the dotted region), but it is not synchronized in SLO because there is a chain of common neighbors of u and v, $\{c_1, c_2, c_3, c_4\}$, such that links $\{\overline{uc_1}, \overline{c_1 c_2}, \overline{c_2 c_3}, \overline{c_3 c_4}, \overline{c_4 v}\}$ have a smaller cost than $\overline{uv}$.

This chapter studies a simplified version of SLO, the Synchronized Link Overlay Triangular (SLOT). SLOT restricts the chain of intermediate common neighbors $\{c_1, c_2, ..., c_n\}$ to a single neighbor. Therefore, a link between two routers u and v is synchronized if and only if there is no router w that is common neighbor of u and v and is closer or at the same distance to u and v than they are to each other. In case of link cost equality (*i.e.*, $m(\overline{uw}) = m(\overline{wv}) = m(\overline{uv})$, m being the metric function), the tie is broken by excluding from synchronization the link that connects those routers with lowest ids.

When the metric m satisfies the three axioms of an Euclidean metric:

(i) Non-negativity: $m(a,b) \geq 0, \forall a, b$;

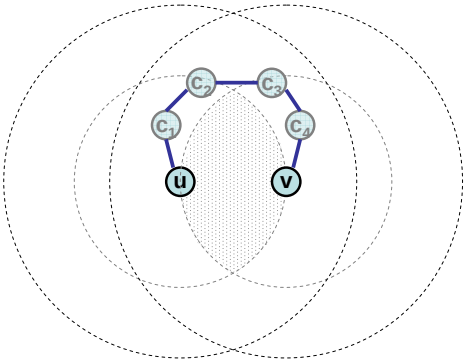


Figure 7.3: $\overline{uv}$ satisfies the condition for RNG, but it is not included in SLO, due to the existence of the chain $\{c_1, c_2, c_3, c_4\}$. *Assuming a metric based on distance for SLO, it is clear that* $m(\overline{c_i c_{i+1}}) \leq m(\overline{uv})$, $\forall 0 \leq i \leq 5$, *with* $u \equiv c_0$ *and* $v \equiv c_5$.

(ii) Symmetry: $m(a, b) = m(b, a), \forall a, b$; and

(iii) Triangle inequality: $m(a, b) \leq m(a, c) + m(c, b), \forall a, b, c$;

Then, the SLOT overlay over a network graph G is equivalent to the Relative Neighborhood Graph computed over the set of locations of the network routers. With an Euclidean metric m, SLOT therefore has the same properties as those which have been shown for RNG. For any set of points S, [131] shows that $RNG(S)$ contains the Minimum Spanning Tree (MST) of S. Hence, the SLOT overlay computed over a network graph G also contains the Minimum Spanning Tree of the set of router positions $S = V(G)$ and, in particular, is a connected and spanning subgraph of G.

7.2.3 SLOT-U and SLOT-D

This section examines the two variations of SLOT, SLOT-U and SLOT-D. The use of different link cost metrics impacts some of the properties of the corresponding overlays.

For SLOT-U, as all the link costs are equal to 1 (hop count), links are selected depending on the ids of the involved routers (tie breaking, see Figure 7.4). In SLOT-D, a link between routers is included in the overlay if there are no routers which are closer to any of the link endpoints that both endpoints to each other. Both the hop count and the distance-based link cost are Euclidean, and thus the corresponding overlays are connecting and spanning over the general network graph

G. Both variants are formally defined as follows:

$$\begin{cases} \text{SLOT-U}(G) & = & \{\overline{xy} \in E(G) : (\nexists z \in V(G), z \in N(x) \cap N(y) : id_z > max\{id_x, id_y\})\} \\ \text{SLOT-D}(G) & = & \{\overline{xy} \in E(G) : (\nexists z \in V(G), z \in N(x) \cap N(y) : m(x,y) \geq max\{m(x,z), m(z,y)\})\} \end{cases} \tag{7.5}$$

SLOT-U does not require any particular mechanism to monitor and measure the link cost: all available links are treated equally, with the same uniform metric. SLOT-D, in contrast, needs a mechanism for establishing the distance between two neighbor routers, something that can be achieved by location-based means (such as GPS).

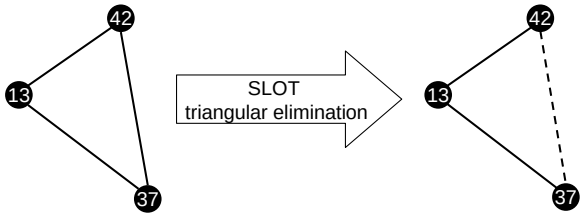


Figure 7.4: The SLOT triangular elimination under unit link cost. *The link connecting routers with the highest ids,* $\overline{42, 37}$ *in the picture, is excluded.*

7.3 Performance Analysis for 2-Dimensional Networks

This section provides a theoretical analysis of the performance of SLOT in 2-dimensional networks. The analysis focuses on two aspects: the overlay density (average number of overlay links per router) and the overlay link stability (rate of overlay link changes, creation or destruction) for SLOT-U and SLOT-D.

Theoretical results presented in this section, and in section 7.4, assume the unit disk graph network model (coverage radio $R = 1$). Routers are distributed uniformly over a square of area A, with constant density ν. Routers are assumed to move freely following independent random walks, with an average speed of s. It is also assumed that the random walk is isotropic, *i.e.*, the stationary probability that a node is in a portion of the map of area σ and has speed in a cone of aperture θ

is exactly $\frac{\theta\sigma}{2\pi A}$. This section also explores the asymptotic case where $\nu \to \infty$ and $A \to \infty$.

Under these assumptions, the average node neighborhood size is $M_f = \pi\nu$. The average link change rate for a router corresponds to $V_f = 2\Delta(s)\nu$ [43], $\Delta(s)$ being the average relative speed between two routers.

Section 7.3.1 examines the overlay link density of SLOT-U and SLOT-D, *i.e.*, the average number of synchronized links per router. Section 7.3.2 examines the overlay link change rate, that is, the rate of creation / destruction of synchronized links. Finally, section 7.3.3 validates these results by simulating both variations of SLOT in mobile and static scenarios.

7.3.1 Overlay Density

Theorems 7.2 and *7.3* show how the overlay density is reduced when using SLOT with unit cost and distance-based cost, respectively.

Theorem 7.2. *The average number of SLOT-U links per router satisfies*

$$M_u(\nu) = \int_{\frac{\pi}{3}}^{\frac{\pi}{2}} d\theta \frac{8\pi}{\nu(A(\theta))^2} sin(2\theta)(\nu A(\theta) + e^{-\nu A(\theta)} - 1) \tag{7.6}$$

where $A(\theta) = 2\theta - sin(2\theta)$, and $M_u(\nu)$ tends when network density $\nu \longrightarrow \infty$, to the following value:

$$M_u = \int_{\frac{\pi}{3}}^{\frac{\pi}{2}} d\theta \frac{8\pi sin(2\theta)}{2\theta - sin(2\theta)} + O\left(\frac{1}{\nu}\right) \approx 3.604 \tag{7.7}$$

Proof: Consider two routers A and B with ids x and y, respectively. Assume that the link (A, B) belongs to the overlay and $y > x$ (this will cover half the cases). Let r be the distance between A and B, and let $S(r)$ be the intersection of the disks of radius 1, respectively centered on A and B, *i.e.* $S(r)$ is the location of the common neighborhood of A and B. Note that the area of $S(r)$ is $A_1(r) = 4\int_{\frac{d}{2}}^{r} \sqrt{1^2 - x^2}dx$. Since (A, B) is an overlay link there is no node in $S(r)$ with id smaller than $\min\{x, y\} = x$.

Since only the router id comparison is considered, there is no loss of generality in assuming that the ids of the routers are scalar numbers, uniformly distributed on the interval $[0, 1]$.

The probability that (i) $y > x$ (half of the cases) and (ii) the link (A, B) belongs to the overlay is $(1 - x)e^{-\nu x|S(r)|}$, where $|S(r)| = A(\theta)$ with $r = 2\cos\theta$. Considering also the case $y < x$, the probability that

(A, B) belongs to the overlay is $2(1-x)e^{-\nu x|S(r)|}$, and the average number of links per router is

$$\begin{aligned} M_u(\nu) &= 2\int_0^1 dx \int_0^1 2\pi\nu(1-x)rdre^{-xA(\theta)\nu} = \int_0^1 \int_{\frac{\pi}{3}}^{\frac{\pi}{2}} 8\pi\nu(1-x)\sin 2\theta d\theta e^{-xA(\theta)\nu} = \\ &= \int_{\frac{\pi}{3}}^{\frac{\pi}{2}} \frac{8\pi}{\nu(A(\theta))^2}\sin 2\theta d\theta(\nu A(\theta) - 1 + e^{-\nu A(\theta)}) \end{aligned}$$

Therefore:

$$M_u(\nu) = \int_{\frac{\pi}{3}}^{\frac{\pi}{2}} \frac{8\pi}{A(\theta)}\sin 2\theta d\theta + O(\frac{1}{\nu}) = M_u + O(\frac{1}{\nu})$$

with

$$M_u = \int_{\frac{\pi}{3}}^{\frac{\pi}{2}} \frac{8\pi\sin 2\theta}{2\theta - \sin 2\theta}d\theta \approx 3.603973720$$

□

Theorem 7.3. *The average number of SLOT-D links per router satisfies*

$$M_d(\nu) = \int_0^1 dr 2\pi\nu r e^{-r^2 A(\frac{\pi}{3})} \tag{7.8}$$

and tends when network density $\nu \longrightarrow \infty$, to

$$M_d = \frac{\pi}{2\frac{\pi}{3} - \frac{\sqrt{3}}{2}} + O(\nu e^{-\nu(\frac{2\pi}{3} - \frac{\sqrt{3}}{2})}) \approx 2.558 \tag{7.9}$$

where $A(\theta) = 2\theta - sin(2\theta)$.

Proof: Consider a link between two nodes A and B at distance r of each other. The condition under which the link belongs to the overlay is that, the intersection of the disks centered in nodes A and B with radius r, contains no nodes other than A and B. The area of this intersection is $A_r(r) = r^2 A(\frac{\pi}{3})$ with $A(\theta) = 2\theta - \sin(2\theta)$. Therefore, the probability that link (A, B) is included in the overlay is $e^{-\nu r^2 A(\frac{\pi}{3})}$.

The average number of links from a random node A, M_d, that belong to the overlay will be

$$\begin{aligned} M_d &= \int_0^1 2\pi\nu r dr e^{-r^2 A(\frac{\pi}{3})\nu} = \int_0^\infty 2\pi\nu r dr e^{-r^2 A(\frac{\pi}{3})\nu} + O(\nu e^{-\nu|B(1)|}) = \\ &= \frac{\pi}{2\frac{\pi}{3} - \frac{\sqrt{3}}{2}} + O(\nu e^{-\nu|B(1)|}) \approx 2.557530242 + O(\nu e^{-\nu|B(1)|}) \end{aligned} \tag{7.10}$$

The constant M_d, Devroye's constant, is known from [125]. □

Figure 7.5 indicates the evolution of SLOT-U and SLOT-D overlay densities as functions of the network density ν. The density reduction, while being relevant for both SLOT variations, is more significant for the distance-based cost: routers have more information about the network topology and can thus perform a more optimized selection of synchronized links.

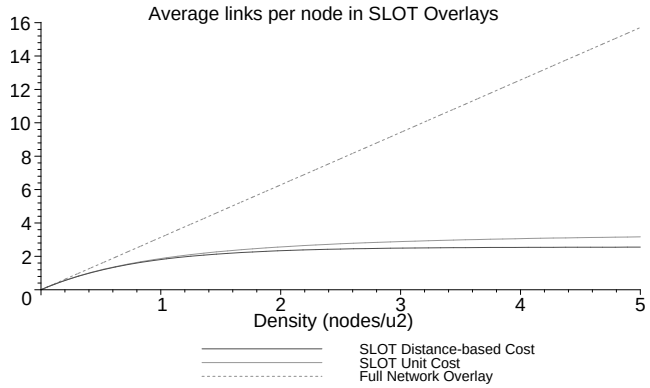


Figure 7.5: Average SLOT overlay density (links per router).

Theorems 7.2 and *7.3* show that the density of SLOT-U and SLOT-D, V_u and V_d respectively, has a finite upper bound independent from network density. That implies that SLOT-U and SLOT-D mechanisms are able to extract a sparse network overlay that connects all routers, with an overlay link density lower than a fixed constant, from a network with arbitrary link density.

7.3.2 Link Stability

Theorems 7.4 and *7.5* show that links that belong to SLOT-U and SLOT-D overlays are significantly more stable (have a longer lifetime) than average links in the full overlay of a mobile network, meaning that SLOT links disappear due to the relative mobility of their endpoints at a lower rate than average links in the network. Figure 7.6 illustrates such stability, measured as the rate of inclusion/destruction of links in the overlay, for a moderate mobility scenario (constant router speed $s = 5m/s$).

Theorem 7.4. *The average rate of link inclusion (and destruction) in SLOT-U is:*

$$V_u(s,\nu) = \Delta(s) \int_{\frac{\pi}{3}}^{\frac{\pi}{2}} d\theta \frac{32\theta sin(2\theta)}{\nu(A(\theta)^3)} (A(\theta)\nu - 2 + e^{-\nu A(\theta)}(2 + \nu A(\theta))) \tag{7.11}$$

where $A(\theta) = 2\theta - sin(2\theta)$ *and* $\Delta(s)$ *is the average relative speed between routers. For constant speed* ($\Delta(s) = \frac{4}{\pi}s$)*, equation (7.11) becomes*

$$V_u(s,\nu) = \frac{128s}{\pi} \int_{\frac{\pi}{3}}^{\frac{\pi}{2}} d\theta \frac{\theta sin(2\theta)}{(2\theta - sin(2\theta))^2} \approx 4.146s + O\left(\frac{4s}{\pi\nu}\right) \tag{7.12}$$

Proof: To get the rate at which overlay links vanish in the uniform cost algorithm (SLOT-U), consider the link (A, B) such that routers are at distance r and their ids are respectively x and y. Assume $x < y$. The area $S(r)$ contains no node with id smaller than x. The rate at which the link (A, B) will disappear as overlay link is equal to the rate at which nodes with id smaller than x will enter the area $S(r)$. This rate is equal to $|\partial S(r)|\frac{\Delta(s)}{\pi}\nu x$. Since $|\partial S(r)| = 4\theta$ with $r = 2\cos\theta$, the rate at which overlay links disappear (including the case $y < x$) is:

$$\begin{aligned} V_u(\nu) &= \Delta(s)\int_0^1 (1-x)x dx \times \int_{\frac{\pi}{3}}^{\frac{\pi}{2}} 32\nu^2\theta\sin 2\theta d\theta e^{-\nu A(\theta)x} = \\ &= \Delta(s)\int_{\frac{\pi}{3}}^{\frac{\pi}{2}} \frac{32\theta\sin 2\theta}{\nu(A(\theta))^3} \times \left(A(\theta)\nu - 2 + (2+\nu A(\theta))e^{-\nu A(\theta)}\right) d\theta = \\ &= V_u + O\left(\frac{\Delta(s)}{\nu}\right) \end{aligned} \tag{7.13}$$

with

$$V_u = 32\Delta(s)\int_{\frac{\pi}{3}}^{\frac{\pi}{2}} \frac{\theta\sin 2\theta}{(2\theta - \sin 2\theta)^2} d\theta$$

When the speed is a constant s, the expression for V_u becomes $V_u = \frac{128s}{\pi}\int_{\frac{\pi}{3}}^{\frac{\pi}{2}} \frac{\theta\sin 2\theta}{(2\theta-\sin 2\theta)^2} d\theta \approx 4.146111863 \times s$.

□

Theorem 7.5. *The average rate of link inclusion (and destruction) in SLOT-D is:*

$$V_d(s,\nu) = \frac{4}{3}\Delta(s)\int_0^1 2\pi\nu^2 r^2 e^{-r^2\nu A(\frac{\pi}{3})} \tag{7.14}$$

where $A(\theta) = 2\theta - sin(2\theta)$ and $\Delta(s)$ is the average relative speed between routers. For constant speed ($\Delta(s) = \frac{4}{\pi}s$), equation (7.14) becomes

$$V_d(s,\nu) \approx 3.471 s\sqrt{\nu} \tag{7.15}$$

Proof: Consider a link (A, B) which belongs to the overlay. $B(r)$, the intersection of the corresponding disks of radius r, is empty. Therefore, the rate at which the link will disappear from the overlay is equal to the rate at which mobile nodes enter $B(r)$. Let $\partial B(r)$ be the border of $B(r)$; its length is then $|\partial B(r)| = \frac{4}{3}\pi r$. The entering rate is therefore $|\partial B(r)|\frac{\Delta(s)}{\pi}\nu$. Therefore, the rate V_d at which overlay links vanish, from a random node A, is

$$\begin{aligned} V_d &= \tfrac{4}{3}\Delta(s)\int_0^1 2\pi\nu^2 r^2 dr e^{-r^2 A(\frac{\pi}{3})\nu} = \tfrac{4}{3}\Delta(s)\int_0^\infty 2\pi\nu^2 r^2 dr e^{-r^2 A(\frac{\pi}{3})\nu} + O(\nu^2 e^{-\nu|B(1)|}) = \\ &= \tfrac{4}{3\sqrt{\pi}}(A(\tfrac{\pi}{3}))^{-\frac{3}{2}}\Delta(s)\sqrt{\nu} + O(\nu^2 e^{-\nu|B(1)|}) \end{aligned} \tag{7.16}$$

Notice that $V_d \approx 3.471762654 \times s\sqrt{\nu}$ when the speed is constant. The rate at which links appear is also V_d.

□

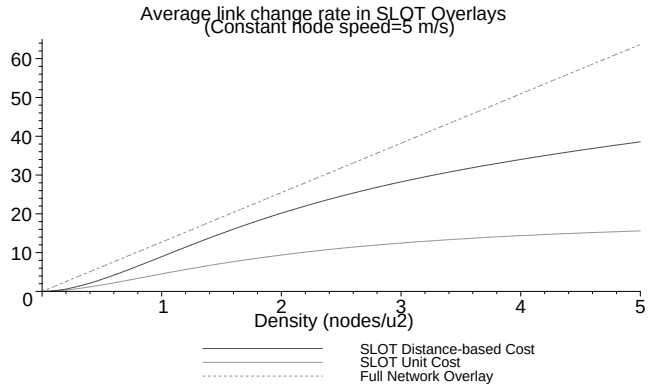


Figure 7.6: Average SLOT links change, for constant speed $s = 5m/s$.

Figure 7.6 indicates that SLOT-D has a higher link change rate than SLOT-U, meaning that links in SLOT-D appear and disappear at a higher rate than in SLOT-U. This implies that SLOT-U link are more stable (have a higher lifetime, in average) than SLOT-D links. This is due to the sensitivity of SLOT-D to changes in routers relative position (and thus link cost). Changes in the cost of links may cause additional SLOT-D link inclusion/exclusion decisions. In contrast, SLOT-U ensures that there will be no changes in the synchronization decisions as long as there are no new routers forcing new triangular eliminations (see Figure 7.2).

7.3.3 Validation

This section presents the most significant results, link density and link change rate, from simulation of SLOT-U and SLOT-D overlays in mobile and static scenarios. Mobile scenarios assume a network model, based on the Unit Disk Graph (UDG), and also assume that routers move within a $6r \times 6r$ grid, with r being the radius of the coverage area of each router (see Figure 7.7.a). Static scenarios also use the UDG network model, but assume a fixed grid of $600m \times 600m$ with coverage radius for routers $r = 150m$ (see Figure 7.7.b). Simulation parameters, mobility model and further details about the performed experiments are described in the Appendix D.

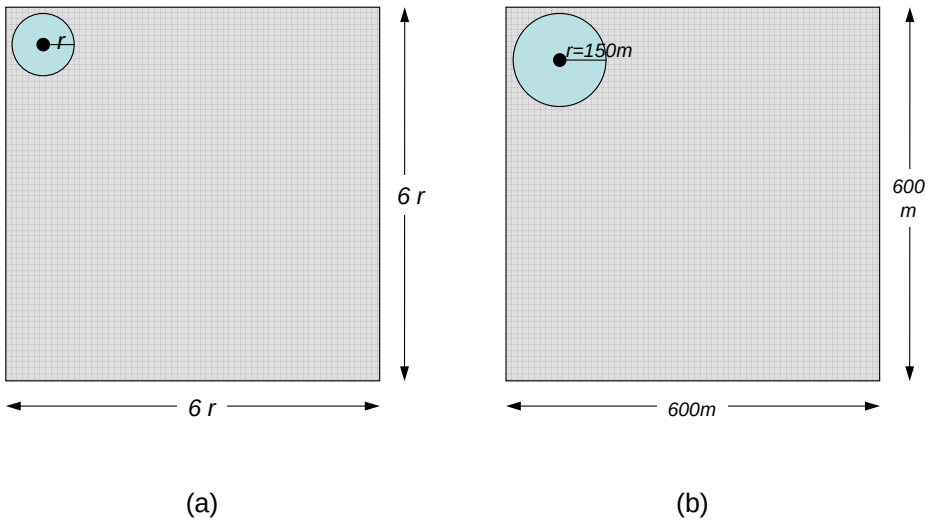


Figure 7.7: **(a)** Fixed $6r \times 6r$ grid for mobile scenarios, and **(b)** Fixed grid ($600m \times 600m$, $r = 150m$) for static scenarios.

Overlay Link Density

Figures 7.8 and 7.9 show, for mobile and static scenarios respectively, the overlay link densities (average number of synchronized links per router) provided by SLOT-U and SLOT-D, and compare these to the average number of links per router in the network. Both figures confirm that SLOT-D overlays are sparser than SLOT-U overlays. This is due to the fact that SLOT-D link synchronization is able to take information about the link length into account, as mentioned in section 7.3.1.

Discrepancies between these simulations and theory can be noticed in Figure 7.8, mainly due to the fact that the simulations run on a finite size grid, while the theoretical analysis was based on the assumption of an infinite grid.

Indeed, even in a rather "big" $6r \times 6r$ grid, more than 55% of the nodes are neighbors of the border, impacting neighbor size and triangle adjacencies occurrence. The theoretical results thus bring a theoretical upper bound for the finite size networks that were simulated (Figure 7.8.a). There are fewer border effects with SLOT-D than with SLOT-U, because SLOT-D priorizes links between close nodes to the detriment of those between distant neighbors, as mentioned in section

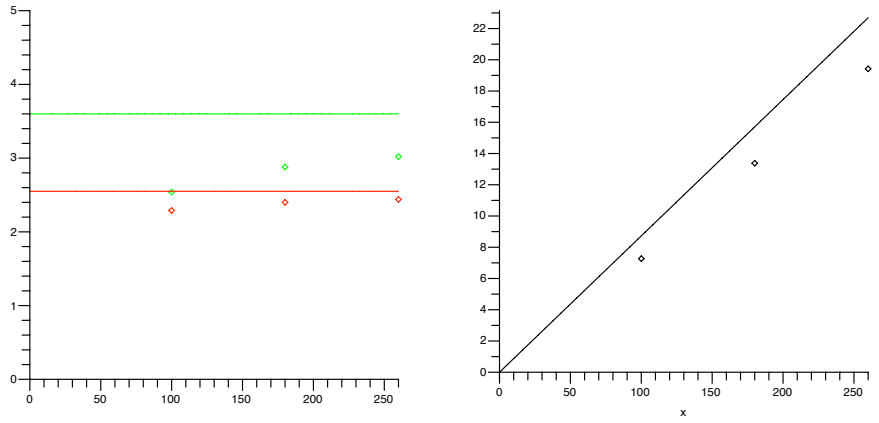


Figure 7.8: Average link density on a 6×6 map, **(a)** SLOT overlay: Maple simulations (dots), theory (plain), SLOT with distance cost (red), uniform cost (green), **(b)** Full network: Maple simulations (dots), theory (plain).

7.5. Regarding the latter, the simulations show results well below the theoretical performance. Simulations with $2r \times 2r$ and $4r \times 4r$ grids were also performed, and as the simulated map becomes bigger, simulation results converge towards the theoretical upper bound. Such border effects can also be observed on the density of links in full network (Figure 7.8.b), where a gap is visible between theory and the simulations.

Overlay Link Change Rate

Figure 7.10 shows the overlay link creation/destruction rate for SLOT-U and SLOT-D, compared to the rate of link creation/destruction in the full network. The gap between theory on infinite map and simulation on finite maps is no longer significant, and the simulations confirm that SLOT-D is outperformed by SLOT-U. This is due to the fact that SLOT-U has a link change rate independent of density while SLOT-D has a link change rate that depends on the square root of density, $\sqrt{\nu}$, as *Theorem 7.5* pointed out.

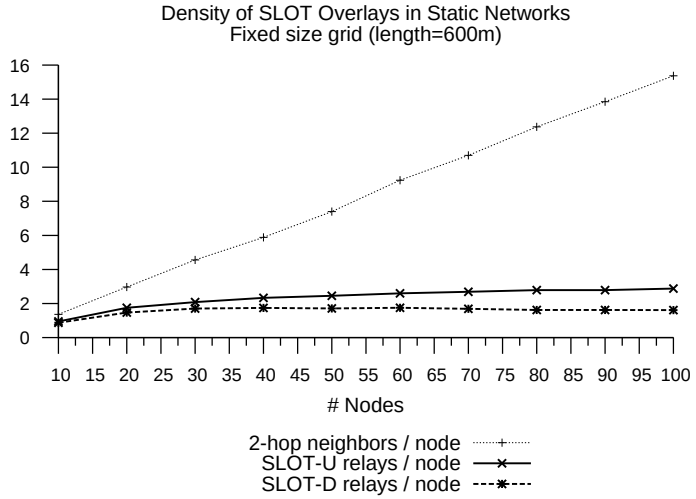


Figure 7.9: Density of SLOT overlays (SLOT-U and SLOT-D) in static networks. *The SLOT-D simulations are computed with the quantized cost: cost*$(\overline{xy}) = \lceil K\frac{d(\overline{xy})}{r} \rceil$*, with* $K = 10$ *and* $d(\overline{xy})$ *being the Euclidean distance between* x *and* y*.*

7.4 Performance Analysis for Other Dimensions

This section extends the analysis from section 7.3 to other dimensions D: a linear network ($D = 1$) and a cubic network ($D = 3$). Results in this section are based on the same network (unit-disk graph) and mobility model described in section 7.3, for dimensions 1 and 3.

7.4.1 1-Dimensional Networks

The one-dimensional case models networks, in which routers are deployed along a single line and where links between routers are determined by the position of these routers along the line. In this case, the positions of the nodes are ordered on the real ($\mathbb{R}$) axis as an increasing sequence $\{x_i\}_{i\in\mathbb{Z}}$, in which closer indexes indicate closer routers, *i.e.*:

$$j - i < k - i \Longrightarrow d(x_i, x_j) \leq d(x_i, x_k)$$

When the router speed s is constant, routers' relative speed in 1-dimensional networks is $\Delta(s) = s$.

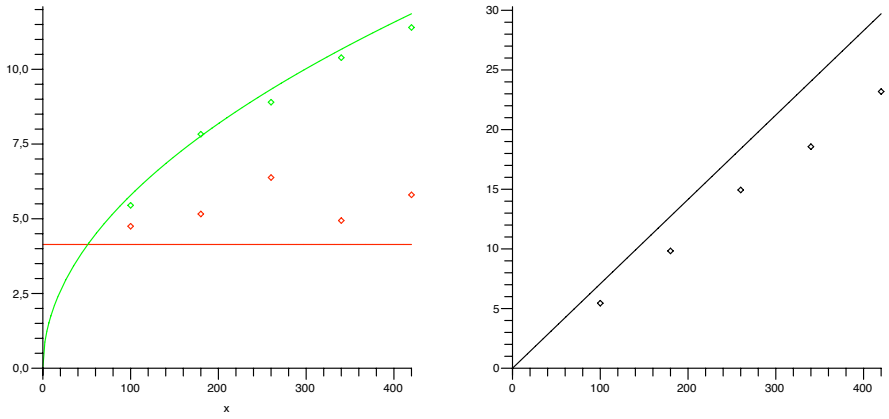


Figure 7.10: Average link creation rate (per node) with speed $1\frac{\text{unit distance}}{\text{time unit}}$, 6×6 map, **(a)** SLOT overlay: Maple simulations (dots), theory (plain), SLOT with distance cost (green), uniform cost (red), **(b)** Full network: Maple simulations (dots), theory (plain).

Theorem 7.6 describes the synchronized links present in the SLOT-D overlay, for 1-dimensional networks:

Theorem 7.6. *The SLOT-D overlay is made of the links* (x_i, x_{i+1}) *provided that* $x_{i+1} - x_i < 1$.

Consequently *Theorem 7.7* about SLOT-D performance in 1-dimensional networks is proved:

Theorem 7.7. *The average number of SLOT-D links per node is*

$$M_d = 2 + O(e^{-\nu})$$

and the average overlay link change rate is

$$V_d = 2\Delta(s)\nu + O(\nu\Delta(s)e^{-\nu})$$

Proof. These results can be deduced directly from *Theorem 7.6*, since:

1. There is one synchronized link between router x_i and x_{i-1}, and one synchronized link between x_i and x_{i+1}, except when those are at distance greater than 1. This happens with probability of order $e^{-\nu}$.
2. A change in existing synchronized links of x_i happens only when router x_{i-1} or x_{i+1} pass another router. A router passes a neighboring router with rate $\Delta(s)\nu$ and therefore the link rate change of node at position x_i is $2\Delta(s)\nu$.

However, these results can also be derived from the previous methodology inspired from the $D = 2$ analysis. In a 1-dimensional space, the set $B(r)$ is an interval of length $|B(r)| = r$, therefore

$$M_d = \int_0^1 e^{-\nu|B(r)|}\nu 2dr = 2 - 2e^{-\nu}$$

In dimension 1, the rate of entrance in a set B is $|\partial B|\frac{s}{2}\nu$ when the flow is isotropic with average speed s. $|\partial B(r)| = 2$ points, given that $B(r)$ is an interval of length r. Therefore the rate at which overlay links disappear, is:

$$V_d = \int_0^1 \int_0^1 2\nu dr e^{-\nu|B(r)|}|\partial B(r)|\nu\frac{\Delta(s)}{2} = 2\Delta(s)\nu - 2\Delta(s)\nu e^{-\nu}$$

□

Theorem 7.8 examines overlay link density and overlay link change rate of SLOT-U for linear networks:

Theorem 7.8. *For linear networks, the average number of SLOT-U overlay link per node is*

$$M_u = 4\log 2 + O\left(\frac{1}{\nu}\right)$$

The average SLOT-U overlay link change rate is

$$V_u = 2\Delta(s) + O\left(\frac{\Delta(s)}{\nu}\right)$$

Proof. The intersection of the neighborhood of two routers at distance r, $S(r)$, has a size (length) corresponding to $|S(r)| = 2 - r$. Then,

$$M_u = 2\int_0^1 (1-x)dx\left(\int_0^1 e^{-\nu|S(r)|x}2\nu dr\right) = 4\log 2 - \tfrac{2}{\nu} + O(e^{-\nu})$$

Regarding overlay link changes:

$$\begin{aligned} V_u &= 2\int_0^1 (1-x)xdx\left(\int_0^1 e^{-\nu|S(r)|x}2\nu^2|\partial S(r)|\tfrac{\Delta(s)}{2}dr\right) = [S(r) = 2-r, |\partial S(r)| = 2] = \\ &= 4\nu^2\int_0^1 dxx(1-x)\int_0^1 dre^{-\nu(2-r)x}\Delta(s) = \Delta(s)\left(\tfrac{4}{\nu e^{\nu}} - \tfrac{1}{\nu e^{2\nu}} + 2 - \tfrac{3}{\nu}\right) = \\ &= 2\Delta(s) - \tfrac{3\Delta(s)}{\nu} + O(\Delta(s)e^{-\nu}) = 2\Delta(s) + O\left(\tfrac{\Delta(s)}{\nu}\right) \end{aligned}$$

□

7.4.2 3-Dimensional Networks

For analysis of SLOT properties in 3-dimensional mobile ad hoc networks, the following geometric results need to be taken into account:

1) Let $S(r)$ denote the intersection of unit spheres whose centers are at distance r apart ($r \leq 1$). Then, it can be proven that:

$$|S(r)| = 2\int_0^{1-r/2} \pi(1-(\tfrac{r}{2}+x)^2)dx = 2\pi((1-\tfrac{r}{2}) - \tfrac{1}{3}(1-\tfrac{r}{2})^3)$$

and

$$|\partial S(r)| = 4\pi\int_0^{1-r/2} dx = 4\pi(1-\tfrac{r}{2})$$

2) The area and border length of $B(r)$ are $|B(r)| = r^3|S(1)|$ and $|\partial B(r)| = r^2|\partial S(1)|$, respectively.

3) The average entrance rate in a volume B is equal to $|\partial B|\frac{s}{4}\nu$ with mobile routers moving at average isotropic speed s.

4) If routers move at constant isotropic speed s, then the relative speed between routers is $\Delta(s) = \frac{4}{3}s$.

Theorem 7.9 describes the overlay link density and link change rate for SLOT-D in cubic networks:

Theorem 7.9. *The overlay link per-node density for SLOT-D in dimension 3 is:*

$$M_d = \frac{4\pi}{3}\left(\frac{12}{11\pi}\right)^{\frac{1}{3}} + O(e^{-11\pi\nu/12})$$

and the per-node overlay link change rate

$$V_d = 2\pi^2\left(\frac{12}{11\pi}\right)^{\frac{5}{3}}\frac{1}{3}\Gamma\left(\frac{5}{3}\right)\Delta(s)\nu^{\frac{1}{3}} + O(\nu e^{-11\pi\nu/12})$$

Proof. The straightforward methodology developed so far is applied with $D = 3$. Therefore,

$$\begin{aligned} M_d &= \int_0^1 e{-}\nu|B(r)|4\pi r^2\nu dr = \int_0^1 e{-}\nu r^3|S(1)|4\pi r^2\nu dr = \\ &= \tfrac{4\pi}{3|S(1)|^{1/3}}(1-e^{-\nu|S(1)|}) \end{aligned}$$

And the overlay link change rate is

$$\begin{aligned} V_d &= \int_0^1 e^{-\nu|B(r)|} 4\pi r^2 \nu dr |\partial B(r)| \tfrac{\Delta(s)}{4} \nu = \\ &= \nu^{\frac{1}{3}} \int_0^{\nu^{1/3}} e^{-|V(1)|x^3} \pi x^4 |\partial S(1)| \Delta(s) dx = \\ &= \nu^{\frac{1}{3}} \left(O(e^{-\nu V(1)}) + \int_0^{\infty} e^{-|V(1)|x^3} \pi x^4 |\partial S(1)| \Delta(s) \right) dx = \\ &= \pi |\partial S(1)| |S(1)|^{-\frac{5}{3}} \tfrac{1}{3} \Gamma(\tfrac{5}{3}) \Delta(s) \nu^{\frac{1}{3}} + O(\nu^{\frac{1}{3}} e^{-\nu|S(1)|}) \end{aligned}$$

□

Theorem 7.10 describes SLOT-U overlay link density and link change rate for cubic networks:

Theorem 7.10. *The per-node link density with the SLOT-U overlay is:*

$$M_u = 64\log(11) - 160\log(2) + 64\sqrt{3}(\text{Arctanh}(\tfrac{\sqrt{3}}{6} - \text{Arctanh}(\tfrac{\sqrt{3}}{3})) + O\left(\tfrac{1}{\nu}\right)$$

and the per-node overlay link change is:

$$V_u = \left(-8\log(2) - \tfrac{24}{11} + 8\log(11) + 16\sqrt{3}(\text{Arctanh}(\tfrac{\sqrt{3}}{6} - \text{Arctanh}(\tfrac{\sqrt{3}}{3})) + O(\tfrac{1}{\nu})\right) \Delta(s)$$

Proof. The node link density is computed as follows:

$$\begin{aligned} M_u &= 2\int_0^1 (1-x)dx \int_0^1 e^{-\nu|S(r)|x} 4\pi r^2 \nu dr = \\ &= 8\pi \int_0^1 \tfrac{\nu|S(r)|-1}{\nu|S(r)|^2} r^2 dr + O(e^{-\nu|S(0)}) = \\ &= 8\pi \int_0^1 \tfrac{1}{|S(r)|} r^2 dr + O(\tfrac{1}{\nu}) \end{aligned}$$

and the link change rate, correspondingly,

$$\begin{aligned} V_u &= \int (1-x)x dx \int_0^1 e^{-\nu|S(r)|x} 2\pi r^2 \nu dr |\partial S(r)| \Delta(s) \nu = \\ &= \int_0^1 2\pi r^2 |\partial S(r)| \tfrac{\nu|S(r)|-2}{\nu|S(r)|^3} dr + O(\Delta(s) e^{-\nu|S(0)|}) = \\ &= \int_0^1 2\pi |\partial S(r)| r^2 \tfrac{1}{|S(r)|^2} dr \Delta(s) + O(\tfrac{\Delta(s)}{\nu}) \end{aligned}$$

□

7.5 Selection of Links depending on Distance

The probability that a network link is included in SLOT depends, among other parameters, on the distance between its two endpoints. The impact of link length in the probability of link

inclusioon in the overlay is different for SLOT-U and SLOT-D. Intuitively, the longer a link is, the less likely it is that there is a common neighbor of both endpoints with a larger router id than both endpoints, which would cause exclusion of that link from SLOT-U. On the contrary, the further two neighbor routers are, the more probable it is for a common neighbor to be closer to both endpoints – thus, the more likely it is that SLOT-D discards such link. Longer distance between neighboring routers, therefore, increases the probability of link selection in SLOT-U and decreases the probability with SLOT-D.

This intuition is formalized in *Proposition 7.11.* Let $\sim$ denote, within this proposition, the relationship between routers connected by way of a synchronized link: $a \sim b$ thus implies that there is a SLOT link between a and b.

Proposition 7.11. *Assume that routers in the network are distributed according to a Poisson punctual process of rate (node density) ν. Then, the probability that a link between two routers x and y at distance d is included in the overlay is as indicated in expressions (7.18) for SLOT-U, (7.19) for SLOT-D with ideal distance-based link cost ($cost(\overline{xy}) = d$) and (7.19) for SLOT-D with discrete distance-based cost ($cost(\overline{xy}) = \lceil K\frac{d}{r} \rceil$, K being the number of discretization steps).*

$$P(x \sim y | m(\overline{xy}) = d, SLOT\text{-}U) = \frac{3}{2} e^{-\nu A_r(d)} \left(\frac{3}{2} e^{\frac{2}{3}\nu A_r(d)} - \frac{3}{2} - \nu A_r(d) \right) \quad (7.17)$$

$$P(x \sim y | m(\overline{xy}) = d, SLOT\text{-}D) = 1 - e^{-\nu d^2 (2\frac{\pi}{3} - sin(2\frac{\pi}{3}))} \quad (7.18)$$

$$P(x \sim y | m(\overline{xy}) = d, SLOT\text{-}D) = 1 - e^{-\nu \lceil \frac{K}{r} d \rceil^2 (2\frac{\pi}{3} - sin(2\frac{\pi}{3}))} \quad (7.19)$$

where $A_r(d) = 4\int_{\frac{d}{2}}^{r} \sqrt{r^2 - x^2} dx$ is the area of intersection between two circles of radius r at a distance d.

