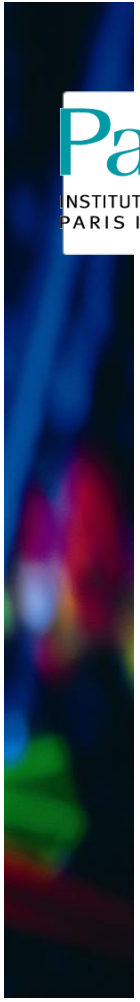
Coordination of Self-Organizing Network (SON) functions in next generation radio access networks

Ovidiu Constantin Iacoboaiea



ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2015-ENST-0067

EDITE ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

Télécom ParisTech

Spécialité “ Signal et Images ”

présentée et soutenue publiquement par

Ovidiu-Constantin IACOBOAIEA

le 2 octobre 2015

Coordination des fonctionnalités auto-organisantes (SON) dans les réseaux d'accès radio du futur

Directeur de thèse : **Pascal BIANCHI**
Directrice de thèse : **Berna SAYRAC**
Co-directrice de thèse : **Sana BEN JEMAA**

Jury

M. Eitan ALTMAN, Directeur de recherche, INRIA, France	Rapporteur
M. Samson LASAULCE, Directeur de recherche CNRS, Supélec, France	Rapporteur
M. Tijani CHAHED, Professor, Telecom SudParis, France	Examinateur
M. Henning SANNECK, Senior Research Engineer, Nokia, Allemagne	Examinateur
M. Christophe LE MARTRET, Senior Research Engineer, Thales, France	Invité
M. Pascal BIANCHI, Professeur, TSI, Télécom ParisTech, France	Directeur de thèse
Mme. Berna SAYRAC, Senior Research Engineer, Orange Labs, France	Directrice de thèse
Mme. Sana BEN JEMAA, Senior Research Engineer, Orange Labs, France	Co- directrice de thèse

T H È S E

Télécom ParisTech
école de l’Institut Mines Télécom – membre de ParisTech
46, rue Barrault – 75634 Paris Cedex 13 – Tél. + 33 (0)1 45 81 77 77 – www.telecom-paristech.fr

Coordination of Self-Organizing Network (SON) functions in next generation radio access networks

- Ph.D. Thesis -

Ovidiu-Constantin IACOBOAIEA

Telecom ParisTech and Orange Labs

Thesis director: Pascal Bianchi
Thesis director: Berna Sayrac
Thesis co-director: Sana Ben Jemaa

October 2nd, 2015

Acknowledgements

I am grateful to Berna Sayrac and Sana Ben Jemaa from Orange Labs, and to Pascal Bianchi from Telecom ParisTech, whose expertise, support and generous guidance allowed me to engage in a very interesting topic. Your rigour and constructing advices have been a constant source of motivation and have urged me to do better. Thank you for your time and effort.

I wish to show my appreciation to the members of the jury for accepting to evaluate the work of my thesis: Mr. Samson Lasaulce, Mr. Eitan Altman, Mr. Henning Sanneck, Mr. Christophe Le Martret and Mr. Tijani Chahed, I deeply value your opinions and I am thankful for being able to benefit from your broad experience.

As most of my work was done in the framework of the European project SEMAFOUR, I am also grateful for the unique experience this project has offered me and for the great moments spent with the project colleagues. I wish to tank Christoph Schmelz and Colin Willcock for the inspiring leadership and valuable advices during the project.

During my thesis I was fortunate to work in the team of Radio CEC, Performance, Optimization and Tools in Orange Labs where I have made some great friends. I wish to acknowledge the the help received from my managers Benoît Badard, Salah Eddine El Ayoubi and Laurent Marceron for their efforts in funding my conference participations and for creating a remarkable working environment. I would also like to express my thanks to my gang: Aymen, Abdoulaye, Sudhir, Hajer, Ahlem, Yu Ting, Yasir, Hind, Maroua and Riad and all the friends I have made here for all the good times that we have spent together.

I would also like to thank my family and all my friends for their support, for their patience, for their frankness and for encouraging me to pursue my aspirations.

Ovidiu

Synthèse en français

1. Introduction

Le trafic de données ne cesse d'augmenter. En conséquence l'Union Internationale des Télécommunications (UIT) a rédigé la demande de l'International Mobile Telecommunications-Advanced (IMT-A) pour la 4*ème* génération de réseaux mobiles (4G) [1]. Afin de rester compétitifs, les opérateurs de réseaux doivent trouver des solutions pour répondre à ces demandes. Le Long Term Evolution (LTE), notamment son évolution le LTE-Advanced (LTE-A), est le candidat le plus important pour un système 4G [2]. LTE est standardisé dans la Release 8 du 3^{rd} Generation Partnership Project (3GPP) et LTE-A est considéré comme la Release 10 du 3GPP. À part le LTE, les réseaux mobiles utilisent les technologies anciennes comme le Global System for Mobile Communications (GSM), l'Universal Mobile Telecommunications System (UMTS), etc. De plus, les réseaux hétérogènes représentent aussi une tendance majeure. Gérer toutes ces technologies se traduit par une augmentation de dépenses en capital (CAPEX) et de dépenses opérationnelles (OPEX).

Envisageant ces augmentations de coûts, 3GPP a introduit, dès la Release 8, les réseaux auto-organisant (Self Organizing Network, SON). Il y a principalement trois catégories de fonctions SON : auto-configuration, auto-optimisation et auto-guérison [2]. Dans cette thèse, nous nous focalisons sur les fonctions d'auto-optimisation. Par la suite, SON fait référence à cette catégorie.

Un opérateur réseau peut choisir d'acheter plusieurs fonctions SON pour atteindre ses objectifs en termes de satisfaction de l'utilisateur final. Avoir plusieurs fonctions SON peut entraîner des conflits, notamment si les fonctions SON viennent de différents fournisseurs, mais pas seulement.

Pour que toutes ces SON fonctionnent efficacement ensemble, nous devons nous assurer qu'ils ne sont pas en conflit, ou si un conflit existe, nous devons le détecter et appliquer un mécanisme de résolution. Pour faire face à ces conflits SON, 3GPP a introduit dans la Release 10 la fonction de coordination SON (SON COordinator, SONCO) [3], [4]. Selon [4], les tâches du SONCO incluent : la prévention des conflits (SONCO-A, Avoidance), la détection et le diagnostic des conflits (SONCO-D), et la résolution des conflits (SONCO-R).

Il y a plusieurs stratégies pour coordonner les fonctions SON. Une des approches est d'avoir un co-design des fonctions SON. Cela est possible si les fonctions SON sont toutes construites par un seul fournisseur (ou si plusieurs fournisseurs collaborent sur une telle conception).

Une autre approche est d'élaborer indépendamment les fonctions SON. Ceci est plus probable dans le cadre d'un environnement multi-fournisseur. Deux cas peuvent être identifiés pour cette approche. Dans un cas, les fournisseurs partagent les algorithmes utilisés par les fonctions SON. Ainsi nous disons que les fonctions SON sont considérées comme des *boîtes blanches*. Le designer de SONCO peut donc concevoir une solution englobant toutes ces connaissances. En tout cas, il est très improbable que les fournisseurs de SON partageront les algorithmes.

Ceci nous amène au deuxième cas où les vendeurs ne partagent pas les algorithmes SON. Dans ce cas, les fonctions SON sont considérées comme des *boîtes noires*. Cela crée un environnement approprié aux fournisseurs pour pouvoir proposer des fonctions SON sans être obligé de révéler les

FIGURE 1: LTE et SON dans 3GPP

algorithmes. Donc, un opérateur réseau peut acheter et orchestrer les fonctions SON provenant de différents fournisseurs en fonction de ses besoins.

Avec les considérations précédentes nous avons l'objectif de concevoir un SONCO capable de gérer les fonctions SON conçues indépendamment, vues comme des *boîtes noires*. Dans cette thèse :

1. Nous abordons la détection et le diagnostic des conflits SON (SONCO-D) en utilisant le Classificateur Naïve Bayésien (NBC). Concrètement une fonction SON, en essayant d'atteindre son objectif, pourrait être trop ambitieuse. Cela peut affecter les autres fonctions SON dans le sens où elles sont empêchées d'atteindre leurs objectifs. Le SONCO-D apprend des expériences précédentes, quelle est la cause la plus probable (SON) de cette sous-performance.

2. Nous abordons le problème de la résolution des conflits (SONCO-R) en utilisant un cadre d'apprentissage par renforcement (Reinforcement Learning, RL). Nous nous concentrons sur la résolution de conflits : sur les mesures et sur les paramètres. Les conflits sur les mesures font référence à la stabilité du réseau, c'est-à-dire l'élimination des changements inutiles de paramètres qui se produisent à cause des interdépendances parasites entre les mesures utilisées et les paramètres réglés par les différentes fonctions SON. La résolution des conflits sur les paramètres traite le cas où deux fonctions SON (au moins) demandent le changement d'un paramètre de réseau dans des directions opposées.

3. Nous faisons des recommandations sur l'utilisation des paramètres de réseau. Tout au long de notre travail, nous avons utilisé des paramètres de HandOver (HO) établis *par-cellule*, c'est-à-dire que les paramètres de HO sont les mêmes pour toutes les cellules voisines. En utilisant des paramètres de HO *par-cellule* nous limitons la manœuvrabilité de la mobilité de l'UE. Si nous utilisons des paramètres de HO *par-paire-de-cellules*, cela peut conduire à la création de zones soumises à des HOs Enchaînés Continus (Continuous Chained HOs, CCH). Nous présentons une analyse détaillée pour motiver notre choix.

2. La modélisation du LTE-A et du SON

Tout au long de notre travail, nous avons utilisé un simulateur LTE-A développé en C/C++ avec une résolution temporelle d'une milliseconde. Nous présentons les détails dans les sous-sections suivantes.

Le LTE

Dans notre simulateur, les UEs apparaissent dans le réseau de manière aléatoire. Leur emplacement est mise à jour chaque chaque milliseconde. Soit $RxPow_n^\kappa$ la puissance moyenne reçue par l'UE κ de la cellule n exprimée en dBs (on ignore l'index temporelle). Si l'UE κ veut télécharger des données, il passe d'abord par une procédure d'*Établissement de Connexion* (*Connection Establishment*, Figure 2) afin de s'attacher a une cellule du réseau. Plus précisément, il s'attache a la cellule :

$$iCell_\kappa = \text{argmax}_n \left(RxPow_n^\kappa + CIO_n \right) \tag{1}$$

où CIO_n est l'offset individuel de la cellule n (Cell Individual Offset, CIO) [5]. Si la connexion a réussi, l'UE passera à la procédure de *Réception de Données (Receive Data)*, autrement l'UE est supprimé. Nous considérons un trafic ressemblant au FTP, c'est-à-dire que chaque UE qui arrive dans le réseau vise à télécharger un fichier (de taille fixe FS [MBits/UE]).

Si la cellule de desserte n'offre plus un bon (ou le meilleur) canal radio, l'UE cherchera une nouvelle cellule qui offre un (ou le) meilleur canal radio. Cette procédure s'appelle *ReConfiguration de Connexion (Connection ReConfiguration)* ou *HandOver* (HO). Plus précisément, un

FIGURE 2: Les procédures des UEs

UE attaché à la cellule *sCell* déclenchera le compte à rebours d'un compteur (Time To Trigger) pour effectuer un HO à une cellule cible $tCell \neq sCell$ si :

$$tCell_{\kappa}(t) = \text{argmax}_n \left(RxPow_n^{\kappa}(t) + CIO_n + HYS_{sCell} \mathbb{I}_{\{n=sCell\}} \right) \quad (2)$$

où HYS_{sCell} est le Hystérésis de HO (HYS) de *sCell* [5].

Au cours de son téléchargement de données et même pendant une procédure de HO, l'UE peut perdre la connexion avec la cellule de desserte. Nous appelons cela un échec de liaison radio (Radio Link Failure, RLF). Si un tel événement se produit alors l'UE entrera dans la procédure de *Rétablissement de Connexion* (*Connection ReEstablishment*) où l'UE tentera de se connecter à une cellule du réseau en utilisant des pas similaires à celles de l'*Établissement de Connexion*. Si un RLF se produit pendant la procédure de *Rétablissement de Connexion,* alors l'UE est supprimée.

En ce qui concerne les entrées des fonctions SON, nous nous intéressons aussi aux événements suivants :

- HO trop tard : Un RLF se produit après que l'UE soit resté pendant une longue période de temps attaché à une cellule ; l'UE tente de rétablir la connexion avec une cellule différente.
- HO trop tôt : Un RLF se produit peu après un HO réussi, d'une cellule de desserte à une cellule cible, ou si un échec de HO se produit pendant la procédure de HO ; l'UE tente de rétablir la connexion à la cellule de desserte[6].
- HO vers la mauvaise cellule : Un RLF se produit peu après un HO réussi, d'une cellule de desserte à une cellule cible, ou si un échec de HO se produit pendant la procédure de HO ; l'UE tente de rétablir la connexion avec une autre cellule que la cellule de desserte et la cellule cible [6],
- HO ping-pong : Un HO réussi, d'une cellule de desserte à une cellule cible, est suivi peu après, par un autre HO réussi vers la cellule de desserte initiale. [7],
- HO enchaîné (Chained HO, CH) : Un HO réussi, survient peu après un autre HO réussi.

A remarquer que dans l'équation (1) et l'équation (2), nous avons rencontré deux paramètres de réseau : le CIO et le HYS. Nous les appelons paramètres de mobilité/HO et ils font aussi l'objet d'études par la suite. Le CIO est utilisé pour déplacer les bordures des cellules. Le HYS est utilisé pour retarder les HOs, ce qui crée un effet d'hystérésis sur la procédure de HO. Nous notons que cette procédure est basée sur l'événement A3 du 3GPP [5]. Le CIO et le HYS sont utilisés dans la résolution de plusieurs problèmes pratiques (voir [6, 8]) parmi lesquels : équilibrer les charges et prévenir les HO ping-pongs et les RLFs.

Nous choisissons d'utiliser les paramètres de HO *par-cellule*, et non *par-paire-de-cellules,* afin d'éviter les zones où les UE sont prise dans des CH Continus (CCH). Nous donnons plus de détails dans une des sections suivantes.

Afin de réduire l'interférence vers leur voisines, certains cellules pourraient être interdites à transmettre. Une trame LTE a une durée de 10 ms. Elle est composée par 10 sous-trames de 1 ms chacune. Pour plus de détails, voir [9]. Nous appelons un Almost Blank Sub-frame (ABS) une sous-trame où il y a pas de données transmises. Cella permet de réduire l'interférence vers les cellules voisines. La technique est utilisée das les réseaux hétérogènes quand la couverture des cellules Pico est étendue. Afin de protéger les UEs forcés à s'attacher aux cellules Pico, nous pouvons employer les ABSs sur les cellules Macro.

FIGURE 3: Les fonctions SON

Figure 4: Les dimensions de coordination

Les fonctions SON et SONCO

Envisageant des augmentations de coûts pour les opérateurs de réseaux mobiles, 3GPP introduit à partir de la Release 8, les fonctionnalités de réseau auto-organisant (Self Organizing Network, SON), voir Figure 3. Une fonction SON est une boucle de contrôle qui règle les paramètres de réseau, afin d'améliorer ses indicateurs de performance (Key Performance Indicators, KPIs). Il y a principalement trois catégories de fonctions SON : auto-configuration, auto-optimisation et auto-guérison [2]. Les fonctions d'auto-configuration sont destinées à fournir des capacités plug-and-play aux équipements de réseau. Les fonctions d'auto-optimisation sont destinées à optimiser les réseaux quand ils sont opérationnels. Les fonctionnalités d'auto-guérison traitent de la résolution des pannes de réseau. Dans cette thèse, nous nous focalisons sur les fonctions d'auto-optimisation. Par la suite, SON fait référence à cette catégorie.

Nous définissons une instance (d'une fonction) SON comme une réalisation d'une fonction SON (ou de l'un de ses composants), exécutée sur une cellule. Les instances SON peuvent collaborer entre elles, surtout si elles sont des instances de la même fonction SON. Mais, ce n'est pas toujours le cas, notamment dans un environnent multi-fournisseur. Cela pourrait créer des conflits et baisser les performances de réseau [2]. Nous considérons deux types de conflits potentiels :

- conflits sur les paramètres : 2 instances SON ciblent le même paramètre.
- conflits sur les mesures : le paramètre ciblé par une instance SON influence les mesures d'une autre instance SON.

Deux dimensions doivent être considérées pour ces conflits (voir Figure 4) :

- La dimension instance SON : Envisagez un scénario avec des instances SON indépendantes appartenant à la même fonction SON, par exemple MLB. Les instances peuvent appartenir à deux fournisseurs différents, mais, même dans le cas d'un seul fournisseur nous pouvons être confrontés à des situations de ce genre.
- La dimension fonction SON : Envisagez un scénario avec des instances SON indépendantes appartenant à des fonctions SON différentes, par exemple MLB et MRO. Cela pourrait être le cas dans un environnent multi-fournisseur, mais pas uniquement.

Pour faire face à des conflits SON potentiels, la Release 10 du 3GPP a introduit la fonction de COordination SON (le SONCO)[3], [4]. Selon [4], les tâches d'un SONCO incluent (voir Figure 5) :

- éviter autant que possible les conflits SON (SONCO-A),
- détecter et diagnostiquer les conflits SON (SONCO-D),
- fournir des mécanismes de résolution des conflits si nécessaire (SONCO-R). Le SONCO-R reçoit toutes les demandes de mise à jour des paramètres.

Nous définissons une instance (d'une fonction) SONCO-A/D/R comme une réalisation d'une fonction SONCO-A/D/R (ou de l'un de ses composants).

En cas de conflit, le SONCO-R doit savoir quand il doit favoriser une instance SON ou une autre. Par exemple, disons que un SONCO-R reçoit deux demandes opposées : l'une d'une

Figure 5: Les composants du SONCO

FIGURE 6: Réseau Hétérogène

instance MLB et l'autre d'une instance MRO. La question est : laquelle d'entre elles il acceptera ? Nous proposons que les demandes de mise à jour comprennent l'information sur leur criticité. Cette information nous permettra d'aborder judicieusement un conflit en faveur de la demande la plus critique. Cella assure une certaine équité entre les fonctions SON. Plus de détails sur la façon dont l'équité est atteinte seront présentés plus tard. Donc, nous proposons que la demande de mise à jour soit une valeur réelle en adoptant la convention suivante :

- $u > 0$ est une demande d'augmentation. Plus $|u|$ est grande, plus la demande est critique,
- $u < 0$ est une demande de réduction. Plus $|u|$ est grande, plus la demande est critique,
- $u = 0$ est une demande de maintenir la valeur.

3. Diagnostiquer les conflits SON

Typiquement, un mécanisme de dépannage est composé par 3 étapes : la détection de défaut, le diagnostic et le déploiement de solution. Dans cette section, nous nous concentrons sur la deuxième étape (SONCO-D). Diagnostiquer les conflits sur les paramètres est simple. Une simple vérification, s'il y a ou pas des demandes opposées ciblant le même paramètre, pourrait être suffisante. Les conflits sur les mesures sont plus difficiles à diagnostiquer, donc nous nous concentrons sur ces conflits.

Le SONCO-D est chargé d'identifier lesquelles des fonctions SON (et potentiellement lesquels de leurs paramètres) sont à l'origine des conflits. Dans un environnement multi-fournisseur l'information sur l'algorithme utilisé par les SONs, en général, n'est pas partagée, c'est-à-dire que les fonctions SON sont considérées comme des *boîtes noires*. Quelques travaux proches peuvent être trouvés dans la littérature. Dans [10], [11, 12]et [13] une fonction d'*undo* est appliquée au cas où un défaut survienne après une mise à jour des paramètres de réseau. Dans [14] les auteurs utilisent des alarmes soulevées par les fonctions SON pour déclencher d'autres fonctions SON. D'autres travaux considèrent connu l'algorithme à l'intérieur des fonctions SON, c'est-à-dire que les fonctions SON sont considérées comme des *boîtes blanches*, par exemple [15].

Dans notre travail, nous considérons les fonctions SON comme des *boîtes noires*. Nous employons un mécanisme de dépannage automatisé, basé sur le Classificateur Naïf Bayésien (NBC) [16] car il a fait preuve d'être une bonne technique de diagnostic pour les réseaux mobiles [17, 18, 11, 19, 20, 12]. Dans ce travail, nous utilisons le NBC pour identifier les erreurs de configuration (mauvais réglages) des fonctions SON.

La contribution de ce travail est résumée comme suit : nous proposons une solution SONCO-D où nous utilisons le NBC en considérant les fonctions SON comme des *boîtes noires* ; nous fournissons une étude de cas dans un scénario basé sur un réseau hétérogène, en utilisant 3 fonctions SON : MLB via Cell Range Expansion (CRE), MRO et eICIC.

La description du système

Considérons un groupe (*cluster*) de cellules constitué par une cellule Macro et $N-1$ cellules Pico sous sa couverture destinées à augmenter la capacité (voir Figure 6). Nous indexons la cellule Macro avec $n = 1$ et les cellules Pico avec $\mathcal{N} = \{2, .., N\}$.

Soient le CIO, le HYS et le nombre d'ABS, les $K = 3$ paramètres de réseau qui nous intéressent, indexés (dans cet ordre) par $\mathcal{K} = \{1, .., K\}$. Le CIO et le HYS sont deux paramètres utilisés pour le management de la mobilité, comme présentés dans l'équation (1) et l'équation(2). L'ABS établit le nombre de sous-trames où la cellule Macro ne transmet pas des données, afin de réduire l'interférence vers les cellules voisines [6].

Nous employons $Z = 3$ fonctions SON : la fonction (MLB via) CRE, la fonction MRO et la fonction eICIC indexées (dans cet ordre) par $\mathcal{Z} = \{1, ..., Z\}$. Nous considérons qu'elles sont synchronisées, c'est-à-dire qu'elles envoient périodiquement les demandes de mise à jour de paramètres, à la fin d'un intervalle de temps $T = 5$min. Toutes les demandes sont acceptées et appliquées immédiatement dans le réseau (par le SONCO-R). En bref, à un moment donné (on fait abstraction de l'index temporel), on a :

MLB via CRE (on l'appelle simplement CRE) prend comme mesures d'entrée le vecteur de charges (LD) de toutes les cellules $M_{1,LD}, ..., M_{N,LD}$. Sa tâche est de réduire la charge maximale (parmi toutes les cellules), c'est-à-dire d'éviter d'avoir des cellules surchargées. Soit :

$$M_{LD} = \max_{n \in \{2,...,N\}} M_{n,LD}. \tag{3}$$

Dans ce but, CRE crée des demandes de mise à jour visant le CIO des cellules Pico (augmentation/ diminution /maintien). Les réglages du CRE sont les seuils utilisés pour déclencher les demandes de mise à jour des paramètres ($\mathbb{T}^H_{LD}$ et $\mathbb{T}^L_{LD}$). Soit Δ_{CIO} la taille de pas pour le CIO. Pour simplifier, nous la considérons comme un réglage du CRE, même si ce n'est pas le cas.

MRO prend comme mesures d'entrée le pourcentage de HOs trop tard (TL) $M_{2,TL}, ..., M_{N,TL}$ et de HOs Ping Pong (PP) $M_{2,PP}, ..., M_{N,PP}$ (par rapport à toutes les HOs) originaires uniquement des cellules Pico. Nous ignorons ceux originaires des cellules Macro parce que, en général, ils sont beaucoup moins fréquents. La tâche du MRO est de réduire les moyennes des deux, sur toutes les cellules. Soit :

$$M_{TL} = \sum_{n \in \{2,...,N\}} M_{n,TL}/N \text{ et } M_{PP} = \sum_{n \in \{2,...,N\}} M_{n,PP}/N. \tag{4}$$

Dans ce but, MRO crée des demandes de mise à jour visant seulement le HYS des cellules Pico (augmentation/ diminution /maintien). Les instances MRO pourraient également cibler le CIO, mais nous n'activons pas cette option (dans cette section) afin d'éviter les conflits sur les paramètres. Les réglages du MRO sont les seuils utilisés pour déclencher les demandes de mise à jour des paramètres ($\mathbb{T}^{H/L}_{TL}$ et $\mathbb{T}^{H/L}_{PP}$). Soit Δ_{HYS} la taille de pas pour le HYS. Pour simplifier, nous la considérons comme un réglage du MRO, même si ce n'est pas le cas.

eICIC prend comme mesures d'entrée le rapport de débits (TR), plus précisément le rapport entre le débit moyen des UEs Macro et le débit moyen des tous les UEs protégés : $M_{1,TR}$. Les UEs protégés sont tous les utilisateurs attachés à une cellule Pico, qui seront attachés à la cellule Macro si les paramètres CIO, HYS et TTT étaient nuls. Soit :

$$M_{TR} = M_{1,TR} \tag{5}$$

Son objectif est de maintenir ce rapport entre certaines limites afin de garantir une certaine équité entre les débits des UEs. Dans ce but, eICIC crée des demandes de mise à jour visant l'ABS sur la cellule Macro (augmentation/ diminution /maintien). Les réglages de l'eICIC sont les seuils utilisés pour déclencher les demandes de mise à jour des paramètres ($\mathbb{T}^H_{TR}$ et $\mathbb{T}^L_{TR}$). Soit Δ_{ABS} la taille de pas pour le nombre d'ABS. Pour simplifier nous la considérons comme un réglage de l'eICIC, même si ce n'est pas le cas.

Nous considérons que toutes les demandes de mise à jour sont acceptées (par le SONCO-R) et immédiatement appliquées dans le réseau. L'indicateur de criticité n'a pas d'impact dans cette section. L'entrée du SONCO-D est constituée par toutes les composantes des $I = 4$ mesures d'intérêt : LD, TL, PP et TR indexées (dans cet ordre) par $\mathcal{I} = \{1, .., I\}$.

Figure 7: SON interdépendance

Les conflits SON

Les fonctions SON regardent les mesures et en conséquence mettent à jour les paramètres de réseau à la fin de chaque intervalle de temps $t \in \mathbb{N}$ (de durée T). Nous utilisons $I = 4$ symptômes où chacun est basé sur l'une des $I = 4$ mesures. Nous définissons les symptômes comme le pourcentage de temps où la moyenne de la mesure (correspondante) est supérieure à un seuil :

$$S_i(t) = \frac{100}{a_i} \sum_{j=0}^{a_i-1} \mathbb{I}\left(\frac{1}{b_i} \sum_{j=0}^{b_i-1} M_i(t-i-j) > c_i \right) \tag{6}$$

où $\mathbb{I}$ est la fonction indicateur ($\mathbb{I}(\texttt{true}) = 1$, $\mathbb{I}(\texttt{false}) = 0$). Les valeurs a_i, b_i et c_i ($\forall i \in \{LD, TL, PP, TR\}$) sont définies plus tard.

Nous définissons $\mathbb{S}_i$, $\forall i \in \{LT, TL, PP, TR\}$ comme la performance ciblée par rapport aux symptômes. Au moment t nous disons qu'il y a un défaut si $\exists i \in \{LT, TL, PP, TR\}$ tel que $S_i(t) > \mathbb{S}_i$.

Selon la Figure 7 nous pouvons voir que le CRE peut avoir un impact sur le MRO et l'eICIC, et que l'eICIC peut avoir un impact sur le CRE. Ainsi, nous définissons un ensemble de causes de 1^{er} *ordre* :$\mathcal{C}_1 = \{\text{CRE}, \text{eICIC}\}$, avec le cardinal $|\mathcal{C}_1| = 2$. Nous notons que le problème peut provenir d'un des différents réglages de la fonction SON, comme les seuils de déclenchement ($\mathbb{T}$) ou la taille de pas pour les paramètres ciblé (Δ). Ainsi, nous définissons un ensemble de causes de $2^{éme}$ *ordre* : $\mathcal{C}_2 = \{\text{CRE}, \text{eICIC}\} \times \{\mathbb{T}, \Delta\}$, avec le cardinal $|\mathcal{C}_2| = 4$. Ensuite, si nous détaillons plus, nous pouvons également cibler d'identifier le degré d'altération (l'éloignement de la bonne valeur) du réglage (SON) problématique : faible (low, `l`), moyenne (`m`), haute (`h`). Ainsi, nous définissons un ensemble de causes de $3^{éme}$ *ordre* : $\mathcal{C}_3 = \{\text{CRE}, \text{eICIC}\} \times \{\mathbb{T}, \Delta\} \times \{\texttt{l}, \texttt{m}, \texttt{h}\}$, avec le cardinal $|\mathcal{C}_3| = 12$. Dans ce travail, nous comparons le diagnostic pour les 3 ensembles de causes.

Le diagnostic

Selon la règle de Bayes nous avons, $\forall c \in \mathcal{C}_l$ (pour toute $l \in \{1, 2, 3\}$) :

$$\mathbb{P}(C = c|S = s) = \frac{\mathbb{P}(S = s|C = c)\,\mathbb{P}(C = c)}{P(S = s)} \tag{7}$$

où $s = (s_i)_{i \in \{LD, TL, PP, TR\}}$ représente l'ensemble de valeurs observées en ce qui concerne les symptômes.

À noter que $P(S = s)$ est le même pour toute cause. Ainsi, afin d'identifier la cause la plus probable, nous ne devons pas le calculer forcément. Donc nous nous concentrons uniquement sur le numérateur. $\mathbb{P}(C = c)$ est la probabilité d'apparition de la cause c, et peut être facilement obtenue. Le calcul/estimation du $\mathbb{P}(S = s|C = c)$ peut être très difficile et peu pratique, surtout si nous avons beaucoup de symptômes. Pour cette raison, nous supposons que i) les symptômes sont indépendants et ii) il peut y avoir une seule cause à la fois. Même si en réalité, c'est rarement le cas, des résultats significatifs ont été obtenus en vertu de ces hypothèses (e.g. [17]). La simplicité du modèle a des avantages évidents. Par conséquent l'équation (7) devient :

$$\mathbb{P}(C = c|S = s) = \frac{\prod_i \mathbb{P}(S_i = s_i|C)\,\mathbb{P}(C = c)}{\mathbb{P}(S = s)} \tag{8}$$

L'équation ci-dessus représente le NBC, et il sert de modèle de diagnostic. La méthode d'inférence de la cause donne simplement la cause la plus probable selon l'équation (8).

Les résultats des simulations

FIGURE 8: La topologie de réseau

SON	CRE			eICIC		MRO
Réglage	$\mathbb{T}^H_{LD}$ [rapport]	$\mathbb{T}^L_{LD} =$ [rapport]	Δ_{CIO} [dB]	$\mathbb{T}_{TR}$ [rapport]	Δ_{nABS} [rapport]	
REF.	0.85	$\mathbb{T}^H_{LD} - 0.1$	1	2.1	1	$\mathbb{T}^H_{TL} = 0.1$ [rapport], $\mathbb{T}^H_{PP} = 0.2$ [rapport], $\Delta_{HYS} = 1$ [dB]
CRE-$\mathbb{T}$-l	0.80		1	2.1	1/40	
CRE-$\mathbb{T}$-m	0.75		1	2.1	1/40	
CRE-$\mathbb{T}$-h	0.70		1	2.1	1/40	
CRE-Δ-l	0.85		3	2.1	1/40	
CRE-Δ-m	0.85		5	2.1	1/40	
CRE-Δ-h	0.85		6	2.1	1/40	
eICIC-$\mathbb{T}$-l	0.85		1	1.9	1/40	
eICIC-$\mathbb{T}$-m	0.85		1	1.7	1/40	
eICIC-$\mathbb{T}$-h	0.85		1	1.5	1/40	
eICIC-Δ-l	0.85		1	2.1	3/40	
eICIC-Δ-m	0.85		1	2.1	5/40	
eICIC-Δ-h	0.85		1	2.1	6/40	

TABLE 1: Le sommaire des scénarios

Nous analysons la faisabilité d'utiliser le NBC pour le design du SONCO-D. Pour cela, nous employons un réseau avec 9 cellules Macro. Sous une des cellules Macro (nous l'appelons "M"), nous mettons au hasard 2 cellules Pico (nous les appelons "P1" et "P2"). Les fonctions SON sont appliquées sur le groupe de cellules formé par M, P1 et P2 (voir la zone en pointillé dans la Figure 6).

Nous considérons un trafic FTP. Les UEs arrivent au hasard dans le réseau. Le taux d'arrivée des UEs est Λ_G[UEs/s] sur tout le réseau, et supplémentaire Λ_{HS}[Mb/s] autour des cellules Pico. Le taux d'arrivée est soumis à une variation périodique (2 heures), $\forall j \in \{G, HS\}$:

$$\Lambda_j\left(\text{temps}[h]\right) = \lambda_j\left(0.9 - 0.1 \cdot \sin\left(\pi \cdot \text{temps}[h]\right)\right)$$

où temps[h] est le temps mesuré en heures et λ_i est le taux d'arrivée moyen.

Nous considérons 13 scénarios avec différents réglages des fonctions SON. L'un d'eux est le scénario de référence (REF) et les 12 autres correspondent à une des causes comprises dans l'ensemble $\mathcal{C}_3$ (et implicitement dans $\mathcal{C}_2$ et $\mathcal{C}_1$). Pour plus de détails, voir la Table 1. Nous utilisons les paramètres de symptômes présentés dans la Table 2.

Nous utilisons 1000 échantillons de chaque scénario sauf REF, étiquetés manuellement , pour entraîner le NBC. Ensuite, considérons que nous nous concentrons sur l'ensemble de causes d'ordre i : $\mathcal{C}_i$. Nous obtenons $\mathbb{P}\left(S|C\right)$ et $\mathbb{P}\left(C\right)$ en utilisant des histogrammes.

Puis, nous utilisons 400 autres échantillons, non étiquetés, et nous testons le SONCO-D. La probabilité d'identification correcte de la faute est présentée dans la Figure 9 pour tous les trois *ordres*. Nous voyons une performances : supérieure à 91,7% pour la détection de la fonction SON problématique (cause de 1^{er} ordre), supérieure à 82,5% si nous voulons également détecter le réglage problématique (cause de $2^{éme}$ ordre) et supérieure à 52% dans le cas où nous voulons détecter le degré d'altération du réglage (cause de $3^{éme}$ ordre). Comme attendu, il y a moins d'erreurs pour détecter seulement la fonction SON problématique que pour l'identification complète du problème : fonction SON problématique, le réglage problématique et son degré d'altération.

FIGURE 9: Le pourcentage de diagnostic correct

Symptôme : $i =$	LD	TL	PP	TR
$\mathbb{S}_i$	1.0%	3.5%	4.0%	1.7%
a_i	1440			
b_i	3	12	12	3
c_i	0.85	0.08	0.015	2.25

TABLE 2: Les paramètres des symptômes

FIGURE 10: Les interactions SONCO-R $\leftrightarrow$ SON

4. La résolution de conflits SON

Dans la littérature, le problème de la résolution des conflits SON (SONCO-R) est abordé de plusieurs façons. Certaines études proposent une élaboration jointe des fonctions SON [21, 22, 23, 24, 25]. Une autre approche considère des fonctions SON conçues indépendamment, dont les algorithmes d'optimisation sont connus [15, 26, 27]. Ainsi, les fonctions SON sont considérées comme des *boîtes blanches*. La troisième approche que nous avons identifiée considère les fonctions SON conçues indépendamment, dont les algorithmes d'optimisation sont inconnus [28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39]. Ainsi, les fonctions SON sont considérées comme des *boîtes noires*.

Dans notre travail, nous considérons les fonctions SON comme des *boîtes noires*. En plus, nous supposons qu'ils sont sans mémoire. Contrairement à la littérature existante, nous proposons d'utiliser les informations sur l'impact de nos décisions passées. Le SONCO-R apprend les décisions optimales de coordination en interagissant avec les fonctions SON. Une technique très efficace et bien connue pour ça est l'apprentissage par renforcement (Reinforcement Learning, RL) [40]. RL a été largement utilisé dans le cadre d'auto-optimisation [41, 23].

Nous employons RL et nous décrivons les compromis qui doivent être faits afin d'obtenir des solutions passables à l'échelle. Nous fournissons la description du système, nous présentons le SONCO-R comme un Processus de Décision Markovien (MDP) et la technique d'approximation et estimation des fonctions avec l'algorithme RL associé. Nous analysons plusieurs scénarios, en employant différentes approximations.

La description du système

Afin de simplifier les notations nous proposons la convention suivante : pour toute variable X (paramètre, demande de mise à jour, action, etc.), si nous l'indexons avec l'ensemble $\mathcal{I}$ le sens est : $X_{\mathcal{I}} = (X_i)_{i \in \mathcal{I}}$.

Considérons un segment de réseau composé par C clusters (c'est-à-dire groupes disjoints) de cellules (voisines), indexe par $\mathcal{C} = \{1, ..., C\}$ (voir Figure 10). Nous nous concentrons sur un seul cluster, considérons le cluster $c \in \mathcal{C}$, comme s'il était indépendant des autres clusters. Soit N le nombre des cellules appartenant à ce cluster, indexé par $\mathcal{N} = \{1, ..., N\}$. Nous définissons $\bar{\mathcal{N}}_n$, $\forall n \in \mathcal{N}$, comme l'ensemble contenant la cellule n et ses voisines (seulement celles appartenant au cluster c). Chaque instance SON envoie des demandes de mise à jour ciblant les paramètres de la cellule hôte. Nous considérons des demandes vides (`void`) pour le cas où l'instance SON n'est pas intéressée d'ajuster certains paramètres.

Considerons que chaque fonction SON ajuste K paramètres (CIO, HYS, le tilt de l'antenne, etc) sur chaque cellule. Nous les indexons par $k \in \mathcal{K} = \{1, ..., K\}$.

À une instance de temps donnée (on fait abstraction de l'index temporel), soit :

- $P_{n,k}$ la valeur du paramètre $k \in \mathcal{K}$ de la cellule $n \in \mathcal{N}$ et $\mathcal{P}_{n,k}$ l'ensemble de valeurs possible de $P_{n,k}$. Soit $P = P_{\mathcal{N},\mathcal{K}}$ et $\mathcal{P} = \mathcal{P}_{\mathcal{N},\mathcal{K}}$.
- $U^z_{n,k}$ la valeur de la demande de mise à jour de l'instance de la fonction SON $z \in \mathcal{Z}$ exécutée sur la cellule $n \in \mathcal{N}$ ciblant le paramètre $k \in \mathcal{K}$ ($P_{n,k}$) et $\mathcal{U}^z_{n,k} \subseteq [-1; 1] \cup \{\texttt{void}\}$ l'ensemble

FIGURE 11: Le *kernel* de transition

de valeurs possibles de $U^z_{n,k}$. Soit $U = U^{\mathcal{Z}}_{\mathcal{N},\mathcal{K}}$ et $\mathcal{U} = \mathcal{U}^{\mathcal{Z}}_{\mathcal{N},\mathcal{K}}$.

Le SONCO-R

Au lieu d'exécuter directement les changements de paramètres souhaités, les instances SON envoient leurs demandes de mise à jour à une instance SONCO-R qui décide si elles sont acceptées (et immédiatement exécutées) ou refusées. Les instances SON sont considérées comme des *boîtes noires* par le SONCO-R, c'est-à-dire qu'il ne connaît pas les algorithmes à l'intérieur des boîtes. Il connaît seulement les demandes de mise à jour actuelles (U) et la configuration actuelle des paramètres de réseau (P). La tâche du SONCO-R est de trouver une résolution juste des conflits, afin d'assurer une certaine équité entre les instances SON. Nous voulons que dans le cas où nous sommes confrontés à une situation de conflit, le SONCO-R choisisse quelles demandes seront acceptées, telles que, lors de la prochaine étape, la criticité maximale parmi toutes les demandes soit minimale.

Les instances SON sont synchronisées. Elles envoient les demandes de mise à jour simultanément, à la fin d'une fenêtre temporelle périodique de taille T. Le SONCO-R est également synchronisé avec les instances SON. Il reçoit les demandes de mise à jour de toutes les instances SON et décide quelles demandes sont acceptées et lesquelles sont refusées. Nous employons une approche de RL car elle permet d'utiliser les informations sur les effets de décisions antérieures. RL est fondée sur un MDP décrit ensuite.

Le Processus de Décision Markovien Nous décrivons la dynamique du système comme une interaction entre la configuration des paramètres actuels P, les demandes de mise à jour des fonctions SON U et la décision du SONCO-R A. Dans ce but, nous les interprétons comme des processus stochastiques sur un espace probabiliste, gouverné par le MDP suivant (voir Figure 11) :

L'espace d'états $\mathcal{S}$: Un état $s = (p, u) \in \mathcal{S}$ a deux composants : la configuration des paramètres de réseau ($p = p_{\mathcal{NK}}$) et les demandes de mise à jour ($u = u^{\mathcal{Z}}_{\mathcal{NK}}$). Nous définissons $S_{(t)} = \left(P_{(t)}, U_{(t)}\right)$ le processus de l'état où $t \in \mathbb{N}$ est l'index temporel.

L'espace d'actions $\mathcal{A}$: Une action $a \in \mathcal{A}$ est un vecteur de taille NK, c'est-à-dire $a = a_{\mathcal{NK}}$. Le composant $a_{n,k}$ de l'action a permet d'augmenter/ diminuer la valeur du paramètre k de la cellule n seulement s'il existe au moins une demande de le faire, et si la valeur n'est pas déjà au maximum/minimum dans l'ensemble des valeurs possibles $\mathcal{P}_{n,k}$. Concrètement la prochaine valeur du paramètre k pour la cellule n est :

$$p^{'}_{n,k} = \begin{cases} \min\left\{x \in \mathcal{P}_{n,k} : x > p_{n,k}\right\}, & \text{si } a_{n,k} > 0,\ p_{n,k} \neq \max\mathcal{P}_{n,k} \text{ et } \exists z \in \mathcal{Z} \text{ t.q. } u^z_{n,k} > 0 \\ \max\left\{x \in \mathcal{P}_{n,k} : x < p_{n,k}\right\}, & \text{si } a_{n,k} < 0,\ p_{n,k} \neq \min\mathcal{P}_{n,k} \text{ et } \exists z \in \mathcal{Z} \text{ t.q. } u^z_{n,k} < 0 \\ p_{n,k}, & \text{autrement} \end{cases} \tag{9}$$

Nous pouvons voir dans l'équation (9) que le vecteur p', représentant la configuration future des paramètres de réseau, est une fonction déterministe de la configuration courante des paramètres p, des demandes de mise à jour u et de l'action a, c'est-à-dire :

$$p' = g\left((p,u),a\right) = \left(g_{n,k}\left(\left(p_{n,k}, u^{\mathcal{Z}}_{n,k}\right), a_{n,k}\right)\right)_{n,k} \tag{10}$$

où les fonctions $g : \mathcal{P}\times\mathcal{U}\times\mathcal{A} \to \mathcal{P}$ et $g_{n,k} : \mathcal{P}_{n,k}\times\mathcal{U}_{n,k}\times\mathcal{A}_{n,k} \to \mathcal{P}$ sont détaillées dans l'équation (9).

Le *kernel* de transition $\mathcal{T}\left(s'|s,a\right)$: décrit la probabilité que $s' = (p', u') \in \mathcal{S}$ est l'état suivant si $s = (p, u) \in \mathcal{S}$ est l'état courant et a est l'action courante :

$$\mathcal{T}\left(s'|s,a\right) = \mathbb{P}\left(S_{t+1} = s' | S_t = s, A_t = a\right).$$

Le regret : $\bar{r}(s,a)$ est l'espérance mathématique du regret associé à la paire état-action (s,a). Nous introduisons $\left(R_{(t)}\right)_t$ le processus des regrets instantanés. En conséquence $\forall t \in \mathbb{N}$:

$$\mathbb{E}\left[R_{(t+1)}|S_{(t)}=s, A_{(t)}=a\right]=\bar{r}\left(s,a\right)$$

Nous supposons aussi que :

$$R_{(t)}=\rho\left(U_{(t)}\right)=\sum_{i\in\mathcal{I}}\rho_i\left(u\right) \tag{11}$$

pour des fonctions $\rho, \rho_i : \mathcal{U} \to \mathbb{R}$ pour tout $i \in \mathcal{I}$.

Les fonctions valeur Pour toute politique π, afin de mesurer ses performances, nous introduisons la *fonction valeur-état* (V^π) et la *fonction valeur-action* (Q^π) :

$$V^\pi(s)=\mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t R_{(t+1)}\,|S_{(0)}=s\right]\ , \tag{12}$$

$$Q^\pi(s,a)=\mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t R_{(t+1)}\,|S_{(0)}=s, A_{(0)}=a\right]. \tag{13}$$

où $0 \leq \gamma < 1$ est le facteur de dévaluation et $\mathbb{E}_\pi$ est l'espérance pour le cas où la politique suivie est π.

Selon [42] $V^*(s)$ est la *valeur-état* optimale de l'état $s \in \mathcal{S}$, si elle représente la valeur minimale :

$$V^*\left(s\right)\leq V^\pi\left(s\right) \tag{14}$$

pour toute politique π, étant donné que le processus commence dans l'état s. De la même façon, $Q^*(s,a)$ est la *valeur-action* optimale si $Q^*(s,a) \leq Q^\pi(s,a)$pour toute politique π.

La simplification de la fonction valeur-action Le théorème suivant permet de simplifier l'équation (13). Nous définissons $\hat{r}_i : \mathcal{P} \to \mathbb{R}$, pour tout $i \in \mathcal{I}$, où :

$$\hat{r}_i\left(p\right)=\mathbb{E}\left[\rho_i\left(U_{(t)}\right)|P_{(t)}=p\right].$$

Théorème 1. *Pour toute politique π il existe certaines fonctions $W_i^\pi : \mathcal{P} \to \mathbb{R}$, telles que pour tout $(s,a) \in \mathcal{S} \times \mathcal{A}$:*

$$Q^\pi\left(s,a\right)=\sum_{i\in\mathcal{I}}W_i^\pi\left(g\left(s,a\right)\right). \tag{15}$$

Aussi, pour tout $i \in \mathcal{I}$, W_i^π résolut l'équation du point fixe :

$$W_i^\pi\left(p\right)=\hat{r}_i\left(p\right)+\gamma\sum_{u\in\mathcal{U}}\mathbb{P}\left[U_{(t)}=u|P_{(t)}=p\right]\cdot\sum_{a\in\mathcal{A}}\pi\left(\left(p,u\right),a\right)\cdot W_i^\pi\left(g\left(\left(p,u\right),a\right)\right). \tag{16}$$

Démonstration. Voir Annexe 4.1. □

Remarque 1. *Il est connu que la politique optimale π^* est une fonction déterministe, c'est-à-dire que $\pi^* : \mathcal{S} \to \mathcal{A}$ et nous avons selon [40] : $\pi^*(s) = arg\ min_a Q^*(s,a)$.*

Approximation linéaire

Même si le Théorème 1 permet de simplifier l'équation (13) à l'ensemble de fonctions avec un domaine réduit $\mathcal{P}$, la complexité grandis de façon exponentielle avec le nombre de cellules N. En conséquence nous effectuons une *Approximation Linéaire de Fonction* [40], au détriment de la performance, mais qui permet de réduire l'espace d'états de $I \cdot |\mathcal{P}|$ à $\sum_i J_i$.

Nous faisons l'approximation de chaque $W_i^*(p)$ avec $\bar{W}_i^*(p)$, qui dépend de p par une fonction linéaire de quelques *descripteurs* $F_i(p)$. Soit J_i le nombre de *descripteurs* , c'est-à-dire $F_i : \mathcal{P} \to \mathbb{R}^{J_i}$. Le choix des *descripteurs* est très important et il est spécifique pour chaque cas. Nous fournissons des exemples plus tard. Donc $\bar{W}_i^*$ est donné par :

$$\bar{W}_i^*\left(p\right)=\left\langle\theta_i^*, F_i\left(p\right)\right\rangle \tag{17}$$

où $\langle x,y\rangle$ est le produit scalaire de x et y, et θ_i^* doit être choisi tel qu'il minimise la différence entre $\bar{W}_i^*(p)$ et $W_i^*(p)$.

Algorithme 1 SONCO-R

Fonction Init :
Initialiser,pour toute $i \in \mathcal{I}$, $\theta_i = 0^{J_i}$
Fonction SONCO-R :
Observer la configuration courante des paramètres p et des demandes u, calculez le regret $r = \rho(u)$,
Calculer $\bar{a} = \arg\ \min_{a \in \mathcal{A}} \sum_{i \in \mathcal{I}} \langle \theta_i, F_i(g((p,u),a)) \rangle$
Mettre à jour $\theta_i = \theta_i + \alpha \left[r + \gamma \langle \theta_i, F_i(g((p,u),\bar{a})) \rangle - \langle \theta_i, F_i(p) \rangle \right] F_i(p)$
Choisir l'action courante a en utilisant la politique ϵ-glouton :
$\pi((p,u),\{a\}) = (1-\epsilon)\,\mathbb{I}_{\{a=\bar{a}\}} + \frac{\epsilon}{3^{NK}}$
Exécuter l'action a.

L'apprentissage par renforcement

Nous ne pouvons pas calculer directement θ_i^* parce que nous n'avons qu'une connaissance partielle sur le *kernel*. Au lieu de cela, (d'après [40]) nous pouvons utiliser la récursion résumée dans l'Alg. 1 qui est basée sur l'apprentissage par renforcement. La **Fonction Init** doit être appelée pour l'initialisation de l'algorithme et la **Fonction SONCO-R** doit être appelée à chaque fois après avoir reçu les demandes des instances SON. À remarquer que nous ne pouvons pas nous attendre à une convergence de l'algorithme parce que nous utilisons un pas de taille fixe α. Au plus, nous pouvons espérer d'avoir une convergence dans la moyenne. Même cela ne peut pas être garanti pour toutes les approximations. En tout cas, ce sera le cas dans notre scénario selon les résultats. À remarquer que $100 \cdot \epsilon\%$ du temps nous explorons et $100 \cdot (1-\epsilon)\%$ du temps nous exploitons.

Le regret multi-dimension Dans cette section, au lieu de considérer que le regret est une somme de I composants, nous supposons que le regret est en fait un vecteur de taille D. Ainsi, pour toute instance temporelle nous pouvons écrire le regret instantané dans l'Algorithme 1 comme un vecteur $r = (r_d)_{d \in \mathcal{D}} \in \mathbb{R}^D$ ($\mathcal{D} = \{1,..,D\}$). Nous considérons encore que le regret r est une fonction des demandes de mise à jour (u) : $r = \rho(u)$ et $r_d = \rho_d(u)$.

Évidemment, pour $D = 1$ nous sommes dans le cadre ci-dessus. Pour $D > 1$ le cadre est toujours valable jusqu'à un certain point. L'extension des définitions des *fonctions valeur* V dans (12) et Q dans (13) est évidente. Elles deviennent tout simplement des vecteurs de taille D. Le premier problème est la politique optimale dans la Remarque 1 parce que l'*arg min* n'a plus de sens pour une fonction Q multi-dimension. Pour résoudre ce problème, nous définissons une méthode pour comparer des vecteurs :

Considérons deux vecteurs de taille D : $x, y \in \mathbb{R}^D$. Soient $x', y' \in \mathbb{R}^D$ les vecteurs x et y, respectivement, triés descendant, c'est-à-dire $x'_1 > x'_2 > ....$ Si $x'_1 > y'_1$ nous disons que x est plus grand que y . Si $x'_1 < y'_1$ nous disons que x est plus petit que y. Si $x'_1 = y'_1$ nous passons à la comparaison de x'_2 et y'_2 de la même manière, et ainsi de suite. Si $x'_d = y'_d$ pour tout $d \in \mathcal{I}$, nous disons que les deux vecteurs sont *égaux* ou *équivalents* (à remarquer que ça n'est pas la même chose que $x = y$). Pour $D = 1$ cela est cohérent avec les règles de comparaison des scalaires.

Afin de simplifier les choses, pour $D > 1$ nous considérons $\gamma = 0$, c'est-à-dire que la politique est myope. En utilisant le même raisonnement que dans le cas unidimensionnel nous nous retrouvons avec l'estimation de $W^*(p)$ qui est un vecteur de taille D. Donc nous approximons chaque composent $W_d^*(p)$ en utilisant un ensemble de J_d *descripteurs* $F_d(p) : \mathcal{P} \to \mathbb{R}^{J_d}$, c'est-à-dire $\bar{W}_d^*(p) = \langle \theta_d^*, F_d(p) \rangle$. L'algorithme que nous proposons pour une instance SONCO-R est résumé dans Alg. 2.

Si $D > 1$ le raisonnement ne tient pas pour $\gamma > 0$ en raison de non-linéarité des opérations min/max.

Algorithme 2 SONCO-R pour le regret multi-dimension

Fonction Init :
Initialiser, pour toute $d \in \mathcal{D}$, $\theta_d = 0^{J_d}$
Fonction SONCO-R :
Observer la configuration courante des paramètres p et des demandes u, calculez le regret $r = \rho(u)$,
Computer $\bar{a} = \arg\ \min_{a \in \mathcal{A}} \langle \theta_d, F_d(g((p,u),a)) \rangle_{d \in \mathcal{D}}$
Mettre à jour $\theta_d = \theta_d + \alpha \left[r + \gamma \langle \theta_d, F_d(g((p,u),\bar{a})) \rangle_{d \in \mathcal{D}} - \langle \theta_d, F_d(p) \rangle_{d \in \mathcal{D}} \right] F_d(p)$
Choisir l'action courante a en utilisant la politique ϵ-glouton :
$\pi((p,u),\{a\}) = (1-\epsilon)\,\mathbb{I}_{\{a=\bar{a}\}} + \frac{\epsilon}{3^{NK}}$
Exécuter l'action a.

Les scénarios d'application

Pour valider notre cadre, nous analysons plusieurs scénarios. Au début, nous utilisons le regret unidimensionnel/scalaire pour deux cas correspondant aux types de conflits : sur les mesures et sur les paramètres. Pour chacun, nous fournissons des évaluations pour quand le regret peut ou ne peut pas être écrit comme une somme de plusieurs composants. Ces évaluations sont effectuées dans un réseau homogène qui ne contient que des cellules Macro.

Ensuite, nous utilisons le regret multi-dimension en mettant l'accent sur la résolution des conflits sur les paramètres. Ces évaluations sont effectuées dans un réseau hétérogène qui contient des cellules Macro et des cellules Pico.

Lorsque le regret peut être décomposé en une somme de sous-regrets, ou en un vecteur de sous-regrets, et que nous avons plusieurs composants W_i, nous disons que l'*apprenant* (*Leaner*) est *Distribué* (LD). Sinon, si le regret ne peut être divisé en sous-regrets alors nous avons un seul composant W_i et nous disons que l'*apprenant* (*Learner*) est *Centralisé* (LC).

Pour avoir un aperçu des scénarios, nous fournissons un résumé dans le Tableau 3 et les détails dans la suite.

Regret	Conflit sur	Apprenant	fonctions SON
Scalaire	mesures	LC	MLB
		LD	
	paramètres	LC	MLB, MRO
		LD	
Vecteur	paramètres	LC	MLB(via CRE), MRO, eICIC
		LD	

TABLE 3: Scénarios d'application

La résolution de conflits sur les mesures

Dans cette section nous abordons le problème de la coordination, notamment sur la dimension de l'instance. Nous utilisons 12 cellules avec une seule fonction SON, spécifiquement MLB, instanciée indépendamment sur chaque cellule. Cela a un impact sur la stabilité du réseau, c'est-à-dire que nous observons des changements inutiles de la configuration de paramètres. Le but de l'instance SONCO-R est d'éliminer cette instabilité en trouvant la configuration (état) avec le plus petit regret, en orientant le réseau vers cette configuration et après en évitant de s'éloigner d'elle.

Regret non-distribuable

Premièrement, nous considérons que le regret est un simple scalaire. Le plus simple design SONCO-R est le cas *optimal,* où il y a pas d'approximation et toutes les cellules forment un seul cluster, c'est-à-dire qu'elles sont toutes gouvernées par une seule instance SONCO-R. Dans

:/3.PHD/PhD Manuscris/graphs/SONCO$_R$/$VTCFall2014/res_regret$".eps

(a) Le regret moyen (fenêtre glissante de 48h)

:/3.PHD/PhD Manuscris/graphs/SONCO$_R$/$VTCFall2014/res_NoCIOupdperhour$".eps

(b) Le numéro moyen des changements de CIO [#/h] pendant les derniers 48h

:/3.PHD/PhD Manuscris/graphs/SONCO$_R$/$VTCFall2014/res_utlz$".eps

(c) La charge moyenne pendant les dernières 48h

FIGURE 12: Résultats : regret scalaire, non-distribuable (conflits sur les mesures)

ce cas F grandis de façon exponentielle avec le nombre de cellules. Pour régler ça, une des solutions est de diviser la zone d'intérêt en plusieurs sous-zones, dont chacune représente un cluster de cellules. Évidemment, cela est une solution *sous-optimale* parce que les clusters ne seront pas totalement indépendants. Nous évaluons 5 scénarios codés comme suit :

SFoff : les instances MLB sont désactivées,

SCoff : les instances MLB sont activées, SONCO-R activé

SC-LC : SONCO-R activé, toutes les cellules das un seule cluster,

SC6-LC : SONCO-R activé, 6 cellules par cluster,

SC3-LC : SONCO-R activé, 3 cellules par cluster.

Nous voyons dans la Figure 12a que le regret moyen est réduit lorsque le SONCO-R est activé, avec le plus gros gain pour le SONCO-R centralisé (SC-LC). Plus le système est distribué, plus le regret est grand. C'est le prix à payer pour le passage à l'échelle. La Figure 12b contient le maximum et la moyenne (sur toutes les cellules) des moyennes-temporelles des numéros de changements du CIO. Nous pouvons voir que le SONCO-R élimine la plupart des changements inutiles (73-84%). Dans la Figure 12c nous traçons le maximum et la moyenne (sur toutes les cellules) des moyennes-temporelles des charges ; nous pouvons voir que les instances MLB arrivent encore à accomplir leur tâche, en gardant la charge plus bas que le seuil ($\mathbb{T}^H_{LD} = 0.8$).

Regret distribuable

Dans cette section nous évaluons les avantages de pouvoir écrire le regret comme une somme de sous-regrets, notamment un par cellule. Chaque sous-regret R_n dépend uniquement des demandes de mise à jour ciblant la cellule n ($U^{\mathcal{Z}}_{n,\mathcal{K}}$). Sans aucune approximation, chaque W_n dépend de tous les paramètres de réseau $P_{\mathcal{N},\mathcal{K}}$. Nous disons que cette solution est *optimale*, c'est-à-dire qu'il n'y a aucune approximation, mails elle n'est pas passable à l'échelle. Pour résoudre ce problème nous considérons que W_n (qui correspond au regret n) ne dépend que des paramètres de la cellule n et ses voisines ($\forall n \in \mathcal{N}$). Pour faire ça, nous utilisons l'*agrégation des états* et nous nous retrouvons avec un F_n de taille $\left|\mathcal{P}_{\mathcal{N}_n,\mathcal{K}}\right|$, donc le passage à l'échelle de F est linéaire avec le nombre des cellules. Au cause des approximations, nous disons que cette solution est *sous-optimale*,

Nous évaluons 4 scénarios codés comme suit :

SFoff : les instances MLB sont désactivées,

SCoff : les instances MLB sont activées, SONCO-R activé

SC-LD : SONCO-R activé, toutes les cellules das un seul cluster, solution *optimale*,

SC-LD-SA : SONCO-R activé, toutes les cellules das un seul cluster, solution *sous-optimale*,

La Figure 13a montre que le regret moyen est réduit lorsque le SONCO-R est activé. La politique *sous-optimale* (SC-LD) converge vers le même regret que la politique *optimale* (SC-LD-SA). Dans la Figure 13b nous traçons le maximum et la moyenne (sur toutes les cellules) des

:/3.PHD/PhD Manuscris/graphs/SONCO$_R$/$PIMRC2014/res_regret$".eps

(a) Le regret moyen (fenêtre glissante de 48h)

:/3.PHD/PhD Manuscris/graphs/SONCO$_R$/$PIMRC2014/res_NoCIOupd$".eps

(b) Le numéro moyen des changements de CIO [#/h] pendant les derniers 48h

:/3.PHD/PhD Manuscris/graphs/SONCO$_R$/$PIMRC2014/res_utlz$".eps

(c) La charge moyenne pendant les dernières 48h

FIGURE 13: Résultats : regret scalaire, distribuable (conflits sur les mesures)

moyennes-temporelles des nombres de changements du CIO. Nous pouvons voir une baisse de 77-82% si SONCO-R est activé. Dans la Figure 13c nous traçons le maximum et la moyenne (sur toutes les cellules) des moyennes-temporelles des charges. Nous pouvons voir que les instances MLB arrivent encore a accomplir leur tâche comme la charge est plus bas que le seuil ($\mathbb{T}^H_{LD} = 0.8$).

La résolution de conflits sur les paramètres

Dans cette section nous abordons le problème de la coordination sur la dimension de la fonction. Nous utilisons 21 cellules avec deux fonctions SON, notamment MLB et MRO, instanciées sur chaque cellule. Les deux ciblent à régler le CIO. À part ça, MRO règle aussi le HYS. Le SONCO-R effectuera une résolution de conflit sur les paramètres, pour assurer un degré d'équité entre les deux fonctions. Comme dans la section précédente le problème ciblé est la taille du vecteur de *descripteurs* (F), c'est-à-dire le passage à l'échelle.

D'abord, nous proposons de négliger les paramètres qui ne sont pas une source importante de conflit. Dans notre cas nous faisons référence au HYS, c'est-à-dire que nous considérons que le HYS n'impacte pas de manière signifiante le MLB. Quand même, sans aucune approximation supplémentaire, nous avons toujours un grand nombre de *descripteurs* : $\left|\mathcal{P}_{\mathcal{N},\{CIO\}}\right|$. Il grandit exponentiellement avec le nombre de cellules. Dans la suite nous analysons les deux stratégies que nous avons utilisées dans la section précédente, par rapport à écrire le regret comme une somme de plusieurs composants.

Regret non-distribuable

Premièrement nous considérons que le regret est un simple scalaire. Le plus simple design SONCO-R est celui où toutes les cellules forment un seul cluster, c'est-à-dire qu'elles sont toutes gouvernées par une seule instance SONCO-R. Dans ce cas, F grandit de façon exponentielle avec le nombre de cellules. Par rapport à ça, une des solutions est de diviser la zone d'intérêt en plusieurs sous-zones, dont chacune représentera un cluster de cellules. Nous évaluons 2 scénarios codés comme suit :

SC-LC-SA : SONCO-R activé, toutes les cellules das un seul cluster,

SC7-LC-SA : SONCO-R activé, 7 cellules par cluster.

Dans les Figures 14a, 14c et 14b nous traçons le maximum (sur toutes les cellules) des moyennes-temporelles des charges (de la cellule), des numéros des HOs trop retardés et des numéros de HOs ping-pongs pour les deux scénarios. Nous pouvons voir qu'une priorité plus grande pour le MLB est bénéfique pour réduire la charge maximale. Par contre, une priorité plus grande pour le MRO est plus bénéfique pour réduire le nombre de HOs trop retardés et de HOs ping-pongs.

Dans la Figure 14d nous pouvons voir que le design centralisé peut atteindre des valeurs plus bas pour le regret.

Regret distribuable

Dans cette section nous évaluons les avantages de pouvoir écrire le regret comme une somme de sous-regrets dans deux scénarios. Dans le premier scénario nous considérons un sous-regret par

:/3.PHD/PhD Manuscris/graphs/SONCO$_R$/$IWSON$/$L0$".eps

(a) La charge moyenne maximale

:/3.PHD/PhD Manuscris/graphs/SONCO$_R$/$IWSON$/$TL0$".eps

(b) Le numéro moyen de HOs trop tardé maximal

:/3.PHD/PhD Manuscris/graphs/SONCO$_R$/$IWSON$/$PP0$".eps

(c) Le numéro moyen de HO ping-pongs maximal

:/3.PHD/PhD Manuscris/graphs/SONCO$_R$/$IWSON$/res_reward".eps

(d) Le regret moyen (pendant les derniers $24h$)

FIGURE 14: Résultats : regret scalaire, non-distribuable (conflits sur les paramètres)

:/3.PHD/PhD Manuscris/graphs/SONCO$_R$/$SIGCOMM2014$/$L0$".eps

(a) La charge moyenne

:/3.PHD/PhD Manuscris/graphs/SONCO$_R$/$SIGCOMM2014$/$TL0$".eps

(b) Le numéro moyen de HOs trop tardé

:/3.PHD/PhD Manuscris/graphs/SONCO$_R$/$SIGCOMM2014$/$PP0$".eps

(c) Le numéro moyen de HO ping-pongs

FIGURE 15: Résultats : regret scalaire, non-distribuable, SA (conflits sur les paramètres)

cellule. Chaque sous-regret R_i dépend uniquement des demandes de mise à jour de la cellule n ($U^{\mathcal{Z}}_{n,\mathcal{K}}$). Cette solution n'est pas passable à l'échelle. Pour résoudre ce problème nous considérons que W_n (qui correspondent au regret n) ne dépend que des paramètres de la cellule n et ses voisines ($\forall n \in \mathcal{N}$). Pour implémenter ça nous utilisons l'*agrégation des états*. Nous évaluons 1 scénario codé comme suit :

SC-LD-SA : SONCO-R activé, toutes les cellules das un seul cluster, nous utilisons l'*agrégation des états*.

Dans le deuxième scénario nous considérons un sous-regret par demande de mise à jour, pour seulement les demandes ciblant le CIO, en dépendant uniquement de cette demande ($U^z_{n,k}$). Cette solution, aussi, n'est pas passable à l'échelle. Pour résoudre ce problème nous considérons que W_i (qui correspond à la demande $U^z_{n,k}$) ne dépend que des paramètres de la cellule n et ses voisines ($\forall n \in \mathcal{N}$). Pour implémenter cela nous utilisons des *descripteurs linéaires*. Nous évaluons 1 scénario codé comme suit :

SC-LD-LF : SONCO-R activé, toutes les cellules das un seul cluster, nous utilisons des *descripteurs linéaires*

Pour le scénario SC-LD-SA, nous traçons dans les Figures 15a, 15b et 15c le maximum (sur toutes les cellules) des moyennes-temporelles des charges, des numéros des HOs trop tardés et des numéros de HOs ping-pongs. Pour le scénario SC-LD-LF, nous traçons les mêmes graphiques dans les Figures 15a, 15b et 15c, respectivement.

Nous pouvons voir que donner une priorité plus grande pour le MLB est bénéfique pour réduire la charge maximale. Par contre, donner une priorité plus grande pour le MRO est plus bénéfique pour réduire le nombre de HOs trop retardés et de HOs ping-pongs.

Les regrets vecteurs

Nous considérons un réseau hétérogène. Les cellules Pico sont destinés à augmenter la capacité. Dans cette section nous définissons le cluster comme un groupe de cellules composé d'une cellule Macro et des cellules Pico sous sa couverture. Nous employons 3 fonctions SON : CRE (qui

:/3.PHD/PhD Manuscris/graphs/SONCO$_R$/$GLOBECOM2014/L0$".*eps*

(a) La charge moyenne

:/3.PHD/PhD Manuscris/graphs/SONCO$_R$/$GLOBECOM2014/TL0$".*eps*

(b) Le numéro moyen de HOs trop tardé

:/3.PHD/PhD Manuscris/graphs/SONCO$_R$/$GLOBECOM2014/PP0$".*eps*

(c) Le numéro moyen de HO ping-pongs

FIGURE 16: Résultats : regret scalaire, non-distribuable, LF (conflits sur les paramètres)

FIGURE 18: La moyenne-temporelle des KPIs

règle le CIO), MRO (qui règle le CIO et le HYS) et eICIC (qui règle le nombre d'ABS). Nous supposons que le HYS n'a pas un impact signifiant sur le CRE et le eICIC, donc nous nous focalisons sur les conflits causés par le CIO et le nombre d'ABS.

Nous évaluons 2 scénarios codés comme suit :

SC-LD-SA : SONCO-R activé, regret uni-dimensionnel $D = 1$.

SC-LD-LF : SONCO-R activé, regret multi-dimensionnel $D = 2(2N - 1) > 1$. Plus précisément deux composants de regret pour chaque demande de mise à jour visant le CIO et le ABS.

Les résultats des 2 scénarios sont mathématiquement différents, mais suivent le même cible : fournir un sens d'équité entre les fonctions SON.

Nous utilisons trois clusters de cellules (voir Figure 17)dont chacun contient une cellule Macro et deux cellules Pico. Nous rappelons que chaque cluster est gouverné par une instance SONCO.

Le réglage de l'équité La Figure 18 montre les KPIs, pour différents poids w, visés par les fonctions SON : la charge moyenne, le nombre moyen des HOs ping-pongs , le nombre des HOs trop tardés moyen, et le rapport moyen entre le débit des utilisateurs Macro et le débit des utilisateurs Pico trouvé au bord de la cellule. Nous pouvons voir que si nous donnons la priorité au CRE la charge baissera. par contre, si nous donnons la priorité au MRO c'est le nombre de HOs trop tardé qui baisse.

FIGURE 17: La topologie de réseau

Les capacités de suivi Nous voulons montrer comment un SONCO-R, qui a déjà convergé, réagit (en termes de regret) aux variations du trafic :

- Le temps nécessaire aux algorithmes pour converger à nouveau
- L'augmentation de regret pendant la période de nouvelle convergence.

Pour chaque design, SC-LD-SA et SC-LD-LF, nous considérons deux cas. Dans le premier cas le trafic augmente de 0.9λ à λ et dans le deuxième de 0.95λ à λ. Dans la Figure 19 nous traçons la moyenne-temporelle du sous-regret instantané maximal. Nous utilisons une fenêtre temporelle glissante de 2,5 jours. Nous pouvons voir que pour le design SC-LD-LF le temps nécessaire aux algorithmes pour converger à nouveau et la performance (en terme de regret) pendant cette période, sont beaucoup mieux que ceux du design SC-LD-SA.

FIGURE 19: Les capacités de suivi

FIGURE 20: Zones CCH

5. Paramètres de mobilité

Tout au long de notre travail, nous avons utilisé des paramètres de HO établis *par-cellule*. Dans cette section nous motivons notre choix. Dans la littérature il y a généralement deux approches :

- paramètres de HO *par-paire-de-cellules* : chaque cellule a des paramètres de HO indépendantes pour chaque voisine [43, 44, 45, 46, 47, 48].
- paramètres de HO *par-cellule* : chaque cellule a les mêmes paramètres de HO pour toutes les voisines [49, 50, 28, 33, 51, 34]. Ceci peut être vu comme un cas particulier de la première approche.

L'utilisation des paramètres de HO *par-paire-de-cellules* peut sembler une stratégie prometteuse permettant des configurations individuelles pour chaque voisine, par rapport à l'utilisation des paramètres de HO *par-cellule* qui limite la maniabilité de la mobilité des UEs. En fait, nous montrons que l'utilisation de paramètres de HO *par-paire-de-cellules* crée le risque d'avoir des zones où nous sommes exposés à des HOs continus enchaînés (Continuous Chained HO, CCH). Un HO enchaîné (Chained HO, CH) est l'événement où un HO réussi est effectué peu de temps après un autre HO réussi. Nous mentionnons que l'événement A3, qui aide au déclenchement du HO, prévoit des paramètres de HO *par-cellule* [5].

La description du problème

Pour un ensemble arbitraire de cellules $\{A, B, ...\}$ et tout UE à l'emplacement $\ell \in \mathcal{D}$ nous définissons la fonction de HO comme suit :

$$f_i(\ell) = \begin{cases} i & \text{si } \mathbb{T}_{i,j}(\ell) \leq 0, \forall j \in \{A, B, ...\} \setminus \{i\} \\ \text{argmax}_{j \in \{A,B,....\} \setminus \{i\}} \mathbb{T}_{i,j}(\ell) & \text{autrement} \end{cases} \tag{18}$$

avec

$$\mathbb{T}_{ij}(\ell) = RxPow_j(\ell) - RxPow_i(\ell) + CIO_{ji} - CIO_{ij} - HYS_{ij} \tag{19}$$

où $RxPow_i(\ell)$ est la puissance moyenne reçue dans l'emplacement ℓ, CIO_{ij} est l'offset de HO de la cellule i vers la cellule j et HYS_{ij} est le hystérésis de HO de la cellule i vers la cellule j. Le fait que $f_i(\ell) = k$ signifie que les UEs dans l'emplacement ℓ attaches à la cellule i doivent effectuer un HO vers la cellule k. Évidemment, si $i = f_i(\ell)$ nous n'aurons pas de HO.

Nous montrons que l'utilisation de paramètres de HO *par-paire-de-cellules*, bien qu'elle puisse avoir les avantages mentionnés, créera très probablement des zones CCH. Nous définissons la zone CCH comme la zone où f est cyclique.

Proposition 1 (HO tests)**.** *Pour une instance temporelle, pour toute $i, j \in \{A, B, ...\}$ où $i \neq j$, ayant $\mathbb{T}_{i,j}(\ell) > 0$ implique $\mathbb{T}_{j,i}(\ell) \leq 0$.*

La proposition 1 garantit qu'aucun UE (statique ou avec une vitesse basse) n'est susceptible d'effectuer en continu des CHs entre deux cellules i et j (dans un environnent statique). Par contre, il y a toujours le risque d'avoir de CHs entre plusieurs cellules. Pour mieux comprendre le problème, nous représentons das la Figure 20 un exemple. Nous utilisons 3 cellules A, B et C. La zone noire représente une zone CCH, dans laquelle toute UE effectuera des HOs sans s'arrêter dans le sens $A \to C \to B \to A$.

Remarque 2 (per-cell parameters)**.** *L'utilisation de paramètres de HO* par-cellule *est un cas particulier de l'utilisation des paramètres* par-paire-de-cellules*, c'est-à-dire que pour un ensemble de cellules $\{A, B, ...\}$ nous avons $CIO_{ij} = CIO_i$ et $HYS_{ij} = HYS_i$ $\forall i, j \in \{A, B, ...\}$, $i \neq j$, où CIO_i et HYS_i sont les paramètres de HO, respectivement l'offset et le hystérésis, définit pour la cellule i.*

Préliminaires

Nous ciblons de rechercher les conditions pour ne pas avoir des zones CCH et de quantifier leur taille lorsque ces zones existent. Pour cela, nous considérons 3 cellules A, B et C (comme dans la section précédente), non-colinéaires, omnidirectionnelles (voir Figure 20).

Nous nous concentrons sur le cas général avec les paramètres de HO *par-paire-de-cellules*. Nous définissons $\gamma = (\mathbf{CIO}, \mathbf{HYS})$ une configuration arbitraire des paramètres de HO, où :

$$\mathbf{CIO} = (CIO_{ij})_{i,j\in\{A,B,C\},i\neq j}\,,\ \mathbf{HYS} = (HYS_{ij})_{i,j\in\{A,B,C\},i\neq j}\,. \tag{20}$$

Nous rappelons que les éléments de la matrice $\mathbf{H}$ sont non négatifs. Nous définissons Γ comme l'ensemble de valeurs possibles de γ. Nous introduisons la *marge de HO* : $HOM_{ij} = CIO_{ij} - CIO_{ji} + HYS_{ij}$ pour toute $i, j \in \{A, B, C\}$, $i \neq j$,.

Nous considérons $\mathcal{A} = (x_A, y_A)$, $\mathcal{B} = (x_B, y_B)$ et $\mathcal{C} = (x_C, y_C)$ comme les coordonnées des 3 cellules A,B et C, respectivement, où $\mathcal{A}, \mathcal{B}, \mathcal{C} \in \mathcal{D}$. Soit $d_{(\ell_1,\ell_2)}$ la distance entre deux points ℓ_1 et ℓ_2, où $\ell_1, \ell_2 \in \mathcal{D}$. Nous notons $c = d_{(\mathcal{A},\mathcal{B})}$, $a = d_{(\mathcal{B},\mathcal{C})}$ et $b = d_{(\mathcal{C},\mathcal{A})}$. Nous explicitons $RxPow_A^{\kappa}(t)$ en utilisant un modèle typique d'affaiblissement du signal :

$$RxPow_A(\ell) = TxPow + k - \alpha 10\log_{10} d_{(\mathcal{A},\ell)} \tag{21}$$

où k est le coefficient de gain de propagation [dB], $\alpha > 0$ est l'exposant de gain de propagation et $TxPow$ est la puissance de transmission des cellules en[dB]. De la même manière, nous pouvons également expliciter $RxPow_B^{\kappa}(t)$ et $RxPow_C^{\kappa}(t)$.

L'équation de la frontière de HO, par exemple pour le HO de la cellule A à la cellule B, peut être écrite comme suit :

$$\mathbb{B}_{AB} = \left\{\ell \in \mathcal{D} : \log_{10} d_{(\mathcal{A},\ell)} - HOM_{AB}/\alpha = \log_{10} d_{(\mathcal{B},\ell)}\right\} = \left\{\ell \in \mathcal{D} : d_{(\mathcal{A},\ell)} \cdot \rho_{AB} = d_{(\mathcal{B},\ell)}\right\} \tag{22}$$

où $\rho_{AB} = 10^{-HOM_{AB}/\alpha}$. De même manière, $\forall i, j \in \{A, B, C\}$, $i \neq j$, $\rho_{ij} = 10^{-HOM_{AB}/\alpha}$. Finalement nous récupérons l'équation d'une ligne ou d'un cercle d'Apollonius [52, 53].

Lemme 1. *Nous considérons le cercle d'Apollonius* $\mathbb{B}(\ell_1, \ell_2, z) = \left\{\ell \in \mathcal{D} : d_{(\ell_1,\ell)} \cdot z = d_{(\ell_2,\ell)}\right\}$ *correspondant aux points* $\ell_1, \ell_2 \in \mathcal{D}$*et pour le rapport* z*. Nous pouvons expliciter les frontières de HO comme suit :*

$$\mathbb{B}_{AB} = \mathbb{B}(\mathcal{A}, \mathcal{B}, \rho_{AB}) = \mathbb{B}(\mathcal{B}, \mathcal{A}, 1/\rho_{AB}) \tag{23}$$

et de même manière toutes les autres frontières. Pour $z = 1$ *le cercle est en fait une ligne.*

Pour les théorèmes dans la section principale qui permettront d'identifier les configurations de réseau qui créent des zones CCH, nous fournissons d'abord deux lemmes qui aident à les prouver. Ainsi, nous considérons 3 variables aléatoires $z_{AB}, z_{BC}, z_{CA} \in (0, \infty)$ et nous analysons l'intersection des suivantes : $\mathbb{B}(\mathcal{A}, \mathcal{B}, z_{AB})$, $\mathbb{B}(\mathcal{B}, \mathcal{C}, z_{BC})$ et $\mathbb{B}(\mathcal{C}, \mathcal{A}, z_{CA})$, c'est-à-dire toutes les frontières de HO pour les 3 cellules (indépendamment de la direction). D'abord, nous identifions les conditions d'intersection par paire.

Lemme 2. *Nous considérons 3 cellules A,B et C. Pour tout* $z_{AB}, z_{BC}, z_{CA} \in (0, \infty)$ *nous avons :*

$$\begin{aligned}
&\mathbb{B}(\mathcal{A}, \mathcal{B}, z_{AB}) \cap \mathbb{B}(\mathcal{B}, \mathcal{C}, z_{BC}) \neq \emptyset \Leftrightarrow (a \cdot z_{CA} + b/z_{BC})^2 \geq c^2 \geq (a \cdot z_{CA} - b/z_{BC})^2\,,\\
&\mathbb{B}(\mathcal{B}, \mathcal{C}, z_{BC}) \cap \mathbb{B}(\mathcal{C}, \mathcal{A}, z_{CA}) \neq \emptyset \Leftrightarrow (c \cdot z_{BC} + a/z_{AB})^2 \geq b^2 \geq (c \cdot z_{BC} - a/z_{AB})^2\,,\\
&\mathbb{B}(\mathcal{C}, \mathcal{A}, z_{CA}) \cap \mathbb{B}(\mathcal{A}, \mathcal{B}, z_{AB}) \neq \emptyset \Leftrightarrow (b \cdot z_{AB} + c/z_{CA})^2 \geq a^2 \geq (b \cdot z_{AB} - c/z_{CA})^2\,.
\end{aligned} \tag{24}$$

Démonstration. Voir Annexe 5.1. □

Ensuite, nous identifions les conditions dans lesquelles toutes les frontières ont un point commun.

Lemme 3. *Nous considérons 3 cellules A,B et C. Pour tout* $z_{AB}, z_{BC}, z_{CA} \in (0, \infty)$ *les suivantes sont équivalentes :*

1. $\mathbb{B}(\mathcal{A},\mathcal{B},z_{AB}) \cap \mathbb{B}(\mathcal{B},\mathcal{C},z_{BC}) \cap \mathbb{B}(\mathcal{C},\mathcal{A},z_{CA}) \neq \emptyset$
2. $z_{AB} \cdot z_{BC} \cdot z_{CA} = 1$ *et* $(b \cdot z_{AB} + c/z_{CA})^2 \geq a^2 \geq (b \cdot z_{AB} - c/z_{CA})^2$
3. $z_{AB} \cdot z_{BC} \cdot z_{CA} = 1$ *et* $(c \cdot z_{BC} + a/z_{AB})^2 \geq b^2 \geq (c \cdot z_{BC} - a/z_{AB})^2$
4. $z_{AB} \cdot z_{BC} \cdot z_{CA} = 1$ *et* $(a \cdot z_{CA} + b/z_{BC})^2 \geq c^2 \geq (a \cdot z_{CA} - b/z_{BC})^2$

Démonstration. Voir Annexe 5.2. □

Résultats principaux

Dans cette section, nous évaluons le domaine de configurations qui nous assure de ne pas avoir de zones CCH. Nous définissons Γ_{ABCA} l'ensemble de configurations γ pour lesquelles nous avons des zones CCH dans le sens $A \to B \to C \to A$:

$$\Gamma_{ABCA} = \{\gamma : \exists \ell \in \mathcal{D} \text{ s.t. } f_\ell(A) = B, f_\ell(B) = C, f_\ell(C) = A\}; \tag{25}$$

de la même manière nous définissons Γ_{ACBA} pour l'autre sens :

$$\Gamma_{ACBA} = \{\gamma : \exists \ell \in \mathcal{D} \text{ s.t. } f_\ell(A) = C, f_\ell(C) = B, f_\ell(B) = A\}. \tag{26}$$

Les théorèmes suivants identifient les configurations de réseau pour lesquelles il existe des zones CCH. Ce genre de zones peut créer des problèmes majeurs. Les UEs qui préforment des HO continus entre les cellules de réseau consomment une quantité importante de ressources qui se traduit par une mauvaise qualité de service pour tous les UEs.

Théorème 2. *Nous considérons 3 cellules A,B et C, une configuration de réseau $\gamma \in \Gamma_{ABCA}$ si et seulement si les conditions suivantes sont simultanément satisfaites :*

$$\begin{cases} \rho_{AB} \cdot \rho_{BC} \cdot \rho_{CA} > 1 \\ c/\rho_{CA} < a + b \cdot \rho_{AB} \\ b/\rho_{BC} < c + a \cdot \rho_{CA} \\ a/\rho_{AB} < b + c \cdot \rho_{BC} \end{cases} \tag{27}$$

A remarquer que $\rho_{AB} \cdot \rho_{BC} \cdot \rho_{CA} > 1$ est équivalent à $HOM_{AB} + HOM_{BC} + HOM_{CA} = -\alpha log_{10}\rho_{AB} \cdot \rho_{BC} \cdot \rho_{CA} < 0$.

Démonstration. Voire Annexe 5.3. □

Le Théorème 2 peut être simplement adapté pour $\gamma \in \Gamma_{ACBA}$ en commutant B et C. Le théorème suivant se concentre sur les paramètres de HO *par-cellule*, qui est un cas particulier de paramètres de HO établi *par-paire-de-cellules*.

Théorème 3. *Dans le cas des paramètres de HO* par-cellule *il n'existe pas de zones CCH.*

Démonstration. Voir Annexe5.2. □

Les résultats numériques.

FIGURE 21: Les configurations sans zones CCH pour $a = b = c$ et $\mathbf{H}[dB] = \mathbf{0}$

Les configurations problématiques de réseau Dans la Figure 21 nous représentons le domaine de (HOM_{AB}, HOM_{BC}, HOM_{CA}) pour lequel nous n'avons pas de zones CCH. Nous considérons : i) $\mathcal{A}$, $\mathcal{B}$ et $\mathcal{C}$ sont les sommets d'un triangle équilatéral ($a = b = c$) et ii) pour tout $i, j \in \{A, B, C\}$, $i \neq j$ nous mettons $H_{ij}[dB] = 0$ (implicitement $HOM_{ij} = -HOM_{ji}$). Nous représentons uniquement les points à l'intérieur de la sphère $HOM_{AB}^2 + HOM_{BC}^2 + HOM_{CA}^2 = 50^2$.

FIGURE 22: La taille de la zone *CCH,* en [%] par rapport à $S_{\bigcirc \mathcal{ABC}}$

L'enveloppe de la sphère reflète la zone d'intérêt. La partie pleine de la sphère représente les configurations sans zones CCH. Pour faciliter la compréhension de la figure, nous identifions pour le lecteur que la partie pleine de la sphère contient 3 *volumes* symétriques et un *plan* ($HOM_{AB} + HOM_{BC} + HOM_{CA} = 0$) qui coupe la sphère. Nous pouvons voir que la sphère est en fait assez vide. Nous pouvons donc tirer la conclusion qu'il y a beaucoup de configurations avec des zones CCH. Selon le Théorème 3 ceci ne peut pas arriver si nous utilisons des paramètres de HO *par-cellule*. Nous pouvons également identifier ces configurations dans la Figure 21 comme le plan mentionné précédemment, qui coupe la sphère.

La quantification de la zone CCH Afin d'avoir une idée de la taille de la zone CCH, nous fournissons ensuite des résultats numériques. Il est trop difficile de présenter des résultats pour toute configuration de réseau, c'est-à-dire $\forall \gamma \in \Gamma$ et $\forall \mathcal{A}, \mathcal{B}, \mathcal{C} \in \mathcal{D}$. Nous choisissons un cas particulier pour faciliter la lecture. Ainsi, nous considérons le cas général en utilisant des paramètres de HO par-paire-de-cellules, où : i) $\mathcal{A}, \mathcal{B}$ et $\mathcal{C}$ sont les sommets d'un triangle équilatéral ii) la distance e inter-site est 500 m et ii) $\alpha = 4$.

Dans cet exemple, nous analysons seulement la taille de la zone CCH dans le sens $A \rightarrow C \rightarrow B \rightarrow A$ que nous notons par $S_{\bullet}$. Nous définissons $S_{\bigcirc \mathcal{ABC}}$ la taille de la surface du cercle circonscrit du triangle ΔABC . Nous remarquons que $S_{\bigcirc \mathcal{ABC}}$ est comparable à la taille d'une cellule.

En regardant le Théorème 2 nous avons eu l'intuition que pour Γ_{ACBA} la valeur de $\rho_{BA} \cdot \rho_{AC} \cdot \rho_{CB}$ pouvait être très importante. Implicitement la valeur de $\Sigma = HOM_{BA} + HOM_{AC} + HOM_{CB}$ semble être très importante. Donc, ce que nous nous attendons est que déjà la valeur de Σ fournisse une quantité importante d'information sur la taille de la zone CCH, sans connaître individuellement HOM_{BA}, HOM_{AC} et HOM_{CB}. Dans la Figure 22 nous donnons des valeurs du rapport $S_{\bullet}/S_{\bigcirc \mathcal{ABC}}$ en fonction de HOM_{BA} et HOM_{CB} pour différentes valeurs de Σ. Implicitement $HOM_{AC} = \Sigma - HOM_{BA} - HOM_{CB}$. Nous pouvons voir que pour $\Sigma = 0$ ($\rho_{BA}\rho_{AC}\rho_{CB} = 1$) la taille de la zone est nulle et, si Σ s'éloigne de 0, la taille de la zone CCH augmente en touchant 25% de $S_{\bigcirc \mathcal{ABC}}$.

L'existence de ce problème a été vérifiée avec un simulateur réseau.

6. Conclusions

Les fonctions de réseau auto-organisé (SON) sont actuellement l'un des principaux défis pour les réseaux mobiles, Elles sont destinées à automatiser les réglages du réseau. Ainsi, les opérateurs de réseaux peuvent remplacer l'intervention humaine coûteuse en ce qui concerne les tâches répétitives. Cela peut considérablement améliorer la vitesse de réponse et la Qualité de Service (QoS) offerte à leurs clients.

Pour pouvoir acheter les fonctions SON des différents fournisseurs, lorsqu'ils sont mis ensemble, nous devons nous assurer d'avoir toujours une bonne performance du réseau. Pour cela, nous avons analysé les conflits potentiels qui peuvent être rencontrés lorsque nous utilisons plusieurs fonctions SON conçues indépendamment. De plus, les fournisseurs SON offrent des informations très limitées sur les algorithmes utilisés. Ainsi, dans notre approche, nous avons considéré les fonctions SON comme des *boîtes noires*, Pour faire face aux conflits SON, 3GPP a proposé l'utilisation d'un COordonnateur SON (SONCO). Le but de cette thèse est de proposer des solutions pour le SONCO.

Tout d'abord, nous montrons comment nous pouvons identifier les conflits SON potentiels. Puis, nous fournissons une méthodologie pour diagnostiquer lesquels des conflits potentiels sont actifs (SONCO-D), c'est-à-dire ils ont un impact négatif sur les KPIs . Pour cela, nous avons utilisé le Classificateur Naïf Bayésien (NBC). Les résultats montrent la capacité d'identifier la cause des conflits, notamment la fonction SON qui est la source du problème. En allant

plus loin dans les détails, nous essayons d'identifier également quel réglage de la fonction SON est responsable des mauvaises performances. Évidemment, dés que nous rentrons plus dans les détails, la précision de diagnostic baisse. Surtout, le NBC a prouvé qu'il est un bon candidat pour le diagnostic des conflits SON.

Après avoir identifié les conflits actifs dans le réseau, la prochaine étape est leur résolution. Une façon de le faire est de rendre les conflits inactifs en reconfigurant les fonctions SON. Une autre solution est d'appliquer un mécanisme de résolution des conflits (SONCO-R). Nous considérons que le SONCO-R reçoit toutes les demandes de mise à jour créées par les fonctions SON et il doit décider lesquelles d'entre elles seront acceptées et lesquelles seront rejetées, en assurant l'équité entre les fonctions SON. Pour cela nous utilisons l'apprentissage par renforcement (RL) car il permet de profiter de renseignements des décisions passées pour améliorer l'avenir. Nous avons analysé plusieurs designs de RL et nous avons montré comment rendre l'algorithme passable à l'échelle (en utilisant l'*agrégation des états* ou des *descripteurs linéaires*). Les résultats montrent que l'*agrégation d'états* est un bon candidat pour réduire l'espace d'états. L'utilisation des *descripteurs linéaires* réduit encore plus l'espace d'états, mais nous devons faire attention parce que l'approximation pourrait être trop imprécise.

Dans toutes nos études, nous avons utilisé des paramètres de HO *par-cellule*. Nous motivons notre choix en montrant que si nous utilisons des paramètres de HO *par-paire-de-cellules* nous risquons de créer des zones où les UE seront en HO continu.

L'une des prochaines étapes devrait être d'améliorer le concept de fonction SON. Les fonctions SON doive être plus plus facile à coordonner. Par exemple, la fonction SON doit mieux comprendre l'environnement et reconnaître les autres fonctions SON exécutées en parallèle. Il serait utile si elle peut prévoir comment elle est impacté par les autres fonctions SON et même déclencher des alarmes si elle estime que sa performance est détériorée à cause de cela.

Les algorithmes du SONCO doive évoluer en conséquence. Il doit être prêt à coordonner des fonctions SON plus intelligentes.

Une lacune du travail actuel est le manque des données réelles. Les fonctions SON ne sont pas encore largement déployées. Une fois que cela se produira, il sera intéressant de tester les algorithmes avec des données réelles.

Contents

Abstract

The demand for mobile data is continuously increasing. Consequently the International Telecommunications Union (ITU) has released the International Mobile Telecommunications-Advanced (IMT-A) requirements for the 4^{th} generation of mobile telecommunications technology (4G) . The Long Term Evolution (LTE), specifically its evolution LTE-Advanced (LTE-A), is the most significant candidate for a 4G system. LTE is standardized in Release 8 of the 3^{rd} Generation Partnership Project (3GPP) and LTE-A is considered to be Release 10 of 3GPP. Continuously adding new technologies in the network leads to an increased management complexity. This translates into increased CAPital EXpenditures (CAPEX) and OPerational EXpenditures (OPEX) especially in the case of Heterogeneous Networks (HetNets).

Envisaging these cost augmentations 3GPP has introduced, already from Release 8, the Self Organizing Network (SON) functions. There are mainly three categories of SON functions: self-configuration, self-optimization and self-healing [2]. In this thesis we focus on the self-optimization functions. In the sequel SON refers particularly to this category.

A network operator might choose to purchase several functions depending on its targets in terms of network performance. Having several SON functions in the network might lead to conflicts, especially in a multi-vendor environment.

For all these SON functions to work properly together we have to ensure that they do not conflict, or if a conflict exists we have to detect it and enforce a resolution mechanism. To deal with potential SON conflicts Release 10 of 3GPP has introduced the SON COordinator (SONCO) function [3], [4]. The 3GPP description of the SONCO is conceptual. No detailed specifications on the SONCO implementation is given. According to [4], the tasks of a SONCO include: conflict avoidance, conflict detection, conflict diagnosis and conflict resolution.

In this thesis we tackle the problem of the coordinating SON functions. Specifically we focus on the SON conflict detection and diagnosis (SONCO-D), and on the SON conflict resolution (SONCO-R). For this purpose we had to develop an LTE-A network simulator, modelling in detail several User Equipment (UE) procedures: *Connection Establishment*, *Connection ReEstablishment* and *Connection ReConfiguration (HandOver, HO)*. The radio channel modelling is based on the 3GPP TS 363.814 [54] and the ITU-R M2134 [55]. Thus, in the thesis manuscript, we provide a brief description of the simulator. Next we follow with an introduction to the SON functions and the formalization of the SON conflict problem. As a solution to this we introduce the SONCO.

For the SONCO-D we propose a framework based on Naive Bayesian Classifier (NBC). The target is to identify which of the SON function is the root of network faults, i.e. bad network Key Performance Indicators (KPIs). We evaluate the performance of the proposed SONCO-D on a scenario with 3 SON functions: MLB via Cell Range Expansion (CRE), MRO and eICIC. NBC proves to be a good candidate.

For the SONCO-R we propose a framework based on Reinforcement Learning (RL). To make the RL solution scalable we make use of two flavours of linear function approximation: state aggregation and linear features. We then analyse several scenarios where we apply the framework. First we focus on the measurement conflicts, i.e when the parameter tuned by one SON instance affects the input measurements of a different SON instance affecting the network stability. Then we focus on the parameter conflicts, i.e. when two SON functions have different requests on how to change the same network parameter. We analyse the benefits of centralized vs. distributed algorithm implementations w.r.t. the achieved network performance, their convergence time and

the computational complexity. In our application examples we make use of several SON functions like: MLB, MRO, MLB via CRE and eICIC.

Throughout our work we have considered the HO parameters to be established *per-cell*, i.e. the HO parameters are the same for all neighbouring cells. We motivate our choice by presenting the risks coming with the alternative, which is using *per-cell-pair* HO parameters, i.e. it allows for different HO parameters w.r.t. each of the neighbouring cells.

List of Figures

List of Tables

Chapter 1

Introduction

In this thesis we consider a mobile network running in parallel several automated optimization algorithms, so called Self-Organizing Network (SON) algorithms. Each algorithm is intended to optimize some network performance indicators by tuning a set of network parameters. We consider the algorithms to be independent, and thus there is a risk that at some point they may conflict. The thesis focuses on the management system of the optimization algorithms. It aims to deal with the potential conflicts between the optimization algorithms. First we tackle the diagnosis problem which identifies the conflict. Then we provide a resolution mechanism which decides when to favour one algorithm or the other, by deciding which of the parameter updates (suggested by the optimization algorithms) will be enforced in the network. Finally, we also analyse the trade-offs for restricting the values of some network parameters in order to avoid having bad network performances.

1.1 Context

The demand for mobile data is continuously increasing. Consequently the International Telecommunications Union (ITU) has released the International Mobile Telecommunications-Advanced (IMT-A) requirements for the 4^{th} generation of mobile telecommunications technology (4G) [1]. They set nominal data rates of 1 Gbit/s for low mobility and 100 Mbit/s for high mobility, peak link spectral efficiency of 15 bit/s/Hz in the downlink and 6.75 bit/s/Hz in the uplink, etc. In order to remain competitive the network operators have to find solutions of answering these demands. The Long Term Evolution (LTE), specifically its evolution LTE-Advanced (LTE-A), is the most significant candidate for a 4G system [2] (see Fig. 1.1). LTE is standardized in Release 8 of the 3^{rd} Generation Partnership Project (3GPP) and LTE-A is considered to be Release 10 of 3GPP. Besides LTE, mobile networks still have a lot of legacy technologies like Global System for Mobile Communications (GSM), Universal Mobile Telecommunications System (UMTS), etc. Moreover, Heterogeneous Networks (HetNets), i.e. the use of small cells besides Macro cells, is also one major trend today. Managing all these technologies translates into increased CAPital EXpenditures (CAPEX) and OPerational EXpenditures (OPEX).

Envisaging these cost augmentations, 3GPP has introduced, already from Release 8, the Self Organizing Network (SON) functions. There are mainly three categories of SON functions (Fig.

Figure 1.1: LTE and SON in 3GPP releases

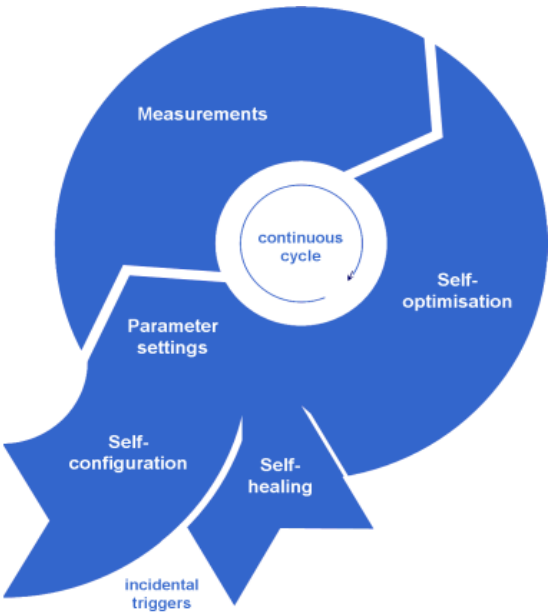


Figure 1.2: SON function classification [56]

1.2): self-configuration, self-optimization and self-healing [2]. The self-configuration functions are meant to provide plug-and-play capabilities to the network equipments. Examples of such functions include: Automatic Neighbour Relationship (ANR), Physical Cell ID (PCI) Allocation, LTE-A relay auto-connectivity, etc. The self-optimization functions tackle the run-time optimization of the network. Examples of such functions include: Mobility Load Balancing (MLB), Mobility Robustness Optimization (MRO), Coverage and Capacity Optimization (CCO), enhanced Inter-Cell Interference Coordination (eICIC), etc. The self-healing functions deal with network failure troubleshooting (e.g. failing cells). Examples of such functions include: Cell Outage Compensation (COC), cell degradation detection and diagnosis, etc.

In this thesis we focus on the self-optimization functions. In the sequel SON refers particularly to this category. A network operator might choose to purchase several SON functions depending on its targets in terms of end-user experience, and could even prefer a multi-vendor solution. In this case we have several SON functions coming from different vendors. As each vendor independently develops its SON solutions, it is likely that these SON functions will conflict.

For all these SON functions to work properly together we have to ensure that they do not conflict, or if a conflict exists we have to detect it and enforce a resolution mechanism. To deal with potential SON conflicts Release 10 of 3GPP has introduced the SON COordinator (SONCO) function [3], [4], see Fig. 1.1. The 3GPP description of the SONCO is conceptual. No detailed specifications on the SONCO implementation are given. According to [4], the tasks of a SONCO include: conflict avoidance, conflict detection, conflict diagnosis and conflict resolution .

Figure 1.3: SONCO components

The conflict avoidance function (SONCO-A, Fig. 1.3) must properly choose the settings of the SON functions such that the conflicts are reduced to a minimum and the network still achieves its target performance.

The conflict detection and diagnosis function (SONCO-D, Fig. 1.3) has to be able to identify what are the potential conflicts and which of them are active conflicts, i.e. they effectively degrade the network performance. One consequent action might be to change the settings of the SON functions in order to prevent the conflict from occurring in the future (via the conflict avoidance function), i.e. make the conflict in-active.

Another consequent action can be to employ a resolution mechanism (SONCO-R, Fig. 1.3)

which does not eliminate the conflict but instead, at the moment when the conflict occurs, it takes the decision to favour (the requests of) one SON function or another based on predefined criteria. The SONCO-D and the SONCO-R should receive relevant information from the SON functions, e.g. parameter update requests, input measurements, etc.

1.2 Motivation

There are several strategies in coordinating the SON functions. One approach is to have a co-design of the SON functions. This is possible if the SON functions are all built by one vendor (or if several vendors collaborate on such a design).

Another approach is to have independent designs of the SON functions. Thus, a network operator can purchase and orchestrate independently designed SON functions. This is useful mostly in a multi-vendor environment, but not only. Two cases can be identified for this approach. In one case the vendors share the algorithms inside the SON functions. Thus we say that the SON functions are seen as *white boxes*. The SONCO designer can thus come up with a solution encompassing all this knowledge. However, it is highly unlikely that the SON vendors would share the algorithm inside the SON functions as it represents their proprietary knowledge.

This takes us to the second case where the vendors do not share the algorithm inside the SON function. Thus, in this case the SON functions are seen as *black boxes*. This creates a proper environment for vendors to be able to propose SON functions without having to reveal the proprietary algorithms.

Under these considerations we target to answer the following questions:

1. Given some network failure (e.g. bad network performance) what is the cause of this? We do not consider the case where the network is under-dimensioned, i.e. there exists at least one behaviour of the SON functions which can achieve the desired performance. In the context of several independent SON functions it is quite difficult to identify the root of a network failure. If one of the SON functions cannot achieve the desired network performance, in order to identify the cause we have to take into account all the inter-dependencies between the SON function.

2. Once the conflict is identified, what resolution method can we apply such that we can judiciously decide when to accept and when to deny the parameter update requests of the conflicting SON functions? This decision is quite problematic. First, we need a criterion based on which to establish the best decision. This criterion should not only focus on the instantaneous situation, it should consider a more long term performance. Second, in finding the best decisions we have to be reasonable in terms of the demanded computational effort and the available information.

3. Are there any restrictions which can be applied on the tuned parameters in order to avoid some undesired behaviour which leads to bad network KPIs? The difficulty here is to identify the trade-offs between the restrictions and benefits.

1.3 Scope and methodology

With the above considerations we target to design a SONCO which is capable of managing independently designed SON functions, seen as *black boxes*. In this thesis we provide some answers to the questions outlined in the previous section.

1. We tackle the SON conflict detection and diagnosis (SONCO-D) making use of a Naive Bayesian Classifier framework. Concretely a SON function, in trying to achieve its target might be over-ambitious. This can affect the other SON functions in the sense that they can no longer achieve their targets. The SONCO-D learns from previous experiences what is the most probable root (SON function) of this under-performance.

2. We tackle the problem of conflict resolution (SONCO-R) making use of a Reinforcement Learning framework. We focus on two problems: measurement conflict resolution and parameter conflict resolution. The measurement conflict occurs when the tuned parameter of one SON function impacts the input of a different SON function, affecting the network stability by triggering unnecessary parameter changes. The parameter conflict resolution treats the case where two SON functions request a network parameter to change in opposite directions. The SONCO-R receives all the update requests from the SON functions and decides which are accepted and which are denied. It learns the best actions from past experience.

3. We study the benefits of applying some restrictions on the network parameters, specifically the mobility/ HandOver (HO) parameters. If we consider a network cell, the general case accounts for different HO parameter configurations w.r.t. all its neighbouring cells, i.e. *per-cell-pair* HO parameters. The restriction considers the same HO parameters w.r.t. all its neighbouring cells, i.e. *per-cell* HO parameters. Restricting the HO parameters to be established *per-cell* eliminates the risks of having areas where users experience continuous HOs, but trades off the manoeuvrability of the user mobility.

We focus on LTE(-A) networks as they are the main candidates for the 4G networks. Our work could be easily extrapolated to other technologies. The work shall serve as an input to SON/SONCO deployments (architecture, algorithms, etc.). It also contributes to the European project SEMAFOUR. The SEMAFOUR project is meant to design and develop a unified self-management system, which enables the network operators to holistically manage and operate their complex heterogeneous mobile networks. The ultimate goal is to create a management system that enables an enhanced quality of user experience, improved network performance, improved manageability and reduced operational costs [57].

1.4 Contributions and structure of the thesis

In this thesis we tackle the problem of designing a SONCO. Specifically we focus on the SON conflict detection and diagnosis, and on the SON conflict resolution. Briefly, the contribution of the thesis consists of the following: i) we build an LTE-A simulator in C/C++ which models in detail the mobility procedures, ii) we propose an enhancement of the SON update requests by including a *criticalness* indicator, iii) we propose, formalize and evaluate a SONCO-D design based on an NBC framework, iv) we propose, formalize and evaluate a SONCO-R design based on a RL framework, showing how we can exploit the specificities of our problem in order to make the RL solution scalable v) we give recommendations on restrictions for the HO parameters in order to avoid having areas where users experience continuous HOs.

In order to have a realistic and accurate platform which can sustain the employed SON functions, in the thesis we have developed an LTE-A simulator, modelling in detail several UE procedures: *Connection Establishment*, *Connection ReEstablishment, Connection ReConfiguration (HO)*, etc. The interest is to have access to details concerning events like Radio Link Failures, HO ping-pongs, etc. The radio channel modelling is based on the 3GPP TS 363.814 [54] and the ITU-R M2134 [55]. We briefly present the simulator in chapter 2. In the same chapter we introduce the SON functions and the problems which come with them, namely the SON conflicts. As a solution to this we follow with the introduction of the SONCO. As most of the work has been done within the framework of the European project SEMAFOUR [57], we provide a short description how our work fits in this project .

In chapter 3 we propose a framework for SON conflict diagnosis (SONCO-D). First we identify the symptoms, i.e. the network Key Performance Indicators (KPIs), which are used to detect a network fault. Second we identify all the potential causes which might have lead to the network fault. Then, by making use of a Naive Bayesian Classifier (NBC) we present the cause inference method. This consists of outputting the most probable cause given the symptoms. We show results for 3 orders of details of the cause dictionary. In this chapter we consider a heterogeneous

network (i.e. two cell sizes: Macro and Pico) and use three SON functions: MLB via Cell Range Expansion (CRE), MRO and eICIC.

In chapter 4 we propose a framework for SON conflict resolution (SONCO-R). We assume that all requests of the SON functions are received by the SONCO-R entity. The SONCO-R decides which requests to accept and which to deny. To do this, we propose that the SON update requests include a *criticalness* indicator reflecting how important the request is w.r.t. the target performance indicators of the SON function. Then we propose a SONCO-R design based on Reinforcement Learning (RL). RL allows to improve the future decisions based on the outcome past experience. Specifically we aim to minimize the highest *criticalness* among the requests and the RL learns which decisions allows us to do so. It is widely know that RL faces scalability problems. To make the RL solution scalable we make use of two flavours of linear function approximation: state aggregation and linear features. Briefly, we make the RL solution scalable by restricting the dependencies of the learned information. We then analyse several scenarios where we apply the framework. We show how our solution deals with measurement conflicts, improving the network stability by eliminating unnecessary parameter changes. Also, we show how we can deal with parameter conflicts, i.e. SON functions which target the same network parameters and have opposite requests on how to change them. Throughout our application examples we fist employ a homogeneous network (only Macro cells) make use of two SON functions: MLB and MRO, and then we employ a heterogeneous network making use of three SON functions: MLB via CRE, MRO and eICIC.

In chapter 5 we analyse the impact of the choice of the HO parameters. In a general case the HO parameters can be established *per-cell-pair*, i.e. each cell has one set of HO parameters for each neighbouring cell. Throughout our work (chapters 3 and 4) we have put a restriction on the HO parameters and considered them to be established *per cell*, i.e. each cell uses the same HO parameters w.r.t. all its neighbouring cells. In this chapter we motivate our choice. Using *per-cell* HO parameters limits the manoeuvrability of the user mobility. Alternatively, if we use *per-cell-pair* HO parameters, this can lead to creating areas where users would HO continuously. We call them Continuous Chained HO (CCH) areas.

In chapter 5 we outline the conclusions of this thesis and provide directions for future work.

1.5 Publications

Under peer review

[J1] O. Iacoboaiea, B. Sayrac, S. Ben Jemaa, and P. Bianchi, "On mobility parameter configurations that can lead to chained Handovers" submitted to IEEE Transactions on Vehicular Technology,

[J2] O. Iacoboaiea, B. Sayrac, S. Ben Jemaa, and P. Bianchi, "SON Coordination in Heterogeneous Networks. A Reinforcement Learning Framework" submitted to IEEE Transactions on Wireless Communications.

Published

[C1] O. Iacoboaiea, B. Sayrac, S. Ben Jemaa, and P. Bianchi, "SON Conflict Diagnosis in Heterogeneous Networks," in IEEE PIMRC 2015,

[C2] O. Iacoboaiea, B. Sayrac, S. Ben Jemaa, and P. Bianchi, "Low Complexity SON Coordination using Reinforcement Learning," in IEEE GLOBECOM 2014,

[C3] O. Iacoboaiea, B. Sayrac, S. Ben Jemaa, and P. Bianchi, "SON Conflict Resolution using Reinforcement Learning with State Aggregation," in SIGCOMM 2014, All things cellular Workshop,

[C4] O. Iacoboaiea, B. Sayrac, S. Ben Jemaa, and P. Bianchi, "Coordinating SON Instances: A Reinforcement Learning Framework," in IEEE VTCFall 2014,

[C5] O. Iacoboaiea, B. Sayrac, S. Ben Jemaa, and P. Bianchi, "Coordinating SON Instances: Reinforcement Learning with Distributed Value Function," in IEEE PIMRC 2014,

[C6] O. Iacoboaiea, B. Sayrac, S. Ben Jemaa, and P. Bianchi, "SON coordination for parameter conflict resolution: A reinforcement learning framework," in IEEE WCNC 2014, SONET Workshop,

Chapter 2

LTE-A and SON modelling

In order to have a realistic and accurate platform which can sustain the employed Self Organizing Network (SON) functions, in the thesis we have developed an LTE-A simulator in C++, with a time granularity of 1 ms. In this section, first we present the details of how we model the LTE-A network in our simulator with all the features which are relevant to the SON functions' implementation. We follow with the description of the SON functions, giving more details on the ones which are later used in our work, and with the description of the SON COordinator (SONCO). Finally, we provide a summary of how the simulator works including the LTE-A features, the SON functions, the SONCO, etc.

2.1 LTE-A introduction

The Long Term Evolution (LTE), specifically its evolution LTE-Advanced (LTE-A), is the most significant candidate for a 4G system [2]. LTE is standardized in Release 8 of the 3^{rd} Generation Partnership Project (3GPP) and LTE-A is considered to be Release 10 of 3GPP. To be able to cope with the LTE characteristics a new system architecture was also introduced in Release 8 of 3GPP, namely the Evolved Packet System (EPS). For details we refer the reader to [9]. The EPS contains two main components (see Fig. 2.1):

- the Evolved Packed Core (EPC) as a result of the System Architecture Evolution (SAE) study. Its main components are:
  - the Mobility Management Entity (MME): It provides a direct logical connection with the User Equipment (UE), it stores the UE context information, it generates the temporary identities for the UE and it deals with the UE mobility.
  - the Serving Gateway (S-GW): Its main task is to forward the data to the UE.
  - the Packet Data Network (PDN) Gateway (P-GW): It is the gateway to the packet data network.
  - the Home Subscriber Service (HSS): It manages the user profiles.
  - the Policy Charging Enforcement Function (PCEF). It enforces the policy and charging rules.
- the Evolved Universal Terrestrial Radio Access Network(E-UTRAN) as a result of the LTE study. It consists only of the evolved Node B (eNB) which interfaces to the UEs. A given eNB is connected with neighbouring eNBs via the X2 interface. Throughout this manuscript the terms eNB, cell and sector are synonymously used.

In our work we have built a network simulator modelling only the E-UTRAN/LTE(-A) component. The EPC is not modelled. In the following section we describe in more details the system model and present our simulation choices and assumptions.