Proof: Under the conditions of the proposition, the probability that a link $x \longleftrightarrow y$ is synchronized under the SLOT-U rule is:

$$P(x \sim y | \text{SLOT-U}) = \left(\frac{2}{3} \right)^{n_{x,y}} \quad (7.20)$$

where $n_{x,y}$ is the number of common neighbors of x and y.

Consequently, the probability that a link between two routers x and y at distance $d < r$ is selected as part of the synchronized overlay is:

$$
\begin{aligned}
P(x \sim y | m(\overline{xy}) = d, \text{SLOT-U}) &= \sum_{k=0}^{\infty} P(n_{x,y} = k) P(x \sim y | m(\overline{xy}) = d, n_{x,y} = k) = \\
&= \sum_{k=0}^{\infty} \left(\frac{2}{3}\right)^k e^{-\nu A_r(d)} \frac{(\nu A_r(d))^{k+2}}{(k+2)!} = \\
&= e^{-\nu A_r(d)} \sum_{k=0}^{\infty} \left(\frac{2}{3}\right)^k \frac{(\nu A_r(d))^{k+2}}{(k+2)!} = \\
&= \frac{3}{2} e^{-\nu A_r(d)} \left(\frac{3}{2} e^{\frac{2}{3}\nu A_r(d)} - \frac{3}{2} - \nu A_r(d)\right) \qquad (7.21)
\end{aligned}
$$

The same argument applies for SLOT-D. If the link cost corresponds exactly to its length, the SLOT-D condition presented in (7.5) leads to:

$$
P(x \sim y | m(\overline{xy}) = d, \text{SLOT-D}) = 1 - e^{-\nu d^2 (2\frac{\pi}{3} - sin(2\frac{\pi}{3}))} \qquad (7.22)
$$

For the case of discrete cost ($cost = \lceil K \frac{d}{r} \rceil$), (7.22) becomes

$$
P(x \sim y | m(\overline{xy}) = d, \text{SLOT-D}) = 1 - e^{-\nu \lceil \frac{K}{r} d \rceil^2 (2\frac{\pi}{3} - sin(2\frac{\pi}{3}))} \qquad (7.23)
$$

□

Figure 7.11 indicates the probability that a link is selected by SLOT (SLOT-U and SLOT-D variations), as a function of its length. It can be observed that discreetization of the length for link cost reduces the probability of selecting a router for synchronization. This is consistent with the effect observed in Figures 7.6 and 7.9, in which the SLOT-D overlay density predicted by the theoretical analysis (with link cost equal to the length) was significantly higher than the average number of links per router obtained in the static simulations of SLOT-D (performed with a discrete link cost with $K = 10$ levels).

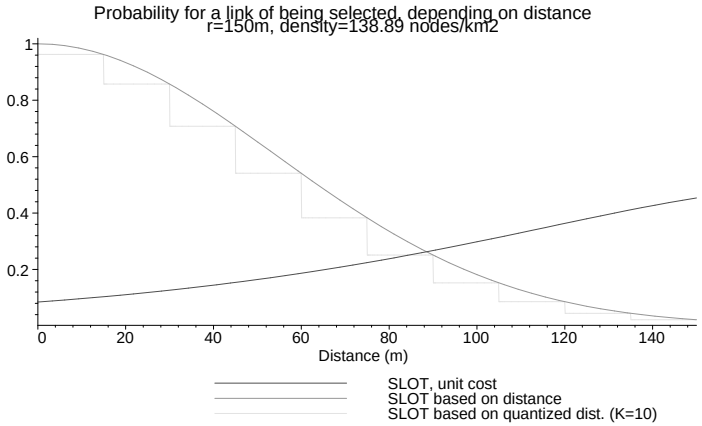


Figure 7.11: Probability for a link of being selected, under SLOT-U, SLOT-D with distance cost and SLOT-D with quantized distance-based cost.

7.6 Conclusion

The Synchronized Link Overlay Triangular (SLOT) technique builds and maintains a distributed connected and spanning overlay on top of a network, relying only on local information about the links towards the 1-hop neighbors of each router. The resulting overlay can be used for LSDB synchronization, so that each pair of routers connected by overlay links are able to efficiently exchange critical data. Two variations are analyzed: one relying in a unit link cost (SLOT-U), and other in which link cost is related to link length (SLOT-D).

SLOT overlays have two remarkable characteristics related to overlay link density and overlay link change rate. For both variations, it can be proven (under the unit disk graph model) that overlay densities (average number of overlay links per router) does not depend on the network link density, that is, there is an upper bound for the overlay densities that is independent from network density $\sim O(\nu)$. Concerning the overlay link change rate, this chapter has shown that SLOT-U provides a per router synchronization rate independent from the network density, and proportional to the average node speed s. When the cost is based on distance (SLOT-D) and not in hop count (SLOT-U), the synchronization rate increases from $O(s)$ (SLOT-U) to $O(s\sqrt{\nu})$ (SLOT-D), which nevertheless remains drastically lower than the total link change rate $O(\nu s)$. These characteristics are interesting for LSDB synchronization overlays, as the reduction in the number of overlay links

and the rate of overlay link changes both imply a reduction in the number of LSDB synchronizations to be performed in the network.

Comparing the two variations of SLOT, SLOT-D is expected to perform better than SLOT-U in real deployments for synchronization processes. Indeed, SLOT-D produces more optimized (in terms of density) overlays than SLOT-U. However, this requires that routers are able to extract information from the network about the link length, which is not needed for SLOT-U. Also, SLOT-D has a link change rate that increases with the network density, with a $\sqrt{\nu}$ term that can become significant for dense networks.

Chapter 8

Multi-Point Relays – MPR

Multi-Point Relaying (MPR) is primarily a technique for efficient flooding in wireless ad hoc networks, in which flooding decisions are based on local information about the 2-hop neighborhood of the corresponding source.

MPR has been widely studied in literature as a flooding technique, since it was first presented by Qayyum *et al.* in 2002 [60, 83, 88]. Several algorithmic improvements over MPR flooding have been also proposed and evaluated [40, 67].

This technique can however be used for additional purposes, in particular for performing the other link-state operations presented in chapter 4 – LSDB synchronization and topology selection. This chapter examines the characteristics of the Multi-Point Relays technique, presents different overlays based on MPR for performing each of the main link-state operations and evaluates the properties of such overlays in light of the requirements detailed in chapter 6 for each of them.

8.1 Outline

Section 8.2 presents the MPR technique and details the main heuristics from literature proposed for Multi-Point Relays selection. Section 8.3 describes the overlay created by MPR for flooding purposes. Section 8.4 presents the LSDB synchronization overlay based on MPR, analyzes

its asymptotic properties of overlay connection and density and discusses the stability of the overlay in mobile ad hoc networks. Section 8.5 examines the *Path MPR* overlay for topology selection purposes. Resulting from this analysis, an improvement of the Path MPR overlay is proposed and evaluated, in order to guarantee that this overlay fulfills the requirements of chapter 6 for topology selection. Finally, section 8.6 summarizes the main properties of MPR and concludes the chapter.

8.2 Definitions and Heuristics

Multi-Point Relays provide an efficient way of disseminating information in ad hoc networks. Instead of requiring every neighbor of a source, s, to retransmit a message from s (pure flooding), MPR flooding requires that s selects a subset of its neighbors (the MPR set of s), and that only these neighbors retransmit messages from s. Neighbors not included in the MPR set are thus excluded from the flooding operation.

Election of such multi-point relays permits to achieve the same coverage as the one obtained by allowing every 1-hop neighbor to transmit, while reducing significantly the number of redundant transmissions – as Figure 8.1 illustrates.

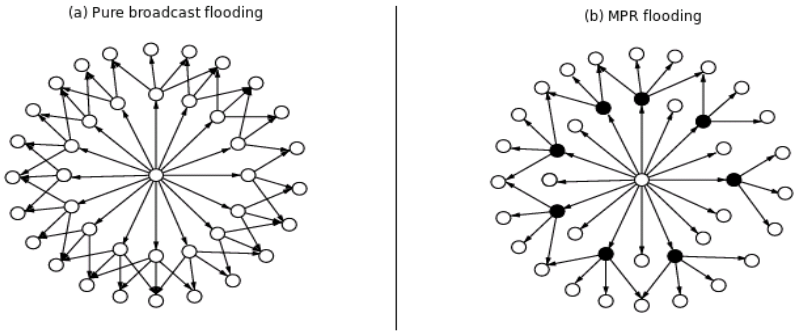


Figure 8.1: **(a)** Pure flooding *vs.* **(b)** flooding based on the Multi-Point Relays (MPR) principle. *Solid dots in (b) represent multi-point relays.*

The selection of relays must ensure that the set of 2-hop neighbors will receive all messages disseminated by the source. Therefore, the MPR selection heuristics must satisfy the following condition:

Definition 8.1 (**MPR coverage criterion**). Every 2-hop neighbor of the computing router must be reachable by (at least) one of the selected multi-point relays.

Therefore, an MPR set of a router x can be formally defined as follows:

$$R(x) \subseteq N(x) \text{ is an MPR set of } x \Longleftrightarrow \forall z \in N_2(x), \exists y \in R(x) : z \in N(y) \tag{8.1}$$

Links connecting routers with those neighbors selected as multi-point relays denoted *MPR links*.

Definition 8.2 (**MPR link**). A link between interfaces a and b is a *MPR link* if and only if:

(i) a is selected MPR of b, or

(ii) b is selected MPR of a.

a is then an *multi-point relay* (MPR) of b, and b is called a *MPR selector* of b.

8.2.1 Heuristics

Different heuristics can be used for selecting multi-point relays, all legitimate as long as they satisfy the MPR coverage criterion. This chapter uses the iterative greedy heuristic indicated in (8.2), proposed and analyzed in [88], studied in detail in [83] and [60] and used, for example, in OLSR [71].

$$\begin{cases} (1)\ MPR(x) = \{\emptyset\} \\ (2)\ MPR(x) \longleftarrow \{y_{excl} \in N(x) : y_{excl} \text{ provides exclusive coverage to one or more 2-hop neighbor(s) of } x\} \\ (3)\ while(\exists \text{ uncovered 2-hop neighbors of } x), \\ \qquad MPR(x) \longleftarrow y \in N(x) : y \text{ covers the maximum \# of uncovered 2-hop neighbors of } x \end{cases} \tag{8.2}$$

Step (2) could be removed from the heuristic without affecting correctness. Relays selected in that step would be selected anyways before every 2-hop neighbor is covered, since there are 2-hop neighbors only covered by them. By including step (2), the algorithm converges faster [88].

Step (3) from heuristic (8.2) first selects those relays that provide coverage to most of 2-hop neighbors. As the distance between the source and one 1-hop neighbor increases, more 2-hop neighbors may be covered by that 1-hop neighbor of the source, as the area coverage of the 1-hop neighbor not included in the area coverage of the source also increases. Therefore, this MPR heuristic gives priority to relays located further from the source, under the assumption that all other conditions (transmission power, error correction mechanisms, interference, etc.) are equal. It has been established [40, 32, 67] that the quality of the "coverage" provided by a 1-hop neighbor strongly depends on the distance to the source and to the covered 2-hop neighbors. The selection of relays based only on the quantitative maximization of the set of uncovered neighbors, as in (8.2), is likely to have a negative impact on the quality of coverage, since neighbors that are further from the source have also longer and possibly less reliable links. New heuristics taking into account the reliability of MPR links are thus proposed, *e.g.* in [67]: for routers able to estimate the quality of the links in their neighborhood, these improved heuristics enable 2-hop neighbors to be covered by links that are more reliable, at the expense of increasing the number of MPRs. These heuristics propose the use of other criteria for MPR selection, different to maximization of the number of covered 2-hop neighbors. They can be seen as extensions and refinements of the basic heuristic of (8.2).

8.2.2 Implications

Any MPR heuristic requires that the source has information about its 2-hop neighbors. Discovery of the 2-hop neighborhood by the source can be performed by way of exchange of Hello packets between neighbors, as indicated in section 3.4.1. Dependency on the 2-hop neighborhood information provided by Hellos has two effects on the MPR properties:

- Since MPR selection for a router may become obsolete due to a change in the 2-hop neighborhood, stability of the MPR set is not only affected by the conditions of MPR links, but

also by MPR recalculations due to changes within the 2-hop neighbors or the way in which they are connected to the 1-hop neighbors of the source (see Figure 8.2). Frequent changes in the 2-hop neighborhood of a router may cause an excessive MPR link change rate. This has further implications for the application of MPR-based overlays in link-state operations such as LSDB synchronization, as described in section 8.4.

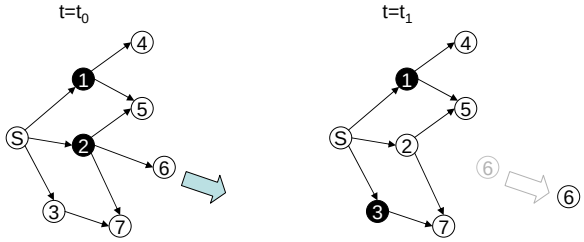


Figure 8.2: MPR recalculation due to changes in the 2-hop neighborhood. *Solid dots represent relays of router S.*

- There is a delay between the time in which a router, y, becomes 2-hop neighbors of a source, s, and the time in which y is taken into consideration for computing the MPR set of s. This delay depends on (i) the time in which y advertises its own presence (via Hello), and (ii) the time in which the 1-hop neighbors of s advertise the presence of y to s. (ii) is related to the network router density, as *Lemma 8.1* points out (see also Figure 8.3). Delay caused by (i) and (ii) is still increased with the interval between the MPR election and the notification to the corresponding neighbor – typically, by way of a Hello message. For sparse MANETs with highly mobile routers, these delays might have a non-negligible impact in terms of MPR selection staleness and, consequently, in the performance of the link-state routing operations relying on it.

  Lemma 8.1. *Assume that routers in a wireless network are distributed according to a Poisson punctual process of rate (node density) ν. Routers have a uniform radio range r (unit disk graph assumed) and all of them advertise their presence and the list of* heard *neighbors with the periodic transmission of Hello messages, at a uniform rate HI. Consider a router s and a router y appearing into the 2-hop neighborhood of s, at distance $d \in [r, 2r]$ of the source. Then, the time τ between the instant in which y appears in the 2-hop neighborhood of s and the instant in which s takes y into consideration for*

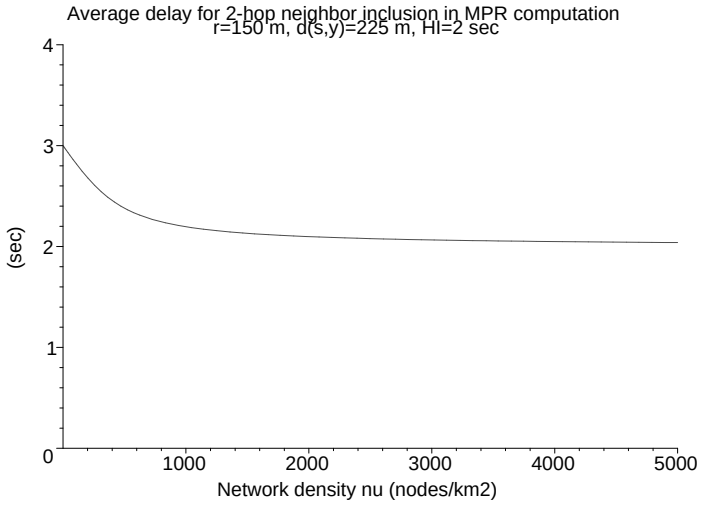


Figure 8.3: Average delay for the inclusion of a 2-hop neighbor in the MPR computation, according to (8.3).

selecting its multi-point relays can be modeled as:

$$E\{\tau\}(d) = HI\left(1 + \frac{e^{-\nu A_r(d)}}{1 - e^{-\nu A_r(d)}} \sum_{i=1}^{\infty} \frac{(\nu A_r(d))^i}{i!(i+1)}\right) \tag{8.3}$$

that depends on the network density and presents an asymptotic behavior such that $E\{\tau\} \longrightarrow HI$ *when* $\nu \longrightarrow \infty$. $A_r(d)$ *denotes the area of intersection between two circles of radius* r*, at distance* d*,* i.e.*:*

$$A_r(d) = 4\int_{d/2}^{r} \sqrt{r^2 - x^2}dx \tag{8.4}$$

Proof. Let s be a source, x be a bidirectional neighbor and y be a new 2-hop neighbor, all three nodes configured with the same Hello interval HI. Consider the time interval between the moment at which y appears in the 2-hop neighborhood of s (*i.e.*, becomes reachable from s in 2 hops) and the moment in which s takes it into consideration in the MPR computation (*i.e.*, s has discovered the presence of y within its 2-hop neighborhood). For the sake of simplicity, assume that y sends a Hello message when it first appears in the 2-hop neighborhood. The delay until it establishes bidirectional communication with x (3-step Hello handshake, see Figure 3.3) is HI. s will notice the presence of y after the reception of the following Hello from x. The time until this Hello is sent can be modeled as a random uniform variable τ_1 within $[0, HI]$, the overall delay τ thus being:

$$\begin{aligned} \tau &= HI + \tau_1 \\ E\{\tau\} &= \frac{3}{2} HI \end{aligned}$$

Consider the case in which n bidirectional neighbors of s ($n \geq 1$) provide coverage to y. Then, $\tau_n = min_{1 \leq i \leq n}\{\tau_1\}$ and $f_{\tau_n}(t) = \prod_{i=1}^{n} f_{\tau_1}(t) = f_{\tau_1}^n(t) = \left(\frac{t}{HI}\right)^n$ for $0 \leq t \leq HI$.

Assume that nodes (excluding s and y, fixed at distance $d > r$) are distributed over the network according to a Poisson punctual process with density ν. In this case, the delay τ can be computed as:

$$\begin{aligned} \tau|(n \geq 1) &= \frac{1}{1 - e^{-\nu A_r(d)}} \sum_{i=1}^{\infty} \tau_i e^{-\nu A_r(d)} \frac{(\nu A_r(d))^i}{i!} \\ E\{\tau|n \geq 1\} &= \sum_{i=1}^{\infty} E\{\tau_i\} \frac{e^{-\nu A_r(d)}}{1 - e^{-\nu A_r(d)}} \frac{(\nu A_r(d))^i}{i!} = \sum_{i=1}^{\infty} \frac{e^{-\nu A_r(d)}}{1 - e^{-\nu A_r(d)}} \frac{(\nu A_r(d))^i}{i!} \int_{t=0}^{t=HI} dt \left(\frac{t}{HI}\right)^i = \\ &= \sum_{i=1}^{\infty} \frac{e^{-\nu A_r(d)}}{1 - e^{-\nu A_r(d)}} \frac{(\nu A_r(d))^i}{i!} \frac{HI}{i+1} = \frac{HI e^{-\nu A_r(d)}}{1 - e^{-\nu A_r(d)}} \sum_{i=1}^{\infty} \frac{(\nu A_r(d))^i}{i!(i+1)} \end{aligned}$$

□

8.3 MPR as a Flooding Overlay

MPR flooding introduces a directed overlay for every flooded message, by requiring a router to forward the message if and only if the following two conditions are satisfied:

(1) the message comes from a MPR selector, and

(2) it is the first time the message is received by that router.

Condition (2) ensures that the flooding process terminates in a finite number of steps. The number of multi-point relays (MPRs) of a router is an upper bound for the number of neighbor (re)transmissions caused by a single broadcast transmission of the router. Both values are the same in the first step of a flooding process, and the latter may decrease with respect to the former in further steps, if there are MPRs of intermediate forwarders that do not satisfy (2). During the

flooding of a message is performed over the network, an increasing part of MPRs of intermediate forwarders have already received the message and thus do not forward it again, until the message reaches routers for which every neighbor has received a copy, and the flooding terminates.

This directed overlay, formed by routers participating in flooding of a message sent by a source s as multipoint relays of s or any of the forwarders, is known to be a Connected Dominating Set (CDS) that depends on the source s [60].

8.4 MPR as a Synchronized Overlay

Multi-Point Relays can also be used for synchronization purposes. Two neighbors synchronize their local instance of LSDB if any of them has selected the other as multi-point relay – that is, if it is a MPR link (def. 8.2). The MPR synchronized overlay thus contains the same links as the MPR flooding overlay. The MPR synchronized overlay is undirected (not directed as the MPR flooding overlay), due to the symmetric nature of the LSDB synchronization operation (see section 6.2.2).

The MPR synchronized overlay is also a denser overlay (that is, with more links per router) than the MPR flooding overlay, given that all MPR links in the network (and not only those that would participate in flooding from a particular source) are included. In the MPR synchronized overlay, a router is connected to all its MPRs and MPR selectors – not only to its MPRs as in the MPR flooding overlay. Figure 8.4 shows the average number of links between a router and its MPRs (MPR flooding overlay density) and the average number of links between a router and its MPRs and MPR selectors (MPR synchronized overlay density), computed via simulations in static, error-free networks with uniformly distributed routers.

8.4.1 Asymptotic Connection and Density

As mentioned in chapter 6, a synchronized overlay needs to be an asymptotically connected overlay. The overlay formed by all MPR links in the network does not necessarily satisfy this condition. Figure 8.5 illustrates two examples of networks in which the union of MPR links (represented

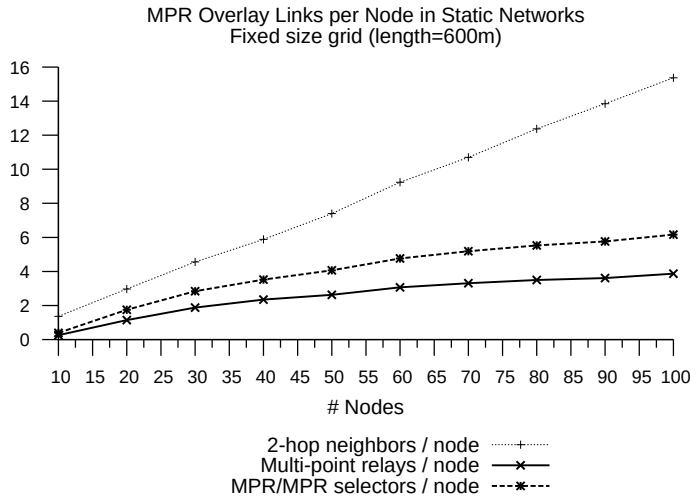


Figure 8.4: Density of MPR overlays for a static, error-free network. *Results from simulations.*

with thick lines, directed from the source to the MPR) produces disconnected overlays. Figure 8.5.b shows that disconnected overlays as such are possible with networks of arbitrary diameter (k).

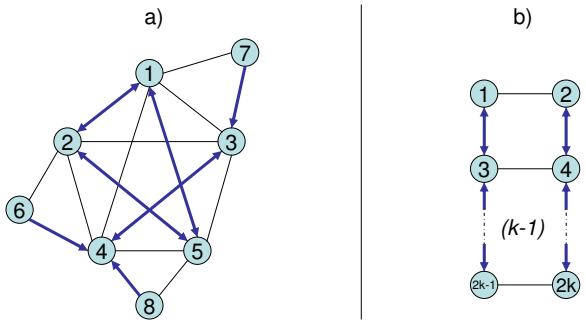


Figure 8.5: **(a)** Disconnection of the MPR set. *Thick directed lines represent MPR selection relationships.* **(b)** Disconnection of the MPR set in a k-diameter network.

Lemma 8.2 proves that, in case of disconnection of the MPR synchronized overlay, all its connected components are dense in the network – meaning that any vertex of the network graph is at distance 1 or 0 from all these components.

Lemma 8.2. *Let $G = (V, E)$ be a network graph, and let $H \subseteq G$ be the subgraph of G containing the links from every vertex in the graph to all its MPRs. Then, every connected component of H is dense over G.*

Proof. Let $H^{cx} \subseteq H$ be a connected component of H. Consider $x \in H^{cx}$. By induction over k, every vertex $z \in G$ at a distance k (in hops, $k < \infty$ because G is connected) from x has (at least) a neighbor that belongs to H^{cx}:

- $k = 1$ is trivial, from the definition.
- $k = 2$, then z is a 2-hop neighbor of x and, by definition of the MPR, there will be a vertex $y \in N(x) \cap N(z)$ so that $\overline{xy} \in H^{cx}$.
- $k \Longrightarrow k+1$. Consider the vertex $y \in G$ satisfying $dist(x, y) = k$, $y \in N(z)$. Note that vertex y exists because $dist(x, z) = k+1$, and by induction hypothesis, y is at a distance ≤ 1 from H^{cx}. Let t be the closest vertex of H^{cx} to y. Then, t is either a neighbor or a 2-hop neighbor of z; in both cases, the argument for $k = 1, 2$ concludes that the $dist(z, H^{cx}) \leq 1$, and thus H^{cx} (and, more in general, every connected component of H) is dense in G.

□

As every connected component of the MPR overlay is dense in the network, the MPR synchronized overlay becomes necessarily connected when all links of any single router belong to the overlay. This provides a sufficient condition for the connection of the overlay, which is proved in the following *Lemma 8.3*.

Lemma 8.3. *Let $G = (V, E)$ be a network graph, and $H \subseteq G$ the subgraph of G consisting of:*

1. *$H_1 \subseteq G$: For every vertex $x \in V$, the edges from x to the neighbor vertices selected by x as MPRs.*
2. *$H_2 \subseteq G$: For a certain $s \in V$, the edges from s to every neighbor of s.*

Then, H is connected.

Proof. In case that there are several connected components of H_1 (that is, H_1 is disconnected), all components are known to be dense over G (Lemma 8.2), *i.e.*, every vertex of G has at least a neighbor belonging to each of them. The subgraph that results from adding the links from any vertex of G (say $s \in G$) to all its neighbors (H_2) to H_1 will necessarily be connected. Note that the argument is valid for an arbitrary s. □

Under these conditions, the MPR-based overlay G_S defined in (8.5) is asymptotically connected.

$$\begin{cases} V(G_S) = V(G) \\ E(G_S) = \{\overline{xy} \in E(G) : x \in MPR(y) \vee y \in MPR(x) \vee (x \equiv s) \vee (y \equiv s), s \in V(G)\} \end{cases} \tag{8.5}$$

8.4.2 Link Change Rate and Persistency

As shown in section 8.2.1, MPR links present a high change rate due to the multi-point relay dependence on 2-hop neighborhood variations. Figure 8.6 illustrates the stability of MPRs when compared to bidirectional neighbors, for a moderately mobile ad hoc scenario (results from simulations, see Appendix E for a detailed description of configuration and parameters).

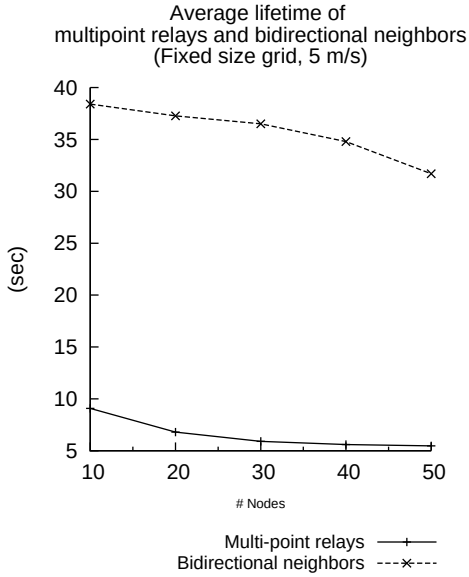


Figure 8.6: Average link lifetime for MPRs and bidirectional neighbors. *Simulation of a moderately mobile ad hoc network (5 m/s).*

Unstable and short-lived links are not desirable for LSDB synchronization purposes. The synchronization between two routers that have established a MPR relationship consists of exchanging and keeping updated their respective Link State Databases (LSDB). It is therefore an expensive process in terms of overhead that may generate an excessive amount of control traffic if it has to be performed too often due to changes in the overlay. The fact that a link has a short lifetime in the

overlay reduces as well the benefits of running a full synchronization process over it, while keeping untouched the cost of synchronizing.

The notion of overlay *persistency* permits partly overcoming these inconvenients by improving artificially the stability of synchronized links within the overlay.

Definition 8.3 (**Persistent Overlay**)**.** A (synchronized) overlay is *persistent* if the condition which a link needs to satisfy in order to be included in the overlay is not the same as the condition for a link in the overlay to not be removed, and the latter condition is less strict than the former. In case the conditions for overlay link inclusion and maintenance are the same, the overlay is **non-persistent**.

Definition 8.4 (**Persistent Link**)**.** In a persistent overlay, a link is *persistent* if it belongs to the overlay, but does not satisfy the condition that links not belonging to the overlay need to satisfy in order to be included. In case that the overlay link satisfies this condition for inclusion in the overlay, it is called **non-persistent**.

Figure 8.7 shows the Finite States Machines (FSM) corresponding to persistent and non-persistent approaches. In Figure 8.7.a, bidirectional links are upgraded to the status of synchronized when they fulfill *synch_condition*, and degraded back to the bidirectional (non-synchronized) status when they stop fulfilling it. In Figure 8.7.b, in contrast, synchronized links are not degraded except that they are no longer bidirectional.

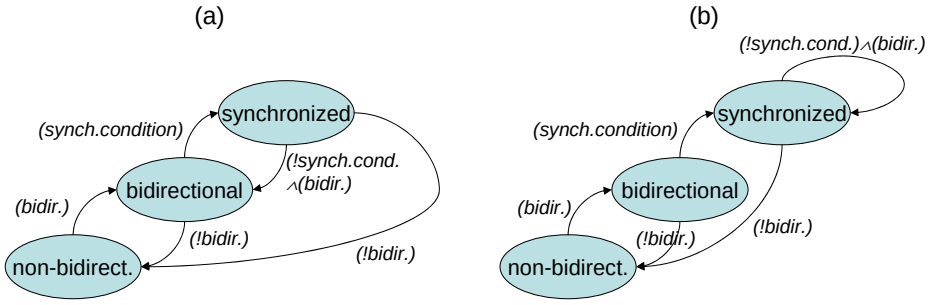


Figure 8.7: **(a)** Non-persistent and **(b)** persistent approaches for link synchronization.

Implementation of persistency in the MPR synchronized overlay leads to the *persistent*

MPR synchronized overlay. This overlay includes, for each router, the existing links to all bidirectional neighbors that had been selected as MPR by this router, even if they were later removed from the MPR set of the router. Once an MPR is elected, the corresponding link is only removed from the synchronized overlay when it loses bidirectionality – or it disappears. When compared to the original MPR synchronized overlay, as defined in (8.5), two main differences can be observed:

- As expected, links in the persistent overlay are more stable (in terms of average lifetime in the overlay) than those of the non-persistent overlay, since the links oscillating between MPR and bidirectional non-MPR status do not oscillate anymore, and for the links that are eventually removed, the removal is delayed until the instant in which the corresponding link is not bidirectionally reachable.

- The persistent overlay is significantly denser (it contains more links) than the non-persistent, and the gap between the two grows bigger as the network becomes less stable (due to mobility or to wireless channel variations). This growth of the overlay, however, does not cause a significant increase in the associated overhead: additional links have been already synchronized, so the cost of maintaining them (in terms of control traffic), if any, is limited to acknowledgment of topology updates – in case reliable transmission is implemented over synchronized links.

The impact of persistency in MPR synchronized overlays deployed over mobile ad hoc networks is further analyzed in Part III.

8.5 MPR as a Topology Selection Rule

Section 6.2 has established that the main requirement for an overlay of advertised links (topology selection overlay) is that it is a spanning subgraph that contains the network-wide shortest paths to all destinations.

Computation of shortest paths involves a metric, *i.e.*, a link cost function which gives sense to the notion of *shortest*. As the MPR mechanism is defined in terms of coverage requirements, rather than cost minimization objectives, it becomes necessary to translate the cost-based optimality

considerations in terms of optimal coverage, in order to reuse and extend MPR as efficient topology selection mechanism.

8.5.1 Path MPR

[24] proposes and specifies a topology selection rule based on MPR, called *Path MPR*. The Path MPR algorithm intends to "provide the router with a Path-MPR set (..) such that for any element of N or N_2 that is not in the Path-MPR set, there exists a shortest path that goes from this element to the router through a neighbor selected as Path-MPR (unless the shortest path is only one hop)" [24]. The subgraph generated by Path MPR selection in every node of the network should thus include, for any node x of the network, the links to x from the neighbors providing local shortest paths (w.r.t. a given *cost* function) from the 2-hop neighborhood of x and to x. These links are directed, meaning that Path MPR supports links with different costs depending on the direction.

The Path MPR algorithm extracts, from the set of 1-hop neighbors of the computing node x, a subset of neighbors (called $N'(x)$) for which the link to x is a local 2-hop shortest path w.r.t. the current metric – that is, there are no other paths of 2 hops, that provide a better (cheaper) cost from that 1-hop neighbor to x (def. 6.6). The algorithm also extracts from the set of 2-hop and 1-hop neighbors of x ($N(x) \cup N_2(x)$), a subset of neighbors (called $N'_2(x)$) for which the local 2-hop shortest path has exactly 2 hops. Then, it executes the MPR algorithm from x over the 2-hop neighborhood subgraph resulting from considering $N'(x)$ as 1-hop neighborhood and $N'_2(x)$ as 2-hop neighborhood. The algorithm can be summarized as follows:

(8.6)

1. **Input:** x, $N(x)$, $N_2(x)$.
2. The following subsets, $N' \subseteq N$, $N'_2 \subseteq N \cup N_2$, are calculated:

$$\begin{cases} N' &= \{n \in N | cost(x,n) = dist(x,n)\} \\ N'_2 &= \{n \in N, N_2 | n \notin N', \exists m \in N' : cost(n,m) + cost(m,x) = dist_2(n,x)\} \end{cases}$$

3. The router runs the MPR selection procedure with arguments x, $N'(x)$ and $N'_2(x)$.

4. **Output:** $PathMPR(x, N, N_2) = MPR(x, N', N_2')$

It is worth noting that the Path MPR algorithm is a MPR-based topology selection algorithm in the sense of section 6.2.3. Therefore, the requirements indicated in that section apply.

Correctness in Unit Link Costs Scenarios

Assume that the network links have a uniform cost, that is,

$$cost(e) = 1, \forall e \in E(G) \tag{8.7}$$

where G is the network graph. Then, the sets $N'(x)$ and $N_2'(x)$ computed by the Path MPR algorithm from a node x are expressed as follows:

$$\begin{cases} N'(x) &= \{n \in N | cost(x,n) = dist_2(x,n)\} = [cost(x,n) = dist_2(x,n) = 1] = N(x) \\ N_2'(x) &= \{n \in N(x) \cup N_2(x) | n \notin N'(x), \exists m \in N'(x) : cost(n,m) + cost(m,x) = dist_2(n,x)\} = \\ &= [N(x) = N'(x)] = \{n \in N_2(x) | \exists m \in N(x) : cost(n,m) + cost(m,x) = dist_2(n,x)\} = \\ &= [dist_2(n,x) = cost(n,m) + cost(m,x) = 2] = N_2(x) \end{cases} \tag{8.8}$$

In these conditions, the Path MPR algorithm produces the same result as MPR. Neighbors selected as Path MPRs by router x provide coverage to/from every 2-hop neighbor of x. Since all paths from 2-hop neighbors to x have a cost of 2 (number of hops), that trivially means that the Path MPR algorithm provides local shortest paths in unit link cost networks.

Correctness in Arbitrary Link Costs Scenarios

In case that the link costs take non-uniform values (*e.g.*, because they use router energy metrics [8], or link reliability metrics such as ETX [82]), the Path MPR algorithm may be unable to identify neighbors that provide shortest paths from 2-hops neighbors towards the computing node. Figure 8.8 illustrates a simple example in which the Path MPR algorithm computed on node *1* selects a neighbor not providing local shortest paths from the 2-hop neighbors of *1* to *1*.

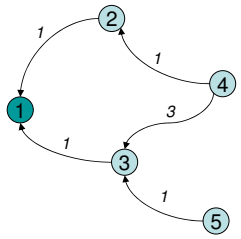


Figure 8.8: Path MPR malfunctioning example, with respect to router (1).

In this case, the sets $N'(1)$ and $N'_2(1)$ have the following composition:

$$\begin{cases} N'(1) &= \{n \in N(1) : cost(n,1) = dist_2(n,1)\} = \{2,3\} \\ N'_2(1) &= \{m \in N(1) \cup N_2(1) : \exists n \in N'(1) : cost(m,n) + cost(n,1) = dist_2(n,1)\} = \{4,5\} \end{cases}$$

Thus, according to the algorithm presented in (8.6), the output from the Path MPR selection would be $PathMPR(1) = \{3\}$, since node (3) would be sufficient for covering all nodes in $N'_2(1)$ (MPR coverage criterion). This election would nonetheless not contain the shortest path from (4) to (1), $p^*_{41} = \{\overline{42}, \overline{21}\}$.

The problem shown in figure 8.8 is caused by the fact that MPR is a cost-agnostic algorithm that relies only on coverage, while the Path MPR algorithm is expected to select links according to cost minimization rules. By executing MPR selection on x over the subgraph formed by $N'(x)$ and $N'_2(x)$, the algorithm may select vertices of $N'(x)$ providing sub-optimal paths (in terms of *cost*) from $N'_2(x)$ to x, if they provide better coverage (in terms of number of covered vertices belonging to $N'_2(x)$) than the vertices providing optimal (local shortest) paths.

8.5.2 Enhanced Path MPR

This section proposes a modification of the previously presented Path MPR mechanism. Figure 8.9 displays the input/output block diagram of this approach, called *Enhanced Path MPR* (ePMPR).

The cost-coverage translation block (see Figure 8.9) extracts the subgraph of (local) shortest

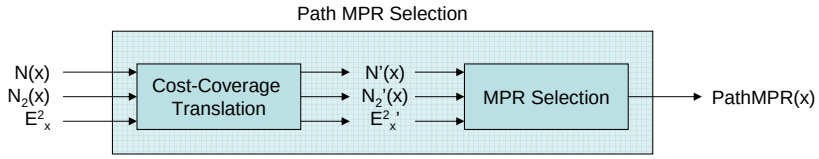


Figure 8.9: Block diagram for a MPR-based topology selection algorithm. $E_x^2 \subset E(G)$ *are the set of edges connecting vertices within* $x \cup N(x) \cup N_2(x)$.

paths from the 2-hop and 1-hop neighbors of x to x. Vertices of this subgraph include x, $N'(x)$ and $N_2'(x)$, while $(E_x^2)'$ is the set of edges. $N'(x)$ contains those routers from $N(x)$ for which the link to x is also the local (2-hop) shortest path to x; and correspondingly, $N_2'(x)$ contains those routers from $N_2(x)$ for which the optimal path from x has 2 hops. Finally, $(E_x^2)'$ contains those edges (links) of E_x^2 that participate in at least one shortest path from a 1-hop or 2-hop neighbor of x to x. The formal definition for the output of the cost-coverage translation block is as follows:

$$\begin{cases} N'(x) = & \{n \in N(x) | m(x,n) = dist_2(x,n)\} \subseteq N(x) \\ N_2'(x) = & \{n \in N(x) \cup N_2(x) | n \notin N'(x), \exists m \in N'(x) : m(n,m) + m(m,x) = dist_2(n,x)\} \subseteq N(x) \cup N_2(x) \\ (E_x^2)' = & \{\overline{nm} \in E(G) : n \in N'(x), m \in N_2'(x), m(x,n) + m(n,m) = dist_2(x,m)\} \cup \\ & \cup \{\overline{xn} \in E(G) : n \in N'(x)\} \subseteq E_x^2 \end{cases} \tag{8.9}$$

Definitions for $N'(x)$ and $N_2'(x)$ are identical to those used in original Path MPR. The difference between Path MPR and the Enhanced mechanism is that the latter is able to prevent those links that do not participate in local shortest paths to take part in the MPR computation. Operation of the Enhanced Path MPR mechanism is shown with an example in Figure 8.10.