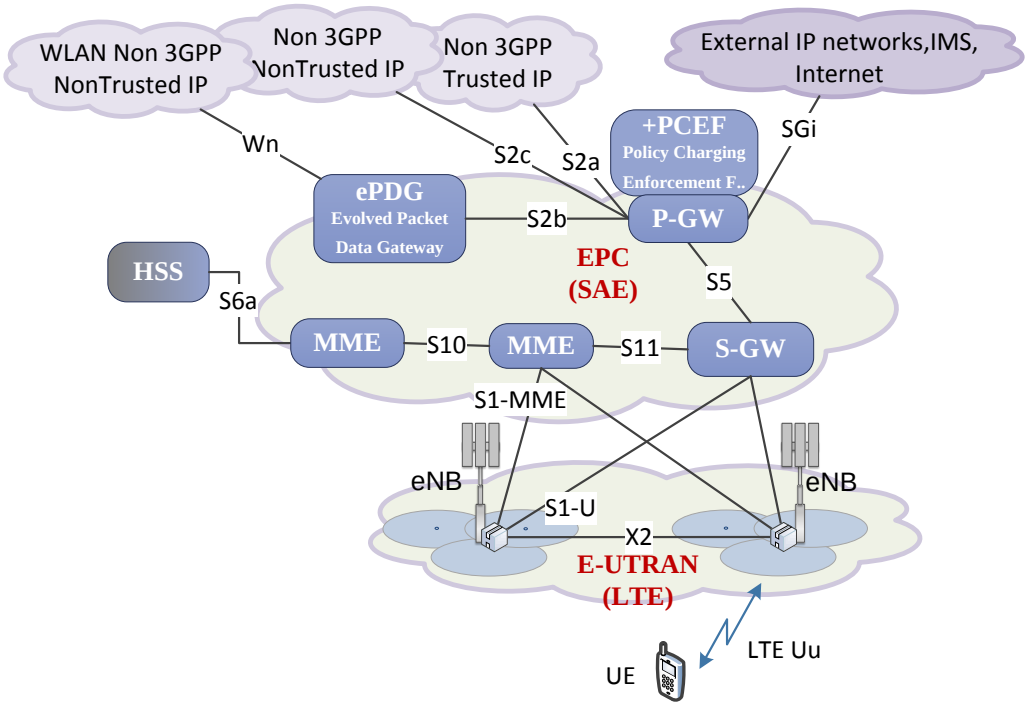


Figure 2.1: EPS architecture

2.1.1 UE mobility

In our simulator UEs pop up randomly in the network according to Space Poisson Point Processes (SPPPs) [58]. They may have different speeds and movement directions. Their location is updated every every millisecond, i.e. every LTE sub-frame. We consider an File Transfer Protocol (FTP)-like traffic, i.e. each UE which arrives in the network aims to download a file (of fixed size FS [MBits/UE]). When a UE arrives in the network and wants to download its file, it first has to attach to a cell. Let $RxPow_n^\kappa$ be the received signal power of UE κ from cell n (layer 3 filtered, see Annex 1.1 for details). Normally it should choose the one which has the best channel conditions.

At a given time instance, if UE κ wants to download data it first goes through the *Connection Establishment* procedure (Fig. 2.2) in order to attach to a network cell. Specifically, it attaches to cell $iCell_\kappa$ (the time index is omitted):

$$iCell_\kappa = \text{argmax}_n \left(RxPow_n^\kappa + CIO_n \right) \tag{2.1}$$

where CIO_n is the Cell Individual Offset (CIO) of cell n [5]. If the *Connection Establishment* is successful then the UE goes into the *Receive Data* procedure, otherwise the UE is dropped.

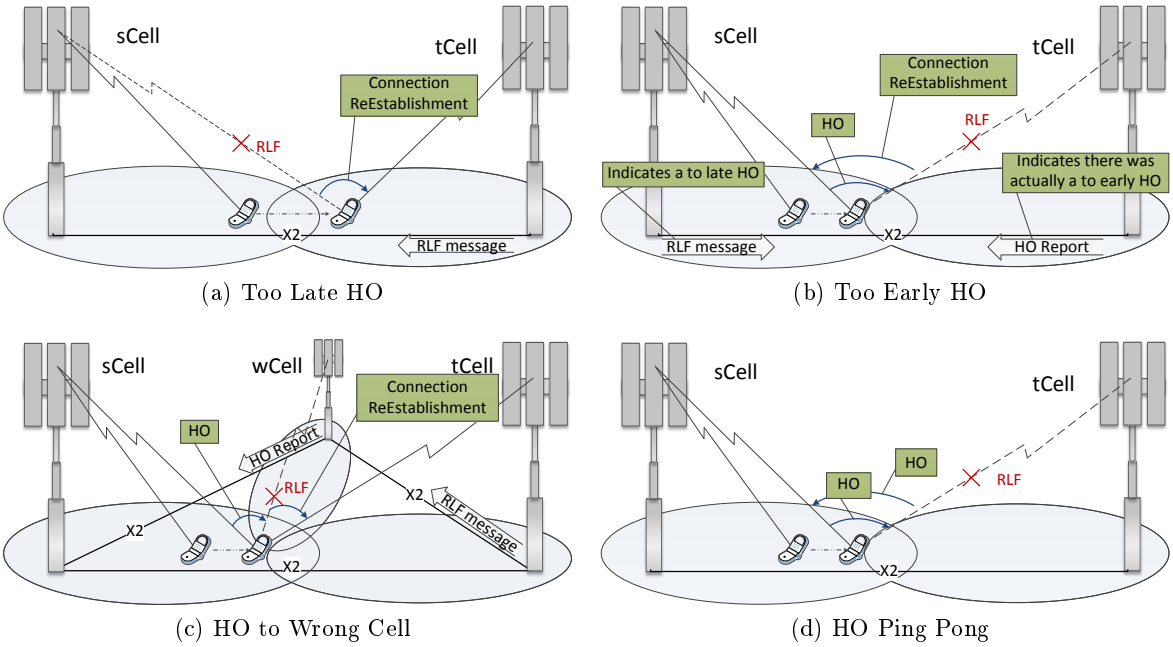
If the serving cell no longer offers good (or the best) radio channel condition then the UE searches for a new target cell which offers better (or the best) channel conditions. This procedure is called *Connection ReConfiguration* or *HandOver* (HO). To be precise a UE attached to cell $sCell$ starts a Time To Trigger (TTT) countdown for a HO to a target cell $tCell \neq sCell$ if:

$$tCell_\kappa = \text{argmax}_n \left(RxPow_n^\kappa + CIO_n + HYS_{sCell} \mathbb{I}_{\{n=sCell\}} \right) \tag{2.2}$$

where HYS_{sCell} is the HO Hysteresis (HYS) of sCell [5] .

During its data download and even during a HO procedure a UE may loose connection with its serving cell. We call this a Radio Link Failure (RLF). If such an event occurs then the UE goes into the *Connection ReEstablishment* procedure where the UE attempts to connect to a cell using similar steps as in the *Connection Establishment*. If the RLF occurs while in the *Connection ReEstablishment* procedure then the UE is dropped.

Figure 2.2: UE procedures



(a) Too Late HO

(b) Too Early HO

(c) HO to Wrong Cell

(d) HO Ping Pong

Figure 2.3: Tracked Events

The previously mentioned procedures (Fig. 2.2) basically consist of a certain message exchange sequence on different LTE protocol layers between an eNB and a UE. We do provide the details of all the procedures as modelled/ implemented in our simulator in Appendix 1.2.

We now present a set of events which are of interest for the SON functions (to be introduced in section 2.2) employed throughout our work:

- Too late HO: An RLF occurs after the UE has stayed for a long period of time in the cell; the UE attempts to re-establish the radio link connection in a different cell [6], see Fig. 2.3a.

- Too early HO: An RLF occurs shortly after a successful handover from a source cell to a target cell or a handover failure occurs during the handover procedure; the UE attempts to re-establish the radio link connection in the source cell [6], see Fig. 2.3b.

- HO to wrong cell: An RLF occurs shortly after a successful handover from a source cell to a target cell or a handover failure occurs during the handover procedure; the UE attempts to re-establish the radio link connection in a cell other than the source cell and the target cell [6], see Fig. 2.3c

- Ping pong HO: A successful HO occurs shortly after another successful HO. The target cell of the second HO is the source cell of the first HO [7], see Fig. 2.3d.

- Chained HO (CH): A successful HO occurs shortly after another successful HO. Note that this is generalization of the ping pong HO.

Note that in eq. (2.1) and eq. (2.2) we have come across two network configuration parameters: the CIO and the HYS. They are called mobility/HO parameters and are involved in many of our studies presented in the sequel. The CIO is used to shift the cell borders. The HYS is used to delay the HOs between cells, creating a hysteresis effect on the HO procedure. Note that this HO procedure is based on the 3GPP A3-event [5]. The A3-event is used to trigger the UE to send measurement reports to the serving cell in order to help prepare a HO. Basically the UE says to the serving cell that a neighbouring cell has better radio channel conditions. The CIO and HYS are used in solving several practical problems (see [6, 8]) among which we mention:

- load balancing: The UEs attached to overloaded cells are forced to HO to neighbouring cells.
- prevent HO ping-pongs: Typically this is a problem that the UEs at the cell border are facing.
- prevent link failures due to too late HOs, too early HOs and HOs to wrong cell.

We have chosen to use *per-cell* HO parameters, and not *per-cell-pair,* in order to avoid Continuous CH (CCH) areas. The *per-cell-pair* HO parameters allow the cells to have individual HO parameters for all its neighbouring cells. The *per-cell* HO parameters sets for each cell the same HO parameter for all its neighbouring cells. This is argued in Section 5 .

2.1.2 Almost Blank Sub-frames

In order to reduce the interference towards the neighbouring cells some cells might be forbidden to transmit at certain time instances. To better understand this we present in Fig. 2.4 the structure of an LTE downlink frame. Thus, a *frame* has a length of 10 ms. It is composed of 10 *sub-frames* of 1 ms each. In our simulator, the resources of each *sub-frame* are allocated to only one user. We assume that all the cells are synchronized on a frame basis. A sub-frame where no user data is transmitted is called an Almost Blank Sub-frame (ABS). The ABS helps reduce the interference towards neighbouring cells. The technique is used typically in heterogeneous networks where sometimes the range of the Pico (or other small) cells is extended and in order to protect the UEs at the cell edge which we forced to attach to the Pico cells we can employ such ABSs on the Macro cells. For details see Appendix 1.3.

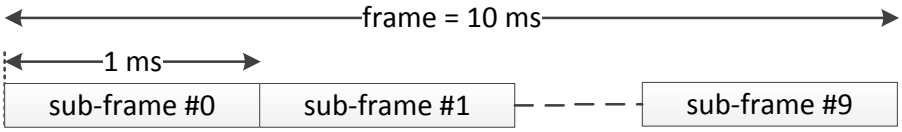


Figure 2.4: LTE frame

2.2 Introduction to Self Organizing Networks

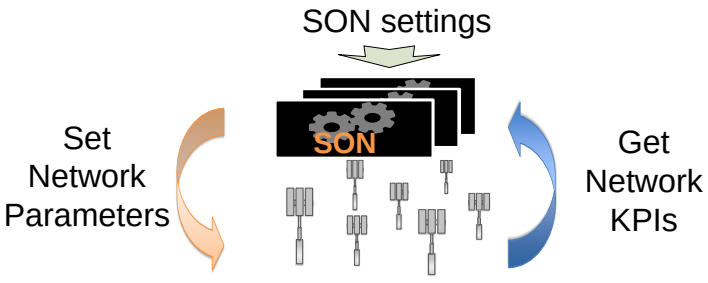


Figure 2.5: SON function

LTE-A is the most significant candidate for a 4G system [2]. However it comes the cost of increased CAPital EXpenditures (CAPEX) and OPerational EXpenditures (OPEX). Envisaging these cost augmentation 3GPP has introduced, already from Release 8, the Self Organizing Network (SON) functions (Fig. 2.5). A SON function is a control loop which tunes the network parameters in order to improve the network Key Performance Indicators (KPIs). It replaces the manual network operation and optimization task. The SON settings allow to alter the behaviour of the SON function, e.g: triggering thresholds, algorithm coefficients etc.

Figure 2.6: Management reference model [100]

There are mainly three categories of SON functions: self-configuration, self-optimization and self-healing [2]. The self-configuration functions are meant to provide a plug-and-play capabilities to the network equipments. Examples of such functions include: Automatic Neighbour Relationship (ANR), Physical Cell ID Allocation, LTE-A relay auto-connectivity, etc. The self-optimization functions tackle the run-time optimization of the network. Examples of such functions include: Mobility Load Balancing (MLB), Mobility Robustness Optimization (MRO), Coverage and Capacity Optimization (CCO), enhanced Inter-Cell Interference Coordination (eICIC), etc. The self-healing functions deal with network failure troubleshooting (e.g. failing cells). Examples of such functions include: Cell Outage Compensation (COC), cell degradation detection and diagnosis, etc.

In the sequel we focus on the self-optimization functions category and SON will refer particularly to this. The 3GPP TR.902 [8] contains the specifications on their input, on their outputs and on their targets. The algorithms inside the SON functions are not standardized, they are vendor specific.

For our work we are particularly interested in the following SON functions:

- MLB : This SON functions aims to balance the load among cells in order to avoid having over-loaded cells. To do so it tunes the mobility parameters, typically the CIO. In the literature we can find a significant amount of work on this: [59, 48, 43, 47, 60, 61, 62, 63, 64, 65] etc.

- MRO: This SON function aims to reduce the number of ping-pong HOs, the number of too late HOs, the number of too early HOs and the number of HOs to wrong cell. It does so by tuning the mobility parameters, typically the CIO and the HYS. In the literature we can find a significant amount of work on this: [66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77] etc.

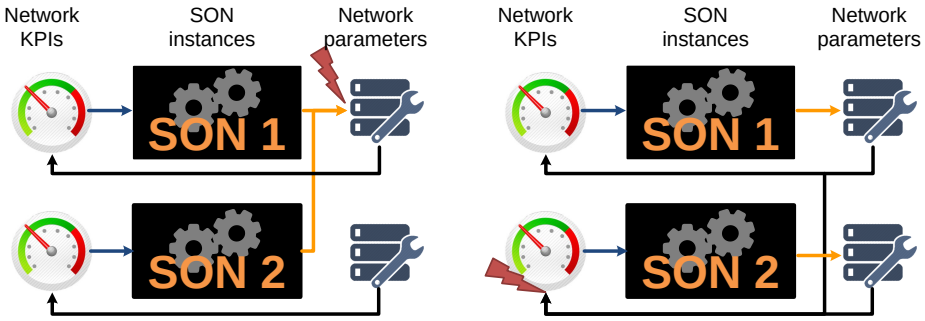
- MLB via Cell Range Expansion(CRE) : This SON function is specific to heterogeneous networks (HetNets), i.e. networks encompassing various cell sizes. It helps the Macro cells off-load on the Pico cells by extending the range of the Pico cells. It tunes the CIOs of the Pico cells. In the literature we can find a significant amount of work on this: [78, 79, 80, 81, 82, 83, 84, 85, 86, 87] etc.

- eICIC: This SON function is HetNet specific. It aims to improve the Quality of Service (QoS) of the Pico users situated at the cell edge. It does so by using ABSs on the Macro cell. Typically it is used to counteract the effects of the CRE. In the literature we can find a significant amount of work on this: [88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99] etc.

In our work we use typical versions of these SON functions. Their details of our SON designs are summarized in Annexes B and C.

The standardization effort on the SON functions architecture (the Integration Reference Point, IRP) is included in the 3GPP 32.500 series, for example TS 32.522 [4]. In Fig. 2.6 we outline the components of this architecture, namely: the Network Manager (NM), the Domain Manager (DM) and the Element Manager (EM), which contain management functions on different levels, and the Network Element (NE) which is the element to be managed (the eNB). The components of the SON functions and of their management entities are to be spread over the components of this architecture depending on their implementation requirements (i.e. if they are centralized, distributed or hybrid - see [4]). Just informatively, we also identify the most important interfaces: the Northbound Interface (Itf-N), the Southbound Interface (Itf-S) and the Peer-to-Peer Interface (Itf-P2P).

We define a SON (function) instance to be a realization of a SON function (or of one of its components) running on one cell. The SON instances might collaborate among themselves,



(a) parameter conflict (b) measurement conflict

Figure 2.7: SON conflicts

Figure 2.8: Coordination dimensions

especially if they are instances of the same SON function. However this is not always the case especially in a multi-vendor environment, and so, having several independent SON instances in the network might create conflicts and lead to bad network KPIs [2]. We consider two types of potential conflicts (see Fig. 2.7):

- parameter conflicts: 2 SON instances target the same parameter and request opposite changes,
- measurement conflicts: the parameter targeted by one SON instance affects the measurements of a different SON instance.

Two dimensions have to be considered for these conflicts (see Fig. 2.8):

- The SON instance dimension: consider a scenario with independent instances of the same SON function: say MLB. The instances can correspond to two vendors, but even in the case of one vendor we might be facing situations like this.
- The SON function dimension: consider a scenarios with independent instances of two different SON functions: say MLB and MRO. As for the previous dimension this might be the case in a multi-vendor environment, but not only.

2.2.1 SON Coordination

SON Coordination in 3GPP

To deal with potential SON conflicts Release 10 of 3GPP has introduced the SONCO function [3], [4]. According to [4], the tasks of a SONCO include:

- to Avoid as much as possible having conflicts (SONCO-A),
- to Detect and to Diagnose the conflicts (SONCO-D),
- to provide conflict Resolution mechanisms when needed (SONCO-R).

The SONCO-A must properly choose the settings of the SON functions such that the conflicts are reduced to a minimum and the network still achieves its target performance. The SONCO-D has to be able to identify what are the potential conflicts and which of them are *active* conflicts, i.e. they effectively degrade the network performance. One consequent actions might be to change the settings of the SON functions in order to prevent the conflict from occurring in the future (via the SONCO-A), i.e. make the conflict *in-active*. Another consequent action could be to employ a resolution mechanism (SONCO-R) which does not eliminate the conflict but

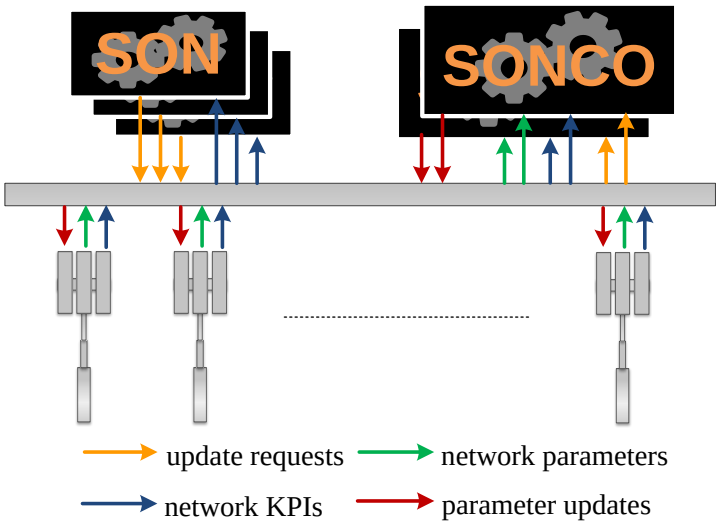


Figure 2.9: SON - SONCO instances interaction

instead, at the moment when the conflict occurs, it shall take the decision to favour one SON function or another based on predefined criteria. Thus, every update request must go through the SONCO-R (see Fig 2.9). Based on the conflict types the tasks of the SONCO-R are:

- measurement conflict resolution: the aim is to improve network stability by reducing the number of unnecessary parameter changes (those which are undone shortly after) caused by the dependence between the parameter tuned by one SON instance and the input of another.
- parameter conflict resolution: the aim is to find a judicious way to decide which of update requests to accept and which to deny when facing opposite requests from different SON instances targeting the same parameter

We define, a SONCO-A/D/R instance to be a realization of a SONCO-A/D/R function (or of one of its components).

SON related requirements

In case of a conflict, the SONCO-R must know when to favour one SON instance or another. For example say that a SONCO-R receives two opposite requests, one from an MLB instance and the other from an MRO instance. The question is which of them to accept? In the literature we typically find update requests encompassing 3 possible values representing the request to increase, decrease and maintain the value of the targeted parameter [28, 29, 44, 33, 21, 30] (Fig. 2.10a). At most it also specifies by how much the value should be changed. We propose that the update requests shall also include the information on how *critical* they are (Fig. 2.10b). This sort of information would allow to judiciously tackle a conflict in favour of the most critical request, thus enforcing a certain degree of fairness among the SON functions. More details on how fairness is achieved will be presented in chapter 4. On this note we propose that each update request u (see Fig. 2.9) is a real value for which we adopt the following convention:

- The case $u > 0$ is an increase request. The bigger $|u|$ is the more critical the request is.
- The case $u < 0$ is a decrease request. As before, the bigger $|u|$ is the more critical the request is.
- The case $u = 0$ is a maintain request.

. We make use of this feature in the SONCO-R designs presented in Section 4. Note that in our work we consider that the *criticalness* indicator is established by the SON instance sending

the request. In case we would not trust the SON instance, i.e. in case it is too aggressive and exaggerates on the criticalness, then this indicator could also be calculated outside the SON instance, by another entity. The output of such an entity should simply reflect the gap between the current network performance and the targeted network performance. In our work, the update requests do not include any information related to by how much the parameters should be altered.

(a) Simple (b) With criticalness indicator

Figure 2.10: Update requests

2.2.2 SEMAFOUR project

Most of the work of this thesis has been done within the framework of the European project SEMAFOUR [57]. In this subsection we provide some details concerning this project and we show how our work fits in it. Fist we note that it is not the first project to tackle SON functions and their management. Similar investigations have been done in UniverSelf [101] and SOCRATES [56]. Due to the increasing interest on the topic, the project SEMAFOUR aimed to further improve the design solutions of the SON functions and of their management. The functional architecture of the Integrated SON Management is represented in Fig. 2.11.

The interaction of the SON functions with the network operator is mainly performed through the Policy-Based SON Management (PBSM). The PBSM introduces a functionality that maps operator-defined technical objectives into the appropriate configuration for the individual SON functions. These settings should instrument the SON functions to work towards the defined objectives. Thus, the network operator can focus on high level objectives, whereas how they are achieved is transparent for it.

The SONCO deals with the conflicts that may occur among the SON functions. On this note it performs the following: conflict detection, conflict resolution and troubleshooting. Our work focuses on the SONCO in particular. As represented in Figure 2.11 its tasks overlap with the ones of some of the other components.

The Monitoring and Diagnosis (MD) collects and evaluates the network performance information. The results are provided to the PBSM, to the SONCO and to the Decision Support System (DSS).

The DSS provides upgrade suggestions to the operator anticipating the network traffic evolution.

2.3 Simulator snapshot

All the previously mentioned elements are put together in an LTE simulator built using C++ and GNUplot. The major components are represented in Fig. 2.12. The simulator has a time granularity of 1 ms, in order to properly assess the HO events. At each time step: i) the UEs move, ii) their radio conditions get updated, iii) the communication procedure gets updated, iv) the UEs (in the *Receive Data* procedure) get scheduled and receive data accordingly, v) users that finished the download (or cannot reconnect to a cell) are deleted, vi) new users are generated in a random manner and vii) the SON and the SONCO instances collect measurements and check their timers.

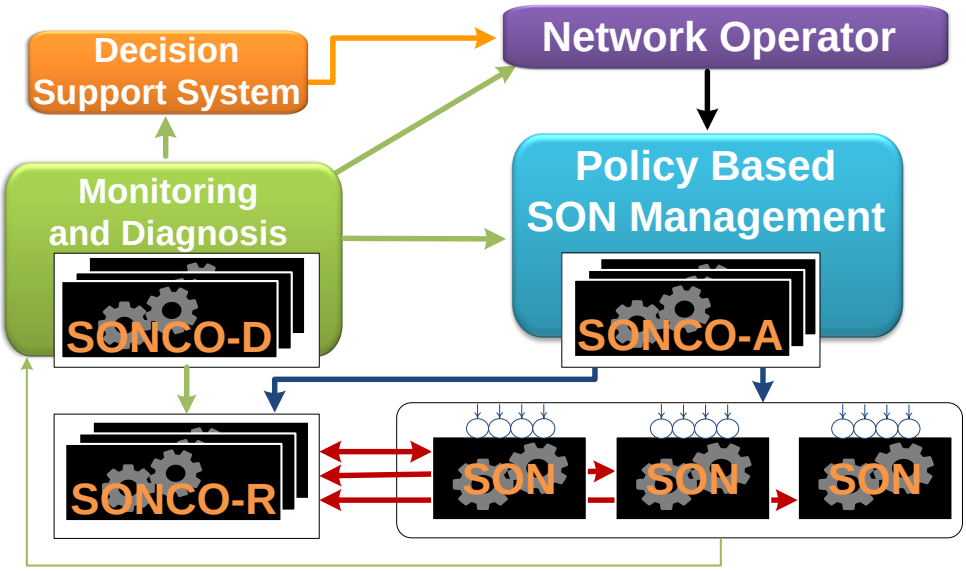


Figure 2.11: SEMAFOUR Integrated SON Management functional architecture [57]

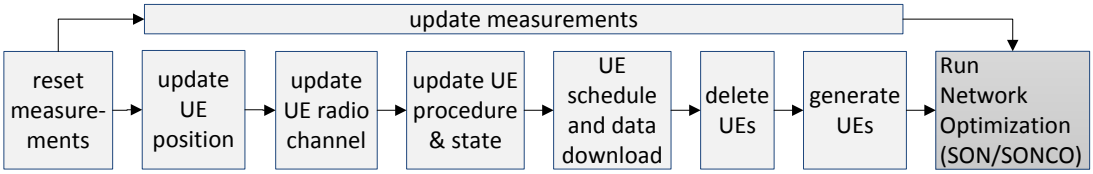


Figure 2.12: Simulator Cycle - Block Diagram

The measurements are for example: the HO event counter, the RLFs counter, the Too late HO event counter, the load of the cells, the counter for number of UEs in the network, the cell/UEs throughout, etc. We reset them at the beginning of each time step. They are updated throughout each time step. The measurements serve as inputs as inputs to the network optimization functions, i.e. the SON functions.

The SON mechanisms make use of a *guard window* where the SON functions do not collect measurements, for instance immediately after a network parameter has been changed (see Fig. 2.13). In all our simulations we have considered the SON functions and the SONCO to be synchronous. They run their algorithm at the end of a time window of size T (a multiple of the time granularity). The SONCO-R receives and enforces the update requests. All SONCO components can have access to measurements and network parameters.

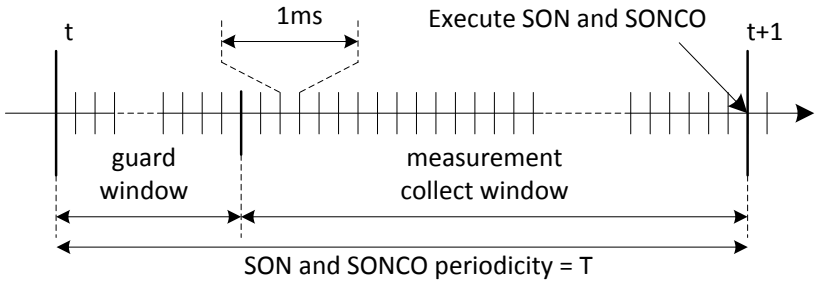


Figure 2.13: SON and SONCO in the simulator

For debugging and presentation purposes we use a graphic interface built with GNUplot. We present it in Fig. 2.14 where we identify Macro cells (sectors) in cyan , Pico cells in green and UEs connected with (black) lines to the serving cells.

Figure 2.14: GNUplot interface

Chapter 3

SON Conflict Diagnosis

In this chapter we propose and formalize a design for the SON functions' conflict diagnosis. First we establish what a faulty network is in terms of network KPIs. Then we show how to identify all potential causes. We consider that the network is properly dimensioned, so the target KPIs could be achieved by using the SON functions. This means that we assume that the only causes of network failures are the misconfiguration of the SON functions' settings. Based on these assumptions we formalize a diagnosis method based on the Naive Bayesian Classifier which we use to identify most probable cause for bad network KPIs.

The contribution of this chapter is summarized as follows: i) we propose an operator-centric SONCO-D, considering the SON functions as *black-boxes*, ii) we propose and formalize a SONCO-D design based on NBC and iii) we provide a study case in a heterogeneous network (HetNet) scenario with 3 SON functions: Mobility Load Balancing (MLB) via Cell Range Expansion (CRE), Mobility Robustness Optimization (MRO) and enhanced Inter-Cell Interference Coordination (eICIC).

In section 3.2 we provide the system description, section 3.3 contains the diagnosis methodology based on NBC, results are contained in section 3.5 and section 3.7 concludes the chapter.

3.1 State of the Art

Typically a troubleshooting mechanism consists of 3 steps: fault detection, cause diagnosis and solution deployment. In this section we focus on the first two steps. We employ a fault detection mechanism which evaluates if we have bad KPIs (making use of some predefined thresholds), and focus mainly on the conflict diagnosis (SONCO-D). Diagnosing parameter conflicts is straightforward, i.e. a simple check whether or not there are opposite requests targeting the same parameter could be sufficient. However measurement conflicts are more difficult to diagnose, thus we focus on these conflicts in particular.

The SONCO-D is responsible for identifying which of the SON functions (and potentially which of their settings) are causing the conflicts. The SONCO-D concept was introduced fairly recently in 3GPP (Release 10).

Some tangent work on the SON diagnosis can be found in the literature. One approach is to consider the algorithm inside the SON function to be unknown, i.e. the SON functions are seen as *black-boxes*. In [10] the authors make use of an *undo* function in case the KPIs have moved away from their targets. The evaluation is done for Automatic Neighbour Relation (ANR) and Physical Cell Identity (PCI) allocation. In [13] an *undo* function is applied in case a fault occurs after a network parameter update. Evaluations are provided with CCO and MRO. In [14] the authors make use of alarms raised by the SON functions which cannot achieve their target performance indicators solely by means of their tuned parameters, and so the alarms trigger other SON functions which can help on this point. Other works consider the algorithm inside the SON functions to be known, i.e. SON functions are seen *as white-boxes*, e.g. [15] where the diagnosis is done by studying the stability of a system of ordinary differential equations which represent the approximations of the SON functions.

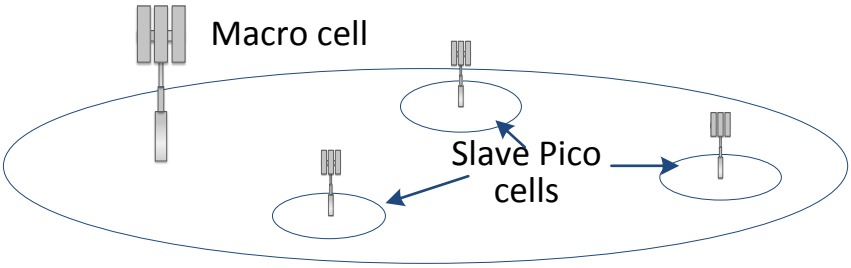


Figure 3.1: HetNet

In our work we consider the SON functions as *black-boxes*. To render the diagnosis process autonomous there are several algorithms which can be used like: linear classifiers, decision trees, neural networks, etc. We employ a mechanism based on a linear classifier, namely the Naive Bayes Classifier (NBC) [16], as it has proved itself to be a good diagnosis technique. The NBC makes use of a probabilistic model, linking the fault symptoms (performance indicators) and a list of potential causes. First this NBC has to be trained using some labelled data, i.e. where the cause is already (manually) identified. Then it can be applied to diagnose the unlabelled data. For example, in [17] and [18] the authors use the NBC in troubleshooting UMTS networks. In [11, 12] and [19]the authors focus on the knowledge acquisition for the NBC model. In [20] we have an evaluation comparing the performances of different levels of NBC model approximations. In this work we use NBC to identify the misconfiguration (bad settings) of the SON functions.

3.2 System description

Consider a network segment containing several Macro cells and Pico cells. For simplicity (of notations) we focus on a cell cluster (i.e. a group of cells) formed by a Macro cell and the $N-1$ slave Pico cells under its coverage meant to boost the capacity (see Fig. 3.1). We index the Macro cell with $n = 1$ and the Pico cells with $\mathcal{N} = \{2, .., N\}$.

Let the Cell Individual Offset (CIO), the HandOver (HO) Hysteresis (HYS) and the number of Almost Blank Sub-frames (ABS) be the $K = 3$ network parameters of interest, indexed (in this order) by $\mathcal{K} = \{1, .., K\}$. The CIO and the HYS are two parameters that are used in mobility management as presented in section 2.1.1. The ABS establishes the number of sub-frames where the (Macro) cell will not transmit data in order to reduce interference to the slave Pico cells [6], see Section 2.1.2.

We employ $Z = 3$ SON functions: the CRE function, the MRO function and the eICIC function indexed (in this order) by $\mathcal{Z} = \{1, ..., Z\}$ (see Fig. 3.2). We consider them to be synchronized, i.e. they perform the desired network parameter changes periodically at the end of a time interval of size $T = 5$min. All the update requests are accepted and immediately enforced in the network. The description of the SON functions is given in Appendix C. Briefly, at a given time instance (we disregard the time index) :

(MLB via) CRE takes as input measurements the vector of loads (LD) of all cells $M_{1,LD}$, ..., $M_{N,LD}$. Its task is to reduce the maximum load (among all cells), i.e. avoid having overloaded cells. For simplicity in the sequel we just call it CRE. Let

$$M_{LD} = \max_{n \in \{2,...,N\}} M_{n,LD}. \tag{3.1}$$

CRE creates update requests targeting the CIO of the Pico cells. The main settings of the CRE are: the thresholds used to trigger parameter changes ($\mathbb{T}_{LD}^{L/H}$). Let Δ_{CIO} be the CIO step size and 0, Δ_{CIO}, $2\Delta_{CIO}$,... the CIO possible values. For simplicity, we consider the problems caused by the size of Δ_{CIO} as a CRE setting problem, even-though Δ_{CIO} is not a CRE setting.

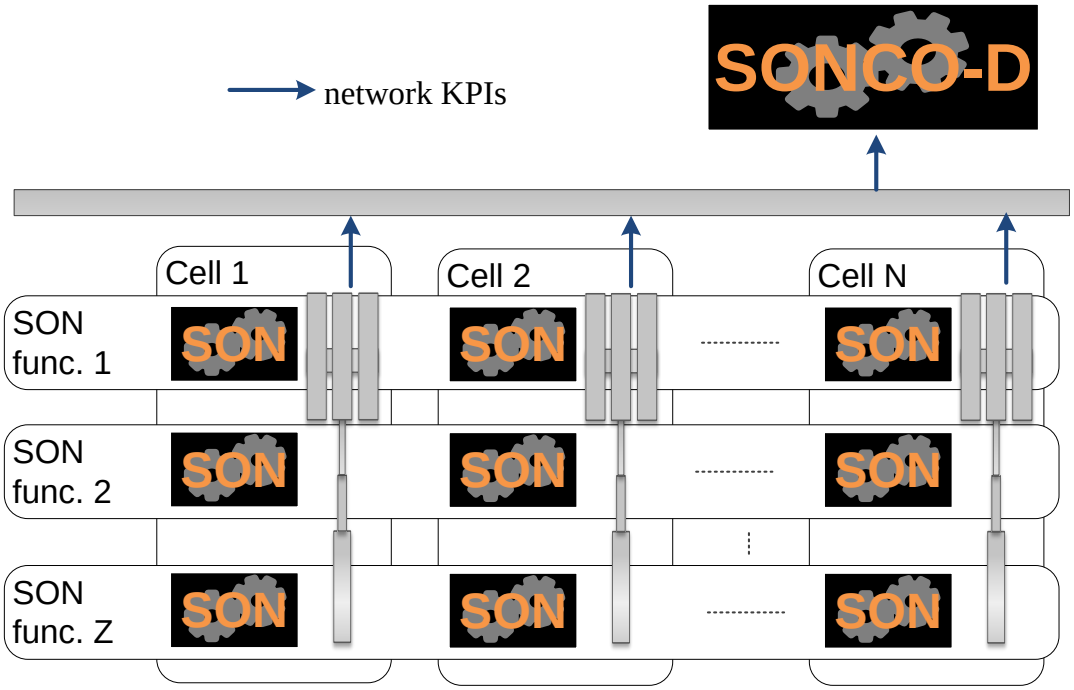


Figure 3.2: SONCO-D ↔ SON interactions

MRO takes as input measurements the percentage of too late HandOvers (HOs) (TL) $M_{2,TL}$, ..., $M_{N,TL}$ and of ping pong HOs (PP) $M_{2,PP}, ..., M_{N,PP}$ (over all HOs) originating only from the Pico cells. We do not look at the ones originating from the Macro cells as typically they are considerably less frequent. Its task is to reduce the averages (over all Pico cells) of the two measurements. Let

$$M_{TL} = \sum_{n\in\{2,...,N\}} M_{n,TL}/N \tag{3.2}$$

and

$$M_{PP} = \sum_{n\in\{2,...,N\}} M_{n,PP}/N. \tag{3.3}$$

MRO creates update requests targeting the HYS of the Pico cells. The MRO instance could also target the CIO but we do not activate this feature in order to avoid parameter conflicts. This feature will be used in other chapters. The settings of the MRO are the thresholds used to trigger parameter changes ($\mathbb{T}_{TL}^{L/H}$ and $\mathbb{T}_{PP}^{L/H}$). Let Δ_{HYS} be the HYS step-size and 0, Δ_{HYS}, $2\Delta_{HYS}$,... the HYS possible values. For simplicity, we consider the problems caused by the size of Δ_{HYS} as a MRO setting problem, even-though Δ_{HYS} is not an MRO setting.

eICIC takes as input measurements the throughput ratio (TR), specifically the ratio between the average throughput of the Macro users and the average throughput of all the *protected users* $M_{1,TR}$. By protected users we refer to all other (Pico) users that would be attached to the Macro cell if the CIO, HYS and TTT were null. Let

$$M_{TR} = M_{1,TR} \tag{3.4}$$

Its target is to keep this ratio between certain bounds in order to guaranty some degree of fairness among the network users (i.e. comparable throughput) . For this eICIC creates update requests targeting the ABS on the Macro cell. The settings of the eICIC are: the thresholds used to trigger parameter changes ($\mathbb{T}_{TR}^{L/H}$). Let Δ_{ABS} be the step-size of the number of ABS and 0, Δ_{ABS}, $2\Delta_{ABS}$,... the ABS possible values. For simplicity, we consider the problems caused by the size of Δ_{ABS} as a eICIC setting problem, even-though Δ_{ABS} is not an eICIC setting.

We consider that all the update requests are accepted (by the SONCO-R) and immediately enforced in the network, thus the criticalness indicator has no impact in this chapter. Hence the input of the SONCO-D consists of all the components of the $I = 4$ measurements of interest: LD, TL, PP and TR which we index (in this order) by $\mathcal{I} = \{1, .., I\}$.

3.3 SON conflict diagnosis

In this section we first identify the network KPIs which we also call symptoms, that are used to identify the network faults. Then we identify the set of potential conflicts among the SON functions representing the set of causes for the network failures. Finally we provide the methodology to diagnose the network faults which links the observation of certain symptom values to the most probable cause.

3.3.1 Symptoms

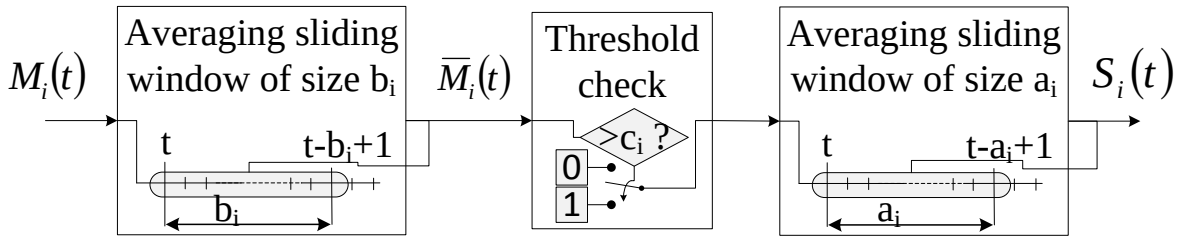


Figure 3.3: Symptom calculation

As mentioned, the SON functions look at the measurements and change the network parameters accordingly at the end of every time step (of size T). We use $I = 4$ symptoms calculated at every time step $t \in \mathbb{N}$ each of which is based on one of the 4 measurements. First we calculate the average of the aggregated measurements $M_i(t)$ within a sliding time window of size b_i (see Fig. 3.3), for any $i \in \{LD, TL, PP, TR\}$, as:

$$\bar{M}_i(t) = \frac{1}{b_i} \sum_{j=0}^{b_i-1} M_i(t-j) \tag{3.5}$$

Next we define the corresponding symptoms as the percentage of time the average $\bar{M}_i(t)$ is larger than a threshold c_i, considering a sliding time-window of size a_i:

$$S_i(t) = \frac{100}{a_i} \sum_{j=0}^{a_i-1} \mathbb{I}\left(\bar{M}_i(t-j) > c_i\right) \tag{3.6}$$

where $\mathbb{I}$ is the indicator function ($\mathbb{I}(\texttt{true}) = 1$, $\mathbb{I}(\texttt{false}) = 0$). The values of a_i, b_i and c_i ($\forall i \in \{LD, TL, PP, TR\}$) are provided later on in section 3.5. Let $S(t) = (S_i(t))_{i \in \{LT, TL, PP, TR\}}$, for any $t \in \mathbb{N}$.

We define $\mathbb{S}_i$, $\forall i \in \{LT, TL, PP, TR\}$, to be the target performance w.r.t. the symptoms. For a given time instance $t \in \mathbb{N}$, we say that there is a network fault if $\exists i \in \{LT, TL, PP, TR\}$ s.t. $S_i(t) > \mathbb{S}_i$. For simplicity, in the sequel we disregard the time index.

3.3.2 Causes

In this section we wish to identify the potential causes of network failures. An easy solution would be to consider that any of the SON functions (for some given settings) could impact some other SON functions by degrading their performance, and so any of them may represent a cause.

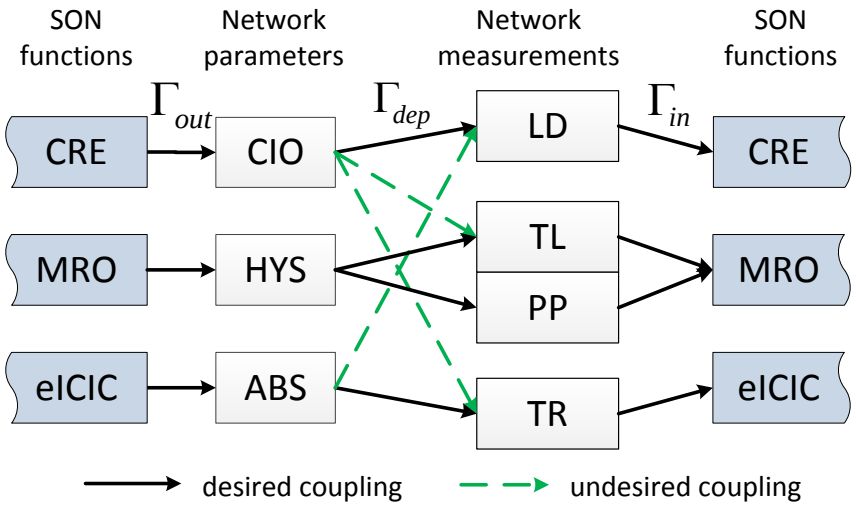


Figure 3.4: SON interaction

However, in order to avoid wasting resources (especially for when we have many SON functions in the network), we next propose a method to precisely identify the potential conflicts, i.e. the minimum dictionary of causes.

Let Γ_{out} be the $Z \times K$ matrix where $\Gamma_{out}(z,k) = 1$ if SON z tunes the parameter k and $\Gamma_{out}(z,k) = 0$ otherwise. In our example (see Fig. 3.4) we have:

$$\Gamma_{out} = \begin{array}{c} \\ \text{CRE} \\ \text{MRO} \\ \text{eICIC} \end{array} \begin{array}{c} \begin{array}{ccc} \text{CIO} & \text{HYS} & \text{ABS} \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \tag{3.7}$$

We note that it just happens that Γ_{out} is a unit matrix, it is not a rule.

Let Γ_{in} be the $I \times Z$ matrix where $\Gamma_{in}(i,z) = 1$ if SON function z takes as input measurement i and $\Gamma_{in}(i,z) = 0$ otherwise. In our example (see Fig. 3.4) we have:

$$\Gamma_{in} = \begin{array}{c} \\ \text{LD} \\ \text{TL} \\ \text{PP} \\ \text{TR} \end{array} \begin{array}{c} \begin{array}{ccc} \text{CRE} & \text{MRO} & \text{eICIC} \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \tag{3.8}$$

Next, let Γ_{dep} be the $K \times I$ (dependencies) matrix where $\Gamma_{dep}(k,i) = 1$ if parameter k is expected to have a significant impact on measurement i and $\Gamma_{dep}(k,i) = 0$ otherwise. This matrix is obtained through expert knowledge, i.e. experienced mobile network engineers. In our example we propose the following matrix:

$$\Gamma_{dep} = \begin{array}{c} \\ \text{CIO} \\ \text{HYS} \\ \text{ABS} \end{array} \begin{array}{c} \begin{array}{cccc} \text{LD} & \text{TL} & \text{PP} & \text{TR} \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \end{array} \tag{3.9}$$

where we consider that the CIO does not significantly impact the PP, the HYS does not significantly impact the LD and the TR, and ABS does not significantly impact the TL and the PP.

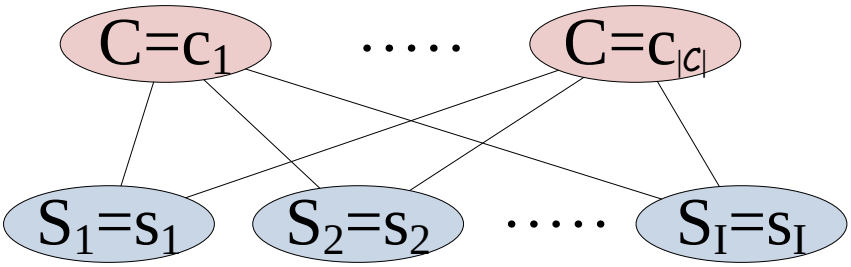


Figure 3.5: Naive Bayes Classifier

In order to find out which SON functions may end up conflicting we simply calculate, using Boolean algebra ($1 \equiv \texttt{true}$, $0 \equiv \texttt{false}$):

$$\Gamma = \Gamma_{out}\Gamma_{dep}\Gamma_{in} = \begin{array}{c} \\ \text{CRE} \\ \text{MRO} \\ \text{eICIC} \end{array} \begin{array}{c} \begin{array}{ccc} \text{CRE} & \text{MRO} & \text{eICIC} \end{array} \\ \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \end{array} \tag{3.10}$$

If $\Gamma(z_1, z_2) = 1$ it means that SON function z_1 may have a negative impact (conflict) on SON function z_2. Note that CRE may impact MRO and eICIC, and eICIC may impact CRE.

From eq. (3.10) we identify that 2 SONs may be the root cause of conflicts, namely CRE and eICIC. Thus we define a 1^{st} *order* cause-set:

$$\mathcal{C}_1 = \{\text{CRE}, \text{eICIC}\} \tag{3.11}$$

with cardinality $|\mathcal{C}_1| = 2$. We note that the problem may come from one of the different settings of the SON functions like: the triggering thresholds ($\mathbb{T}$) or the step size for the tuned parameters (Δ), and so we define the 2^{nd} *order* cause-set:

$$\mathcal{C}_2 = \{\text{CRE}, \text{eICIC}\} \times \{\mathbb{T}, \Delta\} \tag{3.12}$$

with cardinality $|\mathcal{C}_2| = 4$. Moreover, if we go into even more details we can also target to identify the alteration degree of the problematic SON setting (i.e. how far it is from the value which would ensure a good performance) : low (`l`), medium (`m`), high (`h`) and so we define the 3^{rd} *order* cause-set:

$$\mathcal{C}_3 = \{\text{CRE}, \text{eICIC}\} \times \{\mathbb{T}, \Delta\} \times \{\texttt{l}, \texttt{m}, \texttt{h}\}, \tag{3.13}$$

with cardinality $|\mathcal{C}_3| = 12$. In this work we compare the diagnosis performance for the 3 orders of cause-sets.

3.4 Diagnosis based on the Naive Bayes Classifier

In the previous sections we have identified the set of symptoms and 3 sets of causes corresponding to the 3 *orders* of details. If a network fault occurs then to find the most probable cause, say of order l, we have to calculate:

$$\max_{c \in \mathcal{C}_l} \mathbb{P}(C = c | S = s)$$

where $S = (S_i)_{i \in \{LT,TL,PP,TR\}}$ represents the observed symptoms' values (see Fig. 3.5). According to Bayes' rule we can write, $\forall c \in \mathcal{C}$:

$$\mathbb{P}(C = c | S = s) = \frac{\mathbb{P}(S = s | C = c)\,\mathbb{P}(C = c)}{P(S = s)} \tag{3.14}$$

Note that $P(S = s)$ is the same for all causes so, in order to identify the most probable cause, we do not necessarily need to calculate it. Thus we focus only on the numerator. $\mathbb{P}(C = c)$ is

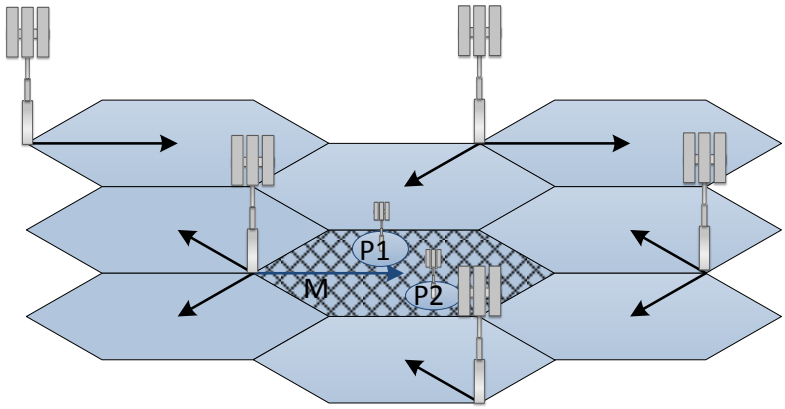


Figure 3.6: Network topology

simply the probability that a given cause may occur, so it can be easily obtained. Calculating/ estimating $\mathbb{P}(S = s|C = c)$ can be quite difficult and impractical especially if we have a lot of symptoms. For this reason we assume i) the symptoms are independent and ii) there can be only one cause at a time. Even if in reality this is rarely the case, still, significant results have been obtained under these assumptions (e.g. [17]). The simplicity of the model has its clear benefits. Consequently eq. (3.14) becomes:

$$\mathbb{P}(C = c|S = s) = \frac{\prod_i \mathbb{P}(S_i = s_i|C = c)\,\mathbb{P}(C = c)}{\mathbb{P}(S = s)} \tag{3.15}$$

The above equation represents a so-called NBC, and it serves as our *diagnosis model*. The cause *inference method* simply outputs the most probable cause according to eq. (3.15). This is known as the *maximum a posteriori* decision rule [16].

The methodology of using the NBC has two steps:

Learning: we estimate $\mathbb{P}(S_i = s_i|C = c)$,$\forall i \in \{LT, TL, PP, TR\}$, and $\mathbb{P}(C = c)$ by using a some labelled data, i.e. where the cause has been manually identified. In a real network this means that an experience engineer has to troubleshoot/label several network faults and create a database containing the symptoms and the cause. In our case we will obtain such data by means of simulations, see section 3.5.

Exploitation: we apply the inference method to unlabelled data. In reality, we would say that the cause was correctly identified if the solution associated to this cause fixes the network failure. In our case, as we are orchestrating the simulations, the cause is known a priory and we compare it with the output of the inference mechanism, see section 3.5.

3.5 Simulation results

In this section we target to analyse the feasibility of using NBC for the design of the SONCO-D considering the 3 different orders of (cause) details. For this we employ a network with 9 Macro cells, see area in Fig. 3.1. In one of the Macro cells (call it 'M') we place randomly 2 slave Pico cells (call them 'P1' and 'P2'). We make use of a wraparound technique. The SON functions will be applied to the cell cluster formed by M, P1 and P2. Thus, the measurement data $M_i(t)$, $\forall i \in \{LD, TL, PP, TR\}$, comes from this cell cluster.

We consider an FTP-like traffic. User Equipments (UEs) arrive randomly in the network according to a Space Poisson Point Process (SPPP) [58]. We consider a UE arrival rate Λ_G[UEs/s] over all the network and an additional HotSpot (HS) UE arrival rate Λ_{HS}[Mb/s] . The arrival rate is subject to periodic (2 hours) variations as follows, $\forall j \in \{G, HS\}$:

$$\Lambda_j(\text{time}[h]) = \lambda_j(0.9 - 0.1 \cdot \sin(\pi \cdot \text{time}[h]))$$

where time$[h]$ is the time stamp measured in hours and λ_j is an average UE arrival rate. UEs arrive in the network, transmit their file (fixed size: $FS[Mb/UE]$) and then leave the network. For details see Table 3.1.

Parameter	Value
Number cells	9 Macro cells, 2 Pico cells
Propagation Model	#2 in Table A.1 (3GPP Case 3 [54])
Bandwidth	10 MHz
TxPow	49 dBm / 30 dBm
Traffic type	FTP-like , constant file size,
FS	$16[Mb/UE]$
$\lambda_G \cdot FS$	$12.5[Mb/s/Macro\ Hexagon\ Area]$
$\lambda_{HS} \cdot FS$	$12.5[Mb/s/Pico\ HS\ Area]$
HotSpot	radius: 100m

Table 3.1: Simulator summary

We consider 13 scenarios with different SON function settings and for each we run simulations of 2400 hours in order to generate enough data to test the NBC. One of them is the *reference* (REF) scenario where there are no network faults and the other 12 correspond to the causes in $\mathcal{C}_3$ (and implicitly in $\mathcal{C}_2$ and $\mathcal{C}_1$). For details see Table 3.2. We make use of the symptom parameters in Table 3.3.

SON	CRE			eICIC			MRO
Setting	$\mathbb{T}^H_{LD}$ [ratio]	$\mathbb{T}^L_{LD} =$ [ratio]	Δ_{CIO} [dB]	$\mathbb{T}^H_{TR}$ [ratio]	$\mathbb{T}^L_{TR}$ [ratio]	Δ_{nABS} [ratio]	
REF.	0.85	$\mathbb{T}^H_{LD} - 0.1$	1	2.1	1.6	1	$\mathbb{T}^H_{TL} = 0.1$ [ratio], $\mathbb{T}^H_{PP} = 0.2$ [ratio], $\Delta_{HYS} = 1$ [dB]
CRE-$\mathbb{T}$-l	0.80		1	2.1	1.6	1/40	
CRE-$\mathbb{T}$-m	0.75		1	2.1	1.6	1/40	
CRE-$\mathbb{T}$-h	0.70		1	2.1	1.6	1/40	
CRE-Δ-l	0.85		3	2.1	1.6	1/40	
CRE-Δ-m	0.85		5	2.1	1.6	1/40	
CRE-Δ-h	0.85		6	2.1	1.6	1/40	
eICIC-$\mathbb{T}$-l	0.85		1	1.9	1.4	1/40	
eICIC-$\mathbb{T}$-m	0.85		1	1.7	1.2	1/40	
eICIC-$\mathbb{T}$-h	0.85		1	1.5	1.0	1/40	
eICIC-Δ-l	0.85		1	2.1	1.6	3/40	
eICIC-Δ-m	0.85		1	2.1	1.6	5/40	
eICIC-Δ-h	0.85		1	2.1	1.6	6/40	

Table 3.2: Summary of scenarios

Symptom: $i =$	LD	TL	PP	TR
$\mathbb{S}_i$	1.0%	3.5%	4.0%	1.7%
a_i	1440			
b_i	3	12	12	3
c_i	0.85	0.08	0.015	2.25

Table 3.3: Symptom parameters

Fig. 3.7 shows, for all the scenarios, the percentile of time each fault occurs (LD, TL, PP, TR) and the percentile of time at least one fault occurs (LD|TL|PP|TR). Consequently these

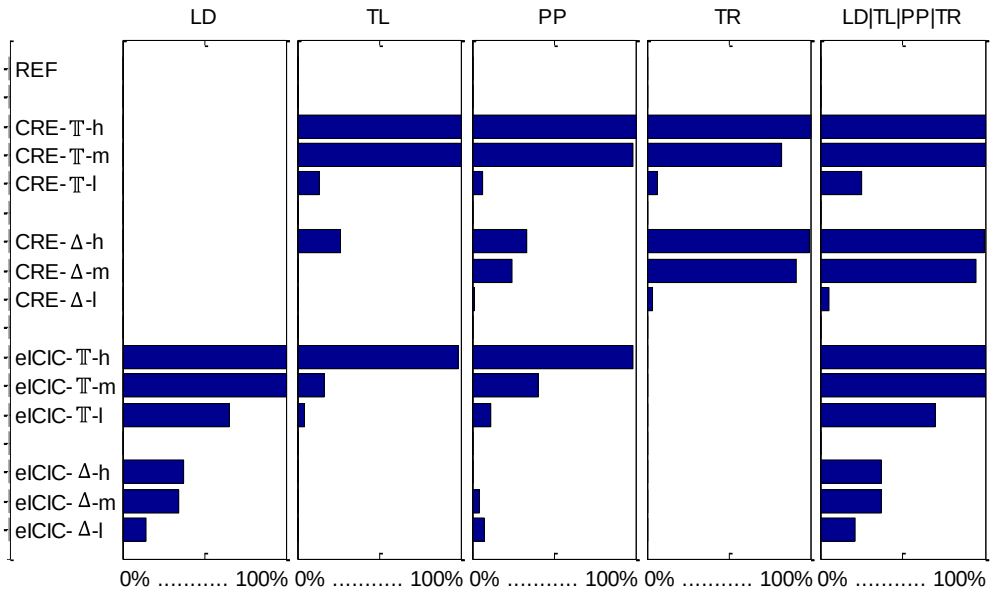


Figure 3.7: Fault frequency

Figure 3.8: Correct fault diagnosis

time-samples where we are experiencing network faults represent the input to our diagnosis as described in Section 3.4.

We select 1000 samples (experiencing network faults) from each of the scenarios except REF which we manually label. Next say that we focus on the l^{th} order cause-set: $\mathcal{C}_l$. We use this labelled data and train the NBC getting $\mathbb{P}(S|C)$ and $\mathbb{P}(C)$ by drawing the corresponding histograms.

We then select 400 other samples (also experiencing network faults) representing the unlabelled data and we test the SONCO-D. The percentiles of correct identification of the fault are presented in Fig. 3.8 for all the 3 orders. We have performances of over 91.7 % for detecting the SON function causing the problem (1^{st} order cause), of over 82.5% if we also want to detect the problematic setting (2^{nd} order cause) and of over 52% in the case where we also want to detect the alteration degree of the setting (3^{rd} order cause). As expected, detecting only the problematic SON is less subject to errors than identifying altogether the problematic SON, the problematic setting and its degree of alteration.

3.6 Logical framework architecture

In this section we present the architecture of the SONCO-D, i.e. how it would be implemented in a real network considering the management reference model in Fig. 2.6. Our SONCO-D only focuses on diagnosing measurement conflicts. Parameter conflicts are straightforward, i.e. only identify opposite requests.

The SONCO-D agents collect information about the network symptoms (e.g. KPIs). This information is forwarded to the SONCO-D algorithm which will output a potential cause (i.e. the malicious SON function). Initially the SONCO-D has to learn the diagnosis model. Thus several manual diagnoses have to be performed. After, when exploiting the diagnosis model, the SONCO-D will provide a potential cause given some symptoms. Consequently some actions have to be taken to deal with the network fault. If the actions corresponding to the cause are efficient, i.e. the fault disappears, then the diagnosis is correct, else it is wrong. This feedback has to be provided to the SONCO-D in order to update the diagnosis model. Thus the SONCO-D instances include several *Agents* (Fig. 3.9) which shall collect, aggregate and forward the

network KPIs to the *Algorithm* entity. The *Algorithm* entity is the one that contains the NBC model and inference mechanism.

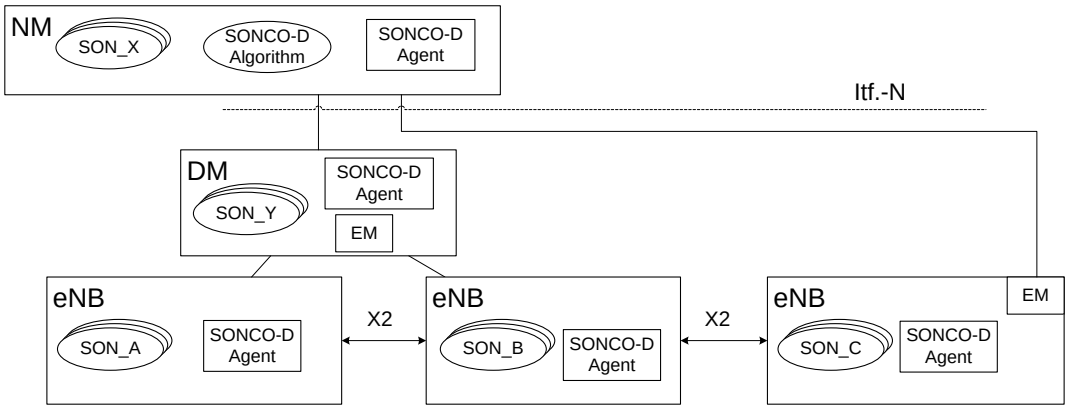


Figure 3.9: SONCO-D architecture

3.7 Concluding remarks

In this chapter we tackled the problem of conflict diagnosis for SON functions. First we provide the methodology to identify all potential conflict sources. Second we present a framework based on NBC that helps identify the most probable cause. As expected, for an increased *order* of details describing the cause (i.e. refined cause dictionary) it is harder to correctly identify the source of a conflict. The performance of detecting only the problematic SON function is the highest followed by the performance of detecting also the problematic setting, and by the performance of detecting also the setting's alteration. Thus we find that NBC represent a promising frame for SON conflict diagnosis. Future work could focus on introducing more KPIs and also alarms which can be raised by the SON functions in order to help the diagnosis. Currently there is not sufficient SON field data in order to apply the SONCO-D to a real scenario, but this could be envisaged in the near future.

Chapter 4

SON Conflict Resolution

After having presented in the previous chapter a method to identify the SON conflicts, in this chapter we propose and formalize a conflict resolution framework (SONCO-R). The SON instances are again considered as *black-boxes*, i.e. the algorithm inside is unknown. All the update requests created by the SON functions are sent to the SONCO-R. In this chapter we exploit the fact that the SON parameter update requests include an additional metric reflecting how *critical* each request is. The SONCO-R decides which requests to accept and which to deny in targeting to minimize a regret function calculated using this *criticalness* indication. As we have no information on the SON *black-box* behaviour, we propose to learn/estimate this behaviour from the past experience, in order to improve the SONCO-R decisions. For this, we formalize the problem in a Reinforcement Learning (RL) framework where we propose and exploit the fact that the RL regret is a function of the update requests. A well known problem of RL is that its complexity does not scale well, in our case w.r.t. the number of cells. We tackle this issue making use of several function approximation techniques. More precisely, given the fact that the Key Performance Indicators (KPIs) of one network cell are not significantly impacted by the parameter configurations of the cells which are faraway, in certain conditions, we are able to estimate the RL regret with local information. Thus we show how, given some specificities of our scenario, we are able to reduce the requirements in terms of memory and computational effort, by using local approximations of the information (regarding the past experience).

As mentioned we employ RL outlining the compromises that have to be made in order to obtain scalable solutions. In section 4.2 we provide the system description, section 4.3 presents the SONCO-R with the underlying Markov Decision Process (MDP) framework and the function approximation, and section 4.4 contains the associated RL algorithm; results are contained in section 4.5 and section 4.7 concludes the chapter.

4.1 State of the Art

In the literature, the problem of SON conflict resolution (SONCO-R) has been addressed in several papers. Some works propose a *joint design* of the SON functions: In [102] we find a joint optimization two SON functions: Coverage and Capacity Optimization (CCO) tuning the antenna tilt and Mobility Load Balancing (MLB) tuning the Cell Individual Offset (CIO). In [21] we find an implicit coordination of MLB and Mobility Robustness Optimization (MRO) obtained through a joint design of the two SON functions. The authors propose a per user optimization that attributes priorities to users for handovers based on the weights attributed to the SON functions. In [22] we find a joint optimization solution for MLB, Inter-Cell Interference Coordination (ICIC) and Energy Saving Management (ESM). In [23] we have a joint implementation of MLB, MRO and Admission Control (AC) using Q-learning and fuzzy learning. In [24] we find a joint optimization scheme for MLB and MRO. This approach only works if the designers of the SON functions are one and the same. In [25] coordination between CCO instances is achieved through cooperation, i.e. message exchanges between CCO instances. In a multi-vendor case, this implies that the messaging between these instances should be standardized which is not always the case.

Another approach is to consider independently designed SON functions whose optimization algorithms are known. Thus the SON functions are seen as *white boxes*. In [15] the authors assume that the algorithm of the SON function is known and propose a mechanism which alters their outputs to eliminate the conflicts. Similar work can be found in [26] and [27].

The third approach that we have identified considers independently designed SON functions whose optimization algorithms are unknown. Thus the SON functions are seen as *black boxes*. On this track we find a great amount of work. It seems that there is a great interest to have multi-vendor SON architecture. In [28] we have a coordination of MLB and MRO through an Alignment function which basically rejects all the MRO requests on the HYS in case we are dealing with overloaded cells (which are the object of MLB). In [29] and [30] we find a coordination between two CCO mechanisms. One tunes the antenna Remote Electrical Tilt (RET) while the other tunes the cell transmission power. The coordination is done making use of a decision tree which checks if the update request of one of the SON functions is conflicting with the requests of the other SON function. In [44] the focus is on conflict avoidance through creating restrictions for the value set of the parameters for MRO and MLB (they both tune the CIO). A similar strategy is found in [32] where the authors consider different Time To Trigger (TTT) values for different User Equipment (UE) speeds. In [33] we find a coordination of MRO and MLB which simply gives priority to one of the SON functions. Specifically, if the SON function with higher priority experiences bad KPIs then the other SON function is shut down or some of its update requests are always denied. In [34] we have a conflict resolution method between MRO and MLB. The authors propose the following algorithm: in case one of the SON function is experiencing bad KPIs, the algorithm truncates the parameters of the other SON function which are presumed to have led to this situation. In [35] we have a coordination mechanism between MLB and MRO based on a decision tree which establishes which of the SON functions is triggered. The coordination mechanism presented in [36] schedules (and enforces) the update requests making use of a token mechanism. In other words each SON function accumulates tokens according to a predefined rule and the update requests are accepted if the SON function has enough tokens. In [37] we find a coordination mechanism for MLB and MRO conflict, where the scopes of the two functions are orthogonalized based on the UE speed as follows: MLB has priority on the low speed UEs and MRO has priority on high speed UEs. In [38] a heuristic framework is proposed where the coordination for the instantaneous requests is performed by means of a decision table. In [39] the authors use a multi-agent Markov Decision Process (MDP) and present two approaches to solve the conflicts between the agents: artificial intelligence and game-theory approaches. No results are provided for any of the approaches. Using a game-theory approach does not guaranty to lead to the optimal solution.

As in all the thesis, in this chapter we consider the SON functions as *black boxes*. Thus, in order to take the best coordination decision, we propose to learn this information from past experience. The SONCO-R shall learn this behaviour and provide the optimal coordination decisions by interacting with the SON functions, i.e. from the consequences of the past decisions. A very effective and well known technique to do this is Reinforcement Learning (RL), specifically the Temporal Difference learning methods [40]. RL is a technique which has been used extensively in the self-optimization context, e.g. for tuning relay-backhaul resource allocation parameters in [41], for tuning the CCO parameters in [103], etc. A well known problem of RL is that its complexity does not scale well, in our case w.r.t. the number of cells. We tackle this issue making use of some function approximation techniques presented in the sequel.

4.2 System Description

In order to simplify notations we make the following convention: for any variable X (be it parameter, update request, action, etc.) if we index it by a set $\mathcal{I}$ the meaning is $X_{\mathcal{I}} = (X_i)_{i \in \mathcal{I}}$.

Consider a network segment composed of C cell clusters (disjoint groups of cells), indexed by $\mathcal{C} = \{1, ..., C\}$ (see Fig. 4.1). The clustering is established on a neighbour relation basis. We focus on one of the clusters, say cluster $c \in \mathcal{C}$, as if it was independent from the other cell

clusters. Let N be the number of cells in this cluster, indexed by $\mathcal{N} = \{1, ..., N\}$. We note that these cells can be of any kind: Macro cells, Pico cells, etc. We also define $\bar{\mathcal{N}}_n$, $\forall n \in \mathcal{N}$, to be the set containing cell n and its direct neighbours (only the ones belonging to cell cluster c).

We employ Z SON functions indexed by $z \in \mathcal{Z} = \{1, ..., Z\}$,e.g.: MLB, MRO, enhanced ICIC (eICIC), MLB via Cell Range Expansion (CRE) etc. Each SON function is instantiated on each and every cell, i.e. N instances per SON function (see Fig. 4.1) . Each SON instance sends update requests to change the parameter configuration of the hosting cell. We also consider `void` requests for the case where the SON instance is not tuning a given parameter.

Say that there are K parameters, indexed by $k \in \mathcal{K} = \{1, ..., K\}$, which can be tuned on any cell by the hosted SON instances, e.g. CIO, HandOver (HO) Hysteresis (HYS), number of Almost Blank Sub-frames (ABS), antenna tilt, etc.

For a given time instance (we disregard the time index), let

- $P_{n,k}$ be the value of parameter $k \in \mathcal{K}$ of cell $n \in \mathcal{N}$ and $\mathcal{P}_{n,k}$ be the set of possible values of $P_{n,k}$. For simplicity, we write $P = P_{\mathcal{N},\mathcal{K}}$ and $\mathcal{P} = \mathcal{P}_{\mathcal{N},\mathcal{K}}$.
- $U^z_{n,k}$ be the update request of the instance of SON function $z \in \mathcal{Z}$ running on cell $n \in \mathcal{N}$ targeting parameter $k \in \mathcal{K}$ ($P_{n,k}$) and $\mathcal{U}^z_{n,k} \subseteq \mathbb{R} \cup \{\texttt{void}\}$ the set of possible values of $U^z_{n,k}$. For simplicity, we write $U = U^{\mathcal{Z}}_{\mathcal{N},\mathcal{K}}$ and $\mathcal{U} = \mathcal{U}^{\mathcal{Z}}_{\mathcal{N},\mathcal{K}}$.

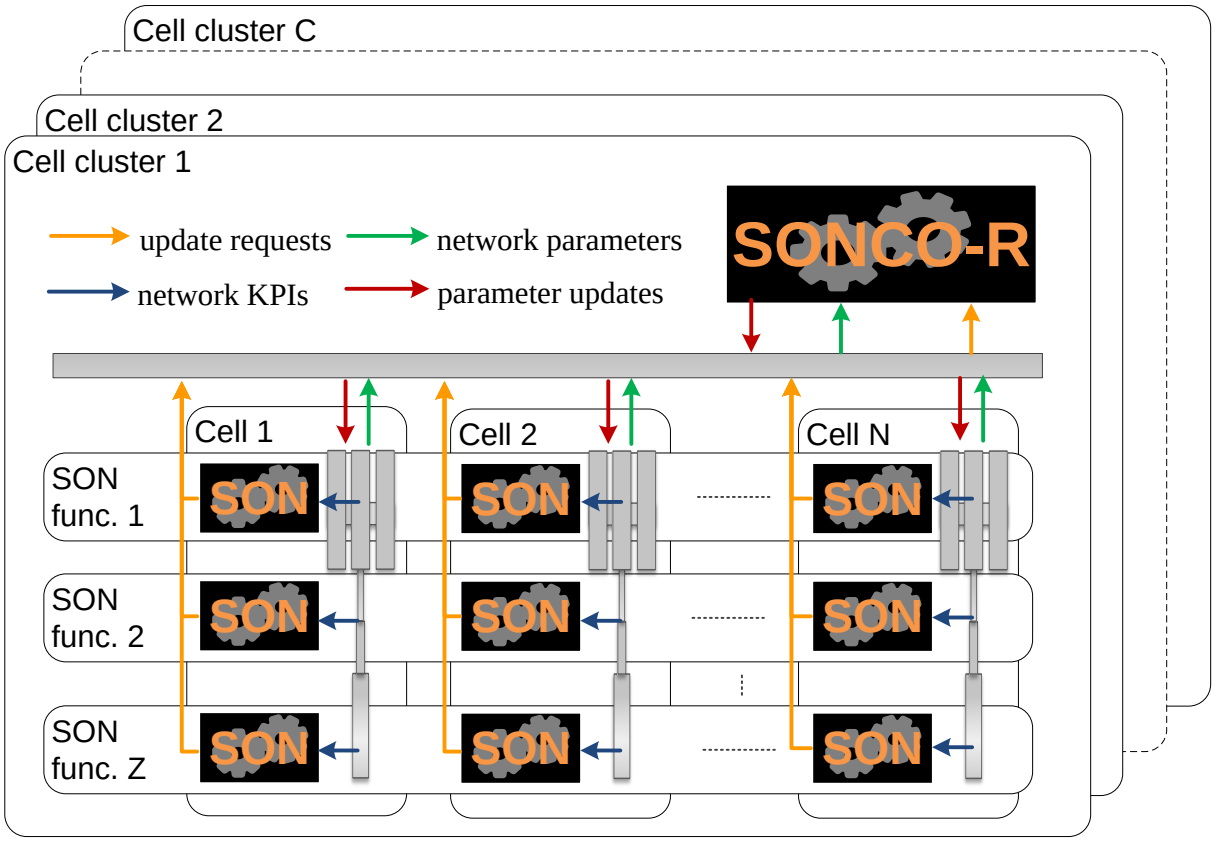


Figure 4.1: SONCO-R ↔ SON interactions

4.3 SON conflict resolution

Instead of directly executing the desired parameter changes in the network, the SON instances send update requests to a SONCO-R instance which decides if they are accepted (and immediately enforced) or denied. The SON instances are seen as *black-boxes* by the SONCO-R, so it does not know the algorithm running inside them. It only knows the current update requests (U) and the current network parameter configuration (P). The task of the SONCO-R is to find a equitable conflict resolution in order to ensure some degree of fairness among SON instances. We

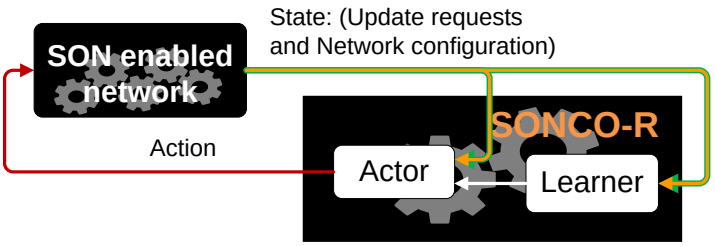


Figure 4.2: Reinforcement Leaner

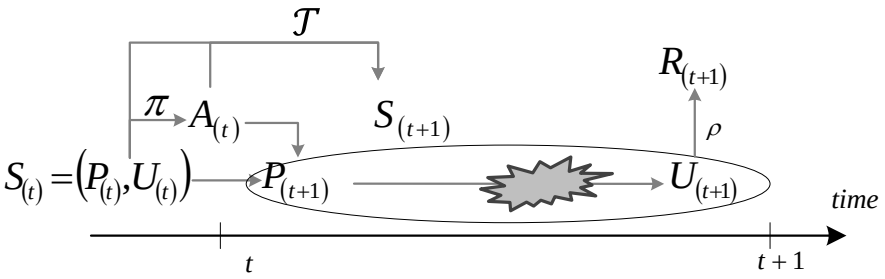


Figure 4.3: Transition Kernel

recall that each update request sent by a SON function also includes an indicator reflecting how *critical* the update is, see Section 2.2.1. Thus, in case we are facing a conflict situation we will choose which requests to accept such that we minimize a *regret* which represents an aggregated *criticalness* (over the update requests). The fairness is ensured through the choice of the *regret* function's by making it reflect the most *critical* update requests.

The SON instances are considered to be synchronized. They send the update requests simultaneously at the end of a periodic time window of size T (see Chapter 2, Fig. 2.13). The SONCO-R is also synchronized with the SON instances. It receives the update requests from all the SON instances and decides which requests are accepted and which are denied. We employ a RL approach as it enables us to use the information on the effect of our past decisions. The RL algorithm consists of a *Learner* and an *Actor* (see Fig. 4.2). The *Leaner* assimilates the information on the impact of past decisions. The *Actor* establishes the current actions (obviously based on the learned information) given the current state. This information is stored in value functions which reflect how satisfied we were with a given configuration. RL is founded on an MDP that is described in the sequel.

4.3.1 Markov Decision Process

We now describe the system dynamics as an interplay between the current network parameter configuration P, the SON update requests U and the SONCO-R decisions A. To that end we interpret them as stochastic processes on some probability space, which are governed by the following MDP (see also Fig. 4.3) :

State space $\mathcal{S}$. A state $s = (p, u) \in \mathcal{S}$ has two components: the configuration of the network parameters ($p = p_{\mathcal{NK}}$) and the update requests ($u = u^{\mathcal{Z}}_{\mathcal{NK}}$). We denote by $\mathcal{S}$ the state space. We define $S_{(t)} = \left(P_{(t)}, U_{(t)}\right)$ as the state process where $t \in \mathbb{N}$ is the time index.

Action space $\mathcal{A}$. An action $a \in \mathcal{A}$ is a N by K matrix, i.e. $a = a_{\mathcal{NK}}$. The action component $a_{n,k}$ allows to increase/decrease the value of a parameter k on cell n only if there exists at least one request to do so, and if the value is not already the maximum/minimum value in the set of possible values $\mathcal{P}_{n,k}$. Concretely the next value of parameter k of cell n given the current value

$p_{n,k}$ is:

$$p'_{n,k} = \begin{cases} \min\{x \in \mathcal{P}_{n,k}\colon\ x > p_{n,k}\}, & \text{if } a_{n,k} > 0,\ p_{n,k} \neq \max\mathcal{P}_{n,k} \text{ and } \exists z \in \mathcal{Z} \text{ s.t. } u^z_{n,k} > 0 \\ \max\{x \in \mathcal{P}_{n,k}\colon\ x < p_{n,k}\}, & \text{if } a_{n,k} < 0,\ p_{n,k} \neq \min\mathcal{P}_{n,k} \text{ and } \exists z \in \mathcal{Z} \text{ s.t. } u^z_{n,k} < 0 \\ p_{n,k}, & \text{otherwise} \end{cases} \tag{4.1}$$

As can be seen from eq. (4.1) the vector p' representing the next network parameter configuration is a deterministic function of the current parameter configuration p, the update requests u and the actions a, i.e:

$$p' = g((p,u),a) = \left(g_{n,k}\left(\left(p_{n,k}, u^{\mathcal{Z}}_{n,k}\right), a_{n,k}\right)\right)_{n,k} \tag{4.2}$$

where the functions $g : \mathcal{P}\times\mathcal{U}\times\mathcal{A} \to \mathcal{P}$ and $g_{n,k} : \mathcal{P}_{n,k}\times\mathcal{U}_{n,k}\times\mathcal{A}_{n,k} \to \mathcal{P}$, for all $(n,k) \in \mathcal{N}\times\mathcal{K}$, are detailed in eq. (4.1). The action is established according to a stochastic policy $\pi : \mathcal{S}\times\mathcal{A} \to [0,1]$ where $\pi(s,a)$ reflects the probability that action a is taken given that the state is s.

Transition kernel $\mathcal{T}$. Let us describe the probability that $s' = (p', u')$ is the next state given that $s = (p,u)$ is the current state and the action is a: $\mathcal{T}(S_{t+1} = s' | S_t = s, A_t = a)$. Recall the key specificity of our transition kernel mentioned above: p' is entirely determined by (s,a): $p' = g(s,a)$. Random effects are only used to model the values of the next update requests as they cannot be deterministically obtained. Recall that the algorithm inside the SON functions is assumed to be unknown and so this lack of knowledge is grasped by the transition kernel. Moreover we make the assumption that the SON functions are memory-less: their output at time t ($\forall t \in \mathbb{N}$) only depend on their inputs at time t which only depend on the network parameter configuration at time t. Formally stated we assume that:

$$\mathbb{P}\left(U_{(t+1)} = u' | S_{(t)} = s, A_{(t)} = a, P_{(t+1)} = p'\right) = \mathbb{P}\left(U_{(t+1)} = u' | P_{(t+1)} = p'\right) \tag{4.3}$$

Therefore by eq. (4.2) and the chain rule our transition kernel model becomes:

$$\begin{aligned} \mathcal{T}\left(S_{t+1} = s' | S_t = s, A_t = a\right) &= \mathbb{P}\left(P_{(t+1)} = p', U_{(t+1)} = u' | S_{(t)} = s, A_{(t)} = a\right) \\ &= \mathbb{I}_{\{p' = g(s,a)\}} \mathbb{P}\left(U_{(t+1)} = u' | P_{(t+1)} = p'\right) \end{aligned}$$

where $\mathbb{I}$ is the indicator function ($\mathbb{I}_{\{\texttt{true}\}} = 1$, $\mathbb{I}_{\{\texttt{false}\}} = 0$).

Regret $\bar{r}(s,a)$ is the expected regret associated to the state-action pair (s,a). We introduce $\left(R_{(t)}\right)_t$ as the process of instantaneous regrets. Consequently, $\forall t \in \mathbb{N}$:

$$\mathbb{E}\left[R_{(t+1)} | S_{(t)} = s, A_{(t)} = a\right] = \bar{r}(s,a)$$

We consider that the instantaneous regret is a function of the update requests:

$$R_{(t)} = \rho\left(U_{(t)}\right) \tag{4.4}$$

for some function $\rho : \mathcal{U} \to \mathbb{R}$. Typically the lower the number of increase/ decrease requests (or the smaller the *criticalness* indicators) the smaller the regret. As an application of the transition kernel, using eq. (4.3) and eq. (4.4), note that:

$$\bar{r}(s,a) = \mathbb{E}\left[\rho\left(U_{(t+1)}\right) | P_{(t+1)} = g(s,a)\right].$$

Assume that ρ can be written as a sum of I components, i.e. $\exists \rho_i : \mathcal{U} \to \mathbb{R}$ (for $i \in \mathcal{I} = \{1, ..., I\}$) such that:

$$\rho(u) = \sum_{i \in \mathcal{I}} \rho_i(u) \tag{4.5}$$

Thus the *instantaneous regret* at time t can be written the sum of I sub-components. For example the regret could be the sum of several sub-components each of which depends only on the update requests of one cell ($U^{\mathcal{Z}}_{n,\mathcal{K}}$). Another example is to have each regret component depending only on one update request ($U^z_{n,k}$). Concrete examples are provided in section 4.5.

4.3.2 Value functions

For any policy π, in order to measure its performance, we introduce the *state-value function* (V^π) and the *action-value function* (Q^π):

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R_{(t+1)} \,|S_{(0)} = s\right] , \tag{4.6}$$

$$Q^\pi(s,a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R_{(t+1)} \,|S_{(0)} = s, A_{(0)} = a\right] . \tag{4.7}$$

where $0 \leq \gamma < 1$ is the regret sum discount factor and $\mathbb{E}_\pi$ is the expectation given that the followed policy is π.

In [42] it is said that $V^*(s)$ is the *optimal state-value* corresponding to state $s \in \mathcal{S}$, if it returns the smallest achievable expected value when the process starts from state s:

$$V^*(s) \leq V^\pi(s) \tag{4.8}$$

for any policy π. Similarly the *optimal action-value* corresponding to state-action pair (s,a), $Q^*(s,a)$ is defined as the minimum expected regret given that the process starts in state s and the first action is a:

$$Q^*(s,a) \leq Q^\pi(s,a)$$

for any policy π.

4.3.3 Action-value function simplification

The following theorem allows to simplify eq. (4.7). We define $\hat{r}_i : \mathcal{P} \to \mathbb{R}$, for all $i \in \mathcal{I}$, where:

$$\hat{r}_i(p) = \mathbb{E}\left[\rho_i\left(U_{(t)}\right) |P_{(t)} = p\right] .$$

Theorem 4.1. *For any policy π there exists some functions $W_i^\pi : \mathcal{P} \to \mathbb{R}$, s.t. for any $(s,a) \in \mathcal{S} \times \mathcal{A}$,*

$$Q^\pi(s,a) = \sum_{i \in \mathcal{I}} W_i^\pi\left(g(s,a)\right) . \tag{4.9}$$

Moreover, for all $i \in \mathcal{I}$, W_i^π solves the following fixed point equation:

$$W_i^\pi(p) = \hat{r}_i(p) + \gamma \sum_{u \in \mathcal{U}} \mathbb{P}\left[U_{(t)} = u | P_{(t)} = p\right] \cdot \sum_{a \in \mathcal{A}} \pi\left((p,u),a\right) \cdot W_i^\pi\left(g\left((p,u),a\right)\right) . \tag{4.10}$$

Proof. See Appendix 4.1. □

Remark 4.1. *The optimal policy π^* is known to be a deterministic policy i.e. (using a small notation abuse) $\pi^* : \mathcal{S} \to \mathcal{A}$ and we have (see [40]): $\pi^*(s) = arg\ min_a Q^*(s,a)$ ($Q^*$is the* optimal action-value function*).*

Corollary 4.1. *According to Theorem 4.1 and Remark 4.1 the optimal policy can be recovered by:*

$$\begin{aligned} &W_i^*(p) = \hat{r}_i(p) + \gamma \textstyle\sum_{u \in \mathcal{U}} \mathbb{P}\left[U_{(t)} = u | P_{(t)} = p\right] W_i^*(p^*) , \\ &p^* = g\left((p,u),a^*\right) , \\ &a^* = \pi^*(p,u) = \arg\min_{a \in \mathcal{A}} \textstyle\sum_{i \in \mathcal{I}} W_i^*\left(g\left((p,u),a\right)\right) \end{aligned} \tag{4.11}$$

Note that $W_{\mathcal{I}}^*$ has to be processed jointly (all together) as the policy π^* is centralized, i.e. it is a function of all W_i^* components, specifically of $\sum_i W_i^*$.

4.3.4 Linear Function Approximation

Although Theorem 4.1 allows to simplify (4.7) to a set of functions with a reduced domain (*state-space*) $\mathcal{P}$, the complexity still scales exponentially with the number of cells N. Therefore in the sequel we set to perform a *linear function approximation* [40] which trades of the optimal policy due to the loss of information coming with the approximation.

We approximate each $W_i^*(p)$ by some $\bar{W}_i^*(p)$ which depends on p trough a linear function of some features $F_i(p)$. Let J_i be the number of features, i.e. $F_i : \mathcal{P} \to \mathbb{R}^{J_i}$. We provide examples in Section 4.5. Thus $\bar{W}_i^*$ is given by:

$$\bar{W}_i^*(p) = \langle \theta_i^*, F_i(p) \rangle \tag{4.12}$$

where $\langle x, y \rangle$ is the scalar product of x and y, and θ_i^* should minimize the mean square error:

$$\theta_i^* = \text{argmin}_{\theta \in \mathbb{R}^J} \sum_{p \in \mathcal{P}} \left(W_i^*(p) - \langle \theta, F_i(p) \rangle \right)^2 \tag{4.13}$$

This projection allows to reduce the state-space from $I \cdot |\mathcal{P}|$ to $\sum_i J_i$. The choice of the features is very important and it is case specific. We now propose some techniques of how to do this, which we use in our application examples (Section 4.5). For any $i \in \mathcal{I}$, in the case of *no approximation* we can write:

$$F_i(p) = \left(\mathbb{I}_{\{p=x\}} \right)_{x \in \mathcal{P}}$$

where θ_i^* would implicitly be $(W_i^*(p))_{p \in \mathcal{P}}$ and the mean square error in eq. (4.13) is null. As mentioned this does not scale well. The first *linear function approximation* method which we propose to evaluate is based on *State Aggregation* (SA). In this case the feature vector simply becomes:

$$F_i(p) = \left(\mathbb{I}_{\{p_{\mathcal{O}_i} \in x\}} \right)_{x \in \mathcal{P}_{\mathcal{O}_i}}$$

for some set $\mathcal{O}_i \subseteq \mathcal{N} \times \mathcal{K}$. Note that in this case we have a lower (or equal) number of features, i.e. $|\mathcal{O}_i| \leq |\mathcal{N} \times \mathcal{K}|$. This can be used for example if the regret component R_i (and implicitly W_i) depends say only on the update requests coming form one single cell say n ($U^{\mathcal{Z}}_{n,\mathcal{K}}$) and it would be reasonable to approximate that they only depend on the parameters of cell n and its neighbours ($P_{\bar{\mathcal{N}}_n,\mathcal{K}}$), i.e. $\mathcal{O}_i = \bar{\mathcal{N}}_n \times \mathcal{K}$. Thus the size of F_i is limited no matter how big the network is, and so the size of F scales linearly with the number of cells. This can be seen as neglecting the dependency of W_i on some non-essential parameters based on geographic considerations, i.e. the parameters of the cells which are far away from cell n.

The second *linear function approximation* which we propose to evaluate is based on the use of *Linear Features* (LF) where in the general sense we can write:

$$F_i(p) = \sum_{n \in \mathcal{N} \times \mathcal{K}} \omega^i_{n,k} p_{n,k}$$

for some $\omega^i_{n,k} \in \mathbb{R}^{J_i}$. As in the *state aggregation* case, by properly choosing the coefficients $\omega^i_{n,k}$ (specifically by making some of them null) we can get $F_i(p)$ to depend only on a subset of the parameters ($\mathcal{O}_i \subseteq \mathcal{N} \times \mathcal{K}$):

$$F_i(p) = \sum_{(n,k) \in \mathcal{O}_i} \omega^i_{n,k} p_{n,k}.$$

Note that this not trivial. Depending on the function ρ_i which gives the regret component R_i a linear approximation of W_i^* may or may not be suitable. This is case specific and is done based on expert knowledge of engineers with network optimization experience. In our application examples (Section 4.5) we chose the regret components functions ρ_i as simple as possible so that it is reasonable to approximate W_i^* making use of *linear features*.

Under these conditions we have to provide a method to estimate θ_i^* and implicitly the approximation of the optimal policy. We provide this in the next section.

Algorithme 4.1 SONCO-R

Function Init :

Initialize, for all $i \in \mathcal{I}$, $\theta_i \leftarrow 0^{J_i}$

Function SONCO-R :

Observe current parameter configurations p and update requests u,

Calculate regret $r_i = \rho_i(u)$, $\forall i \in \mathcal{I}$,

Compute $\bar{a} = \arg\ \min_{a \in \mathcal{A}} \sum_{i \in \mathcal{I}} \langle \theta_i, F_i(g((p,u),a)) \rangle$

Update $\theta_i \leftarrow \theta_i + \alpha \left[r_i + \gamma \langle \theta_i, F_i(g((p,u),\bar{a})) \rangle - \langle \theta_i, F_i(p) \rangle \right] F_i(p)$, $\forall i \in \mathcal{I}$

Choose current action a using the ϵ-greedy policy:

$\pi((p,u),\{a\}) = (1-\epsilon)\,\mathbb{I}_{\{a=\bar{a}\}} + \frac{\epsilon}{3^{NK}}$

Take action a.

4.4 Reinforcement learning

4.4.1 Algorithm

We cannot directly calculate $\bar{W}_i^*$ as we only have partial knowledge on the transition kernel. Instead, (according to [40]) we can use the following recursion: at every time instant $t \in \mathbb{N}$, given the state $S_{(t)} = (p,u)$ we update θ_i ($\forall i \in \mathcal{I}$) as follows:

$$\begin{aligned} &\theta_i \leftarrow \theta_i + \alpha \left[\rho_i(u) + \gamma \langle \theta_i, F_i(g((p,u),\bar{a})) \rangle - \langle \theta_i, F_i(p) \rangle \right] F_i(p) \\ &\bar{a} = \arg\min_{a \in \mathcal{A}} \sum_{i \in \mathcal{I}} \langle \theta_i, F_i(g((p,u),a)) \rangle \end{aligned} \tag{4.14}$$

Consequently it would be natural to define the action at time t as $\bar{A}_{(t)} = \bar{a}$.

In practice, the construction of the feature function F (as suggested in Section 4.3.4) is insensitive (independent) to certain components of the parameter p and their associated actions. Denote by $\mathcal{O} = \bigcup_{i \in \mathcal{I}} \mathcal{O}_i \subseteq \mathcal{N} \times \mathcal{K}$ the set of (n,k) pairs where F is dependent on $p_{n,k}$. This means that we may have several potential outputs of the *arg min* in eq. (4.14). One option is to choose randomly one of them. However we choose to select the one which leads to accepting the biggest number of change requests.

In order to ensure that the entire state space is explored we choose and ϵ-greedy policy (see [40]) w.r.t. $\bar{a}$ as follows:

$$\pi((p,u),\{a\}) = (1-\epsilon)\,\mathbb{I}_{\{a=\bar{a}\}} + \frac{\epsilon}{3^{NK}} \tag{4.15}$$

This means that during $100 \cdot \epsilon\%$ of the time we *explore* the state space and during $100 \cdot (1-\epsilon)\%$ of the time we exploit the learned information.

The proposed algorithm for a SONCO-R instance is summarized in Alg. 4.1. **Function Init** should be called for the initialization of the algorithm and **Function SONCO-R** should be called every time after receiving the requests of the SON instances. Note that we cannot expect the convergence of the algorithm as we are using a fixed step size α. The use of a fixed step size is to envisage for a non-stationary environment, i.e. the optimal policy may change during time. At most we can expect to have a convergence in the average. This cannot be guaranteed for any function approximation. However this will be the case in our scenario according to the results in Section 4.5. For this, the features F have been carefully chosen such that they allow for a decent approximation of W.

The following proposition focuses our effort on improving the scalability the particular case of using *linear features*. It provides a distributed way to calculate $\bar{a}_{n,k}$ in eq. (**??**), thus making the computational effort per SONCO-R instance scale linearly with the number of governed cells.

Proposition 4.1. *If $\forall i \in \mathcal{I}$, the feature $F_i(p)$ in eq. (4.12) is a linear function w.r.t. the components of p, i.e. :*

$$F_i(p) = \sum_{n\in\mathcal{N}} \sum_{k\in\mathcal{K}} \omega^i_{n,k} p_{n,k}$$

where $\omega^i_{n,k} \in \mathbb{R}^{J_i}$, then each component of $\bar{a}_{n,k}$ in eq. (??)/ Algorithm 4.1 can be computed as:

$$\forall (n,k) \in \mathcal{N}\times\mathcal{K},\ \bar{a}_{n,k} = \arg\min_{a_{n,k}\in\mathcal{A}_{n,k}} g_{n,k}\left(\left(p_{n,k}, u^{\mathcal{Z}}_{n,k}\right), a_{n,k}\right)\Omega_{n,k} \tag{4.16}$$

where $\Omega_{n,k} = \sum_{i\in\mathcal{I}} \left\langle \theta_i, \omega^i_{n,k} \right\rangle$.

Proof. See Appendix 4.2 □

4.4.2 Multi-dimension regret

Say that a network operator wants to diminish the most *critical* update request. In this case it would be suitable to consider the regret function ρ as:

$$\rho(u) = \max_{n,k,z} \left|u^z_{n,k}\right|$$

Thus in this case it is clear that we cannot write the regret as a summation of sub-regrets and it is difficult to find a suitable approximation for this regret function. In this section, instead of assuming that the regret is a summation of I sub-components, we assume that the regret is actually a vector of D sub-components. Thus for any given time instance we can write the instantaneous regret in Algorithm 4.1 as a vector:

$$r = (r_d)_{d\in\mathcal{D}} \in \mathbb{R}^D$$

($\mathcal{D} = \{1,..,D\}$). We still consider that the instantaneous regret (r) is a function of the update request (u): $r = \rho(u)$ and $r_d = \rho_d(u)$, where u is the current update request.

For $D = 1$ obviously we fall in to the above framework. For $D > 1$ the framework is still valid up to a certain point. The extension of the definitions of the value functions V in (4.6) and Q in (4.7) is straight forward, they simply become vectors of size D. The first bottleneck is the optimal policy in Remark 4.1 as the *arg min* has no longer a sense for multi-dimension Q-function. To solve this problem we first define a methodology to compare vectors.

Consider two vectors of size D: $x, y \in \mathbb{R}^D$. Let $x', y' \in \mathbb{R}^D$ be x and y sorted in descending order, respectively, i.e. $x'_1 > x'_2 > ....$ If $x'_1 > y'_1$ we say that x is larger than y (we write $x \succ y$). If $x'_1 < y'_1$ we say that x is smaller than y (we write $x \prec y$). If $x'_1 = y'_1$ then we move to comparing x'_2 and y'_2 in the same way, and so on. If $x'_d = y'_d$ for all $d \in \mathcal{I}$, then we say that the two vectors are *equal* (we write $x \asymp y$). Note that $x \asymp y$ is not the same thing as $x = y$. For $D = 1$ this is coherent with the scalar comparison rules.

By extending the previous framework for any value of D, we say that a policy π^* is optimal if it achieves the optimal value function ($V^*(s)$) in all states ($\forall s \in \mathcal{S}$), i.e. for any policy π we have:

$$V^*(s) \preceq V^\pi(s). \tag{4.17}$$

To simplify things for $D > 1$ we only consider $\gamma = 0$, i.e. myopic policies. It is straight foreword to see that the value function of any policy including the optimal one is given by:

$$Q^*(s,a) = Q^\pi(s,a) = \bar{r}(s,a)$$

Using the same reasoning as in the uni-dimensional case we end up with estimating $W^*(p)$ which is a vector of size D. Thus we approximate each component $W^*_d(p)$ using a set of J_d features $F_d(p) : \mathcal{P} \to \mathbb{R}^{J_d}$, i.e. $\bar{W}^*_d(p) = \langle \theta^*_d, F_d(p) \rangle$. The proposed algorithm for a SONCO-R instance is summarized in Alg. 4.2.

If $D > 1$ the reasoning does not hold for $\gamma > 0$ due to the non-linearity of min/max operations in Remark 4.1.

Algorithme 4.2 SONCO-R for multi-dimension regret

Function Init :

Initialize, for all $d \in \mathcal{D}$, $\theta_d \leftarrow 0^{J_d}$

Function SONCO-R :

Observe current parameter configurations p and update requests u,

Calculate regret $r_d = \rho_d(u)$, $\forall d \in \mathcal{D}$,

Compute $\bar{a} = \arg\ \min_{a \in \mathcal{A}} \langle \theta_d, F_d(g((p,u),a)) \rangle_{d \in \mathcal{D}}$

Update $\theta_d \leftarrow \theta_d + \alpha \left[r_d + \gamma \langle \theta_d, F_d(g((p,u),\bar{a})) \rangle_{d \in \mathcal{D}} - \langle \theta_d, F_d(p) \rangle_{d \in \mathcal{D}} \right] F_d(p)$, $\forall d \in \mathcal{D}$

Choose current action a using the ϵ-greedy policy:

$\pi((p,u),\{a\}) = (1-\epsilon)\,\mathbb{I}_{\{a=\bar{a}\}} + \frac{\epsilon}{3^{NK}}$

Take action a.

4.4.3 Complexity analysis

In this section we analyse the memory requirements and the computational complexity of the framework.

Memory requirements The main problems in terms of memory come from the size of θ, which has the same size as the feature vector F. For one cell cluster, i.e. one SONCO-R instance, the size of F is given by:

$$complexity:\ F-size = \sum_{i \in \mathcal{I}} |\mathcal{J}_i| \tag{4.18}$$

Computational complexity The number of operations that the algorithm is supposed to perform may also represent a bottleneck. Thus we outline the main components that require most of the computational effort:

- update of θ: the number of parameter (θ) that have to be updated (worst case scenario)

$$complexity:\ \theta-update\ (general) = \sum_{i \in \mathcal{I}} |\mathcal{J}_i| \tag{4.19}$$

  Note that in some cases this may be simplified. As we are not using eligibility traces ([40]), when the function approximation is equivalent to a state aggregation (including the no-approximation cases), we actually have to update one θ per SONCO-R instance, i.e. just the one corresponding to the current state:

$$complexity:\ \theta-update\ (SA) = 1 \tag{4.20}$$

- computing $\bar{a}$: basically the problem is that we have to calculate the argmin in (4.11). In the worst case scenario this comes to exhaustively evaluating all possible actions:

$$complexity:\ a-search\ (general) = |\mathcal{A}| = 3^{|\mathcal{O}|} \tag{4.21}$$

  where we recall that $\mathcal{O} \subseteq \mathcal{N} \times \mathcal{K}$ the set of (n,k) pairs where F is dependent on $p_{n,k}$. Note that in some cases Prop. 4.16 allows to simplify eq. (4.21) to:

$$complexity:\ a-search\ (LF\ scalar) = 3\,|\mathcal{O}| \tag{4.22}$$

4.5 Application Scenarios

To validate our framework we analyse several scenarios. At first we use the uni-dimensional/scalar regret and we focus on two cases corresponding to the conflict types: measurement conflicts and parameter conflicts. For each of the two we provide evaluations for when the regret can or cannot be written as a summation of several sub-components. These evaluations are done in a homogeneous network which contains only Macro cells.

Next we use the multi-dimension regret focusing on parameter conflict resolution. These evaluations are done in a Heterogeneous Network (HetNet) which contains Macro cells and Pico cells.

When the regret can be broken into a summation of sub-regrets or into a vector of sub-regrets and we deal with several components of W, we say that the *Learner is Distributed* (LD). Otherwise, if the regret cannot be broken into sub-components then we only have one component of W and we say that the *Learner is Centralized* (LC).

To have an overview of the scenarios we provide a summary in Table 4.1, while the details are presented in the following sub-sections.

Regret	Conflict type	Learner	SON functions	Section
Scalar	measurement	LC	MLB	4.5.1
		LD		
	parameter	LC	MLB, MRO	4.5.2
		LD		
Vector	parameter	LC	(MLB via) CRE), MRO, eICIC	4.5.3
		LD		

Table 4.1: Application scenarios

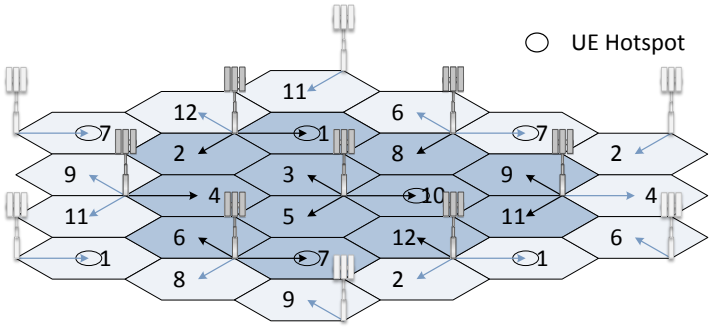
4.5.1 Scalar regret: measurement conflict resolution

In this section we tackle the measurement conflict resolution for a scalar regret. We focus on the instance dimension of the SONCO-R (see Section 2.2). We make use of one SON function, namely MLB, independently instantiated on each and every cell (see Appendix B). Briefly, an MLB instance tunes the CIO of the hosting cell in order to maintain its load under a predefined threshold ($\mathbb{T}^H_{LD}$). We note that the MLB instance running on one cell impacts the MLB instance on a neighbouring cell. This is a measurement conflict which may affect the *network stability*, i.e. we experience unnecessary parameter changes. Improving the network stability (i.e. eliminating unnecessary parameter changes) could be done by forbidding any parameter changes, but this would prevent SON functions from doing their job. Thus, the purpose of the SONCO-R instance is to eliminate this instability by finding the network configuration with the smallest *regret*, to steer the network towards this configuration and to prevent as much as possible diverting from it. For this we make two evaluations. First, we consider that the regret is a scalar value which cannot be written as a sum of sub-regrets. Second, we consider that the regret is distributable and can be expressed as a sum of sub-regrets (all scalars), e.g. one per cell.

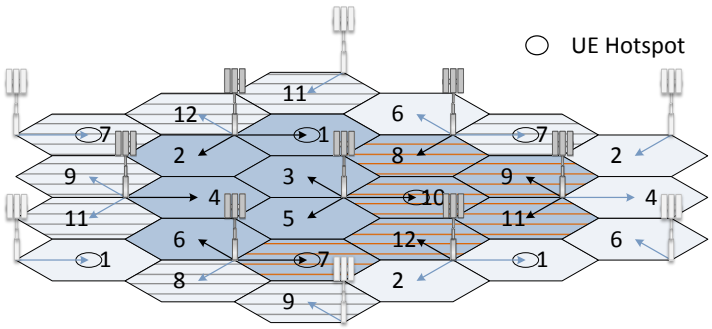
Non-distributable regret

In this sub-section we consider that the regret is a simple scalar, more specific it can not be written as a summation of several sub-regrets ($\mathcal{I} = \{1\}$). We choose a regret which penalizes having requests to change the network parameter configuration, especially the *off-load* requests (i.e. the decrease CIO requests) as they represent that the cells are overloaded (i.e. their load is above a threshold). The regret is minimum in the case of a maintain requests as it reflects that the MLB instances are satisfied with the current network parameter configuration. More precisely we choose the following regret function:

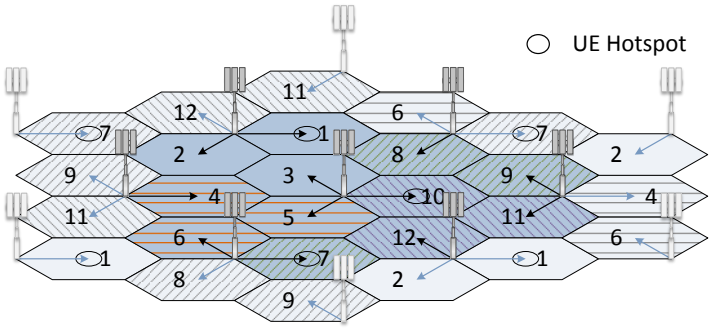
$$\rho_i(u) = \max_{n \in \mathcal{N}} \left(-\mathbb{I}_{\{u^{MLB}_{n,CIO}=0\}} - 9\mathbb{I}_{\{u^{MLB}_{n,CIO}\geq 0\}} \right)$$



(a) One cluster



(b) Two clusters



(c) Four clusters

Figure 4.4: Network topology

for all $i \in \mathcal{I}$.

The simplest SONCO-R design is the one where no approximation is performed. In the *optimal* case there is only one cell cluster containing all the cells, i.e. all cells are governed by one SONCO-R instance. The feature vector can be expressed as:

$$F_i(p) = \left(\mathbb{I}_{\{p_{\mathcal{O}_i}=x\}}\right)_{x \in \mathcal{P}_{\mathcal{O}_i}}$$

for any $p \in \mathcal{P}$ and for all $i \in \mathcal{I}$ where $\mathcal{O}_i = \mathcal{N} \times \mathcal{K}$. The size of F_i is $|\mathcal{P}_{\mathcal{O}_i}|$ which scales exponentially with the number of cells. So we say that the solution is not scalable. To make it scalable one easy approach would be to divide the area of interest into several sub-areas, each of which represents a cell cluster and is governed by an independent SONCO-R instance. Obviously this is a *sub-optimal* solution as it ignores the fact that the cells from one cluster impact the performance of the cells from other clusters. Thus in this section we set out to evaluate precisely this *sub-optimality*. We examine 5 scenarios coded as follows:

SFoff: MLB is off,

SCoff: MLB is on, SONCO-R is off,

SC-LC: MLB and SONCO-R are on, all cells are part of 1 cell cluster, see Fig. 4.4a.

SC6-LC: MLB and SONCO-R are on, 6 cells per cluster, see Fig. 4.4b.

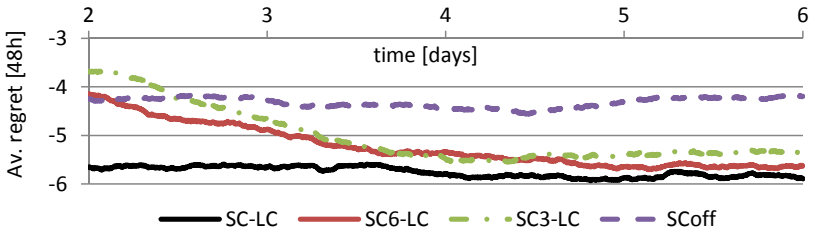
SC3-LC: MLB and SONCO-R are on, 3 cells per cluster, see Fig. 4.4c.

The rest of the details can be found in Table 4.2.

One can see in Fig. 4.5a that the average regret is decreased when the SONCO-R is on, with the biggest gain being (as expected) in the centralized SONCO-R deployment (SC-LC). We can see that the more distributed the system is the the larger the regret. This is the price to pay in order to gain in scalability. It is easy to foresee that if the cell clusters of the SONCO-R instances in the distributed cases were independent then the performance would have been the same as for the centralized case. In practice the cell clusters should be established such that between them there is as less interaction as possible, for example if the border cells have almost static parameter configuration, but good results can also be obtained if this is not possible.

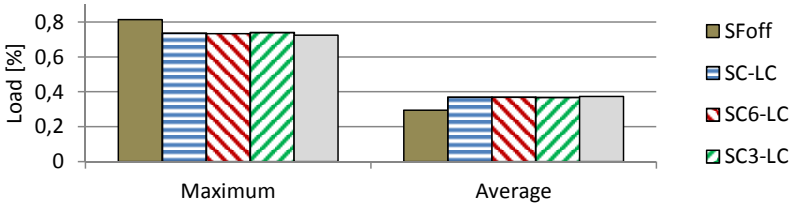
Since our aim is to reduce the number of CIO changes, we plot in Fig. 4.5b the maximum and the average (over all cells) of the time-averaged number of CIO changes. We can see a significant decreased (73-84%) eliminating most of the unnecessary configuration changes and the accompanying signalling. Also, by steering the CIOs configuration towards the one with the lowest regret, the frequency of overload events (when cells send off-load requests) is reduced by 30-46%.

In Fig. 4.5c we plot the maximum and the average (over all cells) of the time-averaged loads; one can see that the MLB instances are still fulfilling their task keeping the load well below the *off-load* threshold ($\mathbb{T}^H_{LD} = 0.8$). Note that the differences between all the scenarios with *MLB on* are very small.



(a) Average regret (48h sliding window)

(b) Average number of CIO changes [#/h] over last 48h



(c) Average loads over last 48h

Figure 4.5: Results: scalar non-distributable regret (for measurement conflict resolution)

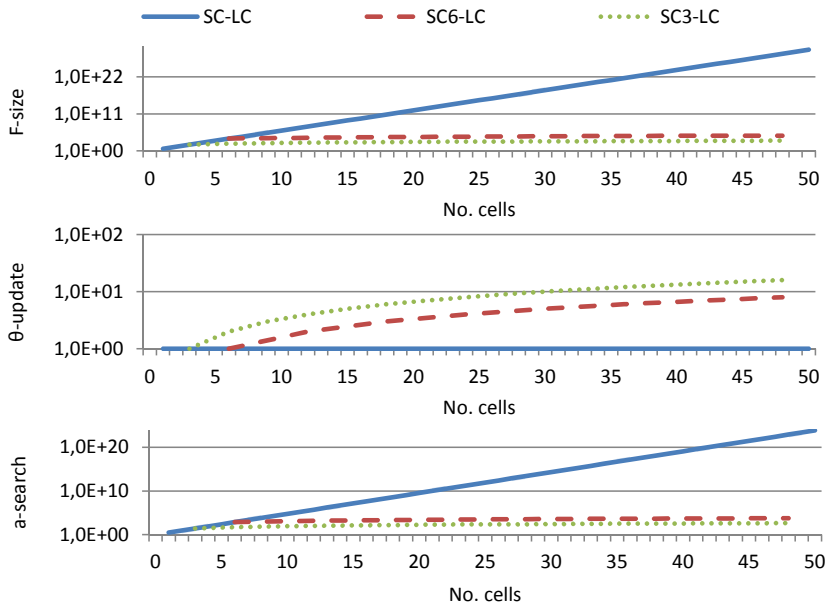


Figure 4.6: Complexity: scalar non-distributable regret (for measurement conflict resolution)

We plot in Figure 4.6 the complexity of F-size, θ-update and a-search, w.r.t. the number of cells. One can see that for the centralized design (SC-LC) although the complexity of the θ-update is constant, the complexities of F-size and a-search are exponential. In the case where we form smaller cluster of 3 (SC3-LC) respectively 6 (SC6-LC) cells the scalability of the feature vector and of the a-search become linearly at the cost of having the complexity of the θ-update slightly increased.