From definitions of (8.9) it follows that the Enhanced Path MPR mechanism, as defined in Figure 8.9, returns a set of relays that provide (local) shortest paths from every 2-hop neighbor of x to x: if a path $p_{zy} = \{\overline{zy}, \overline{yx}\}$ is not optimal, with $y \in N'(x)$ and $z \in N_2'(x)$, then $\overline{yz}$ will not belong to $E(S_x')$. That ensures that Enhanced Path MPR is able to select the local (2 hops) shortest paths to the computing router x, given that every 2-hop neighbor of x is included in $N_2'(x)$.

A topology selection mechanism based on the advertisement by each router of the En-

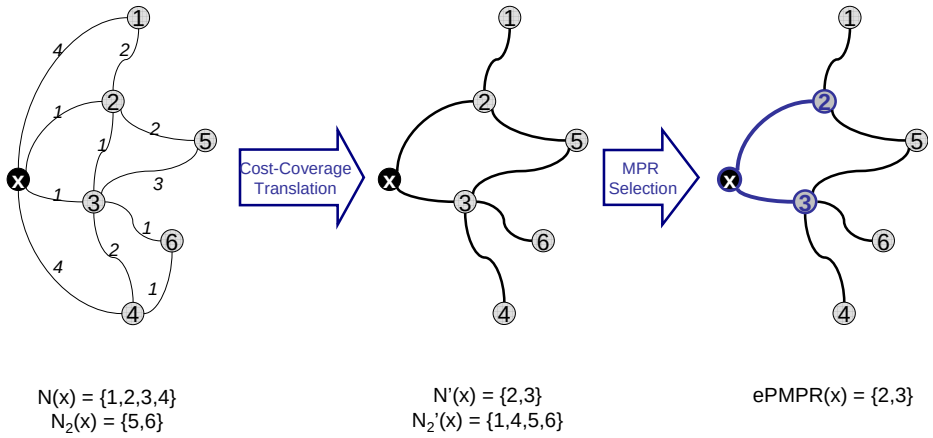


Figure 8.10: Enhanced Path MPR operation over the 2-hop neighborhood of router x.

hanced Path MPR set, generates a network-wide overlay that contains, for every router x, the 1-hop neighbors of x that provide shortest paths (in a 2 hop scope) from 2-hop neighbors of x to x. The requirements for topology selection overlays identified in section 6.2 included however:

- Overlay connection.
- Preservation of network-wide (and not only local) shortest paths.

Connection of an MPR overlay can be achieved (*Lemma 8.3*) by adding to the overlay all the links maintained by a single arbitrary router. *Lemma 8.4* shows that the overlay that results of adding this additional router (the computing router itself, for Path MPR) contains network-wide shortest paths from every destination of the network to the computing router:

Lemma 8.4. *Let $G = (V, E)$ be a connected network graph, an edge metrics function cost$(e \in E(G))$, a router $s \in V(G)$ and a subgraph $G'_s = (V, E'_s)$ including:*

1. *the edges connecting s to its 1-hop neighbors, and*
2. *for every router x of the network, the edges from x to those 1-hop neighbors of x providing local shortest paths from every 2-hop neighbor of x to x.*

Then, the Dijkstra algorithm computed on a source router s over G'_s selects the shortest paths in G from the source to every possible destination.

Proof. Since the Dijkstra algorithm selects the shortest paths of the graph (w.r.t. a given metrics *cost*) over which it is computed, it needs to be proved that the shortest paths from s in G are contained in G'_s, *i.e.*, $SPT_s(G) \subset G'_s \subset G$. Let z be an arbitrary router $z \in V$, $\overline{sz}_{sh-p}$ be the shortest path (w.r.t. *cost*) between s and z, and let $d(x, y)$ be the distance in hops between x and y.

- If $d(s, z) = 1$, $\overline{sz}_{sh-p} \in G'$ by condition 1 of the hypothesis.
- For $d(s, z) = n > 1$, let $\{m_i\}$ be the intermediate routers of $\overline{sz}_{sh-p}$, so that $d(s, m_i) = i$. The edge $\overline{sm_1}$ belongs to G'_s by definition of G'_s (condition 1). The edge $\overline{m_i m_{i+1}}$ (consider $\overline{m_1 z}$ if $n = 2$) is included in G'_s because m_i is part of the local shortest path from s (2-hop neighbor of m_{i+1}) to m_{i+1} (condition 2 of the hypothesis about G'_s). Repeating the argument along $\overline{sz}_{sh-p}$ for $\{m_j\}_{1 \leq j < n}$, leads to the conclusion that all segments $\overline{sm_1}, ..., \overline{m_i m_{i+1}}, ..., \overline{m_{n-1} z}$ belong to G'_s and thus $\overline{sz}_{sh-p}$ belongs too.

□

As other improvements are possible (such as including not only $N(x)$ but also $N_2(x)$), the previous lemma states a sufficient condition for the asymptotic correctness of an MPR-based topology selection overlay.

8.6 Conclusion

The Multi-Point Relays (MPR) technique is known and has been widely studied in the literature as an efficient distributed flooding mechanism for wireless multi-hop networks that only requires router's local knowledge of their 2-hop neighborhood. As a flooding technique, MPR is able to generate a flooding overlay that reaches every node in the network with a significant reduction in the number of transmissions (therefore, the network links contained in the overlay) with respect to the pure flooding procedure (full network flooding overlay), as it can be observed in Figure 8.4.

The MPR principle is also useful for the other operations related to link-state routing, MPR-based overlays being thus suitable as well for link synchronization and topology selection purposes.

Concerning topology reduction, the chapter focuses on the Path MPR mechanism specified in [24]. While this mechanism is correct for unit link cost, it does not guarantee the inclusion of network-wide shortest paths in its associated overlay with more general link metrics. The chapter identifies a sufficient condition for ensuring the inclusion of shortest paths. It also proposes a modification of Path MPR, so-called Enhanced Path MPR, such that the resulting overlay is granted to contain network-wide shortest paths, thus enabling routers to compute optimal routes to every destination in the network.

In terms of link synchronization, the overlay generated by all the MPR links in the network is granted to be a Connected Dominating Set (CDS) if all the links of one router in the network are included. As a synchronized overlay, however, the resulting MPR-based overlay has significant drawbacks: the overlay is significantly denser than the MPR flooding overlay and has a poor performance in terms of link stability when nodes are mobile. This is due to the fact that MPR links are sensitive to changes in the 2-hop neighborhood. Preserving the links in the overlay as far as they stay bidirectional (persistent MPR links), even if the connected endpoints have no longer an MPR relationship, improves the link stability of the synchronized overlay, at the expense of increasing its density.

The MPR principle can be used for building and maintaining flooding, link synchronization and topology selection distributed overlays only relying on the local information from the 2-hop neighborhood of the corresponding routers. However, the MPR-based overlays have better properties for flooding and topology selection purposes than for link synchronization objectives. In this latter case, both the density and the link change rate in mobile deployments can increase significantly the cost of updating the MPR-based synchronized overlay.

Chapter 9

The Smart Peering Technique – SP

The Smart Peering technique is a simple rule for constructing a network overlay in a distributed fashion. This rule enables routers to determine whether a bidirectional link should be included or rejected in the overlay. The Smart Peering overlay is used for link-state synchronization and flooding purposes.

Unlike the other techniques examined in chapters 7 and 8, Smart Peering does not only takes local information (from the 1-hop neighborhood for SLOT, from the 2-hop neighborhood for MPR) into consideration, but also information about the whole network topology. This feature turns the Smart Peering technique to be representative of the family of overlay techniques based on global information. The analysis performed in this chapter thus leads to general conclusions that apply for techniques in which nodes taking decisions about their links take into consideration information beyond its neighborhood.

9.1 Outline

Section 9.2 presents Smart Peering and details the way that links are selected to be part of the Smart Peering overlay. Section 9.3 describes asymptotic properties of the Smart Peering overlay, in particular the connection and spanning properties identified in chapter 6. Section 9.4 analyzes

the stability of Smart Peering links by focusing on a particular aspect of SP: the ability of this technique to select or reject links depending on the relative speed between the two attached routers. Section 9.5 concludes the chapter.

9.2 Definition and Specification

The Smart Peering rule was proposed in [45] as a mechanism for link-state database synchronization and flooding in ad hoc networks using the link-state routing protocol OSPF[1]. In Smart Peering, a router x synchronizes its local instance of LSDB with the local instance of LSDB of a bidirectional neighbor y if and only if:

- There are not *enough* available paths from x to y within the synchronized overlay (consisting on links already selected through Smart Peering).
- The link between x and y provides a *significantly* cheaper path from x to y than those already present in the synchronized overlay.

The precise meaning of *enough* and *significantly* defines different possible variations of Smart Peering. In this manuscript, the most elementary version is considered: a neighbor is synchronized if and only if the (already existing) synchronized overlay does not contain paths towards this neighbor. Figure 9.1 shows the Smart Peering flowchart for a router that detects a new bidirectional neighbor and decides whether it performs an LSDB synchronization with such neighbor or not.

Taking Smart Peering decisions requires that every router is able to determine whether a *synchronized path* (def. 4.2) exists over the network between itself and the neighboring router that candidate to synchronization.

The link between two routers is synchronized if either of the involved routers (but not necessarily both) decides to perform such synchronization. When using a link-state routing protocol,

[1]For a detailed description of OSPF, refer to chapter 10.

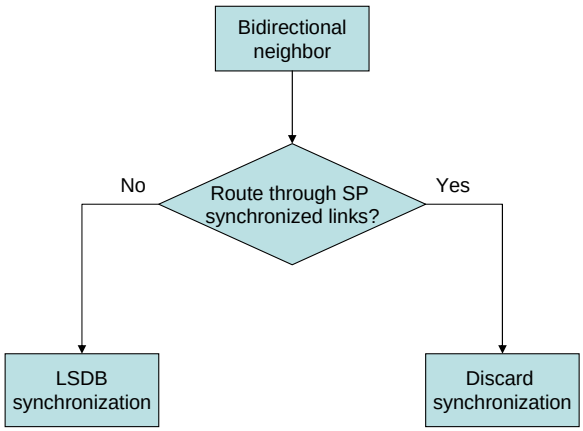


Figure 9.1: The Smart Peering (SP) flowchart for a router that detects a new bidirectional neighbor, for deciding whether to synchronize the link between itself and such neighbor.

such synchronized paths can be locally searched within the network topology information. This requires a topology selection mechanism in which routers advertise their synchronized links. This way, every router in the network is able to identify synchronized links within the links described in its local instance of LSDB.

9.3 Asymptotic Properties

The overlay provided by the Smart Peering rule fulfills the topological requirements for synchronized and flooding overlays, as defined in section 6.2: *Lemma 9.1* shows that Smart Peering decisions lead to an asymptotically connected overlay (def. 6.2). From the definition, since every router synchronizes its local instance of LSDB with at least one neighbor's, the Smart Peering overlay contains all routers in the network and is therefore an asymptotically spanning overlay (def. 6.4).

Lemma 9.1. *Using Smart Peering, every pair of routers* (x, y) *of a connected network are connected through at least one synchronized path.*

Proof. Let d be the minimum distance in hops from x to y ($d < \infty$). Let two routers be SP-connected if there is a synchronized path between them, with Smart Peering.

- $d = 1$: if x and y are not already connected via a synchronized path, the two routers will synchronize their local instances of LSDB, by definition of Smart Peering.

- $d \Rightarrow d+1$. Consider the set of bidirectional neighbors of x, $N(x)$. There exists at least one $z \in N(x)$ for which $d(z,y) = (d+1) - 1 = d$, and z is thus SP-connected to y (induction hypothesis). Denoting $\overline{xz}$ the SP-route between x and z (which exists as shown for the case $d = 1$), and $\overline{zy}$ the synchronized path between z and y, it is clear that the route $\overline{xz} \cup \overline{zy}$ is an synchronized path between x and y, and that concludes the proof.

□

Unlike the techniques presented in previous chapters, Smart Peering decisions do not depend exclusively on topology. The overlay, produced by the Smart Peering rule for a given ad hoc network, thus cannot be deduced from the relations between routers. Rather, it may be significantly affected by aspects such as the order of appearance of the routers in the network, the trajectory of mobile routers and their mobility patterns (speed, pauses in movement). Dependency on such mobility patterns is probably one of the most interesting features of the Smart Peering rule for mobile ad hoc networks, and it will be addressed in section 9.4.

For a static and stable network with error-free links, in which synchronization decisions are taken independently and concurrently, the overlay induced by Smart Peering is roughly equivalent to the full network overlay presented in section 6.3. When a router first appears in a network, and is discovered by its neighbors, none of them will have any entry corresponding to it in their local instance of LSDB. Consequently, all such neighbors will initiate synchronizations with this new router (the argument is also valid from the point of view of such new router).

9.4 Reaction to Mobility

In wireless ad hoc networks, where communication is subject to channel failures and packet losses, the Smart Peering overlay does not necessarily contain all links in the network. From the definition, a links is only synchronized when no other synchronized paths between the two attached interfaces are known to be available. For an unstable link –*i.e.*, a link that is only available part of the time–, such availability of synchronized paths is tested every time that the link is available after breaking down. The existence of available synchronized paths is more probable if the attached

interfaces have completed LSDB synchronization processes with some neighbors. Therefore, the probability that such link is synchronized decreases as the link is less stable.

For mobile scenarios, dependency on link stability is more clear, as Smart Peering excludes links between routers with high relative speed. Once a router, R, has completed its first synchronization process, and the existence of a synchronized link towards R is advertised to the whole network, no other router will perform a new synchronization with R as long as the LSDB entry corresponding to the synchronized link remains valid. Highly mobile routers will therefore have difficulties establishing new synchronized links after the completion of the first synchronization process, while routers with a low relative speed to their neighbors will have more chances to maintain their synchronized links.

This behavior is confirmed empirically (via simulations) in Part III, but it can be also illustrated theoretically, as *Proposition 9.2* shows:

Proposition 9.2. *Assume a linear stationary wireless network formed by k fixed nodes with wireless interfaces $\{n_i\}_{1 \leq i \leq k}$ positioned along a line, at distances d. Each node is reachable along a linear interval (denominated coverage interval) with radius r and length $2r$, centered in the node, with $d < r < 2d$. Consider that the node n_i is placed in the position:*

$$x_i = 2r + (i-1)d$$

Consider also a mobile wireless node, m, with the same coverage properties as fixed nodes. m is placed at $x = 0$ and moves in the direction of increasing x at a constant speed v.
Assume that all nodes (mobile and fixed):

- *periodically transmit Hello messages, with an interval HI,*
- *declare a neighbor dead (def. 3.3) if there is no Hello received within a time interval DI ($DI > HI$), and*
- *synchronize their local instances of LSDB using Smart Peering.*

Let s be the time that takes to synchronize the link between two nodes. Then, the number of complete (Smart Peering) synchronization processes performed by m decreases linearly with the speed v.

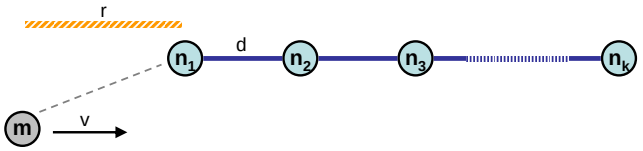


Figure 9.2: Scenario of *Proposition 9.2*.

Proof. The Smart Peering technique enables a node to synchronize the link with a neighbor if it is unable to find a synchronized path towards that neighbor in its local instance of LSDB. When a link is synchronized, both attached interfaces advertise the existence of such link to the rest of the network, by way of topology flooding. Similarly, nodes flood their topology descriptions when they detect that a synchronized link disappears (because the interfaces are no longer reachable to each other).

Given a speed v, the time that the mobile node m is reachable through the coverage interval of a fixed node, in a linear network, is $\frac{2r}{v}$. The time that two nodes need for establishing bidirectional communication and synchronizing their LSDB corresponds to $HI + s$, where s depends on to the size of the LSDB. The time that a node needs to detect that a neighbor is no longer reachable is DI. Two additional (binary) variables are introduced:

$$p(v) = H\left(\frac{2r}{v} - (HI + s)\right) = \begin{cases} 1 & , v \leq \frac{2r}{HI+s} \\ 0 & , \text{ otherwise} \end{cases}$$

$$x_i(v) = H\left(\frac{di}{v} - (DI + HI + s)\right) = \begin{cases} 1 & , v \leq \frac{di}{DI+HI+s} \\ 0 & , \text{ otherwise} \end{cases}$$

$p(v)$ indicates whether the mobile node stays within the coverage interval of a fixed node for duration necessary for performing a link synchronization. $x_i(v)$ indicates whether the mobile node, after having synchronized itself with a fixed node, will perform a new synchronization with the fixed node that is placed i positions later. This requires that the synchronized link is declared dead and a new synchronization process is completed while the mobile node is within the coverage interval of the new fixed node.

Let s_i represent whether the link $\overline{n_i m}$ was synchronized according to the Smart Peering rule. m will synchronize its link with n_1 if it can complete the synchronization before leaving the coverage length of n_1. Therefore,

$$s_1(v) = p(v)$$

Consider $s_2(v)$. The link between m and n_2 will only be selected for Smart Peering synchronization if m was not synchronized with n_1 or it had been synchronized but the synchronized link disappeared.

$$s_2(v) = s_1(v)x_1(v) + (1 - s_1(v))p(v)$$

This can be generalized for $i \leq k$:

$$\begin{aligned} s_i(v) &= s_{i-1}(v)x_1(v) + (1 - s_{i-1}(v))s_{i-2}(v)x_2(v) + ... \\ &\quad +... + \left[\prod_{j=1}^{i-1}(1 - s_{i-j}(v))\right] p(v) \end{aligned}$$

Note that the product $(1 - s_1(v))p(v) = 0, \forall v$, and thus the expression of $s_i(v)$ can be simplified as follows:

$$\begin{aligned} s_i(v) &= s_{i-1}(v)x_1(v) + (1 - s_{i-1}(v))s_{i-2}(v)x_2(v) + ... = \\ &= \sum_{j=1}^{k-2}\left[\prod_{l=1}^{j-1}(1 - s_{k-l}(v))\right] s_{k-j}(v)x_j(v) \end{aligned}$$

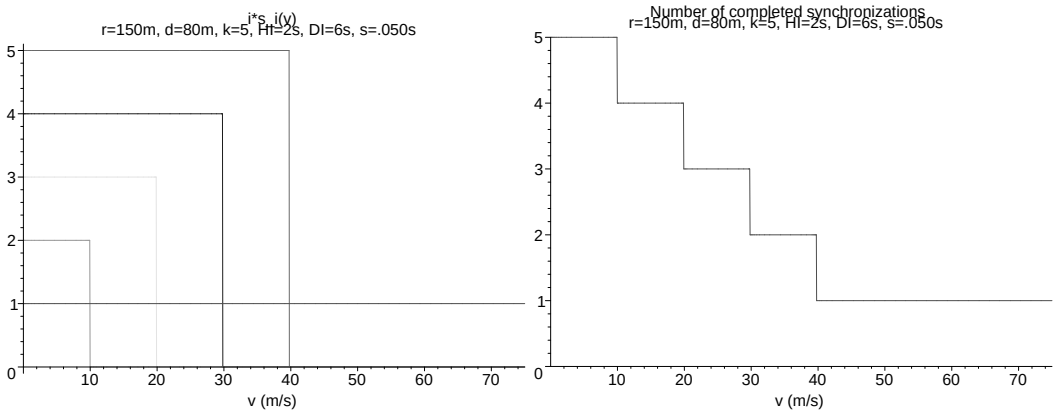


Figure 9.3: **(a)** Scaled functions $s_i(v)$. **(b)** Number of performed synchronizations depending on the speed v.

Then, the function $S(v)$ representing the number of synchronizations completed by the mobile node m,

$$S(v) = \sum_{i=1}^{k} s_i(v)$$

is linearly decreasing with v, as Figure 9.3.b. Figure 9.3 displays the shape of functions $\{s_i(v)\}$ and the number of synchronizations $S(v)$. The fact that the figure is displayed for specific values does not imply any loss of generality for the proposition.

□

Basing overlay decisions on information from the whole network (in particular, related to the current state of the Smart Peering overlay) enables routers to take into consideration dynamic aspects of the Smart Peering candidates, such as their evolution within the overlay or their degree of integration with it. In the case of the most basic version of the Smart Peering rule (see flowchart of Figure 9.1), the technique is able to discriminate the relative speed of the two endpoints of the corresponding links, excluding those with higher relative speed. By discarding fast-moving neighbors, Smart Peering routers are able to to minimize the overlay incorporation of short-lived links, which naturally tends to improve the overall stability of the Smart Peering overlay.

9.5 Conclusion

The Smart Peering technique enables routers to select links to neighbors for the overlay, relying on the relationship that such neighbors maintain with the current overlay. In its most basic version (the one explored in this chapter), the link to a neighbor is discarded for synchronization if that neighbor is already reachable through an already existing synchronized path. Despite its simplicity, the analysis of this technique indicates the benefits that may be achieved by relying on global scope information, as the current state of the Smart Peering overlay. Such feature enables SP to perceive dynamic properties of candidate links and thus discriminate them, for instance, in terms of link stability.

Smart Peering produces a distributed synchronized overlay, and was designed for operation in mobile ad hoc networks. In synchronized overlays, link stability and minimization of link selection events are required properties, as each link that joins the overlay triggers a database exchange process

between the two involved routers. Such a database exchange is expensive in general, as chapter 6 pointed out. The fact that a router takes decisions about links synchronization depending on the stability of such links is therefore one of the most important advantages of Smart Peering as a technique for producing and maintaining a synchronized overlay. Moreover, the SP overlay satisfies the asymptotic properties of connection and dominance over the network (defs. 6.2 and 6.3).

Other aspects, however, discourage the use of this technique for other link-state routing operations: the fact that overlay decisions do not take into account the network topology implies that there is no guarantee that the Smart Peering overlay includes network-wide shortest paths.

Part III

APPLICATION TO OSPF

Chapter 10

LS Routing Protocols within an AS

Different techniques have been presented in Part II for optimizing the performance of link-state routing operations in ad hoc networks. In this Part of the manuscript, these link-state optimization mechanisms for MANETs and overlay techniques are applied to OSPF. The performance of OSPF extensions based on the different presented overlay techniques is evaluated in chapter 12, and the behavior of extended OSPF in compound internetworks is examined in chapter 13.

OSPF and IS-IS are two of the most prominent Interior Gateway Protocols (IGPs) in use in the Internet [98]. Both protocols are based on proactive link-state mechanisms that rely on Dijkstra's algorithm [135] for computing network-wide optimal paths.

Figure 10.1 displays the evolution of the number of ASes in the Internet, by monitoring the amount of AS numbers (ASNs) assigned by regional Internet registries (RIRs) to AS owners (Internet Service Providers and end users)[1]. In 1998, when the number of assigned ASNs in the Internet was around 8000 (see Figure 10.1), the number of routing domains using OSPF was estimated in 4300, according to RFC 2329 [106]. This implies that OSPF was used as IGP in about half of the existing ASes in 1998 – and this trend has not changed substantially since then.

Both protocols, OSPF and IS-IS, are based on proactive link-state mechanisms that rely

[1]Image available at `http://www.potaroo.net/tools/asn32`, website of Geoff Huston, accessed on May 31th, 2011.

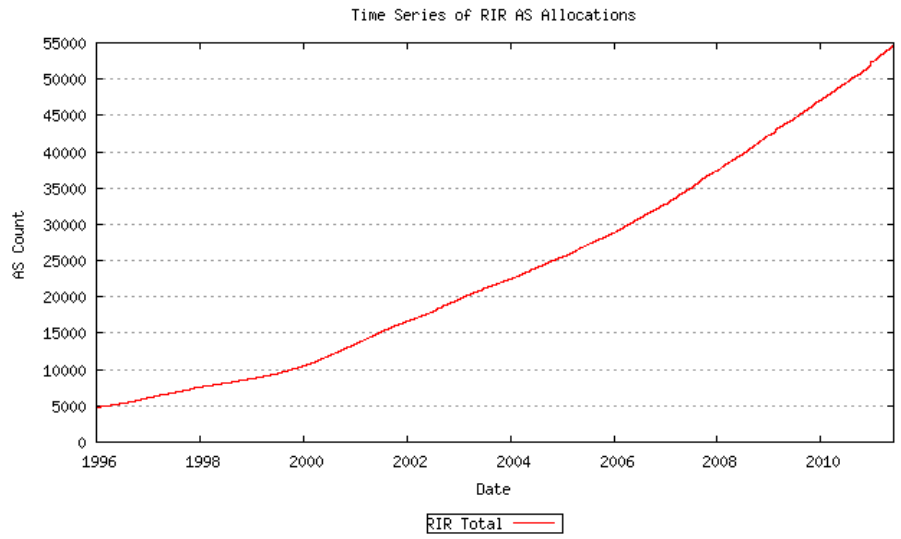


Figure 10.1: Amount of Autonomous System Numbers (ASNs) assigned by Regional Internet Registers (RIRs) to Internet Service Providers (ISPs) and end users.

on Dijkstra's algorithm [135] for computing network-wide optimal paths. The basic features of these protocols are described in this chapter. The remaining of this Part of the manuscript focuses on OSPF.

10.1 Outline

This chapter focuses mainly on the description of the main features, architecture and performance of OSPF (section 10.2). A short overview of IS-IS is provided for completeness in section 10.3, in order to show similarities and differences of this protocol with OSPF. Section 10.4 concludes the chapter.

This Part of the manuscript explores the use of a single routing protocol in compound Autonomous Systems. In consequence, this chapter uses the term *Autonomous System* to refer to the internetwork in which the same instance of OSPF or IS-IS is employed for routing. It assumes that a single instance of the routing protocol is used in all routers of the AS, although multiple instances of the same protocol, and of OSPF in particular, can run within an Autonomous System. This assumption is consistent with definition 1.15 of AS and OSPF terminology [107] – and corresponds

to the case of an OSI *Administrative Domain* that contains a single *Routing Domain*, as these terms are defined in ISO/IEC TR 9575 [115] and used in IS-IS [85].

10.2 Open Shortest Path First – OSPF

The Open Shortest Path First protocol (OSPF) [28, 107] is a link-state routing protocol for IP networks[2]. The first specification of OSPF was released in 1989 by the IETF, and the protocol was designed for replacing RIP[3] as a standard interior gateway protocol. RIP is a distance-vector routing protocol and presents significant disadvantages with respect to link-state protocols, in terms of scalability and convergence, as discussed in section 1.3. These disadvantages motivated the design of a new link-state protocol able to support, in particular, bigger networks – this protocol was OSPF [70]. The first release of OSPF was followed by OSPFv2 [107], and OSPFv3 [28], adapted to IPv6.

10.2.1 Architecture and Terminology

Routers in OSPF maintain a local instance of the *Link State Database* (LSDB), which contains information about the AS topology. Topology descriptions are distributed over the AS in order to ensure that such local instances of LSDB of different routers contain the same information, and thus paths maintained by different routers are consistent to each other. Paths to all possible destinations are derived from the *Shortest Path Tree* (SPT) that every router computes, by way of Dijkstra's algorithm [135].

Routers acquire local topology information and announce their own presence and their list of neighbors by exchanging *Hello* packets with all their 1-hop neighbors (neighbor sensing). With such signaling, each router discovers its immediate topology, *i.e.* its 2-hop neighborhood. This also allows verification of bidirectional connectivity with 1-hop neighbors (then called *bidirectional* or *two-way* neighbors).

---

[2]That is, OSPF runs on top on the network layer, meaning that OSPF packets are encapsulated by IP.

[3]Routing Information Protocol, specified in RFC 1058 [124], and updated in RFC 2453 [102] (RIPv2) and RFC 2080 [110] (RIPng, adapted to IPv6).

Routers also advertise and acquire topology information by exchanging *Link State Advertisements* (LSA). Each router generates LSAs that contain topology descriptions of parts of the AS. These LSAs are disseminated over the AS in a reliable manner (that is, requiring explicit acknowledgments, called *Link State Acknowledgments*, and retransmitting the LSA if no acknowledgement is received) – this operation is called *LSA flooding*. LSAs received from other routers in the AS enable a router to update its own instance of the LSDB.

In order to ensure that topology information is acquired by every interface in the AS, each router performs an explicit pairwise synchronization of a subset of its bidirectional links – that is, each router synchronizes its local instance of LSDB with LSDB local instances of a subset of its bidirectional neighbors. When the local instance of LSDB of a router contains the most recent Link State Advertisements (LSAs) generated in the AS, such router is able to compute shortest paths towards any possible destination in the AS. Synchronization between two neighbors implies that such neighbors share their LSDB (by exchanging database summaries, denominated *Database Description* packets or DBDs), request from the other and install the most recent LSAs of each database.

OSPF introduces the term *adjacency* to denominate a synchronized link. For OSPF, a synchronized link is a link in which (i) local instances of LSDB of both endpoints have been exchanged and updated, and (ii) changes in the local instance of LSDB of any of the routers lead to changes in the other.

Adjacency An *adjacency* is a synchronized link.

Adjacent neighbor The neighbor of a router's interface is *adjacent* if the link between the interface and such neighbor is an adjacency.

The set of adjacencies is required to form a AS-wide connected synchronized overlay that connect all routers in the AS. The use of this synchronized overlay is two-fold: first, Link State Advertisements are flooded through adjacent links; and second, the list of links advertised in a LSA include at least the adjacent links of the generating router. That implies, in particular, that any router that has formed adjacencies must advertise this periodically by way of generating an LSA and performing LSA flooding.

Topology information acquired via LSA flooding or LSDB synchronization is then used for the construction of the Shortest Path Tree, *i.e.*, the set of optimal routes to every possible destination in the AS: each router computes the shortest paths over the set of LSAs in its local instance of LSDB.

OSPF thus classifies links into three categories:

a) links belonging to shortest paths,

b) adjacent links and

c) bidirectional links.

Each of these is a subcategory of the lower categories, as Figure 10.2 illustrates. A subset of bidirectional links in the AS becomes adjacent (synchronized). Among these adjacent links, a new subset is selected to be part of the Shortest Path Tree. While data traffic is routed on shortest paths belonging to the SPT, control traffic is sent over adjacent links.

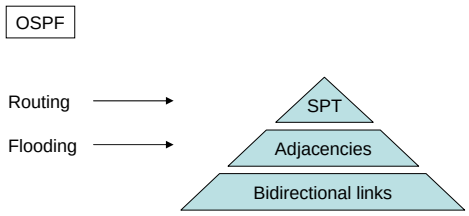


Figure 10.2: Link hierarchy in OSPF.

The use of synchronized links for routing data packets ensures that forwarding decisions along a routing path are consistent, as they are based in synchronized instances of the LSDB. Routing packets through unsynchronized links may lead to routing loops if intermediate routers along the routing path maintain different topology information. Restricting flooding to synchronized (adjacent) links allows to concentrate in such adjacencies the impact of control traffic, without affecting the other links in the AS.

10.2.2 Areas, Interfaces and Neighbors

Information exchange in OSPF involves three entities: the network interface through which a router communicates with other routers, the neighbors that are reachable through an interface, and the OSPF area in which an interface is located. The behavior of each network interface depends on the properties of the network in which the interface participates: the specification of OSPF provides support to five different interface types. The relationship between an interface and one of its neighbors depends on the state of the link (def. 1.5) to such neighbor – *i.e.*, if it is symmetric, asymmetric or synchronized. Finally, the notion of area permits OSPF to provide an efficient logical topology to ASes that contain a large number of networks. Figure 10.3 illustrates the notions of interfaces, neighbors and areas for an OSPF router.

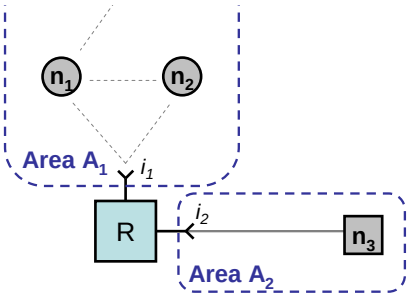


Figure 10.3: Areas, interfaces and neighbors of an OSPF router. *Router R has two interfaces, i_1 and i_2. i_1 belongs to area A_1 and has two direct neighbors n_1 and n_2, while i_2 belongs to area A_2 and has a single neighbor n_3.*

Areas

The pure link-state routing mechanism described in section 1.3 and in chapter 4 does not scale for ASes involving a large number of routers and links, as the overhead required for topology flooding has quadratic growth $\sim O((nm)^2)$ with respect to the product of routers n and links per router m (see section 6.3.1). In particular, the requirements that local instances of the LSDB are identical for all routers in the AS, and this LSDB contains a complete view of the AS topology, generates a control overhead for LSA flooding and LSDB synchronization that may become excessive

when the AS grows or (part of) the links present a non-negligible change rate, as it was shown in section 6.3.

In order to address this issue, OSPF allows splitting an AS into logical routing *areas*.

Definition 10.1 (**OSPF Area**)**.** In OSPF, an *area* is a group of networks in which the interfaces have the same LSDB and thus share the same topology information in their local instances of LSDB.

The fact that routers in the same area have the same topology information implies that information maintained by any of such routers is sufficient for performing intra-area routing (routing from nodes in the area towards nodes in the area).

In multi-area OSPF deployments, communication between OSPF areas is performed by way of the *OSPF backbone*.

Definition 10.2 (**OSPF Backbone**)**.** In OSPF, the *backbone* is an OSPF area, also denominated *Area Zero*, to which any other area needs to be connected by way of one or more routers.

The OSPF backbone has two defining characteristics:

(i) There is only one backbone in the Autonomous System and it is connected, by way of physical or virtual links.

(ii) Each router with interfaces in more than a single area – that is, that provides connectivity between different areas – participates in the backbone.

Depending on the areas in which router interfaces participate, routers in OSPF are classified as follows:

IR *Internal Routers* have all interfaces connected to a single area.

ABR *Area Border Routers* have at least one interface connected to the backbone and at least one interface connected to another area.

ASBR *Autonomous System Boundary Routers* have at least one interface connected to an OSPF area and at least one interface connected to a network outside the Autonomous System.

BR *Backbone Routers* are those that have at least one interface connected to the backbone. BRs may be internal routers, if all interfaces are connected to the backbone, or ABRs, otherwise.

ABRs are the only routers connecting different areas and maintain a local instance of the LSDB of each area in which have an interface. These routers are thus necessary for performing inter-area routing, *i.e.*, routing of packets for which the destination area is different from the area of the source. Packets sent to a destination not in the same area of the source traverse therefore a maximum of three areas: the area of the source, the area of the destination and the backbone to which both areas are connected via ABRs.

A router computes the tree of shortest paths over the network graph described in a local instance of the LSDB that contains information about the area topology. Therefore, inter-area routing may use suboptimal paths, as they result from the juxtaposition of paths that are optimal in each area, but may not be optimal when considered together.

Such a partition requires a 2-level hierarchy of routers (the bottom level including IRs connected to areas other than the backbone, and the top level containing all BRs). Router hierarchy enables OSPF to restrict most of the impact of a topology change (in terms of control traffic overhead to update local instances of LSDBs) to the area in which that change occurred. Control traffic in the backbone may also be affected, but no effect should be perceived in areas not directly connected to the area whose topology changed.

Interfaces

Rules for flooding and adjacency handling vary for the different *interface types* supported by OSPF. Three main interface types are specified in [107]:

- *Point-to-point interfaces* participate in point-to-point links (def. 1.7). Such a link only permits communicating with a single (neighboring) interface. Point-to-point interfaces are used for connecting to PPP or HDLC[4] links.

[4]High-Level Data Link Control protocol.

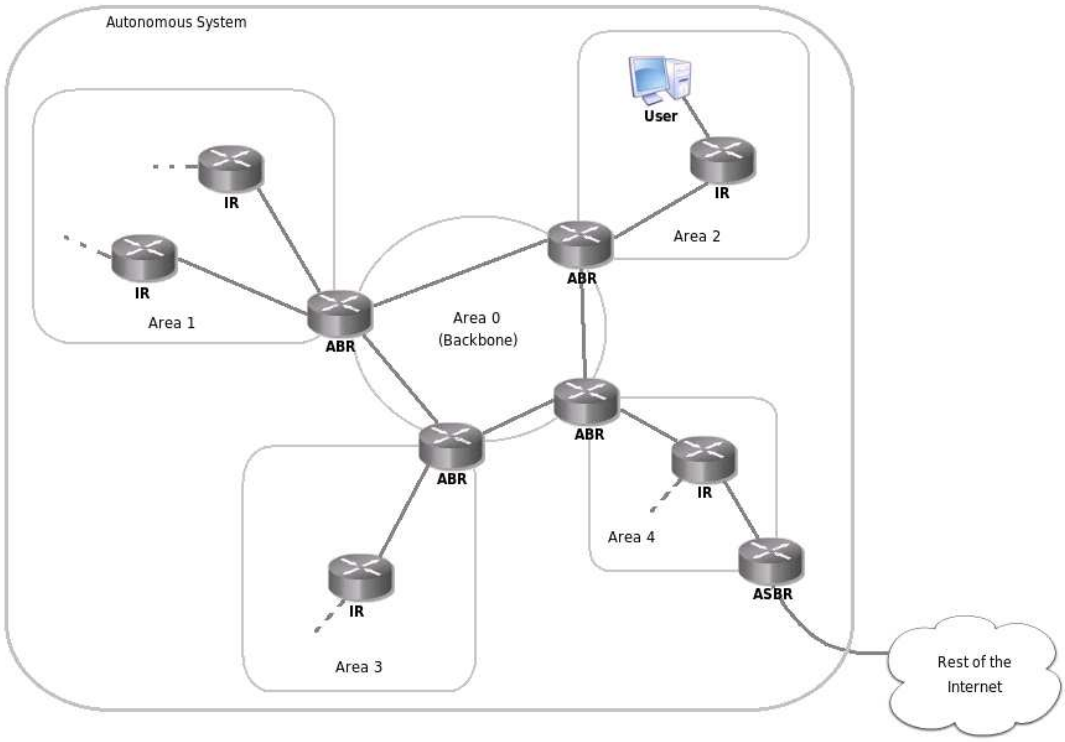


Figure 10.4: Area partition of an Autonomous System under OSPF.

- *Broadcast interfaces* participate in a broadcast link (def. 1.8). Classic example of broadcast link is Ethernet.

- *Virtual link interfaces* can emulate direct communication between two Backbone Routers (BRs) that are not physically connected but have interfaces connected to a common area. These links are used to ensure the connection of every area to the backbone or to guarantee the connection of the backbone itself [51]. This last case is illustrated in Figure 10.5.

Links of non-broadcast networks are not supported explicitly by any of these types. In these networks, an interface may be able to directly communicate with several other interfaces, and therefore the interface does not participate in a point-to-point link (def. 1.7). Since it cannot be ensured that a single transmission is received by all the interfaces to which direct communication is possible, such link cannot be classified as a broadcast link neither (def. 1.8). For handling links in

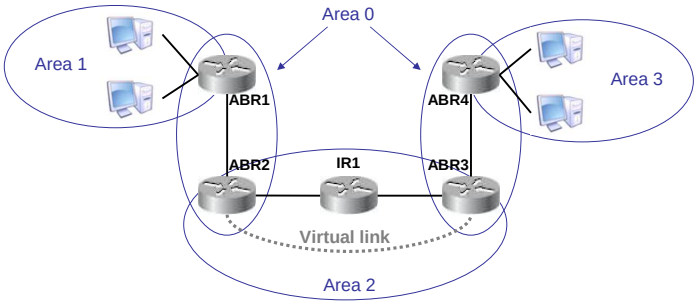


Figure 10.5: Virtual link between ABR_2 and ABR_3 through Area 2 provides connection of Area 0 (*backbone*).

such non-broadcast networks, OSPF provides two additional interface types:

- *Non-Broadcast Multiple Access (NBMA) interface*, for non-broadcast networks in which each pair of interfaces can communicate directly (*i.e.*, by way of a link as defined in def. 1.5). Typical examples of these type of networks are ATM with Switched Virtual Circuits (SVC) or X.25.