Distributable regret

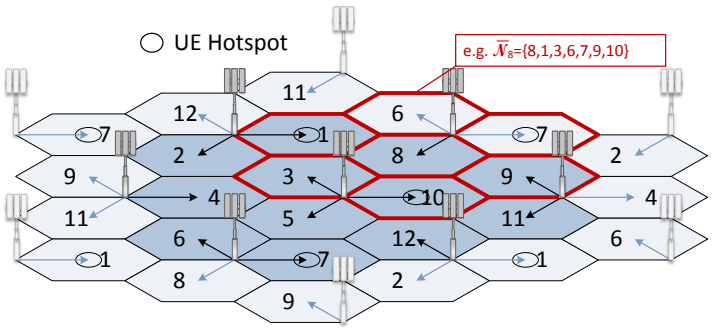


Figure 4.7: Network topology - local learning

In this section we evaluate the benefits of writing the regret as a sum of several sub-regrets. Specifically we consider one sub-regret per cell ($\mathcal{I} = \mathcal{N}$) depending only on the update requests of that particular cell. Thus, here also, we choose a regret which penalizes having requests to change the network parameter configuration, especially the *off-load* requests (i.e. the decrease CIO requests):

$$\rho_i(u) = -\mathbb{I}_{\{u_{i,CIO}^{MLB}=0\}} - 9\mathbb{I}_{\{u_{i,CIO}^{MLB}\geq 0\}}$$

for any $i \in \mathcal{I}$. Without any approximation, W_i depends on all the network parameters $P_{\mathcal{N},\mathcal{K}}$, i.e. the feature vector is given by

$$F_i(p) = \left(\mathbb{I}_{\{p_{\mathcal{O}_i}=x\}}\right)_{x\in\mathcal{P}_{\mathcal{O}_i}}$$

for any $i \in \mathcal{I}$ where $\mathcal{O}_i = \mathcal{N} \times \mathcal{K}$. We say this solution is *optimal* (i.e. no approximation), but it does not scale well. The size of each the feature vector F_i is $|\mathcal{P}_{\mathcal{N},\mathcal{K}}|$, which means that it scales exponentially with the number of cells. To solve this problem we consider that W_i ($\forall i \in \mathcal{N}$) only depends on the parameters of cell i and its neighbours ($P_{\bar{\mathcal{N}}_i,\mathcal{K}}$), see Fig. 4.7. To this end we apply a *state aggregation* simply by considering $\mathcal{O}_i = \bar{\mathcal{N}}_i \times \mathcal{K}$, i.e. we account for any configuration of the CIO of cell i and of its neighbours. Thus the size of the feature vector F_i becomes $\left|\mathcal{P}_{\bar{\mathcal{N}}_i,\mathcal{K}}\right|$. This solutions is *sub-optimal*, but it clearly has its benefits: the size of the vector F containing all features scales linearly with the number of cells.

We examine 4 scenarios coded as follows: :

SFoff: MLB is off.

SCoff: MLB is on, SONCO-R is off.

SC-LD: MLB and SONCO-R are on, 1 cluster with all the cells, we use the *optimal* solution (no approximation).

SC-LD-SA: MLB and SONCO-R are on, 1 cluster with all the cells, we use the *sub-optimal* solution.

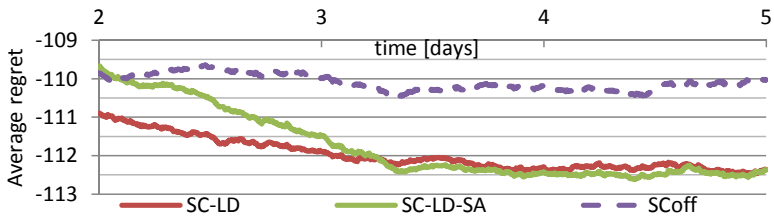
The rest of the details can be found in in Table 4.2.

Fig. 4.8a shows that the average regret is decreased when the SONCO-R is on. Moreover the sub-optimal policy (SC-LD) converges to the same regret as the optimal policy (SC-LD-SA). One can see that the convergence of the algorithm remains slow. This is due to the fact that the algorithm learns from scratch and needs to visit all the possible states a sufficient number of times. Once the learning algorithm is trained offline on a realistic network simulator, it can be implemented in the real network. The algorithm should then be capable of tracking and adapting dynamically to the traffic variations. We tackle this in a future section.

As mentioned our aim is to reduce the number of parameter changes. In Fig. 4.8b we plot the maximum and the average (over all cells) of the time-averaged number of CIO changes. One can see that he proposed approach has eliminated a big amount of the unnecessary configuration changes (77-82%) and implicitly the accompanying signalling. By steering the network towards the most rewarding CIO configuration we also reduce the frequency of overload events (concretely the number of off-load requests) by 33-40%.

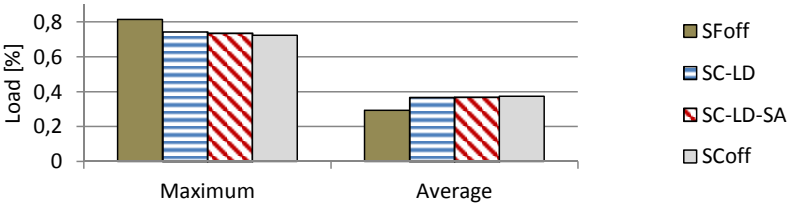
In Fig. 4.8c the maximum and the average (over all cells) of the time-averaged loads are plotted. We can see that for the SONCO-R-on cases (SC-LD, SC-LD-SA) the MLB instances still accomplish their task (keeping the load below $\mathbb{T}^H_{LD} = 0.8$) meaning that the SONCO-R does not prevent the MLBs from balancing the cell loads.

We plot in Figure 4.9 the complexity of F-size, θ-update and a-search, w.r.t. the number of cells. One can see that for the optimal design (SC-LD) the F-size complexity scales exponentially, where as in the case using the state aggregation (SC-LD-SA) it scales linearly. The θ-update complexity scales linearly in both cases. The a-search complexity scales exponentially (worst case scenario).



(a) Average regret (48h sliding window)

(b) Average number of CIO changes [#/h] over last 48h



(c) Average loads over last 48h

Figure 4.8: Results: scalar distributable regret (for measurement conflict resolution)

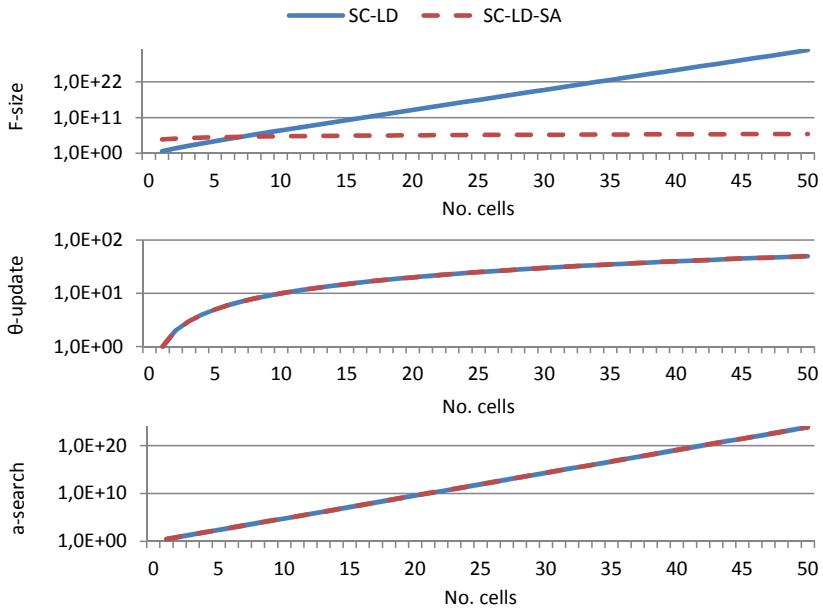


Figure 4.9: Complexity: scalar distributable regret (for measurement conflict resolution)

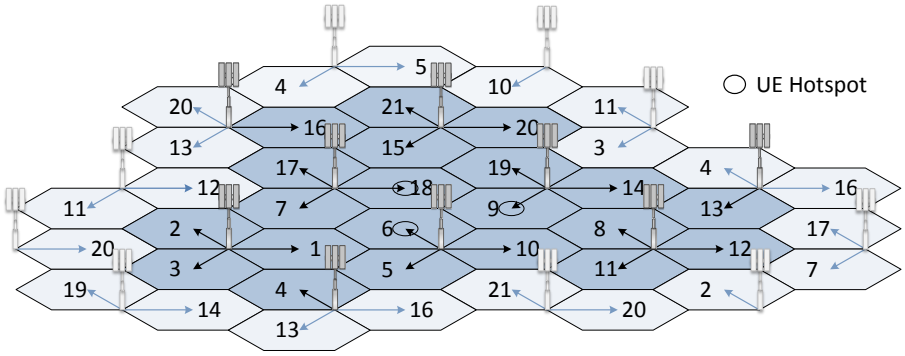
Scenario:	SC-LC SC6-LC SC3-LC	SC-LD	SC-LD-SA
No. cells	12 Macro cells - hexagonal topology, with wraparound		
Clusters	1 vs 2 vs 4	1	
ISD	500 m		
Carrier/Bw/TxPow	Carrier = 2 GHz, Bandwidth = 10 MHz, CellTxPower = 46 dBm		
Propagation Model	Channel model #1 in Table A.1, (3GPP Case 1 [54])		
Traffic type	FTP-like , constant file size $FS = 16[Mb/UE]$, UE arrival: SPPP, arrival rates: general over-all (λ_G, per hexagon) and HotSpot (λ_{HS}, per HS)		
$(\lambda_G; \lambda_{HS})\, FS[Mb/s]$	$(6; 21)$		
Simulated time	6 days	5 days	
SON functions	$\mathcal{Z} = \{MLB\}$ (for a Macro network, see Appendix B)		
Tuned parameters	$\mathcal{K} = \{CIO\}$		
CIO set [dB] $\mathcal{P}_{n,CIO}$	$\{-9, -6, -3, 0\}$		
HYS set [dB] $\mathcal{P}_{n,HYS}$	$\{1\}$ (fixed)		
ABS set [#] $\mathcal{P}_{n,ABS}$	$\{0\}$ (fixed)		
TTT	160ms		
Time window T	2.5 min		
SON settings	$\left(\mathbb{T}^L_{LD}; \mathbb{T}^H_{LD}\right) = (0.5; 0.8)$		
SON priority weights	-		
Conflict-solving target	measurement conflict resolution (network stability)		
Criticalness indicator	not used, i.e. we only use the sign of the update request		
state reduction	no state reduction	no state reduction	state aggregation
Distr. Learner (W)	no (centralized)	per cell	
SONCO-R $(\alpha, \gamma, \epsilon)$	$(0.05, 0.8, 0.1)$		
No. Learners $I =$	$I = 1$	$I = \lvert\mathcal{N}\rvert$	
b is a bijection	-		
Learners focus $\forall i \in \mathcal{I}$, $\mathcal{O}_i =$	$\mathcal{N} \times \mathcal{K}$	$\{\mathcal{N} \times \mathcal{K}\}$	$\{\bar{\mathcal{N}}_i \times \mathcal{K}\}$
$\forall i \in \mathcal{I}$,$\mathcal{J}_i =$	$\{1, .., \lvert\mathcal{P}_{\mathcal{O}_i}\rvert\}$		
$\forall p \in \mathcal{P}$, $\forall i \in \mathcal{I}$, $F_i(p) =$	$\left(\mathbb{I}_{\{p_{\mathcal{O}_i} = x\}}\right)_{x \in \mathcal{P}_{\mathcal{O}_i}}$		
size of F	$\sum_{i \in \mathcal{I}} \lvert\mathcal{P}_{\mathcal{O}_i}\rvert$		
$\forall u \in \mathcal{U}$, $\forall i \in \mathcal{I}$, $\rho_i(u) =$	$\max_{n \in \mathcal{N}} \left(-\mathbb{I}_{\{u^{MLB}_{n,CIO} = 0\}} - 9\mathbb{I}_{\{u^{MLB}_{n,CIO} \geq 0\}}\right)$	$-\mathbb{I}_{\{u^{MLB}_{i,CIO} = 0\}} - 9\mathbb{I}_{\{u^{MLB}_{i,CIO} \geq 0\}}$	
multi-dimension frame	no		

Table 4.2: Scenario summaries: scalar regret for measurement conflict resolution

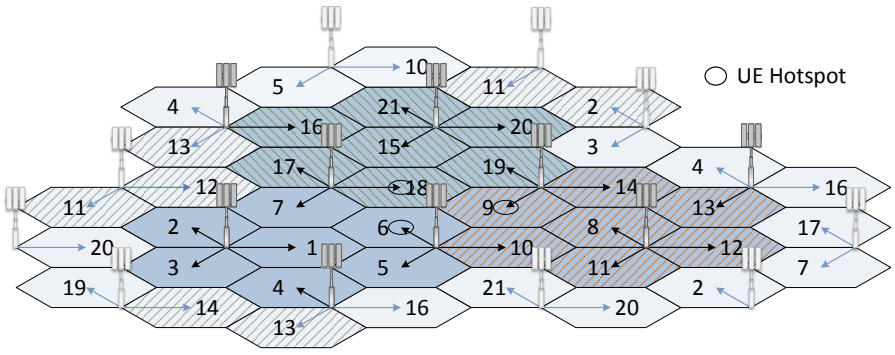
4.5.2 Scalar regret: parameter conflict resolution

In this section we tackle the parameter conflict resolution for a scalar regret. We focus on the function dimension of the SONCO-R (see Section 2.2). We use 2 SON functions: namely MLB and MRO. An MLB instance tunes the CIO of the hosting cell in order to maintain its load under a predefined threshold ($\mathbb{T}^H_{LD}$). An MRO instance tunes the CIO and the HYS of the hosting cell in order to reduce the number of ping pong HOs and the number of too late HOs originating from it. For more details on how the two SON functions work see Appendix B. We remark that both MLB and MRO tune the CIO. Thus we aim for the SONCO-R to perform a parameter conflict resolution ensuring a certain degree of fairness between the two SON function. As in the previous section we tackle potential solutions of reducing the size of the feature vector (F). First, we consider that the regret is a scalar value which cannot be written as a sum of sub-regrets. Second, we consider that the regret is distributable and can be expressed as a sum of sub-regrets (all scalars), e.g. one per cell or one per update-request.

Non-distributable regret



(a) One cluster



(b) Three clusters

Figure 4.10: Network topology

In thus sub-section we assume that the regret is non-distributable, i.e. it cannot be written as a sum of sub-regrets. As we focus on the parameter conflict on the CIO we define a regret function:

$$\rho_i(u) = \max_{(n,z)} \left| u^z_{n,CIO} \right| w_z$$

for any $i \in \mathcal{I} = \{1\}$, i.e. it reflects the most *critical* request targeting the CIO, where w_z is a weight attributed to the SON function z. Next we reduce the number of feature by performing a *state aggregation* as follows:

$$F_i(p) = \left(\mathbb{I}_{\{p_{\mathcal{O}_i} = x\}} \right)_{x \in \mathcal{P}_{\mathcal{O}_i}}$$

for any $i \in \mathcal{I} = \{1\}$ where $\mathcal{O}_i = \mathcal{N} \times \{CIO\}$, i.e. the impact of the HYS is neglected from the conflict perspective due to its low impact on the inputs of the MLB. This still leaves us with

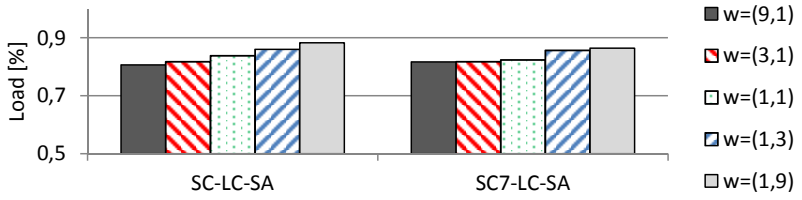
a big number of features, specifically $\left|\mathcal{P}_{\mathcal{N},\{CIO\}}\right|$, which scales exponentially with the number of cells. Therefore, we analyse the solution to divide the area of interest into sub-areas each of which represents a cell cluster and so it is governed by one independent SONCO-R instance. We examine 2 scenarios coded as follows: :

SC-LC-SA: SONs and SONCO-R are on, all cells part of 1 cluster, see Fig. 4.10a.

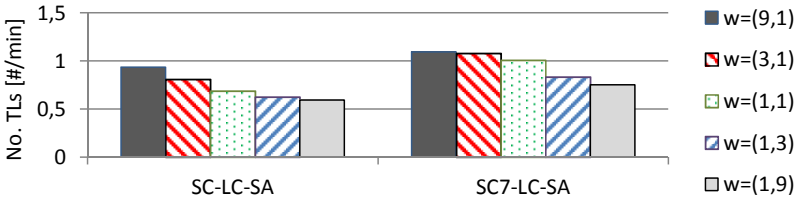
SC7-LC-SA: SONs and SONCO-R are on, 7 cells per cluster, see Fig 4.10b.

The rest of the details can be found in in Table 4.3.

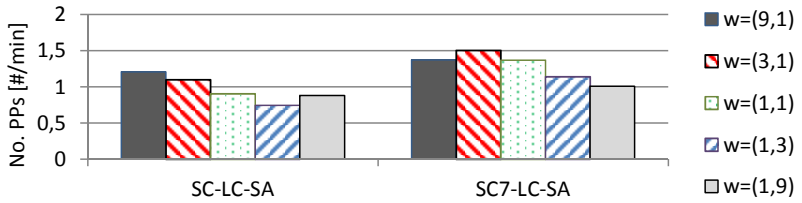
In Figures 4.11a, 4.11c and 4.11b we plot the maximum (over all cells) of the time-averaged cell load, the time-averaged number of too late HOs and the time-averaged number of HO ping-pongs, respectively, for both centralized and distributed scenarios. Note that w is the weights vector. The time-average is done over the last half of the simulation time.



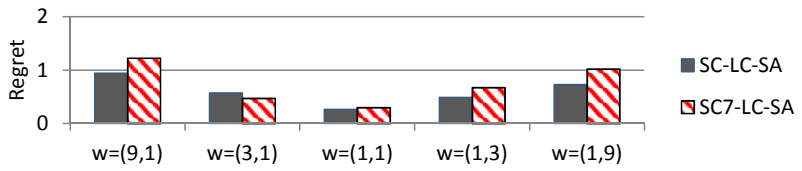
(a) Maximum Average Load



(b) Maximum Average Number of Too Late HOs



(c) Maximum Average Number of Ping-Pongs



(d) Average regret (last $24h$)

Figure 4.11: Results: scalar non-distributable regret (for parameter conflict resolution)

In Fig. 4.11a one can see that a better load balancing is achieved by giving higher weights to MLB. On the other hand giving a bigger priority to MRO causes the maximum load to degrade by up to 9.5% for SC-LC-SA and 5.7% for SC7-LC-SA, but the maximum number of too late HOs is reduced by up to 36.5% for SC-LC-SA and 31.2% for SC7-LC-SA (Fig. 4.11c). Also, the number of ping-pongs is reduced by up to 38.2% in both scenarios (Fig. 4.11b).

In fig 4.11d one can see that the centralized SONCO-R is most-likely to reach a smaller regret. However, a centralized solution is not always feasible as the state space scales exponentially with the number of cells.

We plot in Figure 4.12 the complexity of F-size, θ-update and a-search, w.r.t. the number of cells. As in Section 4.5.1 the scenario with one cluster (SC-LC-SA) does not scale well for the complexity of the F-size and a-search. The scenario where we have 7 cells per cluster (SC7-LC-SA) solves this issue making these complexities scale linearly with the number of cells.

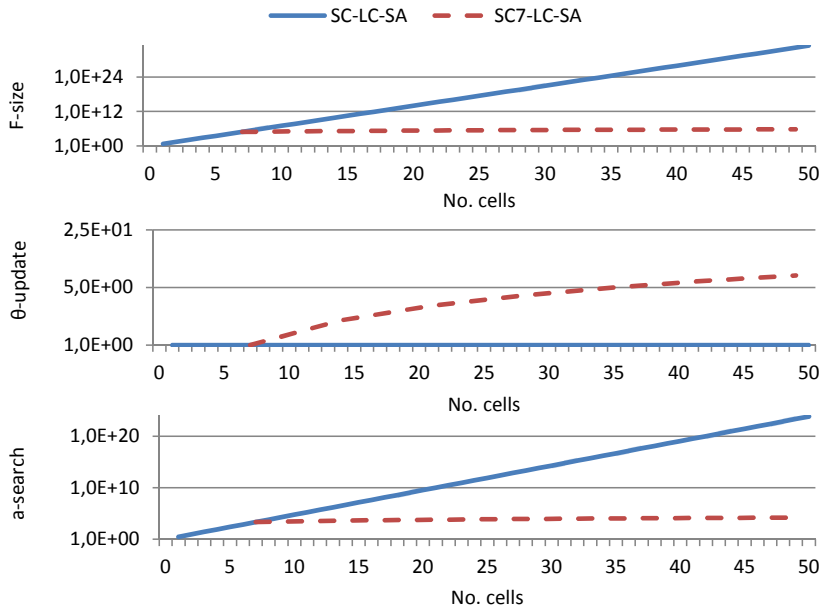


Figure 4.12: Complexity: scalar distributable regret (for parameter conflict resolution)

Distributable regret

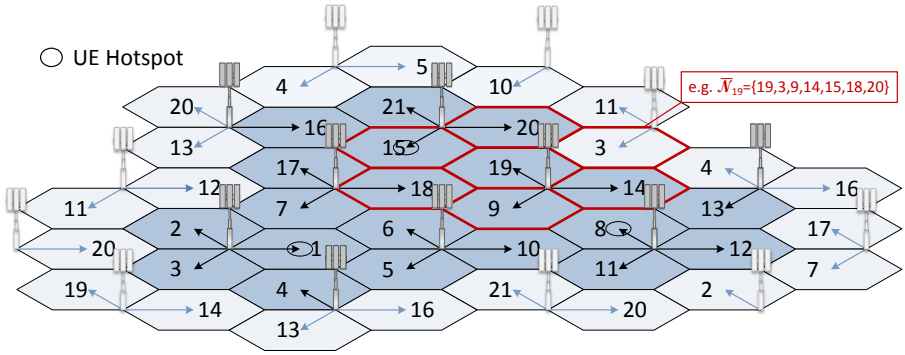


Figure 4.13: Network topology

As compared to the scenarios in the previous sub-section we now evaluate the parameter conflict resolution in the case where the regret can be expressed as a summation of several sub-components.

For this we consider two cases: In the first case we consider one regret component per cell, i.e. $\mathcal{I} = \mathcal{N}$. Each regret-component R_i depends only on the update requests of cell i ($U^{\mathcal{Z}}_{i,\{CIO\}}$). The regret function, is given by ($\forall i \in \mathcal{I}$, $\forall u \in \mathcal{U}$):

$$\rho_i(u) = \max_{z \in \mathcal{Z}} \left| u^z_{i,CIO} \right| w_z$$

where w_z represents the weight attributed to the SON function z. This reflects the most critical request targeting the CIO w.r.t. the two SON functions. To make the solution scalable we consider that W_i $(\forall i \in \mathcal{N})$ only depends on the CIO parameters of cell i and of its neighbours $(P_{\bar{\mathcal{N}}_i,\{CIO\}})$, see Fig. 4.13. We perform a *state aggregation* after which the feature vector comes to:

$$F_i(p) = \left(\mathbb{I}_{\{p_{\mathcal{O}_i}=x\}}\right)_{x\in\mathcal{P}_{\mathcal{O}_i}}$$

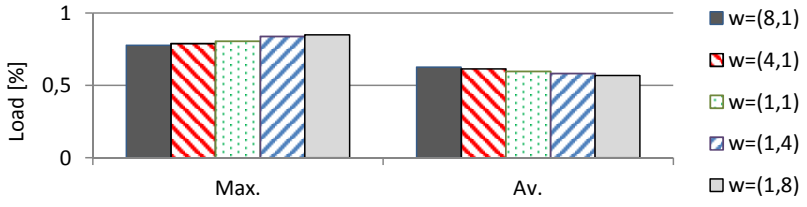
where $\mathcal{O}_i = \bar{\mathcal{N}}_i \times \{CIO\}$. This means that the size of F_i becomes $\left|\mathcal{P}_{\bar{\mathcal{N}}_i,\{CIO\}}\right|$, i.e. accounting for any configuration of the CIO of cell i and its neighbours. The solution clearly has its benefits: the size of the vector containing all features (F) scales linearly with the number of cells.

We examine one scenario (see Table 4.3) coded as follows:

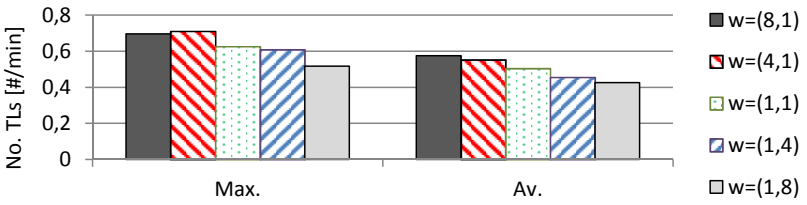
SC-LD-SA: SON and SONCO-R are on, 1 cell cluster with 21 cells, function approximation through *state aggregation*.

The rest of the details can be found in in Table 4.3.

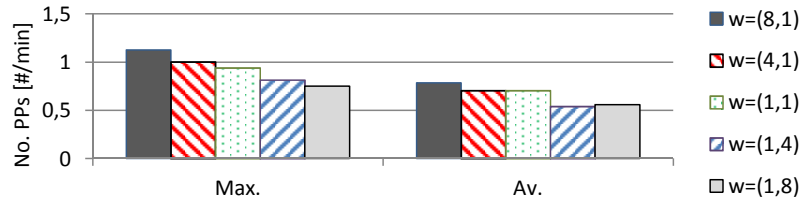
In Figures 4.14a, 4.14b and 4.14c we plot the maximum and the average (over all cells) of the time-averaged cell load, the time-averaged number of too late HOs and the time-averaged number of ping-pongs, respectively. The time-average is done over the last half of the simulation time. Note that w is the weights vector.



(a) Average Load



(b) Average Number of Too Late HOs



(c) Average Number of Ping Pongs

Figure 4.14: Results: scalar distributable regret - SA (for parameter conflict resolution)

In Fig. 4.14a one can see that a better load balancing is achieved by giving bigger weights to the MLB instances. Basically the overloaded cells (the ones containing the traffic hotspots) are allowed to *off-load* more. On the other hand giving a bigger priority to the MRO prevents these cells from *off-loading* as much as the MLB would want (i.e. to decrease the CIO) causing the maximum load to degrade by up to 9.9%. The HO borders are pushed *further away* from

the overloaded cells. Thus the number of Too Late HOs towards the overloaded cells from neighbouring base stations is reduced at the cost of slightly increasing the number of Too Late HO from these cells towards their neighbours. Overall we reduce the maximum number of too late HOs by up to 28.7% (Fig. 4.14b).

Typically the number of ping pongs depends mostly on the Hystereses, but as we can see in Fig. 4.14c they are also impacted by the weights. If we reduce the number of too late HOs by means of CIO configurations then the MRO has more flexibility in tuning the Hysteresis in order to further reduce the number of HO ping-pongs (at most 33.6%).

In the second case we consider an approximation which uses *linear features*, i.e. the features are linear functions of the network parameters. For this the regret function has to be as simple as possible such that the use of *linear features* makes sense. For this purpose we choose to have one regret component per update request (only for the requests targeting the CIO). Thus we define a bijection $b : \mathcal{I} \to \mathcal{N} \times \{CIO\} \times \mathcal{Z}$ and we set the regret function:

$$\rho_i(u) = \left|u^z_{n,k}\right| \cdot w_z$$

where $(n, k, z) = b(i)$. Consequently we consider the feature vector corresponding to the regret component i to be a linear function of: a constant and the differences between the CIO of cell n and the CIOs of its neighbouring cells:

$$F_i(p) = \left(\left(p_{n',k} - p_{n,k}\right)_{n' \in \bar{\mathcal{N}}_n \setminus \{n\}}, \quad 0.1 \right).$$

In this case also, the feature vector scales linearly with the number of cells. Moreover it is considerably smaller than in the first case. We examine one scenario coded as follows:

SC-LD-LF: SON and SONCO-R are on, 1 cell cluster with all cells, function approximation through *linear features*.

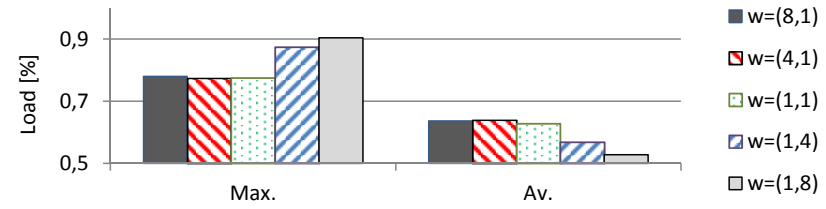
The rest of the details can be found in in Table 4.3.

We plot the maximum and the average (over all cells) of: the time-averaged cell load, the time-averaged number of too late HOs and the time-averaged number of ping-pongs, in Figures 4.15a, 4.15b and 4.15c respectively. The time-average is done over the last half of the simulation time. Note that w is the weights vector.

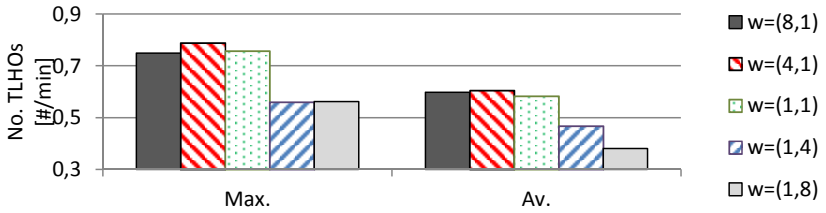
Fig. 4.15a shows that assigning a bigger weight to the MLB function makes the load balancing work better, reducing the maximum average load (bellow $\mathbb{T}_{LD}$). In our scenario what happens is that the cells with the traffic hotspots are allowed to offload more on their neighbours. This off-loading generates more too late HOs as the HO borders are not any-more within the actual coverage borders (Fig. 4.15b). An increased number of too late HOs complicates the job of the MRO instances and thus the number of ping pongs also increases (Fig. 4.15c).

On the other hand giving a bigger weight to the MRO prioritizes the reduction of the number of too late HOs (up to 25% reduction in Fig. 4.15b). As expected the price to pay is in terms of load balancing (Fig. 4.15a) where the maximum average load increases (up to 15%). The number of ping pongs is also impacted by the weights, but as it is not directly part of the update request sent by the MRO instances on the CIO we cannot see a trend, but the max is clearly lower than $\mathbb{T}_{PP}/T = 2\,[\#/min]$ for any weight.

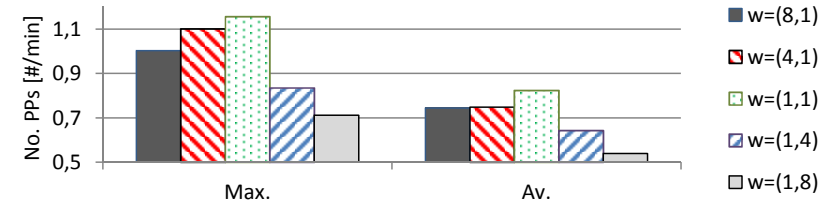
Note that in Fig. 4.15a the load difference between the cases $w = (1, 1)$, $w = (1, 4)$ is quite abrupt, whereas otherwise it is almost constant. This is due to the *Linear Features*. As the value function is linear w.r.t. the network parameters, the optimal action is to steer the network in configurations where the parameters are either maximum or minimum (depending on the sign of their coefficient). So basically what we can see in Fig. 4.15a for the two weight ($w = (1, 1)$, $w = (1, 4)$) is exactly the outcome of a jump of the network parameter configuration from one extremity to the other.



(a) Average Load



(b) Average Number of Too Late HOs [#/min]



(c) Average Number of Ping Pongs [#/min]

Figure 4.15: Results: scalar distributable regret - LF (for parameter conflict resolution)

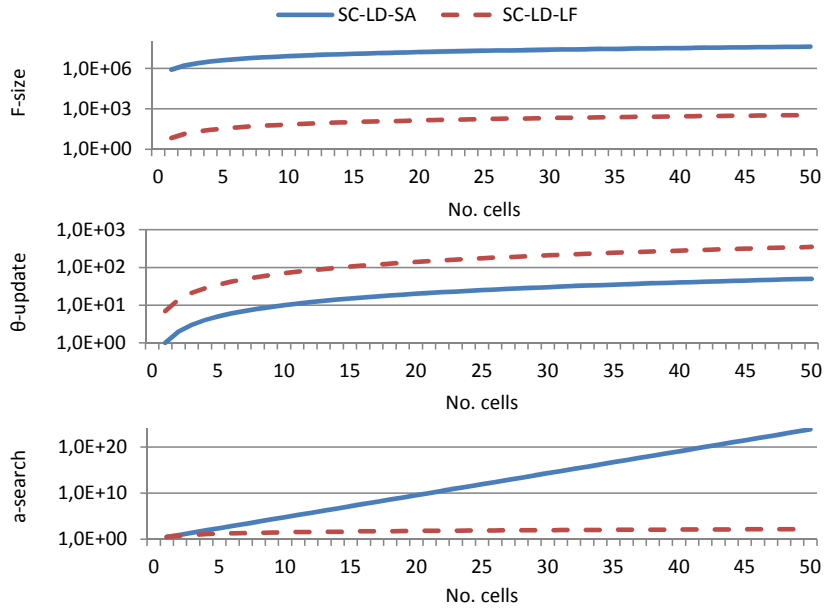


Figure 4.16: Complexity: scalar distributable regret (for parameter conflict resolution)

We plot in Figure 4.16 the complexity of the F-size, θ-update and a-search, w.r.t. the number of cells, for the two scenarios SC-LD-SA and SC-LD-LF. Both prove to have a linear scalability of the F-size and θ-update. However, the a-search complexity scales linearly in the case of using *linear features* (SC-LD-LF) and exponentially in the other case (SC-LD-SA).

Scenario	SC-LC-SA SC7-LC-SA	SC-LD-SA	SC-LD-LF
No. cells	21 Macro cells - hexagonal topology, with wraparound		
Clusters	1 vs 3	1	
ISD	1732 m		
Carrier/Bw/TxPow	Carrier = 2 GHz, Bandwidth = 10 MHz, CellTxPower = 46 dBm (40W)		
Propagation Model	Channel model #2 in Table A.1, (3GPP Case 3 [54])		
Traffic type	FTP-like , constant file size $FS = 16[Mb/UE]$, UE arrival: SPPP, arrival rates: general (λ_G, per hexagon) and HotSpot (λ_{HS}, per HS)		
$(\lambda_G, \lambda_{HS})\, FS\ [Mb/s]$	$(9; (30; 25; 20))$	$(9; 30)$	
Simulated time	16 days	2 days	2 days
SON functions	$\mathcal{Z} = \{MLB, MRO\}$ (for a Macro network, see Appendix B)		
Tuned parameters	CIO , Hysteresis, $\mathcal{K} = \{CIO, HYS\}$,		
CIO set [dB] $\mathcal{P}_{n,CIO}$	$\{-12, -9, .., 0\}$	$\{-12, -8, ..., 0\}$	
HYS set [dB] $\mathcal{P}_{n,HYS}$	$\{0, 1, ..., 12\}$		
ABS set [#] $\mathcal{P}_{n,ABS}$	$\{0\}$ (fixed)		
TTT	160ms		
Time window T	5 min		
SON settings	$\left(\mathbb{T}^L_{LD}; \mathbb{T}^H_{LD}\right) = (0.3; 0.8)$, $\mathbb{T}^H_{TL} = 8$, $\mathbb{T}^H_{PP} = 10$, $\mathbb{T}^H_{TL2} = 6$		
SON priority weights	$w = w_{\mathcal{Z}}$ (one weight per SON function)		
Conflict-solving target	parameter conflict resolution		
Criticalness indicator	used: $U^z_{n,k} \in [-1, 1]$		
state reduction	state aggregation	state aggregation	linear features
Distr. Learner (W)	no (centralized)	yes (per cell)	yes (per update request)
SONCO-R $(\alpha, \gamma, \epsilon)$	$(0.2; 0.8; 0.1)$		$(0.1; 0.8; 0.1)$
No. Learners $I =$	1	$I = \lvert\mathcal{N}\rvert$	$I = \lvert\mathcal{N}\rvert \cdot \lvert\{CIO\}\rvert \cdot \lvert\mathcal{Z}\rvert$,
b is a bijection			$b : \mathcal{I} \to \mathcal{N} \times \{CIO\} \times \mathcal{Z}$
Learners focus $\forall i \in \mathcal{I}$, $\mathcal{O}_i =$	$\mathcal{N} \times \{CIO\}$	$\bar{\mathcal{N}}_i \times \{CIO\}$	$\bar{\mathcal{N}}_n \times \{k\}$ where $(n, k, z) = b\,(i)$,
$\forall i \in \mathcal{I}$,$\mathcal{J}_i =$	$\{1, .., \lvert\mathcal{P}_{\mathcal{O}_i}\rvert\}$		$\left\{1, .., \lvert\bar{\mathcal{N}}_n\rvert\right\}$
$\forall p \in \mathcal{P}$, $\forall i \in \mathcal{I}$, $F_i(p) =$	$\left(\mathbb{I}_{\{p_{\mathcal{O}_i} = x\}}\right)_{x \in \mathcal{P}_{\mathcal{O}_i}}$		$\left(\left(p_{n',k} - p_{n,k}\right)_{n' \in \bar{\mathcal{N}}_n \setminus \{n\}}, \quad 0.1 \right)$ where $(n, k, z) = b\,(i)$,
size of F	$\sum_{i \in \mathcal{I}} \lvert\mathcal{P}_{\mathcal{O}_i}\rvert$		$\sum_{n \in \mathcal{N}} \lvert\bar{\mathcal{N}}_n\rvert$
$\forall u \in \mathcal{U}$, $\forall i \in \mathcal{I}$, $\rho_i\,(u) =$	$\max_{(n,z)} \left\lvert u^z_{n,CIO}\right\rvert w_z$	$\max_{z \in \mathcal{Z}} \left\lvert u^z_{i,CIO}\right\rvert w_z$	$\left\lvert u^z_{n,k}\right\rvert \cdot w_z$, where $(n, k, z) = b\,(i)$,
multi-dimension frame	no		

Table 4.3: Scenario summaries: scalar regret for parameter conflict resolution

4.5.3 Vector regret: parameter conflict resolution

In this section we tackle the parameter conflict resolution for a multi-dimension/vector regret. Consider a network with Macro and Pico cells. The Pico cells are meant to help the Macro cells providing them with a capacity bust. In this sense we say that a Macro cell may have several slave Pico cells, in our scenarios we consider the Pico cells under its coverage. A Pico cell is the slave of one and only one Macro cell. We identify a cell cluster to be composed of a Macro cell and its slave Pico cells. Focusing on one cell cluster, let $\mathcal{N}_M$ and $\mathcal{N}_P$ be the set of indexes of the Macro cells and Pico cells, respectively, ($\mathcal{N}_M = \{1\}$).

We use 3 SON functions: (MLB via) CRE, MRO and eICIC. For details see Appendix C. The CRE is a function which keeps the load of the Macro cell lower than a given threshold by tuning the CIOs of the slave Pico cells in order to *off-load* on them when needed. The MRO aims to reduce the number of ping-pong HOs and of too late HOs originating from the Pico cells by tuning the CIO and the HYS on the Pico cells. The eICIC aims to protect the users which have been forced to HO from the Macro cell to the slave Pico cells by tuning the number of ABS on the Macro cell.

We employ a network topology as represented in Fig. 4.17 with 3 cell clusters each of which is composed of 1 Macro cell and 2 *slave* Pico cells. As mentioned earlier, we consider one independent SONCO-R instance governing each of the cell clusters.

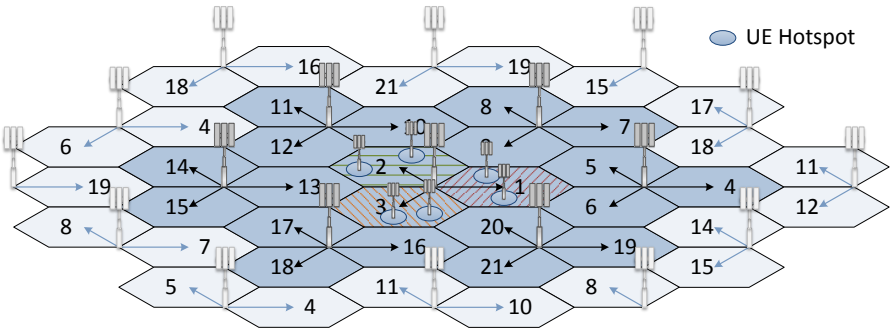


Figure 4.17: Network topology

In the SONCO-R designs, first of all we propose to neglect the parameters which are not a significant source of conflict. In our case we refer to the HYS, i.e. we assume that the HYS does not significantly impact the performance of the CRE or of the eICIC. We consider two scenarios:

SC-LC-SA: In the first case the regret is a simple scalar calculated over the entire cluster as follows:

$$\rho_d(u) = \max_{(n,k,z)\in\mathcal{O}_d\times\mathcal{Z}} \left|u^z_{n,k}\right| w_z$$

for any $d \in \mathcal{D} = \{1\}$ where $\mathcal{O}_d = \mathcal{N}_M \times \{ABS\} \cup \mathcal{N}_P \times \{CIO\}$. The feature vectors reflect all the possible configurations of the CIO and of the ABS:

$$F_d(p) = \left(\mathbb{I}_{\{p_{\mathcal{O}_d}=x\}}\right)_{x\in\mathcal{P}_{\mathcal{O}_d}}$$

SC-LD-LF: In the second case we target to enforce an approximation using *linear features*. The complicated formula used to calculate the regret in the fist scenario does not make it suitable to use *linear features*. Thus we consider a multi-dimension regret $D = 2(2N-1) > 1$. More precisely 2 regret components for each update requests targeting the ABS or the CIO. For this we first define a bijection $b : \mathcal{I} \to \mathcal{N}_M \times \{ABS\} \times \{eICIC\} \times \{\pm 1\} \cup \mathcal{N}_P \times \{CIO\} \times \{CRE, MRO\} \times \{\pm 1\}$ and then the regret function as:

$$\rho_d(u) = \left|u^z_{n,k}\right| w_z \mathbb{I}_{\{u^z_{n,k}\cdot sg>0\}}$$

for any $d \in \mathcal{D}$, where $(n,k,z,sg) = b(d)$. We approximate W_d with *linear features* as follows:

- for $z = CRE$, $k = CIO$ and $n \in \mathcal{N}_P$ and for $z = eICIC$, $k = 3$ and $n \in \mathcal{N}_M$ we approximate with a linear function of the CIOs of the cells $n' \in \mathcal{N}_P$, the ABSs of cells $n \in \mathcal{N}_M$ and one constant:

$$F_d(p) = 0.1 \cdot \left(10, \left(p_{n',ABS}\right)_{n' \in \mathcal{N}_M}, \left(p_{n',CIO}\right)_{n' \in \mathcal{N}_P}\right)$$

- for $z = MRO$, $k = CIO$ and $n \in \mathcal{N}_P$ we approximate with a linear function of the CIO of cell n and one constant:

$$F_d(p) = 0.1 \cdot (10, p_{n,CIO})$$

Leaving aside the approximations performed in the two designs, if we have a closer look we can see that there is a small difference w.r.t. what they optimize (reflected in ρ_d). The two targets can be summarized as follows:

SC-LC-SA: minimization of the expected worst regret:

$$\min_\pi \mathbb{E}_\pi \left[\sum_t \gamma^t \max_{(n,k,z)\in\mathcal{O}} \left|U^z_{(t),n,k}\right| \cdot w_z\right] =$$
$$\min_\pi \mathbb{E}_\pi \left[\sum_t \gamma^t \max_{(n,k,z,h)\in\mathcal{O}\times\{\pm 1\}} \left|U^k_{(t),n,k}\right| \mathbb{I}_{\left\{U^k_{(t),n,k} h > 0\right\}} \cdot w_z\right]$$

SC-LD-LF: minimization of the worst expected regret :

$$\min_\pi \max_{(n,k,z,h)\in\mathcal{O}\times\{\pm 1\}} \mathbb{E}_\pi \left[\sum_t \gamma^t \cdot \left|U^k_{(t),n,k}\right| \cdot \mathbb{I}_{\left\{U^k_{(t),n,k} h > 0\right\}} \cdot w_z\right]$$

with a myopic approach, i.e. $\gamma = 0$.

One can see that the outcome of the 2 are mathematically different but they are in the same spirit which is to provide a sense of fairness among the SON functions, as they minimize the maximum *criticalness* among the requests of the SON functions.

Fairness biasing In this sub-section we show the results aimed at proving the biasing mechanism, i.e. the capability of favouring one SON function or another. For this we perform simulations with stationary traffic λ (see Table 4.4) We evaluate the network parameters and performances during the last half of the simulation time. Fig. 4.18 shows averages over the parameters controlled by the SON functions: the average CIO (over time and all Pico cells), the average HYS (over time and all Pico cells) and the average no. of ABS (over time and all Macro cells), respectively, for different weights w attributed to the SON functions. The 2 designs can be identified by their names. They are followed by the weight of the 3 SON functions in the following order: CRE, MRO and eICIC . For simplicity we focus only on the conflict between CRE and MRO.

One can see how different weights impact the mentioned parameters. First we try to give some intuition in order to help the reader interpret the graphs. The CIO is the main target of CRE. In order to offload the Macro cell we increase the CIO of the Pico cells. However, in case that the CIO of the Pico cell increases too much we create the risk of having a lot of too late HOs; thus the MRO can get suffocated. This is why MRO also tunes the CIO to prevent this extreme off-loading.

We expect that a higher priority on the CRE should lead to more aggressive off-loading, i.e. bigger values of the CIO of Pico cells, and on the other hand a bigger priority of the MRO would restrict this offloading, i.e. lower CIO values. The no. of ABS will also have the same trend as eICIC reacts to the changes of the CRE: the bigger the CIO the more UEs to protect and thus the bigger the number of ABS sub-frames. This trend can be seen in Fig. 4.18 for the 2 designs SC-LC-SA and SC-LD-LF. The average HYS is also impacted by the different weights as for example a small CIO value would reduce the number of too late HOs and would thus allow for bigger HYS values to also reduce the ping-pong HOs.

As mentioned a bigger weight on the CRE leads to higher average CIO values. This forces the MRO to reduce the values of HYS, as can be seen in Fig. 4.18, in order to reduce the number of too late HOs. Naturally a bigger weight on the MRO will have the opposite effect, i.e. the average HYS values will be bigger as the number of too late HOs will not be so critical. MRO will profit from this by increasing the HYS in order to reduce the no. of ping-pong HOs.

In Figure 4.19 we represent averages over the KPIs used as inputs for the SON functions: the average load (over time and all Macro cells, over time and all Pico cells), the average no. of ping-pong HOs and of too late HOs (over time and over all Pico cells), and the average ratio of the bit-rates of UEs attached to the Macro cells by cell-edge UEs attached to its slave Pico cells (over time and all Macro cells). The latter is scaled by 0.4 in the figure. For more details on the KPI calculation see Appendixes C. One can again see the effects of giving higher priority to either CRE or to MRO, especially on the load of the Pico cells and also on the no. of ping pong HOs and the too late HOs. More precisely the load of the Pico cells, the no. of ping-pong HOs and the no. of too late HOs increases if we give higher priority to CRE and consequently decreases if a higher priority is given to MRO. We do not expect precise trends on the other KPIs, given the more complex dependencies.

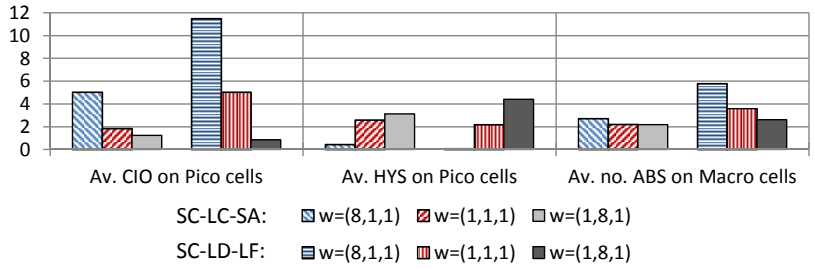


Figure 4.18: Time-averaged parameters

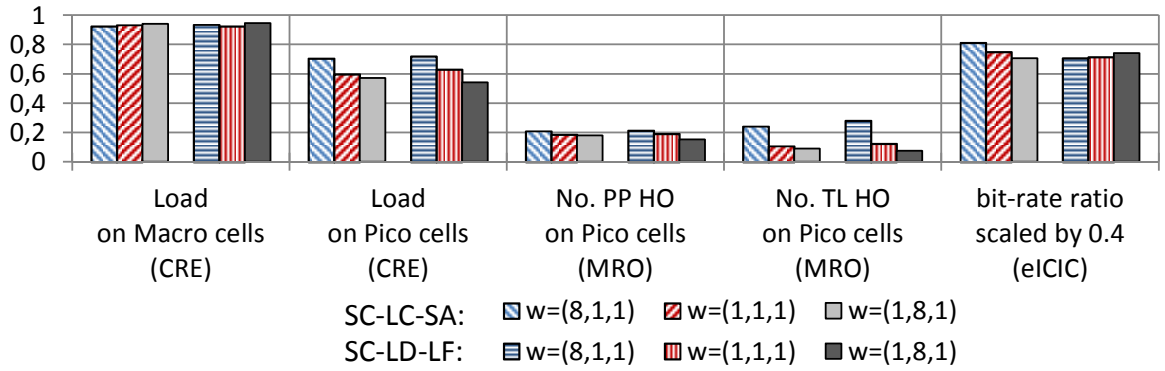


Figure 4.19: Time-averaged KPIs

Tracking capabilities In this sub-section we target to show how a trained SONCO-R reacts (in terms of the abstract regret) to variations of the traffic:

- the time needed for the algorithms to re-converge.
- the regret augment during the re-convergence period as compared to after the re-convergence.

For each design, SC-LC-SA and SC-LD-LF, we consider two scenarios. In the first scenario we split the simulation time into 2 equal time periods of 10 days. During the first period we consider a stationary traffic of 0.90λ (see Table 4.4) and during the second time period we consider

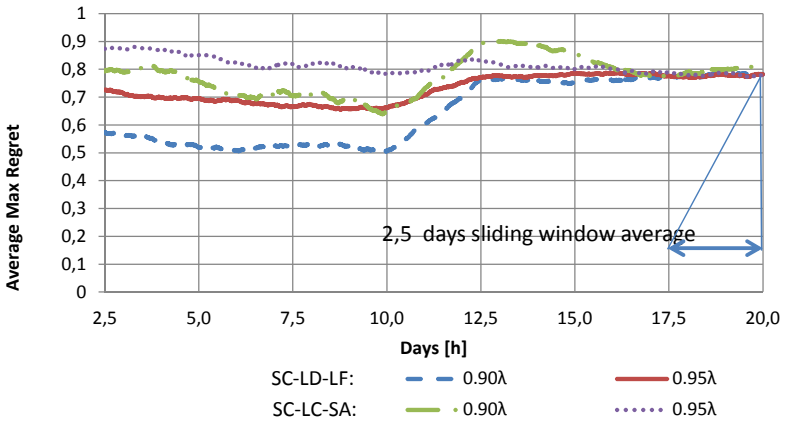


Figure 4.20: Tracking capabilities

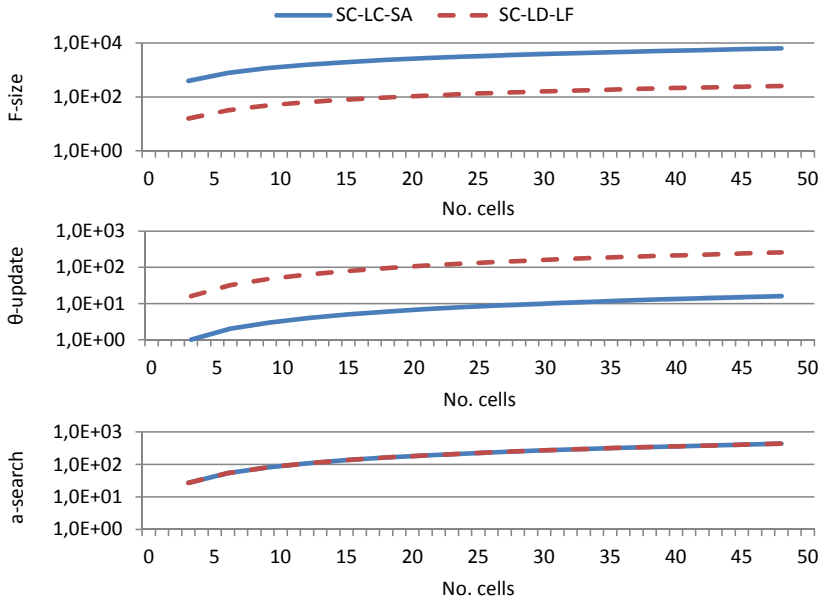


Figure 4.21: Complexity: vector regret, distributed learner (parameter conflict resolution)

a stationary traffic of λ. The second scenario is similar to the first one, the only difference being that during the first time period we consider a stationary traffic of 0.95λ. In Fig. 4.20 we plot the time-average of the maximum over all instantaneous regret components making use of a sliding time window of 2.5 days during the 20 days simulation period. From the performed tests one can see that the tracking capability of the SC-LD-LF design is much better then the one of the SC-LC-SA design. The convergence time is considerably shorter for the SC-LD-LF design and the regret augmentation during the re-convergence period as compared to the regret after re-convergence is imperceptible.

We plot in Figures 4.21 the complexity of the F-size, θ-update and a-search, w.r.t. the number of cells. Both scenarios prove to have good (linear) scalability of the F-size, θ-update and a-search. Having a more detailed look one can see that the scenario which uses *linear features* (SC-LD-LF) has better (smaller) values of the F-size complexity while the other one which uses *state aggregation* (SC-LC-SA) has better (smaller) values of the θ-update complexity.