- *Point-to-multipoint interface*, for those non-broadcast networks in which direct communication between any pair of interface cannot be ensured. Examples of this type of non-broadcast networks are Frame Relay networks that only support Permanent Virtual Circuits (PVC), for which not every pair of interfaces has a PVC between them.

In the case of NBMA interfaces, OSPF emulates the behavior of a *broadcast* link, *e.g.* by replicating transmission of Hello and LSA packets to all neighbors via unicast. In case of a point-to-multipoint interface, the non-broadcast network is handled as a set of *point-to-point* links, one per neighbor. Hello exchange and LSA flooding are performed in NBMA and point-to-multipoint interfaces as it is performed for broadcast and (a collection of) point-to-point interfaces, respectively.

Figure 10.6 displays the Finite States Machine (FSM) of network interfaces in OSPF. When an interface is switched on, it checks the type of link in which it is expected to participate. The information contained in the first Hello packet received over the interface enables detecting whether the link corresponds to the point-to-point or point-to-multipoint type (decision (1) in Figure 10.6). If the link is either point-to-point or point-to-multipoint, the interface is configured *point-to-point*.

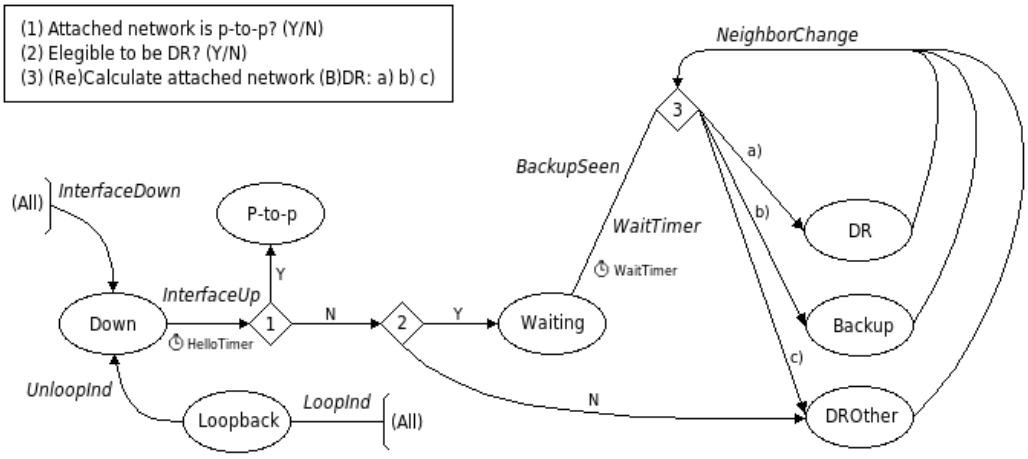


Figure 10.6: Finite States Machine (FSM) for network interfaces in OSPF.

Otherwise, the interface type is set to broadcast/NBMA type, and then stays in state *Waiting* until the interface state is selected among states DR, BDR and $BDROther$ according to a procedure described in section 10.2.4.

The FSM presented in Figure 10.6 permits only the automatic acquisition of point-to-point (or point-to-multipoint) and broadcast/NBMA interface types. In case they are required, virtual links need to be configured manually.

Neighbors

An interface keeps track of the state of the neighboring interfaces (neighbors) it can directly communicate with. The state of a neighbor indicates the communication capabilities of the link between the interface and such neighbor. Such capabilities concern two main aspects: (i) bidirectionality of communication and (ii) synchronization of local instances of LSDB.

Figure 10.7 displays the Finite States Machine (FSM) for a neighboring interface. OSPF classifies the state of a neighbor for an interface in eight categories [107]:

Down There is no neighbor or the interface does not receive any packet from this neighbor. Therefore, no packets are sent towards this neighbor.

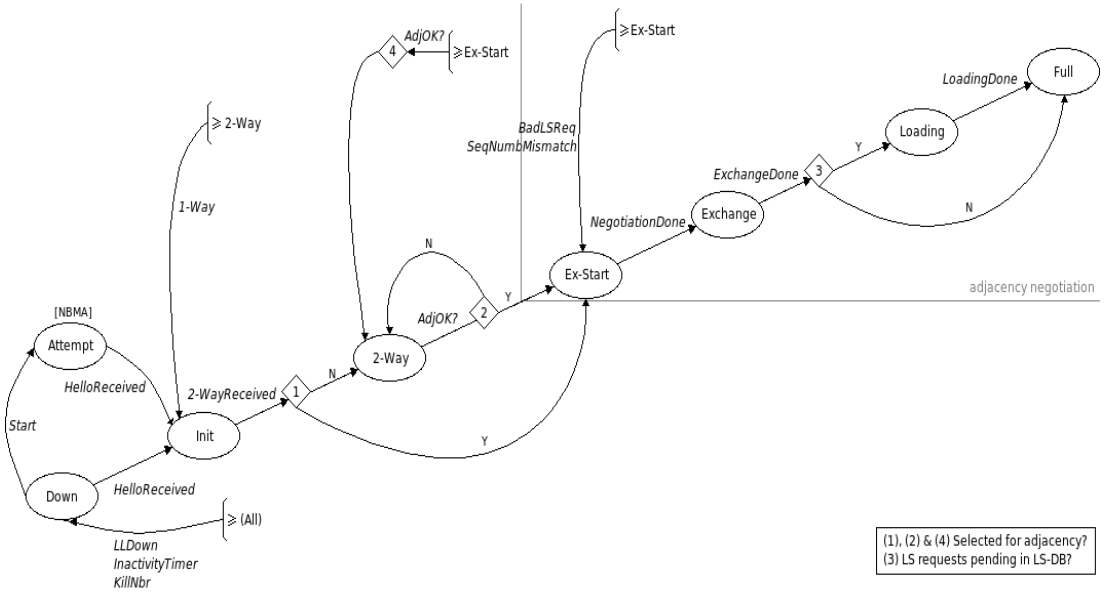


Figure 10.7: Finite States Machine for neighbors in OSPF.

Attempt (only for NBMA networks) The interface does not receive any packet from this neighbor, but still tries to establish communication with it.

Init Unidirectional communication (from the neighbor towards the interface) is available.

Two-Way Bidirectional communication (from the interface towards the neighbor and vice versa) is available.

Ex-Start The neighbor has been selected for LSDB synchronization. While in this state, the interface negotiates with the neighbor their respective role in the adjacency-forming process.

ExChange The interface and the neighbor are exchanging their respective local instances of LSDB, by way of sending and transmitting Database Description ($DBDesc$) packets.

Loading The interface has requested the neighbor for LSAs that are more recent in the neighbor's local instance of LSDB than in its own, and is waiting for the neighbor's reply.

Full The neighbor is fully adjacent to the interface.

Init and *Two-Way* states depend on Hello exchange. When a Hello packet from a neighbor is received over a router's interface, and such packet does not advertise the receiving router, the neighbor state is set to *Init*. If the receiving router is listed among the neighbors in the received Hello packet, the neighbor state is upgraded to *Two-Way*, if it was lower, or kept in a higher state, otherwise.

States between *Two-Way* and *Full* (both excluded) correspond to intermediate stages of the LSDB synchronization process that includes (i) negotiation of the role assumed by each neighbor in the process (*Ex-Start* state), (ii) exchange of summaries of the respective LSDBs (*ExChange* state) and, (iii) eventually, request and transmission of missing LSA updates (*Loading* state). The decision of starting an adjacency-forming process (transition from *Two-Way* state to *Ex-Start* state) is taken following an procedure which is specific of the interface type. Section 10.2.4 describes this procedure in detail for the case of broadcast/NBMA interfaces, for other interfaces and a more exhaustive explanation of the neighbor state machine, see [107].

10.2.3 Packet and Message Types

Section 10.2.1 mentioned the main types of packets and messages that are used in OSPF operation: *Hello* packets for neighbor sensing, *Database Description* packets for LSDB synchronization and *Link State Advertisements* (LSAs) for topology reliable flooding and update. Several LSAs may be sent in a single *Link State Update* packet (LSU). Several LSA acknowledgements may also be grouped in a single *Link State Acknowledgment* (LSAck) packet. The topology of an OSPF multi-area AS is described via different types of LSAs.

Different types of routers are responsible for originating and flooding different types of LSAs over their *flooding scope*.

Definition 10.3 (**Flooding Scope**)**.** The *flooding scope* of a LSA is the set of routers that are expected to receive and acknowledge the LSA in the AS.

Table 10.1 lists the flooding scope, the originating routers and the contents for each LSA type, as well as the LSA denominations in the two main specifications of OSPF (OSPFv2 [107] and

OSPFv3 [28]).

Denomination[5]	Scope	Originator(s)	Contains
Router-LSA	Area	Every router	State of the interfaces of the originating router attached to the area.
Network-LSA	Area	DRs *(bc/NBMA)*	Interfaces participating in a broadcast or NBMA link.
Inter-Area-Prefix-LSA *Summary-LSA (type 3)*	Area	ABRs	Prefix out of the area, inside the AS.
Inter-Area-Router-LSA *Summary-LSA (type 4)*	Area	ABRs	Routes to ASBRs.
AS-external-LSA	AS	ASBRs	Routes to destinations outside the AS, and default route of the AS.
Link-LSA*	Link	Every router	Link-local address of the orig. router.
Intra-Area-Prefix-LSA*	Area	Every router	Map between router id and IPv6 prefixes

Table 10.1: LSA formats in OSPF.

Each router in the Autonomous System originates a Router-LSA, which describes the links maintained by all interfaces of the router. For broadcast and NBMA links, the *Designated Router* (see section 10.2.4) originates a Network-LSA, describing the interfaces that participate in such network.

ABRs are responsible for distributing routing information between those areas to which they have interfaces. More precisely, ABRs originate two types of LSAs:

- By way of Inter-Area-Prefix-LSAs, ABRs inject prefixes attached to one area into another. A single Inter-Area-Prefix-LSA is flooded per each prefix to be advertised.

- By way of Inter-Area-Router-LSAs, an ABR describes a route towards an ASBR. An ABR sends a single Inter-Area-Router-LSA per advertised ASBR and per attached area not containing the advertised ASBR.

AS Boundary Routers originate and flood AS-external-LSAs to advertise destinations external to the AS (*e.g.*, a default route) over routers within the Autonomous System.

[5]In *italic*, the denomination for OSPFv2, where different from the used in OSPFv3. An asterisk (*) indicates the LSA formats specific of OSPFv3, not existing in OSPFv2.

Two additional LSA types were introduced in OSPFv3: Link-LSAs and Intra-Area-Prefix-LSAs, as shown in table 10.1. Link-LSAs permit every interface to advertise its link-local address over the link in which it participates. Such link-local addresses are used in Hello packet exchange. Intra-Area-Prefix-LSAs are originated by every router in the AS and advertise the IPv6 prefixes used by the originating router. This way, addressing semantics are not necessary in the payload of other OSPFv3 packets – instead, routers are identified with a 32 bit router id, and other routers with interfaces in the link are able to match the interface link-local address with the id of its router.

10.2.4 Single-Area OSPF for Non-Broadcast Networks

The architecture of OSPF, as it is specified in RFCs 2328 [107] and 5340 [28], is not adapted for operation in wireless ad hoc networks. The hierarchical routing model based on the existence of a backbone to which several routing areas are connected, in particular, cannot be directly deployed when network topology is unknown and may changed dynamically [73]. Some hierarchical approaches for link-state routing (in particular, for OLSR [53, 30]) have been proposed for large ad hoc networks; these approaches explore the use and management of clustering techniques and link-state routing areas in mobile ad hoc networks. This manuscript, however, concentrates on OSPF routing solutions based on a single area.

None of the interface types described in OSPF specification captures the characteristics of wireless ad hoc networks. However, such networks are a particular case of non-broadcast networks, as they were described in section 10.2.2. Two interface types are defined in OSPF for non-broadcast networks: NBMA and point-to-multipoint interfaces. This section shortly describes the operation of these two types of interfaces in a non-broadcast network organized in a single OSPF area, and discusses the applicability of these interface types for wireless ad hoc networks.

Non-Broadcast Multiple Access (NBMA)

Non-broadcast multiple access networks are non-broadcast networks for which any pair of routers can communicate directly – *i.e.*, using a link between two interfaces of such routers [107].

In OSPF, the main responsible for topology flooding in NBMA networks is the *Designated Router* (DR). Such router is elected in a distributed manner among all interfaces attached to the network. Each interface selects its Designated Router based on the information received via Hello packets from other interfaces, and the procedure converges when all interfaces select the same DR. Figure 10.8 displays the algorithm executed by an interface after switching on.

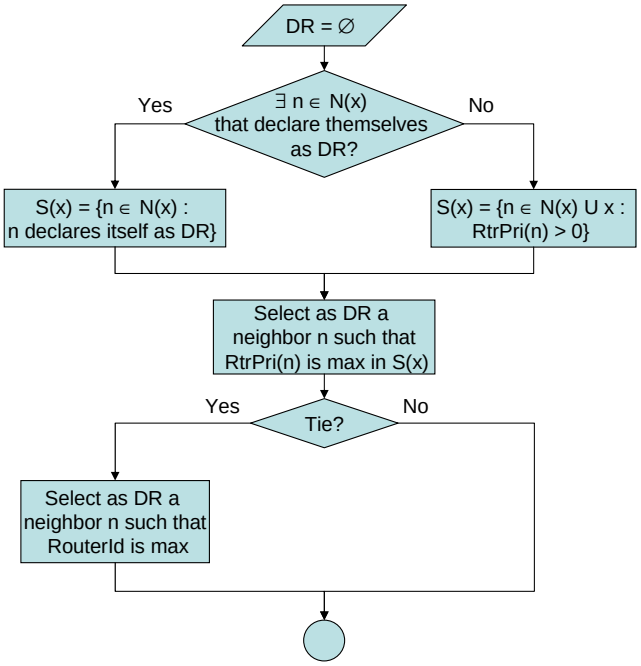


Figure 10.8: Procedure of election of Designated Router (DR) for a broadcast or NBMA interface.

The DR election procedure takes into consideration the willingness (*RouterPriority*, or *RtrPri*) of each router to become DR, as well as the DR that each router has selected. Both elements are advertised in Hello packets. When a router switches on an interface in a broadcast/NBMA network, the interface selects as Designated Router the existing one, in case it has been already elected. If several interfaces declare themselves as DRs (*i.e.*, if no DR has been still agreed in the network), the interface selects as DR the self-declared neighbor with highest *RtrPri*. If no neighbor has declared itself as a DR, the interface selects as DR the router with highest *RtrPri* among the set composed of the neighbors and the computing router itself. In case of tie between several routers,

the router with highest router id is selected as DR.

This procedure ensures that the appearance of interfaces in the network after the election of a DR does not cause a new election procedure, as new interfaces assume the existing DR when such DR has been already elected. A DR is not changed as long as it does not disappear from the network. This reduces the number of changes in the identity of the Designated Router. The fact that all interfaces receive Hellos from each other implies that the DR election procedure is consistent in all interfaces.

Every newly elected DR requires a significant amount of control traffic, mostly related to LSDB synchronization processes, and a time lapse before it is fully operational; therefore, minimization of the number of DR changes is desired. The Designated Router has two main responsibilities:

- The DR originates and floods over the area a Network-LSA that describes the set of interfaces taking part in the NBMA network.
- The DR also receives Router-LSAs originated by any other router in the network and floods them over the area.

This is performed by forming adjacencies between the DR and any other neighbor in the network – no more adjacencies are needed. Network-LSAs thus contain a list of the neighbors adjacent to the DR, and Router-LSAs from such adjacent neighbors are flooded over the network. When the DR is adjacent to all its neighbors, the NBMA network emulates the operation of OSPF in broadcast networks, in which LSAs from a neighbor are received in every other neighbor, by way of DR operation.

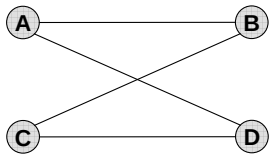
In order to maintain network operation when the DR is replaced (*e.g.*, due to the failure of the router selected as DR), a Backup Designated Router (BDR) is also elected. Routers also synchronize their links with the BDR. The procedure of election of BDRs is similar to the procedure for DRs, and is described in [107]. A stochastic analysis of the cost of DR and BDR election procedures (in broadcast networks) can be found in [50]. The synchronized overlay in NBMAs is thus formed by adjacencies between routers and the selected DR and BDR.

The election of a Designated Router and the optimization that such DR enables in terms of flooding and adjacencies, rely on the assumption that each pair of interfaces are able to directly communicate. Since this cannot be ensured at any time in wireless ad hoc networks with dynamic topology, the NBMA interface type should not be used for wireless interfaces in such networks.

Moreover, the use of the NBMA interface type in a wireless ad hoc network implies that routers are unable to distinguish among links to different wireless neighbors. The Router-LSA of a router describes the link to which a NBMA interface is attached as a single NBMA link, with a single Designated Router, when elected, and a single metric value. This implicitly assumes that the quality of links towards all wireless neighbors is the same – which prevents the use of link metrics other than hop count.

Point-to-Multipoint

When direct communication is not available between every pair of interfaces in a non-broadcast network, the Designated Router cannot be unambiguously elected. Figure 10.9 illustrates an example in which the procedure described for DR election in NBMA networks (Figure 10.8) causes that (some) interfaces do not agree in the DR.



	RtrId	*RtrPri*
A	1	1
B	3	1
C	2	1
D	4	1

Figure 10.9: Example of non-broadcast network with no direct communication between every pair of interfaces.

In the example of Figure 10.9, routers B and D select themselves as DRs, and A and C select as DR the self-declared neighbor with higher $RtrId$, *i.e.* D. As C does not receive Hellos from D, it has no self-declared neighbors and keeps selecting the router with highest $RtrId$ among A, B and C – thus selecting itself as DR.

Interfaces attached to a non-broadcast network in which direct communication cannot be ensured for every pair of interfaces should not operate as NBMA interfaces. Instead, their type

can be set to *point-to-multipoint*. Interfaces of this type communicate with each of their neighbors as if there was a point-to-point link between the interface and the neighbor. For each neighbor, a point-to-multipoint interfaces behaves in OSPF as a point-to-point interface. No Network-LSA is then originated on behalf of the network, all the links are then synchronized (as they are for point-to-point interfaces) and thus the flooding of Router-LSAs and other LSAs within the area is performed over all such links. As links from a router towards each of their neighbors, and their costs, are described separately in Router-LSAs, link metrics other than hop count can be used.

There are no theoretical restrictions for the use of OSPF point-to-multipoint interfaces in wireless ad hoc networks. Due to the fact that all links become adjacent, the amount of overhead produced by such interfaces in MANETs becomes excessive as the number of routers increases: experimental results show that OSPF operation in mobile ad hoc networks with point-to-multipoint interfaces does not scale over 20 routers [72].

10.3 Intermediate System to Intermediate System – IS-IS

IS-IS[6] is a proactive link-state protocol for routing within Autonomous Systems (def. 1.16). Its core mechanisms are thus very similar to those used in OSPF, and both protocols rely on the Djikstra's Shortest Path algorithm.

IS-IS is an OSI routing protocol. It uses the OSI addressing model and operates independently from the network protocol. That implies, in particular, that IS-IS packets are directly processed at layer 2. A version of IS-IS, denominated *Integrated IS-IS*, was specified in RFC 1195 [121] for operation in IP networks – *i.e.*, for running over IP at layer 3.

10.3.1 Architecture and Network Partitioning

IS-IS supports network partitioning into several areas. Figure 10.10 displays an example of an IS-IS network partitioned into 3 areas.

[6]Specified by ISO in 1992 (ISO standard ISO/IEC 10589:1992, and revised by ISO/IEC 10589:2002), also available in RFC 1142 [122].

Definition 10.4 (**IS-IS Area**)**.** In IS-IS, an *area* is a set of routers that "maintains detailed routeing information about its own internal composition, and also maintains routeing information which allows it to reach other routeing subdomains" [122]. Every router in an IS-IS network is attached to one (and only one) area.

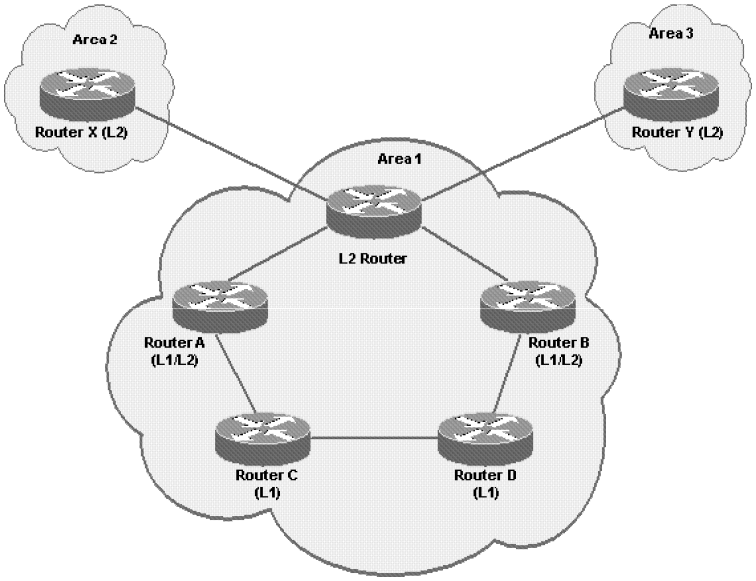


Figure 10.10: Area partition of an Autonomous System under IS-IS.

Two levels of routing are distinguished in a multi-area IS-IS network. Routing within a single area is denominated *Level 1* routing. Routing between areas is denominated *Level 2* routing, and is performed by way of the *IS-IS backbone*.

Definition 10.5 (**IS-IS backbone**)**.** In an IS-IS network, the *backbone* is the set of routers able to directly communicate with routers attached to areas other than their own.

The fact that IS-IS routers are located in only one area implies that the IS-IS backbone is not an area. Instead of forming an area by themselves, backbone routers belong to the different existing areas (at least one backbone router per area) and provide connection between such areas. These routers are responsible for L2 routing, while L1 routing can be performed by routers only able to communicate with neighbors in its area. Such L1 routing only requires that routers have

topology information from their area. Depending on their routing capabilities, routers maintain a local instance of the LSDB of their own area and a local instance of the LSDB of the IS-IS backbone. A router that only has information about its own area can only forward packets addressed to other ares towards the nearest router that maintains a local instance of the backbone LSDB.

In this section, when a router can only perform L1 routing, it is denominated *L1 router*, and when it is only able to perform L2 routing, it is denominated *L2 router*. Routers may be able to perform L1 and L2 routing – they are then denominated *L1L2 routers*. For convenience, the terms *L1* routers* and *L2* routers* are used to denote routers able to perform L1 routing and L2 routing, respectively – regardless on their capability to perform routing in the other level. With these terms, the three types of routers in IS-IS (L1, L2 and L1L2) are described as follows:

- A L1 router maintains a local instance of the its area LSDB, with topology information describing the area in which they are located. They only become neighbors with other L1* routers in the same area.

- L1L2 routers maintain local instances of two LSDBs, one corresponding to the topology of the area they belong to, and the other containing the backbone links. L1L2 routers can become neighbors with L1* routers in their area and L2* routers in other areas.

- L2 routers are located in the backbone, and they maintain the topology of the network subgraph connecting L2 routers. They can become neighbors with L2* routers in other areas.

There are different Hello formats for L1 and L2 routing levels. This implies that L1L2 routers need to exchange Hellos of both formats in order to discover and maintain L1* and L2* neighbors. There are also level-specific formats for topology flooding and LSDB synchronization packets. By performing separately link-state operations in each routing level (L1 and L2), IS-IS enables a network partition both in terms of areas and routing levels, consisting of:

- The backbone, formed by L2* routers of all areas.

- For each area, the set of L1* routers in the area.

Figure 10.11 identifies these elements in the network example of Figure 10.10. Only L1L2 routers have a complete information about topology of *Area 1*, as local instances of LSDB stored by L1 routers and L2 routers contain only partial information about *Area 1* topology.

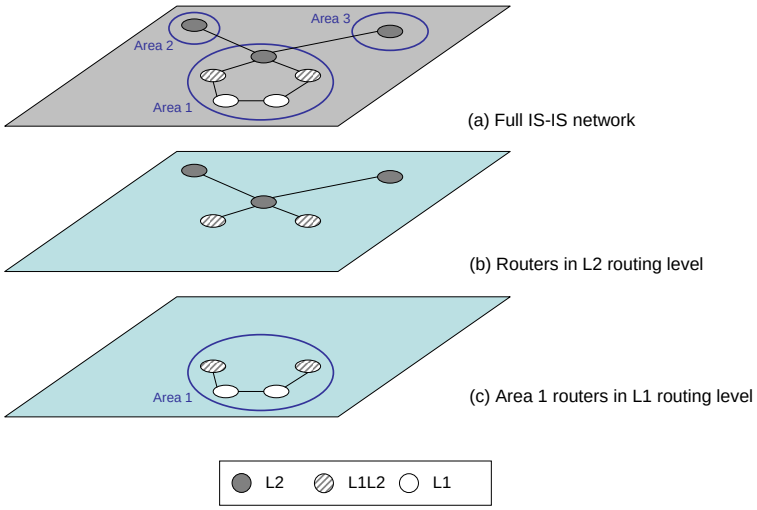


Figure 10.11: Routing level and area partition of the IS-IS network example of Figure 10.10.

10.3.2 Interface Types

IS-IS provides support for two interface types: *point-to-point* interfaces, for point-to-point links (def. 1.7), and *broadcast* interfaces, for broadcast links (def. 1.8). In IS-IS, links other than broadcast or point-to-point are treated as one of these two. In particular, interfaces attached to non-broadcast links should be split into several point-to-point subinterfaces, one per reachable neighbor [34, 84].

A broadcast link is represented in IS-IS by way of a virtual entity denominated *Pseudo-Node* (PSN). A *Designated Intermediate System* (DIS) is elected among the routers of the broadcast network, creates the PSN and acts on behalf of it. As DRs in OSPF, the election of the DIS is based on the exchange of Hello packets, and takes into account the willingness of each router to become DIS. Unlike Designated Routers in OSPF, the DIS in IS-IS may change if a router with higher willingness than the current DIS joins a broadcast network. Each DIS is responsible for keeping

updated (synchronized) the local instances of LSDB of the rest of routers, by periodically flooding its own topology information. In broadcast IS-IS networks there is no explicit LSDB exchange between the DIS and the other routers as in OSPF: instead, periodic dissemination of the topology by the DIS ensures that all routers' local instances of LSDB stay synchronized to each other's [52].

For point-to-point links, the two involved routers synchronize their local instances of LSDB by alternatively transmitting packets that announce the link-state advertisements (denominated *Link-State Packets* or LSP in IS-IS) contained in each local instance. Both interfaces can then request each other for reliable transmission of particular LSPs. The LSDB synchronization process is similar to OSPF's adjacency-forming process.

Different LSPs are used for L1 and L2 routing levels, meaning that L1* routers originate and collect topology information from L1 LSPs, and similarly for L2* routers. For each routing level, LSPs advertise the list of bidirectional neighbors of the originating router. The set of LSPs received by a router for a particular routing level form the local instance of the corresponding LSDB (L1 LSDB or L2 LSDB). Since routers compute shortest paths over local instances of LSDBs of their supported routing levels, route optimality is only guaranteed for routing in a single area. Multi-area architectures may lead to suboptimal routing, if the juxtaposition of locally optimal routes computed over area (L1) LSDBs and backbone (L2) LSDBs is not an globally optimal route.

10.4 Conclusion

OSPF and IS-IS are the main protocols for link-state routing within an Autonomous System. Both can be used for routing in mobile ad hoc networks or compound ASes, and they use similar mechanisms to perform link-state operations – topology selection, flooding and LSDB synchronization.

This chapter has focused on the description of the main concepts and mechanisms of OSPF. The protocol is used in the following as a base protocol for routing in MANETs (chapters 11 and 12) and compound Autonomous Systems (chapter 13). The chapter also presents the basics of IS-IS, in order to show the similarities and differences with OSPF.

The main concept in the OSPF architecture is the concept of synchronized link, or *adjacency*. Adjacencies are links between interfaces that maintain the same topology information in their local instances of LSDB. The three link-state operations are performed only by way of adjacencies, in order to ensure consistency of flooding and routing decisions from any interface in the network. The way that an interface performs such operations depends in OSPF on the type of link to which the interface is attached – OSPF provides support for four interface types, two of them designed for interfaces attached non-broadcast networks: Non-Broadcast Multiple Access (NBMA) and point-to-multipoint interface types.

The use of these types in interfaces attached to MANETs raises fundamental problems in both cases: the mechanisms used in NBMA interfaces are not appropriate and point-to-multipoint interfaces have scalability constraints. NBMA interfaces get a significant optimization in performance of the link-state operations, mostly by way electing Designated Reouters, but this optimization relies on the assumption that two interfaces can always directly communicate – which is not necessarily the case in a wireless ad hoc network. The point-to-multipoint interface type enables OSPF operation in non-broadcast networks not fulfilling this assumption. OSPF operation in such point-to-multipoint interfaces, however, requires a control traffic amount (for LSDB synchronization and topology flooding) in the order of $O(n^2)$, where n is the number of interfaces in the network, as shown in chapter 6. These amounts of control traffic may become excessive as ad hoc networks become bigger and more dynamic.

Whilst link-state routing and, in particular, OSPF routing can be used in mobile ad hoc networks, the protocol needs to be adapted so as it can perform the three link-state operations efficiently in MANETs. Given that the interface types specified in OSPF are not suitable for MANETs, such adaptation of OSPF is explored in this manuscript by defining an additional interface type for MANET operation. Several approaches to this new OSPF interface type are examined and proposed in chapter 11, and evaluated via simulations in chapter 12.

Chapter 11

OSPF MANET Extensions

The way that topology flooding and LSDB synchronization are performed in OSPF "as-is" is not suitable for MANETs, mostly due to the excessive overhead that such operations imply in dynamic topology networks. The modular architecture of OSPF, however, enables development of new extensions – extensions specifically designed for MANET operation. Development of such extensions makes possible to perform routing in compound ASes, with both ad hoc and wired networks, and where the particularities of each such network are managed by appropriate mechanisms – all within the same routing protocol instance.

11.1 Outline

This chapter focuses on the multiple OSPF extensions for MANET operation in a single area that have been standardized by the IETF, including MPR-OSPF [24], OR / SP [19] and OSPF-MDR [22]. Section 11.2 describes the main properties and behavior of each of these extensions. Two of these three extensions, MPR-OSPF and OR / SP, are based on Multi-Point Relaying (MPR), described in chapter 8. Section 11.3 explores other extensions that improve or combine techniques from these MPR-based extensions. Finally, section 11.4 concludes the chapter.

11.2 IETF Standard Extensions

IETF standardization efforts with respect to the extension of OSPF for MANET operation have led to three different extensions: the Multi-Point Relays extension (MPR-OSPF), the Overlapping Relays & Smart Peering extension (OR / SP), and the MANET Designated Routers' extension (OSPF-MDR). Both MPR-OSPF and OR / SP use the MPR technique presented in chapter 8, OR / SP incorporating also the Smart Peering technique detailed in chapter 9. They are described in sections 11.2.1 and 11.2.2, respectively. OSPF-MDR takes a fundamentally different approach, inspired by the notion of Designated Router used in broadcast and NBMA interfaces of standard OSPF, and is presented for completeness in section 11.2.3.

11.2.1 Multipoint Relays – MPR-OSPF

The MPR extension of OSPF for mobile ad hoc networks performs the three main operations of a link-state routing protocol, topology selection, flooding and link synchronization, by using network overlays based on Multi-Point Relays. Each router selects its multi-point relays (MPRs) from among its bidirectional neighbors, and such MPRs are incorporated into the flooding, link synchronization and topology selection procedures performed by that router.

Flooding and Adjacencies

In any OSPF MANET extension, topology information is disseminated in Router-LSAs over the network, by way of two mechanisms:

- Selective retransmission (reliable flooding over a selected subset of neighbors), and
- LSDB synchronization (adjacency-forming processes and adjacency maintenance).

In MPR-OSPF, selective retransmission follows the MPR principle: a router only forwards Router-LSAs if they have been received from one of the router's MPR selectors. Acknowledgments are only sent as reply to LSA transmissions from an adjacent neighbor.

A router triggers an adjacency-forming process with all its multi-point relays, thus becoming adjacent of:

(i) its MPRs, and

(ii) those neighbors that have selected such router as MPR (MPR selectors).

The set of MPRs of a router includes the neighbors selected for flooding (Flooding MPRs) and the neighbors selected for topology selection (Path MPRs). Flooding MPRs are selected following heuristic (8.2), while Path MPR selection is performed as described in (8.6):

$$MPR(x) = Flooding\_MPR(x) \cup Path\_MPR(x)$$

While both sets (Flooding MPRs and Path MPRs) are identical when a hop count metric is used, as shown in equations (8.8), they may be different for other link metrics. The set of adjacencies corresponding to (i) and (ii) does not guarantee the connection of the synchronized overlay, as it was shown in section 8.4. This means that a synchronized overlay based only on conditions (i) and (ii) may be disconnected in several connected components, as the examples of Figure 8.5 pointed out. In this case, there may be pairs of routers for with no synchronized path (def. 4.2) exists between them, thus implying that shortest paths are not necessarily synchronized.

Lemma 8.3 proved that the addition of links between a single router and all its neighbors to the MPR overlay is a sufficient condition for the connection of such overlay. Such an additional router is denominated *synch* router in MPR-OSPF. With the addition of *synch* links to the synchronized overlay, a router R becomes also adjacent of:

(iii) the *synch* router, if such *synch* router is neighbor of R, or

(iv) all its neighbors, if R is the *synch* router.

With this definition, the adjacent overlay contains more links than the MPR flooding overlay generated by any interface in the network. Adjacent routers are expected to exchange their local instances of Link-State Database (LSDB) to report, and acknowledge to each other, changes

in their own LSDB local instance. Absent acknowledgment from an adjacent neighbor, LSAs are retransmitted. Router-LSAs received as part of adjacency-forming processes may be flooded over the network by the receiving interface if the LSA contains topology information more recent than that stored in the local instance of LSDB of the receiving interface.

In order to address the issues related to the stability of MPR links mentioned in section 8.4, MPR-OSPF does not tear down adjacencies as long as the corresponding links stay bidirectional, even if such adjacencies are no longer MPR links. This *persistent* approach for maintaining adjacencies is discussed and explored for other link-state operations in section 11.3, by way of persistent variations of MPR-OSPF.

Topology Selection

Links advertised in Router-LSAs describe local topology (set of maintained links or *link-state*) of the originating router. As specified in OSPF, only adjacent neighbors are advertised, in particular those selected as Path MPR and Path MPR selectors. Section 8.5 has shown that Path MPRs are sufficient for allowing every other router in the network to compute optimal routes to all destinations. Path MPR selectors are added for redundancy of the topology overlay – note that the same link is advertised twice, one by the interface being selected as Path MPR and another by its selector.

Neighbor Sensing

Routers discover and track their neighbors by exchanging Hello packets. Information contained in such Hellos enables routers to select and maintain their MPRs, both Flooding and Path MPRs.

Hello packets contain the list of neighbors of the originating router, as well as the list of neighbors which the router is adjacent with. Hello packets also advertise the cost of links between the originating router and its neighbors.

Information contained in Hello packets of all neighbors permits each router to identify the routers that are reachable in 2 hops or less, as well as the relationships between neighbors (routers

reachable in 1 hop) and routers reachable in 2 hops. In particular, a router x keeps two lists of neighbors:

- $N(x)$, the list of bidirectional neighbors, and
- $N_2(x)$, the list of 2-hop bidirectional neighbors, *i.e.*, bidirectional neighbors of at least one bidirectional neighbor of x.

In order to perform an accurate selection of Path MPRs, a router needs also to identify those neighbors that are both reachable in 1 and 2 hops.

Link Hierarchy

MPR-OSPF preserves the philosophy of OSPF routing, based on the principle that data traffic is sent over (i) shortest paths, and (ii) synchronized links (see section 10.2.1). This implies that the OSPF link hierarchy is respected, meaning that synchronized links (the synchronized overlay) are selected from among the graph of bidirectional links (the full network overlay), and in turn, the Shortest Path Tree (SPT) is computed over a topology selection overlay that includes all adjacent (synchronized) links. The use of the Path MPR algorithm for topology selection ensures that the paths computed by way of Dijkstra over the synchronized (adjacent) overlay are the shortest paths over the network.

11.2.2 Overlapping Relays & Smart Peering – OR/SP

The Overlapping Relays & Smart Peering extension for OSPF is based on the combination of two different mechanisms: Overlapping Relays, a variation of MPRs, and Smart Peering, presented and analyzed in chapter 9. In short, Smart Peering is used for LSDB synchronization (adjacent overlay creation and maintenance) and topology selection purposes. Reliable flooding is performed in two steps: LSAs from a router are first forwarded by its primary relays, which are the MPRs of the router selected among adjacent neighbors over the SP overlay. In case of failure of primary transmissions, additional (secondary) retransmissions may be performed by other adjacent neighbors.

Topology Information Dissemination

The Smart Peering decision is based on the ability of the computing router to determine whether a candidate neighbor is already part of the Smart Peering overlay (*i.e.*, is reachable through a path of Smart Peering-synchronized links) or not. While, in theory, the information about the Smart Peering overlay should be the same for both endpoints of the corresponding link, in practice the two involved routers may take different decisions, due to the impact of mobility, unreliability and topology information staleness. Since consistency cannot be ensured in the adjacency-forming decisions, LSDB synchronization processes are triggered when any of the involved routers (not necessarily both) selects the other as adjacent.

Flooding of Router-LSAs from its originator towards the rest of the network is performed over the synchronized (Smart Peering) overlay, in two steps. MPRs of a router are selected from among its adjacent neighbors so as to cover all its 2-hop adjacent neighbors – these MPRs are the *active overlapping relays* of the router, responsible for primary forwarding of LSAs flooded by that router. If a message is not acknowledged by any of the 2-hop adjacent neighbors of such router, the overhearing (overlapping) non-active relays (those adjacent 1-hop neighbors of the router that were not elected MPRs) can retransmit it until the missing acknowledgment is received.

As in OSPF, topology flooding requires that each router advertises its set of its (Smart Peering) synchronized links via LSAs. Nonetheless, since the Smart Peering overlay of a network does not necessarily include the network-wide shortest paths, RFC 5820 supports a complementary mechanism to advertise as well some additional bidirectional links, denominated *unsynchronized adjacencies*. These additional links are those for which synchronization was deemed unnecessary (and thus stay in bidirectional state) because they were already reachable (*routable*, in the terminology of OR / SP) through Smart Peering-synchronized paths. As the union of links that were synchronized and such *unsynchronized adjacencies* contains all available bidirectional links, this mechanisms permits ensuring route optimality, at the cost of requiring that all bidirectional links in the network are advertised in Router-LSAs.