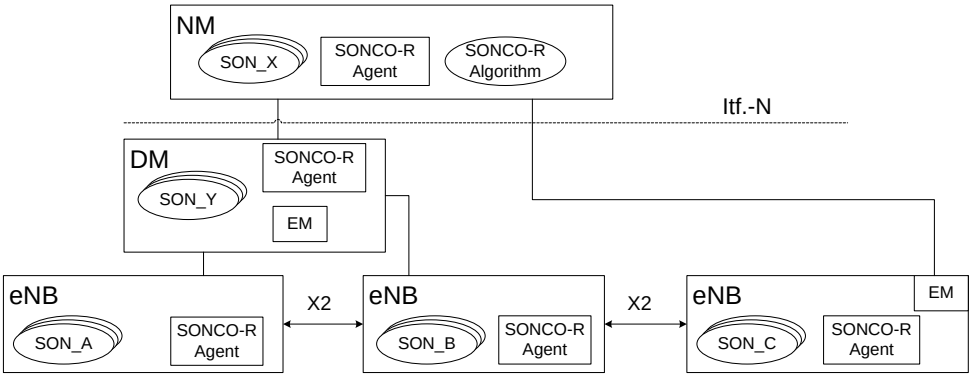
Scenario		SC-LC-SA	SC-LD-LF
No. cells		21 Macro cells, 6 Pico cells - hexagonal topology, with wraparound	
Clusters		3 (each contains 1 Macro cell and 2 Pico cells)	
ISD		1732 m	
Carrier/Bw/TxPow		Carrier = 2 GHz, Bandwidth = 10 MHz, MacroCellTxPower = 49 dBm, PicoCellTxPow=30 dBm	
Propagation Model		Channel model #21 in Table A.1, (3GPP Case 3 [54] with Pico)	
Traffic type		FTP-like , constant file size $FS = 16[Mb/UE]$, UE arrival: SPPP, arrival rates: general (λ_G, per hexagon) and HotSpot (λ_{HS}, per HS)	
$(\lambda_G, \lambda_{HS})\, FS\ [Mb/s]$		$(14; 14)$	
Simulated time		20 days	20 days
SON functions		$\mathcal{Z} = \{CRE, MRO, eICIC\}$ (for a HetNet, see Appendix C)	
Tuned parameters		CIO, Hysteresis $\mathcal{K} = \{CIO, HYS, ABS\}$	
CIO set [dB]	$n \in \mathcal{N}_M$	$\{0, 2, ..., 12\}$	$\{0, 1, ..., 12\}$
$\mathcal{P}_{n,CIO}$	$n \in \mathcal{N}_p$	$\{0\}$	
HYS. set [dB]	$n \in \mathcal{N}_M$	$\{0, 1, ..., 12\}$	
$\mathcal{P}_{n,HYS}$	$n \in \mathcal{N}_p$	$\{2\}$	
ABS set [#]	$n \in \mathcal{N}_M$	$\{0\}$ (out of 40)	
$\mathcal{P}_{n,ABS}$	$n \in \mathcal{N}_p$	$\{0, 4, ..., 36\}$ (out of 40)	
TTT		160ms	
Time window T		5 min	
SON settings		$\mathbb{T}^H_{LD} = 0.85$, $\mathbb{T}^H_{TL} = 0.1$, $\mathbb{T}^H_{PP} = 0.2$, $\mathbb{T}^L_{TR} = 1.0$ and $\mathbb{T}^H_{TR} = 2.5$	
SON priority weights		$w = w_{\mathcal{Z}}$ (one weight per SON function)	
Conflict-solving target		parameter conflict resolution	
Criticalness indicator		used: $U^z_{n,k} \in [-1, 1]$	
state reduction		state aggregation	linear features
Distr. Learner (W)		no (centralized)	yes (per update request)
SONCO-R $(\alpha, \gamma, \epsilon)$		$(0.2, 0.5, 0.1)$	$(0.5, 0.0, 0.1)$
No. Learners $D =$		1	$2\,(2\,\lvert\mathcal{N}\rvert - 1)$
b is a bijection			$b : \mathcal{I} \to \mathcal{N}_M \times \{ABS\} \times \{eICIC\} \times \{\pm 1\} \cup \mathcal{N}_P \times \{CIO\} \times \{CRE, MRO\} \times \{\pm 1\}$
Learners focus $\forall d \in \mathcal{D}$, $\mathcal{O}_d =$		$\mathcal{N}_M \times \{ABS\} \cup \mathcal{N}_P \times \{CIO\}$	$\begin{cases} \mathcal{N}_M \times \{ABS\}, & \text{if } k = ABS \\ \mathcal{N}_P \times \{CIO\}, & \text{if } k = CIO \end{cases}$ where $(n, k, z, sg) = b\,(d)$,
$\forall d \in \mathcal{D}$,$\mathcal{J}_d =$		$\{1, .., \lvert\mathcal{P}_{\mathcal{O}_d}\rvert\}$	$\begin{cases} N + 1, & \text{if } z \neq MRO \\ 2, & \text{if } z = MRO \end{cases}$
$\forall p \in \mathcal{P}$, $\forall d \in \mathcal{D}$, $F_d(p) =$		$\left(\mathbb{I}_{\{p_{\mathcal{O}_d} = x\}}\right)_{x \in \mathcal{P}_{\mathcal{O}_d}}$	if $z \neq MRO$: $0.1 \cdot \left(10, (p_{n',ABS})_{n' \in \mathcal{N}_M}, (p_{n',CIO})_{n' \in \mathcal{N}_P}\right)$, if $z = MRO$: $0.1 \cdot (10, p_{n,CIO})$, where $(n, k, z, sg) = b\,(d)$,
size of F		$\sum_{d \in \mathcal{D}} \lvert\mathcal{P}_{\mathcal{O}_d}\rvert$	$(\lvert\mathcal{N}_M\rvert + \lvert\mathcal{N}_P\rvert)\,(1 + \lvert\mathcal{N}_M\rvert + \lvert\mathcal{N}_P\rvert) + 2\,\lvert\mathcal{N}_P\rvert$
$\forall u \in \mathcal{U}$, $\forall d \in \mathcal{D}$, $\rho_d\,(u) =$		$\max_{(n,k,z) \in \mathcal{O}_d \times \mathcal{Z}} \left\lvert u^z_{n,k} \right\rvert w_z$	$\left\lvert u^z_{n,k} \right\rvert w_z \mathbb{I}_{\left\{u^z_{n,k} \cdot sg > 0\right\}}$, where $(n, k, z, sg) = b\,(d)$
multi-dimension frame		yes	

Table 4.4: Scenario summaries: vector regret for parameter conflict resolution

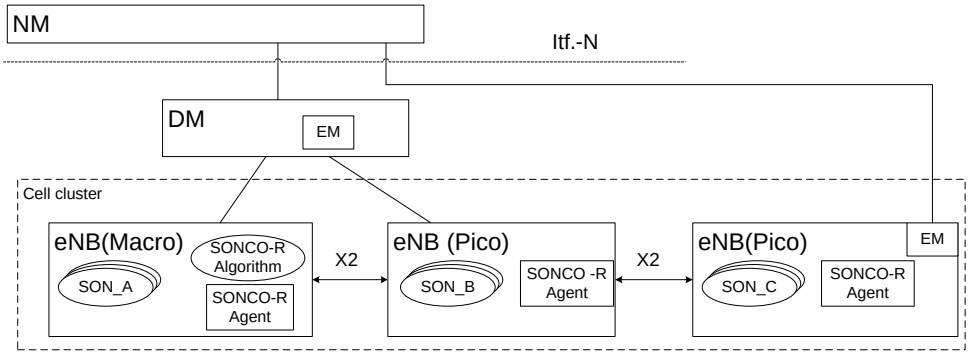
4.6 Logical framework architecture

For the implementation of the SONCO-R we consider 2 perspectives w.r.t. the number of network layers: Macro Network and HetNet. In the case of Macro Network the scope area of the SONCO-R might be very large and thus (normally) as the accept/deny update request decision is centralized it has to be taken higher up the hierarchy, i.e. above the Itf-N. The SONCO-R *Agents* have to collect the necessary information from the SON function and forward it to the corresponding SONCO-R *Algorithm*. The Itf-N has to be able to sustain this forwarding. We note that the employed SON functions send update requests every 5 minutes. Thus, at this time scale, feeding information to the SONCO-R in the NM should not be a problem. The proposed architecture is outlined in Fig. 4.22a.

In the case of the HetNet scenario, we remind that the Pico cells act as slaves of the Macro cell, i.e. they are meant to help the Macro cell by providing additional coverage and capacity. Thus we considered cell clusters composed by a Macro cell and its slave Pico cells. Each cell cluster can be led by its Macro cell. Thus in this scenario we have proposed that each SONCO-R instance governs independently a cell cluster, and the SONCO-R Algorithm instance will be running on the Macro cell/eNB. We expect the SONCO-R response time to be very good due to the limited scope-area. Obviously this is valid if the scope of the SON functions is limited to one cell cluster. The SONCO-R *Agents* have to collect the necessary information and ensure that it gets to the corresponding SONCO-R *Algorithm* directly or via the X2 interface . The proposed architecture is outlined in Fig. 4.22b.



(a) Macro Network



(b) HetNet

Figure 4.22: SONCO-R architecture

4.7 Concluding remarks

This work is different from current work found in the literature by the fact that we propose a SONCO-R that considers the SON instances as *black-boxes*, i.e. the algorithm inside is unknown, and we provide a RL-based SONCO-R solution. One important aspect is that we assume that

the parameter update requests sent by the SON functions include an additional metric reflecting how *critical* each request is. The SONCO-R decides which requests to accept and which to deny in targeting to minimize a regret function calculated using this *criticalness* indication. A well known problem of RL is that its complexity does not scale well. We tackle this issue making use of two function approximation techniques: *state aggregations* and *linear features*. We show how we are able to reduce the requirements in terms of memory and computational effort.

Breaking the regret into several components provides a means to make the solutions scalable. More precisely, we perform an approximation for each component by reducing its dependencies to a local scope. This is reasonably easy to be done through *state aggregation*. Moreover, the use of *linear features* (to approximate these components) allows for a much faster convergence and has low memory requirements. However this is suitable only in the case of a vector of components and not in the case where the components are summed up to form the regret.

Breaking the focus area into several sub-areas and creating cell clusters, is a good solution to reduce the a-search complexity. However it should be carefully done. Ideally the cell clusters should be formed such that the inter-cluster dependencies are minimal.

Chapter 5

Mobility parameters

In Section 2.1.1 we have introduced the HandOver (HO) parameters: namely the Cell Individual Offset (CIO) and HO Hysteresis (HYS). Throughout our work (chapters 3 and 4) we have considered that the HO parameters are established *per-cell* as opposed *to per-cell-pair*. In this section we motivate our choice of using *per-cell* HO parameters. In the literature we typically find two approaches:

- *per-cell-pair* HO parameters: on each cell we have a set of HO parameters for each neighbouring cell [43, 44, 45, 46, 47, 48].

- *per-cell* HO parameters: on each cell we have only one set of HO parameters used for all neighbouring cells [49, 50, 28, 33, 51, 34]. Note that this can be seen as a particular case of the first approach.

The use of *per-cell-pair* HO parameters at first seems to be a promising strategy, allowing individual HO configurations w.r.t. each neighbour, as compared to using *per-cell* HO parameters which would limit the manoeuvrability of the user mobility. However, using *per-cell-pair* HO parameters creates the risk of having areas where we may continuously experience Chained-HOs (CH). We consider a CH to be the event where two successful HOs are performed one shortly after another, regardless of the target cell. Note that a HO ping-pong event (see Section 2) is a special case of a CH where the the target of the second HO is the serving cell of the first HO. We mention that the A3-event envisages *per-cell* parameters [5].

To the best of our knowledge this problem has not been signalled in the literature. In this chapter:

- We show that using *per-cell-pair* HO parameters creates the risk of having Continuous CH (CCH) areas,

- We give the necessary and sufficient conditions under which CCH areas are avoided for the case of *per-cell-pair* HO parameters,

- We show that one solution to avoid the CCH areas is to use *per-cell* HO parameters,

- We quantify the size of the CCH areas,

- We show the effects of CCH areas in an LTE network simulator.

The remainder of this section is structured as follows: sub-section 5.1 describes formally the problem, sub-section 5.2 presents some preliminary results, sub-section 5.3 presents the network configurations that cause CCH areas, sub-section 5.4 contains numeric results and simulation results and sub-section 5.5 concludes the chapter.

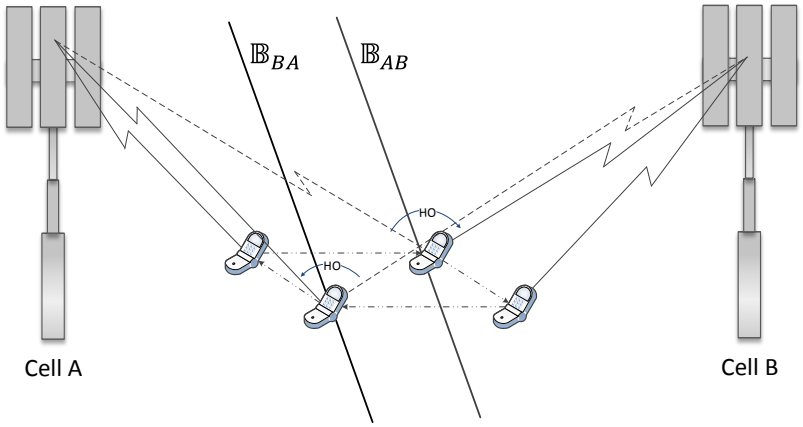


Figure 5.1: HandOver mechanism

5.1 Problem description

5.1.1 HandOver test: two cells

We consider a network formed by several cells and UEs placed on a geographical area $\mathcal{D} \subset \mathbb{R}^2$. As mentioned, concerning the HO parameters we have identified 2 approaches: the first one uses *per-cell-pair* HO parameters and the second one uses *per-cell* HO parameters. The second approach is a particular case of the first approach, as in the first case one can always set equal values for the HO parameters w.r.t. all neighbours. Therefore we formally describe the *per-cell-pair* HO model as the general case.

Figure 5.1 shows how these parameters are used for an example with two cells: A and B. Consider a static UE at a given location $\ell \in \mathcal{D}$. Assume that this UE is served by cell A. At a given time instance this UE checks if it should HO to a neighbouring cell, in this case the only option is cell B. Say it receives an average power $RxPow_A$ from cell A and $RxPow_B$ from cell B (in dB). In order to take a HO decision. By generalizing eq. (2.2), a HO decision is taken in if the following test returns a strictly positive value :

$$\mathbb{T}_{AB}(\ell) = RxPow_B(\ell) - RxPow_A(\ell) + CIO_{BA} - CIO_{AB} - HYS_{AB} \tag{5.1}$$

where CIO_{AB} and CIO_{BA} are the CIOs concerning the cell pair A-B, $HYS_{AB} \geq 0$ is the HYS for a HO from cell A to cell B, all in dB. More precisely, this condition has to be met for a minimum time duration, the so-called Time To Trigger (TTT) [5]. For now to simplify things we assume the TTT to be 0. Note that the HO hysteresis in dB is non-negative as it is intended to delay the HO, in this case from cell A to cell B. Similarly, a UE attached to cell B will perform a HO from cell B to cell A if $\mathbb{T}_{BA}(\ell) > 0$.

Each pair of two cells, say A and B, has two HO offsets CIO_{AB} and CIO_{BA} and two HO hystereses HYS_{AB} and HYS_{BA}. We define the HO border from cell A to cell B as:

$$\mathbb{B}_{AB} = \{\ell \in \mathcal{D} : \mathbb{T}_{AB}(\ell) = 0\} \tag{5.2}$$

Similarly we define $\mathbb{B}_{BA}$. We note that $\mathbb{B}_{BA} \neq \mathbb{B}_{AB}$ unless the HO hystereses are null: $HYS_{BA} = HYS_{AB} = 0$. Typically HYS_{BA} and HYS_{AB} should be sufficiently large in order to avoid HO ping-pongs for the users moving close to cell borders.

5.1.2 Generalization

For an arbitrary set of cells $\{A, B, ...\}$ and any UE at location $\ell \in \mathcal{D}$ we define the HO function as follows:

$$f_i(\ell) = \begin{cases} i & \text{if } \mathbb{T}_{i,j}(\ell) \leq 0, \forall j \in \{A, B, ...\} \setminus \{i\} \\ \text{argmax}_{j \in \{A,B,....\} \setminus \{i\}} \mathbb{T}_{i,j}(\ell) & \text{otherwise} \end{cases} \tag{5.3}$$

where $f_i(\ell) = k$ means that a user (UE) in location ℓ when attached to cell i should handover to cell k. Obviously if $f_i(\ell) = i$ there will be no HO. The HO procedure is described in Alg. 5.1 by means of a *HO automata*.

Algorithme 5.1 HO automata

For a UE in location ℓ, let $i(t=0)$ be the initial cell attachment,
At each time instance, $t \in \mathbb{N}$, the UE will perform a HO from $i(t)$ to $i(t+1) = f_{i(t)}(\ell)$

5.1.3 Problematic HO parameter configurations

Next we show that using *per-cell-pair* HO parameters, although it may have the mentioned advantages, it is very likely to create CCH areas. A CH is the UE event where a new HO is performed shortly after another (regardless of the targeted cell). Considering the *HO automata* in Alg. 5.1, we define the CCH area as the area where f is cyclic:

$$\text{CCH area} = \{\ell | \ell \in \mathcal{D}, f \text{ does not converge, it is cyclic}\} \tag{5.4}$$

Proposition 5.1 (HO tests)**.** *Given some time instance, for any $i, j \in \{A, B, ...\}$ where $i \neq j$, having $\mathbb{T}_{i,j}(\ell) > 0$ implies $\mathbb{T}_{j,i}(\ell) \leq 0$.*

Proof. Indeed, as $HYS_{ij} \geq 0$ and $HYS_{ij} \geq 0$ we get:

$$0 \leq HYS_{ij} + HYS_{ji}. \tag{5.5}$$

Thus, say that for a point $\ell \in \mathcal{D}$ we have $\mathbb{T}_{i,j}(\ell) > 0$. Lets assume that $\mathbb{T}_{j,i}(\ell) > 0$. Then, by explicitating using eq. (5.1) and adding up the two inequalities, we get that eq. (5.5) should be false, and thus our assumption is false. □

Proposition 5.1 ensures that no (static or low speed) UE is susceptible to continuously experiencing pairwise CHs between cells i and j (in a static environment). However, there is still a considerable risk of having non-pairwise CCH areas. To better understand the problem, we plot in Figure 5.2 an example of such a CCH area, making use of 3 cells A, B and C. We note that in fact the HO borders are not necessarily lines, so Figure 5.2 contains a rough representation. To ease the reading we explicitate the HO function $f_A(\ell)$ in Table 5.1 w.r.t. the corresponding tests. The black area represents the CCH area. For example, say a UE arrives in this area. Without loss of generality assume it is attached to cell A. It cannot HO to B ($\mathbb{T}_{AB}(\ell) \leq 0$), it can and will perform a HO to cell C, as it is past border $\mathbb{B}_{AC}$ ($\mathbb{T}_{AC}(\ell) > 0$). Attached to cell C, it cannot HO to cell A ($\mathbb{T}_{CA}(\ell) \leq 0$), but it can and will perform a HO to cell B as it is past border $\mathbb{B}_{CB}$ ($\mathbb{T}_{CB}(\ell) > 0$). Attached to cell B, it cannot HO to cell C ($\mathbb{T}_{BC}(\ell) \leq 0$) but it can and will perform a HO to cell A as it is past border $\mathbb{B}_{BA}$ ($\mathbb{T}_{BA}(\ell) > 0$). Being back in cell A, this will repeat infinitely. Thus the *HO automata* is cyclic.

$\mathbb{T}_{AB}(\ell) > 0$	0	1	0	1	1	1
$\mathbb{T}_{AC}(\ell) > 0$	0	0	1	1	1	1
$\mathbb{T}_{AC}(\ell) > \mathbb{T}_{AB}(\ell)$	-	-	-	0	1	0
$\mathbb{T}_{AB}(\ell) > \mathbb{T}_{AC}(\ell)$	-	-	-	1	0	0
$\mathbb{T}_{AB}(\ell) = \mathbb{T}_{AC}(\ell)$	-	-	-	0	0	1
$f_A(\ell)$	A	B	C	B	C	B or$C$

Table 5.1: HO automata for cell A

Remark 5.1 (per-cell parameters)**.** *Using* per-cell *HO parameters is a particular case of using* per-cell-pair *HO parameters, i.e. consider a set of cells $\{A, B, ...\}$ then $\forall i, j \in \{A, B, ...\}$, $i \neq j$, we have that $CIO_{ij} = CIO_i$ and $HYS_{ij} = HYS_i$, where CIO_i and HYS_i are the HO offset and HO hysteresis, respectively, defined for cell i w.r.t. all neighbouring cells.*

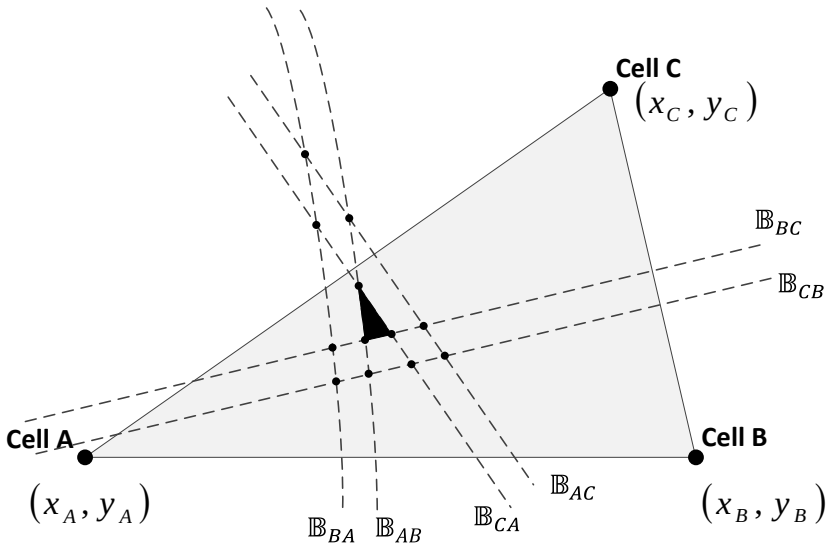


Figure 5.2: CCH areas

5.2 Preliminaries

We target to investigate the conditions of not having CCH areas and to quantify their size when such areas exist. For this we consider 3 cells A,B and C (as in the previous section), non-collinear, omnidirectional (see Fig. 5.2) and LTE capable.

We focus on the general case with *per-cell-pair* HO parameters. Let $\gamma = (\mathbf{CIO}, \mathbf{HYS})$ represent an arbitrary network configuration of the HO parameters, where:

$$\mathbf{CIO} = (CIO_{ij})_{i,j\in\{A,B,C\},i\neq j}\,,\ \mathbf{HYS} = (HYS_{ij})_{i,j\in\{A,B,C\},i\neq j}\,. \tag{5.6}$$

We recall that the elements of matrix **HYS** are non-negative. Let Γ be the set of all possible values of γ. We introduce the notation of the HO margin (HOM): $HOM_{ij} = CIO_{ij} - CIO_{ji} + HYS_{ij}$ for any $i, j \in \{A, B, C\}$, $i \neq j$,. This allows for a characterization of the cell borders under the form:

$$\mathbb{B}_{ij} = \{\ell \in \mathcal{D} : RxPow_i(\ell) + HOM_{ij} = RxPow_j(\ell)\}\,, \forall i, j \in \{A, B, C\}\,, i \neq j. \tag{5.7}$$

Consider $\mathcal{A} = (x_A, y_A)$, $\mathcal{B} = (x_B, y_B)$ and $\mathcal{C} = (x_C, y_C)$ to be the coordinates of the 3 cells A,B and C, respectively, where $\mathcal{A}, \mathcal{B}, \mathcal{C} \in \mathcal{D}$. Let $d_{(\ell_1,\ell_2)}$ be the distance between 2 points on the map ℓ_1 and ℓ_2, where $\ell_1, \ell_2 \in \mathcal{D}$. In particular we denote $c = d_{(\mathcal{A},\mathcal{B})}$, $a = d_{(\mathcal{B},\mathcal{C})}$, and $b = d_{(\mathcal{C},\mathcal{A})}$. We explicitate $RxPow_A^{\kappa}(t)$ making use of a typical pathloss model:

$$RxPow_A(\ell) = TxPow + k - \alpha 10\log_{10} d_{(\mathcal{A},\ell)} \tag{5.8}$$

where k is the path gain coefficient in [dB], $\alpha > 0$ the path gain exponent and $TxPow$ is the cell transmission power (the same for all cells) in [dB]. Similarly we can also explicitate $RxPow_B^{\kappa}(t)$ and $RxPow_C^{\kappa}(t)$.

Remark 5.2 (pathloss model)**.** *A more complicated model could have been used in eq. (5.8), for example including shadowing, fast fading, etc. This would significantly increase the complexity of derivations, making them difficult to follow, without providing a significant change in the conclusions. However, to check if the conclusions still hold in a more realistic scenario, we provide simulations in the results Section 5.4.3 considering the* 3GPP Case 1 *scenario [54] (see #1 in Table A.1).*

Note that the border equation in (5.7), say for the HO from cell A to cell B, can be written as:

$$\mathbb{B}_{AB} = \left\{\ell \in \mathcal{D} : \log_{10} d_{(\mathcal{A},\ell)} - HOM_{AB}/\alpha = \log_{10} d_{(\mathcal{B},\ell)}\right\} = \left\{\ell \in \mathcal{D} : d_{(\mathcal{A},\ell)} \cdot \rho_{AB} = d_{(\mathcal{B},\ell)}\right\} \tag{5.9}$$

where $\rho_{AB} = 10^{-HOM_{AB}/\alpha}$. We extend this notation to $\forall i, j \in \{A, B, C\}$, $i \neq j$, $\rho_{ij} = 10^{-HOM_{ij}/\alpha}$. In the end we recover the equation of an Apollonian circle or line [52, 53].

Lemma 5.1. *Consider the Apollonian circle* $\mathbb{B}(\ell_1, \ell_2, z) = \left\{\ell \in \mathcal{D} : d_{(\ell_1,\ell)} \cdot z = d_{(\ell_2,\ell)}\right\}$ *corresponding to location points* $\ell_1, \ell_2 \in \mathcal{D}$ *and ratio* z. *We can explicitate the HO borders as:*

$$\mathbb{B}_{AB} = \mathbb{B}(\mathcal{A}, \mathcal{B}, \rho_{AB}) = \mathbb{B}(\mathcal{B}, \mathcal{A}, 1/\rho_{AB}) \tag{5.10}$$

and similarly all the other borders. For $z = 1$ *it is in fact a line.*

Note that $\forall i, j \in \{A, B, C\}$, $i \neq j$, $\rho_{ij} \in (0, \infty)$ given $HOM_{ij} \in \mathbb{R}$, and as a consequence of eq. (5.5) we have $\rho_{ij}\rho_{ji} \leq 1$.

In figure 5.3 we represent $\mathbb{B}_{AB}$ corresponding to different values of HOM_{AB}. One can easily identify that the borders are in fact Apollonian circles [52, 53]. For $HOM_{AB}[dB] \to 0$ ($\rho_{AB} \to 1$) the border is not any more a circle but a line, the perpendicular bisector of the segment $[\mathcal{AB}]$.

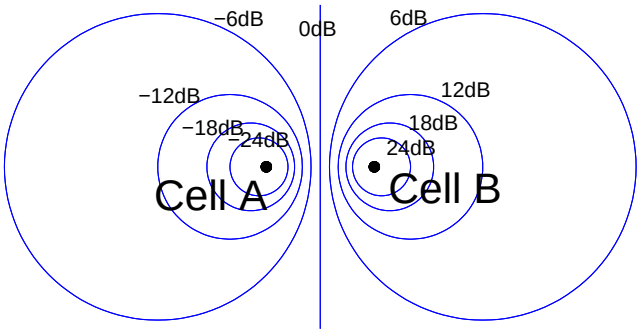


Figure 5.3: Border representation $\mathbb{B}_{AB}$ for different values of HOM_{AB} (Apollonian circles)

Next we provide two lemmas that help in proving the theorems in the main section 5.3, which will identify the network configurations which are subject to CCH areas. Thus, we consider 3 arbitrary variables $z_{AB}, z_{BC}, z_{CA} \in (0, \infty)$ and we analyse the intersection of the following (see Fig. 5.4): $\mathbb{B}(\mathcal{A}, \mathcal{B}, z_{AB})$, $\mathbb{B}(\mathcal{B}, \mathcal{C}, z_{BC})$ and $\mathbb{B}(\mathcal{C}, \mathcal{A}, z_{CA})$, i.e. any possible HO border w.r.t. the 3 cell pairs (regardless of the direction).

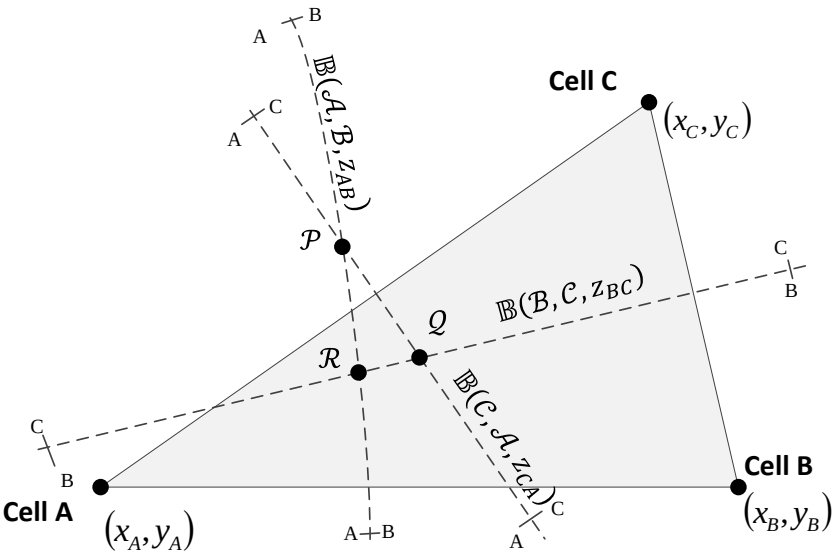


Figure 5.4: Border intersections

The following two theorems identify the conditions under which the three borders intersect. First we identify the pairwise intersection conditions.

Lemma 5.2. *Considering 3 cells A,B and C, then for any $z_{AB}, z_{BC}, z_{CA} \in (0, \infty)$ we have:*

$$\begin{aligned}
&\mathbb{B}(\mathcal{A}, \mathcal{B}, z_{AB}) \cap \mathbb{B}(\mathcal{B}, \mathcal{C}, z_{BC}) \neq \emptyset \Leftrightarrow (a \cdot z_{CA} + b/z_{BC})^2 \geq c^2 \geq (a \cdot z_{CA} - b/z_{BC})^2, \\
&\mathbb{B}(\mathcal{B}, \mathcal{C}, z_{BC}) \cap \mathbb{B}(\mathcal{C}, \mathcal{A}, z_{CA}) \neq \emptyset \Leftrightarrow (c \cdot z_{BC} + a/z_{AB})^2 \geq b^2 \geq (c \cdot z_{BC} - a/z_{AB})^2, \\
&\mathbb{B}(\mathcal{C}, \mathcal{A}, z_{CA}) \cap \mathbb{B}(\mathcal{A}, \mathcal{B}, z_{AB}) \neq \emptyset \Leftrightarrow (b \cdot z_{AB} + c/z_{CA})^2 \geq a^2 \geq (b \cdot z_{AB} - c/z_{CA})^2.
\end{aligned} \tag{5.11}$$

Proof. See appendix 5.1. □

Second we identify the conditions under which all the borders have a common point.

Lemma 5.3. *Considering 3 cells A,B and C, for $z_{AB}, z_{BC}, z_{CA} \in (0, \infty)$ the following are equivalent:*

1. $\mathbb{B}(\mathcal{A}, \mathcal{B}, z_{AB}) \cap \mathbb{B}(\mathcal{B}, \mathcal{C}, z_{BC}) \cap \mathbb{B}(\mathcal{C}, \mathcal{A}, z_{CA}) \neq \emptyset$
2. $z_{AB} \cdot z_{BC} \cdot z_{CA} = 1$ *and* $(b \cdot z_{AB} + c/z_{CA})^2 \geq a^2 \geq (b \cdot z_{AB} - c/z_{CA})^2$
3. $z_{AB} \cdot z_{BC} \cdot z_{CA} = 1$ *and* $(c \cdot z_{BC} + a/z_{AB})^2 \geq b^2 \geq (c \cdot z_{BC} - a/z_{AB})^2$
4. $z_{AB} \cdot z_{BC} \cdot z_{CA} = 1$ *and* $(a \cdot z_{CA} + b/z_{BC})^2 \geq c^2 \geq (a \cdot z_{CA} - b/z_{BC})^2$

Proof. See Appendix 5.2. Lemma 5.2 is used in this proof. □

An interesting observation is that given the fact that the borders are Apollonian circles/lines of a triangle ($\triangle \mathcal{ABC}$) means that if they have a common intersection point then there exists at most one more intersection point, and there are no other intersection points between any 2 borders. They are called isodynamic points [52].

5.3 Main results

In this section we evaluate the configuration domain that ensures us to have no CCH areas. There are 2 possible CCH areas. We define Γ_{ABCA} the configurations domain of γ where we have CCH areas in the sense $A \to B \to C \to A$:

$$\Gamma_{ABCA} = \{\gamma : \exists \ell \in \mathcal{D} \text{ s.t. } f_\ell(A) = B, f_\ell(B) = C, f_\ell(C) = A\}; \tag{5.12}$$

analogous we also define Γ_{ACBA} for the opposite sense:

$$\Gamma_{ACBA} = \{\gamma : \exists \ell \in \mathcal{D} \text{ s.t. } f_\ell(A) = C, f_\ell(C) = B, f_\ell(B) = A\}. \tag{5.13}$$

The following theorems identify the network configurations under which we have CCH areas. This kind of area can create some major problems. UEs that continuously HO between cells consume a significant amount of resources which translates into bad Quality of Service for all the UEs.

Theorem 5.1. *Considering 3 cells A,B and C, a network configuration $\gamma \in \Gamma_{ABCA}$ if and only if the following conditions are simultaneously met:*

$$\begin{cases}
\rho_{AB} \cdot \rho_{BC} \cdot \rho_{CA} > 1 \\
c/\rho_{CA} < a + b \cdot \rho_{AB} \\
b/\rho_{BC} < c + a \cdot \rho_{CA} \\
a/\rho_{AB} < b + c \cdot \rho_{BC}
\end{cases} \tag{5.14}$$

Note $\rho_{AB} \cdot \rho_{BC} \cdot \rho_{CA} > 1$ is equivalent to $HOM_{AB} + HOM_{BC} + HOM_{CA} = -\alpha log_{10} \rho_{AB} \cdot \rho_{BC} \cdot \rho_{CA} < 0$.

Proof. See Appendix 5.3. □

Note that Theorem 5.1 can be adapted for $\gamma \in \Gamma_{ACBA}$ simply by switching B and C. The next theorem focuses on the *per-cell* HO parameters, which is a particular case of *per-cell-pair* HO parameters.

Theorem 5.2. *In the case of per-cell HO parameters there are no CCH areas.*

Proof. See Appendix 5.2. □

Thus, when using *per-cell* HO parameters we are CCH free. However, when using *per-cell-pair* HO parameters special precautions have to be taken to avoid having CCH areas.

We now discuss the above results. We outline two solutions for avoiding CCH in case we use *per-cell-pair* parameters. One is that the tuning mechanism used by the system designer should envisage a simple check that the conditions in Theorem 5.1 are not met. The second solution is to patch the HO procedure by trying to hint that a UE might be caught in a CCH and forbid it to perform further HOs. This requires some caution in order not to generate radio link failures due to the HO restrains. We consider the later to be outside the scope of the work as it targets the HO procedure not the HO parameter optimization.

5.4 Numerical results.

5.4.1 Problematic network configurations

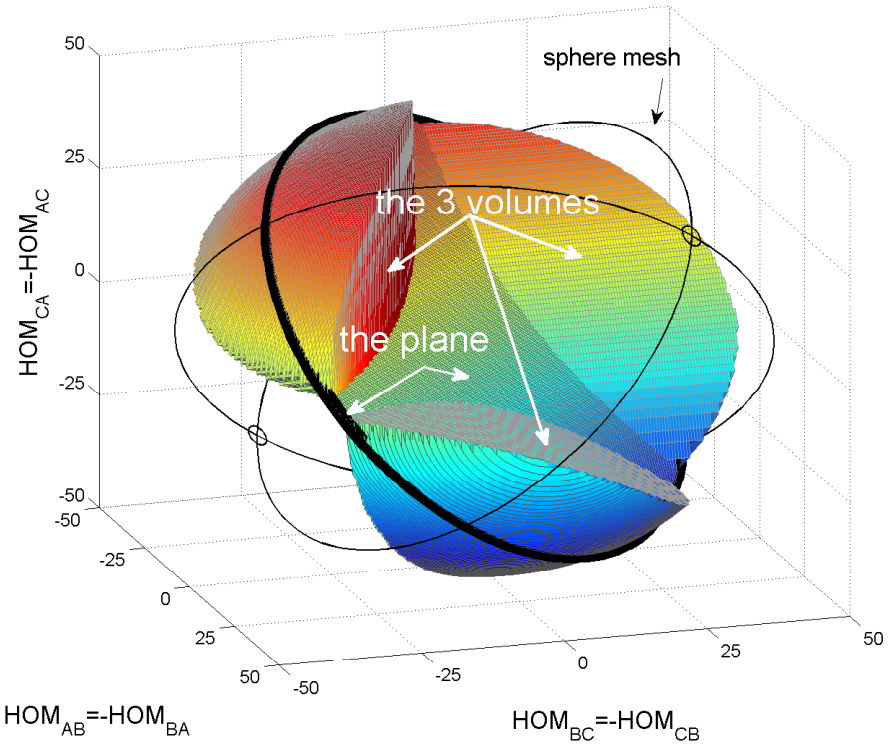


Figure 5.5: CCH free configurations for $a = b = c$ and $\mathbf{HYS}[dB] = \mathbf{0}$

In Fig. 5.5 we represented the domain of $(HOM_{AB},\ HOM_{BC},\ HOM_{CA})$ for which we do not have CCH areas. In this scenario we consider: i) $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ are the vertices of an equilateral triangle ($a = b = c$) and ii) for any $i, j \in \{A, B, C\}$, $i \neq j$ set $HYS_{ij} = 0$ (this implies $HOM_{ij} = -HOM_{ji}$). We represent only the points inside the sphere $HOM_{AB}^2 + HOM_{BC}^2 + HOM_{CA}^2 = 50^2$.

The *sphere mesh* (of radius 50) simply reflects our focus area. The full part of the sphere represents CCH free configurations. To ease the understanding of the figure we identify for the reader that the full part of the sphere contains 3 symmetric *volumes* and a *plane* ($HOM_{AB} +$

HOM_{BA}	$S_\bullet$	$\frac{S_\bullet}{S_{\triangle\mathcal{ABC}}}$	$\frac{S_\bullet}{S_{\bigcirc\mathcal{ABC}}}$
-3	496 m^2	0.45%	0.19%
-6	1993 m^2	1.84%	0.75%
-9	4674 m^2	4.31%	1.78%
-12	9559 m^2	8.83%	3.64%

Table 5.2: CCH area examples

$HOM_{BC} + HOM_{CA} = 0$) that cuts the sphere. One can see that the sphere is actually quite empty, thus we can say that there are a great deal of configurations that create problems. According to Theorem 5.2 this could be avoided by using *per cell* HO parameters instead. You can also identify these configurations in Fig. 5.5 as the previously mentioned *plane* that cuts the sphere.

5.4.2 CCH area quantification

In order to give an idea on how big the problematic area is, we next provide some numerical results. It is too complex to present the results for any network configuration, i.e. $\forall\gamma \in \Gamma$ and $\forall\mathcal{A},\mathcal{B},\mathcal{C} \in \mathcal{D}$. We pick a particular case to ease the reading. Thus we consider the general case using *per-cell-pair* HO parameters where: i) $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ are the vertices of an equilateral triangle, ii) the Inter-Site Distance is 500 m and ii) the pathloss exponent is $\alpha = 4$. Note that the area of triangle $\triangle\mathcal{ABC}$ is $S_{\triangle\mathcal{ABC}} \approx 1.08 \cdot 10^5\ m^2$ and the area of its circumscribed circle is $S_{\bigcirc\mathcal{ABC}} \approx 2.62 \cdot 10^5\ m^2$.

In this example we will only analyse the size of the CCH area in the sense $A \to C \to B \to A$.

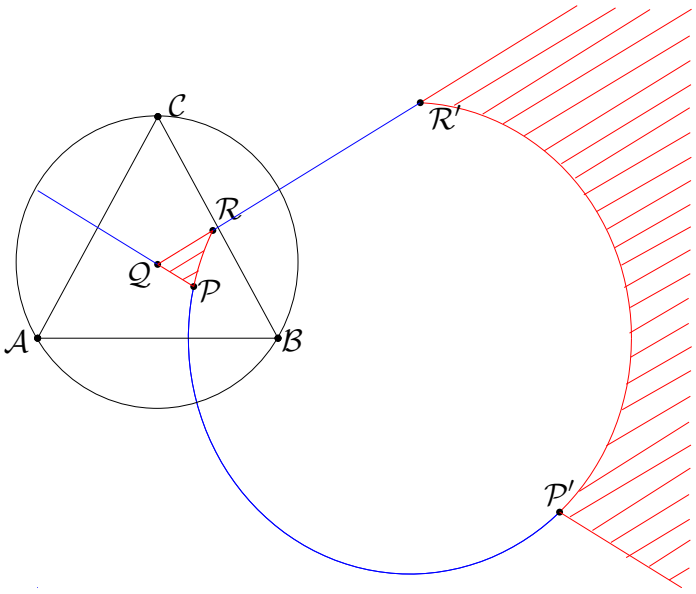
Lets first consider $HOM_{CB} = 0$, $HOM_{AC} = 0$, $HOM_{BA} = x$ (i.e. a variable), and any possible values for HOM_{AB}, HOM_{BC}, HOM_{CA}. For example, in Fig. 5.6a for $HOM_{BA} = -9$ the dashed area represents the area where we will have CCH in the sense $A \to C \to B \to A$. One can see that the dashed area delimited by $\mathcal{P}'$ and $\mathcal{R}'$ will most-likely be covered by other cells. So in the following we limit our focus on the CCH area inside the circumscribed circle of $\triangle\mathcal{ABC}$; we denote the surface of this CCH area by $S_\bullet$.

For various values of $HOM_{BA}[dB] = x$, represented in Fig. 5.6b, one can see in Table 5.2 that as $|HOM_{BA}[dB]|$ increases the problematic area rises close to 10^4m^2.

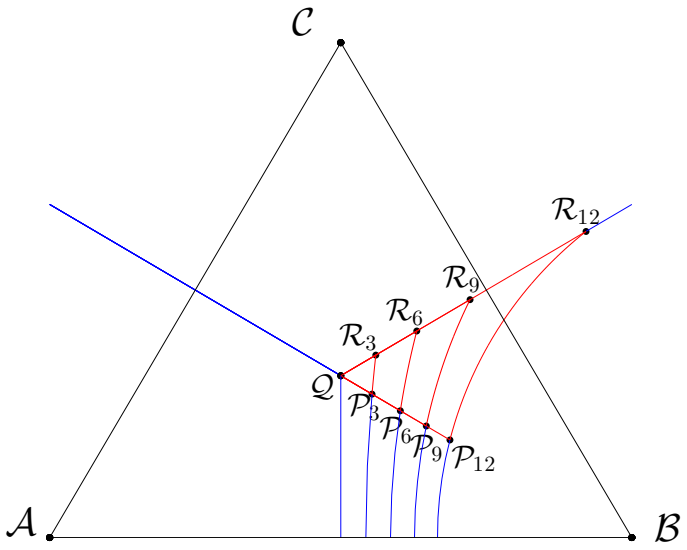
Note that $S_\bullet$, the area corresponding to CCH in the sense $A \to C \to B \to A$, depends on HOM_{BA}, HOM_{AC} and HOM_{CB}. Looking at all the Theorem 5.1 we got the intuition that for Γ_{ACBA} the value of $\rho_{BA} \cdot \rho_{AC} \cdot \rho_{CB}$ may be very important. Implicitly the value of $\Sigma = HOM_{BA} + HOM_{AC} + HOM_{CB}$ is very important. So what we expect is that the value of Σ provides a significant amount of insight on the size of the CCH area. In other words knowing Σ already gives a good idea on how big the CCH area may be, without individually knowing HOM_{BA}, HOM_{AC} and HOM_{CB}. So, in Fig. 5.7 you can find more values of $S_\bullet/S_{\bigcirc\mathcal{ABC}}$ as a function of HOM_{BA} and HOM_{CB} for different values of Σ. Obviously we consider $HOM_{AC} = \Sigma - HOM_{BA} - HOM_{CB}$ is implicit. One can see that when $\Sigma = 0$ ($\rho_{BA}\rho_{AC}\rho_{CB} = 1$) we have a null area and as Σ gets further away from 0 the CCH area increases and can reach 25% of $S_{\bigcirc\mathcal{ABC}}$ ($S_{\bigcirc\mathcal{ABC}}$ comparable to the coverage area of a cell).

5.4.3 Simulation results

To check if the conclusions are still valid in a more realistic scenario, we consider the 3GPP case 1 [54] scenario with 3 cells (Channel model #1 in Appendix 1.1, Table A.1). We make use of a wraparound technique, see Fig. 5.8. We employ a random UE arrival rate based on a Space Poisson Point Process [58]. Each UE stay 3 minutes and then leave the network. During this period we evaluate their number CHs. We set the TTT to 160 ms. To show the problems which occur when using the *per-cell-pair* HO parameters we make use of 6 scenarios as presented in Table 5.3. We note LS_BC and HS_BC are scenarios where $O_{AB} = O_{AB} = O_A$ thus this is



(a) Area identification



(b) Examples (P, R are indexed by $|HOM_{BA}[dB]|$)

Figure 5.6: CCH area evaluation

Figure 5.7: Surface of *CCH area* in [%] w.r.t. $S_{\bigcirc \mathcal{ABC}}$

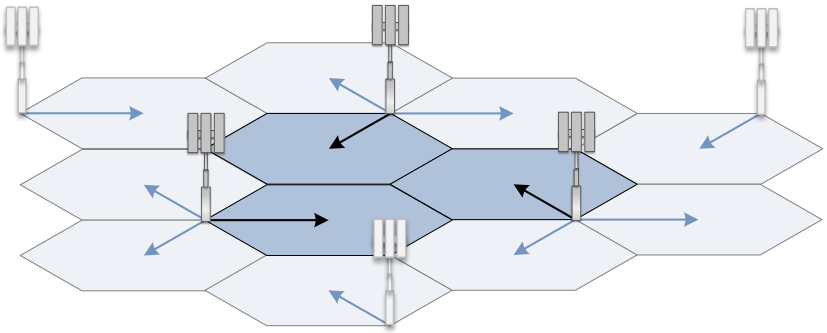


Figure 5.8: Simulation map

equivalent to using *per cell* HO parameters (Remark 5.1).

	LS_B	LS_C	LS_BC	HS_B	HS_C	HS_BC
UE speed	3km/h			30km/h		
O_{AB}	-20dB	0dB	-20dB	-20dB	0dB	-20dB
O_{AC}	0dB	-20dB		0dB	-20dB	
$O_{i,j}, \forall i \in \{B,C\}$, $\forall j \in \{A,B,C\}, i \neq j$	0dB					
$H_{i,j}$, $\forall i,j \in \{A,B,C\}$, $i \neq j$	3dB					

Table 5.3: Scenarios

Figure 5.9a presents the average number of CH per UE and Figure 5.9b presents the percentage of CH (over all HOs). Both figures focus on the total number of CHs, the number of CHs in the sense $A \to B \to C \to A$ and the number of CHs in the sense $A \to C \to B \to A$. One can see that the LS_BC and HS_BC experience only a few CHs. All the other scenarios experience a large number of CHs. One can also see the amount of CHs for each of the 2 senses. This checks the result of Theorem 5.1, i.e. LS_B and HS_B have a high number of CH in the sense $A \to B \to C \to A$ while for LS_C and HS_C the predominant CHs are in the other sense.

(a) Number of CH per UE

(b) Percentage of CH (over all HOs)

Figure 5.9: CH evaluations

5.5 Concluding remarks

In this work we focus on potential risks of using *per-cell-pair* HO parameters (offset and hysteresis) as compared to using *per-cell* HO parameters. Although we have a bigger manoeuvrability, most likely it leads to the creation of CCH areas. We have provided the necessary and sufficient conditions that create these CCH areas for the case of using *per-cell-pair* HO parameters. Results of Theorem 5.1 with an example in Fig. 5.5 show that the conditions to have a CCH free configuration are quite restrictive. We also showed that for the case of *per-cell* HO parameters the risk of having CCH areas is null. We have validated our results through simulations for different UE movement speeds. Therefore we conclude that using *per-cell* HO parameters is safer than using *per-cell-pair* HO parameters when considering the CCH areas, however it provides less manoeuvrability since we cannot individually tune the HO parameters w.r.t. each neighbouring cell. Future work can focus on a deeper analysts of the pros and cons of using *per-cell-pair* HO parameters.

Chapter 6

Conclusions and future work

Self Organizing Networks (SON) is currently one of the main challenges for mobile networks along with Internet of Things (IoT), Software Defined Networks (SDN), Network Function Visualization (NFV), Cloud-RAN (C-RAN), etc. SON functions are meant to automate the network tuning. Thus the network operators can replace the costly human intervention for repetitive tasks and can significantly improve their reactivity and the Quality of Service (QoS) offered to their end-users.

A network operator might choose to purchase several SON functions depending on its targets in terms of end-user experience, and could even prefer a multi-vendor solution. Thus, a SON management framework is required. This framework should insure that SON functions can be bought from different vendors and when put together they still offer good network Key Performance Indicators (KPIs). In this spirit we have analysed potential conflicts which may occur when using several independently designed SON functions. Moreover, SON vendors will most-likely provide very limited information on the algorithms inside the SON functions. Thus in our approach we have also considered the SON functions as *black-boxes*, i.e. the algorithm inside is unknown. To deal with the SON conflicts 3GPP has proposed the use of a SON COordinator (SONCO). The purpose of this thesis is to propose some SONCO designs and analyse their performance.

In **Chapter 3** we show how how these potential conflicts can be identified. Then we provide a methodology for diagnosing which of the potential conflicts are active conflicts, i.e. they actually degrade the network KPIs (SONCO-D). For this we have introduce a framework based on the Naive Bayesian Classifier (NBC). Results show that we are able to identify the root of our conflicts, i.e. the SON whose settings makes it impossible for other SON functions to achieve their target KPIs. Going further into more details, we also try to identify which of the SON function's setting is responsible for the bad KPIs. Obviously, the further we go into details the less accurate the results will be. Overall, NBC has proved to be a good candidate for SON conflict diagnosis.

After identifying the active conflicts in the network the next rational step is to deal with them. One way of doing this is to try to eliminate the conflict by reconfiguring the SON functions, worst case scenario turn some SON functions off. Another solution is to enforce a conflict resolution mechanism (SONCO-R). In **Chapter 4** we provide such a mechanism design. Basically the SONCO-R receives all the update requests form the SON instances and it has to decide which of them to accept and which to reject, ensuring some degree of fairness among the SON instances. We make use of Reinforcement Learning (RL) as it allows to make use of past information to improve the future decisions. We have analyse several RL designs showing how to make the RL algorithm scalable using two linear function approximation flavours: *state aggregation* and *linear features*. We also study the performance costs for each of these. Results show that *state aggregation* is a good candidate for reducing the RL complexity. The use of *linear features* reduces even more the complexity, but it has be carefully done as the approximation might be to imprecise.

In all our studies we use *per-cell* HO parameters. In the literature we have also come across a more general case that uses *per-cell-pair* HO parameters. In **Chapter 5** we motivate our choice.

We show that using *per-cell-pair* HO parameters can lead to creating areas of Continuous Chained HandOvers (CCH). On the other hand using *per-cell* HO parameters eliminates this risk, but limits the manoeuvrability of the user mobility.

Future work There is still a lot of work to be done on the SON coordination. Now that we have got a taste of what are the challenges of coordinating the SON functions, one of the next steps should be to improve the SON function concept. The SON functions should evolve into facilitating their coordination. For example, a SON function may better understand the environment, acknowledge the SON functions running in parallel, foresee how they can be impacted by them and even create alarms if it considers that its performance is deteriorated because this.

The algorithms of the SON coordinator should consequently evolve. They should be ready to deal with smarter SON functions.

Other machine learning algorithms could be tested for the SON conflict resolution function. A fully distributed algorithm could improve the convergence time. Some approximations on the Naive Bayesian Classifier could be tested for the SON conflict detection. Also, one could also evaluate other classifiers: decision trees, neural networks, etc. Another perspective would be to create a meta control loop to optimize the SON settings via high level policies.

One shortcoming of the current work is the lack of real data. SON functions are not yet widely deployed. Once this happens, it is interesting to test the algorithms with real data and update them accordingly.

Appendix A

Simulator details

1.1 Channel Model

Modelling the radio channel comes down to estimating power received by the UEs from the network cells. Let $\overline{RxPow}_n^{\kappa}(t)$ be the power received by UE κ from cell n at time t expressed in dBs. We model it as follows:

$$\overline{RxPow}_n^{\kappa}(t) = TxPow_n(t) - FL_n + AG_n + AP_n^{\kappa}(t) + PG_n^{\kappa}(t) + Sh_n^{\kappa}(t) + Ff_n^{\kappa}(t) - PtL_n \quad \text{(A.1)}$$

where $TxPow_n(t)$ is the transmission power, FL_n is the feeder loss, AG_n^{κ} is the antenna gain, $AP_n^{\kappa}(t)$ is the antenna pattern, $PG_n^{\kappa}(t)$ is the path gain, $Sh_n^{\kappa}(t)$ is the shadowing effect, $Ff_n^{\kappa}(t)$ is the fast fading effect and PtL_n is the penetration loss.. For all the components the bases are the 3GPP TS 363.814 [54] (Case 1 and 3) and the ITU-R M2134 [55] (Urban Macro) considering also the extensions for Heterogeneous Networks (HetNets), specifically Pico cells. The details for the channel models available in our simulator are presented later on in this section.

We define the filtered signal, which is to be used for HO purposes, as follows:

$$RxPow_n^{\kappa}(t) = FtC \cdot \overline{RxPow}_n^{\kappa}(t) + (1 - FtC) \cdot RxPow_n^{\kappa}(t-1) \quad \text{(A.2)}$$

where FtC is the filter coefficient [104].

We employ a hexagonal topology (see Fig. A.1) subject to a wraparound and consider 8 radio channel models as summarized in Table A.1.

We benchmark the channel models #0 and #1 (Table A.1) w.r.t. ITU-R UMa and 3GPP Case 1, respectively. We consider a 2 tiers topology, see Fig. A.1. In Fig. A.2a we plot the c.d.f. of the signal attenuation (*RxPow-TxPow*). In Fig. A.2b we plot the c.d.f. of the geometric SINR (all cells transmit with maximum power). The fast fading is disregarded for both figures.

ID	LOS/NLOS distinction		Simulator reference
	Macro	Pico	
#0	Yes	-	CM_ITU_UMa
#1	No	-	CM_3GPP_case1
#11		No	CM_3GPP_case1_wPico_model1
#12		Yes	CM_3GPP_case1_wPico_model2
#2		-	CM_3GPP_case3
#21		No	CM_3GPP_case3_wPico_model1
#22		Yes	CM_3GPP_case3_wPico_model2S
#23		Yes	CM_3GPP_case3_wPico_model2RS

Table A.1: Radio channel models

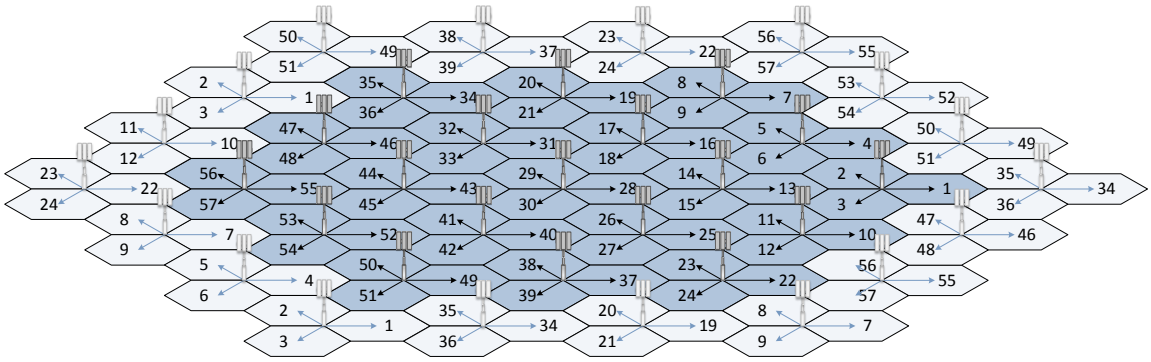
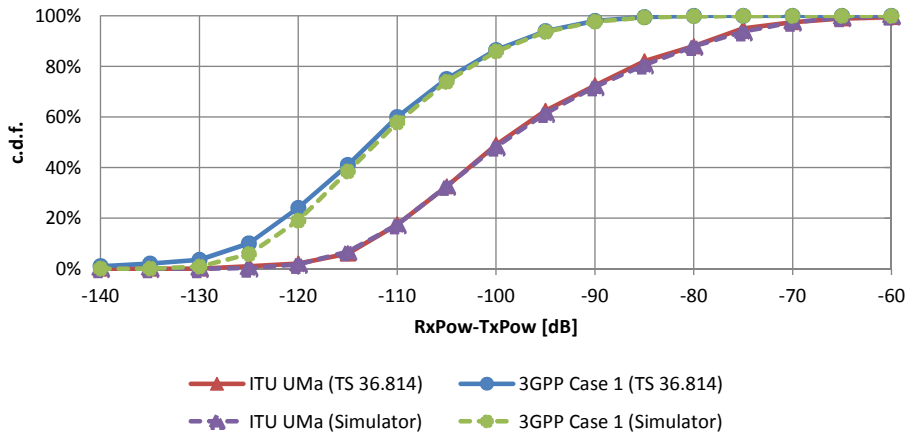
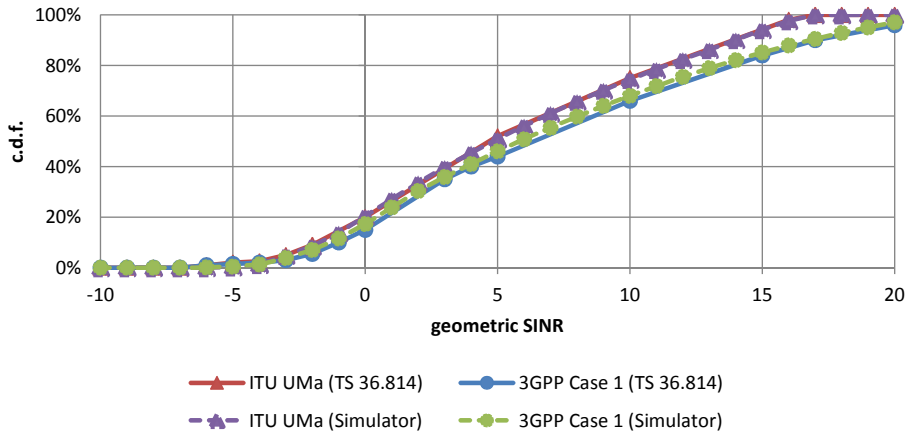


Figure A.1: Map Wrap Around: 2 tiers



(a) Benchmark *RxPow-TxPow* (no fast fading)



(b) Benchmark geometric SINR (no fast fading)

Figure A.2: Benchmark

The Inter Site Distance (ISD) values are summarized in Table A.2.

The LOS probability is detailed in Table A.3.

The PathLoss details are presented in Table A.4.

ISD	#0	#1	#11	#12	#2	#21	#22	#23
Macro to Macro	500.0 m				1732.0 m			
Macro to Pico	-		>75m		-	>75m		
Pico to Pico	-		>40m		-	>40m		

Table A.2: Inter Site Distance

	LOS probability
#0	Macro: $\min\{18.0/d_H, 1.0\}(1.0 - \exp(-d_H/63)) + exp(-d_H/63)$
#1	0
#11	0
#12	Macro: $\min\{18.0/d_H, 1.0\}(1.0 - \exp(-d_H/63)) + \exp(-d_H/63)$ Pico:$0.5 - \min\{0.5; 5.0\exp(-156.0/d_H)\} + \min\{0.5; 5.0\exp(-d_H/30)\}$
#2	0
#21	0
#22	Macro:$\exp(-(d_H - 10.0)/200.0)$ Pico: $0.5 - \min\{0.5, 3.0 * \exp(-300.0/d_H)\} + \min\{0.5, 3.0\exp(-d_H/95.0)\}$
#23	Macro: $exp(-(d_H - 10)/1000.0)$ Pico: $0.5 - \min\{0.5, 3.0\exp(-300.0/d_H)\} + \min\{0.5, 3.0\exp(-d_H/95.0)\}$

d_H is the horizontal distance between the UE and the Cell

Table A.3: LOS/NLOS probability

	Pathloss
#0	$\begin{cases} 34.0206 + 22.0\log_{10}(d_H) & \text{if } LOS \text{ and } d_H < 320 \\ -11.0232 + 40.0\log_{10}(d_H) & \text{if } LOS \text{ and } d_H \geq 320 \\ 19.5653 + 39.1\log_{10}(d_H) & \text{if } NLOS \end{cases}$
#1 #2	$128.1 + 37.6\log_{10}(d_H/1000)$
#11 #21	Macro: $128.1 + 37.6\log_{10}(d_H/1000)$ Pico: $140.7 + 36.7\log_{10}(d_H/1000)$
#12 #22 #23	Macro:$\begin{cases} 103.4 + 24.2\log_{10}(d_H/1000) & \text{if } LOS \\ 131.1 + 42.8\log_{10}(d_H/1000) & \text{if } NLOS \end{cases}$ Pico: $\begin{cases} 103.8 + 20.9\log_{10}(d_H/1000) & \text{if } LOS \\ 145.4 + 37.5\log_{10}(d_H/1000) & \text{if } NLOS \end{cases}$

d_H is the horizontal distance between the UE and the Cell

Table A.4: Pathloss

	#0	#1	#2	#11, #12	#21, #22, #23
Feeder Loss	2[dB]	0[dB]		0[dB]	

Table A.5: Feeder loss

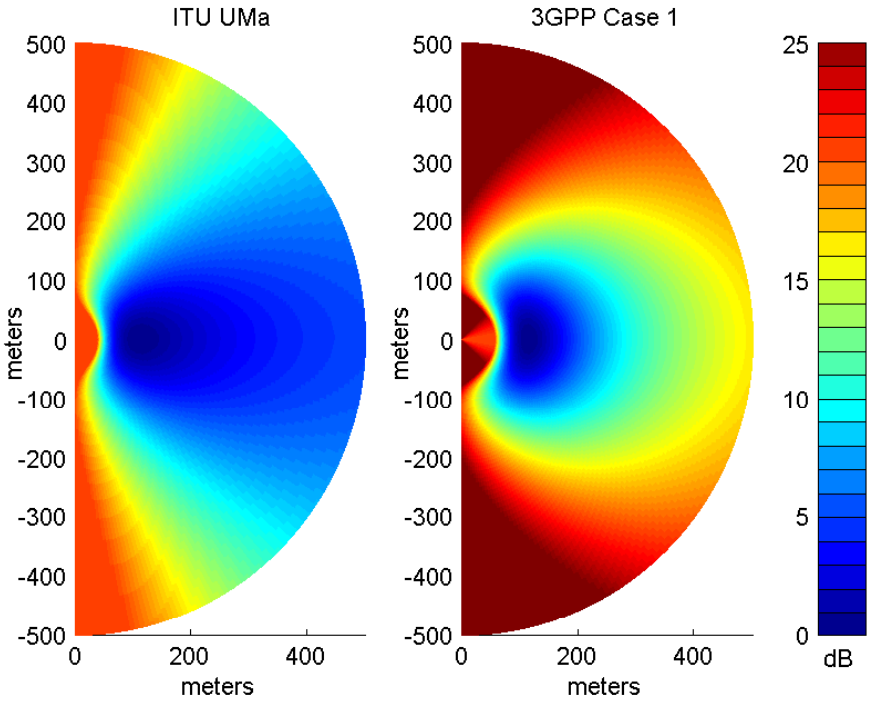


Figure A.3: Antenna pattern #0 (ITU UMa) and #1 (3GPP Case 1)

The feeder loss values are summarized in Table A.5.

The antenna gain include: antenna gain, antenna directivity horizontal and vertical. We also present the feeder loss in this section. The details are summarized in Table A.6. The Macro cells use directive antennas (3-sectorial sites). The Pico cells use omnidirectional antennas. In Fig. A.3 we plot the antenna patter for scenario #0 (ITU-UMa).

The penetration loss is established per UE. We define the mean and the standard deviation in Table A.7.

	#0	#1	#2	#11, #12	#21, #22, #23
Antenna Gain	17[dB]	14[dB]	14[dB]	Macro:14[dB], Pico: 5[dB]	
Antenna Pattern H	$A_H(\varphi) = -\min\left[12\left(\frac{\varphi}{\varphi_{3dB}}\right)^2, A_m\right]$ dB.$\varphi_{3dB} = 70°$				
A_m	20	25		Macro: 25	
Antenna Pattern V	$A_V(\theta) = -\min\left[12\left(\frac{\theta-\theta_{tilt}}{\theta_{3dB}}\right)^2, SLA_v\right]$ dB.$SLA_v = 20$ dB				
θ_{3dB}	15°	10°		Macro: 10°	
Antenna Tilt	12°	15°	6°	Macro: 15°	Macro: 6°
Antenna Pattern 3D	Macro:$A(\varphi,\theta) = -min\{-[A_H(\varphi) + A_V(\theta)], A_m\}$dB Pico: 0dB				

Table A.6: Antenna characteristics

Penetration Loss	#0	#1	#2	#11, #12	#21, #22, #23
Mean	Outdoor-to-Incar: 9[dB]	Outdoor-to-Indoor: 20[dB] Outdoor-to-Outdoor: 0[dB]			
Std. Dev.	5[dB]	0[dB]			

Table A.7: Penetration loss

Shadowing	#0	#1	#2	#11, #12	#21, #22, #23
Std. Dev.	LOS: 4[dB] NLOS:6[dB]	8[dB]		Macro: 8[dB] Pico: 10[dB]	Macro: 8[dB] Pico: $\begin{cases} 3\text{[dB]} & \text{if LOS} \\ 4\text{[dB]} & \text{if NLOS} \end{cases}$
Corr. Dist.	LOS: 37[m] NLOS:50[m]	50[m]		Macro: 50[m], Pico: $\begin{cases} 10\text{[m]} & \text{if LOS} \\ 13\text{[m]} & \text{if NLOS} \end{cases}$	

Table A.8: Shadowing

The shadowing is established per UE w.r.t. every site. We define the standard deviation and the correlation distance in Table A.8. As the UEs are moving we update the shadowing (w.r.t. all cells) at every time instance. Each update is correlated with the previous.

The correlation is implemented for the shadowing of UE κ w.r.t. site n as follows: First let the shadowing correlation coefficient be:

$$corrSh_{\kappa}(t) = \exp\left(-\left|d_{(\ell_{\kappa}(t),\ell_{\kappa}(t-1))}\right| / corrDistSh_n^{\kappa}(t)\right) \tag{A.3}$$

where $corrDistSh_n^{\kappa}(t)$ is the correlation distance established for site n and UE κ (i.e. the correlation distance depends on the location of UE κ) according to Table A.8. We remind that $\ell_{\kappa}(t) \in \mathbb{R}^2$ is the location of UE κ at time t and $d_{(\ell_1,\ell_2)}$ is the distance between ℓ_1 and ℓ_2 ($\ell_1, \ell_2 \in \mathbb{R}^2$). The shadowing correlations is performed as follows:

$$Sh_n^{\kappa}(t) = \sqrt{(corrSh_n^{\kappa}(t))^2}Sh_n^{\kappa}(t-1) + \sqrt{1-(corrSh_n^{\kappa}(t))^2}Y_n^{\kappa}(t) \tag{A.4}$$

where $Sh_n^{\kappa}(t)$ is the shadowing at time t experienced by UE κ w.r.t. site n in [dB], $Y_n^{\kappa}(t)$is a random variable (normal distribution) with mean 0 and standard deviation $stdDevSh_n^{\kappa}(t)$ (i.e. the standard deviation depends on the location of UE κ).

We check the mean as follows:

$$\begin{aligned}
&\mathbb{E}\left[Sh_n^{\kappa}(t)\right] \\
&= \mathbb{E}\left[corrSh_n^{\kappa}(t)Sh_n^{\kappa}(t-1) + \sqrt{1-(corrSh_n^{\kappa}(t))^2}Y_n^{\kappa}(t)\right] \\
&= \mathbb{E}\left[corrSh_n^{\kappa}(t)Sh_n^{\kappa}(t-1)\right] + \mathbb{E}\left[\sqrt{1-(corrSh_n^{\kappa}(t))^2}Y_n^{\kappa}(t)\right] \\
&= corrSh_n^{\kappa}(t)\mathbb{E}\left[Sh_n^{\kappa}(t-1)\right] + \sqrt{1-(corrSh_n^{\kappa}(t))^2}\mathbb{E}\left[Y_n^{\kappa}(t)\right] \\
&= 0
\end{aligned} \tag{A.5}$$

We check the standard deviation as follows:

$$\begin{aligned}
&\mathbb{E}\left[Sh_n^\kappa(t)\,Sh_n^\kappa(t)\right] \\
&= \mathbb{E}\left[\left(corrSh_n^\kappa(t)\,Sh_n^\kappa(t-1) + \sqrt{1-(corrSh_n^\kappa(t))^2}Y_n^\kappa(t)\right)^2\right] \\
&= \mathbb{E}\left[(corrSh_n^\kappa(t)\,Sh_n^\kappa(t-1))^2\right] + \mathbb{E}\left[\left(\sqrt{1-(corrSh_n^\kappa(t))^2}Y_n^\kappa(t)\right)^2\right] \\
&\qquad + 2\mathbb{E}\left[corrSh_n^\kappa(t)\sqrt{1-(corrSh_n^\kappa(t))^2}Sh_n^\kappa(t-1)\,Y_n^\kappa(t)\right] \\
&= (corrSh_n^\kappa(t))^2\,\mathbb{E}\left[(Sh_n^\kappa(t-1))^2\right] + \left(1-(corrSh_n^\kappa(t))^2\right)\mathbb{E}\left[(Y_n^\kappa(t))^2\right] \\
&\overset{t\to\infty}{=} (stdDevSh_n^\kappa(t))^2
\end{aligned} \tag{A.6}$$

where the last equality holds if UE κ does not move (i.e. $corrSh_n^\kappa(t)$ and $stdDevSh_n^\kappa(t)$ are constant).

We check the correlation as follows:

$$\begin{aligned}
&\mathbb{E}\left[Sh(t)\,Sh(t-1)\right] \\
&= \mathbb{E}\left[\left(corrSh_n^\kappa(t)\,Sh_n^\kappa(t-1) + \sqrt{1-(corrSh_n^\kappa(t))^2}Y_n^\kappa(t)\right)Sh_n^\kappa(t-1)\right] \\
&= \mathbb{E}\left[corrSh_n^\kappa(t)\,(Sh_n^\kappa(t-1))^2\right] + \mathbb{E}\left[\sqrt{1-(corrSh_n^\kappa(t))^2}Y_n^\kappa(t)\,Sh_n^\kappa(t-1)\right] \\
&= corrSh_n^\kappa(t)\,\mathbb{E}\left[(Sh_n^\kappa(t-1))^2\right] + \sqrt{1-(corrSh_n^\kappa(t))^2}\mathbb{E}\left[Y_n^\kappa(t)\,Sh_n^\kappa(t)\right] \\
&= corrSh_n^\kappa(t)\,\mathbb{E}\left[(Sh_n^\kappa(t-1))^2\right]
\end{aligned} \tag{A.7}$$

The fast fading has a Rayleigh distribution, i.e. square root of the sum of the squares of two variables of normal distribution. The fast fading is updated at each time step. As in the shadowing case we correlate the current sample with the one from the previous time step. Let $\bar{Ff}_n^\kappa(t)$ be the fast fading (not in dB!) at time t corresponding to cell n and UE κ: $Ff_n^\kappa(t) = 10\log_{10}\bar{Ff}_n^\kappa(t)$. Let $\bar{Ff}1_n^\kappa(t)$ and $\bar{Ff}2_n^\kappa(t)$ be the two variables with normal distribution that compose $\bar{Ff}_n^\kappa(t)$:

$$\bar{Ff}_n^\kappa(t) = \sqrt{\left(\bar{Ff}1_n^\kappa(t)\right)^2 + \left(\bar{Ff}2_n^\kappa(t)\right)^2} \tag{A.8}$$

We apply the correlation on the two components as follows:

$$\begin{aligned}
\bar{Ff}1_n^\kappa(t) &= corrFF_\kappa(t)\,\bar{Ff}1_n^\kappa(t-1) + \sqrt{1-(corrFf_\kappa(t))^2}Y1_n^\kappa(t)\sqrt{2\pi^{-1}} \\
\bar{Ff}2_n^\kappa(t) &= corrFf_\kappa(t)\,\bar{Ff}2_n^\kappa(t-1) + \sqrt{1-(corrFf_\kappa(t))^2}Y2_n^\kappa(t)\sqrt{2\pi^{-1}}
\end{aligned} \tag{A.9}$$

where $corrFf_\kappa(t) = \exp\left(-2\left|d_{(\ell_\kappa(t),\ell_\kappa(t-1))}\right| f_c/speedOfLight\right)$ is the correlation coefficient, f_c is the carrier frequency, $Y1_n^\kappa$ and $Y2_n^\kappa$ are random variables with normal distribution of mean 0 and standard deviation 1 ($\mathcal{N}(0,1)$), $\sqrt{2\pi^{-1}}$ is a correction to bring the average back to 1 and $speedOfLight = 3\cdot 10^8 m/s$. We remind $\ell_\kappa(t) \in \mathbb{R}^2$ is the location of UE κ at time t and $d_{(\ell_1,\ell_2)}$ is the distance between ℓ_1 and ℓ_2 (ℓ_1, $\ell_2 \in \mathbb{R}^2$). The mean and correlation can be checked in a similar manner as for the shadowing.

The thermal noise is -174.0 dBm.

The noise figure values are presented in Table A.9.

The carrier frequency is 2GHz.

	#0	#1, #2, #11, #12, #21, #22, #23
Noise Figure	7[dB]	9[dB]

Table A.9: Noise Figure

Bw.	1.4 [MHz]	3 [MHz]	5 [MHz]	10 [MHz]	15 [MHz]	20 [MHz]
PRBs	6	15	25	50	75	100
REs	76	181	301	601	901	1201

Table A.10: Bandwidth

The bandwidth can be selected as presented in Table A.10.

1.2 UE procedures

While in the network the UEs trough several procedures: *Connection Establishment*, *Connection ReEstablishment*, *Connection Reconfiguration* (HO) and *Receive Data* (see Fig. 2.2). The procedures basically consist of a certain message exchange sequence. The messages correspond to two a protocol stacks according to the control plane and user plane (see Figure A.4).

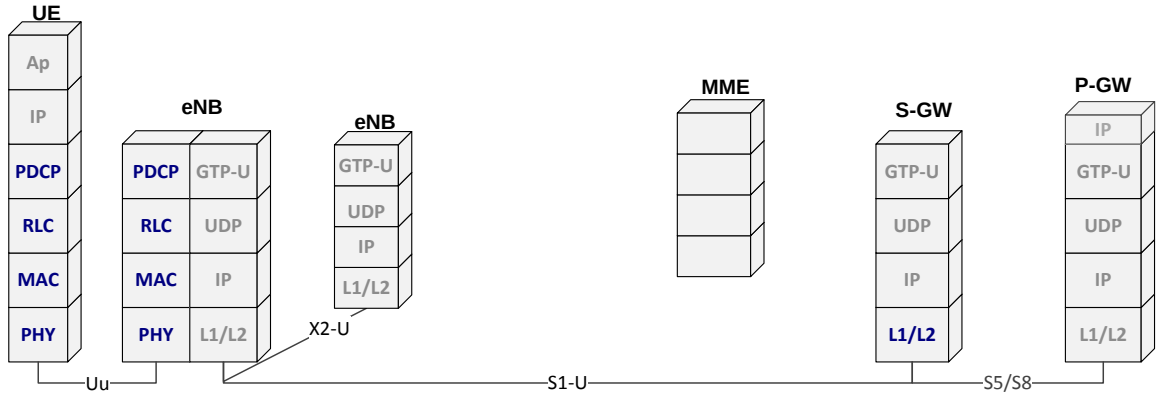
In our simulator we only model the the messages exchanged between the eNB and the UE on the Media Access Control (MAC) layer, the Radio Link Control (RLC) layer, and the Radio Resource Control (RRC) layer. For more details we refer the reader to [9].

In the implementation we define one UE state for each of these messages as presented in Figures A.5, A.6, A.7, A.8, A.9. For more details we refer the reader to [5]. To ease the understanding of the figures we provide a short legend:

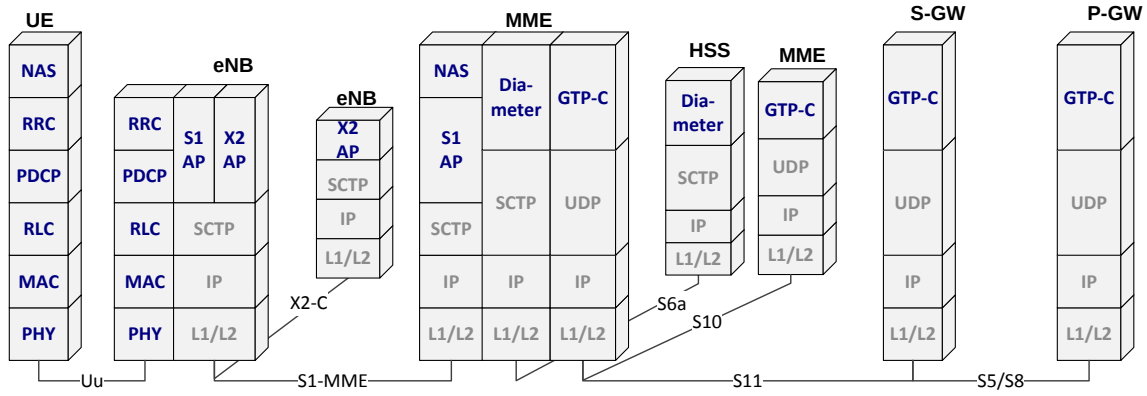
- the grey square boxes represent the states
- the text colour indicates the layer as follows: red is RRC (the RRC timers), green is RLC and blue is MAC. See Tables A.11 and A.12.
- the message prefix meaning is defined from a UE perspective as follows: "s_" is for a sent message, "w_" is for waiting for the message, "r_" is for received message, "pr_" is for process received message (i.e. message has been receive correctly) and "fr_" is for failed to receive.

1.3 Almost Blank Sub-frames

In order to reduce the interference towards the neighbouring cells some cells might be forbidden to transmit at certain time instances. To better understand this we present in Fig. A.10 the structure of an LTE downlink frame (for 2 transmission antennas). Thus, a *frame* has a length of 10 ms. It is composed of 10 *sub-frames* of 1 ms each. A sub-frame contains 2 *slots* of 0.5 ms each. A slot contains 6-7 so called Orthogonal Frequency-Division Multiplexing (OFDM) symbols, which can be seen as *batches* of data. The OFDM symbols can be further divided into Resource Elements (REs). In Fig. A.10 one can see the employment of the REs and how they are grouped to form Resource Blocks (RBs). The RBs can be then allocated to different UEs. For more details we refer the reader to [9]. In our simulator, all the RBs of one *sub-frame* are allocated only to one user. This is reasonable for a FTP-like traffic, which is the case in our simulator. We assume that all the cells are synchronized on a frame basis. A sub-frame where no user data is transmitted is called an Almost Blank Sub-frame (ABS).



(a) Control plane protocol stack [105]



(b) User plane protocol stack [105]

Figure A.4: LTE - SAE protocol stack

Figure A.5: UE states Connection Establishment

Figure A.6: UE states Connection Reconfiguration HO

Figure A.7: UE states Connection ReEstablishment

Figure A.8: UE states Physical and MAC Connection

Figure A.9: UE states Receive Data

Parameter		Simulator	Standard Values TS 36.331
RRC			
T300	[ms]	100	100, 200, 300, 400, 600, 1k, 1.5k, 2k
T301	[ms]	100	100, 200, 300, 400, 600, 1k, 1.5k, 2k
T304	[ms]	100	100, 200, 400, 500, 1k, 2k, 4k, 8k
T311	[ms]	1000	(1k), 3k, 5k, 1k, 1.5k, 20k, 30k
T311a	[ms]	200	parameter not standardized
T310	[ms]	1000	0, 50, 100, 200, 500, (1k), 2k
N310	[#]	1	(1), 2, 3, 4, 6, 8, 10, 20
N311	[#]	1	(1), 2, 3, 4, 5, 6, 8, 10
T_N310	[ms]	= N310×200	parameter not standardized
T_N311	[ms]	= N311×100	parameter not standardized
TTT	[ms]	100	0, 40, 64, 80, 100, 128, 160, 256, 320, 480, 512, 640, 1024, 1280, 2560, 5120

Table A.11: Timers

Parameter		Simulator	Standard Values TS 36.331
MAC			
preambleTransMax	[#]	3	3, 4, 5, 6, 7, 8, 10, 20, 50, 100, 200
re_ResponseWindowSize	[#]	2	2, 3, 4, 5, 6, 7, 8, 9, 10
BackoffIndicator	[sf]	10	0, 10, 20, 30, 40, 60, 80, 120, 160, 240, 320, 480
~~numberOfRA_Preambles~~	~~[#]~~	Not used	2, 8, 12, 16 64
~~sizeOfRA_PreambleGroupA~~	~~[#]~~	Not used	
maxHARQ_Tx	[#]	5	1, 2, 3, 4, (5), 6, 7, 8, 9, 10, 12, 16, 20, 24, 28
~~maxHARQ_Msg3Tx~~	~~[#]~~	Not used	1, 2 ,3 ,4, 5, 6, 7, 8
HARQ RTT Timer (ReTx)	[sf]	8	≥ 8 (TS 36.321)
RLC			
maxRetxThreshold	[#]	4	1, 2, 3, (4), 6, 8, 16, 32
t_PollRetransmit	[ms]	45	5, 10, 15, ... (45),,,250, 300, 350, 400, 450, 500
RRC			
filterCoefficientRSRP	[ms]	4	0, 1, 2, 3, (4), 5, 6, 7, 8, 9, 11, 13, 15, 17, 19,
filterCoefficientRSRQ	[ms]	4	
cellIndividualOffset	[dB]		Q-OffsetRange
Q-OffsetRange	[dB]		-24, -22, -20, -18, -16, -14, -12, -10, -8, -6, -5, -4, -3, -2, -1, (0), 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24

Table A.12: Parameters

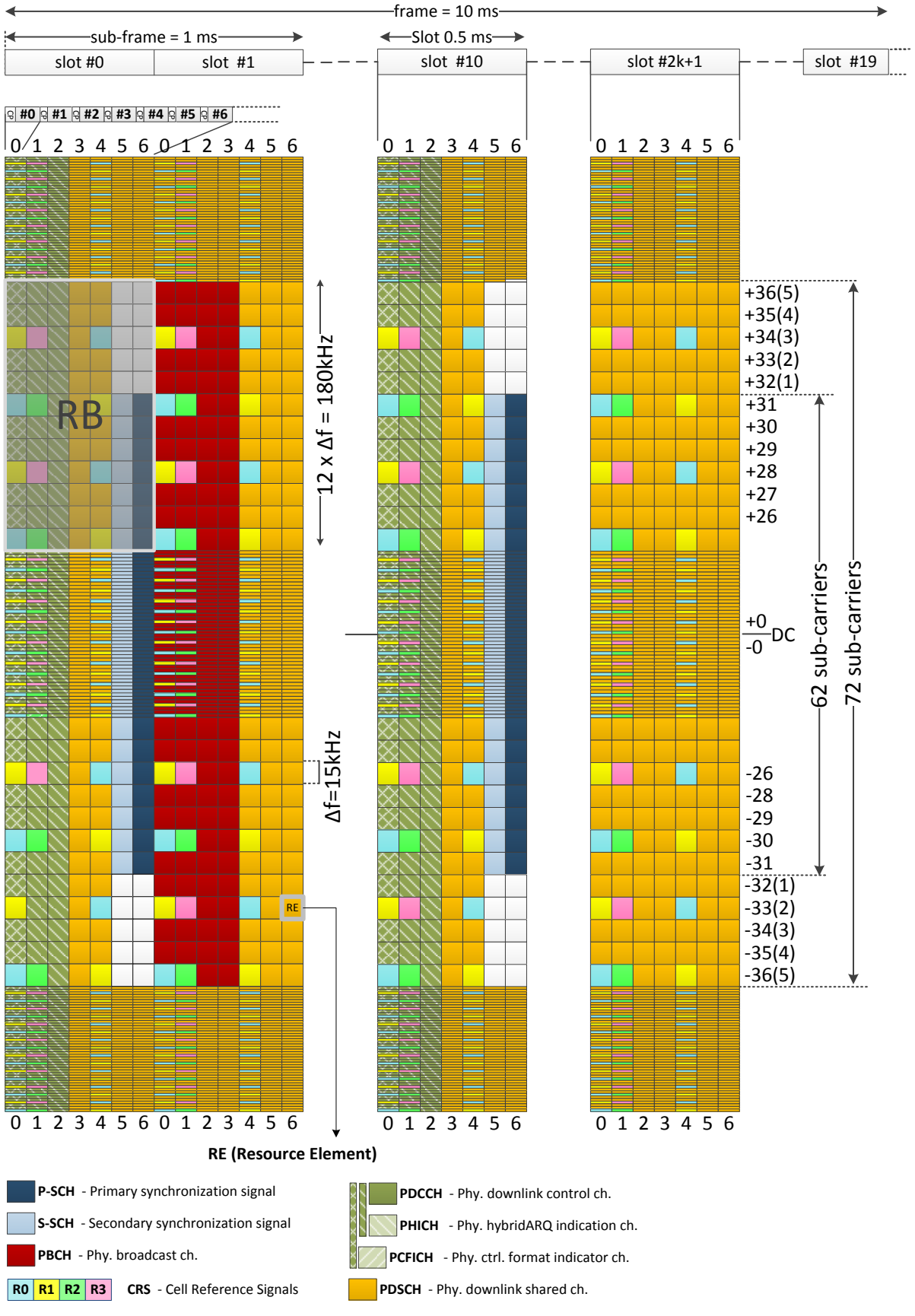


Figure A.10: LTE frame

Appendix B

SON functions for Macro Network scenarios

The SON functions are defined in this appendix considering N Macro cells (indexed by $n \in \{1, .., N\}$).

2.1 MLB for a Macro Network

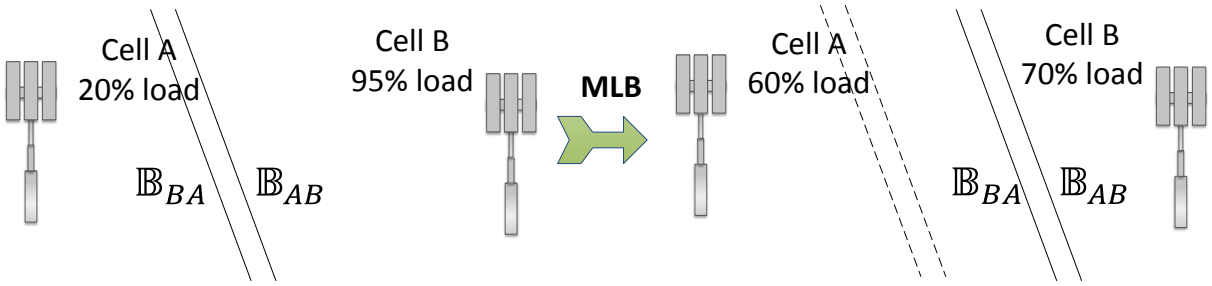


Figure B.1: MLB Function

We consider one independent MLB instance per cell. The MLB instance targets to reduce the load of the host cell, see Fig. B.1. The tuned parameter is the CIO of the hosting cell.

Given some time instance (we disregard the time index) we define:

- $M_{n,LD}$ the load of cell n calculated as the average number of occupied Physical Resource Blocks (PRBs, [106]) of the hosting cell (the cell on which the MLB instance runs)
- $\mathbb{T}^H_{LD}$ and $\mathbb{T}^L_{LD}$ are triggering thresholds
- φ is the logistic function:

$$\varphi\left(mid, sloap, x\right) = \frac{1}{1 + \exp^{-sloap(x-mid)}} \tag{B.1}$$

For cell n, the update request of the MLB instance at time t for the CIO can be expressed as:

$$U^{MLB}_{n,CIO} = \begin{cases} -\varphi\left(0.9, 30, M_{n,LD}\right), & \text{if } M_{n,LD} > \mathbb{T}^H_{LD} \\ 1 - \varphi\left(-0.2, 5, M_{n,LD}\right), & \text{if } M_{n,LD} < \mathbb{T}^L_{LD} \\ 0, & \text{otherwise} \end{cases} \tag{B.2}$$

2.2 MRO for a Macro Network

We consider a distributed implementation of the MRO where the MRO instances running on neighbouring cells communicate with each other. The target of the MRO instances is to reduce

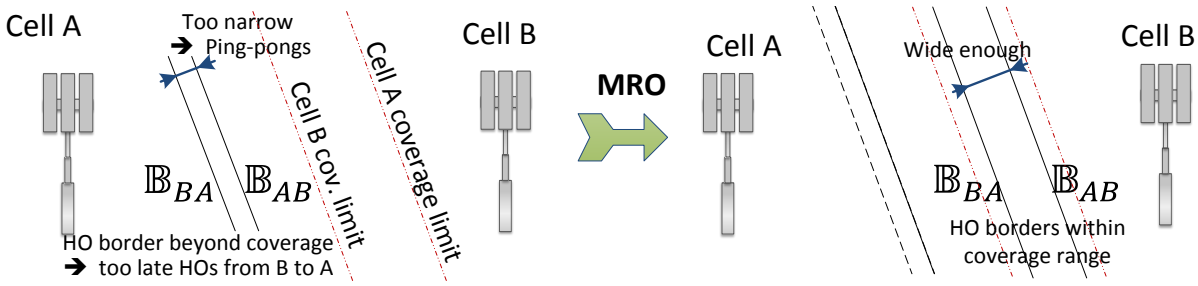


Figure B.2: MRO Function

the no. of ping-pongs, the no. of too late HOs, the no. of too early HOs and the number of HOs to wrong cells. We only focus on the first two, as the last two are considerably less frequent. To do this MRO tunes mainly the HYS (see Fig. B.2). Increasing the HYS should reduce the no. of ping-pongs. Reducing the HYS should reduce the no. of too late HOs. But when we want to do both we turn to the CIO. Reducing the CIO of one cell should reduce the no. of too late HOs originating from that cell, without affecting the no. of ping-pongs.

Given some time instance (we disregard the time index) we define:

- $M_{n,PP}$ the number of HO ping-pongs originating from cell n and $M_{n,TL}$ the number of too late HOs also originating from cell n,
- $M_{n,TL2}$ the number of too late HOs where cell n is the HO target cell.
- $\mathbb{T}^H_{TL}$, $\mathbb{T}^H_{PP}$, $\mathbb{T}^L_{TL} = 0.25\mathbb{T}^H_{TL}$, $\mathbb{T}^L_{PP} = 0.25\mathbb{T}^H_{PP}$ and $\mathbb{T}^L_{TL2}$ are triggering thresholds
- $\varphi\left(mid, sloap, x\right)$ is the logistic function, see eq. (B.1)

Since the MRO tunes two parameters, the CIO and the Hysteresis, the MRO instance of cell n sends 2 simultaneous requests: one for each parameter. The one regarding the Hysteresis can be expressed as follows:

$$U^{MRO}_{n,HYS} = \begin{cases} -1, & \text{if } M_{n,TL} > \mathbb{T}^H_{TL} \text{ and } M_{n,PP} < \mathbb{T}^L_{PP} \\ 1, & \text{if } M_{n,PP} > \mathbb{T}^H_{PP} \text{ and } M_{n,TL} < \mathbb{T}^L_{TL} \\ 0, & \text{otherwise} \end{cases} \tag{B.3}$$

where $\mathbb{T}_{TL}$ and $\mathbb{T}_{PP}$ are thresholds used to trigger events.

The request regarding the CIO can be expressed as follows:

$$U^{MRO}_{n,CIO} = \begin{cases} \varphi\left(1.5, 5, M_{n,TL2}/\mathbb{T}^H_{TL2}\right) - 1 & \text{if } \exists \dot{n} \in \mathcal{N} \backslash \{n\} \text{ s.t. } \breve{U}_{\dot{n}\to n} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{B.4}$$

where $\breve{U}_{\dot{n}\to n}$ is an *intra-MRO message*. The MRO instance running on cell n will receive such messages from all its neighbouring MRO instances ($\dot{n} \in \mathcal{N}_{-n}$) requesting a CIO increase request, if needed:

$$\breve{U}_{\dot{n}\to n} = \begin{cases} 1 & \text{if } \left(\left(M_{\dot{n},TL} > \mathbb{T}^H_{TL} \text{ or } M_{\dot{n},PP} \geq \mathbb{T}^L_{PP}\right) \text{ or } \left(M_{\dot{n},TL} > \mathbb{T}^H_{TL} \text{ or } M_{\dot{n},PP} \geq \mathbb{T}^L_{PP}\right)\right) \\ & \text{and } n \in \xi\left(\dot{n}\right) \\ 0 & \text{otherwise} \end{cases} \tag{B.5}$$

where $\xi\left(\dot{n}\right)$ is the set on cell from which there are Too Late HOs to cell $\dot{n}$. Increasing the CIO values of neighbouring cells reduces the number of too late HOs originating from them.

Appendix C

SON functions for HetNet scenarios

The SON functions are defined in this appendix considering one Macro cell (indexed by $n = 1$) and its $N - 1$ slave Pico cells (indexed by $n \in \{2, .., N\}$). For the same SON function, the instance on the Macro cell collaborate with the ones on the Pico cells.

3.1 (MLB via) CRE for a HetNet

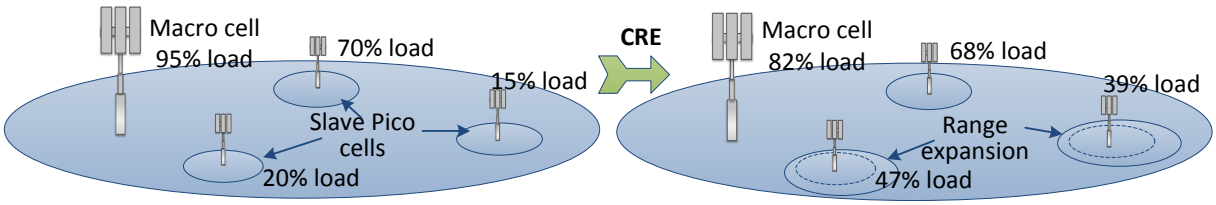


Figure C.1: HetNet CRE Function

The purpose of a CRE instances (Fig. C.1) is to *off-load* (force UEs to HO to neighbouring cells) the Macro cells onto Pico cells when they are *over-loaded*, i.e. have a high resource utilization. The mechanism also deals with the case where the Pico cells are (or we anticipate they will become) *over-loaded*.

Given some time instance (we disregard the time index), we define:

- $M_{n,LD}$ the resource utilization of cell n, i.e. the no. of used resources by the no. of available resources (the ABS are not considered available resources),
- $\mathbb{T}^H_{LD}$ and $\mathbb{T}^L_{LD} = \mathbb{T}^H_{LD} - 0.1$ are triggering thresholds,
- $\varphi\left(mid, sloap, x\right)$ is the logistic function, see eq. (B.1)

The CRE instances send update requests targeting the CIOs of the Pico cells ($n \in \{2, ..., N\}$) as follows:

$$U^{CRE}_{n,CIO} = \begin{cases} \varphi\left(0.9, 20, M_{1,LD}\right), & \text{if } M_{1,LD} > \mathbb{T}^H_{LD} \text{ and } M_{n,LD} < \mathbb{T}^L_{LD} \\ -\varphi\left(0.9, 20, M_{n,LD}\right), & \text{if } M_{n,LD} > \mathbb{T}^H_{LD} \text{ or} \\ & \left(M_{1,LD} < \mathbb{T}^L_{LD} \text{ and } n = \text{argmax}_i M_{i,LD}\right) \\ 0, & \text{otherwise} \end{cases} \tag{C.1}$$

3.2 MRO for a HetNet

The target of the MRO instances (Fig. C.2) is to reduce the no. of ping-pongs, the no. of too late HOs, the no. of too early HOs and the number of HOs to wrong cells. We only focus on the first two, as the last two are considerably smaller. To do this MRO tunes mainly the HYS. Increasing the HYS should reduce the no. of ping-pongs. Reducing the HYS should reduce the no. of too late HOs. But when we want to do both we turn to the CIO. Reducing the CIO of one cell should reduce the no. of too late HOs originating from that cell, without affecting the no. of ping-pongs. The CIO tuning feature can be deactivated (it is the case in some of our scenarios, specifically throughout chapter 3).

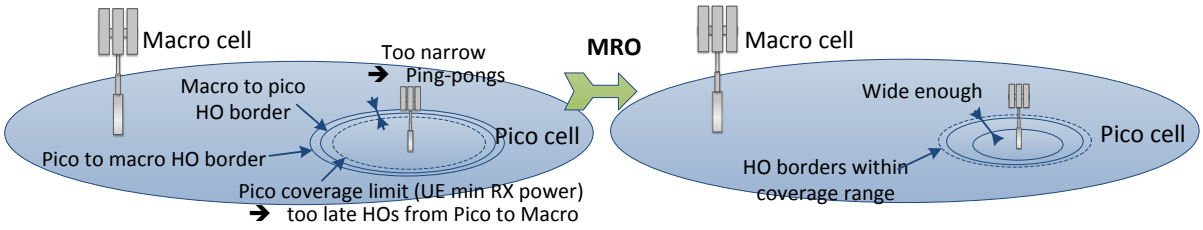


Figure C.2: HetNet MRO Function

Typically the no. of too late HOs originating from the Pico cells is significantly bigger then the ones that originate from the Macro cell. Thus the MRO instances focus on reducing the no. of too late HOs and ping-pong HOs originating from the Pico cells. Given some time instance (we disregard the time index), we define:

- $M_{n,PP}$ and $M_{n,TL}$ are the percentages of no. of ping-pongs and of too late HOs, respectively, originating from cell n,
- $\mathbb{T}^H_{PP}$, $\mathbb{T}^H_{TL}$, $\mathbb{T}^L_{PP} = 0.25\mathbb{T}^H_{PP}$and $\mathbb{T}^L_{TL} = 0.25\mathbb{T}^H_{TL}$ are thresholds used to trigger parameter changes.
- $\varphi\left(mid, sloap, x\right)$ is the logistic function, see eq. (B.1)

The MRO instances send update requests targeting only the parameters of the Pico cell ($n \in \{2, ..., N\}$), specifically the HYS:

$$U^{MRO}_{n,HYS} = \begin{cases} 1, & \text{if } M_{n,PP} > \mathbb{T}_{PP} \text{ and not } \left(M_{n,TL} \geq \mathbb{T}^L_{TL} \text{ and } \frac{\varphi\left(0.3,20,M_{n,PP}\right)}{\varphi\left(0.2,30,M_{n,TL}\right)} \leq 2 \right) \\ -1, & \text{if } M_{n,TL} > \mathbb{T}_{TL} \text{ and not } \left(M_{n,PP} \geq \mathbb{T}^L_{PP} \text{ and } \frac{\varphi\left(0.2,30,M_{n,TL}\right)}{\varphi\left(0.3,20,M_{n,PP}\right)} \leq 2 \right) \\ 0, & \text{otherwise} \end{cases} \tag{C.2}$$

and the CIO (if activated):

$$U^{MRO}_{n,CIO} = \begin{cases} -\max \left\{ \begin{array}{c} \varphi\left(0.3, 20, M_{n,PP}\right), \\ \varphi\left(0.2, 20, M_{n,TL}\right) \end{array} \right\}, & \text{if } \left(M_{n,PP} > \mathbb{T}^H_{PP} \text{ and } M_{n,TL} \geq \mathbb{T}^L_{TL}\right) \text{ or } \left(M_{n,PP} \geq \mathbb{T}^L_{PP} \text{ and } M_{n,TL} > \mathbb{T}^H_{TL}\right) \\ 0, & \text{otherwise} \end{cases} \tag{C.3}$$

3.3 eICIC for a HetNet

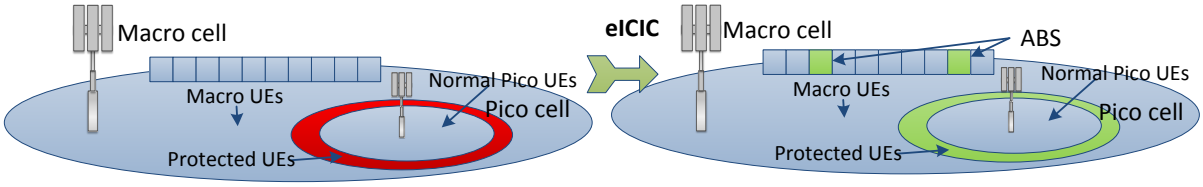


Figure C.3: HetNet eICIC Function

To protect the UEs which are at the Pico cell edge, especially when having CRE, the eICIC (Fig. C.3) instances tune the no. of ABS of the Macro cells. We consider that we may have at most 36 ABS sub-frames out of 40 sub-frames, i.e. sub-frames 0 and 5 of every frame will never be ABS. The ABS pattern is random and we reset it every frame (40 sub-frames). The protected UEs, i.e. the ones which were forced to HO from the Macro cell to the Pico cells, will take advantage of the ABS sub-frames due to the proportional fair (PF) scheduler that we employ, as the instantaneous channel conditions should be significantly better during the ABS.

Only the Macro cell will change the no. of ABS sub-frames. Given some time instance (we disregard the time index), we define:

- $M_{n,TR}$, is the ratio between the throughput of the UEs attached to cell n and the throughput all the protected UEs, which would normally be attached to cell n if all the CIOs, HYSs and TTTs were null.
- $\mathbb{T}^L_{TR} = 1.0$ and $\mathbb{T}^H_{TR} = 2.5$ are thresholds used to trigger parameter changes.
- $\varphi\left(mid, sloap, x\right)$ is the logistic function (see eq. B.1)

The eICIC instances send update requests targeting the ABS of the Macro cell ($n = 1$):

$$U^{eICIC}_{n,ABS} = \begin{cases} \varphi\left(4.0, 1.0, M_{n,TR}\right) & \text{, if } M_{n,TR} \geq \mathbb{T}^H_{TR} \\ -1.0 + \varphi\left(0.7, 5.0, M_{n,TR}\right) & \text{, if } M_{n,TR} \leq \mathbb{T}^L_{TR} \\ 0 & \text{, otherwise} \end{cases} \tag{C.4}$$

Appendix D

Reinforcement Learning

4.1 Proof of Theorem 4.1

Proof. To simplify (4.7), we first calculate ($\forall t \in \mathbb{N}$) :

$$\begin{array}{l}
\mathbb{P}\left(U_{(t+1)} = u' | S_{(0)} = s, A_{(0)} = a, P_{(1)} = p'\right) \\
= \sum_{u''} \mathbb{P}\left(U_{(t+1)} = u', U_{(1)} = u'' | S_{(0)} = s, A_{(0)} = a, P_{(0)} = p'\right) \\
= \sum_{u''} \mathbb{P}\left(U_{(t+1)} = u' | S_{(1)} = (p', u''), S_{(0)} = s, A_{(0)} = a\right) \cdot \\
\qquad \mathbb{P}\left(U_{(t+1)} = u'' | S_{(0)} = s, A_{(0)} = a, P_{(1)} = p'\right) \\
\overset{(\divideontimes)}{=} \sum_{u''} \mathbb{P}\left(U_{(t+1)} = u' | S_{(t+1)} = (p', u'')\right) \mathbb{P}\left(U_{(1)} = u'' | P_{(1)} = p'\right) \\
= \mathbb{P}\left(U_{(t+1)} = u' | P_{(1)} = p'\right)
\end{array} \tag{D.1}$$

where ($\divideontimes$) comes from using the Markov property on the first term and eq. (4.3) on the second term.

Now, from (4.7) we have:

$$\begin{array}{rl}
Q^{\pi}(s,a) & = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R_{(t+1)|} S_{(0)} = s, A_{(0)} = a\right] = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \rho\left(U_{(t+1)}\right) | S_{(0)} = s, A_{(0)} = a\right] \\
& \overset{A4.2}{=} \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \rho\left(U_{(t+1)}\right) \ | P_{(1)} = g(s,a), S_{(0)} = s, A_{(0)} = a\right] \\
& \overset{(D.1)}{=} \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \rho\left(U_{(t+1)}\right) \ | P_{(1)} = g(s,a)\right] = W^{\pi}(g(s,a))
\end{array} \tag{D.2}$$

Denote $p = g(s,a)$. We can further develop $W^{\pi}(p)$, $\forall p \in \mathcal{P}$, as follows:

$$\begin{array}{rl}
W^{\pi}(p) & = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \rho\left(U_{(t+1)}\right) \ | P_{(1)} = p\right] \\
& = \sum_i \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \rho_i\left(U_{(t+1)}\right) \ | P_{(1)} = p\right] \\
& = \sum_i W_i^{\pi}(p)
\end{array} \tag{D.3}$$

where we can further develop $W_i^{\pi}(p)$ as follows:

$$\begin{array}{rl}
W_i^{\pi}(p) & = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \rho_i\left(U_{(t+1)}\right) \ | P_{(1)} = p\right] \\
& = \mathbb{E}\left[\rho_i\left(U_{(1)}\right) \ | P_{(1)} = p\right] + \mathbb{E}_{\pi}\left[\sum_{t=1} \gamma^t \rho_i\left(U_{(t+1)}\right) | P_{(1)} = p\right] \\
& = \hat{r}_i(p) + \gamma \mathbb{E}_{\pi}\left[W_i^{\pi}\left(P_{(2)}\right) | P_{(1)} = p\right]
\end{array} \tag{D.4}$$

The conclusion for a policy π follows simply by detailing $\mathbb{E}_{\pi}$. □

4.2 Proof of proposition 4.1

Proof. To prove Prop. 4.1 we start from the expression of $\bar{a}$ in eq. (**??**):

$$\begin{array}{rl}
\bar{a} & = \arg\min_{a \in \mathcal{A}} \sum_{i \in \mathcal{I}} \langle \theta_i, F_i(g((p,u),a)) \rangle \\
& = \arg\min_{a \in \mathcal{A}} \sum_{i \in \mathcal{I}} \left\langle \theta_i, \sum_{n \in \mathcal{N}} \sum_{k \in \mathcal{K}} \omega_{n,k}^i g_{n,k}\left(\left(p_{n,k}, u_{n,k}^{\mathcal{Z}}\right), a_{n,k}\right) \right\rangle \\
& = \arg\min_{a \in \mathcal{A}} \sum_{i \in \mathcal{I}} \sum_{n \in \mathcal{N}} \sum_{k \in \mathcal{K}} g_{n,k}\left(\left(p_{n,k}, u_{n,k}^{\mathcal{Z}}\right), a_{n,k}\right) \left\langle \theta_i, \omega_{n,k}^i \right\rangle \\
& = \arg\min_{a \in \mathcal{A}} \sum_{n \in \mathcal{N}} \sum_{k \in \mathcal{K}} g_{n,k}\left(\left(p_{n,k}, u_{n,k}^{\mathcal{Z}}\right), a_{n,k}\right) \sum_{i \in \mathcal{I}} \left\langle \theta_i, \omega_{n,k}^i \right\rangle \\
& = \arg\min_{a \in \mathcal{A}} \sum_{(n,k) \in \mathcal{N} \times \mathcal{K}} g_{n,k}\left(\left(p_{n,k}, u_{n,k}^{\mathcal{Z}}\right), a_{n,k}\right) \Omega_{n,k}
\end{array} \tag{D.5}$$

As the sum components are independent we can maximize them individually, thus $\forall n, k$ we can write:

$$\bar{a}_{n,k} = \arg\min_{a_{n,k} \in \mathcal{A}_{n,k}} g_{n,k}\left(\left(p_{n,k}, u_{n,k}^{\mathcal{Z}}\right), a_{n,k}\right) \Omega_{n,k} \tag{D.6}$$

□

Appendix E

Mobility Parameters

5.1 Proof of Lemma 5.2

Proof. Consider the border $\mathbb{B}(\mathcal{A},\mathcal{B},z_{AB})$ and border $\mathbb{B}(\mathcal{C},\mathcal{A},z_{CA}) = \mathbb{B}(\mathcal{A},\mathcal{C},1/z_{CA})$. Denote $\bar{z}_{AC} = 1/z_{CA}$. We want to prove that the condition for the two borders to intersect $\mathbb{B}(\mathcal{A},\mathcal{B},z_{AB}) \cap \mathbb{B}(\mathcal{A},\mathcal{C},\bar{z}_{AC}) \neq \emptyset$ is:

$$(bz_{AB} + c\bar{z}_{AC})^2 \geq a^2 \geq (bz_{AB} - c\bar{z}_{AC})^2 . \tag{E.1}$$

We explicitate $\mathbb{B}(\mathcal{A},\mathcal{B},z_{AB})$ by developing $d_{(\mathcal{A},\ell)} \cdot z_{AB} = d_{(\mathcal{B},\ell)}$ $(\ell = (x,y))$ as follows:

$$\left[(x-x_A)^2 + (y-y_A)^2\right]\rho_{AB}^2 = \left[(x-x_B)^2 + (y-y_B)^2\right] \tag{E.2}$$

from where we easily get:

$$\begin{aligned}\left(x^2+y^2\right)\left(z_{AB}^2-1\right) - 2x\left(x_A z_{AB}^2 - x_B\right) - 2y\left(y_A z_{AB}^2 - y_B\right) + \\ \left(x_A^2 z_{AB} - x_B^2 + y_A^2 z_{AB} - y_B^2\right) = 0.\end{aligned}$$

If $z_{AB} \neq 1$ then $\mathbb{B}(\mathcal{A},\mathcal{B},z_{AB})$ is a circle of radius and center coordinates:

$$\begin{aligned} r_{AB}^2 &= \frac{(x_A-x_B)^2+(y_A-y_B)^2}{\left(z_{AB}^2-1\right)\left(1-1/z_{AB}^2\right)} = \frac{z_{AB}^2 c^2}{\left(z_{AB}^2-1\right)^2} \\ x_{AB} &= \frac{x_A z_{AB}^2 - x_B}{z_{AB}^2-1} \\ y_{ij}(z) &= \frac{y_A z_{AB}^2 - y_B}{z_{AB}^2-1} \end{aligned}$$

If $z_{AB} = 1$ then $\mathbb{B}(\mathcal{A},\mathcal{B},z_{AB})$ is a line $xm_{AB} + yn_{AB} = p_{AB}$ where:

$$\begin{aligned} m_{AB} &= 2(x_A - x_B) \\ n_{AB} &= 2(y_A - y_B) \\ p_{AB} &= \left(x_A^2 - x_B^2 + y_A^2 - y_B^2\right) \end{aligned} \tag{E.3}$$

Similarly we express $r_{AC}, x_{AC}, y_{AC}, m_{AC}, n_{AC}$ and p_{AC} when explicitating $\mathbb{B}(\mathcal{A},\mathcal{C},\bar{z}_{AC})$.Next we detail 4 cases based on the values of z_{AB} and $\bar{z}_{AC}$.

First case is $z_{AB} \neq 1$ and $\bar{z}_{AC} \neq 1$ so the borders $\mathbb{B}(\mathcal{A},\mathcal{B},z_{AB})$ and $\mathbb{B}(\mathcal{A},\mathcal{C},\bar{z}_{AC})$ have circle equations. The condition for the 2 circles to intersect is:

$$r_{AB} + r_{AC} \geq \sqrt{(x_{AB}-x_{AC})^2 + (y_{AB}-y_{AC})^2} \geq |r_{AB} - r_{AC}| \tag{E.4}$$

Developing the conditions we obtain:

$\Rightarrow (r_{AB}+r_{AC})^2 \geq (x_{AB}-x_{AC})^2 + (y_{AB}-y_{AC})^2 \geq (r_{AB}-r_{AC})^2$

$\Rightarrow 2r_{AB}r_{AC} \geq x_{AB}^2 + x_{AC}^2 - 2x_{AB}x_{AC} + y_{AB}^2 + y_{AC}^2 - 2y_{AB}y_{AC} - r_{AB}^2 - r_{AC}^2 \geq -2r_{AB}r_{AC}$

We partially replace the variables and rearrange the terms, coming to:

$$\Rightarrow 2r_{AB}r_{AC} \geq \left[\left(\frac{x_A z_{AB}^2 - x_B}{z_{AB}^2-1}\right)^2 - \frac{(x_A-x_B)^2 z_{AB}^2}{\left(z_{AB}^2-1\right)^2}\right] + \left[\left(\frac{x_A \bar{z}_{AC}^2 - x_C}{\bar{z}_{AC}^2-1}\right)^2 - \frac{(x_A-x_C)^2 \bar{z}_{AC}^2}{\left(\bar{z}_{AC}^2-1\right)^2}\right] - 2 \cdot \frac{x_A z_{AB}^2 - x_B}{z_{AB}^2-1} \cdot \frac{x_A \bar{z}_{AC}^2 - x_C}{\bar{z}_{AC}^2-1} + \left[\left(\frac{y_A z_{AB}^2 - y_B}{z_{AB}^2-1}\right)^2 - \frac{(y_A-y_B)^2 z_{AB}^2}{\left(\nu_{AB}^2-1\right)^2}\right] + \left[\left(\frac{y_A \bar{z}_{AC}^2 - y_C}{\bar{z}_{AC}^2-1}\right)^2 - \frac{(y_A-y_C)^2 \bar{z}_{AC}^2}{\left(\bar{z}_{AC}^2-1\right)^2}\right] - 2 \cdot \frac{y_A z_{AB}^2 - y_B}{z_{AB}^2-1} \cdot \frac{y_A \bar{z}_{AC}^2 - y_C}{\bar{z}_{AC}^2-1} \geq -2r_{AB}r_{AC}$$

and we first process the square parenthesis and then group the relevant terms obtaining:

$$\Rightarrow 2r_{AB}r_{CAC} \geq \frac{-(x_A-x_C)^2 z_{AB}^2 - (x_A-x_B)^2 \bar{z}_{AC}^2 + (x_B-x_C)^2}{\left(z_{AB}^2-1\right)\left(\bar{z}_{AC}^2-1\right)} + \frac{-(y_A-y_C)^2 z_{AB}^2 - (y_A-y_B)^2 \bar{z}_{AC}^2 + (y_B-y_C)^2}{\left(z_{AB}^2-1\right)\left(\bar{z}_{AC}^2-1\right)} \geq -2r_{AB}r_{AC}$$

From this we easily get:

$$\Rightarrow 2r_{AB}r_{AC} \geq \frac{-b^2 z_{AB}^2 - c^2 \bar{z}_{AC}^2 + a^2}{\left(z_{AB}^2-1\right)\left(\bar{z}_{AC}^2-1\right)} \geq -2r_{AB}r_{AC}$$

and we also replace r_{AB} and r_{AC} getting to:

$$\Rightarrow \frac{2cb\nu_{AB}\bar{\nu}_{AC}}{\left|z_{AB}^2-1\right|\left|\bar{z}_{AC}^2-1\right|} \geq \frac{-b^2 z_{AB}^2 - c^2 \bar{z}_{AC}^2 + a^2}{\left(z_{AB}^2-1\right)\left(\bar{z}_{AC}^2-1\right)} \geq -\frac{2cbz_{AB}\bar{z}_{AC}}{\left|z_{AB}^2-1\right|\left|\bar{z}_{AC}^2-1\right|}$$

$$\Rightarrow 2bcz_{AB}\bar{z}_{AC} \geq -b^2 z_{AB}^2 - c^2 \bar{z}_{AC}^2 + a^2 \geq -2bcz_{AB}\bar{z}_{AC}$$

Thus in the end we reach eq. (E.1).