Neighbor Sensing

Router interfaces exchange their lists of neighbors by way of Hello packets sent periodically. As some of the listed neighbors may be the same in consecutive Hello transmissions, the Overlapping Relays & Smart Peering extension provides a Hello optimization mechanism, denominated *Incremental Hellos*, that enables interfaces to advertise only the changes in the neighborhood occurred from the last Hello transmission. In case that a Hello packet from a particular interface is lost, receiving interfaces may request a full list of the neighbors of the such interface. This mechanism is described in detail in chapter 12

Link Hierarchy

In its basic form, *i.e.* without unsynchronized adjacencies, the Overlapping Relays & Smart Peering extension preserves the main principles of the OSPF link hierarchy (see Figure 10.2): flooding and topology selection are performed over the adjacent overlay, primary forwarding being managed through a MPR overlay built on top of the SP overlay, as Figure 11.1 indicates.

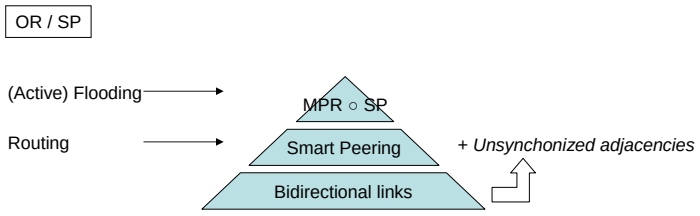


Figure 11.1: Link hierarchy in OR/SP.

As Smart Peering links may be not sufficient for providing shortest paths in routing, *unsynchronized links* may be also announced in Router-LSAs. The link hierarchy, however, is then different from the one of OSPF: rather than routing over a subset of synchronized links, the use of *unsynchronized adjacencies* allows forwarding of data traffic through all routable links in the network, both synchronized (SP) or not.

11.2.3 MANET Designated Routers – OSPF-MDR

The MDR extension of OSPF for mobile ad hoc networks adapts the mechanism of OSPF for adjacency optimization in broadcast networks (the Designated Router mechanism, see section 10.2.4) for MANET interfaces. OSPF-MDR proposes an interface hierarchy based on three states: one for MANET Designated Routers (MDR state), one for Backup MDRs ($BMDR$ state) and another one for interfaces without specific responsibilities ($MDROther$ state).

MDRs and BMDRs are elected by way of distributed algorithms based on 2-hop neighborhood information (thus requiring the exchange of Hello messages) that are executed by an interface any time there are significant changes in the neighborhood (*e.g.*, disappearance or appearance of bidirectional links [22]), or periodically before every Hello transmission. Unlike Designated Routers, MDRs are not necessarily persistent, according to the OSPF-MDR specification[1]: every new execution of the algorithm may lead to election of a new MDR. MDRs become adjacent to:

(i) all their neighbors not elected as MDRs, and

(ii) other MDRs.

The resulting overlay forms a Connected Dominating Set (CDS) [22]. MDRs assume the tasks of primary forwarding, although non-flooding MDRs may also be elected. Secondary forwarding, in case that the first retransmission is not acknowledged, is performed by Backup MDRs. It is worth to observe that, in OSPF-MDR, the decision of forwarding an LSA coming from a neighbor, describing the link with a neighbor or synchronizing its local instance of LSDB with a neighbor does not depend on the relationship with that neighbor, as in MPR-OSPF or OR/SP extensions. Rather, it depends on its own state and the state of the neighbor.

The performance of OSPF-MDR over a network may change depending on the value of three parameters that determine the way that link-state operations are performed: $AdjConnectivity$ affects LSDB synchronization, $LSAFullness$ affects topology selection and $MDRConstraint$ has effect on topology flooding.

[1]MDR persistency can be implemented as a variation, as it is done for instance in Boeing's implementation [54].

- *AdjConnectivity* (values from 0 to 2) regulates the adjacency-forming rule and enables adjacent overlays that may include the full network (value 0), form a uni-connected backbone (every router is adjacent to its MDRs, and MDRs become adjacent to each other in a uni-connected backbone, value 1) or a bi-connected backbone (each router is adjacent to MDRs and BMDRs, both MDRs and BMDRs becoming adjacent to each other, value 2).

- *MDRConstraint* determines the number of MDR elected in a network, high values implying more MDRs in the network and, therefore, more flooding overhead and higher number of adjacencies.

- *LSAFullness* (value from 0 to 4) defines the contents (set of advertised links) of Router-LSAs, all non-zero values providing shortest paths, the maximum (4) enabling routers to describe all their maintained bidirectional links in the LSAs.

OSPF-MDR provides a Hello optimization mechanism in order to prevent redundant neighborhood information to be sent in consecutive Hello transmissions, denominated *Differential Hellos*. The principle is similar to the one of *Incremental Hellos* (see section 11.2.2): interfaces in OSPF-MDR advertise changes in their neighborhood, and periodically they send a full list of their neighbors so that Hello packet losses can be overcome. This mechanism is described in detail and evaluated, together with the *Incremental Hellos* mechanism, in chapter 12.

11.3 Improved MPR-based Extensions

Two of the three IETF standard extensions (MPR-OSPF and OR / SP) are inspired, at least partially, by the Multi-Point Relaying technique. In this section, additional modifications of these two extensions are presented. These modifications are based on techniques described in chapters 7, 8 and 9. In particular, three modifications are considered: generalization of the persistency technique for operations other than adjacency selection, in section 11.3.1; implementation of the SLOT mechanism in MPR-OSPF, in section 11.3.2; and combination of MPR and Smart Peering in the MPR+SP extension, in section 11.3.3.

11.3.1 Persistency Variations of MPR-OSPF

As mentioned in chapter 8, an interface selects its multi-point relays in order to cover its 2-hop neighborhood. Changes in the 2-hop neighborhood may thus cause changes in the MPR set of a router, which leads to low average lifetimes of MPR links in dynamic networks, as shown in Figure 8.6. As this implies a high number of adjacency-forming processes when using MPR for LSDB synchronization, MPR-OSPF permits that a MPR-based adjacency is maintained in the synchronized overlay when the link is no longer an MPR link (def. 8.2), as long as the link remains bidirectional (see section 11.2.1). This approach provides a *persistent adjacency* technique for MPR.

The same persistency principle can be applied also to other link-state routing operations, such as topology selection and flooding. Four persistent variations of MPR-OSPF are examined in this Part of the manuscript (see also Table 11.1)[2]:

- *PPM (adjacency and flooding persistency).* The MPR synchronized overlay is persistent (def. 8.3), and Router LSAs are flooded through all adjacent links (including persistent adjacent links, in the sense of def. 8.4).
- *PMP (adjacency and topology persistency).* The MPR synchronized overlay is persistent, and all adjacent links (including those persistent) are advertised in LSAs.
- *PMM (only adjacency persistency).* The MPR synchronized overlay is persistent, but only non-persistent adjacencies (*i.e.*, links to Path MPRs or Path MPR selectors) are advertised in Router-LSAs. LSAs are only flooded over MPR links.
- *MMM (non-persistent approach).* Links are no longer adjacent when none of the involved nodes is MPR of the other.

PMM corresponds to the standard behavior of MPR-OSPF, as specified in RFC 5449 [24]. The analysis of the other variations (PMP and PPM) permits to measure the impact of persistency in each link-state operation, and MMM is included for completeness.

[2]Acronyms for the considered variations correspond to [P]ersistent / non-persistent [M]PR for (i) adjacencies, (ii) flooding and (iii) topology selection.

MPR-OSPF Variation	Adjacencies	Flooding	Topology
PPM	Persistent	Persistent	Path MPRs
PMP	Persistent	MPR selectors	Persistent
PMM (RFC 5449)	Persistent	MPR selectors	Path MPRs
MMM (Non-Persistent)	MPRs	MPR selectors	Path MPRs

Table 11.1: Considered MPR-OSPF variations.

11.3.2 SLOT over MPR-OSPF – SLOT-OSPF

SLOT-OSPF is a variation of MPR-OSPF that uses the same mechanisms for flooding optimization, reliability and topology reduction as specified in RFC 5449 [24], while exploring a new rule for adjacency-forming decisions. Instead of using the MPR overlay for adjacency selection and maintenance, SLOT-OSPF uses the overlay produced by the SLOT algorithm for unit costs (SLOT-U) described in chapter 7.

Adjacency Selection

The election of adjacent links is performed on the basis of triangular elimination: given a triangle, formed by links between interfaces x, y and z ($\{\overline{xy}, \overline{yz}, \overline{zx}\}$), one of the links (the one connecting interfaces with highest ids) is removed from the overlay, the others two being added. Let id_x be the identity of interface x, then:

$$(id_x > id_z) \wedge (id_y > id_z) \iff id_{\overline{xy}} > id_z \tag{11.1}$$

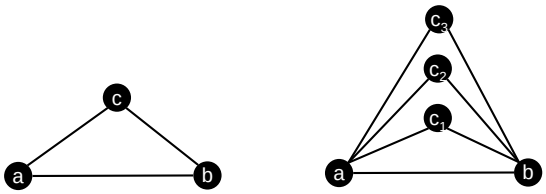
In case of multi-triangles (see Figure 11.2), that is, if there are several common neighbors z_i of x and y ($i > 1$), then a link not included in the overlay if and only if

$$\exists i : id_{\overline{xy}} > id_{z_i} \tag{11.2}$$

Equivalently, a link $\overline{xy}$ is only added to the adjacent overlay if and only if

$$\forall z_i, id_{\overline{xy}} \leq id_{z_i} \tag{11.3}$$

Since all the links and interfaces considered in the (multi-)triangles are known by interfaces x and y attached to the link $\overline{xy}$, the adjacency-forming decision is consistent in both endpoints: an



(a) Triangle abc (b) Multi-triangle around ab

Figure 11.2: Link triangles and multi-triangles around nodes a and b.

adjacency is only formed when both interfaces agree to add the link between them to the synchronized overlay.

11.3.3 Multipoint Relays + Smart Peering – MPR+SP

MPR+SP combines the techniques described in chapters 8 and 9, already used in RFC 5449 [24] and RFC 5820 [19]. The MPR algorithm is used for topology flooding and for selection of links taking part in the Shortest Path Tree (SPT) computation. Adjacencies are selected by way of Smart Peering, in order to minimize the overhead caused by LSDB synchronization in ad hoc networks.

Topology Information Diffusion

MPR+SP performs reliable LSA flooding in the same way as MPR-OSPF: Router-LSAs originated or flooded by an interface are forwarded by the MPRs of such interface. Interfaces acknowledge the reception of Router-LSAs such Router-LSAs come from an adjacent neighbor. Smart Peering is used for adjacencies: a link is synchronized when any of the two attached interfaces selects the other interface by means of the Smart Peering rule. As in MPR-OSPF, Router-LSAs requested and received as part of adjacency-forming processes may be flooded by the receiving interface over the network, if the LSA contains topology information more recent than the one locally stored on the receiving interface's local instance of LSDB.

The topology information, collected by Router-LSAs and Hello packets, is used for com-

puting the Shortest Path Tree (SPT). In MPR+SP, routers re-construct a network subgraph that contains the following components:

1) Path MPRs of every router in the network, listed in the corresponding Router-LSAs.

2) Adjacencies maintained by every router in the network, as reported in Router-LSAs.

3) 1-hop and 2-hop neighbors of the router that performs the computation, as reported via Hello packets.

From *Lemma 9.1*, the subgraph formed by components 1) and 3) contains the shortest path of the computing router to every other router in the network (vertex in the network graph). Adjacencies are, however, required for the Smart Peering adjacency selection. This is due to the fact that the synchronization of a link between two interfaces depends on whether there is an existing synchronized path between such interfaces over the network (see chapter 9).

Figure 11.3 illustrates in a simple static network the three components of the subgraph that node *1* generates. Figure 11.3.a displays the complete network graph, and Figures 11.3.a, b, c and d indicate (thick lines) the subgraphs corresponding to the Path MPRs overlay, node 1's 1-hop and 2-hop neighborhood and the Smart Peering adjacent overlay, respectively. The SP overlay in a static network cannot be unambiguously deduced from the network graph, as explained in section 9.3. For the example in Figure 11.3.d, it has been assumed that (i) the order of appearance of the nodes correspond to their id (that is, node i will appear in the network before node j if $i < j$), (ii) adjacency-forming processes are not concurrent, and (iii) older nodes have priority to form an adjacency to a new neighbor. It can be observed that the three components may overlap.

Inclusion of Path MPR links and the Smart Peering overlay in the LSDB leads to a dual network topology representation: the complete graph is used for computation of optimal routes and thus for data traffic routing, whereas the restricted subgraph containing SP links is only used for adjacency selection purposes.

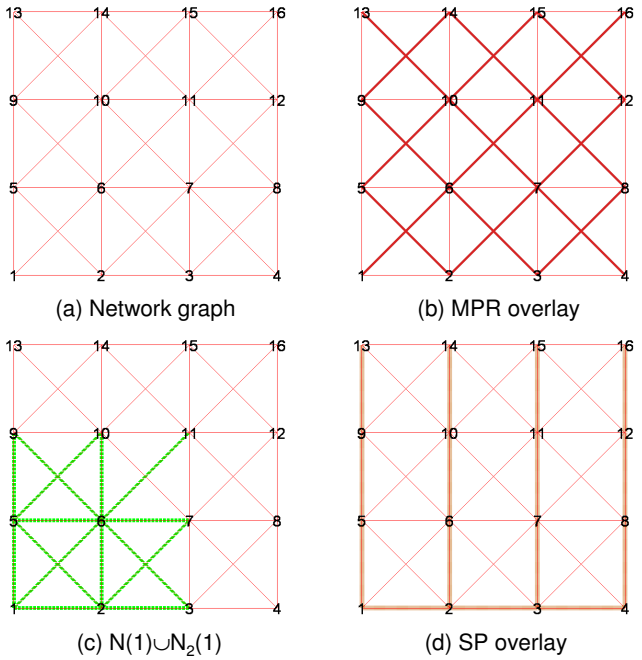


Figure 11.3: Example of static network and the components of the topology subgraph reconstructed by node (1): **(a)** Network graph, **(b)** Path MPR overlay, **(c)** 1-hop and 2-hop neighborhood of (1), and **(d)** (a possible) Smart Peering overlay.

Neighbor Sensing

As in MPR-OSPF, MPRs are selected in MPR+SP from among bidirectional 1-hop neighbors of the computing interface, and are expected to cover all its bidirectional 2-hop neighbors. Neighbors selected as MPRs by an interface are identified as MPRs in Hello packets of such interface. Given that Hello format is mostly maintained as in MPR-OSPF, Hello packets advertise the identity of the transmitting interface's neighbors and the cost of links between such interface and its neighbors. Unlike MPR-OSPF, in which adjacent neighbors were either Path MPR or Path MPR selectors, in MPR+SP Path MPRs are not necessarily adjacent, adjacencies being selected by way of Smart Peering. It is thus necessary to advertise independent lists of (i) Path MPRs, Path MPR selectors and (ii) adjacent (Smart Peering) neighbors.

Link Hierarchy

MPR+SP's architecture has a non-negligible impact on the link hierarchy present in OSPF (see Figure 10.2) and some of its MANET extensions (*e.g.*, RFC 5449). Figure 11.4 indicates the changes that MPR+SP implies in this regard.

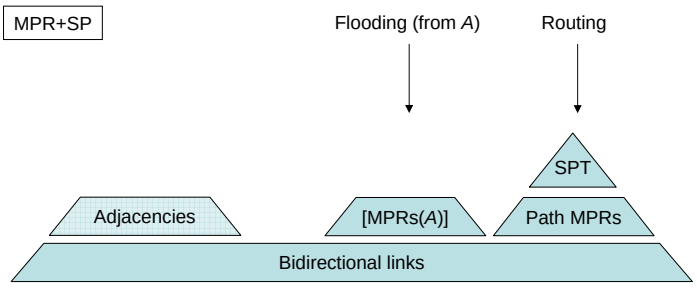


Figure 11.4: Link hierarchy in MPR+SP.

For each interface x from the network, MPR+SP generates two subgraphs based on the graph of bidirectional links within the network:

- the MPR subset, formed by the MPRs of x, the MPRs of these MPRs and so on; and
- the Path MPR subgraph containing Path MPRs of every node in the network.

These two subgraphs are used in MPR+SP for topology flooding and data traffic routing, respectively: flooding of Router LSAs is performed over the MPR subgraph, while the Shortest Path Tree of x is mostly extracted from the Path MPR subset. Unlike OSPF and the studied IETF extensions for MANETs (MPR-OSPF and OR/SP), in MPR+SP none of these subgraphs (flooding and advertised links) is necessarily contained in the subgraph of adjacencies. The adjacencies' subgraph is only used in MPR+SP for LSDB synchronization purposes.

Contrary to OSPF and its IETF standard extensions for MANET extensions, neither of these subgraphs is necessarily contained in the subgraph of adjacencies. Such subgraph is only used for LSDB synchronization purposes.

11.4 Conclusion

OSPF can be used for routing on MANETs and, more in general, as a single routing protocol for compound Autonomous Systems. This, however, requires the deployment of a MANET-specific interface type in OSPF, as the interface types specified in OSPF do not provide support for ad hoc operation, as shown in chapter 10. Such a MANET interface for OSPF is intended to address efficiently ad hoc networking issues while keeping a single, well-known protocol such as OSPF for handling routing in the whole compound AS.

Three extensions of OSPF for MANET operation have been proposed by the IETF: the Multi-Point Relays extension (MPR-OSPF, RFC 5449), the Overlapping Relays & Smart Peering extension (OR / SP, RFC 5820) and the MANET Designated Routers extension (OSPF-MDR, RFC 5614). All extensions have different approaches to a MANET interface type for OSPF. Two of these extensions (MPR-OSPF and OR / SP) explore variations of the multi-point relaying technique for routing in MANETs. The third extension, OSPF-MDR, takes a different approach that extends the OSPF notion of Designated Router, designed to centralize the flooding in broadcast networks, to ad hoc networks. MPR-OSPF and OR / SP are evaluated in the rest of Part III of this manuscript.

This chapter has presented, together with these IETF extensions of OSPF, some additional modifications of MPR-based extensions based on the techniques described in Part II of this manuscript. SLOT-OSPF is a variation of MPR-OSPF in which adjacencies are selected and maintained according to the SLOT-U technique. MPR+SP combines the multi-point relays from MPR-OSPF with the Smart Peering technique for adjacency-forming purposes. Moreover, the adjacency persistency of MPR-OSPF is explored and discussed for other link-state operations. The performance of all the MPR-based extensions presented in this chapter will be examined through an extensive simulation-based analysis in chapter 12.

Chapter 12

Performance Evaluation of OSPF via MANET Simulations

The MPR-based OSPF extensions for MANET operation presented in chapter 11 use, in different ways, the techniques described in Part II of this manuscript. While each one defines a particular MANET interface for OSPF, these extensions take slightly different approaches concerning the main aspects of OSPF behavior: route selection for user data, topology flooding and role of adjacencies.

This chapter performs a qualitative and quantitative analysis of the overall performance of these various extensions, and evaluates the impact that the mechanisms used in each of them have in such performance. The quantitative evaluation is based on simulations through GTNetS [74] of each extension in (mobile) ad hoc networks – without considering fixed networks. Simulated scenarios focus on networks carrying a substantial amount of user data traffic with moderate mobility profiles. The performed experiments test the protocols performance for different values of network size and density, the network reaction to different data traffic loads and the protocol robustness with respect to the wireless link quality. For a detailed description of the parameters used in simulations, see Appendix E.

12.1 Outline

Section 12.2 discusses the adaptation of the two main elements of OSPF, shortest paths and adjacencies, to the networking conditions of MANETs. Section 12.3 discusses and compares two optimization techniques for neighbor sensing (Hello) traffic that are present in OR / SP and OSPF-MDR standard extensions. Such Hello optimization techniques are independent of the link-state operations, and can therefore be applied to virtually any discussed protocol that requires neighbor sensing – in particular, MPR-based extensions of OSPF for MANET. Section 12.4 studies the way that such link-state operations are performed in each of the MPR-based extensions. Section 12.5 explores the use of the persistency mechanism, by evaluating its impact in different variations of MPR-OSPF. Finally, section 12.6 concludes the chapter.

12.2 Synchronization & Optimal Routes in OSPF and MANET Extensions

The concept of a *synchronized link* or *adjacency* is essential in the architecture of OSPF. Adjacencies assume a leading role not only in terms of LSDB exchange (link synchronization), but also in control traffic dissemination (flooding) and Shortest Path Tree computation (topology selection).

Standard OSPF operation can be summarized in the two following principles:

(1) User data is always forwarded over the network-wide shortest paths.

(2) User data and control traffic is only forwarded over links between routers whose local instances of LSDB have been explicitly synchronized.

In wired networks, the first principle aims at reducing delays and overhead endured by data traffic. The second principle aims at reducing risks of routing loops. The effect of these principles is, however, not evident in the performance of routing in multi-hop wireless ad hoc networks. The use of a single, multipurpose overlay for the three operations may yield a suboptimal routing performance,

if requirements of route optimality are not fulfilled; or lead to inefficient results, if route optimality is achieved by not performing other possible partial optimizations. In this context, the OSPF MANET extensions take different approaches for determining the role of adjacencies and their relationship with the three main link-state operations. In consequence, the OSPF MANET extensions explore different approaches with respect to such principles. This section overviews the most significant conclusions about applicability of these principles, obtained from comparing the performance of MPR-based OSPF MANET extensions. Results of this comparison are detailed in the rest of the chapter.

12.2.1 User Data over Shortest Paths

The concept of *shortest paths* employed in principle (1) depends on the metric used to compare the cost of links and paths. Experiments presented in this chapter use a hop-count metric, as this is one of the simplest and most widely used route metrics in ad hoc networks [25]. The obtained results, however, are not particular to this metric and can be generalized to any additive metric.

The results presented throughout this chapter (see section 12.4.2) show a significant performance penalty when routing is not performed over (asymptotically) optimal paths. In the simulated scenarios, data paths suboptimality increases significantly the amount of user data traffic in the AS, thus reducing the available bandwidth in the network and affecting negatively the overall routing quality. This effect should be taken into account when considering other possible side benefits associated to approaches not providing optimal paths. Even in the case of a dynamic topology, in which topology information transmitted over the network may become stale within a short time, the use of shortest paths has a positive impact on the routing quality of the protocol.

12.2.2 User Data & Control Traffic over Synchronized Links

The concept of *synchronized links*, as described for OSPF in section 10.2.1 and used in OSPF routing according to principle (2), raises three main issues when applied in ad hoc networks:

- **Existence of synchronized links.** Due to the short lifetime of links in ad hoc networks, compared to wired links, it may be wasteful to use bandwidth in LSDB synchronization processes; there may not even be enough time to finish the synchronization before the link breaks.
- **Definition of synchronized links.** According to section 10.2.1, link synchronization implies that local instances of LSDB of the two interfaces of the link maintain the same topology information and changes in one of them cause changes in the other. In terms of overlays, this implies that the synchronized overlay is contained in (or equivalent to) the flooding overlay of any interface in the network. In OSPF MANET extensions such as MPR+SP and SLOT-OSPF, however, links used for flooding are not necessarily synchronized links.
- **Use of synchronized links.** The participation of adjacencies in the three link-state operations varies in the OSPF MANET extensions.

The existence of LSDB synchronization in all OSPF MANET extensions is due to two main reasons: (i) adjacencies are a legacy from OSPF, so they are kept for compatibility in OSPF MANET extensions, and (ii) in the context of routing within compound ASes based on extended OSPF, LSDB synchronization is useful for distributing topology information between fixed and ad hoc networks of the AS, as shown in section 4.4.3.

There are differences, however, in the use of adjacencies in the OSPF MANET extensions. The results presented throughout this chapter indicate the following conclusions:

- For user data, a clear advantage in terms of data delivery could not be identified in simulations between (i) using paths made only of synchronized links, and (ii) using paths made both with synchronized and other non-synchronized links in MANETs. Equivalently, no significant performance differences could be observed between extensions for which all advertised links were necessarily synchronized ($\{\text{advertised links}\} \subseteq \{\text{synchronized links}\}$) and those extensions for which synchronization was not a requisite for advertising a link.
- Extensions that include synchronized links in topology selection (*i.e.*, those for which $\{\text{synchronized links}\} \subseteq \{\text{advertised links}\}$), however, achieve a better routing quality (data delivery) than those for

which the SPT does not take into consideration all synchronized links. It is worth to remark that this observation is not contradictory with the previous one.

These conclusions suggest that synchronized links play a non-negligible role in OSPF routing on ad hoc networks. They cannot be used in the same way as they were used in standard OSPF, due to the higher relative cost of LSDB synchronization (with respect to the available bandwidth) and the short lifetime of links in wireless ad hoc networks. The presence of synchronized links in the computation of the Shortest Path Tree, however, has a positive effect in routing quality. Such synchronized links may be completed if necessary with additional links so as to enable selection of network-wide shortest paths.

12.3 Neighbor Sensing Optimization

Although the traffic caused by Hello packet exchange is a relatively small source of control traffic for routing protocol in mobile networks [44], some optimization techniques for information carried by Hello packets have been explored in the framework of the research efforts for extending OSPF to MANET operation, as mentioned in chapter 11. This section explores the optimization that consists of avoiding the transmission of redundant information in Hello packets by only reporting changes in the neighborhood occurred since the last Hello transmission. Throughout the section, the term *synchronism* is used to denote the situation in which the neighbor of an interface has complete information about the neighborhood of the interface, updated the last time that the interface sent a Hello packet.

The use of Hello optimization techniques implies that the failure of a single Hello transmission performed by an interface may cause the loss of Hello synchronism and prevent the neighbors of such interface to track changes in the neighborhood of the transmitting interface. Techniques exploring this optimization need thus to provide Hello loss detection and synchronism recovery mechanisms in order to restore accuracy of the information maintained by neighbors of the Hello transmitting interface.

These techniques have two main drawbacks. First, the artificial reduction of Hello packet sizes may lead routers to an unrealistic overestimation of link quality, as shorter packets are more likely to be successfully delivered than longer packets [55, 86]. While having a small impact in the overall overhead, such techniques may thus damage the routing quality, as the experiments from Chakeres suggest [86]. Second, delays in reestablishing synchronism in case of loss of a single Hello packet may be harmful in terms of accuracy of neighborhood information, in particular if routing and flooding decisions rely on such knowledge (*e.g.*, MPR-OSPF, OSPF-MDR or SLOT-OSPF; but not OR/SP).

This section focuses on the evaluation of both techniques in terms of traffic overhead.

12.3.1 Proactive and Reactive Synchronism Recovery

Two approaches have been explored in the framework of OSPF MANET extensions: *differential Hellos* or proactive synchronism recovery mechanism of OSPF-MDR, and *incremental Hellos* or reactive synchronism recovery mechanism for OR / SP. Both approaches provide sequence numbers in Hello packets in order to detect losses of synchronism, and provide different mechanisms for restoring synchronism when a loss is detected.

Although they are part of two OSPF MANET extensions (OSPF-MDR and OR/SP), both approaches can be analyzed independently from such extensions, and could be applied to any neighbor sensing protocol based on the periodic exchange of messages.

- **Proactive Synchronism Recovery**. This approach, specified in [22], allows router interfaces to report only changes in the neighborhood, via differential Hello packets. Such differential Hello packets only contain information about neighbors having changed its neighbor state since the last Hello packet transmission. Once every n Hello transmissions (configurable), a router transmits a full Hello packet instead of a differential Hello packet. In case that any differential packet is lost, these periodical full transmissions (with interval $nHelloInterval$) permit every neighbor to recover Hello synchronism. The number n of differential Hello transmissions per full Hello transmission indicates the trade-off between the reduction in the amount of Hello

information (higher reduction as n increases) and the average time that a receiving interface would require in order to restore synchronism in case of Hello transmission failure (also higher as n increases).

- **Reactive Synchronism Recovery**. This approach is specified as an additional feature in [19]. Unlike the differential mechanism, in which the interface that receives a Hello packet assumes a passive role, in the incremental approach the receiving interface is responsible for synchronism recovery. When an interface joins the network or detects a Hello packet transmission failure (by detecting a gap between two consecutive received Hellos), the interface requests the corresponding Hello originating node(s) for a full transmission.

Failure detection and synchronism recovery mechanisms are needed in both neighbor optimization techniques, in particular when such techniques are used for neighbor sensing over unreliable wireless links. The performance of detection and synchronism recovery mechanisms can be evaluated by way of the maximum time interval that incremental and differential approaches need to detect and restore synchronism after a Hello packet loss. Figure 12.1 illustrates the different behavior of the two mechanisms in a context of single Hello packet loss.

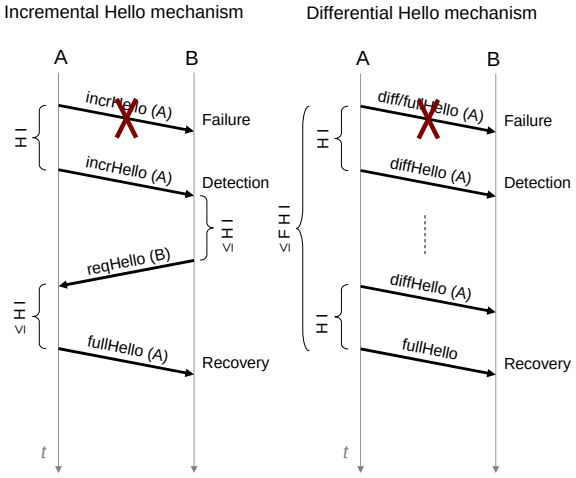


Figure 12.1: Differential and incremental behavior in case of a single Hello packet transmission failure. *HI denotes the time between two consecutive Hello packet transmissions, and FHI denotes the time interval between two consecutive full Hello packet transmissions.*

12.3.2 Overhead Impact

Figure 12.2 shows the impact of these two optimization techniques, in terms of relative Hello traffic reduction. The overhead reduction achieved by using such techniques remain is low: less than a 18% reduction of Hello traffic is achieved, at best, which represents less than 2% reduction of the total control traffic.

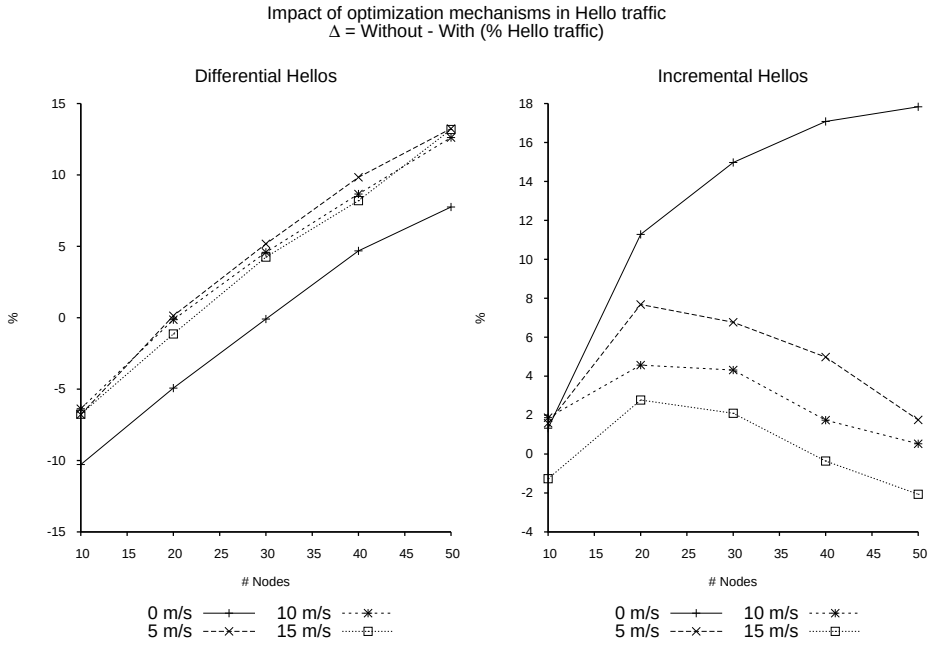


Figure 12.2: Impact of optimization mechanisms in Hello traffic (%).

In some cases, these optimizations are counterproductive, meaning that they generate more overhead than when not using them. This is caused by the additional overhead required to signal neighbor changes. The incremental approach is not able to significantly reduce Hello traffic in mobile and dense scenarios. In these scenarios, the presence of many neighbors sharing the same wireless medium imply that transmitted packets, and Hello packets in particular, are more likely to be lost. When incremental Hellos are used, this situation causes additional requests and full Hello transmissions in reply.

Differential Hellos achieve a slightly higher overhead reduction (maximum, about 13%) than

incremental Hellos (maximum, about 8%), for mobile scenarios. For a fair comparison, however, it must be taken into account that the performance of differential Hellos in terms of overhead reduction is at the expense of not being proactive in the recovery of synchronism after a Hello packet loss. Interfaces using incremental Hellos and detecting a Hello packet loss do not request immediately for a full Hello packet transmission; they can only wait until the next full transmission is performed (see Figure 12.1). This implies that, in case of Hello packet loss, interfaces using differential Hellos stay a longer time interval, in average, before recovering Hello synchronism than interfaces using incremental Hellos.

12.4 Main Link-State Operations

Differences in the role of adjacencies and the link hierarchy used by the OSPF MANET extensions have a significant impact in the performance on all the link-state operations. This section describes the impact observed in simulations for flooding (section 12.4.1), topology selection (section 12.4.2) and LSDB synchronization (section 12.4.3).

12.4.1 Flooding

Reliable flooding is performed in the OSPF extensions by way of two different mechanisms. The first mechanism is used in MPR-OSPF and its variations (SLOT-OSPF, MPR+SP and persistency variations of MPR-OSPF); the second is used in OR/SP. While both mechanisms are based on the multi-point relaying (MPR) technique, they differ in two aspects: (i) the election of MPRs and (ii) the mechanism to ensure that such LSAs are correctly received by the intended destinations – that is, the reliability mechanism.

MPR Selection

This section explores the implications of the MPR election over the SP-synchronized overlay, for OR/SP, and over the overlay containing all bidirectional links, for MPR-OSPF. When compared with the selection of Multi-Point Relays among the bidirectional neighbors of an interface, Overlap-

ping Relays presents a lower amount of MPRs per router and a significantly higher stability of such relays, as shown in Figures 12.3.a and 12.3.b.

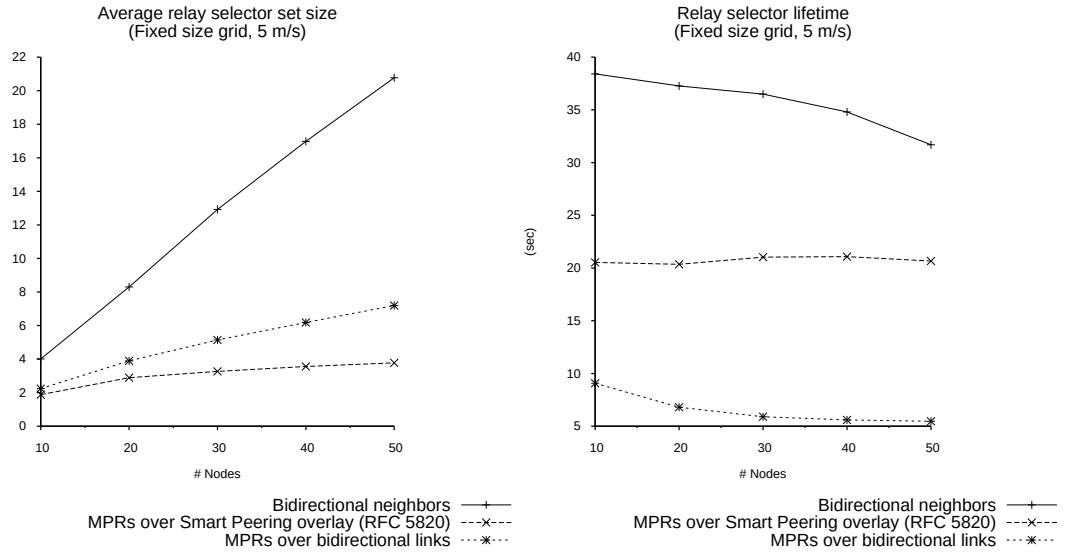


Figure 12.3: **(a)** Average size of the MPR set and **(b)** average relay lifetime ($5m/s$). **(c)** LSA retransmission ratio, depending on the link quality (30 routers, $5m/s$). *The LSA retransmission ratio is the number of backup LSA retransmissions over the number of primary LSA transmissions.*

The drawbacks of computing MPRs over a restricted overlay (such as the Smart Peering overlay) are however significant. In first term, computing MPRs this way weakens the main advantage of using Multi-Point Relays for flooding, which is the ability of reaching all the 2-hop neighbors of the interface while avoiding redundant transmissions. Since the neighborhood topology in which MPR operates is pruned by the Smart Peering selection rule, the set of reachable 2-hop neighbors becomes also affected and the quality of the flooding operation becomes damaged, as Figure 12.4 indicates.

In second term, MPR selection over the Smart Peering overlay makes the MPR computation nearly irrelevant. If the probability of relaying an MPR flood is close to $\frac{M_r}{M}$ (with M_r being the average number of relays per router and M the average number of bidirectional neighbors), the situation in sparse networks (such as the Smart Peering overlay) is close to $M_r = M$, meaning that almost every SP-synchronized neighbor will become a multi-point relay, thus making wasteful the relay selection process.

Reliability

Reliability mechanisms based on acknowledgments present issues in mobile ad hoc networks, in particular those with high mobility patterns. Due to the relative mobility between neighboring routers, the approach in which the transmitter either receives an acknowledgment or retransmit the corresponding packet until the acknowledgement is received (or the transmitter stops retransmitting) may incur in inefficiency and additional overhead, as detailed in section 4.3.1.

Issues related to additional overhead are more significant if the acknowledgement mechanism involves more routers than the transmitter and the intended receiver(s) of a packet, as it is the case of the Overlapping Relays extension. The reliability mechanism of OR uses a third type of routers, the non-active relays of the source overhearing the communication between the active relays (transmitters) and the 2-hop neighbors of the source to be covered. Such non-active relays retransmit the packet forwarded by active relays in case that no acknowledgment is received. That increases the complexity of the mechanism, both in terms of synchronization and buffer management.

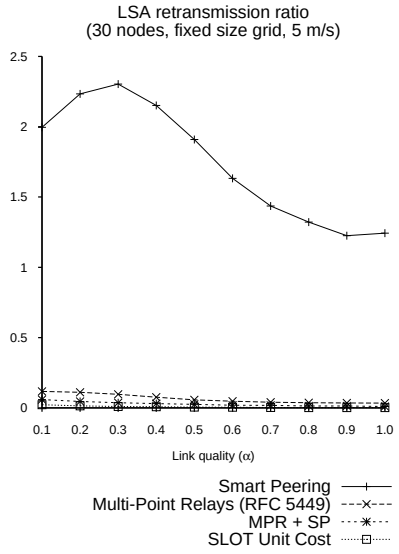


Figure 12.4: LSA retransmission ratio, depending on the link quality (30 routers), $5m/s$. *The LSA retransmission ratio is the number of backup LSA retransmissions over the number of primary LSA transmissions.*

Figure 12.4 compares the impact in flooding performance of such acknowledgement mech-

anism with the single acknowledgment of standard OSPF and RFC 5449-like extensions – *i.e.*, MPR-OSPF, SLOT-OSPF and MPR+SP. The presence of a set of additional retransmitting neighbors, the non-active relays, leads to a substantial increase in the number of LSA retransmissions, and thus in the amount of control traffic overhead. This additional overhead, however, does not provide any significant improvement of the overall routing quality of the protocols – as it is shown in further sections of this chapter.

12.4.2 Topology Selection

Within the considered MPR-based extensions, two mechanisms for topology selection can be distinguished, as shown in chapter 11. In MPR-OSPF and the extensions based on MPR-OSPF (MPR+SP and SLOT-OSPF), each interface describes in its Router-LSA a set of links that is sufficient for computing network-wide shortest paths. Such set of links includes the set of Path MPRs of such interface. In OR/SP without *unsynchronized adjacencies*, in contrast, interfaces only describe the set of SP-synchronized links, and thus network-wide shortest paths are not necessarily included in the network overlay contained in local instances of LSDB.

Figure 12.5.a shows the average path length provided by these two mechanisms, and indicates that Smart Peering provides substantially longer paths. This implies that for sending the same amount of data traffic, extensions not providing shortest paths (such as OR/SP without *unsynchronized adjacencies*) require that a significantly higher amount of traffic is forwarded over the network, as shown in Figure 12.5.b. This is due to the fact that each data packet is forwarded, at least, a number of times equal to the length (in hops) of its path towards the destination. The use of shortest paths has a positive significant effect for minimizing in ad hoc networks the amount of data traffic required for delivering a fixed amount of data over the network. Similar results were also observed in other scenarios, with different speeds. The difference between both mechanisms in the amount of data traffic and overall traffic in the network (Figures 12.5.b and 12.6.b) can only be expected to grow wider with more user data input – presented results report up to 2Mbps.

The probability that a data packet cannot be delivered to its intended destination increases

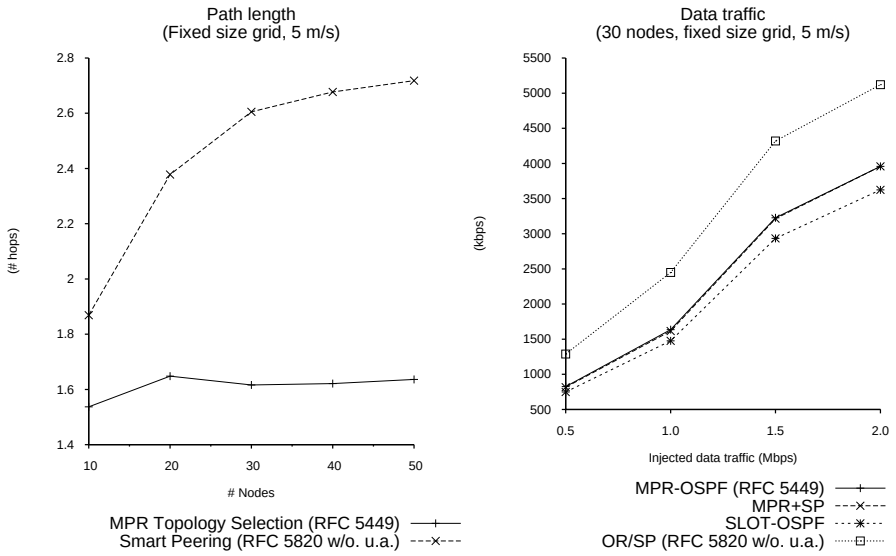


Figure 12.5: **(a)** Path length, **(b)** User data traffic, for $5m/s$.

as the path towards such destination is longer, as Figure 12.6.a indicates. Losses may happen if any of the wireless transmissions of a data packet fails, which is more likely when more wireless hops are used.

Figure 12.6.a also shows that extensions using shortest paths (MPR-OSPF, SLOT-OSPF, MPR+SP) present significant differences in terms of data delivery ratio: MPR-OSPF and MPR+SP are able to deliver about a 4% more data packets than SLOT-OSPF. Whereas in all these extensions the interfaces advertise Path MPRs in their Router-LSAs, in MPR-OSPF and MPR+SP interfaces include in the set of advertised links all their adjacencies. In contrast, in SLOT-OSPF only those adjacent neighbors that are selected Path MPRs of the transmitting interface are included in the Router-LSA.

The effect of these differences is due to the dynamic nature of topology in ad hoc networks. In absence of such dynamism, advertisement of Path MPRs is sufficient for computing shortest paths over the network (*Lemma 8.4*). When topology is dynamic, shortest paths computed by an interface and based on the received Router-LSAs may not correspond to the optimal routes over the network. Then, the fact that routes are computed over a network graph that includes all synchronized links

(adjacencies), as in MPR-OSPF and MPR+SP, proves to be more efficient, in terms of data packet delivery, than computing routes over a network graph which does not include all synchronized links. In this last case, it is more probable that data packets are routed over non-synchronized links. In such links, attached interfaces may maintain non-consistent information in their local instances of LSDB, making thus routing loops possible.

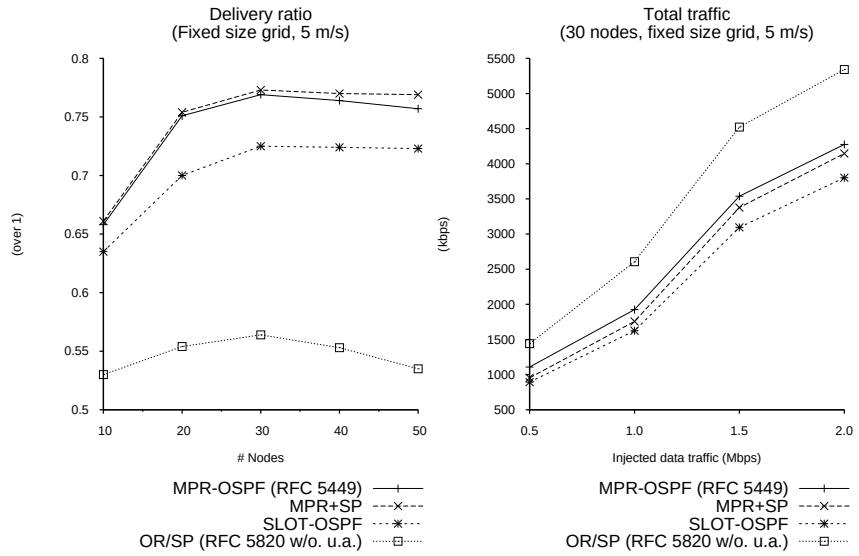


Figure 12.6: **(a)** Data delivery ratio and **(b)** total traffic (data + control) in the network (30 nodes, $5m/s$).

Finally, these results confirms the argument, presented in section 12.2, that it is beneficial to handle LSDB synchronization and topology selection independently in mobile ad hoc networks. Indeed, these two link-state operations are associated with different requirements that cannot be simultaneously fulfilled without incurring in suboptimality and inefficiency issues.

12.4.3 LSDB Synchronization

Three synchronization techniques are used in OSPF MANET extensions: the (persistent) MPR synchronized overlay, presented in chapter 8 and used by MPR-OSPF, the Smart Peering overlay, described in chapter 9 and used by OR / SP and MPR+SP extensions, and the SLOT overlay, proposed in chapter 7 and implemented in OSPF through the SLOT-OSPF extension. All

algorithms are evaluated for the case of hop count link metrics, which in particular implies that the SLOT-U variation is taken into consideration.

Figure 12.7 presents the average number of adjacencies per node and the average stability of adjacencies in each of the OSPF extensions. Figure 12.7.a indicates that Smart Peering (in OR/SP and MPR+SP) reduces significantly the number of adjacencies with respect to MPR-OSPF. This latter reaches its maximum in the displayed scenario (fixed size grid, 5 m/s) with $9.34\frac{adj}{node}$, before decreasing due to network traffic overload. A substantial part of MPR-OSPF adjacencies are, however, *persistent* adjacencies and thus do not imply significant additional overhead: in terms of adjacencies triggering a database exchange process (non-persistent MPR links, def. 8.4), MPR-OSPF achieves a slightly lower amount of synchronized links per interface than Smart Peering. The impact of such persistent adjacencies, and other persistent approaches, is discussed in detail in section 12.5.

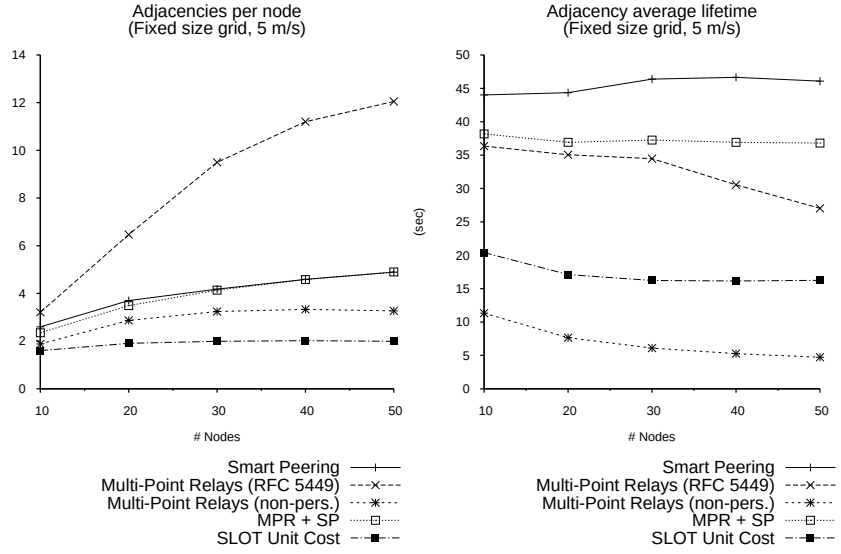


Figure 12.7: (a) Average number of adjacencies per node and (b) average adjacency lifetime, for different network densities (fixed size grid, 5 m/s).

Without considering persistent adjacencies of MPR-OSPF, the MPR technique produces a synchronized overlay that contains more links per interface than SLOT-U. This comes directly from the definition of each technique, and can be shown by considering the interfaces x and y, distant two hops through a common neighbor z:

- The MPR coverage criterion (def. 8.1) ensures that there is a synchronized path between x and y with length exactly 2 – as y is a 2-hop neighbor of x, and vice versa.
- In SLOT, LSDB synchronization decisions are based on 1-hop information, as shown in chapter 7. SLOT ensures therefore that there is a synchronized path between x and z, and between z and y. Each of these synchronized paths may have length 1 or 2, so the synchronized path between x and y may have length (in number of hops) 2, 3 and 4.

As SLOT synchronized paths between interfaces can be longer than MPR synchronized paths, the number of synchronized links per interface within SLOT is necessarily lower than within MPR.

Behavior of Smart Peering cannot be explained by way of this argument. From the SP definition (see chapter 9), synchronized paths between two interfaces x and y at distance 2 hops could have any length ≥ 2. The average number of adjacencies per interface shown in Figure 12.7.a, however, is significantly higher than for MPR-OSPF (considering non-persistent synchronized links) and SLOT-OSPF. This is due to the fact that Smart Peering decisions are based on global information obtained via Router-LSAs – unlike MPR-OSPF and SLOT-OSPF, both based on local information acquired via Hellos. If an interface has not received information about the existence of a synchronized path over the network with a neighbor, for instance due to losses in the flooding procedure, such interface may become adjacent to such neighbor.

The average adjacency lifetime for each of the extensions is displayed in Figure 12.7.b. As described in chapter 9, adjacencies selected through the SLOT or the Smart Peering techniques (both in MPR+SP and Overlapping Relays) are significantly more stable than those selected by MPR-OSPF (not taking into account persistent MPR adjacencies). The Smart Peering capacity for selecting the most stable links is also illustrated in Figure 12.9, where the size of the SP synchronized overlay does not change significantly depending on the wireless channel quality ($\alpha \in [0,1]$, $\alpha = 1$ for ideal wireless channel), and stays constant for $\alpha \geq 0.5$. In MPR-OSPF, in contrast, the set of adjacencies per interface keeps growing as the wireless channel quality α increases. Concerning SLOT-U, the improvement over MPR-OSPF is significant and confirms the theoretical analysis of

chapter 7. The fact that only (static) 1-hop neighborhood information is taken into consideration reduces the ability to provide stable links with respect to the global-based Smart Peering technique.

There is a non-negligible gap between the adjacency lifetime traces from MPR+SP and from Overlapping Relays. This gap has no relationship with the LSDB synchronization technique, because Smart Peering is used in both cases. Rather, it is due to the neighbor keep-alive mechanism. In OSPF, a interface declares a neighbor *dead* if it has not received a Hello packet from it during a *DeadInterval* period. However, in a lossy channel Hello packets can be lost with a probability that increases with the length of the packet (see the lossy channel model in [54]). Figure 12.8 shows the average Hello packet size for the three extensions.

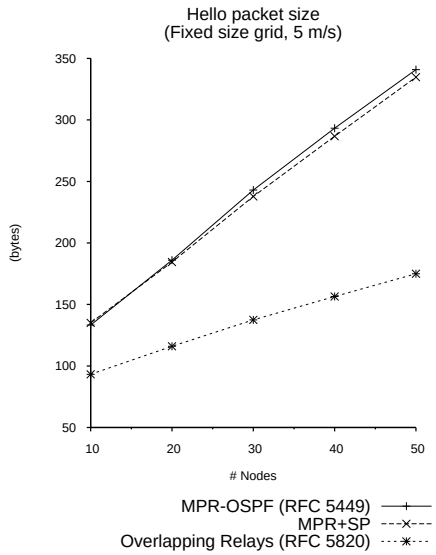


Figure 12.8: Average size of Hello packets (fixed grid, 5 m/s).

Aside from the fact that such keep-alive does not take packets other than Hellos into account, this policy causes extensions with a longer Hello packet format (such as MPR-OSPF or MPR+SP) to be more likely to declare false *dead* neighbors in lossy channels than those with a shorter Hello packet format (such as Overlapping Relays). That makes the adjacency stability of extensions with a longer Hello packet format more depending on wireless channel quality, as shown in Figure 12.9.b.

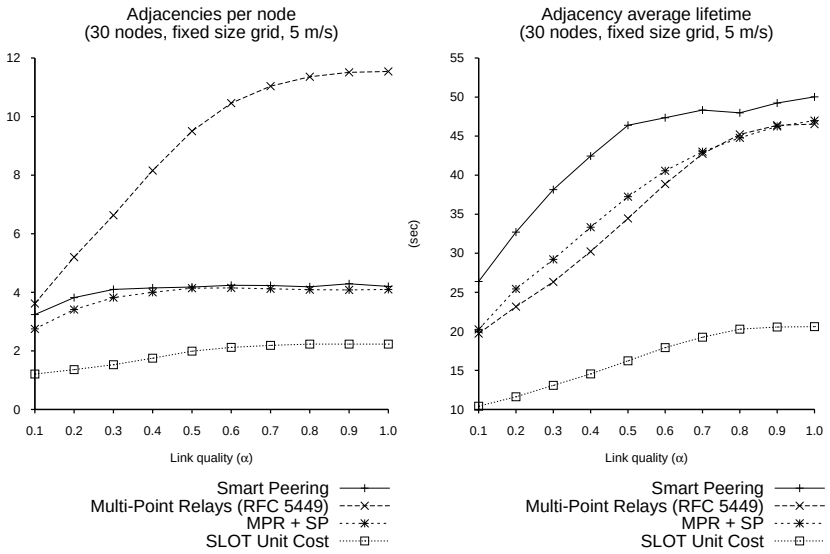


Figure 12.9: (a) Average number of adjacencies per node, and (b) average adjacency lifetime (30 nodes, fixed size grid, 5 m/s).

Figure 12.10 illustrates the effect of the keep-alive configuration in the adjacency lifetime value. It shows the adjacency lifetime achieved with MPR+SP in normal conditions (keep-alive only based on Hello reception), and the value achieved with the same configuration, when Link State Update (LSU) packets are used as keep-alives together with Hellos.

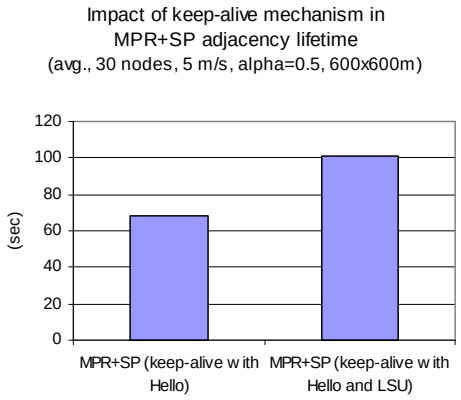


Figure 12.10: OSPF keep-alive (*InactivityTimer*) impact in adjacency lifetime.

12.4.4 Control & Total Traffic

The control traffic used in every extension is displayed in Figure 12.11, together with the overall traffic (Figure 12.12). The control traffic has two main components: traffic dedicated to adjacency-forming processes, which depends on the synchronized overlay, and reliable flooding. Figure 12.11 shows that the analyzed MPR-based extensions can be grouped in three categories:

- extensions using MPR for flooding, topology selection and synchronization purposes (MPR-OSPF),

- extensions using MPR only for flooding and topology selection (SLOT-OSPF and MPR+SP), and

- extensions using techniques other than MPR for the three link-state operations (Overlapping Relays, which relies on Smart Peering).

The results indicate that, while MPR flooding yields in general a better performance (in terms of overhead) than other flooding overlays, the use of MPR for LSDB synchronization purposes has significant shortcomings in terms of overhead – a conclusion that is consistent with chapter 8. Therefore, extensions exploring less dense overlays for synchronization, while keeping MPR as the reliable flooding overlay, obtain more balanced trade-offs between flooding quality and control traffic overhead.

Figure 12.12 shows the impact of suboptimal routing in the total traffic load handled by the network. When the data traffic load injected into the network is significant ($1Mbps$ in the figure), extensions reducing control overhead at the expense of route suboptimality, such as OR / SP without unsynchronized adjacencies, need to handle a higher amount of overall traffic than extensions computing optimal routes (MPR-OSPF, MPR+SP and SLOT-OSPF).

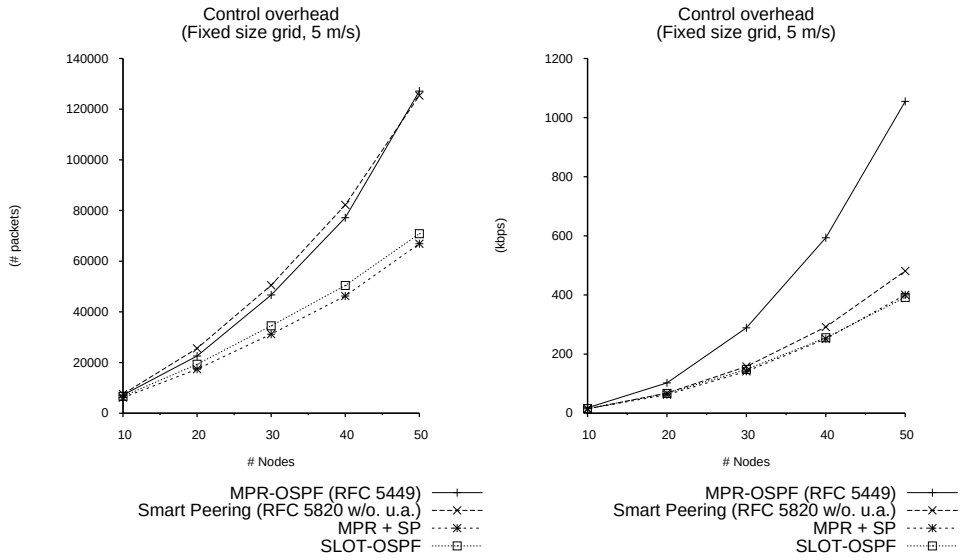


Figure 12.11: **(a)** Control traffic overhead, in number of packets, **(b)** in kbps ($5m/s$).

12.5 Persistency Impact on MPR-OSPF

The notion of link *persistency* in overlays was introduced and developed in sections 8.4 and 11.3. While first applied to the synchronized overlay based on MPRs, this principle could be applied to any overlay technique, and any link-state operation. This section provides an evaluation of the impact of persistency on the different link-state operations (topology flooding, topology selection and LSDB synchronization) in MPR-OSPF. Four different variations of MPR-OSPF (see table 11.1) are evaluated: PMM (standard MPR-OSPF, as is specified in RFC 5449), PMP, PPM and MMM (non-persistent variation). This latter variation is included for reference.

Implementation of the persistent approach in any of the link-state operations in OSPF –link synchronization, LSA flooding, route construction– necessarily implies an increment of the density (number of links) of the corresponding overlay. This section aims to identify the cost of such density increase for the different link-state overlays.

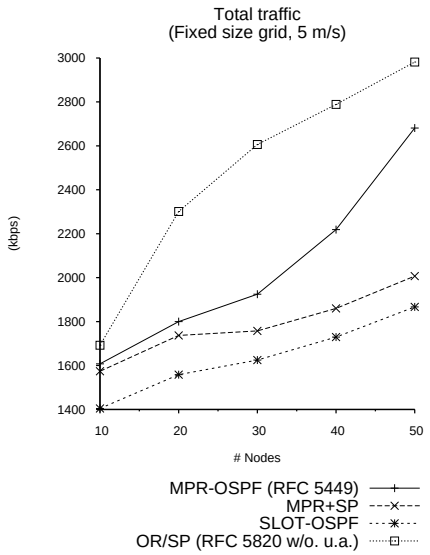


Figure 12.12: Total (data + control) traffic, in kbps ($5m/s$, with $1Mbps$ of data traffic load).

12.5.1 Persistent Adjacencies and Data Routing Quality

Figure 12.13 shows the data delivery ratio and end-to-end delay for each variation. Persistent variations (PMM, PMP and PPM) present a significantly better behavior than MMM. Moreover, Figure 12.13.b shows that the performance of RFC 5449 [24], which corresponds to PMM, can be still improved in terms of delivery ratio by implementing the persistency principle in OSPF operations other than LSDB synchronization, as PPM and PMP do.

The improvement of routing quality shown in Figure 12.13 in persistent variations is at the expense, first of all, of a substantial growth in the number of links contained in the synchronized overlay, as shown in Figure 12.14.a. Figure 12.14.b shows the gap between persistent adjacencies, with significantly longer lifetime, and non-persistent ones.

The difference in the size (in number of links) of synchronized overlays is more significant as the network density increases: in a 50 nodes network, for the same grid dimensions ($400 \times 400m$), about 80% of the adjacencies are persistent. The cost of such persistent adjacencies is extremely low in terms of control traffic exchange. These links became persistent after the exchange of local instances of LSDB had completed, and thus can be conserved with very little additional overhead,

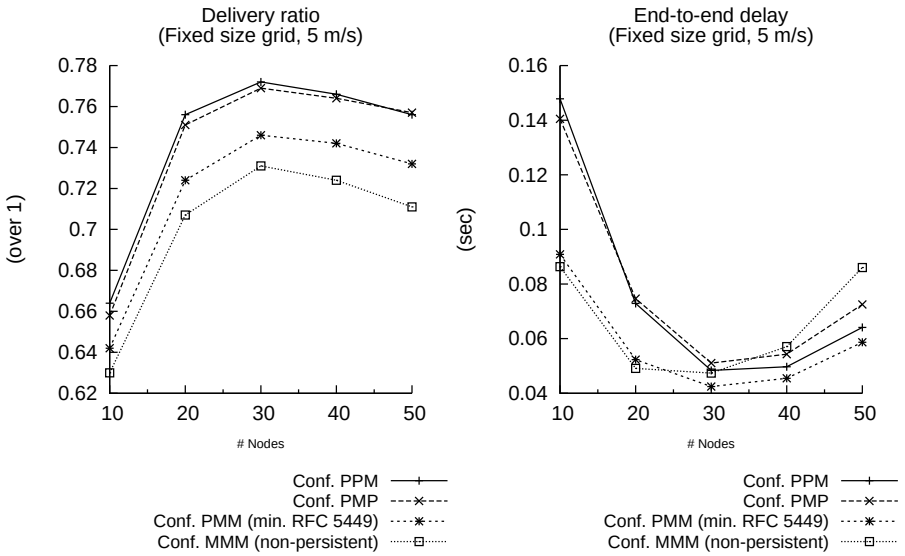


Figure 12.13: **(a)** Delivery ratio, and **(b)** End-to-end packet delay (5 m/s).

corresponding to the acknowledgements to LSA transmissions over adjacencies. While forming an adjacency is a costly process (in terms of overhead) for ad hoc networks, mainly due to the rigid conditions in which local instances of LSDB are exchanged, to maintain such adjacency when it is no longer an MPR link has almost no cost, as explained in section 8.4.

12.5.2 Control Traffic Structure

The amount of OSPF control traffic and the structure of such control traffic traffic –*i.e.*, the presence of the different types of OSPF packets– vary significantly for MMM, PMM, PMP and PPM.

Figure 12.15.a shows that persistent variations, with the exception of PPM, require less control overhead than MMM. Comparison between PMM and MMM indicates that the use of persistency for LSDB synchronization reduces the amount of control traffic required for performing link-state operations. As adjacencies are not removed from the synchronized overlay when they are no longer MPR links, additional LSDB synchronization processes in such links are not necessary when any of the attached interfaces select the other again as MPR.

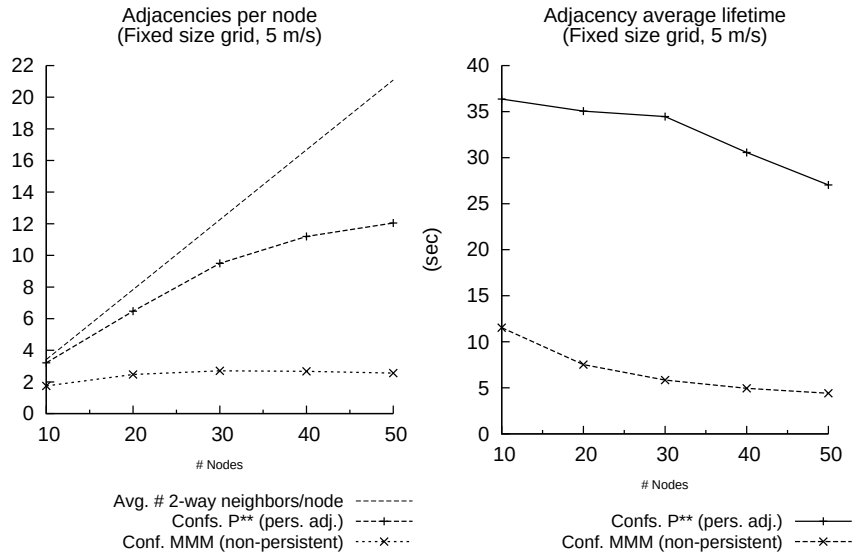


Figure 12.14: **(a)** Average number of adjacencies per node and **(b)** average adjacency lifetime (5 m/s).

PPM has the highest amount of control traffic, as a consequence of performing reliable flooding over all synchronized adjacencies – including persistent adjacencies. Other variations (MMM, PMP and PMM) perform flooding only over the MPR flooding overlay (see section 11.2.1). The use of an overlay that contains a higher number of links, as Figure 12.7.a points out, enables PPM to achieve a significantly better quality in flooding, with a lower LSA retransmission ratio (Figure 12.15.b) than other variations.

This is at the expense, however, of requiring the highest amount of control traffic overhead among all variations. This is caused by the increase of flooding control traffic (LSUpdate packets sent via multicast). According to Figure 12.16.a, the number of packets flooded in PPM triples the number of packets in other variations (PMP, PMM and MMM).

Figure 12.16.b shows the amount of synchronization control traffic (DBDesc packets, LSReq packets and LSUpdate packets sent via unicast) under each variation. Variations with low levels of flooding traffic are as well those with more significant amounts of synchronization control traffic, and vice versa. Figure 12.15.a indicates that the use of persistency in the topology selection (PMP) produces a less significant impact on the overall control overhead than the use of persistency in flooding

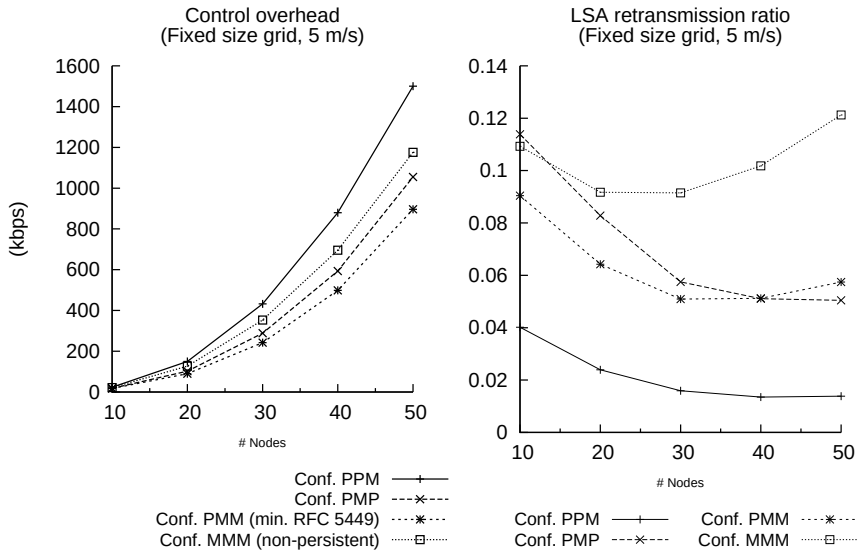


Figure 12.15: **(a)** Total control overhead, **(b)** LSA retransmission ratio (5 m/s).

(PPM). Both PMP and PPM achieve, however, equivalent levels of delivery ratio (Figure 12.13) and keep reasonable flooding quality values (in terms of LSA retransmission ratio, below 10%, see Figure 12.15.b). These results suggest the existence of a trade-off between control traffic dedicated to synchronization and LSA flooding via multicast, for similar levels of routing quality (data delivery ratio).

12.6 Conclusion

This chapter has presented the results of a simulation-based evaluation of the performance of MPR-based OSPF extensions for MANETs. The analysis has focused on the impact in MANETs of the different approaches on the role of adjacencies, the link hierarchy or the properties of data traffic routes provided by the different OSPF MANET extensions. Some additional aspects were also explored, such as the neighbor sensing optimization techniques and the impact of persistency mechanisms. Among the results obtained from the simulations, the following aspects can be highlighted:

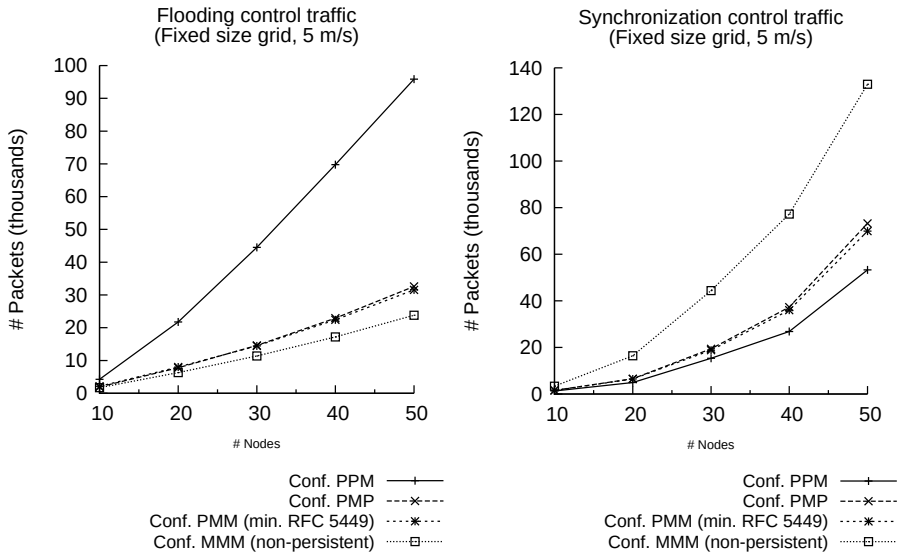


Figure 12.16: **(a)** Flooding control traffic (LSUpdate packets via multicast), and **(b)** Synchronization control traffic (DBDesc packets, LSReq packets and LSUpdate packets via unicast), in number of packets (5 m/s).

- The two main principles of OSPF routing (data traffic over shortest paths, data and control traffic over adjacencies) can be preserved in MANETs – this is the approach of MPR-OSPF. But maintaining a synchronized overlay that contains the flooding overlay, as well as sufficient links for computing network-wide shortest paths, requires a high amount of control traffic, partly caused by LSDB synchronization processes. This overhead can be reduced, without affecting the routing performance, by performing reliable LSA flooding and topology selection operations in different overlays not necessarily contained in the synchronized overlay.

- In terms of topology selection, topology information disseminated over the network should be sufficient to enable each interface to compute accurate shortest paths to any destination in the AS. Performed simulations indicate that the amount of data traffic increases and the quality of routing (in terms of data delivery ratio) decreases significantly if this condition is not satisfied. Moreover, routing quality also improves when all synchronized links are taken into account in shortest paths computation – this way, routing loops caused by inconsistent routing decisions are less probable.

- In terms of flooding, the results show that excessive pruning of the flooding overlay in order to optimize the control overhead may lead to insufficient coverage of the network and may also cause an increase of control traffic due to the need for more retransmissions. The Multi-Point Relay (MPR) technique enables routers to perform efficient flooding over the network, but such performance is only possible if the relays selected by an interface make reachable in two hops all 2-hop neighbors of such interface – and not only part of them.

- In all the OSPF MANET extensions, flooding is performed in a reliable fashion as in OSPF. While the acknowledge mechanism is costly and may cause unnecessary transmissions in MANET conditions, attempts to involve routers other than transmitter and receiver in the acknowledge operation (for instance by establishing a 2-level acknowledgment scheme) may increase the number of required retransmissions without improving substantially the overall quality of the protocol.

Together with the evaluation of the MPR-based extensions, the chapter has also examined some additional issues that are not specific of any single extension:

- Two techniques have been proposed for reducing the traffic involved in neighbor sensing. Such reduction is performed by only announcing in Hello information newer than the last information transmitted in the previous Hello transmission. Drawbacks of these techniques are related to the impact of a single Hello packet loss. Beyond such drawbacks (Chakeres [86]), performed simulations indicate that these techniques yield little benefit in terms of overhead and may become counterproductive in mobile ad hoc scenarios.

- The persistency mechanism, proposed in MPR-OSPF (RFC 5449) for adjacencies, proves to be efficient for stabilizing an overlay with high link change rates, as it is the case for the MPR synchronized overlay. Improving stability of links of the synchronized overlay is equivalent to reducing the rate of completed adjacency-forming processes, with the consequent reduction of LSDB synchronization control traffic. The persistent strategy shows also good properties when applied in MPR-OSPF for topology selection purposes. Topology flooding over all adjacencies

–persistent adjacencies included–, however, causes a substantial growth in the control traffic without improving significantly the quality of the flooding operation.

Chapter 13

Experiments with OSPF on a Compound Internetwork Testbed

Previous chapters have presented simulation-based results about the performance of different OSPF MANET extensions in Mobile Ad hoc Networks. The simplifications provided by network simulation permit to study the performance of different mechanisms of the evaluated extensions in a wide range of conditions of router mobility and population. However, experiments over a real testbed are needed to fully validate OSPF operation in compound internetworks, in order to go beyond assumptions and simplifications from theoretical models and simulation-based analysis.

This chapter documents the experiments performed on a testbed consisting in an internetwork composed of 6 computers that form a static topology – computers do not move during network lifetime. OSPFv3 is used as a routing protocol in the internetwork: wired interfaces run OSPF as specified in RFC 2328 [107] and RFC 5340 [28], wireless interfaces are configured as MANET interfaces as specified in RFC 5449 [24]. The chapter focus on the effect of wireless links in the routing quality of multi-hop communication and the structure of OSPF control traffic.

13.1 Outline

Section 13.2 describes the characteristics of the deployed testbed, the topology of the internetwork and the routing configuration of the attached computers' interfaces. Performed experiments are described in section 13.3, together with the obtained results and a discussion about their implications. Finally, section 13.4 concludes the chapter.

13.2 Testbed Description

This section describes the characteristics of the employed networking testbed. Section 13.2.1 presents the distribution of computers in the testbed and the network topology that they form. Section 13.2.2 details the implications of such topology in OSPF routing.

13.2.1 Interfaces Configuration and Network Topology

The testbed is composed of 6 computers (routers/hosts) attached to two interconnected networks: a wired network and a wireless network. Table 13.1 indicates the network interfaces of each computer. For more details about computers' hardware, see Appendix F.

Computer	Abbr.	Wired ifs.	Wireless ifs.
`server`	S	eth0, eth1	–
`hybrid1`	h_1	eth0	wlan0
`hybrid2`	h_2	eth0	wlan0
`wless1`	w_1	–	wlan0
`wless2`	w_2	–	wlan0
`wless3`	w_3	–	wlan0

Table 13.1: Network interfaces of testbed computers.

Physical Topology

The internetwork connecting these computers was deployed in the Computer Science Lab (*Laboratoire d'Informatique*, LIX) of École Polytechnique, in Paris (France). Three scenarios –I, II

and III– were configured over the resulting internetwork. These scenarios permit to test the communication between computers `wless3` and `server`, for different situations. The physical distribution of computers at LIX is displayed in Figure 13.1.

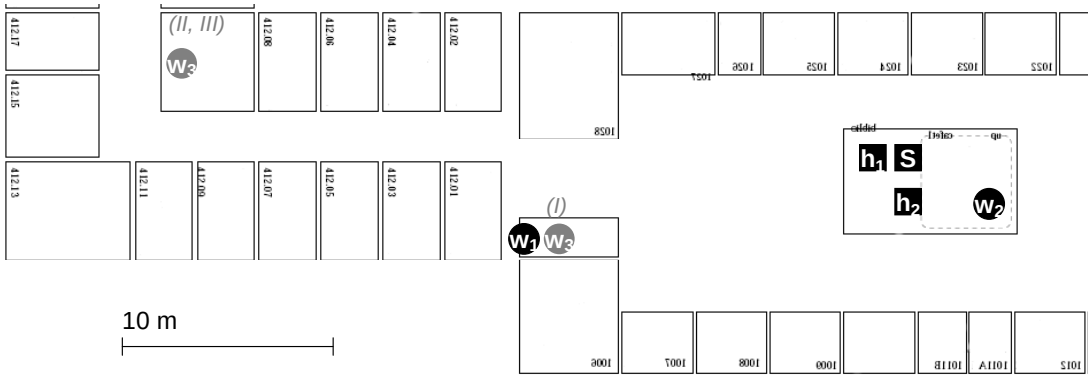


Figure 13.1: Computers position over the plan of LIX.

Positions of computers do not change, except for the case of `wless3`, which has a different position for scenario I and for scenarios II and III, as shown in Figure 13.1.

Logical Internetwork Topology

Each scenario corresponds to a specific internetwork topology. Figure 13.2 indicates the internetwork topology graphs for scenarios I, II and III. In the wired network, computers communicate through the IEEE 802.3 (Ethernet) standard protocol, `server` is connected with `hybrid1` by way of interface eth0 and with `hybrid2` by way of interface eth1, as shown in Figure 13.2. In the wireless network, interfaces communicate through the IEEE 802.11b WLAN standard protocol, and all wireless routers (`hybrid1`, `hybrid2`, `wless1`, `wless2` and `wless3`) use their wireless interface wlan0. The topology that results from wireless reachability among computers `hybrid1`, `hybrid2`, `wless1`, `wless2` and `wless3` is modified by means of MAC filtering in order to disable links $h_1 \longleftrightarrow h_2$, $w_{1,3} \longleftrightarrow w_2$ and $w_{1,3} \longleftrightarrow h_2$. In scenario III, link $w_2 \longleftrightarrow h_1$ is suppressed by disabling interface `wlan0` at computer `hybrid1`.

The use of MAC filtering for suppressing links implies that the filtered traffic is not visible

for upper layers of the wireless and hybrid routers. This filtered traffic, however, has an impact in the performance of network communication, as it requires energy consumption at transmission and reception and may cause, for instance, packet collision or interference. The quality and the capacity of wireless links is therefore underestimated in upper layers when part of the traffic is discarded at the MAC layer. As the links suppressed by way of MAC filtering are the same in the three considered scenarios (I, II and III, see Figure 13.2), this underestimation is equally presented in all scenarios and therefore it does not invalidate the qualitative trends and conclusions drawn from the experiments.

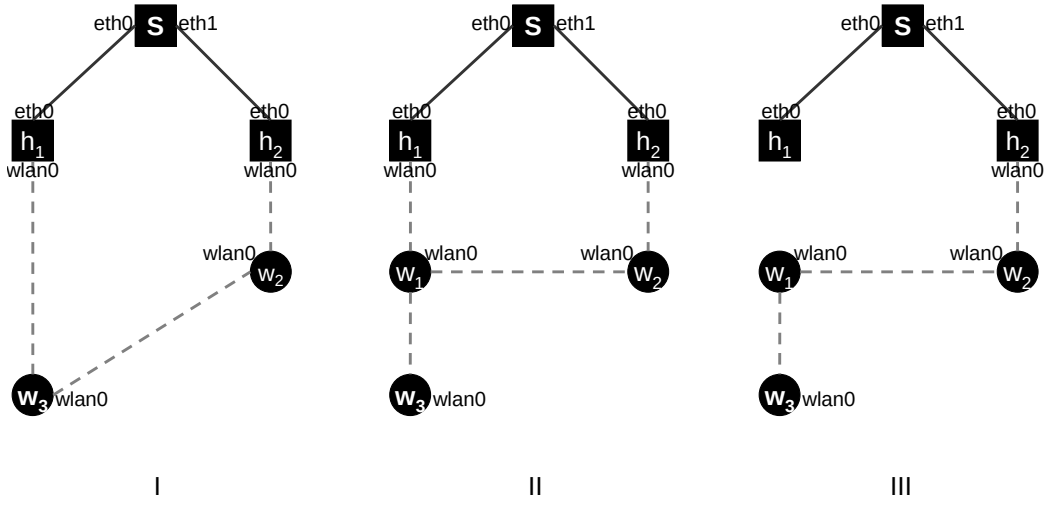


Figure 13.2: Considered topologies for scenarios I, II and III.

13.2.2 OSPF Routing Configuration

All interfaces use the extended OSPFv3 routing protocol, wired and wireless interfaces using different interface types. Wired interfaces are configured as *point-to-point interfaces*, as they are specified in RFCs 2328 [107] and 5340 [28]. Wireless interfaces are configured as *MANET interfaces*, as specified in the MPR-OSPF MANET extension for OSPF (RFC 5449 [24]).

OSPF Adjacencies and MPRs

According to the specification of OSPF and MPR-OSPF extension, all links in any of the considered topologies for scenarios I, II and III are adjacent. Within the wired network, every point-to-point link is an adjacency. In the wireless network, wireless links are adjacent if they are MPR links (def. 8.2). The list of MPRs of every wireless interface, for each scenario, is displayed in Table 13.2.

Interface	I	II	III
`hybrid1:wlan0`	w_1	w_1	–
`hybrid2:wlan0`	w_2	w_2	w_2
`wless1:wlan0`	–	w_2	w_2
`wless2:wlan0`	w_3	w_1	w_1
`wless3:wlan0`	w_2	w_1	w_1

Table 13.2: MPRs selected by each wireless interface, for each scenario.

It can be observed that all links are MPR links, and therefore all are declared adjacent. In this topology, the presence of a *synch* router (see section 11.2.1) is thus redundant.

OSPF Flooding

Flooding in the wired network is performed through adjacent links – that means, $S \longleftrightarrow h_1$ and $S \longleftrightarrow h_2$. In the wireless network, flooding is performed:

- through the MPR links (from a wireless router towards its MPR), and
- through all links connecting an interface to a hybrid router (`hybrid1` and `hybrid2`).

13.3 Experiments and Results

For each scenario (I, II and III), communication between `wless3` and `server` is tested by way of two experiments. Displayed results show the averaged measures over tens of samples (see Appendix F for further details on configuration of the experiments):

- Transmission of ICMPv6[1] requests (*ping*) from `wless3` to `server`. The measure of time between the transmission of an ICMP request and its reply corresponds to the Round Trip Time (RTT) of the *ping* through the evaluated path.

- Transmission of a constant bit rate data UDP flow from `wless3` to `server`. Comparison between packets sent and packets received permits to test the quality of the traversed paths and the wireless links that compose them in each scenario. Characteristics of these UDP flows are summarized in Table 13.3.

Nominal sender bit rate	100 pkts/s
Packet payload	1024 bytes
CBR traffic rate	300 kbps
Flow duration	5 min/flow

Table 13.3: Characteristics of transmitted UDP flows.

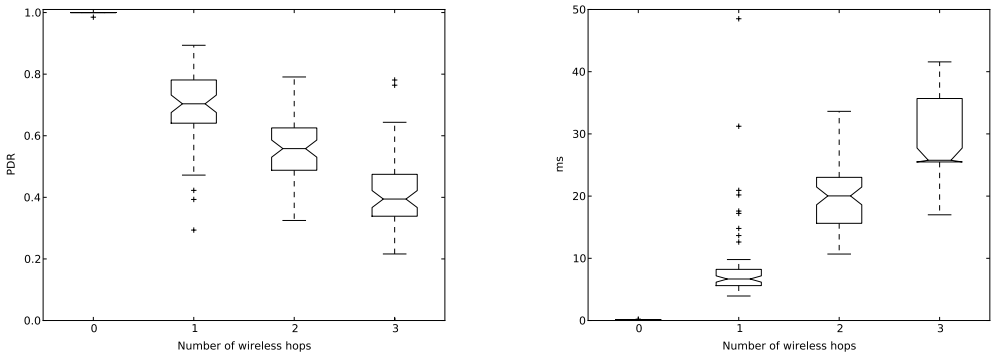
The three considered scenarios are complemented by another scenario in which information is transmitted and measured through the wired link $h_1 \longleftrightarrow S$. Results on this scenario are added for completeness and reference. Section 13.3.1 presents the results obtained in both experiments, for each scenario, in terms of quality of wireless links. Section 13.3.2 examines the amount and structure of control traffic used in OSPF for enabling routing of packets within the internetwork.

13.3.1 Wireless Multi-hop Communication

Figures 13.3.a and 13.3.b display the results of the performed experiments, in particular the delay for ICMP requests (*pings*) and the packet delivery ratio of CBR UDP data flows.

Both Figures 13.3.a and 13.3.b indicate the degradation of the quality of communication between routers `wless3` and `server` as the number of wireless links between them increases. As expected, the wired link $h_1 \longleftrightarrow S$ has an almost-ideal behavior: 100% PDR and no significant delay. The negative impact of wireless links in the path from source to destination is close-to-linear with the number of traversed wireless links, as shown in Figure 13.3.a: more than 30% of transmitted

[1]Internet Control Messaging Protocol for IPv6, RFC 4443 [42].



(a) Packet Delivery Ratio (PDR)

(b) Round Trip Time (RTT)

Figure 13.3: Box-and-whiskers plots for Packet delivery ratio (PDR) of UDP flows and Round Trip Time (RTT) of ICMP requests, both depending on the number of wireless hops.

packets are lost in the first wireless link, and such percentage increases about a 15% per additional wireless link included in the path. Figure 13.3.b shows that such degradation is also evident in terms of round trip time (RTT). Replies to ICMP requests are immediately delivered through a wired link, but the average and the variation of delays grow with the number of wireless links involved – is in the order of tens of miliseconds for 2 and 3 wireless links.

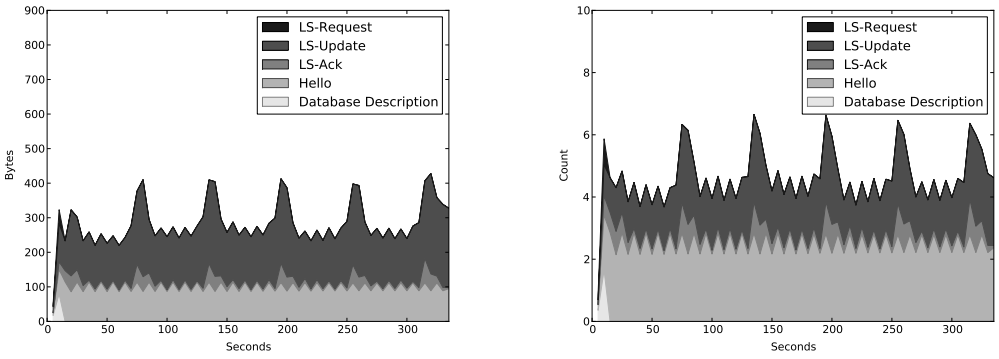
While the degradation due to the use of wireless links depends on the specific topology and the network technology that is used, these experimental results suggest that selection of accurate shortest paths is essential in wireless networks, as additional wireless hops in the route of data packets in the network imply a significant degradation of the quality of communication between the involved computers.

While the impact on communication due to the use of wireless links depends on the specific topology and the network technology that is used, two conclusions can be drawn from these experimental results. As each additional wireless hop in the route of data packets in the network implies a significant degradation of the quality of communication, routing in wireless networks should preserve the principle of shortest (wireless) paths, meaning that the number of wireless links traversed by data packets should be minimized. In the context of compound internetworks with wired and wireless links, such minimization implies that wired links should be used when available, even at

the expense of increasing the number of hops of the overall path through the internetwork, as wired links provide a significantly better quality than wireless ones. Metrics for compound internetworks should thus be able to take into account not only the length (in hops) of an internetwork path, but also the presence of wireless and wired links.

13.3.2 OSPF Control Traffic Pattern

Figures 13.4, 13.5, 13.6 and 13.7 display the evolution of OSPF control traffic transmitted by wireless interfaces `wless3:wlan0` and `hybrid1:wlan0`, on one side, and wired interfaces `hybrid1:eth0` and `server:eth0`, on the other. The five packet formats used in OSPF (Hello, LSUpdate, LSRequest, LSAck and DBDesc, see section 10.2.3) can be distinguished in these figures. Measures were taken with the topology of scenario I, each point corresponding to the number of packets or bytes sent within an interval of 5 seconds. The traffic load of the internetwork was composed of a CBR UDP data traffic flow from `wless3` towards `server` (see Table 13.3 for details), and OSPF control traffic. The figures show the structure of such control traffic, both in terms of number of packets and number of bytes, during the first 335 seconds of network operation, *i.e.*, after routers' startup. All interfaces are configured with the same OSPF parameters, in order to facilitate the comparison between control traffic patterns of each of them. See Appendix F for further details.



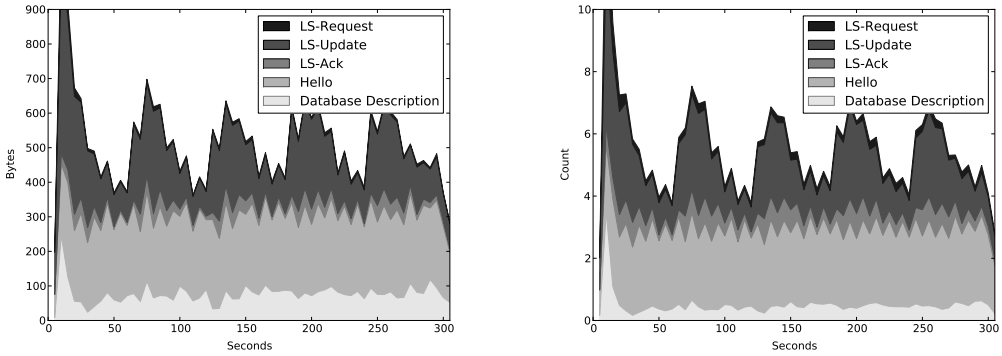
(a) Bytes

(b) # Packets

Figure 13.4: Control traffic overhead at `server:eth1`.

Hello Packets

The amount of Hello packets sent by each interface is kept constant along the monitored time. As $HelloInterval = 2sec$, interfaces transmit 2.5 Hello packets per interval of 5 seconds. The length of Hello packets is significantly longer in wireless interfaces (Figures 13.7.a and 13.5.a) than in wired interfaces (Figures 13.4.a and 13.6.a). For the same number of neighbors, Hellos from `hybrid1:eth0` have 40 bytes while those from `hybrid1:wlan0` have 75.34 bytes. This is due to the fact that Hello packet format in RFC 5449 [24] includes additional information about link costs, adjacencies and MPR selection, which is added to the format specified in OSPF [107] and OSPFv3 [28].



(a) Bytes

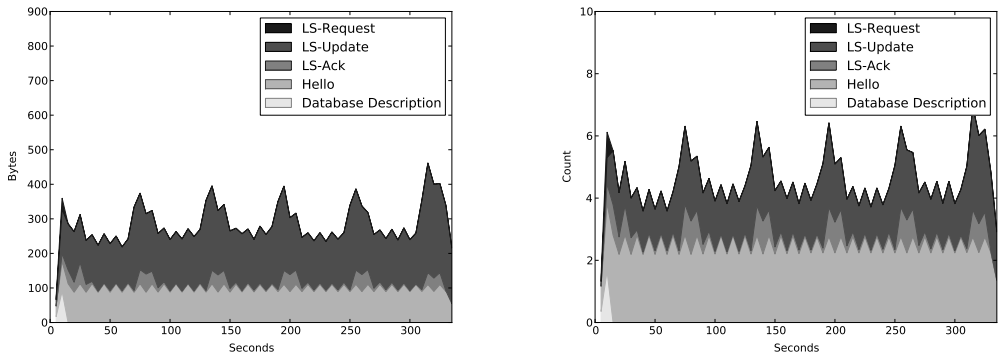
(b) # Packets

Figure 13.5: Control traffic overhead at `wless3:wlan0`.

LSDB Synchronization

The existence of ongoing LSDB synchronization processes during the monitored time interval can be noticed in the OSPF control traffic structure by way of the presence of Database Description (DBDesc) packets. The fact that such packets are only present, for wired interfaces, in the first part of the monitored interval (from $t = 0sec$ to $t = 10sec$, as shown in Figures 13.4 and 13.6) indicates that links become synchronized only when the routers are switched on. In contrast, DBDesc are transmitted in the whole monitored interval for wireless interfaces. This is consistent

with the fact that wired links are mostly stable and therefore there is no need to repeat synchronization process after the first LSDB exchange. Wireless links, in contrast, are more prone to packet losses and link failures, and need thus to be synchronized several times during the network lifetime, even in the absence of router mobility. The same phenomenon can be observed with LSRequest packets, which can only be sent during the last phase of the LSDB synchronization process, when the synchronizing neighbor stays in *Loading* state after having exchanged its local instance of LSDB (see Figure 10.7 in section 10.2.2).



(a) Bytes

(b) # Packets

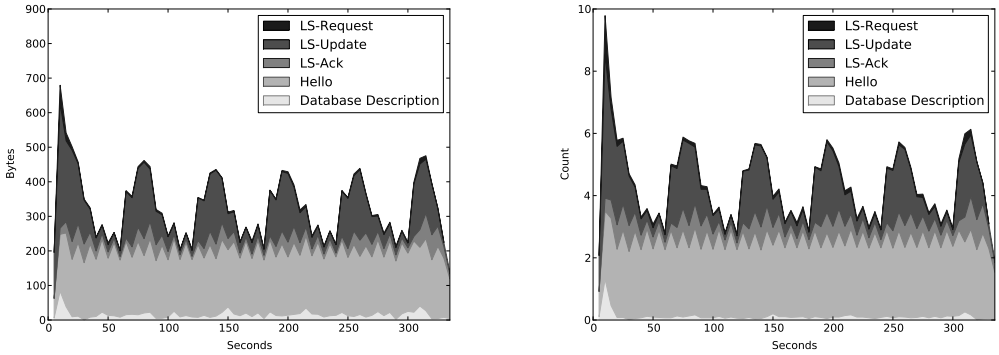
Figure 13.6: Control traffic overhead at `hybrid1:eth1`.

Link State Updates, Requests and Acknowledgements

LSUpdate packets contain one or more Link State Advertisements (LSAs). Such LSAs can be either originated by the sending interface, either originated by another interface and flooded (forwarded) by the sending interface. Transmission of LSUpdate packets follows a common pattern in all the interfaces in the internetwork. Thus pattern consists of periodic peaks followed by time intervals (*valleys*) in which the number and size of LSUpdate transmissions is lower and roughly constant.

The time interval between two consecutive peaks corresponds to the value of parameter $LSRefresh$, set to $60sec$ for all interfaces. This is the time interval at which an interface floods its

topology description (periodically) if there are no topology changes in the meanwhile.



(a) Bytes

(b) # Packets

Figure 13.7: Control traffic overhead at `hybrid1:wlan0`.

Despite the common pattern in the LSUpdate traffic, several differences can be observed between wired and wireless interfaces. This section concentrates on three particular aspects: peak width, height of valleys between consecutive peaks and transient state (after routers are switched on).

- **Transient state.** Immediately after switching on, wireless interfaces transmit a high number of packets – mostly, LSUpdate packets sent in response to LSRequest packets received during the first LSDB synchronization processes in all wireless links (Figures 13.7.a and 13.5.a, between $t = 0sec$ and $t = 50sec$). This amount of transmissions involves traffic rates above $220Bps$ ($1.1kB$ per interval of $5sec$), then decreases and stabilizes in a slightly lower level (maximum peak of $130Bps$). The opposite behavior is found in wired interfaces (Figures 13.4.a and 13.6.a), in which the initial transient period of low LSUpdate traffic rate (about $26Bps = \frac{130B}{5sec}$ for `hybrid1:eth0`) is followed by a steady period in which the minimum LSUpdate rate is slightly higher (about $30Bps = \frac{150B}{5sec}$ for `hybrid1:eth0`). These different behaviors can be explained by the different roles that flooding has over wired and wireless links. Due to their stability, packets sent over wired links are mostly forwarded packets – that is, they come from other interfaces than those involved in the links. In the first instants in which there is

no flooding over the network because adjacencies have not been formed in the network and flooding links have not yet been identified, the overall traffic traversing such wired links is temporarily low. The opposite is observed in wireless links.

- **Peak width.** Peaks are narrower in wired interfaces ($\sim 10sec$ for `server:eth1`, $\sim 15sec$ for `hybrid1:eth0`) than in wireless interfaces ($\sim 25sec$ for `hybrid1:wlan0`, $\sim 30sec$ for `wless3:wlan0`). For interfaces attached to wireless links, there is a high probability that a topology change causes a new topology update before the $LSRefresh$ interval – therefore, intervals between consecutive transmission of interfaces' topology descriptions are shorter than $LSRefresh$ and the width of the peak increases. In stable wired links, in contrast, intervals between consecutive transmissions are closer to the $LSRefresh$ parameter and, therefore, LSUpdate transmission events are less spread in time.

- **Height of valleys.** Besides the peaks caused by transmission of its own topology description, either periodic or as a reaction to a topology change, two other events may lead an interface to transmit Link State Advertisements (LSAs): (i) forwarding of LSAs originated by other interfaces in the internetwork, and (ii) retransmission of LSAs not acknowledged by their intended destinations. Both additional events explain the presence of valleys with significant traffic rate, *i.e.*, a non-zero minimum level of LSUpdate transmissions in the monitored interfaces. In wired (reliable) links such as `server:eth1` and `hybrid1:eth0`, such transmissions are caused by flooding, and involve about $25Bps$ ($127B$ per interval of $5sec$). Wireless interfaces such as `wless3:wlan0` have a minimum LSUpdate transmission rate of about $16Bps$ ($80B$ per interval of $5sec$) caused by LSA retransmissions and flooding.

13.4 Conclusion

Results from the performed experiments over the testbed confirm that wireless links are significantly slower and less reliable than wired links. Moreover, the quality of routes over wireless links becomes worse as the number of wireless links included increases. Degradation of communi-

cation due to wireless links implies that suboptimal paths should be avoided in routing on ad hoc networks, as the quality of resulting communication may get severely damaged. For internetworks combining wired and wireless networks, shortest paths need to be computed over the whole internetwork, taking advantage of all available links in the internetwork. These effects encourage the use of a single routing protocol in the whole internetwork. Scenario III permits to illustrate the case of an internetwork in which the presence of different routing protocols in wired and wireless networks prevents the use of the link $h_1 \longleftrightarrow w_1$ – and restricts interaction between wired and wireless networks to router `hybrid2`, that needs to become a *gateway* in the sense of def. 3.5.

The analysis of the pattern of OSPF control traffic over the wireless network also illustrates the differences between wireless and wired links and points out some particular aspects of control traffic over wireless links. Even with a very small number of neighbors per wireless interface, link synchronization processes may involve an excessive amount of traffic, in particular if they require the exchange of LSUpdate packets. For static internetworks such as the one deployed in the testbed, link synchronizations that involve LSUpdate packets are only the first ones performed in the by wireless interfaces. For mobile ad hoc networks and compound Autonomous Systems, however, these synchronizations may be present during the whole network lifetime. While the evolution of control traffic in wireless interfaces highly depends on the testbed characteristics and the employed OSPF MANET extension, the cost of LSDB synchronization in terms of traffic overhead confirms that the number of synchronized links should be minimized in OSPF routing for ad hoc networks.

Conclusions

This manuscript has addressed the problem of routing in compound Autonomous Systems in the Internet, *i.e.*, Autonomous Systems that contain (mobile) ad hoc networks interconnected with fixed networks.

Research on ad hoc networks, their properties, the requirements to enable communication in such networks and their applications has been very intense from the 1990s, in particular since the IETF defined the concept of MANET in 1997. This research has mostly focused on isolated MANETs, *i.e.* mobile ad hoc networks considered on their own, and MANETs belonging to the *edge* of the Internet, *i.e.* MANETs only able to handle Internet traffic when either the source or the destination is in the MANET.

As wireless and mobile access to the Internet becomes more common (see Figure 13.8), however, wireless ad hoc networking needs to be studied in the framework of the Internet, increasingly based on the interconnection of fixed and ad hoc networks, wired and wireless links. This approach is explored in this manuscript. Instead of focusing on MANETs outside or in the edge of the Internet, it examines the coexistence of wired and wireless dynamic links in the core of the Internet. The manuscript thus addresses the problem of enabling the full exploitation of communication capabilities provided by wireless ad hoc networks in such core, in the interior of Autonomous Systems able to relay traffic from external sources towards external destinations in the Internet.

This is done by using a single routing protocol for compound ASes. The manuscript addresses the extension and enhancement of interior gateway protocols (IGPs) in order to to use them for routing inside compound ASes. As main IGPs are link-state routing protocols, the manuscript

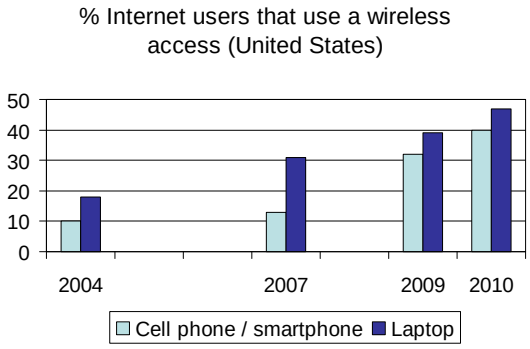


Figure 13.8: Wireless access to the Internet in the United States (source: Pew Internet & American Life Project). *Percentages are not exclusive.*

focuses on such routing technique.

There are several reasons for this approach, as it has been detailed throughout the manuscript. By using a single protocol for routing in an AS, management and routing maintenance of the AS is simpler, as no gateways between the different existing protocols are required. In the case of link-state routing, a single protocol ensures the optimality of computed routes contained in the AS topology. Moreover, the fact that protocols already used in ASes are extended for MANET operation permits that such ASes are able to handle ad hoc networks without requiring fundamental changes in their routing architecture.

Summary of Contributions

The contributions of this manuscript can be summarized in two main categories: the optimization of link-state operations for MANET operation, addressed in Part II; and the extension of OSPF so that extended OSPF can be used for routing in compound ASes, addressed in Part III of the manuscript.

Optimization and Analysis of Link-State Operations

Link-state routing over a network requires that three link-state operations are performed by routers attached to such network: (i) description of the network topology, (ii) flooding of such topol-

ogy information over the network, and (iii) synchronization, exchange and update of the topology information stored in neighboring routers. The manuscript examines the properties and requirements of these three link-state operations in MANETs, by way of an overlay-based analysis in which each link-state operation is associated with an overlay. This analysis shows that the constraints imposed by ad hoc networks over such overlays and the features that the overlays need to fulfill in order to provide support for their corresponding operations, are different and not always compatible for the different overlays. Flooding overlays need to be able to reach all routers in the network. Topology selection overlays should include network-wide shortest paths. Synchronized overlays include all routers in the network, but should optimize the stability of overlay links in order to reduce the number of LSDB synchronization processes to perform, as such processes are costly in terms of overhead. Optimization of link-state routing over MANETs, therefore, requires that the different link-state operations are performed separately and through independent overlays.

In the case of topology flooding, the manuscript focuses on the analysis of the impact of jittering in forwarding decisions. This technique was designed for reducing the probability of packet collisions in wireless networks. When applied to forwarding decisions, jittering consists in introducing a random delay before forwarding a received packet, in order to prevent collisions due to concurrent retransmissions of the same packet performed by neighboring routers. When a random delay expires, all packets waiting to be forwarded are transmitted together. The manuscript provides an analytical model to evaluate the delay in flooding introduced by the use of jitter in an interface, firstly, and the reduction in the forwarded packets rate, secondly.

The conclusions on the overlay-based analysis of link-state operations are used for proposing and evaluated different distributed overlay techniques. Three techniques are examined: the Synchronized Link Overlay Triangular (SLOT), inspired on the Relative Neighbor Graph (RNG); the Multi-Point Relays (MPR) technique; and the Smart Peering technique. MPR and SP are already used in OSPF extensions over MANETs, the MPR technique has been widely analyzed as a flooding technique. The manuscript proposes the SLOT technique and provides a theoretical evaluation of the size (number of links) of the associated overlay and the stability (lifetime) links included in such

overlay. Results indicate that the SLOT overlay over a MANET is more stable than the network graph. The manuscript also explores the use of MPR for other link-state operations, and shows that a modified MPR-based overlay can be used for topology selection purposes – for LSDB synchronization purposes, in contrast, the MPR overlay is excessively unstable. The evaluation of Smart Peering proves its correctness as a synchronization overlay and shows the ability of this technique to select stable links from a MANET. The theoretical and qualitative analysis of these three techniques is completed by way of the performance evaluation of these techniques, as part of the OSPF extension for MANETs.

Extended OSPF for Compound ASes

The second main contribution of this manuscript concentrates on the extension of OSPF for MANET operation, the final objective being to achieve an extended version of OSPF able to perform routing in compound Autonomous Systems. Taking advantage of the modular architecture of the protocol, MANET extensions of OSPF are implemented as new MANET interface types. Each of such MANET interface types is able to coexist with the other existing interface types defined in OSPF specification.

OSPF routing is based on two key elements: (i) the use of shortest paths for data packet routing, and (ii) the use of synchronized links for control traffic flooding and data traffic routing. This implies that the three link-state operations are performed in such synchronized links, denominated *adjacencies* in OSPF. Such a scheme cannot be applied "as-is" in mobile ad hoc networks because it does not scale. In consequence, the manuscript explores several extensions that adapt OSPF operation to challenges and restrictions of mobile ad hoc networks.

One of the most fundamental questions that arise when exploring extensions of OSPF is whether the very concept of *adjacency* is suitable in mobile ad hoc networks. Synchronization processes are costly in terms of overhead, and the effect of link synchronization is weakened by the short average lifetime of wireless links. In this manuscript, it is shown that the existence of adjacencies in OSPF MANET extensions and, more precisely, the presence of a synchronized

spanning overlay, is yet useful in MANETs that belong to compound ASes. The reasons for this are not only related to OSPF compatibility considerations, but also because the formation of adjacencies facilitates the efficient exchange of topology information between fixed and ad hoc networks of the same AS. The rate of links joining the synchronized overlay, however, should be minimized in order to reduce the amount of overhead dedicated to LSDB synchronization processes.

Consistently with the analysis of link-state overlays, the simulation-based evaluation of these extensions confirms that adjacencies should not be responsible for all link-state operations when OSPF routing is performed over MANETs. Instead, a separate optimization of each link-state operation is preferable.

The main elements of OSPF are therefore affected from this separate optimization. From the comparison among examined OSPF MANET extensions, the following conclusions are extracted:

(i) The use of shortest paths in routing should be preserved in MANETs. This implies that topology selection overlays need to include network-wide shortest paths. It is not required that all links in the topology selection overlay are synchronized links –*i.e.*, shortest paths may not be synchronized–, but synchronized links should be present in the computation of shortest paths over MANETs.

(ii) Flooding does not need to be performed over synchronized links. No significant differences were observed between flooding (only) over synchronized links and over a set of synchronized and non-synchronized links, as long as the flooding overlay permits a packet to reach all routers in the network.

Perspectives and Future Work

This manuscript inscribes itself in a broader effort to introduce and exploit ad hoc networking capabilities in the core of the Internet. Unlike developments in isolated networks or in the edge of the Internet, the transition from a wired networking infrastructure towards a compound internetworking infrastructure in core ASes needs to be performed gradually and preserving compatibility

with protocols already operating in current ASes. The final objective is that the Internet core is able to take advantage of all available communication capabilities, including those provided by ad hoc and mobile ad hoc networks.

Different issues arise and need to be handled in this context. The problem of defining link metrics for wireless ad hoc networks, for instance, is closely intertwined with the use of link-state routing techniques in such networks, as metrics designed for wired links (and in particular, hop count) are not sufficient for describing accurately the properties and characteristics of wireless links – which has a negative impact on the quality of shortest paths based on such metrics. This is still an open issue which has not been addressed in this manuscript. While the existence of other metrics is taken into account in the theoretical analysis of overlay techniques in Part II, the study of the presented techniques may need to be extended for particular non-trivial metrics for wireless links (*e.g.* ETX [82]).

The coexistence of wired and wireless networks in the same AS poses some additional problems in terms of link metrics. Given the qualitative differences between wired and wireless links, metrics used for computation of optimal paths in compound internetworks should enable simultaneously the use of wired links when available and wireless links where necessary, while ensuring an optimal usage of the overall existing bandwidth resources.

The full exploitation of wireless ad hoc capabilities requires further and deeper optimizations in compound Autonomous Systems. The very principle of the proactive routing techniques, the selection and use of a single shortest path between a source and a destination, may need to be adapted to constraints and features of wireless ad hoc networks. In particular, the fact that shortest paths in such networks may have a short lifetime, first, and that data packet transmissions may be overheard by routers other that their intended destinations, second. This implies that possible disruptions in the selected route may be overcome by enabling routers in the neighborhood of the last forwarder to cooperate in the packet delivery. Techniques based on similar observations have been developed for reactive routing protocols [97]; these observations may be inspiring as well for proactive and link-state routing on wireless ad hoc networks and compound ASes.

Final Remarks

The emergence and growth of the Internet has changed the way communication between users, information exchange and access to contents is conceived and performed. The development of more dynamic types of computer networks, wireless networks and mobile ad hoc networks, is causing in its turn a significant evolution in the concept and the architecture of the Internet. This is moving from a set of computer networks interconnected deterministically by way of high-capacity wired links, to a model in which such infrastructure will be reinforced with an additional layer of wireless, dynamic and non-predictable connectivity. The impact of wireless and ad hoc networking in the Internet is being observed in the *edge* of the Internet, with the use of sensor networks and MANETs in general able to connect to the Internet. This manuscript addresses the exploitation of communication facilities provided by ad hoc networks in the *core* of the Internet by way of exploring the use of protocols and techniques already existing and operating in the core of the Internet for routing in internetworks that contain MANETs. The transition towards an ad-hoc-compatible Internet core based on these techniques, would imply a significant improvement in the communication capabilities of the whole Internet, without requiring the substitution of the existing networking infrastructure.

Bibliography

[1] J. A. Cordero; M. Philipp; E. Baccelli (2011). *Compound Wired/Wireless Internetworking with OSPF*, INRIA Research Report 7642, June 2011.

[2] Various authors (2011). *Maximum posted speed limits*, Insurance Institute for Highway Safety, available at `www.iihs.org/laws/speedlimits.aspx`, last modification in February 2011.

[3] Cordero, J. A.; Baccelli, E.; Clausen, T.; Jacquet, P. (2011). Wired/Wireless Compound Networking. In: Wang, X. (*Ed.*) (2011). *Mobile Ad-Hoc Networks: Applications*, pp. 349-374, InTECH Publishers, ISBN 978-953-307-416-0.

[4] Cordero, J. A.; Clausen, T.; Baccelli, E. (2011). MPR+SP – Towards a Unified MPR-based MANET Extension for OSPF, *Proceedings of the 44th Hawaii International Conference on System Sciences (HICSS)*, pp. 1-10, Hawaii, HI (United States), January 2011.

[5] Stanica, R.; Chaput, E.; Beylot, A.-L. (2011). Broadcast Communication in Vehicular Ad-Hoc Network Safety Applications, *Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC 2011)*, pp. 462-466, Las Vegas, NV (United States), January 2011.

[6] Goyal, M.; Xie, W.; Hosseini, H.; Bashir, Y. (2011). AntSens – An Ant Routing Protocol for Large Scale Wireless Sensor Networks, *Proceedings of the 5th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA-2010)*, pp. 41-48, Fukuoka (Japan), November 2010.

[7] Various authors (2010). *Standard legal speed limits (km/h) - unless otherwise stated by traffic signs*, European Road Safety Observatory, DG Mobility & Transport, European Commission, available at `ec.europa.eu/transport/road_safety/observatory/doc/speed_rules.pdf`, last modification in October 2010.

[8] Kunz, T.; Alhalimi, R. (2010): Energy-efficient proactive routing in MANET: Energy metrics accuracy. In: *Ad Hoc Networks*, Vol. 8, Issue 7 (September 2010), pp. 755-766.

[9] Baccelli, E.; Townsley, M. (2010). RFC 5889, *IP Addressing Model in Ad Hoc Networks*, IETF, September 2010.

[10] Baccelli, E.; Cordero, J. A.; Jacquet, P. (2010). Optimization of Critical Data Synchronization via Link Overlay RNG in Mobile Ad Hoc Networks. *Proceedings of the 7th IEEE International Conference on Mobile Ad-hoc and Sensor Networks (MASS)*, pp. 402-411, San Francisco, CA (United States).

[11] Cordero, J. A. (2010). Adjacency Persistency in OSPF MANET. *Proceedings of the 4th IET China-Ireland International Conference on Information and Communication Technologies (CI-ICT)*, pp. 46-53, Wuhan (China).

[12] Cordero, J. A. (2010). MPR-based Pruning Techniques for Shortest Path Tree Computation. *Proceedings of the 18th IEEE International Conference on Software Telecommunications and Computer Networks (SoftCOM)*, pp. 240-244, Split (Croatia).

[13] Cordero, J. A.; Baccelli, E.; Jacquet, P. (2010). OSPF over Multi-Hop Ad Hoc Wireless Communications. In: *International Journal of Computer Networks and Communications (IJCNC)*, Volume 2, Number 5, pp. 37-56, AIRCC, September 2010.

[14] Baccelli, E.; Cordero, J. A.; Jacquet, P. (2010). Using Relative Neighborhood Graphs for Reliable Database Synchronization in MANETs. *Proceedings of the 5th IEEE SECON Workshop on Wireless Mesh Networks (WiMESH)*, pp. 1-6, Boston, MA (United States).

[15] Bose, P. *et al.* (2010). Some properties of higher order Delaunay and Gabriel graphs, *Proceedings of the 22nd Canadian Conference on Computational Geometry*, pp. 13-16, Winnipeg (Canada).

[16] Kacimi, R.; Dhaou, R.; Baylot, A.-L. (2010). Energy-balancing Strategies for Lifetime Maximizing in Wireless Sensor Networks, *Proceedings of the IEEE International Conference on Communications (ICC'2010)*, pp. 1-5, Cape Town (South Africa), May 2010.

[17] Baccelli, E.; Perkins, C. (2010). *Multi-hop Ad Hoc Wireless Communication*, Internet-Draft `draft-baccelli-multi-hop-wireless-communication-04`, IETF, May 2010. *(work in progress)*

[18] Clausen, T.; Dearlove, C.; Jacquet, P. *et al.* (2010). *The Optimized Link State Routing Protocol version 2*, Internet-Draft `draft-ietf-manet-olsrv2-11`, IETF, April 2010. *(work in progress)*

[19] Roy, A.; Chandra, M. (2010). RFC 5820, *Extensions to OSPF to Support Mobile Ad Hoc Networking*, IETF, March 2010.

[20] Baccelli, E.; Cordero, J. A.; Jacquet, P. (2009). Multi-Point Relaying Techniques with OSPF on Ad Hoc Networks. *Proceedings of the 4th IEEE International Conference on Sensor Networks and Communications (ICSNC)*, pp. 53-62, Porto (Portugal).

[21] Tarkoma, S. (2010). *Overlay Networks: Toward Information Networking*, Auerbach Publications, ISBN-13 978-1-4398-1373-7.

[22] Ogier, R.; Spagnolo, P. (2009). RFC 5614, *Mobile Ad Hoc Network (MANET) Extension of OSPF Using Connected Dominating Set (CDS) Flooding*, IETF, August 2009.

[23] Baccelli, E.; Clausen, T.; Herberg, U.; Perkins, C. (2009). IP Links in Multihop Wireless Networks?, *Proceedings of the 17th IEEE International Conference on Software Telecommunications and Computer Networks (SoftCOM)*, Split (Croatia).

[24] Baccelli, E.; Jacquet, P.; Nguyen, D.; Clausen, T. (2009). RFC 5449, *OSPF Multipoint Relay (MPR) Extension for Ad Hoc Networks*, IETF, February 2009.

[25] L-W. Chen; W. Chu; Y.-C. Tseng; J.-J. Wu (2009). Route Throughput Analysis with Spectral Reuse for Multi-Rate Mobile Ad Hoc Networks, *Journal of Information Science and Engineering*, pp. 1593-1604, Vol. 25, Number 1, January 2009.

[26] Kacimi, R.; Dhaou, R.; Beylot, A.-L. (2008). Energy-aware self-organization algorithms for wireless sensor networks, *Proceedings of the IEEE Global Communications Conference (GLOBECOM 2008)*, pp. 1-5, New Orleans, LA (United States), December 2008.

[27] Yick, J.; Mukherjee, B.; Ghosal, D. (2008). Wireless Sensor Network Survey, *Computer Networks*, Volume 52, Number 12, pp. 2292-2330, Elsevier, August 2008.

[28] Coltun, R.; Ferguson, D.; Moy, J. (2008). RFC 5340, *OSPF for IPv6*, IETF, July 2008.

[29] Clausen, T.; Dearlove, C.; Adamson, B. (2008). RFC 5148, *Jitter Considerations in Mobile Ad Hoc Networks (MANETs)*, IETF, February 2008.

[30] Canourgues, L.; Lephay, J.; Soyer, L.; Beylot, A.-L. (2008). A Scalable Adaptation of the OLSR Protocol for Large Clustered Mobile Ad hoc Networks, *Proceedings of the IFIP Annual International Ad Hoc Networking Conference (MED-HOC-NET 2008)*, pp. 97-108, Palma de Mallorca (Spain).

[31] Amoss, J. J.; Minoli, D. (2008). *Handbook of IPv4 to IPv6 Transition: Methodologies for Institutional and Corporate Networks*, Auerbach Publications, ISBN-13 978-0-84938-516-2.

[32] Lehsaini, M.; Guyennet, H.; Feham, M. (2007). MPR-based Broadcasting in Ad hoc and Wireless Sensor Networks with a Realistic Environment. In: *International Journal of Computer Science and Network Security (IJCSNS)*, pp. 82-89, Vol. 7, Number 10 (October 2007).

[33] Razafindralambo, T.; Mitton, N. (2007). Analysis of the Impact of Hello Protocol Parameters over a Wireless Network Self-Organization, *Proceedings of the 4th ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor and Ubiquitous Networks (PE-WASUN'07)*, pp. 46-53, Crete (Greece), October 2007.

[34] Stewart, B. D. (2007). *CCNP BSCI Official Exam Certification Guide*, 4th Edition, Cisco Press, ISBN-13 978-1-58720-147-9.

[35] Thompson, S.; Narten, T.; Jinmei, T. (2007). RFC 2462, *IPv6 Stateless Address Autoconfiguration*, IETF, September 2007.

[36] Mbarushimana, C.; Shahrabi, A. (2007). Comparative Study of Reactive and Proactive Routing Protocols Performance in Mobile Ad Hoc Networks, *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, pp. 679-684, Ontario (Canada), August 2007.

[37] IEEE-SA Standards Board (2007). IEEE Std 802.11–2007. IEEE Standard for Information Technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE, June 2007.

[38] Johnson, D.; Hu, Y.; Maltz, D. (2007). RFC 4728, *The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4*, IETF, February 2007.

[39] Clausen, T. H. (2007). *A MANET Architectural Model*, INRIA Research Report 6145, January 2007.

[40] Ingelrest, F.; Simplot-Ryl, D. (2006). Maximizing the probability of delivery of multipoint relay broadcast protocol in wireless ad hoc networks with a realistic physical layer. In: *Mobile Ad hoc and Sensor Networks, LNCS*, Springer Berlin / Heidelberg, Vol. 4325, pp. 143-154 (November 2006).

[41] Spagnolo, P.; Henderson, T. (2006). Comparison of Proposed OSPF MANET Extensions, *Proceedings of the Military Communications Conference (MILCOM'06)*, pp. 1-6, Washington DC (United States), October 2006.

[42] Conta, A.; Deering, S.; Gupta, M. (*ed.*) (2006). RFC 4443, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*, IETF, March 2006.

[43] Jacquet, P. (2006). Control of mobile ad hoc networks, *Proceedings of the IEEE Information Theory Workshop*, pp. 97-101, Punta del Este (Uruguay), March 2006.

[44] Baccelli, E. (2006). *Routing and Mobility in Large Heterogeneous Packet Networks*. Ph.D Thesis, École Polytechnique, Paris (France), January 2006.

[45] Roy, A. (2005). *Adjacency Reduction in OSPF Using SPT Reachability*, Internet-Draft `draft-roy-ospf-smart-peering-01`, IETF, November 2005.

[46] Gast, M. S. (2005). *802.11 Wireless Networks*, 2nd Edition, O'Reilly, ISBN 0-596-10052-3.

[47] Tse, D.; Viswanath, P. (2005). *Fundamentals of Wireless Communications*, Cambridge University Press, ISBN 978-0-521-84527-4.

[48] Cisco Systems Inc. (2005). *Understanding and Configuring the* `ip unnumbered` *Command*, Cisco Design Technote no. 13786, available in `www.cisco.com/en/US/tech/tk648/tk362/technologies_tech_note09186a0080094e8d.shtml`, last modification in October 2005.

[49] Hinden, R.; Haberman, B. (2005). RFC 4193, *Unique Local IPv6 Unicast Addresses*, IETF, October 2005.

[50] Goyal, M.; Xie, W.; Hosseini, S. H.; Vairavan, K.; Rohm, D. (2005). Improving OSPF Dynamics on a Broadcast LAN, *Proceedings of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pp. 105-114, Atlanta, GA (United States), September 2005.

[51] Cisco Systems Inc. (2005). *OSPF Design Guide*, Cisco White paper no. 7039, available in `www.cisco.com/en/US/tech/tk365/technologies_white_paper09186a0080094e9e.shtml`, last modification in August 2005.

[52] Gredler, H.; Goralski, W. (2005). *The Complete IS-IS Routing Protocol*, Springer, ISBN 1-85233-822-9.

[53] Villaseñor-González, L.; Ying, G.; Lament, L. (2005). HOLSR: A Hierarchical Proactive Routing Mechanism for Mobile Ad hoc Networks. In: *IEEE Communications Magazine*, Volume 43, Issue 7, pp. 118-125, IEEE.

[54] Henderson, T.; Spagnolo, P.; Pei, G. (2005). *Evaluation of OSPF MANET Extensions*, Technical Report D950-10897-1, The Boeing Company, July 2005.

[55] Stojmenovic, I.; Nayak, A.; Kuruvila, J.; Ovalle-Martinez, F.; Villanueva-Peña, E. (2005): Physical layer impact on the design and performance of routing and broadcasting protocols in ad hoc and sensor networks. In: *Journal of Computer Communications*, Volume 28, Issue 10, Elsevier, June 2005.

[56] Giruka, V. K.; Singhal, M. (2005). Hello Protocols for Ad-Hoc Networks: Overhead and Accuracy Tradeoffs, *Proceedings of the 6th IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM'05)*, pp. 354-361, Taormina-Giardini (Italy), June 2005.

[57] Cheshire, S.; Aboba, B.; Guttman, E. (2005). RFC 3927, *Dynamic Configuration of IPv4 Link-Local Addresses*, IETF, May 2005.

[58] Busson, A.; Mitton, N.; Fleury, E. (2005). An analysis of the MPR selection in OLSR and consequences, *Proceedings of the 4th Mediterranean Ad Hoc Networking Workshop (MedHocNet'05)*, Île de Porquerolles (France), June 2005.

[59] Cartigny, J.; Ingelrest, F.; Simplot, D.; Stojmenovic, I. (2005). Localized LMST and RNG based minimum-energy broadcast protocols in ad hoc networks, In: *Ad hoc Networks*, Elsevier, Vol. 3, Number 1, pp. 1-6, January 2005.

[60] Adjih, C.; & Viennot, L. (2005). Computing connected dominated sets with multipoint relays. In: *Journal of Ad Hoc and Sensor Wireless Networks*, Volume 1, Number 1-2, pp. 27-39, Old City Publishing, 2005.

[61] Rőmer, K.; Mattern, F. (2004). The Design Space of Wireless Sensor Networks. In: *IEEE Wireless Communications*, Volume 11, Issue 6, December 2004, pp. 54-61, IEEE.

[62] Ogier, R.; Templin, F.; Lewis, M. (2004). RFC 3684, *Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)*, IETF, February 2004.

[63] Carle, J.; Simplot, D. (2004). Energy-Efficient Area Monitoring for Sensor Networks, In: *Computer*, pp. 40-46, IEEE Computer Society, February 2004.

[64] Aguayo, D.; Bicket, J.; Biswas, S.; Judd, G.; Morris, R. (2004). Link-level measurements from an 802.11b mesh network. *Proceedings of the 2004 ACM SIGCOMM Conference*, Volume 34, Issue 4, pp. 121-132, Portland, OR (United States), September 2004.

[65] Cisco Systems *et al.* (2004). *Internetworking Technologies Handbook*, 4th Edition, Cisco Press, ISBN 1-58705-119-2.

[66] Blum, J. *et al.* (2004). Connected Dominating Set in Sensor Networks and MANETs. In: Zu, D.-Z. & Pardalos, P. (*eds.*): *Handbook of Combinatorial Optimization*, pp. 329-369, Kluwer Academic Publishers.

[67] Mans, B. & Shrestha, N. (2004). Performance Evaluation of Approximation Algorithms for Multipoint Relay Selection, *Proceedings of Med-Hoc-Net'04*, Bodrum (Turkey).

[68] Luo, J.; Hubaux, J.-P. (2004). *A survey of inter-vehicle communication*, Technical Report IC/2004/24, School of Computer and Communication Sciences, EPFL, 2004.

[69] Adjih, C.; Baccelli, E.; Clausen, T.; Jacquet, P.; Rodolakis, G. (2004). Fish Eye OLSR Scaling Properties. In: *IEEE Journal of Communications and Networks (JCN)*, Special Issue on Mobile Ad Hoc Wireless Networks, Volume 6, Number 4, pp. 343-351, IEEE, December 2004.

[70] Tanenbaum, A. S. (2003). *Computer Networks*, Prentice Hall, ISBN 0-13-066102-3.

[71] Clausen, T.; Jacquet, P. (2003). RFC 3626, *Optimized Link State Routing Protocol (OLSR)*, IETF, October 2003.

[72] Henderson, T. *et al.* (2003). A Wireless Interface Type for OSPF, *Proceedings of the IEEE Military Communications Conference (MILCOM)*, pp. 137-145, Boston, MA (United States), October 2003.

[73] Adjih, C.; Baccelli, E.; Jacquet, P. (2003). Link-State Routing in Wireless Ad hoc Networks, *Proceedings of the IEEE Military Communications Conference (MILCOM)*, pp. 1274-1279, Boston, MA (United States), October 2003.

[74] Riley, G. F. (2003). The Georgia Tech Network Simulator, *Proceedings of the ACM SIGCOMM Workshop on Models, Methods and Tools for Reproducible Network Research*, pp. 5-12, ACM Press, Karlsruhe (Germany), August 2003.

[75] Perkins, C.; Belding-Royer, E.; Das, S. (2003). RFC 3561, *Ad hoc On-Demand Distance Vector (AODV) Routing*, IETF, July 2003.

[76] Kotz, D.; Newport, C.; Elliott, C. (2003). *The mistaken axioms of wireless-network research*, Technical report TR2003-647, Dartmouth CS Department, July 2003.

[77] Droms R. (*ed.*) *et al.* (2003). RFC 3315, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*, IETF, July 2003.

[78] Cartigny, J.; Ingelrest, F.; Simplot, D. (2003). RNG Relay Subset Flooding Protocols in Mobile Ad-hoc Networks, In: *International Journal of Foundations of Computer Science*, Volume 14, Number 3, pp. 253-265, World Scientific, June 2003.

[79] Tseng, Y.-Ch.; Ni, S.-Y. & Shih, E.Y. (2003). Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Network. *IEEE Transactions on Computers*, Vol. 52, No. 5, pp. 545-557, May 2003.

[80] Goyal, M.; Ramakrishnan, K. K.; Feng, W. (2003). Achieving Faster Failure Detection in OSPF Networks, *Proceedings of the IEEE International Conference on Communications (ICC'03)*, Vol. 1, pp. 296-300, May 2003.

[81] Cartigny, J.; Simplot, D.; Stojmenovic, I. (2003). Localized minimum-energy broadcasting in ad-hoc networks, *Proceedings of the IEEE INFOCOM 2003*, pp. 2210-2217, San Francisco, CA (United States), April 2003.

[82] De Couto, D. S. J.; Aguayo, D., Bicket, J.; Morris, R. (2003). A High-Throughput Path Metric for Multi-Hop Wireless Routing. *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp. 134-146, San Diego, CA (United States), September 2003.

[83] Jacquet, P.; Laouiti, A.; Minet, P.; Viennot, L. (2002). Performance of Multipoint Relaying in Ad Hoc Mobile Routing Protocols, *Proceedings of the 2nd International IFIP-TC6 Networking Conference (NETWORKING'2002)*, pp. 387-398, May 2002.

[84] Cisco Systems Inc. (2002). *Intermediate System-to-Intermediate System Protocol*, Cisco White paper, available in `www.cisco.com/en/US/products/ps6599/products_white_paper09186a00800a3e6f.shtml`, 2002.

[85] ISO (2002): ISO/IEC 10589:2002(E), *Information technology – Telecommunications and information exchange between systems – Intermediate System to Intermediate System intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service (ISO 8473)*, International Organization for Standardization, 2002.

[86] Chakeres, I. D.; Belding-Royer, E.-M. (2002). The Utility of Hello Messages for Determining Link Connectivity, *Proceedings of the 5th International Symposium on Wireless Personal Multimedia Communications*, Vol. 2, pp. 504-508, October 2002.

[87] Clausen, T.; Jacquet, P.; Viennot, L. (2002). Comparative Study of Routing Protocols for Mobile Ad Hoc Networks, *Proceedings of the 1st Annual Mediterranean Ad Hoc Networking Workshop*, September 2002.

[88] Qayyum, A.; Viennot, L. & Laouiti, A. (2002). Multipoint Relaying for Flooding Broadcast

Messages in Mobile Wireless Networks, *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, Vol. 9, pp. 298-308, IEEE ComSoc, Hawaii, HI (United States), January 2002.

[89] Various Authors (2002). Charter of the MANET Working Group, IETF Secretariat, *Proceedings of the 55th IETF Meeting*, Atlanta, GA (United States).

[90] Stoica, I.; Morris, R.; Karger, D.; Frans Kaashoek, M.; Balakrishnan, H. (2001). Chord: a scalable peer-to-peer lookup service for Internet applications, *Proceedings of the 2001 ACM Conference on Applications, Technologies, Architectures and Protocols for Computer Communications (SIGCOMM'01)*, pp. 149-160, San Diego, CA (United States), August 2001.

[91] Doyle, J.; DeHaven Carroll, J. (2001). *Routing TCP/IP*, Volume II, Cisco Press, ISBN 1-57870-089-2.

[92] Perkins, C. E. (2001). Ad Hoc Networking – An Introduction, In: Perkins, C. E. (*Ed.*) (2001). *Ad Hoc Networking*, Addison-Wesley, ISBN 0-201-30976-9.

[93] Freebersyser, J. A.; Leiner, B. (2001). A DoD Perspective on Mobile Ad Hoc Networks, In: Perkins, C. E. (*Ed.*) (2001). *Ad Hoc Networking*, Addison-Wesley, ISBN 0-201-30976-9.

[94] Rappaport, T. S. (2001). *Wireless Communications Principles and Practice*, 2nd Edition, Prentice Hall, ISBN-10 0-13-042232-0.

[95] Jannotti, J.; Gifford, D. K.; Johnson, K. L.; Fraans Kaashoek, M.; O'Toole Jr., J. W. (2000). Overcast: Reliable Multicasting with an Overlay Network, *Proceedings of the Usenix OSDI Symposium 2000*, pp. 197-212, October 2000.

[96] Chu, Y.-H.; Rao, S. G.; Zang, H. (2000). A Case of End System Multicast, *Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'00)*, pp. 1-12, Santa Clara, CA (United States), June 2000.

[97] Lee, S.-J.; Gerla, M. (2000). AODV-BR: Backup Routing in Ad hoc Networks, *Proceedings of the Wireless Communications and Networking Conference (WCNC)*, pp. 1311-1316, Chicago, IL (United States), 2000.

[98] Halabi, S.; McPherson, D. (2000). *Internet Routing Architectures*, 2nd Edition, Cisco Press, ISBN 1-57870-233-X.

[99] Ni, S.Y.; Tseng, Y.-Ch.; Chen, Y.-S.; Sheu, J.-P. (1999). The Broadcast Storm Problem in a Mobile Ad Hoc Network, *Proceedings of the Annual International Conference on Mobile Computing and Networking*, pp. 151-161, ACM Press, Seattle (United States), August 1999.

[100] Perlman, R. (1999). *Interconnections: Bridges, Routers, Switches and Internetworking Protocols*, 2nd Edition, Addison-Wesley, ISBN-13 978-0201634488.

[101] Conta, A.; Deering, S. (1998). RFC 2463, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*, IETF, December 1998.

[102] Malkin, G. (1998). RFC 2453, *RIP Version 2*, IETF, November 1998.

[103] Das, S.R.; Castañeda, R.; Jiangtao Yan; Sengupta, R. (1998). Comparative performance evaluation of routing protocols for mobile, ad hoc networks, *Proceedings of the 7th International Conference on Computer Communications and Networks*, pp. 153-161, October 1998.

[104] Broch, J.; Maltz, D.; Johnson, D.; Hu, Y.-C.; Jetcheva, J. (1998). A Performance Comparison of Multi-hop Wireless Ad Hoc Network Routing Protocols, *Proceedings of ACM Annual International Conference on Mobile Computing and Networking (MobiCom'98)*, pp. 85-97, Dallas, TX (United States), October 1998.

[105] Hinden, R.; Deering, S. (1998). RFC 2373, *IP Version 6 Addressing Architecture*, IETF, July 1998.

[106] Moy, J. (1998). RFC 2329, *OSPF Standardization Report*, IETF, April 1998.

[107] Moy, J. (1998). RFC 2328, *OSPF Version 2*, IETF, April 1998.

[108] Droms, R. (1997). RFC 2131, *Dynamic Host Configuration Protocol*, IETF, March 1997.

[109] Leiner, B. M.; Cerf, V. G. *et al.* (1997). A Brief History of the Internet, version 3.2. Publicly available in *ArXiv*, `arxiv.org/html/cs.NI/9901011`. *(last revision in January 27th, 1997)*

[110] Malkin, G.; Minnear, R. (1997). RFC 2080, *RIPng for IPv6*, IETF, January 1997.

[111] Hawkinson, J.; Bates, T. (1996). RFC 1930, *Guidelines for creation, selection and registration of an Autonomous System (AS)*, IETF, March 1996.

[112] Elz, R. (1996). RFC 1924, *A compact representation of IPv6 addresses*, IETF, April 1996.

[113] Strater, J.; Wollman, B. (1996). *OSPF Modeling and Test Results and Recommendations*, Mitre Technical Report 96W0000017, Xerox Office Products Division, March 1996.

[114] Rekhter, Y.; Moskowitz, B.; Karrenberg, D.; De Groot, G. J; Lear, E. (1996). RFC 1918, *Address Allocation for Private Internets*, IETF, February 1996.

[115] ISO (1995): ISO/IEC TR 9575:1995(E), *Information technology – Telecommunications and information exchange between systems – OSI Routeing Framework*, International Organization for Standardization, 1995.

[116] Baker, F. (1995). RFC 1812, *Requirements for IP Version 4 Routers*, IETF, June 1995.

[117] Rekhter, Y.; Li, T. (1995). RFC 1771, *A Border Gateway Protocol 4 (BGP-4)*, IETF, March 1995.

[118] Jaromczyk, J.; Toussaint, G. T. (1992). Relative Neighborhood Graphs and their Relatives, In: *Proceedings of the IEEE*, Vol. 80, Number 9, pp. 1502-1517, September 1992.

[119] Simpson, W. (1994). RFC 1661, *The Point-to-Point Protocol (PPP)*, IETF, July 1994.

[120] Clark, B. N.; Colbourn, C. J. & Johnson, D. S. (1990). Unit Disk Graphs. In: *Discrete Mathematics*, Volume 86, pp. 165-177.

[121] Callon, R. (1990). RFC 1195, *Use of OSI IS-IS for Routing in TCP/IP and Dual Environments*, IETF, December 1990.

[122] Oran, D. (1990). RFC 1142, *OSI IS-IS Intra-domain Routing Protocol*, IETF, February 1990.

[123] Braden, R. (1989). RFC 1122, *Requirements for Internet Hosts – Communication Layers*, IETF, October 1989.

[124] Hendrik, C. (1988). RFC 1058, *Routing Information Protocol*, IETF, June 1988.

[125] Devroye, L. (1988). The Expected Size of some Graphs in Computational Geometry. In: *Computers and Mathematics with Applications*, Volume 15, pp. 53-64.

[126] Mills, D. L. (1986). RFC 975, *Autonomous Confederations*, IETF, February 1986.

[127] Rosen, E. C. (1982). RFC 827, *Exterior Gateway Protocol (EGP)*, IETF, October 1982.

[128] Postel, J. *(Ed.)* (1981). RFC 791, *Internet Protocol*, IETF, September 1981.

[129] Postel, J. *(Ed.)* (1981). RFC 790, *Assigned Numbers*, IETF, September 1981.

[130] Matula, D. W.; Sokal, R. R. (1980). Properties of Gabriel graphs relevant to geographic variation research and clustering of points in the plane, In: *Geographical Analysis*, Volume 12, Issue 3, pp. 205-222, July 1980.

[131] Toussaint, G. (1980). The Relative Neighborhood Graph of Finite Planar Set. In: *Pattern Recognition*, Vol. 12, Number 4, pp. 261-268.

[132] Gabriel, K. R.; Sokal, R. R. (1969). A new statistical approach to geographic variation analysis. In: *Systematic Zoology*, Vol. 18, Number 3, pp. 259-270.

[133] Ford, L. R. Jr. & Fulkerson, D. R. (1962). *Flows in Networks*, Princeton University Press.

[134] Kleinrock, L. (1962). *Message Delay in Communication Nets with Storage*. PhD Thesis, Massachusetts Institute of Technology (MIT), Cambridge, MA (United States), December 1962.

[135] Dijkstra, E. W. (1959). A Note on Two Problems in Connection with Graphs, In: *Numersiche Mathematik*, No. 1, pp. 269-271.

[136] Bellman, R. (1958). On a Routing Problem, In: *Quarterly of Applied Mathematics*, No. 16, pp. 87-90.

[137] Delaunay, B. (1934). Sur la sphère vide, In: *Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk*, Vol. 7, pp. 793-800.

APPENDICES

Appendix A

Link Equivalence

Definition A.1 (Equivalent links)**.** Let $l_1 : s_1 \longrightarrow d_1$ and $l_2 : s_2 \longrightarrow d_2$ be links. Then, l_1 and l_2 are said to be *equivalent* (denoted as $l_1 \equiv l_2$) if and only if any of the following conditions are fulfilled:

(i) $s_1 = s_2$, $d_1 = d_2$

(ii) $s_1 = s_2$, $d_1 \neq d_2$ and any packet sent from s_1 to d_1 through l_1 is also received by d_2 through l_2 (and vice versa)

(iii) $s_1 \neq s_2$ and $\exists l^*_{12} : s_1 \longrightarrow s_2, l^*_{21} : s_2 \longrightarrow s_1$ such that any packet sent from s_1 to d_1 through l_1 is also received by:

- s_2, through l^*_{12}, and
- d_2, through l_2

and vice versa.

Proposition A.1. *Relation $\equiv$ between links is an* equivalence relation *in the mathematical sense, thus satisfying:*

- *Reflexivity:* $l \equiv l$.
- *Symmetry:* $l_1 \equiv l_2 \Longrightarrow l_2 \equiv l_1$.

- *Transitivity:* $l_1 \equiv l_2, l_2 \equiv l_3 \Longrightarrow l_1 \equiv l_3$.

Proof. Reflexivity is evident. Symmetry is also evident, as the conditions from the definition are symmetric with respect to the two considered links.

For the study of transitivity (*i.e.* $l_1 \equiv l_2, l_2 \equiv l_3 \Longrightarrow l_1 \equiv l_3$,), the following cases have to be distinguished:

a) $l_1 \equiv l_2$ fulfills (i); $l_2 \equiv l_3$ fulfills (i)

b) (i) – (ii)

c) (i) – (iii)

d) (ii) – (ii)

e) (ii) – (iii)

f) (iii) – (iii)

which contain also the symmetric cases.

Cases a), b) and c) are evident.

Case d): The fact that $l_1 \equiv l_2$ and $l_2 \equiv l_3$ fulfill (ii) implies that $s_1 = s_2 = s_3 = s$. When a packet is sent over $s \xrightarrow{l_1} d_1$, it is also received by d_2 through link l_2, as $l_1 \equiv l_2$. And, since $l_2 \equiv l_3$, this packet is also received by d_3 through l_3. Therefore, the relation between l_1 and l_3 satisfies condition (ii) and $l_1 \equiv l_3$.

Case e): Consider the links $s_1 \xrightarrow{l_1} d_1$, $s_2 \xrightarrow{l_2} d_2$ and $s_2 \xrightarrow{l_2} d_2$. The fact that $l_1 \equiv l_2$ fulfills (ii) implies that $s_1 = s_2$ and $d_1 \neq d_2$, that a packet that is sent over $s_1 \xrightarrow{l_1} d_1$ is received by d_2 through link l_2, and, symmetrically, a packet that is sent over $s_2 \xrightarrow{l_2} d_2$ is also received by d_1 through link l_1. The fact that $l_2 \equiv l_3$ fulfills (iii) implies that $s_2 \neq s_3$ and that there exists a link $s_2 \xrightarrow{l^*_{23}} d_3$, such that a packet sent over $s_2 \xrightarrow{l_2} d_2$ is also received by s_3 (through link l^*_{23}, and d_2 (through link l_2). Symmetrically, there exists a link $s_3 \xrightarrow{l^*_{32}} d_2$, such that a packet sent over $s_3 \xrightarrow{l_3} d_3$ is also received by s_2 (through link l^*_{32}, and d_3 (through link l_3).

In this case, given that $s_1 \neq s_3$, it has to be shown that relation between l_1 and l_3 fulfills condition (iii). Since $s_1 = s_2$, existence of a link $s_1 \xrightarrow{l^*_{13}} s_3$ is equivalent to the existence of a link $s_2 \xrightarrow{l^*_{23}} s_3$. Consider a packet sent over $s_1 = s_2 \xrightarrow{l_3} d_3$. Then, this packet is received by s_3 (through link l^*_{23}) and d_2 (through l_2), as $l_2 \equiv l_3$. When a packet is received by d_2 through l_2, it is also received by d_1 through l_1, because $l_1 \equiv l_2$. The same argument applies for the existence of a link $s_3 \xrightarrow{l^*_{31}} s_1$. Therefore, relation between l_1 and l_3 fulfills condition (iii) and $l_1 \equiv l_3$.

Case f): The fact that $l_1 \equiv l_2$ fulfills (iii) implies that $s_1 \neq s_2$, and that there exists a link $s_1 \xrightarrow{l^*_{12}} d_2$ such that packets sent over $s_1 \xrightarrow{l_1} d_1$ are also received by s_2 (through l^*_{12}) and d_2 (through l_2). Symmetrically, there exists a link $s_2 \xrightarrow{l^*_{21}} d_1$ such that packets sent over $s_2 \xrightarrow{l_2} d_2$ are also received by s_1 (through l^*_{21}) and d_1 (through l_1).

Also, the fact that $l_2 \equiv l_3$ fulfills (iii) implies that $s_2 \neq s_3$, and that there exists a link $s_2 \xrightarrow{l^*_{23}} d_3$ such that packets sent over $s_2 \xrightarrow{l_2} d_2$ are also received by s_3 (through l^*_{23}) and d_3 (through l_3). Symmetrically, there exists a link $s_3 \xrightarrow{l^*_{32}} d_2$ such that packets sent over $s_3 \xrightarrow{l_3} d_3$ are also received by s_2 (through l^*_{32}) and d_2 (through l_2).

Then, two subcases need to be considered to prove that $l_1 \equiv l_3$: first, $s_1 \neq s_3$; and second, $s_1 = s_3$.

Subcase f.1) $s_1 \neq s_3$. The existence of a link $s_1 \xrightarrow{l^*_{13}} s_3$ has to be proved. It is known that a packet sent over $s_1 \xrightarrow{l_1} d_1$ is also received by d_2 through link l_2. This implies that the packet is also received by s_3 (through l^*_{23}) and d_3 (through l_3). Therefore, there is a link between s_1 and d_3 – let l^*_{13} be such link. Existence of link $s_3 \xrightarrow{l^*_{31}} d_1$ is proven symmetrically.

Subcase f.2) $s_1 = s_3$. The existence of a link $s_1 \xrightarrow{l^*_{13}} s_3$ is then trivial. □

Appendix B

Wireless Channel Models

B.1 Unit Disk Graph – UDG

The Unit Disk Graph [120] assumes that the coverage area of a wireless interface is a circle of radius r. Therefore, a wireless interface receives successfully a transmission from another interface if and only if its distance is lower than the coverage radio r of the transmitting interface. In terms of power, this assumption is equivalent to ignore transmission losses:

$$P_r = \begin{cases} P_t & , d \leq r \\ 0 & , d > r \end{cases} \tag{B.1}$$

While this model is not realistic, it provides a reasonable framework for studying analytically some relevant properties of wireless networks and its performance.

B.2 Two-Ray Model

The *Two-Ray Ground Reflection Model* considers the contribution of two different signal paths: the direct path between transmitter and receiver and an additional path reflected on the ground (see Figure B.1).

$$\frac{P_r}{P_t} = G_t G_r \left(\frac{h_t h_r}{d^2} \right)^2 \tag{B.2}$$

where h_t and h_r are the heights of the transmitting and receiving antennas, respectively. This model predicts a more significant signal attenuation with respect to the distance (order d^4) as a consequence of the interference of the reflected ray.

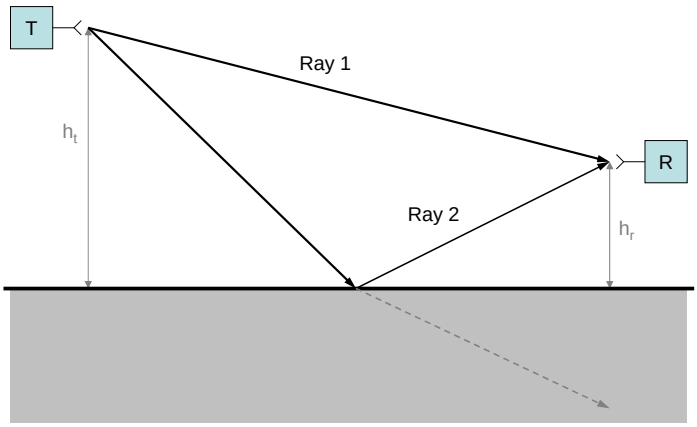


Figure B.1: Illustration of the two-ray propagation model.

Appendix C

IEEE 802.11 Standards

Table C.1 summarizes the most relevant features of the main physical layer (PHY) standards of the IEEE 802.11 family for WLAN.

	Release	Max. rate	Frequency band	Modulation & coding[1]	Indoor rg	Outdoor rg
-	1997	1, 2 Mbps	$2.4GHz$	DSSS, FHSS	20 m	100 m
a	1999	54 Mbps	$5GHz$	OFDM	35 m	120 m
b	1997	1, 2 Mbps	$2.4GHz$	DSSS (Barker)	38 m	140 m
	1999	5.5, 11 Mbps	$2.4GHz$	HR / DSSS (CCK, PBCC)	38 m	140 m
g	2003	54 Mbps	$2.4GHz$	OFDM, DSSS	38 m	140 m
n	2009	$\sim$600 Mbps	$2.4/5GHz$	MCS	70 m	250 m
p	2010	54 Mbps	$5.9GHz$	OFDM	1000 m	

Table C.1: IEEE 802.11 family of standards [46].

The first specification, from 1997, was complemented in 1999 with the standards a and b, not compatibles between them. a operates in the $5GHz$ band, which is expected to suffer less interferences but required more expensive devices. Specification b from 1997 used the Barker coding

[1]DSSS: Direct Sequence Spread Spectrum; HR/DSSS: High Rate DSSS; FHSS: Frequency Hopping Spread Spectrum; CCK: Complementary Code Keying; PBCC: Packet Binary Convolutional Code; OFDM: Orthogonal Frequency-Division Multiplexing; MCS: Modulation and Coding Schemes. DSSS schemes are used with Barker sequence coding and DBPSK/DQPSK (Differential Binary Phase-Shift Keying/Differential Quadrature Phase-Shift Keying) modulation techniques in 802.11b; in High Rate DSSS other coding techniques, as CCK (together with QPSK modulation) or PBCC (with 64-QAM modulation), are used instead. OFDM schemes are used together with different modulation techniques such as BPSK, QPSK or QAM (Quadrature Amplitude Modulation). 802.11n defines 77 Modulation and Coding Schemes (MCS) that use BPSK, QPSK and QAM techniques.

sequence for spreading the signal spectrum, which enabled maximum rates of 1 and $2Mbps$. The use of other coding techniques such as CCK or PBCC in High Rate DSSS (HR / DSSS) schemes, in the 1999 specification of 802.11b, permitted improving the modulation efficiency and enabled the b extension to achieve nominal transmission rates of $5.5Mbps$ and $11Mbps$; this standard became widely spread due to the reduced cost of the associated deployments. g is also based on b and devices produced under g standard keep backwards compatibility (b/g): it introduces the OFDM modulation scheme and increases the theoretical throughput to $54Mbps$. n was designed to increase significantly the transmission rate with respect to versions a and g. Several improvements were thus incorporated: the channel bandwidth was doubled (from $20MHz$ to $40MHz$ around the channel carrier), multiple antennas were allowed to take profit of multipaths (MIMO[2] techniques) and support was provided for both the $2.4GHz$ and the $5GHz$ bands. These improvements permit to achieve a maximum (theoretical) transmission rate of $600Mbps$. Finally, p defines physical and MAC layers of the Wireless Access for Vehicular Environments (WAVE) family of standards, intended to adapt IEEE 802.11 to the requirements of car-to-car communication.

[2]Acronym of Multiple Input – Multiple Output.

Appendix D

SLOT Simulations

D.1 Mobile Scenarios

Basics

Simulations of SLOT-U and SLOT-D overlays are performed in Maple, and use the unit disk graph model. Link density and link creation rates are measured in a $6r \times 6r$ grid, with the number of nodes in the network varying from a few to several hundreds.

Mobility Model

The mobility model used in these simulations is the following: nodes move independently according to a random walk with a constant speed of one unit per second. The nodes change direction every 0.01 second. When a mobile node encounters the border it bounces as in a billiard, the outcoming speed vector being the mirror image of the incoming speed vector. The new overlay link creation rate is measured in this context – which is, of course, equal to the average overlay link failure rate in order to have a constant average density.

D.2 Static Scenarios

Routers are distributed uniformly over a finite square scenario ($600m \times 600m$ grid). The distance-based costs of SLOT-D as computed as $m_d(\overline{xy}) = \lceil \frac{K}{r} d(x,y) \rceil \in \mathbb{N}$ ($d(x,y)$ measuring the Euclidean distance between x and y), that discretizes the link length into a number between 1 and K.

Appendix E

Simulation Parameters

Simulation results shown in this paper were obtained using the Quagga/Zebra OSPF implementation, via the `ospf6d` daemon, and simulations with the GTNetS [74] simulator. The implementation of OR / SP, detailed in [54] and [41], follows the specification in [19]. The implementation of MPR-OSPF follows the specification in [24]. The implementation of SLOT-OSPF follows the algorithms detailed in [10].

E.1 Scenario, Traffic and Protocol Configuration

The following tables describe the simulation environment parameters. Routers have one wireless interface. Table E.1 shows the default values of the main parameters (when not explicitly mentioned in the figures). Tables E.2 and E.3 show the parameters specific to the configurations considered in this paper.

E.2 α Parameter for Wireless Transmission Model

The α parameter determines the probability of successful transmission through a wireless channel, for GTNetS simulations based on [54].

Table E.1: General Simulation Parameters.

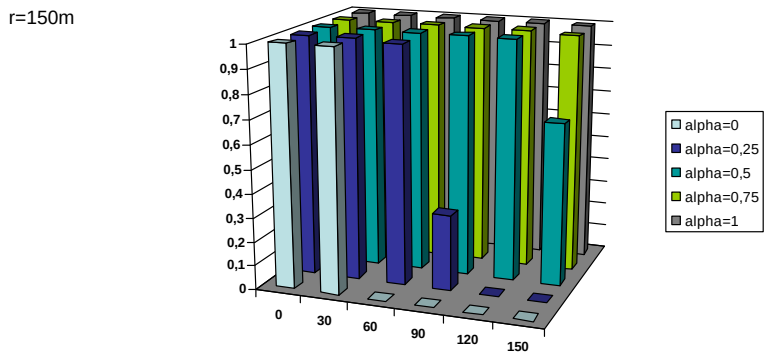
Name	Value	
	General Evaluation	Hello Evaluation
Experiment Statistic Parameters		
Seed	0	
Samples/experiment	20	5
Traffic Patterns		
Type of traffic	CBR UDP	
Packet payload	1472 bytes	40 bytes
Packet rate	85 pkts/sec	10 pkts/sec
Traffic rate	1 Mbps	3,2 Kbps
Scenario		
Mobility	Random waypoint model	
Speed	Constant $v = 0,5\frac{m}{s}$	Uniform $\sim U[0, v_{mx}]$ $v_{mx} = 0,5\frac{m}{s}$
Grid shape and size	Square, 400 m $\times$ 400 m	
Radio range	150 m	
Propagation	Two-ray ground model	
Wireless α	0.5	
Pause time	0 sec	40 sec
MAC protocol	IEEE 802.11b	
OSPF General Configuration		
HelloInterval	2 sec	
DeadInterval	6 sec	
RxmtInterval	5 sec	
MinLSInterval	5 sec	
MinLSArrival	1 sec	
LSRefreshInterval	20 sec	

Table E.2: RFC 5820 (OR/SP) Specific Parameters.

Name	Value
AckInterval	1800 msec
PushbackInterval	2000 msec
Optimized Flooding?	Yes
Smart Peering?	Yes
Unsynch. adjacencies?	Yes
Surrogate Hellos?	Yes
Incremental Hellos?	No

Table E.3: MPR-OSPF (and Variations MPR+SP and SLOT-OSPF) Specific Parameters.

Name	Value
AckInterval	1800 msec
Flooding MPR?	Yes
Topology Reduction	MPR Topology Reduction
Adjacency Selection	MPR Adjacency Reduction SLOT-U Adjacency Policy Smart Peering



	0	30	60	90	120	150
alpha=0	1	1	0	0	0	0
alpha=0,25	1	1	0,99	0,313	0	0
alpha=0,5	1	1	1	1	0,998	0,675
alpha=0,75	1	1	1	1	1	0,99
alpha=1	1	1	1	1	1	1

Figure E.1: Impact of the α parameter in the probability of successful reception, for $r = 150m$.

Appendix F

Testbed Configuration

F.1 Hardware and Software Description

F.1.1 Hardware

Networking interface drivers were the following:

- Wired interfaces: Digital Equipment Corporation DECchip 21140.
- Wireless interfaces: Broadcom BCM4306 WLAN.

F.1.2 Software

Software used in all computers was as follows:

- Operating System: Ubuntu v.10.04 with kernel 2.6.32.
- Routing Protocol Implementation: `ospf6d` daemon of Quagga/Zebra routing suite v.0.99.15.
- Used interface types:
  - Wired interfaces: Point-to-point.
  - Wireless interfaces: MANET, as specified in RFC 5449 [24].

F.2 Experiments Setup

- Routers were switched on between $t = 0sec$ and $t = 2sec$.

F.2.1 PDR and RTT Measures

- PDR results averaged over 84 iterations.
- `ospf6d` daemon NOT restarted in each iteration.
- UDP flows, started $60sec$ after `ospf6d` daemon switch-on:

Nominal sender bit rate	100 pkts/s
Packet payload	1024 bytes
CBR real traffic rate	$\sim$ 300 kbps
Flow duration	5 min/flow

Table F.1: Characteristics of transmitted UDP flows.

- RTT results averages over 60 iterations (ICMPv6 requests).
- ICMPv6 request did not overlap with UDP flows.

F.2.2 Control Traffic Measures

- Results averaged over 84 iterations.
- `ospf6d` daemon restarted in each iteration.
- UDP flows: see Table F.1.

F.3 OSPF Parameters

Name	Value
OSPF General Configuration	
HelloInterval	2 sec
DeadInterval	10 sec
RxmtInterval	5 sec
AckInterval	2 sec
Jitter (max.)	100 msec
MinLSInterval	5 sec
MinLSArrival	1 sec
LSRefreshInterval	60 sec

Table F.2: General Simulation Parameters.