Second case is $z_{AB} \neq 1$ and $\bar{z}_{AC} = 1$ so the border $\mathbb{B}(\mathcal{A}, \mathcal{B}, z_{AB})$ is a circle and $\mathbb{B}(\mathcal{C}, \mathcal{A}, z_{CA})$ is a line. Applying the condition for a circle to intersect a line in our case we have:

$$r_{AB}^2\left(n_{AC}^2 + m_{AC}^2\right) - \left(x_{AB}m_{AC} + y_{AB}n_{AC} - p_{AC}\right)^2 \geq 0 \tag{E.5}$$

Developing the condition we obtain:

$$\Rightarrow c^2\left(4\left(x_C - x_A\right)^2 + 4\left(y_C - y_A\right)^2\right) z_{AB}^2 - \left[2 \cdot \left(x_C - x_A\right) \cdot \left(x_A z_{AB}^2 - x_B\right) + 2 \cdot \left(y_C - y_A\right) \cdot \left(y_A z_{AB}^2 - y_B\right) - \left(x_C^2 - x_A^2 + y_C^2 - y_A^2\right) \cdot \left(z_{AB}^2 - 1\right)\right]^2 \geq 0$$

By grouping terms we easily get:

$$\Rightarrow 4c^2b^2 z_{AB}^2 - \left[-\left(\left(x_A - x_C\right)^2 + \left(y_A - y_C\right)^2\right) z_{AB}^2 + \left(x_B - x_C\right)^2 + \left(y_B - y_C\right)^2 - \left(x_B - x_A\right)^2 + \left(y_B - y_A\right)^2\right]^2 \geq 0$$

$$\Rightarrow 4c^2b^2 z_{AB}^2 - \left(-b^2 z_{AB}^2 + a^2 - c^2\right)^2 \geq 0$$

$$\Rightarrow \left(2cbz_{AB} - b^2 z_{AB}^2 - c^2 + a^2\right) \cdot \left(2cbz_{AB} + b^2 z_{AB}^2 + c^2 - a^2\right) \geq 0$$

$$\Rightarrow \left(a^2 - (b-c)^2\right)\left((bz_{AB} + c)^2 - a^2\right) \geq 0$$

Looking at the 2 parenthesis one option is they should be both negative:

$$(bz_{AB} + c)^2 < a^2 < (bz_{AB} - c)^2$$

which is clearly impossible because $z_{AB} > 0$, $a > 0$, $b > 0$ and $c > 0$. The other options are: both positive, first parenthesis is null or second parenthesis is null:

$$\begin{aligned} &\text{I}: \quad (bz_{AB} + c)^2 > a^2 > (bz_{AB} - c)^2 \\ &\text{II}: \quad (bz_{AB} + c)^2 = a^2 \\ &\text{III}: \quad a^2 = (bz_{AB} - c)^2 \end{aligned}$$

Using the fact that $(bz_{AB} + c)^2 > (bz_{AB} - c)^2$ we can rewrite the conditions in II and III as:

$$\begin{aligned} &\text{II}: \quad (bz_{AB} + c)^2 = a^2 > (bz_{AB} - c)^2 \\ &\text{III}: \quad (bz_{AB} + c)^2 \geq a^2 = (bz_{AB} - c)^2 \end{aligned}$$

Reuniting $(\text{I} \cup \text{II}) \cup \text{III}$ we get the condition in eq. (E.1) with $\bar{z}_{AC} = 1$.

The third case would be $z_{AB} = 1$ and $\bar{z}_{AC} \neq 1$ and it can be easily solved as the second case.

The fourth case is $z_{AB} = 1$ and $\bar{z}_{AC} = 1$ so the borders are lines, i.e. they may intersect in at most one point. In this case the intersection point is actually the center of the circumscribed circle of $\triangle\mathcal{ABC}$. As we assumed that $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ are the vertices of a triangle (and they are not collinear) this means that we will always have a point of intersection. Note that if we replace $z_{AB} = \bar{z}_{AC} = 1$ in eq. (E.1) we get :

$$b + c \geq a \geq |b - c|$$

which is always true (due to the triangle rule), so we can say that also the fourth case has to respect the condition in eq. (E.1); although it will actually lead to no constraints but it helps to have a general condition $\forall z_{AB}, \bar{z}_{AC}$.

In all four cases we obtained the condition in eq. (E.1). Analogously we can prove the remaining statements. □

5.2 Proof of Lemma 5.3

Proof. First we prove 1) is equivalent to 2).

If $\mathbb{B}(\mathcal{A}, \mathcal{B}, z_{AB}) \cap \mathbb{B}(\mathcal{B}, \mathcal{C}, z_{BC}) \cap \mathbb{B}(\mathcal{C}, \mathcal{A}, z_{CA}) \neq \emptyset$ then this implies that $\mathbb{B}(\mathcal{A}, \mathcal{B}, z_{AB}) \cap \mathbb{B}(\mathcal{B}, \mathcal{C}, z_{BC}) \neq \emptyset$ and according to Lemma 5.2:

$$\mathbb{B}(\mathcal{A}, \mathcal{B}, z_{AB}) \cap \mathbb{B}(\mathcal{B}, \mathcal{C}, z_{BC}) \neq \emptyset \Leftrightarrow (b \cdot z_{AB} + c/z_{CA})^2 \geq a^2 \geq (b \cdot z_{AB} - c/z_{CA})^2$$

Also if the 3 borders have at least one common intersection point then $\exists \ell \in \mathcal{D}$ s.t. :

$$\frac{\begin{cases} d_{(\mathcal{A},\ell)} z_{AB} = d_{(\mathcal{B},\ell)} \\ d_{(\mathcal{B},\ell)} z_{BC} = d_{(\mathcal{C},\ell)} \\ d_{(\mathcal{C},\ell)} z_{CA} = d_{(\mathcal{A},\ell)} \end{cases}}{z_{AB} z_{BC} z_{CA} = 1}$$

so we see that the conditions are necessary.

To prove that they are sufficient we start from the fact that we have $(bz_{AB} + c/z_{CA})^2 \geq a^2 \geq (bz_{AB} - c/z_{CA})^2$ and therefore according to Lemma 5.2 $\mathbb{B}(\mathcal{A}, \mathcal{B}, z_{AB}) \cap \mathbb{B}(\mathcal{B}, \mathcal{C}, z_{BC}) \neq \emptyset$. So we can say that $\exists \ell \in \mathcal{D}$ s.t.:

$$\frac{\begin{cases} d_{(\mathcal{A},\ell)} z_{AB} = d_{(\mathcal{B},\ell)} \\ d_{(\mathcal{C},\ell)} z_{CA} = d_{(\mathcal{A},\ell)} \\ 1 = z_{AB} z_{BC} z_{CA} \end{cases}}{d_{(\mathcal{C},\ell)} = d_{(\mathcal{B},\ell)} z_{BC}}$$

so the intersection point of $\mathbb{B}(\mathcal{A}, \mathcal{B}, z_{AB})$ and $\mathbb{B}(\mathcal{C}, \mathcal{A}, z_{CA})$ is also on the border $\mathbb{B}(\mathcal{B}, \mathcal{C}, z_{BC})$ and thus $\mathbb{B}(\mathcal{A}, \mathcal{B}, z_{AB}) \cap \mathbb{B}(\mathcal{B}, \mathcal{C}, z_{BC}) \cap \mathbb{B}(\mathcal{C}, \mathcal{A}, z_{CA}) \neq \emptyset$.

In a similar manner you can easily prove 1) is equivalent to 3) and to 4). □

5.3 Proof of Theorem 5.1

Proof. If we have a CCH area in the sense $A \to B \to C \to A$ this means that $\exists \ell \in \mathcal{D}$ (a point on our map) s.t.: $f_A(\ell) = B$ and $f_B(\ell) = C$ and $f_C(\ell) = A$. Now we detail the conditions under which this could happen.

First $f_A(\ell) = B$ means according to Table 5.1 that we have the following possibilities:

$$\begin{array}{ll} \text{S1:} & \mathbb{T}_{AB}(\ell) > 0 \text{ and } \mathbb{T}_{AC}(\ell) \leq 0 \\ \text{S2:} & \mathbb{T}_{AB}(\ell) > 0 \text{ and } \mathbb{T}_{AC}(\ell) > 0 \text{ and } \mathbb{T}_{AB}(\ell) \geq \mathbb{T}_{AC}(\ell) \end{array}$$

Next we add a redundant condition in S1, namely $\mathbb{T}_{AB}(\ell) \geq \mathbb{T}_{AC}(\ell)$, and by reuniting with S2 we get:

$$\text{S}_{AB}: \quad \mathbb{T}_{AB}(\ell) > 0 \text{ and } \mathbb{T}_{AB}(\ell) \geq \mathbb{T}_{AC}(\ell)$$

In the same manner for $f_B(\ell) = C$ and $f_C(\ell) = A$, we get S_{BC} and S_{CA}. The intersection $\mathrm{S} = \mathrm{S}_{AB} \cup \mathrm{S}_{BC} \cup \mathrm{S}_{CA}$ comes to:

$$\begin{array}{ll} \text{S:} & \mathbb{T}_{AB}(\ell) > 0 \text{ and } \mathbb{T}_{BC}(\ell) > 0 \text{ and } \mathbb{T}_{CA}(\ell) > 0 \text{ and} \\ & \mathbb{T}_{AB}(\ell) \geq \mathbb{T}_{AC}(\ell) \text{ and } \mathbb{T}_{BC}(\ell) \geq \mathbb{T}_{BA}(\ell) \text{ and } \mathbb{T}_{CA}(\ell) \geq \mathbb{T}_{CB}(\ell) \end{array} \tag{E.6}$$

Note that according to Proposition 5.1 $\mathbb{T}_{AB}(\ell) > 0$ implies $\mathbb{T}_{BA}(\ell) \leq 0$. Thus in eq. (E.6) and as $\mathbb{T}_{BC}(\ell) > 0$ this makes $\mathbb{T}_{BC}(\ell) \geq \mathbb{T}_{BA}(\ell)$ redundant. Similarly we find that $\mathbb{T}_{AB}(\ell) \geq \mathbb{T}_{AC}(\ell)$ and $\mathbb{T}_{CA}(\ell) \geq \mathbb{T}_{CB}(\ell)$ are redundant. Thus eq. (E.6) is equivalent to:

$$\text{S':} \quad \mathbb{T}_{AB}(\ell) > 0 \text{ and } \mathbb{T}_{BC}(\ell) > 0 \text{ and } \mathbb{T}_{CA}(\ell) > 0 \tag{E.7}$$

and by explicitating S' we get:

$$\left\{ \begin{array}{l} AvRxPow_A(\ell) + HOM_{AB} < AvRxPow_B(\ell) \\ AvRxPow_B(\ell) + HOM_{BC} < AvRxPow_C(\ell) \\ AvRxPow_C(\ell) + HOM_{CA} < AvRxPow_A(\ell) \end{array} \right. \Leftrightarrow \left\{ \begin{array}{l} d_{(\mathcal{A},\ell)}\rho_{AB} > d_{(\mathcal{B},\ell)} \\ d_{(\mathcal{B},\ell)}\rho_{BC} > d_{(\mathcal{C},\ell)} \\ d_{(\mathcal{C},\ell)}\rho_{CA} \geq d_{(\mathcal{A},\ell)} \end{array} \right. , \tag{E.8}$$

This is true if and only if $\exists \pi_{AB} > 1$,$\pi_{BC} > 1$ and $\pi_{AC} \geq 1$ s.t.:

$$\left\{ \begin{array}{l} d_{(\mathcal{A},\ell)}\rho_{AB} = \pi_{AB} d_{(\mathcal{B},\ell)} \\ d_{(\mathcal{B},\ell)}\rho_{BC} = \pi_{BC} d_{(\mathcal{C},\ell)} \\ d_{(\mathcal{C},\ell)}\rho_{CA} = \pi_{CA} d_{(\mathcal{A},\ell)} \end{array} \right. \Leftrightarrow \left\{ \begin{array}{l} d_{(\mathcal{A},\ell)}\frac{\rho_{AB}}{\pi_{AB}} = d_{(\mathcal{B},\ell)} \\ d_{(\mathcal{B},\ell)}\frac{\rho_{BC}}{\pi_{BC}} = d_{(\mathcal{C},\ell)} \\ d_{(\mathcal{C},\ell)}\frac{\rho_{CA}}{\pi_{CA}} = d_{(\mathcal{A},\ell)} \end{array} \right. \tag{E.9}$$

We can see in (E.9) the equations of tree new borders:

$$\begin{array}{l} \mathbb{B}'_{AB} = \mathbb{B}\left(\mathcal{A}, \mathcal{B}, \frac{\rho_{AB}}{\pi_{AB}}\right) \\ \mathbb{B}'_{BC} = \mathbb{B}\left(\mathcal{B}, \mathcal{C}, \frac{\rho_{BC}}{\pi_{BC}}\right) \\ \mathbb{B}'_{CA} = \mathbb{B}\left(\mathcal{C}, \mathcal{A}, \frac{\rho_{CA}}{\pi_{CA}}\right) \end{array}$$

Using Theorem (5.3) we write the conditions to have an intersection of the 3 borders (so that the 3 equations have a solution):

$$\left\{ \begin{array}{ll} \text{I}: & b\frac{\rho_{AB}}{\pi_{AB}} + c\frac{\pi_{CA}}{\rho_{CA}} \geq a \geq \left| b\frac{\rho_{AB}}{\pi_{AB}} - c\frac{\pi_{CA}}{\rho_{CA}} \right| \\ \text{II}: & \frac{\rho_{AB}}{\pi_{AB}}\frac{\rho_{BC}}{\pi_{BC}}\frac{\rho_{CA}}{\pi_{CA}} = 1 \\ \text{III}: & \pi_{AB} > 1 \\ \text{IV}: & \pi_{BC} > 1 \\ \text{V}: & \pi_{CA} \geq 1 \end{array} \right.$$

and we have to check the condition to have at least one solution (i.e. $\exists \pi_{AB}, \pi_{BC}, \pi_{CA}$) that solves our system. We develop I obtaining:

$$\Leftrightarrow \left\{ \begin{array}{ll} \text{I.i}: & b\frac{\rho_{AB}}{\pi_{AB}} + c\frac{\pi_{CA}}{\rho_{CA}} \geq a \\ \text{I.ii}: & a \geq b\frac{\rho_{AB}}{\pi_{AB}} - c\frac{\pi_{CA}}{\rho_{CA}} \\ \text{I.iii}: & a \geq -b\frac{\rho_{AB}}{\pi_{AB}} + c\frac{\pi_{CA}}{\rho_{CA}} \\ \text{II}: & \frac{\rho_{AB}}{\pi_{AB}}\frac{\rho_{BC}}{\pi_{BC}}\frac{\rho_{CA}}{\pi_{CA}} = 1 \\ \text{III}: & \pi_{AB} > 1 \\ \text{IV}: & \pi_{BC} > 1 \\ \text{V}: & \pi_{CA} \geq 1 \end{array} \right. \Leftrightarrow \left\{ \begin{array}{ll} \text{I.i,iii}: & \frac{b\rho_{AB}}{\left| a - c\frac{\pi_{CA}}{\rho_{CA}} \right|} \geq \pi_{AB} \\ \text{I.ii}: & \pi_{AB} \geq \frac{b\rho_{AB}}{a + c\frac{\pi_{CA}}{\rho_{CA}}} \\ \text{II}: & \frac{\rho_{AB}}{\pi_{AB}}\frac{\rho_{BC}}{\pi_{BC}}\frac{\rho_{CA}}{\pi_{CA}} = 1 \\ \text{III}: & \pi_{AB} > 1 \\ \text{IV}: & \pi_{BC} > 1 \\ \text{V}: & \pi_{CA} \geq 1 \end{array} \right.$$

$$\Leftrightarrow \left\{ \begin{array}{ll} \text{I,III}: & \pi_{AB} \in \left[\frac{b\rho_{AB}}{a + c\frac{\pi_{CA}}{\rho_{CA}}}, \frac{b\rho_{AB}}{\left| a - c\frac{\pi_{CA}}{\rho_{CA}} \right|} \right] \bigcap (1, \infty) \\ \text{II}: & \pi_{BC} = \frac{\rho_{AB}\rho_{BC}\rho_{CA}}{\pi_{AB}\pi_{CA}} \\ \text{IV}: & \pi_{BC} \in (1, \infty) \\ \text{V}: & \pi_{CA} \in [1, \infty) \end{array} \right.$$

Clearly in order to have a solution, specifically for π_{AB} to exist in condition I,III we have the condition: $1 < \frac{\rho_{AB}}{\left|a-\frac{\pi_{CA}}{\rho_{CA}}c\right|}b$. To simplify writing of the proof, for any $u \in \mathbb{R}$ we denote u^+ the smallest real number greater than u and u^- the largest real number smaller than u.
Explicitating the *absolute* of our previous condition as C1 and C2 we get:

$$\begin{cases} \text{C1} & a - c\frac{\pi_{CA}}{\rho_{CA}} < b\rho_{AB} \\ \text{C2} & -a + c\frac{\pi_{CA}}{\rho_{CA}} < b\rho_{AB} \end{cases} \Leftrightarrow \begin{cases} \text{C1:} & \pi_{CA} \in \left((a-b\rho_{AB})\frac{\rho_{CA}}{c}, \infty\right) \\ \text{C2:} & \pi_{CA} \in \left(-\infty, (a+b\rho_{AB})\frac{\rho_{CA}}{c}\right) \end{cases}$$

Thus our equations system is equivalent to:

$$\begin{cases} \text{I,III :} & \pi_{AB} \in \left[\max\left\{\frac{b\rho_{AB}}{a+c\frac{\pi_{CA}}{\rho_{CA}}}, 1^+\right\}, \frac{b\rho_{AB}}{\left|a-c\frac{\pi_{CA}}{\rho_{CA}}\right|}\right] \neq \emptyset \\ \text{C1,C2,V:} & \pi_{CA} \in \left((a-b\rho_{AB})\frac{\rho_{CA}}{c}, (a+b\rho_{AB})\frac{\rho_{CA}}{c}\right) \bigcap [1, \infty) \\ \text{II :} & \pi_{BC} = \frac{\rho_{AB}\rho_{BC}\rho_{CA}}{\pi_{AB}\pi_{CA}} \\ \text{IV,II :} & \frac{\rho_{AB}\rho_{BC}\rho_{CA}}{\pi_{AB}\pi_{CA}} \in (1, \infty) \end{cases}$$

In order to have a solution for π_{CA} (in C1,C2,V) the condition is $1 < (a+b\rho_{AB})\frac{\rho_{CA}}{c}$ which you can write as C3:

$$\Leftrightarrow \begin{cases} \text{I,III :} & \pi_{AB} \in \left[\max\left\{\frac{b\rho_{AB}}{a+c\frac{\pi_{CA}}{\rho_{CA}}}, 1^+\right\}, \frac{b\rho_{AB}}{\left|a-c\frac{\pi_{CA}}{\rho_{CA}}\right|}\right] \neq \emptyset \\ \text{C1,C2,V:} & \pi_{CA} \in \left[\max\left\{\left((a-b\rho_{AB})\frac{\rho_{CA}}{c}\right)^+, 1\right\}, (a+b\rho_{AB})\frac{\rho_{CA}}{c}\right) \neq \emptyset \\ \text{II :} & \pi_{BC} = \frac{\rho_{AB}\rho_{BC}\rho_{CA}}{\pi_{AB}\pi_{CA}} \\ \text{IV,II :} & \pi_{AB} \in \left(0, \frac{\rho_{AB}\rho_{BC}\rho_{CA}}{\pi_{CA}}\right) \\ \text{C3:} & \frac{c}{\rho_{CA}} < a + b\rho_{AB} \end{cases}$$

At this point we have to intersect I,III and IV,II. We have a non-void intersection if and only if 2 conditions are satisfied:

$$\begin{cases} \text{C4:} & \frac{\rho_{AB}\rho_{BC}\rho_{CA}}{\pi_{CA}} > 1 \\ \text{C5:} & \frac{\rho_{AB}\rho_{BC}\rho_{CA}}{\pi_{CA}} > \frac{b\rho_{AB}}{a+c\frac{\pi_{CA}}{\rho_{CA}}} \end{cases} \Leftrightarrow \begin{cases} \text{C4:} & \pi_{CA} < \rho_{AB}\rho_{BC}\rho_{CA} \\ \text{C5:} & a\rho_{CA} > \pi_{CA}\left(\frac{b}{\rho_{BC}} - c\right) \end{cases}$$

We separate in 2 cases based on the sign of $\frac{b}{\rho_{BC}} - c$.

Case 1: if $\frac{b}{\rho_{BC}} - c > 0$ then C5 becomes C5': $\pi_{CA} < \frac{\rho_{CA}a}{\left(\frac{b}{\rho_{BC}}-c\right)}$ which gets us to:

$$\begin{cases} \text{I-IV :} & \pi_{AB} \in \left[\max\left\{\frac{b\rho_{AB}}{a+c\frac{\pi_{CA}}{\rho_{CA}}}, 1^+\right\}, \min\left\{\frac{b\rho_{AB}}{\left|a-c\frac{\pi_{CA}}{\rho_{CA}}\right|}, \left(\frac{\rho_{AB}\rho_{BC}\rho_{CA}}{\pi_{CA}}\right)^-\right\}\right] \neq \emptyset \\ \text{C1,C2,V:} & \pi_{CA} \in \left[\max\left\{\left((a-b\rho_{AB})\frac{\rho_{CA}}{c}\right)^+, 1\right\}, (a+b\rho_{AB})\frac{\rho_{CA}}{c}\right) \neq \emptyset \\ \text{II :} & \pi_{BC} = \frac{\rho_{AB}\rho_{BC}\rho_{CA}}{\pi_{AB}\pi_{CA}} \\ \text{C3} & \frac{c}{\rho_{CA}} < a + b\rho_{AB} \\ \text{C4,C5':} & \pi_{CA} \in \left(-\infty, \min\left\{\frac{a\rho_{CA}}{\left(\frac{b}{\rho_{BC}}-c\right)}, \rho_{AB}\rho_{BC}\rho_{CA}\right\}\right) \end{cases}$$

Considering C4,C5' and C1,C2,V the 2 *equations* have common solutions only if we also have C6', C7', C8' and C9':

$$\begin{cases} \text{C6':} & \rho_{AB}\rho_{BC}\rho_{CA} > 1 \\ \text{C7':} & \rho_{AB}\rho_{BC}\rho_{CA} > (a-b\rho_{AB})\frac{\rho_{CA}}{c} \\ \text{C8':} & \frac{a\rho_{CA}}{\left(\frac{b}{\rho_{BC}}-c\right)} > 1 \\ \text{C9':} & \frac{a\rho_{CA}}{\left(\frac{b}{\rho_{BC}}-c\right)} > (a-b\rho_{AB})\frac{\rho_{CA}}{c} \end{cases} \Leftrightarrow \begin{cases} \text{C6':} & \rho_{AB}\rho_{BC}\rho_{CA} > 1 \\ \text{C7':} & \frac{a}{\rho_{AB}} < b + c\rho_{BC} \\ \text{C8':} & a\rho_{CA} > \frac{b}{\rho_{BC}} - c \\ \text{C9':} & 2ac\rho_{BC} + b^2\rho_{AB} > ab + bc\rho_{BC}\rho_{AB} \end{cases}$$

We next prove that C9' is always true given C7' and case 1 assumption ($\frac{b}{\rho_{BC}} - c > 0$). If $a - b\rho_{AB} \leq 0$ clearly C9' is always true. If $a - b\rho_{AB} > 0$ considering also C7' we have :

$$\begin{cases} a > b\rho_{AB} |^{\cdot 2c\rho_{BC}} \\ b\rho_{AB} + c\rho_{BC}\rho_{AB} > a |^{\cdot b} \end{cases} \Rightarrow \overline{2ac\rho_{BC} + b^2\rho_{AB} > ab + cb\rho_{BC}\rho_{AB}}$$

and therefore C9' is again always true. So we can omit C9' an thus in C3, C6' and C7' and C8' we recover the conditions in equation (5.14).

Case 2: if $\frac{b}{\rho_{BC}} - c \leq 0$ then C5 becomes C5": $\pi_{CA} \in \mathbb{R}$ which gets us to :

$$\begin{cases} \text{I-IV :} & \pi_{AB} \in \left[\max\left\{ \frac{b\rho_{AB}}{a + c\frac{\pi_{CA}}{\rho_{CA}}}, 1^+ \right\}, \min\left\{ \frac{b\rho_{AB}}{\left| a - c\frac{\pi_{CA}}{\rho_{CA}} \right|}, \left(\frac{\rho_{AB}\rho_{BC}\rho_{CA}}{\pi_{CA}} \right)^- \right\} \right] \neq \emptyset \\ \text{C1,C2,V:} & \pi_{CA} \in \left[\max\left\{ \left((a - b\rho_{AB}) \frac{\rho_{CA}}{c} \right)^+, 1 \right\}, (a + b\rho_{AB}) \frac{\rho_{CA}}{c} \right) \neq \emptyset \\ \text{II :} & \pi_{BC} = \frac{\rho_{AB}\rho_{BC}\rho_{CA}}{\pi_{AB}\pi_{CA}} \\ \text{C3} & \frac{c}{\rho_{CA}} < a + b\rho_{AB} \\ \text{C4,C5":} & \pi_{CA} \in (-\infty, \rho_{AB}\rho_{BC}\rho_{CA}) \end{cases}$$

In order to have a common solution for C4,C5" and C1,C2,V we need that the following conditions C6", C7" are satisfied:

$$\begin{cases} \text{C6":} & \rho_{AB}\rho_{BC}\rho_{CA} > 1 \\ \text{C7":} & \rho_{AB}\rho_{BC}\rho_{CA} > (a - \rho_{AB}b) \frac{\rho_{CA}}{c} \end{cases} \Leftrightarrow \begin{cases} \text{C6":} & \rho_{AB}\rho_{BC}\rho_{CA} > 1 \\ \text{C7":} & \frac{a}{\rho_{AB}} < b + c\rho_{BC} \end{cases}$$

Because $\frac{b}{\rho_{BC}} - c \leq 0$ (case 2 condition) and $0 < a\rho_{CA}$ we can add an extra condition C8":$b/\rho_{BC} < c + a \cdot \rho_{CA}$ which will actually have no impact , but we do this in order to be able to generalize. In the end (for case 2) in C3, C6", C7" and C8" we recover the conditions in eq. (5.14). Thus both cases recover the conditions in eq. (5.14).

Note that

$$\begin{aligned} &HOM_{AB} + HOM_{BC} + HOM_{CA} \\ &= CIO_{AB} + HYS_{AB} - CIO_{BA} + CIO_{BC} + HYS_{BC} - CIO_{CB} + CIO_{CA} + HYS_{CA} - CIO_{AC} \\ &= CIO_A + HYS_A - CIO_B + CIO_B + HYS_B - CIO_C + CIO_C + HYS_C - CIO_A \\ &= H_A + H_B + H_C \geq 0 \end{aligned} \tag{E.10}$$

where the last inequality comes from the fact that, $\forall i \in \{A, B, C\}$, $HYS_i \geq 0$. Thus according to Theorem 5.1 we have that $\Gamma_{ABCA} = \emptyset$. Similarly $\Gamma_{ACBA} = \emptyset$. □

5.4 Proof of Theorem 5.2

Proof. Note that

$$\begin{aligned} &HOM_{AB} + HOM_{BC} + HOM_{CA} \\ &= CIO_{AB} + HYS_{AB} - CIO_{BA} + CIO_{BC} + HYS_{BC} - CIO_{CB} + CIO_{CA} + HYS_{CA} - CIO_{AC} \\ &= CIO_A + HYS_A - CIO_B + CIO_B + HYS_B - CIO_C + CIO_C + HYS_C - CIO_A \\ &= H_A + H_B + H_C \geq 0 \end{aligned} \tag{E.11}$$

where the last inequality comes from the fact that, $\forall i \in \{A, B, C\}$, $HYS_i \geq 0$. Thus according to Theorem 5.1 we have that $\Gamma_{ABCA} = \emptyset$. Similarly $\Gamma_{ACBA} = \emptyset$. □

Bibliography

[1] ITU-R, "Requirements related to technical performance for imt-advanced radio interface(s)," Rep. M.2134, 2008.

[2] S. Hämäläinen, H. Sanneck, and C. Sartori, *LTE Self-Organising Networks (SON): Network Management Automation for Operational Efficiency*. Wiley, 2011.

[3] 3GPP, "UMTS; LTE; Telecommunication management; SON; NRM; IRP; Requirements ," TS 32.521, 3rd Generation Partnership Project (3GPP), 2013.

[4] 3GPP, "Digital cellular telecommunications system (Phase 2+); UMTS; LTE; Telecommunication management; SON; NRM; IRP; Information System," TS 32.522, 3rd Generation Partnership Project (3GPP), 2013.

[5] 3GPP, "LTE; E-UTRA; Radio Resource Control; Protocol specification," TS 36.331, 3rd Generation Partnership Project (3GPP), 2012.

[6] 3GPP, "LTE; E-UTRA and E-UTRAN; Overall description," TS 36.300, 3rd Generation Partnership Project (3GPP), 2013.

[7] 3GPP, "E-UTRA; Mobility enhancements in heterogeneous networks," TR 36.839, 2012.

[8] 3GPP, "E-UTRA; Self-configuring and self-optimizing network (SON) use cases and solutions," TR 36.902, 3rd Generation Partnership Project (3GPP), 2011.

[9] S. Sesia, I. Toufik, and M. Baker, *LTE - The UMTS Long Term Evolution: From Theory to Practice 2nd Edition*. Wiley, 2011.

[10] T. Tsvetkov, S. Novaczki, H. Sanneck, and G. Carle, "A configuration management assessment method for son verification," in *Wireless Communications Systems (ISWCS), 2014 11th International Symposium on*, pp. 380–384, Aug 2014.

[11] R. Barco, P. Lázaro, V. Wille, and L. Díez, "Knowledge acquisition for diagnosis in cellular networks based on bayesian networks.," in *KSEM* (J. Lang, F. Lin, and J. Wang, eds.), vol. 4092 of *Lecture Notes in Computer Science*, pp. 55–65, Springer, 2006.

[12] R. Barco, P. Lázaro, V. Wille, L. Díez, and S. Patel, "Knowledge acquisition for diagnosis model in wireless networks.," *Expert Syst. Appl.*, vol. 36, no. 3, 2009.

[13] H. S. Tsvetko Tsvetkov, Szabolcs Novaczki and G. Carle., "A post-action verification approach for automatic configuration parameter changes in self-organizing networks.," in *In 6th International Conference on Mobile Networks and Management (MONAMI 2014),*, 2014.

[14] H. S. B. B. Christoph Frenzel, Tsvetko Tsvetkov and G. Carle., "Operational troubleshooting-enabled coordination in self-organizing networks.," in *In 6th International Conference on Mobile Networks and Management (MONAMI 2014),*, 2014.

[15] A. Tall, R. Combes, Z. Altman, and E. Altman, "Distributed coordination of self-organizing mechanisms in communication networks," *Control of Network Systems, IEEE Transactions on*, vol. 1, pp. 328–337, Dec 2014.

[16] F. JENSEN and T. Nielsen, *Bayesian Networks and Decision Graphs*. Information Science and Statistics, Springer, 2007.

[17] R. Khanafer, B. Solana, J. Triola, R. Barco, L. Moltsen, Z. Altman, and P. Lazaro, "Automated diagnosis for umts networks using bayesian network approach," *Vehicular Technology, IEEE Transactions on*, vol. 57, pp. 2451–2461, July 2008.

[18] R. Khanafer, L. Moltsen, H. Dubreil, Z. Altman, and R. Barco, "A bayesian approach for automated troubleshooting for umts networks," in *Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on*, pp. 1–5, Sept 2006.

[19] R. Barco, V. Wille, L. Díez, and M. Toril, "Learning of model parameters for fault diagnosis in wireless networks," *Wirel. Netw.*, vol. 16, pp. 255–271, Jan. 2010.

[20] R. Barco, L. Díez, V. Wille, and P. Lázaro, "Automatic diagnosis of mobile communication networks under imprecise parameters.," *Expert Syst. Appl.*, vol. 36, no. 1, pp. 489–500, 2009.

[21] J. Chen, H. Zhuang, B. Andrian, and Y. Li, "Difference-based joint parameter configuration for mro and mlb," in *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th*, pp. 1–5, 2012.

[22] J. Bartelt, A. Fehske, H. Klessig, G. Fettweis, and J. Voigt, "Joint bandwidth allocation and small cell switching in heterogeneous networks," in *Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th*, pp. 1–5, Sept 2013.

[23] K. T. Dinh and S. Kuklinski, "Joint implementation of several lte-son functions," in *Globecom Workshops (GC Wkshps), 2013 IEEE*, pp. 953–957, Dec 2013.

[24] J. Zhou, Y. Mo, and B. Wang, "Joint optimization between mlb and mro based on cell load balance for lte networks," in *Wireless Communications Signal Processing (WCSP), 2013 International Conference on*, pp. 1–5, Oct 2013.

[25] S. Berger, P. Zanier, M. Soszka, A. Fehske, I. Viering, and G. Fettweis, "What is the advantage of cooperation in self-organizing networks?," in *Wireless Days (WD), 2013 IFIP*, pp. 1–6, Nov 2013.

[26] K. Tsagkaris, N. Koutsouris, P. Demestichas, R. Combes, and Z. Altman, "Son coordination in a unified management framework," in *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, pp. 1–5, June 2013.

[27] N. Koutsouris, K. Tsagkaris, P. Demestichas, Z. Altman, R. Combes, L. Mamatas, S. Clayman, A. Galis, P. Peloso, and L. Ciavaglia, "Conflict free coordination of son functions in a unified management framework: Demonstration of a proof of concept prototyping platform," in *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pp. 1092–1093, May 2013.

[28] L. Schmelz, M. Amirijoo, A. Eisenblaetter, R. Litjens, M. Neuland, and J. Turk, "A coordination framework for self-organisation in lte networks," in *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*, pp. 193–200, 2011.

[29] T. Bandh, H. Sanneck, and R. Romeikat, "An experimental system for son function coordination," in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pp. 1–2, 2011.

[30] T. Bandh, R. Romeikat, H. Sanneck, and H. Tang, "Policy-based coordination and management of son functions," in *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*, pp. 827–840, 2011.

[31] K. Munasinghe, A. Jamalipour, and D. Sharma, "Eco-inspired load optimization for lte eutran," in *Wireless Communications and Networking Conference (WCNC), 2014 IEEE*, pp. 2085–2089, April 2014.

[32] Y. Li, M. Li, B. Cao, and W. Liu, "A conflict avoid method between load balancing and mobility robustness optimization in lte," in *Communications in China (ICCC), 2012 1st IEEE International Conference on*, pp. 143–148, Aug 2012.

[33] A. Lobinger, S. Stefanski, T. Jansen, and I. Balan, "Coordinating handover parameter optimization and load balancing in lte self-optimizing networks," in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pp. 1–5, 2011.

[34] P. Munoz, R. Barco, and S. Fortes, "Conflict resolution between load balancing and handover optimization in lte networks," *Communications Letters, IEEE*, vol. 18, pp. 1795–1798, Oct 2014.

[35] H. Y. Lateef, A. Imran, and A. Abu-Dayya, "A framework for classification of self-organising network conflicts and coordination algorithms," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*, pp. 2898–2903, Sept 2013.

[36] R. Romeikat, H. Sanneck, and T. Bandh, "Efficient, dynamic coordination of request batches in c-son systems," in *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, pp. 1–6, June 2013.

[37] N. Zia, S. Mwanje, and A. Mitschele-Thiel, "A policy based conflict resolution mechanism for mlb and mro in lte self-optimizing networks," in *Computers and Communication (ISCC), 2014 IEEE Symposium on*, pp. 1–6, June 2014.

[38] X. Gelabert, B. Sayrac, and S. Ben Jemaa, "A heuristic coordination framework for self-optimizing mechanisms in lte hetnets," *Vehicular Technology, IEEE Transactions on*, vol. 63, pp. 1320–1334, March 2014.

[39] J. Moysen and L. Giupponi, "A functional architecture for self-coordination in lte networks," in *European Wireless 2014; 20th European Wireless Conference; Proceedings of*, pp. 1–5, May 2014.

[40] R. Sutton and A. Barto, *Reinforcement Learning: an introduction*. A Bradford book, A Bradford Book, 1998.

[41] R. Combes, Z. Altman, and E. Altman, "Self-organizing relays: Dimensioning, self-optimization, and learning," *Network and Service Management, IEEE Transactions on*, vol. 9, no. 4, pp. 487–500, 2012.

[42] C. Szepesvári, *Algorithma for Reinforcement Learning*. G - Reference,Information and Interdisciplinary Subjects Series, Morgan & Claypool, 2010.

[43] M. Sheng, C. Yang, Y. Zhang, and J. Li, "Zone-based load balancing in lte self-optimizing networks: A game-theoretic approach," *Vehicular Technology, IEEE Transactions on*, vol. 63, pp. 2916–2925, July 2014.

[44] Z. Liu, P. Hong, K. Xue, and M. Peng, "Conflict avoidance between mobility robustness optimization and mobility load balancing," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pp. 1–5, 2010.

[45] P. Fotiadis, M. Polignano, D. Laselva, B. Vejlgaard, P. Mogensen, R. Irmer, and N. Scully, "Multi-layer mobility load balancing in a heterogeneous lte network," in *Vehicular Technology Conference (VTC Fall), 2012 IEEE*, pp. 1–5, Sept 2012.

[46] W. Kim and Y.-J. Suh, "Enhanced adaptive periodic mobility load balancing algorithm for lte femtocell networks," in *Network of the Future (NOF), 2013 Fourth International Conference on the*, pp. 1–5, Oct 2013.

[47] N. Zia and A. Mitschele-Thiel, "Self-organized neighborhood mobility load balancing for lte networks," in *Wireless Days (WD), 2013 IFIP*, pp. 1–6, Nov 2013.

[48] P. Munoz, R. Barco, J. Ruiz-Aviles, I. de la Bandera, and A. Aguilar, "Fuzzy rule-based reinforcement learning for load balancing techniques in enterprise lte femtocells," *Vehicular Technology, IEEE Transactions on*, vol. 62, pp. 1962–1973, Jun 2013.

[49] O. Iacoboaiea, B. Sayrac, S. Ben Jemaa, and P. Bianchi, "Low complexity SON coordination using reinforcement learning," in *GLOBECOM 2014*, (Austin, USA), December 2014.

[50] O. Iacoboaiea, B. Sayrac, S. Ben Jemaa, and P. Bianchi, "SON conflict resolution using reinforcement learning with state aggregation," in *ACM SIGCOMM 2014 - All Things Cellular 2014 Workshop*, (Chicago, USA), August 2014.

[51] O. Iacoboaiea, B. Sayrac, S. Ben Jemaa, and P. Bianchi, "SON coordination for parameter conflict resolution: A reinforcement learning framework," in *IEEE WCNC 2014 - Workshop on Self-Organizing Networks (WCNC'14 - SONET Workshop)*, (Istanbul, Turkey), Apr. 2014.

[52] T. A. Moon, "The apollonian circles and isodynamic points," *Mathematical Reflections*, 2010.

[53] J. Cox and M. B. Partensky, "Spatial Localization Problem and the Circle of Apollonius," *ArXiv Physics e-prints*.

[54] 3GPP, "E-UTRA; Further advancements for E-UTRA physical layer aspects," TR 36.814, 3rd Generation Partnership Project (3GPP), 2010.

[55] ITU-R, "Guidelines for evaluation of radio interface technologies for imt-advanced," Rep. M.2135, 2008.

[56] UniverSelf project web page `http://fp7-socrates.eu/`.

[57] SEMAFOUR project web page `http://fp7-semafour.eu/`.

[58] A. Baddeley, P. Gregori, J. Mahiques, R. Stoica, and D. Stoyan, *Case Studies in Spatial Point Process Modeling*. Lecture Notes in Statistics, Springer-Verlag New York Incorporated, 2006.

[59] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. Andrews, "User association for load balancing in heterogeneous cellular networks," *Wireless Communications, IEEE Transactions on*, vol. 12, pp. 2706–2716, June 2013.

[60] Z. Altman, S. Sallem, R. Nasri, B. Sayrac, and M. Clerc, "Particle swarm optimization for mobility load balancing son in lte networks," in *Wireless Communications and Networking Conference Workshops (WCNCW), 2014 IEEE*, pp. 172–177, April 2014.

[61] C. Yang, "Concurrent mobility load balancing in lte self-organized networks," in *Telecommunications (ICT), 2014 21st International Conference on*, pp. 288–292, May 2014.

[62] Y. Yang, P. Li, X. Chen, and W. Wang, "A high-efficient algorithm of mobile load balancing in lte system," in *Vehicular Technology Conference (VTC Fall), 2012 IEEE*, pp. 1–5, Sept 2012.

[63] C. Yang, M. Sheng, H. Tian, and J. Li, "Double threshold design for mobility load balancing in self-optimizing networks," in *Vehicular Technology Conference (VTC Spring), 2014 IEEE 79th*, pp. 1–5, May 2014.

[64] J. Ruiz Aviles, M. Toril, S. Luna-Ramirez, V. Buenestado, and M. Regueira, "Analysis of limitations of mobility load balancing in a live lte system," *Wireless Communications Letters, IEEE*, vol. PP, no. 99, pp. 1–1, 2015.

[65] C. Yang, H. Li, and Y. Dai, "Net utility-oriented mobility load balancing in self-optimizing networks," in *Computer and Information Technology (CIT), 2014 IEEE International Conference on*, pp. 14–19, Sept 2014.

[66] P. Munoz, R. Barco, and I. de la Bandera, "On the potential of handover parameter optimization for self-organizing networks," *Vehicular Technology, IEEE Transactions on*, vol. 62, pp. 1895–1905, Jun 2013.

[67] C. Fischione, G. Athanasiou, and F. Santucci, "Dynamic optimization of generalized least squares handover algorithms," *Wireless Communications, IEEE Transactions on*, vol. 13, pp. 1235–1249, March 2014.

[68] D. Aziz, A. Ambrosy, L. T. W. Ho, L. Ewe, M. Gruber, and H. Bakker, "Autonomous neighbor relation detection and handover optimization in lte," *Bell Lab. Tech. J.*, vol. 15, pp. 63–83, Dec. 2010.

[69] I. Viering, B. Wegmann, A. Lobinger, A. Awada, and H. Martikainen, "Mobility robustness optimization beyond doppler effect and wss assumption," in *Wireless Communication Systems (ISWCS), 2011 8th International Symposium on*, pp. 186–191, Nov 2011.

[70] G. Zhou and P. Legg, "An evolved mobility robustness optimization algorithm for lte heterogeneous networks," in *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, pp. 1–5, June 2013.

[71] A. Klein, N. Kuruvatti, J. Schneider, and H. Schotten, "Fuzzy q-learning for mobility robustness optimization in wireless networks," in *Globecom Workshops (GC Wkshps), 2013 IEEE*, pp. 76–81, Dec 2013.

[72] H. Zhang, "A user mobility analysis assistive mro algorithm for handover parameters optimization in lte son system," in *Wireless Advanced (WiAd), 2012*, pp. 143–148, June 2012.

[73] L. Abdullah, M. Baba, and S. Ali, "Parameters optimization for handover between femtocell and macrocell in lte-based network," in *Control System, Computing and Engineering (ICCSCE), 2014 IEEE International Conference on*, pp. 636–640, Nov 2014.

[74] B. Wegmann, A. Awada, K. Kordybach, and I. Viering, "Inter-rat mro in 3gpp rel.11: What works, and what does not," in *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, pp. 1–5, June 2013.

[75] S. Mwanje and A. Mitschele-Thiel, "Distributed cooperative q-learning for mobility-sensitive handover optimization in lte son," in *Computers and Communication (ISCC), 2014 IEEE Symposium on*, vol. Workshops, pp. 1–6, June 2014.

[76] W. Qin, Y. Teng, M. Song, Y. Zhang, and X. Wang, "Aq-learning approach for mobility robustness optimization in lte-son," in *Communication Technology (ICCT), 2013 15th IEEE International Conference on*, pp. 818–822, Nov 2013.

[77] Y. Watanabe, H. Sugahara, Y. Matsunaga, and K. Hamabe, "Inter-enb coordination-free algorithm for mobility robustness optimization in lte hetnet," in *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, pp. 1–5, June 2013.

[78] K. Kikuchi and H. Otsuka, "Proposal of adaptive control cre in heterogeneous networks," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*, pp. 910–914, Sept 2012.

[79] K. Kikuchi and H. Otsuka, "Parameter optimization for adaptive control cre in hetnet," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*, pp. 3334–3338, Sept 2013.

[80] S. Matsuoka, K. Kikuchi, and H. Otsuka, "Performance analysis of cooperative control between cso on cre and transmission power for pico-enb in hetnet," in *Ubiquitous and Future Networks (ICUFN), 2014 Sixth International Conf on*, pp. 505–506, July 2014.

[81] S.-S. Sun, W. Liao, and W.-T. Chen, "Traffic offloading with rate-based cell range expansion offsets in heterogeneous networks," in *Wireless Communications and Networking Conference (WCNC), 2014 IEEE*, pp. 2833–2838, April 2014.

[82] A. Daeinabi, K. Sandrasegaran, and P. Ghosal, "A dynamic cell range expansion scheme based on fuzzy logic system in lte-advanced heterogeneous networks," in *Telecommunication Networks and Applications Conference (ATNAC), 2014 Australasian*, pp. 6–11, Nov 2014.

[83] C. Qvarfordt and P. Legg, "Evaluation of lte hetnet deployments with realistic traffic models," in *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2012 IEEE 17th International Workshop on*, pp. 307–311, Sept 2012.

[84] T. Kudo and T. Ohtsuki, "Cell range expansion using distributed q-learning in heterogeneous networks," in *Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th*, pp. 1–5, Sept 2013.

[85] B. Yasir, G. Su, and N. Bachache, "Range expansion for pico cell in heterogeneous lte-a cellular networks," in *Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference on*, pp. 1235–1240, Dec 2012.

[86] P. Tian, H. Tian, L. Gao, J. Wang, X. She, and L. Chen, "Deployment analysis and optimization of macro-pico heterogeneous networks in lte-a system," in *Wireless Personal Multimedia Communications (WPMC), 2012 15th International Symposium on*, pp. 246–250, Sept 2012.

[87] T. Kudo and T. Ohtsuki, "Cell selection using distributed q-learning in heterogeneous networks," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*, pp. 1–6, Oct 2013.

[88] S. Vasudevan, R. Pupala, and K. Sivanesan, "Dynamic eicic - a proactive strategy for improving spectral efficiencies of heterogeneous lte cellular networks by leveraging user mobility and traffic dynamics," *Wireless Communications, IEEE Transactions on*, vol. 12, pp. 4956–4969, October 2013.

[89] M. Kamel and K. Elsayed, "Performance evaluation of a coordinated time-domain eicic framework based on absf in heterogeneous lte-advanced networks," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, pp. 5326–5331, Dec 2012.

[90] Y. Wang, B. Soret, and K. Pedersen, "Sensitivity study of optimal eicic configurations in different heterogeneous network scenarios," in *Communications (ICC), 2012 IEEE International Conference on*, pp. 6792–6796, June 2012.

[91] A. Weber and O. Stanze, "Scheduling strategies for hetnets using eicic," in *Communications (ICC), 2012 IEEE International Conference on*, pp. 6787–6791, June 2012.

[92] L. Jiang and M. Lei, "Resource allocation for eicic scheme in heterogeneous networks," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*, pp. 448–453, Sept 2012.

[93] A. Li and L. Qiu, "Resource allocation in heterogeneous networks with eicic," in *Wireless Communications and Signal Processing (WCSP), 2014 Sixth International Conference on*, pp. 1–6, Oct 2014.

[94] S. Lembo, P. Lunden, O. Tirkkonen, and K. Valkealahti, "Optimal muting ratio for enhanced inter-cell interference coordination (eicic) in hetnets," in *Communications Workshops (ICC), 2013 IEEE International Conference on*, pp. 1145–1149, June 2013.

[95] F. B. Tesema, P. Zanier, I. Viering, A. J. Fehske, and G. P. Fettweis, "Simplified scheduler model for son algorithms of eicic in heterogeneous networks," in *European Wireless 2014; 20th European Wireless Conference; Proceedings of*, pp. 1–6, May 2014.

[96] Q. Li, H. Xia, Z. Zeng, and T. Zhang, "Dynamic enhanced inter-cell interference coordination using reinforcement learning approach in heterogeneous network," in *Communication Technology (ICCT), 2013 15th IEEE International Conference on*, pp. 239–243, Nov 2013.

[97] G. Bartoli, R. Fantacci, D. Marabissi, and M. Pucci, "Adaptive muting ratio in enhanced inter-cell interference coordination for lte-a systems," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2014 International*, pp. 990–995, Aug 2014.

[98] M. Behjati and J. Cosmas, "Self-organizing interference coordination for future lte-advanced network qos improvements," in *Broadband Multimedia Systems and Broadcasting (BMSB), 2014 IEEE International Symposium on*, pp. 1–6, June 2014.

[99] K. Pedersen, Y. Wang, S. Strzyz, and F. Frederiksen, "Enhanced inter-cell interference coordination in co-channel multi-layer lte-advanced networks," *Wireless Communications, IEEE*, vol. 20, pp. 120–127, June 2013.

[100] 3GPP, "Digital cellular telecommunications system (Phase 2+); UMTS; LTE; Telecommunication management; Principles and high level requirements," TS 32.101, 3rd Generation Partnership Project (3GPP), 2014.

[101] UniverSelf project web page `http://univerself-project.eu/`.

[102] A. Fehske, H. Klessig, J. Voigt, and G. Fettweis, "Concurrent load-aware adjustment of user association and antenna tilts in self-organizing radio networks," *Vehicular Technology, IEEE Transactions on*, vol. 62, pp. 1974–1988, Jun 2013.

[103] M. ul Islam and A. Mitschele-Thiel, "Reinforcement learning strategies for self-organized coverage and capacity optimization," in *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, pp. 2818–2823, 2012.

[104] 3GPP, "LTE; E-UTRA; Physical layer; Measurements," TS 36.214, 3rd Generation Partnership Project (3GPP), 2012.

[105] Master Telecom Faster web page `http://www.mastertelecomfaster.com`.

[106] 3GPP, "LTE; E-UTRA; Physical channels and modulation," TS 36.211, 3rd Generation Partnership Project (3GPP), 2013.

Coordination des fonctionnalités auto-organisantes (SON) dans les réseaux d'accès radio du futur

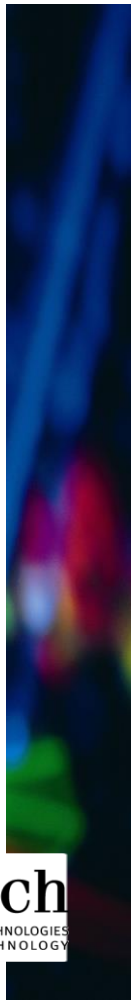
RESUME : Pour rester compétitifs les opérateurs de réseau doivent améliorer en permanence les capacités de leur réseau. Cela augmente la complexité de gestion qui se traduit par une augmentation des dépenses en capital (CAPEX) et des dépenses opérationnelles (OPEX). Le standard 3GPP a donc introduit dès la Release 8 les fonctions de réseau auto-organisantes (SON) qui automatisent les tâches de gestion et d'optimisation du réseau. Pour que toutes ces fonctions SON travaillent correctement ensemble (notamment dans un environnement multifournisseur), nous devons détecter si elles sont en conflit et appliquer un mécanisme de résolution si c'est le cas. Pour cela, la Release 10 du 3GPP a introduit la fonction de COordonnation des fonctions SON (SONCO). Dans cette thèse, nous nous intéressons aux deux fonctions principale de la coordination SON : la détection (SONCO-D) et la résolution (SONCO-R) de conflits. Le SONCO-D doit être en mesure d'identifier les conflits potentiels et de détecter ceux qui sont actifs, c'est-à-dire ceux qui dégradent effectivement la performance du réseau. Pour ce faire, nous utilisons un Classificateur Naïf Bayésien. Une fois que le conflit est identifié, un mécanisme SONCO-R peut être déclenché. Il peut donc décider de favoriser une fonction SON par rapport à une autre selon des critères prédéfinis. Comme nous considérons les fonctions SON comme des boites noires, nous utilisons une technique d'apprentissage par renforcement qui permet d'améliorer les décisions du SONCO-R en profitant de son expérience passée. Tout au long de notre travail, nous considérons que les paramètres de mobilité sont établis par cellule. Nous motivons ce choix dans un chapitre dédié.

Mots clés : Auto-optimisation; SON; coordination SON; SONCO; le diagnostic des conflits; la résolution des conflits; LTE-A; LTE; HetNets; CAPEX; OPEX.

Coordination of Self-Organizing Network (SON) functions in next generation radio access networks

ABSTRACT : To remain competitive operators have to continuously improve the capabilities of their networks. This increases the management complexity translating into increased CAPital EXpenditures (CAPEX) and OPerational EXpenditures (OPEX). Consequently Release 8 of 3GPP introduced the Self Organizing Network (SON) functions. For all these SON functions to work properly together (especially in a multi-vendor environment) we have to detect if they conflict and enforce a resolution mechanism if this is the case. For this purpose Release 10 of 3GPP introduced the SON COordinator (SONCO) function. In this thesis we provide two frameworks. The first one is for SON conflict diagnosis (SONCO-D). The SONCO-D has to be able to identify potential and active conflicts, i.e. which ones effectively degrade the network performance. For this purpose, we use Naive Bayesian Classifiers. The second framework is for SON conflict resolution (SONCO-R). Once a conflict is identified, a SONCO-R mechanism can be applied. It decides when to favor one SON function or another based on a predefined criterion. To this end, we use a Reinforcement Learning framework as it allows us to improve our decisions based on past experience. Throughout our work we consider the handover parameters to be established per cell (the same for all neighboring cells). We motivate this choice in a dedicated chapter.

Keywords : Self-optimization; SON; SON coordination; SONCO; SON conflict diagnosis; SON conflict resolution; LTE-A; LTE; HetNets; CAPEX; OPEX.



TELECOM ParisTech

EDITE

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY