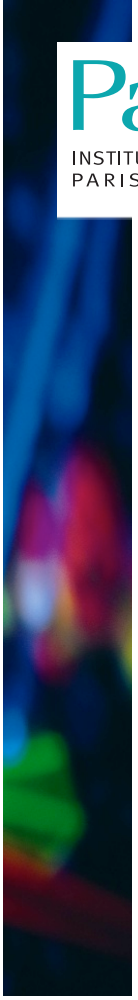
Performance analysis of video streaming services in mobile networks

Yu-Ting Lin



ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUT OF TECHNOLOGY

TELECOM ParisTech

2016-ENST-0080

EDITE - ED 130

Doctorat ParisTech

THESE

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité Informatique et Réseaux

présentée et soutenue publiquement par

Yu-Ting LIN

le 09 décembre 2016

Analyse de performance des services de vidéo streaming dans les réseaux mobiles

Directeur de thèse: **Thomas BONALD**
Co-directeur de thèse: **Salah Eddine ELAYOUBI**

Jury

M. Gerardo RUBINO, Directeur de Recherches, INRIA Rennes / IRISA, France	Rapporteur
M. Rachid EL AZOUZI, Professeur, Université d'Avignon, France	Rapporteur
M. Lin CHEN, Maître de Conférences, Université de Paris Sud, France	Examinateur
M. Philippe MARTINS, Professeur, Télécom ParisTech, France	Examinateur
M. Thomas BONALD, Professeur, Télécom ParisTech, France	Directeur de thèse
M. Salah Eddine EL AYOUBI, Ingénieur de Recherche, Orange Labs, France	Co-directeur de thèse

TELECOM ParisTech
Ecole de l'Institut Mines-Télécom - membre de ParisTech
46 rue Barrault 75013 Paris - (+33) 1 45 81 77 77 - www.telecom-paristech.fr

THÈSE

PhD ParisTech

A dissertation presented

in fulfillment of the requirements for the degree of Doctor of Philosophy of

TELECOM ParisTech

Speciality Computer Networking and Telecommunications

presented and defended in public by

Yu-Ting LIN

On 09 December 2016

Performance analysis of video streaming services in mobile networks

PhD Director: **Thomas Bonald**

À mes chers parents et ma soeur.

Acknowledgements

First of all, I would like to thank my supervisor Thomas Bonald from Télécom ParisTech for his patience, encouragements and instructions during these three years. Without his helps, I would not be able to complete my thesis. I would also like to thank Dr. Salah Eddine Elayoubi for his supports, supervision and providing me an opportunity to work in Orange Labs. The discussions that I had with them really help me clarify my subject of thesis and motivate me towards the end of Ph.D. I still remember the beginning moment of my thesis when I could not speak very good French so I appreciate their kindness and patience. From them, I learn not only the professional knowledge but also the attitude of working.

I also appreciate other colleagues including the colleagues from Orange (Berna, Eduardo, Nivine, Ridha and Sana) and colleagues from Texa A&M (I-Hong, Ping-Chen) who work together with me and contribute a lot on the discussion and inspiration. I am glad that during my Ph.D thesis I can work with so many people from different background and countries.

My Ph.D would not be perfect if I did not have the accompany and encouragements from my colleagues. As I work half of my time in Orange Lab located at Issy Les Moulineaux in the first two years and Chatillon in the last year, I really appreciate my three managers, Laurent Marceront, Benoit Badard and Pierre Dubois for their patience and always being willing to help me. I am also thankful for other CDI colleagues Arturo, Guillaume, Frank, Zwi, Thierry, Jean-Batispe, Julie who are always willing to teach me and open to share any information. Moreover, I also appreciate the Post-Doc and Ph.D students including Ahlem, Yasir, Ovidiu, Hajer, Abdoulaye, Aymen, Hind, Vaggelis, Habib from the previous team, RES and Thomas, Rita, Stefan, Mohamad from the later team, RIDE. I also appreciate the internship students who share the unforgettable moment with me including Thomas, Hatem, Xintao and Xinbao.

Due to the chance of participating in a French ANR project, IDEFIX, I am thankful to people who work and contribute within the project including Alain, Alberto, Tijani, Tania, Bruno, Hervé, Raluca, Calvin, Philippe etc for sharing a lot of different opinions which really broaden my horizon and inspire me a lot.

LINCS is the place where I worked the most frequent during my Ph.D. Thanks to the opportunity to stay in LINCS this open space, I can make so many friends who shared brilliant idea with me. I really appreciate Rémy who taught me how to access the servers of Télécom ParisTech, Marc-Olivier and Jordan who are always willing to answer me different questions. Also I have some colleagues like Maura, Andrea, Leonardo, Léonce, The Dang, Yixi, Christian, Rim, Dalia, Céline, Natalya, Mars, Sara, Diego who share the wonderful lunch moment and coffee break with me.

Last but not least, I would like to thank my parents who really supports me on pursuing my dream and my expectation and my sister, Karen, who encourages me a lot and always brings me a lot of food from Taiwan. I am satisfied on what I have experienced during my Ph.D career. This journey will end but I know that research will never end.

Abstract

As the traffic of video streaming increases significantly in mobile networks, it is essential for operators to account for the features of this traffic when dimensioning and configuring the network. The focus of this thesis is on traffic models of video streaming in mobile networks. For real-time streaming traffic, we derive the analytical form of an important Quality-of-Service (QoS) metric, the packet outage rate, and utilize the model for dimensioning. For HTTP adaptive streaming modeling, we propose to observe other QoS metrics such as mean video bit rate, service time of flows, deficit rate and buffer surplus for understanding the trade-off of video resolution and playback smoothness. Deficit rate corresponds to the probability that the users' throughput is lower than the selected video bit rate and buffer surplus stands for the mean buffer variation in a time unit. These QoS metrics are introduced to understand the QoE, in terms of the video starvation probability. We study by simulation the impacts of some key parameters of the system. We show that using smaller chunk duration, fewer video coding rates or round-robin scheduling scheme may provide a smoother video playback but decrease the mean video resolution. Moreover, we show that users' mobility could enlarge the system stability condition but performance trade-off still exists among different scheduling schemes. We also propose to adapt the number of chunks downloaded in an HTTP request so that each HTTP request has the same data volume.

Finally, we apply machine learning techniques to analyze the correlation between the quality of experience (QoE) of users and system characteristics, such as number of flows from a large amount of data generated by our simulator. This approach offers a chance to understand correlations that can not be easily resolved through developing analytical forms for QoE. We believe that this can provide insights for future research on network management in the presence of video streaming.

KEY-WORDS: Streaming Video, Real-Time Streaming, Adaptive Streaming, Quality of Experience, Video Chunk, Flow-Level Model, Machine Learning.

Résumé

Le trafic de vidéo streaming étant en très forte augmentation dans les réseaux mobiles, il devient essentiel pour les opérateurs de tenir compte des spécificités de ce trafic pour bien dimensionner et configurer le réseau. Dans cette thèse, nous nous intéressons à la modélisation du trafic de vidéo streaming dans les résaux mobiles. Pour le trafic de vidéo streaming en temps-réel, nous obtenons une forme analytique pour une métrique de qualité-de-service (QoS) importante, le taux de perte de paquets, et utilisons ce modèle pour proposer des méthodes de dimensionnement. Pour le trafic streaming HTTP adaptatif, nous utilisons d'autres métriques de performance telles que le débit moyen, le taux de déficit et le surplus de buffer pour comprendre le compromis entre résolution de vidéo et fuildité de la diffusion vidéo. Le taux de déficit correspond à la probabilité que le débit des utilisateurs soit inférieur au débit selectionné. Le surplus de buffer réprésente la variation moyenne du contenu du buffer par unité de temps. Ces métriques de QoS sont proposées pour mieux comprendre la qualité d'expérience (QoE). Nous étudions par simulation l'impact de quelques paramétres clés du système. Nous montrons que l'utilisation de segments vidéo plus courts, d'un nombre réduit d'encodages vidéo et de l'ordonnancement de type round-robin améliore la fluidité de la vidéo tout en diminuant sa résolution. On a aussi montré que la mobilité des utilisateurs pouvait améliorer les conditions de stabilité du système, selon le schéma d'ordonnancement. Nous proposons par ailleurs d'adapter le nombre des segments téléchargés dans une requête HTTP de sorte que chaque requête corresponde au même volume de données.

Enfin, nous appliquons les techniques de l'apprentissage automatique pour étudier les corrélations entre la QoE des utilisateurs et les caracteristiques du système, telles que le nombre de flux vidéos en cours, à partir d'un grand nombre de données générées par notre simulateur. Cette approche permet de mieux comprendre ces corrélations, qui ne sont pas faciles à évaluer de manière analytique. Nous pensons que cela ouvre des perspectives de recherche intéressantes sur la gestion des réseaux avec vidéo streaming.

MOTS-CLEFS: Streaming Vidéo, Streaming en Temps Réel, Streaming Adaptatif, Qualité de l'Expérience, Segment Vidéo, Modèle Niveau Flow, L'Apprentissage Automatique.

Synthèse en français

1. Introduction

Alors que la technologie 4G devient beaucoup plus mature, les services à large bande sont facilement accessibles et abordables pour tous. Avec la technologie à large bande, divers types de services qui sont difficiles à soutenir en 2G et 3G ont un taux extrêmement croissant dans les réseaux 4G. Dans le rapport Cisco, ils ont donné une prévision intéressante montrant que le trafic vidéo a déjà représenté plus de 50% du trafic mondial de données mobiles en 2015 et qu'il a un taux de croissance relativement plus élevé que les autres types de trafic. Les phénomènes que le trafic vidéo soit le trafic le plus important devient alors inévitable pour les fournisseurs de services Internet.

Selon que le contenu vidéo est généré en même temps qu'il soit regardé ou non, nous pouvons simplement diviser les services vidéo en deux catégories. Ils sont respectivement vidéo à la demande et vidéo en temps réel. **La vidéo** peut être abrégée en VoD, ce qui signifie que les programmes de radiodiffusion sur la base des besoins des utilisateurs sera sélectionnée sur la demande. Différent de la diffusion traditionnelle de télévision, les utilisateurs peuvent arrêter ou lire la vidéo à tout moment comme ils le souhaitent. Le meilleur exemple de ce type de service est fourni par YouTube, Netflix, Hulu et Dailymotion qui ont une croissance explosive et sont devenus l'un des sujets de recherche les plus populaires depuis 2005. Tout simplement parlant, le contenu vidéo de la VoD est stocké sur le *cloud* et les utilisateurs peuvent accéder à tout moment selon leur bonne vouloir. **Vidéo en temps réel** Le contenu du service de diffusion en temps réel est généré en même temps que la diffusion du contenu. Contrairement au service de streaming en mode VoD, les utilisateurs de streaming en temps réel ne peuvent pas rejouer la vidéo comme ils le souhaitent et doivent suivre l'horaire des fournisseurs de contenu vidéo. La vidéo en temps réel peut facilement être classée en deux parties. L'une d'entre elle est connue sous le nom de Live Streaming. Les meilleurs exemples de ce type de service sont tous les services Web TV comme BBC, Orange TV, BFM TV direct, etc. En outre, la vidéo en temps réel comprend également la conférence audio fournie par Skype et d'autres messages instantanés comme WhatsApp et FB Messengers. Afin de supporter différents types de services vidéo, plusieurs protocoles de communication sont proposés et nous allons les résumer ici. Certains d'entre eux sont des normes *open* et d'autres sont seulement comprises pour les usages spécifiques.

1.1 Motivation

Les objectifs de cette thèse sont la construction de plusieurs modèles de trafic streaming afin d'analyser la mesure objective de la vidéo dans les réseaux sans fils pour différents types de services vidéo. Nous commençons par prendre en compte le service de streaming en temps réel. En supposant que le streaming en temps réel a la plus haute priorité, nous vérifierons la relation entre la charge de trafic et le taux de panne de paquets (PLR). Comme la vidéo à la demande compte pour une plus grande partie du trafic réseau, notre thèse se concentre principalement sur l'étude de ce type de service et en particulier, le streaming adaptatif HTTP, en raison de la maturité de la technologie. La propriété d'adapter le débit binaire vidéo est censée fournir une liberté d'équilibrage entre le débit binaire moyen et la performance du tampon, la fluidité de la vidéo. Cependant, les impacts sur la performance des paramètres ne sont pas clairs, à la fois dans les réseaux sans fils et dans le système de livraison vidéo. Par conséquent, dans cette partie de la recherche, nous nous concentrerons sur les impacts des paramètres suivants du réseau:

-Durée du morceau vidéo
-Nombre de débits binaires vidéo
-Programmes de planification
-Mobilité des utilisateurs

En appliquant notre modèle de trafic, les opérateurs comprennent comment concevoir correctement les paramètres réseau et vidéo associés pour offrir une meilleure expérience de streaming adaptatif. Dans cette thèse, premièrement nous développerons le modèle de trafic correspondant à l'aide de la dynamique des flux et nous démontrerons ensuite les impacts de performance et proposerons des méthodes de déploiement améliorées.

1.2 Contributions

Dans cette thèse, notre principale contribution est de proposer un modèle analytique basé sur le modèle de niveau de flux pour l'évaluation de la performance vidéo dans différents scénarios. D'autres contributions détaillées serons ensuite précisées:

Dans la section 2, nous présenterons quelques connaissances de base pour cette thèse.

Dans la section 3, nous contribuerons à développer la distribution des paquets en temps réel des services de streaming en temps réel dans une cellule sans fils. Nous modéliserons une station de base (BS) en appliquant la théorie des files d'attente et en fonction de la propriété quasi-stationnaire, où nous calculerons le délai de paquets en combinant la dynamique des niveaux de paquets et des flux.

Dans la section 4, à l'aide d'un modèle de niveau de flux, nous considérons les impacts de la dynamique du trafic sur la performance du streaming adaptatif HTTP. Nous commençons par considérer la durée de fragment vidéo de façon significativement courte. Ensuite, nous étendons notre modèle de trafic au niveau de flux avec la configuration d'une durée de morceau vidéo considérablement importante. Les modèles respectifs représentent une performance extrême liée à toutes les configurations de durées intermédiaires. Ces impacts sur la performance ont été observés en calculant les indicateurs clés de performance (KPI) comme étant le débit binaire moyen de la vidéo (video bitrate) pour la résolution vidéo et

le taux moyen de déficit, le temps de service moyen et le surplus moyen de tampon pour la fluidité vidéo. Le modèle adaptatif de circulation en continu est également étendu pour intégrer les effets des conditions radioélectriques hétérogènes, des schémas de programmation et la coexistence avec le trafic élastique en général. Pour compléter les travaux, nous prendrons également en compte la mobilité intra-cellulaire dans notre modèle de niveau de flux.

Dans la section 5, nos contributions peuvent être divisées en deux parties. L'une consiste à valider notre modèle de trafic proposé pour le streaming adaptatif par simulation. L'autre consiste à examiner les répercussions sur la performance des différentes configurations de systèmes.

Dans la section 6, nous étudions la qualité vidéo de l'expérience par une autre approche, où nous appliquons la technique d'apprentissage automatique pour prédire l'une des principales métriques de qualité d'expérience (QoE), l'assèchement du tampon. Nous démontrons la performance de prédiction de différents flux HTTP et montrons que le streaming statique et adaptatif possède la plus haute précision de prédiction. Nous démontrerons également que différents paramètres de réseau ont des significations importantes et différentes pour prédire l'assèchement. En utilisant la technique d'apprentissage automatique, nous pouvons encore comprendre la relation entre les métriques de performance et les données du système lorsqu'aucun modèle mathématique exact n'est disponible. Cela donne aux opérateurs un accès permettant de comprendre en profondeur la QoE des utilisateurs.

2. Contexte

2.1 Système sans fils

La capacité de canal d'une liaison sans fils entre une paire émetteur-récepteur est limitée par des altérations dues à l'environnement, par exemple. L'affaiblissement de canal vu précédemment et par d'autres transmissions simultanées sur la même bande de fréquence voisine ou adjacente qui génèrent une interférence. Nous utilisons toujours le bruit blanc gaussien (AWGN) pour modéliser une liaison sans fils affectée par le bruit thermique, qui est due à l'agitation thermique des électrons dans les dispositifs électroniques. Avec P_u, représentant le signal reçu et I_u pour l'interférence globale perçue par un utilisateur spécifique u, la qualité du signal est déterminée par le rapport signal sur interférence et bruit (SINR) donné par:

$$SINR_u = \frac{f_u P_u}{I_u + N_0},$$

Où f_u représente l'effet de chute de canal que nous avons mentionné précédemment, habituellement il est décrit par l'Information de l'Etat du Canal (CSI). Une fois que nous avons obtenu le canal d'évanouissement d'un chemin de signal, nous pouvons calculer la capacité de canal théorique point à point en utilisant la formule de Shannon. Et nous pouvons exprimer la capacité d'un canal comme

$$R = W \log(1 + SINR_u),$$

Où W représente la bande passante du système. Sur la base de la valeur R, l'émetteur s'adaptera à un système de codage de modulation (MCS) approprié pour la transmission. Il est intéressant de mentionner que la formule de Shannon nous offre une borne supérieure pour la capacité du canal, ce qui est un résultat optimiste.

2.2 Modèle trafic

Nous présentons ici la modélisation de base du trafic élastique dans le cas des réseaux mobiles: Nous considérons un ensemble arbitraire de classes d'UE indexées par $i \in \mathscr{C}$ pour

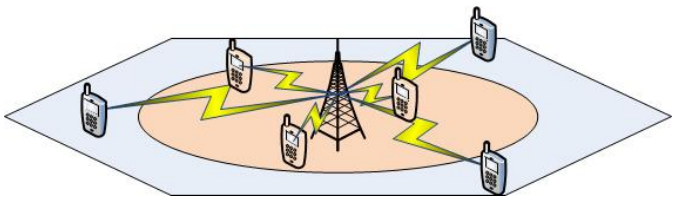


Figure 1: Une cellule typique pour des modèles en niveau flux.

refléter les différentes conditions radio, R_i (c'est-à-dire les emplacements) dans la cellule considérée. En pratique, le taux de transmission dépend de l'environnement radio et varie dans le temps en raison de la mobilité de l'utilisateur. Sauf indication contraire, nous ignorons les effets de décoloration rapide. Par conséquent, la vitesse de crête R_i dépend de la position de l'utilisateur dans la cellule. Nous supposons que le taux de transmission est constant pendant le transfert de données à moins que l'utilisateur ait une grande position change. Dans chaque classe, nous supposons que les flux de données arrivent selon un processus de Poisson avec une intensité λ dans la cellule de référence. Chaque flux reste dans le système tant que les données correspondantes n'ont pas été transmises avec succès à l'UE. On suppose que les tailles flux sont indépendantes et distribuées exponentiellement avec des bits σ moyens, bien que tous nos résultats soient sensiblement insensibles à la distribution. L'intensité du trafic est $\lambda \times \sigma$ en bit/s. Le taux d'arrivée total λ est composé du taux d'arrivée à chaque classe-i, où $\lambda_i = \lambda p_i$ et

$$\sum_{i \in \mathscr{C}} p_i = 1.$$

$X(t)$ représente le nombre d'utilisateur et suivre un processus de Markov dont le taux de transition dépend du schéma d'ordonnancement, dont nous discuterons dans la section 4. Nous supposons ici que Round Robin (RR) Programme d'ordonnancement. Les métriques de performance considérées sont le débit (en bit/s). Soit τ_i est la durée moyenne du flux de classe-i. Selon la formule de Little, $E(X_i) = \lambda_i \tau_i$ et nous avons

$$\gamma_i = \frac{\sigma}{\tau_i} = \frac{\lambda_i \sigma}{E(X_i)}. \tag{1}$$

Il s'agit du rapport entre l'intensité du trafic de la classe-i et le nombre moyen d'flux de classe-i. Cette métrique reflète l'expérience de l'utilisateur, en tenant compte à la fois des

conditions radio et de la nature aléatoire du trafic, à travers la distribution stationnaire du processus de Markov $X(t)$. Le débit moyen dans la cellule est donné par:

$$\gamma = \frac{\sigma}{\tau}, \tag{2}$$

où τ est la durée moyenne d'flux de la cellule,$\tau = \sum_{i\in\mathscr{C}} p_i\tau_i$. On a

$$\gamma = \Big(\sum_{i\in\mathscr{C}} \frac{p_i}{\gamma_i}\Big)^{-1}. \tag{3}$$

C'est la moyenne harmonique pondérée des débits d'flux par classe, avec des poids donnés par les intensités de trafic par classe. L'idée de la moyenne harmonique des débits a été proposée dans [34]. En appliquant le schéma d'ordonnancement RR, la propriété de la balance est vérifiée et le système de file d'attente peut être considéré comme un réseau de Whittle [32]. Avec la définition de charge

$$\rho_i = \frac{\lambda_i\sigma}{R_i},\ \rho = \sum_{i\in\mathscr{C}} \rho_i = \frac{\lambda\sigma}{\hat{R}}, \tag{4}$$

où $\hat{R} = \Big(\sum_{i\in\mathscr{C}} \frac{p_i}{R_i}\Big)^{-1}$. Par conséquent, la distribution stationnaire du nombre d'flux dans la cellule, $\boldsymbol{x}$ est formulé comme

$$\pi(\boldsymbol{x}) = (1-\rho)\frac{|\boldsymbol{x}|!}{\prod_{i\in\mathscr{C}} x_i!}\prod_{i\in\mathscr{C}} \rho_i^{x_i}, \tag{5}$$

avec $|\boldsymbol{x}| = \sum_i x_i$.

2.3 Apprentissage automatique

L'apprentissage automatique est un outil populaire généralement utilisé pour faire des prédictions, des décisions ou une classification basée sur une grande quantité de données. Il est largement appliqué à la reconnaissance des formes et l'intelligence artificielle, par exemple. Il est étroitement lié aux statistiques informatiques. Comme certaines métriques QoE dans notre modèle de trafic peuvent être trop compliquées pour exprimer sous une forme mathématique exacte, nous essayons d'utiliser l'apprentissage automatique pour découvrir la corrélation entre les caractéristiques du réseaux et les résultats de sortie, plus spécifiquement la QoE des utilisateurs. Dans cette section, nous présentons quelques antécédents d'apprentissage automatique utiles pour la section 6. D'une manière générale, il existe deux types d'apprentissage automatique. L'un est l'apprentissage supervisé. L'autre est l'apprentissage non supervisé. Dans cette thèse, nous nous concentrons sur l'apprentissage supervisé.

Un problème général d'apprentissage supervisé est formulé comme Figure 2 . En supposant qu'il existe m paires de données d'apprentissage. Pour les données de formation i-th, nous utilisons le vecteur $\boldsymbol{x}_i \in \mathscr{R}^n$ pour représenter les variables d'entrée, également

appelées caractéristiques d'entrée. Ici, n représente le nombre de caractéristiques dans $\boldsymbol{x}_i$. $\boldsymbol{y}_i$ est notée comme la variable de sortie ou cible que nous essayons de prédire. Une paire $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ est appelée un exemple d'apprentissage dans l'ensemble de données que nous utilisons pour apprendre est appelé ensemble d'entraînement, $\mathscr{Y} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i) : i = 1, \cdots, m\}$. $\mathscr{X} = \{\boldsymbol{x}_i\}_{i=1,\cdots,m}$ désigne l'espace des valeurs d'entrée, et $\mathscr{Y} = \{\boldsymbol{y}_i\}_{i=1,,m}$ l'espace des valeurs de sortie. Pour décrire le problème d'apprentissage supervisé un peu plus formellement,

> **L'objectif est d'apprendre une fonction $h : \mathscr{X} \to \mathscr{Y}$ de sorte que $h(\boldsymbol{x}_i)$ est un bon prédicteur pour la valeur correspondante de y_i.**

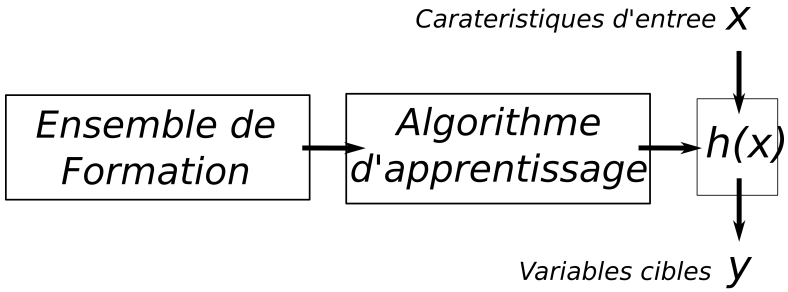


Figure 2: Diagramme d'apprentissage supervisé.

Cette fonction, h est appelée une hypothèse. Lorsque la variable cible que nous essayons de prédire est continue, par ex. $\boldsymbol{x}_i \in \mathscr{R}^n$ et $\boldsymbol{y} \in \mathscr{R}^m$, nous appelons le problème d'apprentissage un problème de régression. Lorsque y peut prendre un petit nombre de valeurs discrètes, par ex. $\boldsymbol{x}_i \in \mathscr{R}^n$ et $\boldsymbol{y} \in \{1, -1\}^m$, alors le problème est appelé **classification** problème.

3. Modèle de trafic streaming en temps réel

La qualité des services de données élastiques est principalement évalué sur le débit moyen des utilisateurs en tant que métriques de QoS. Comme nous l'avons mentionné avant, de nombreuses métriques de QoE sont proposées pour le streaming en temps réel. Pour étudier la performance du streaming en temps réel, dans les recherche comme [35], [21] and [57], les auteurs ont choisi d'autres paramètres appelés taux de blocage de flux ou taux d'indisponibilité en tant que les principales mesures de performance pour le streaming en temps réel. Sur la base de la métrique, dans [36], la performance des utilisateurs élastiques est évaluée avec la présence d'utilisateurs en streaming utilisant un modèle de niveau d'flux. Nous ne considérons ici que ceux liés à la QoS du réseau, le panne ou la perte de paquets, une métrique de QoS importante pour les services en temps réel mentionnés dans [96] et [72]. Différentes applications en temps réel ont des contraintes de délai de paquet différentes. Les paquets avec un délai supérieur à la contrainte de retard sont considérés comme inutiles. Par conséquent, les opérateurs ont besoin d'un bon modèle pour prédire les performances de retard de paquets sous une intensité de trafic donnée afin de déployer une capacité de système appropriée et de concevoir la politique de contrôle d'admission en conséquence.

Notre contribution consiste à développer un modèle de trafic pour les utilisateurs de streaming en temps réel en supposant que les utilisateurs en streaming viennent au système de façon indépendante et que les paquets générés par les utilisateurs sont servis avec une durée de service différente en fonction de leurs propres conditions de canal et leur débit binaire de la vidéo choisi. Pour le service de streaming en temps réel, la station de base ne desservira qu'un seul paquet d'utilisateur à la fois, par rapport aux données vocales, la taille du paquet en continu étant toujours suffisante pour occuper tout le RB dans un TTI. En considérant la contrainte de délai de paquets pour différents types de services de streaming, nous calculons dans la section la capacité maximale du système de streaming en temps réel sous la contrainte que 95% de paquets ont un retard inférieur à un retard d'application spécifique D. Nos autres contributions incluent:

-Développement d'un modèle de calcul de la capacité des services de streaming en temps réel compte tenu des retards des paquets.

-Proposition d'une méthode de calcul plus simple en utilisant le modèle de fluide.

-Extension et validation de notre modèle avec des effets de décoloration rapide.

-Nous utilisons la dynamique du niveau d'flux pour décrire la dynamique des utilisateurs dans le système.

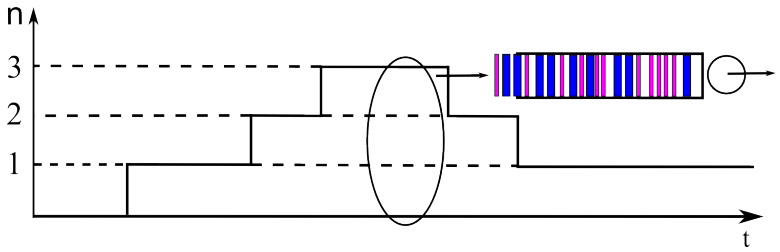


Figure 3: Schéma d'arrivée des paquets avec modélisation en deux niveaux.

3.1 Modèle avec niveaux flux et paquets

Afin d'obtenir la charge maximale du système, nous modélisons le système à deux niveaux, niveau d'flux et niveau de paquet comme Fig. 3. Au lieu d'utiliser le processus de Poisson modulé par MMPP [65] étant la processus d'arrivée, nous supposons que la dynamique du niveau d'flux se produit sur une échelle de temps relativement lente par rapport à la dynamique des paquets, une propriété indiquée dans [75]. En réalité, un service de streaming généré par un utilisateur reste dans une échelle de temps de secondes et le temps de service de paquets est toujours dans une échelle de temps de millisecondes. Par conséquent, la performance de retard au niveau du paquet atteindra approximativement une sorte d'état stationnaire entre les changements dans la population du modèle de niveau d'flux. Comme les paquets sont générés périodiquement et chaque utilisateur va générer indépendamment ses paquets avec un intervalle d'arrivée moyenne, nous modélisons le processus d'arrivée des paquets comme un simple processus d'arrivée de Markov.

Dynamique au niveau flux

Au niveau du flux, on considère d'abord une classe d'utilisateurs qui ont le même canal et nous modélisons le nombre d'utilisateurs de streaming en temps réel utilisant le chaine de Markov en temps continuous avec le taux d'arrivée, $\lambda_f = A_f^{-1}$, inverse de l'intervalle d'arrivée du flux et du débit de service, $\mu_f = S_f^{-1}$, inverse de l'intervalle de service du flux qui sont indépendants du comportement de départ et de départ de l'autre utilisateur. Sur la base de la formule d'Erlang [32], nous savons que la distribution d'état stationnaire avec des utilisateurs infinis et finis peut être exprimée comme

$$\pi_f(n) = \begin{cases} e^{-\rho_f} \dfrac{{\rho_f}^n}{n!} & \text{, when } n \in [0, \infty] \\ \dfrac{\frac{\rho_f^n}{n!}}{1 + \rho_f + \cdots + \frac{{\rho_f}^m}{m!}} & \text{, when } n \in [0, m] \end{cases} \tag{6}$$

où $\rho_f = \frac{\lambda_f}{\mu_f} = \frac{S_f}{A_f}$ représente la charge de niveau flux pour les utilisateurs streaming en temps réel.

Dynamique au niveau paquets

Sur la base du régime quasi stationnaire, chaque état n, correspondre à un nombre d'utilisateurs au niveau de flux, correspondant à un régime en niveau de paquet. Dans la file d'attente de paquets, on suppose que chaque utilisateur génèrera périodiquement ses paquets de service à intervalle fixe A_p et sera desservie par la station de base à intervalle fixe S_p. Comme chaque utilisateur va générer ses paquets de diffusion en continu périodiquement et de nombreux utilisateurs de générer les paquets à des moments différents. L'arrivée des paquets est aléatoire et nous l'approchons d'un processus de Poisson, nous utilisons la file d'attente M/D/1 pour modéliser le système de streaming en temps réel au niveau du paquet. A l'état n, nous modélisons le comportement d'arrivée des paquets en tant que processus de Poisson avec le taux d'arrivée:

$$\lambda_p(n) = \frac{n}{A_p} \tag{7}$$

Nous considérons que tous les utilisateurs appartiennent à la même condition de canal. La vitesse de départ des paquets à l'état n est indépendante de l'état n: $\mu_p(n) = {S_p}^{-1}$. Avec n utilisateurs dans le système, en utilisant les deux équations précédentes, nous définissons la charge de la file d'attente de paquets comme

$$\rho_p(n) = \frac{nS_p}{A_p} = n\rho_p, \quad \text{où } \rho_p = \frac{S_p}{A_p} \tag{8}$$

Avec la dérivation détaillée de la fonction CDF, la distribution du temps d'attente est

représentée dans l'équation (9).

$$P_n(T \leq x) = \begin{cases} 0 & , \ \rho_p(n) \geq 1, \\ (1-n\rho_p)\sum_{k=0}^{\lfloor x' \rfloor} \frac{(n\rho_p(k-x'))^k}{k!} e^{n\rho_p(k-x')} & , \ \rho_p(n) < 1, \end{cases}$$

où la function $\lfloor x \rfloor$ représente le plus grand entier inférieur ou égal à x variable et $x' = \frac{x}{S_p}$. Parce que cette équation nous donne la distribution du temps d'attente, pour obtenir la distribution du temps de réponse, il suffit de déplacer la distribution par un S_p. Dans l'hypothèse d'un système quasi-stationnaire à deux niveaux et basé sur le théorème bayésien, la distribution globale du retard, $\mathrm{P}(T \leq x)$ est le retard moyen de la distribution du retard de chaque état, n. Par conséquent, avec l'équation (6) and (9), on obtient

$$P(T \leq x) = \sum_n \pi_f(n) P_n(T \leq x). \tag{9}$$

Comme tout délai de paquet supérieur à une contrainte de délai donnée, D, est inutile pour le service sensible au retard, nous pouvons obtenir le taux de panne de paquets comme

$$\gamma(D) = P(T > D) \tag{10}$$

Compte tenu de ρ_p, de la tolérance de paquets ϵ et d'une certaine contrainte de retard D, nous sommes capables de calculer ρ_f maximum, charge système, faisant $\gamma(D) = \epsilon$.

3.2 Extension vers des conditions radio hétérogènes

Du point de vue du dimensionnement du système, les utilisateurs peuvent utiliser un taux de codec différent et peuvent avoir des conditions de canal différentes. Par conséquent, nous étendons notre modèle à des utilisateurs de classes multiples avec un modèle M/D/1 modifié et un modèle de fluide. Dans le cas de classes multiples, nous modélisons le système avec plusieurs classes d'utilisateurs ayant des temps de distribution de paquets différents. En outre, en raison de la difficulté d'obtenir la forme fermée de M/D/1 avec plusieurs classes et plusieurs fois de service, le modèle de fluide pourrait devenir un bon modèle pour faciliter le calcul.

Dynamique au niveau flux

Dans la section précédente, nous utilisons la dynamique du niveau de flux pour modéliser le nombre d'utilisateurs dans le système. En supposant qu'il existe K classes d'utilisateurs qui représentent les utilisateurs avec des canaux différentes et chaque classe $k \in \{1, \cdots, K\}$ dispose d'un service de streaming en temps réel avec taux d'arrivée Poissonien λ_k et taux de départ μ_k. Avec les deux paramètres, on dénote la charge du processeur de classe k par $\rho_k = \lambda_k/\mu_k$. On dénote le nombre d'appels d'une classe donnée demandant par $n(t)$. Streaming à l'instant t et $n(t) = (n_1(t), \cdots, n_K(t))$ désigne le nombre d'flux dans chaque

classe. Sur la base de [32], la distribution stationnaire de l'état $\pi(n)$ décrivant le nombre d'flux de chaque classe est donnée par

$$\pi(\boldsymbol{n}) = \prod_{k=1}^{K} e^{-\rho_k} \frac{{\rho_k}^{n_k}}{n_k!} \tag{11}$$

Dynamique au niveau paquets

Correspondant à différentes conditions de canal, chaque classe a son temps de service spécifique $S = \{S_1, S_2, \cdots, S_K\}$. Comme plus d'une classe d'utilisateurs coexistent dans le système, nous modifions le taux d'indisponibilité M/D/1 dans l'équation (12) avec $\rho_f = (\rho_1, \cdots, \rho_K)$.

$$\gamma_{\text{MD1,m}} = \sum_{\boldsymbol{n}} \pi(\boldsymbol{n}) P_{\boldsymbol{n}}(T \leq D, \rho_p) \tag{12}$$

En considérant plus d'un temps de service, le CDF de délai des paquets peut être calculé par le résultat numérique de la transformée de Laplace inverse obtenue dans l'équation (14), qui est également le modèle M/G/1 montré dans [61] avec multiple temps de service discret, S_k et la probabilité correspondante à $n_k/\bar{n}$.

$$\tilde{P}_{\boldsymbol{n}}(s) = \mathscr{L}\Big\{P_{\boldsymbol{n}}(T \leq x)\Big\} = \frac{\rho - 1}{\Big(\lambda - s - \lambda B(s)\Big)} \tag{13}$$

Où la fonction $B(s)$ est exprimée comme

$$B(s) = \sum_{k=1}^{K} \frac{n_k}{\bar{n}} e^{-sS_k} \tag{14}$$

et l'autres variables comme

$$\rho = \lambda\Big(\sum_{k} \frac{n_k}{\bar{n}} S_k\Big) = \frac{\sum_k n_k S_k}{A_p}, \tag{15}$$

$$\lambda = \frac{\sum_k n_k}{A_p}, \quad \bar{n} = \sum_k n_k. \tag{16}$$

3.3 Résultats de simulations

Dans cette section, nous présentons les performances du modèle M/D/1 et du modèle de fluide avec différents services correspondant à différentes configurations de contraintes de paquets délai. Basé sur [78], le délai tolérant humain pour le service interactif tel que la vidéo conférence est environ 150ms. Nous configurons les contraintes de délai en tant que 500ms pour le streaming TV en direct. Nous montrons que le modèle de fluide peut être utilisé pour simplifier le streaming TV en direct et qu'il vaut mieux rester avec le modèle M/D/1 dans le dimensionnement de la vidéo conférence.

Validation du modèle avec single classe

Dans le tableau.1, nous supposons que le temps d'arrivée du flux moyen est $S_f = 10$s qui est cent fois plus grand que le temps moyen d'arrivée des paquets $A_p = 100ms$. Basé sur la distribution du SINR obtenue par [21] et sur la configuration de la spécification 3GPP [9][8], le débit moyen LTE est calculé comme $\tau = 9.4$Mbps. Avec différents codecs, différents S_p s'appliquent. Les paramètres sont affichés dans le tableau.1, avec c désignant le codec choisi.

$$S_p = \frac{c \times A_p}{\tau} \tag{17}$$

On peut observer que $S_f, A_f >> S_p, A_p$, qui suit le régime quasi-stationnaire nous supposons.

Paramètres	Symboles	Valeur
Temps moyen d'arrivé d'un flux (s)	A_f	$[4, 20]$
Temps moyen départ d'un flux (s)	S_f	10
Temps moyen d'arrivé d'un paquet (ms)	A_p	100
Temps moyen départ d'un paquet (ms)	S_p	21.3 (2Mbps) et 10.6 (1Mbps) 5.45 (512kbps) et 2.72 (256kbps)
Nombre maximum d'utilisateurs	m	100

Table 1: Configuration de simulations pour utilisateurs avec une seul class.

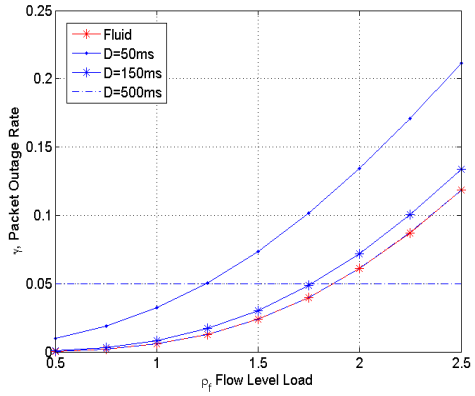


Figure 4: Taux de paquets en panne avec 2Mbps codec.

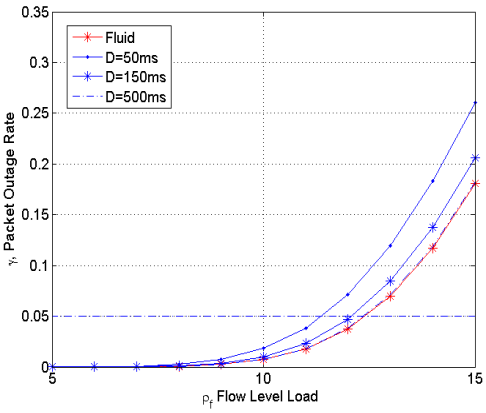


Figure 5: Taux de paquets en panne avec 512kbps codec.

Validation du modèle avec multiple classes

Pour valider l'extension de nos modèles au scénario de plusieurs classes, nous prenons un exemple d'utilisateurs avec deux classes, $S = \{S_c, S_e\}$, représentant respectivement les

utilisateurs de bordures des cellules et de centres des cellules. Dans la validation, nous supposons que les utilisateurs utilisent le codec avec un taux de codage de 512kbps. Sur la base de la même distribution SINR et de l'équation (17), nous avons calculé le débit moyen et le temps de service d'un paquet en tant que $S_c = 3.5$ms calculé par $\tau_c = 14.63$Mbps pour les utilisateurs du cell centre et $S_e = 13.73$ms calculé par $\tau_e = 3.73$Mbps pour les utilisateurs de bord de la cellule.

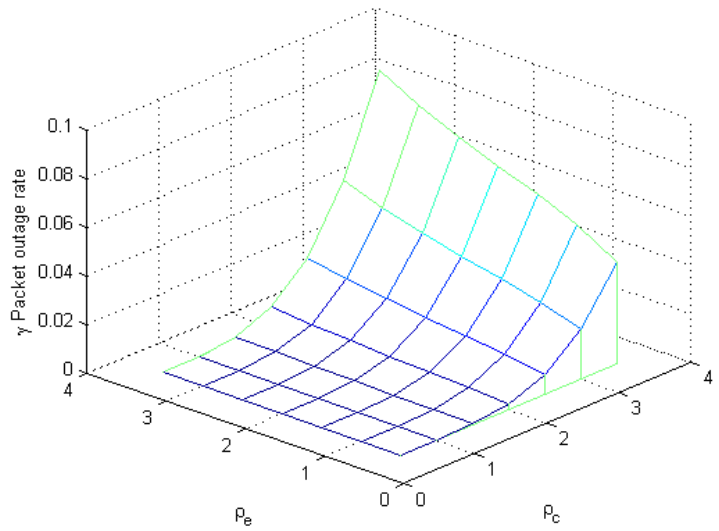
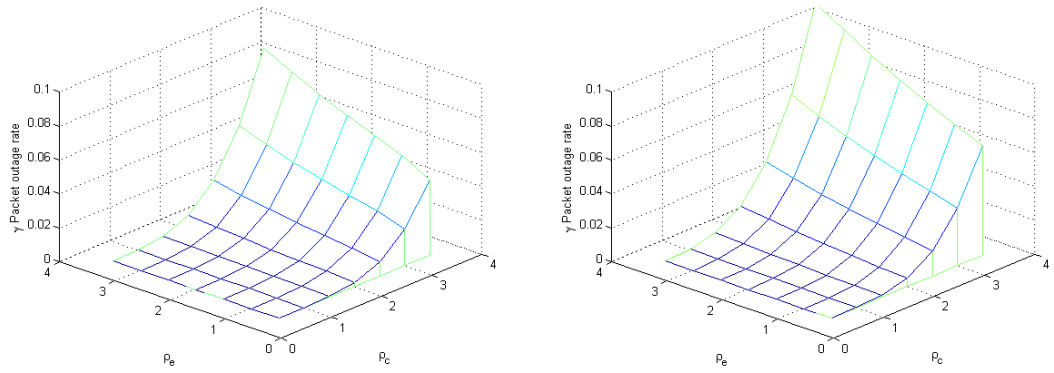


Figure 6: Taux de panne paquets calculés par modèle fluid avec les utilisateurs de multi-classes.



(a) TV Streaming en temps réel

(b) Conférence vidéo

Figure 7: Taux de panne paquets pour different applications.

4. Modèle de trafic streaming adaptatif

Dans la section précédente, nous avons présenté un modèle de trafic streaming en temps réel. Comme nous savons que la vidéo à la demande compte pour une plus grande proportion de trafic que le streaming en temps réel et le fait que les services comme YouTube et Netflix [4] devient très populaire, un modèle de trafic pour analyser l'impact de la configuration différente de streaming d'HTTP est nécessaire, en particulier Streaming adaptatif d'HTTP (HAS) devient une solution technique mature et populaire selon [71][69][44][15][45]. Comme nous l'avons mentionné dans la section introduction, la plus grande différence entre le streaming en temps réel et le streaming d'HTTP est l'existence d'un tampon de lecture vidéo. De plus, TCP est le protocole de couche de transport utilisé pour le streaming d'HTTP. Dans cette section, nous développons un modèle de trafic général qui vise à aider les opérateurs à évaluer la qualité de service perçue par leurs utilisateurs et à dimensionner correctement leurs réseaux. Nous appliquons le modèle de niveau flux bien connu, où un flux peut représenter une session de streaming vidéo ou une session élastique. Il est également appliqué pour le streaming en temps réel. Par exemple, les auteurs de [35] ont étudié l'intégration des services élastiques et de streaming en modélisant le streaming adaptatif en temps réel en tant que flux et en fournissant uniquement la performance liée car la propriété d'insensibilité ne tient pas. Toutefois, les services de streaming considérés sont modélisés comme un type spécifique de diffusion en continu, en temps réel adaptatif. Par rapport à la modélisation en temps réel en streaming, la modélisation pour le streaming HTTP est encore à ses débuts.

La plupart des travaux existants se concentrent sur l'évaluation du streaming HTTP. Il n'existe pas de modèle de trafic mature pour le streaming adaptatif HTTP et ses compromis de performances. Les modèles de niveau flux mentionnés ci-dessus se sont concentrés sur les services de streaming élastiques et en temps réel classiques et ne tiennent pas compte des impacts du tampon. Le modèle de niveau de flux a été appliqué dans des études de streaming HTTP [100] [99] avec des tampons vidéo infinis. Les KPI comme la probabilité d'assèchement du tampon ont été calculés en utilisant une analyse tampon détaillée. Cependant, le modèle mentionné n'est pas adapté pour être adapté pour évaluer la performance du streaming adaptatif HTTP en raison du manque de considération pour l'adaptabilité de débit. Les travaux de HAS incluent [102], où les auteurs proposent un cadre analytique pour le streaming adaptatif HTTP sous l'hypothèse d'une fréquence d'arrivée de cadres fixes pour différents débits binaires vidéo et [98]. Comme l'arrivée de fluide modulée par Markov. Les deux ne tiennent pas compte des répercussions de l'autre trafic et des charges globales du système. Pour d'autres études, il est préférable de combiner les impacts de l'autre trafic au débit d'arrivée des paquets, ce qui est la partie la plus difficile.**Comment allouer des ressources pour le streaming et les services élastiques devient une question pour les opérateurs**. Il est bien connu que la capacité du système sans fils peut être améliorée avec la diversité multi-utilisateurs en utilisant des ordonnanceurs opportunistes de [90] [10] [19]. Toutefois, comme le service de diffusion vidéo en continu comme YouTube et Netflix représente la plus grande partie du trafic système, s'il existe un modèle de flux adaptatif HTTP, il facilitera aux opérateurs de comprendre si ces suggestions sont toujours valables pour les services de

diffusion en continu.

Organization

On commence par introduire le contexte qui explique comment un vidéo était transmis par le système cellulaire et des paramètres systèmes. Et puis on introduit les modèles de trafic de streaming adaptatif d'HTTP avec une configuration de très petit segment de vidéo et très large segment de vidéo. Nous formulerons les KPIs pour mesurer la qualité d'espérance par rapport à la définition de vidéo et la fluidité de la vidéo aussi. Mais les extensions de modèle considérant plusieurs conditions radio, différent ordonnance, différent mobilité et l'intégration des service élastiques ne sont pas démontrés dans la synthèse française mais que dans la thèse anglais.

4.1 Description du système

Cette section présente deux aspects clés qui influent sur la performance des services de streaming adaptatif HTTP fournis dans les réseaux sans fils: la configuration du contenu vidéo et le réseau d'accès sans fils.

Configuration de contenu vidéo

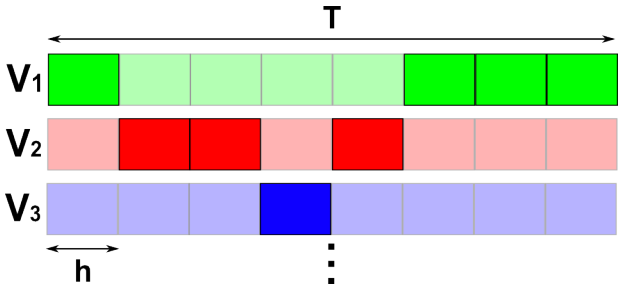


Figure 8: Un exemple de délivrer des segments vidéo pour un utilisateur avec T, durée du vidéo, v_i, débit binaire vidéo.

Selon le mécanisme de streaming adaptatif HTTP que nous avons introduit dans la première section, une vidéo est séparée par plusieurs segments vidéo (segments) et ils sont demandés les uns après les autres par des requêtes d'HTTP. Le débit binaire correspondant est sélectionné au début de chaque téléchargement. Figure. 8 donne un exemple montrant comment un téléchargement vidéo est composé par des tas de morceaux. La durée du bloc, h, est un paramètre système que les fournisseurs de services peuvent contrôler.

Intuitivement parlant, en choisissant une durée plus courte, les utilisateurs ont plus de chances d'adapter son débit binaire vidéo. Dans ce section, nous allons étudier les résultats analytiques de deux configurations de durées de blocs extrêmes illustrées dans le tableau 2 suivant.

Configurations du segment vidéo	h	Section	Transition du débit binaire
Petite segments vidéo	$h \approx 0$	4.2	Plus
Grande segments vidéo	$h \approx \infty$	4.3	Moins

Table 2: Deux extrêmes configurations pour la durée de segments vidéo.

Réseau d'accès sans fils

Dans cette section, nous nous concentrons sur la performance du streaming adaptatif livré dans une cellule typique comme Fig. 9. Les utilisateurs mobiles de la cellule téléchargent le trafic en flux continu vers leur tampon par les ressources allouées de la cellule. Nous commençons, pour la facilité de compréhension du modèle, par une condition radio homogène où tous les utilisateurs sont supposés voir une capacité R obtenu par utiliser l'état radio moyen sur la cellule. En suite, nous montreront comment étendre ces modèles à de multiples conditions radio et comment intégrer d'autres services comme le trafic élastique.

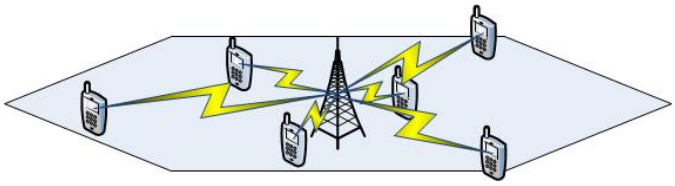


Figure 9: Système sans fils avec single région capacité.

4.2 Modèle de petite vidéo segments

Pour modéliser la dynamique du nombre de flux, $X(t)$, nous utilisons le modèle de file d'attente avec la propriété, partage de processeur. Avec l'hypothèse d'un état radio homogène et d'une petite durée de blocage, nous modélisons tous les flux dans une classe. Comme $X(t) = x$, ce qui signifie que x flux sont servis dans le système. Et le taux de départs, $\mu(x)$, peut être exprimé comme

$$\mu(x) = \frac{\phi(x)}{\sigma(x)}, \tag{18}$$

Où $\phi(x)$ représente le débit physique alloué à tous les UEs de la cellule à l'état x et $\sigma(x)$ représente la taille de flux restante à l'état x. Comme nous considérons une taille de tampon infinie du côté des utilisateurs, les utilisateurs peuvent profiter au maximum du débit qui leur est alloué, occupant ainsi le temps de programmation global de la cellule, conduisant à une allocation $\phi(x) = R$ dans ce cas. A partir de la taille de flux restante, $\sigma(x)$, étant donné que nous considérons une petite longueur de bloc conduisant à une adaptation instantanée, le débit binaire vidéo $v(x)$ à l'état x dépend seulement du nombre des flux x et

non sur l'historique du système. En outre, comme la durée vidéo est supposée d'être exponentielle, la propriété sans mémoire implique que la taille du flux à l'état x est également exponentiellement distribuée avec sa moyenne

$$\sigma(x) = v(x)T. \tag{19}$$

$X(t)$ est donc un processus de Markov dont les vitesses de départ dépendent de la sélection du débit binaire vidéo. Nous montrons dans les sections suivantes comment ce débit est calculé. La vitesse de départ du flux est obtenue comme $\mu(x) = \frac{\phi(x)}{v(x)}$. Et le système peut être facilement montré avoir une distribution stationnaire de produit-forme calculée comme

$$\pi(x) = \pi(0) \prod_{n=1}^{x} \frac{\lambda}{\mu(n)}, \tag{20}$$

Où $\pi(0) = \left(1 + \sum_{x=0}^{\infty} \prod_{n=1}^{x} \frac{\lambda}{\mu(n)}\right)^{-1}$.

Sélection du débit binaire vidéo

Le débit vidéo choisi par chaque flux est déterminé en fonction du débit instantané, $\gamma(x)$, que l'utilisateurs observent. Dans le modèle de niveau flux, en appliquant la politique d'ordonnancement round-robin, le débit instantané peut être calculé comme

$$\gamma(x) = \frac{R}{x}. \tag{21}$$

En réalité, les débits binaires vidéo ne sont pas continus. Au lieu de cela, les flux sélectionnent un débit vidéo spécifique à partir d'un ensemble de débits binaires discrets $\mathcal{V} = \{v_1, \cdots, v_I\}$, où nous supposons $v_1 > \cdots > v_I$. Connaissant le débit, les utilisateurs sélectionnent le débit binaire

$$v(x) = \begin{cases} \lfloor \gamma(x) \rfloor, & \text{when } \gamma(x) > v_I, \\ v_I, & \text{when } \gamma(x) \leq v_I, \end{cases} \tag{22}$$

Où $\lfloor z \rfloor$ représente une fonction qui sélectionne une valeur maximale dans $\mathcal{V}$ mais inférieure à z. On peut également observer que $\gamma(x)$ est toujours égal à $v(x)$ ou supérieur à $v(x)$ seulement lorsque $\gamma(x) < v_I$. Avec le mécanisme de sélection de débit vidéo défini, lorsque $\gamma(x) \geq v_I$, le tampon vidéo des utilisateurs augmente ou reste le même. Au lieu de cela, le tampon vidéo ne diminuera que lorsque $\gamma(x) < v_I$.

4.3 Modèle de grande vidéo segments

Avec la même caractéristique système mentionnée dans la section 1, nous considérons ici le système de streaming avec une durée de blocs infiniment grande, où les flux choisissent leur débit binaire vidéo au début de leur arrivée et gardent celui-ci jusqu'à la fin du téléchargement. Différent du cas de la petite quantité de morceau, où nous modélisons le

système avec une classe d'utilisateur qui choisissent le même débit binaire vidéo en même temps. Pour la configuration d'une durée de blocs infinie, plusieurs classes de files d'attente sont nécessaires pour décrire le nombre de débits qui choisissent des débits binaires vidéo différents à un moment donné. Comme dans la section précédente, l'ensemble discret des débits binaires vidéo est noté $\mathcal{V} = \{v_1, \cdots, v_I\}$, où $|\mathcal{V}| = I$.

Parce que les flux avec un débit binaire différent possèdent différents débits d'arrivée et de départ, nous désignons l'état du système comme $\boldsymbol{x} = (x_1, \cdots, x_I) \in N^I$, Où x_i représente le nombre de flux qui a choisi le débit binaire vidéo v_i à son arrivée. Le débit d'arrivée du flux-i dépend du nombre total d'flux dans le système, $|\boldsymbol{x}| = \sum_i x_i$. Étant donné un état $\boldsymbol{x}$, l'arrivée du flux ne sélectionnera que le débit binaire v_i. Par conséquent, nous avons

$$\forall i,\ \lambda_i(\boldsymbol{x}) = \begin{cases} \lambda, & \text{if } v(\boldsymbol{x} + \boldsymbol{e}_i) = v_i, \\ 0, & \text{otherwise,} \end{cases} \tag{23}$$

Où $v(\boldsymbol{x})$ est définie comme Eq. (22) et $\boldsymbol{e}_i$ représente un vecteur avec une valeur unitaire à la classe i. Appliquer le même concept de l'équation (18), le débit de sortie du flux et la ressource allouée de la classe i est noté

$$\mu_i(\boldsymbol{x}) = \frac{\phi_i(\boldsymbol{x})}{v_i T}, \tag{24}$$

$$\phi_i(\boldsymbol{x}) = \frac{x_i R}{|\boldsymbol{x}|}, \tag{25}$$

Avec le même réglage de Round-Robin ordonnancement. On peut alors calculer la distribution stationnaire $\pi(\boldsymbol{x})$ en utilisant le taux d'arrivée et de départ des deux cas dans les équations (23 - 24) et en résolvant les équations d'équilibre notées

$$\begin{aligned} \forall \boldsymbol{x},\ & \sum_i \Big(\lambda_i(\boldsymbol{x}) + \mu_i(\boldsymbol{x})\Big)\pi(\boldsymbol{x}) \\ & = \sum_i \lambda_i(\boldsymbol{x} - \boldsymbol{e}_i)\pi(\boldsymbol{x} - \boldsymbol{e}_i) + \mu_i(\boldsymbol{x} + \boldsymbol{e}_i)\pi(\boldsymbol{x} + \boldsymbol{e}_i). \end{aligned} \tag{26}$$

Avec la distribution stationnaire, $\pi(\boldsymbol{x})$ calculée dans le cas d'une petite quantité de morceaux et d'une grande quantité de morceaux, nous définissons ensuite les principaux indicateurs de performance dans la section suivante.

4.4 Condition stabilité

Le débit d'arrivée du flux doit être inférieur au débit de départ maximum. Dans le cas où v_n est la sélection du débit binaire vidéo de n-ième flux, la vitesse de départ du débit maximal implique que $\forall n, v_n = v_I$, conduisant à la condition de stabilité suivante pour les deux configurations de durée de fragment et taux d'arrivée maximum, $\lambda_{\max}$:

$$\lambda < \frac{R}{v_I T} \Rightarrow \lambda_{\max} = \frac{R}{v_I T}. \tag{27}$$

Si $v_I = 0$, le système est toujours stable. Cependant, lorsque $v_I > 0$, le système devient instable quand il y aura un grand nombre d'arrivées.

4.5 Définition des KPIs

Pour évaluer le QoE du service de streaming adaptatif, nous proposons quatre indicateurs de performance clés, le débit binaire moyen, le temps de service moyen, le taux de déficit et le surplus de tampon. Tous sont définis sur la base de la distribution stationnaire $\pi(\boldsymbol{x})$.

Débit binaire de la vidéo

Le débit binaire moyen correspond au débit binaire moyen d'un flux lors de la lecture de la vidéo. Lorsque le débit vidéo moyen est élevé, l'utilisateur a une meilleure expérience vidéo. Nous calculons le débit binaire moyen de la cellule en utilisant le concept suivant,

$$\text{Débit binaire de la vidéo} = \frac{\text{Resources allouées}}{\#\text{Flux servi}}, \tag{28}$$

où nous divisons toute les ressources allouées sur le nombre de flux multiplié la durée moyenne de la vidéo pour calculer le débit binaire moyen. Ensuite, nous définissons le débit binaire global moyen, $\bar{\nu}$ pour une durée de fragment petite et grande,

$$\bar{\nu} = \frac{\sum_{x:x>0} \pi(x)\phi(x)}{\lambda T}, \quad \bar{\nu} = \frac{\sum_{\boldsymbol{x}:|\boldsymbol{x}|>0} \pi(\boldsymbol{x}) \sum_i \phi_i(\boldsymbol{x})}{\lambda T}, \tag{29}$$

où $|\boldsymbol{x}| = \sum_i x_i$.

Un indicateur populaire QoE utilisé pour évaluer la performance en streaming est la probabilité d'assèchement du tampon [100]. Même si la pause de la vidéo ne se produit que lorsque le débit binaire vidéo est plus grand que le débit instantané, cette dernière condition n'est pas une condition suffisante pour la pause de la vidéo car le tampon peut contrecarrer l'impact de courtes périodes de faible débit. Le calcul de la probabilité de d'assèchement du tampon doit prendre en compte la mémoire du système en introduisant la taille du tampon dans l'analyse markovienne comme dans [100]. Ici, nous introduisons et examinons trois KPIs appelé temps de service, taux de déficit et excédent de tampon.

Temps de service

Pour évaluer la probabilité d'assèchement, nous proposons le première KPI, temps de service moyen d'un flux vidéo, qui est calculé par la formule de Little comme

$$S = \frac{L}{\lambda} = \frac{\bar{x}}{\lambda}, \quad S = \frac{L}{\lambda} = \frac{\bar{\boldsymbol{x}}}{\lambda}, \tag{30}$$

Avec $\bar{x} = \sum_{x>0} x\pi(x)$, $\bar{\boldsymbol{x}} = \sum_{|\boldsymbol{x}|>0} |\boldsymbol{x}|\pi(\boldsymbol{x})$ et que $L = \bar{x}$ représente le nombre moyen de flux. En observant S, nous pouvons impliquer la probabilité de vacuité, qui est positivement liée à S. Dire que le plus petit S pourrait impliquer une plus petite probabilité de vacuité.

Taux de déficit

Comme [32] mentionné, les flux ont une probabilité plus élevée de rester dans l'état pour le téléchargement $v(x)$ parce que x utilisateurs existent. Par conséquent, en pondérant les métriques correspondantes à l'état x par le nombre de flux, x, également appelé distribution de biais de taille, nous définissons les paramètres suivants. Le taux de déficit est égal à la probabilité qu'un flux voit son débit instantané est inférieur à son débit binaire vidéo choisi. Comme on suppose que l'adaptation du débit binaire vidéo se produit instantanément en réaction aux variations du débit observé, le taux de déficit est défini par la probabilité que le débit instantané,

$$\gamma(x) = \frac{\phi(x)}{x}, \tag{31}$$

est inférieur à $v(x)$ dans le cas d'une petite durée de fragment vidéo ou

$$\gamma_i(x) = \frac{\phi_i(x)}{\sum_i x_i}, \tag{32}$$

Est inférieur à v_i dans le cas d'une longue durée de fragment vidéo. Notez que pour la configuration de la petite taille de la section, le déficit ne se produit que lorsque $\gamma(x) < v_I$, basé sur le mécanisme de sélection, Eq. (22). Le taux de déficit global est défini en pondérant la distribution stationnaire à différents états x avec le nombre de flux:

$$D = \Pr\{\gamma(x) < v(x)\} = \sum_{x:x>0} \frac{x\pi(x)}{\bar{x}} \mathbf{1}_{\{\gamma(x)<v(x)\}}, \tag{33}$$

Où $\mathbf{1}$ représente la fonction indicateur. Pour le cas de configuration de la durée de gros morceau vidéo,

$$D = \Pr\{\gamma_i(\boldsymbol{x}) < v_i(\boldsymbol{x})\} = \sum_{\boldsymbol{x}:|\boldsymbol{x}|>0} \frac{\pi(\boldsymbol{x})}{\bar{\boldsymbol{x}}} \sum_i x_i \mathbf{1}_{\{\gamma_i(\boldsymbol{x})<v_i\}}. \tag{34}$$

La probabilité de vacuité est aussi positivement liée au taux de déficit. Par conséquent, un taux de déficit plus important entraînera une plus grande probabilité de vacuité.

Surplus de tampon

Nous introduisons également une autre métrique de performance appelée excédent de tampon, qui représente la variation moyenne de tampon de chaque débit en une seconde. Il est calculé en pondérant toute la variation du tampon $\frac{\gamma(x)-v(x)}{v(x)}$, à chaque état x as

$$B = \sum_{x:x>0} \frac{x\pi(x)}{\bar{x}} \Big(\frac{\gamma(x)-v(x)}{v(x)} \Big), \tag{35}$$

Pour la petite configuration de durée de morceau vidéo. Lorsque $\gamma(x) > v(x)$, le tampon des utilisateurs accumule une certaine durée de la vidéo. Lorsque $\gamma(x) < v(x)$, l'utilisateur

commence à consommer les paquets vidéo stockés dans la mémoire tampon, ce qui réduit les valeurs de la mémoire tampon moyenne excédentaire. Pour le cas d'une longue durée, le surplus de tampon est calculé comme

$$B = \sum_{\boldsymbol{x}:|\boldsymbol{x}|>0} \frac{\pi(\boldsymbol{x})}{\bar{\boldsymbol{x}}} \sum_i x_i \Big(\frac{\gamma_i(\boldsymbol{x}) - \nu_i}{\nu_i} \Big). \tag{36}$$

Un plus grand surplus de tampon diminuera la probabilité de vacuité. Par conséquent, il est négativement lié à la probabilité de vacuité.

5. Simulation de la performance streaming adaptatif

5.1 Organization

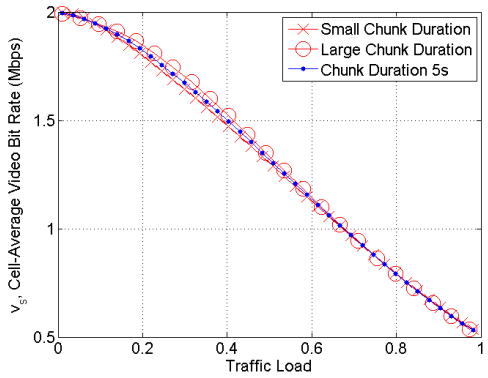
Après avoir présenté le modèle de flux adaptatif basé sur HTTP dans la section précédente, nous commencerons par montrer les impacts de différentes configurations de système soit dans le système de distribution vidéo soit dans les réseaux d'accès sans fils, y compris la durée du segment vidéo, le nombre de débits binaires vidéo disponibles et différent ordonnancements. Nous démontrons également l'impact sur le rendement de la mobilité des utilisateurs.

Nous ne présentons pas le modèle d'approximation qui peut réduire la complexité du calcul ici dans la synthèse française. Dans la partie anglais, nous appliquons le modèle d'approximatif pour illustrer comment utiliser notre modèle pour le dimensionnement du réseau et étudier les impacts de la limitation du débit binaire vidéo sur la performance en streaming.

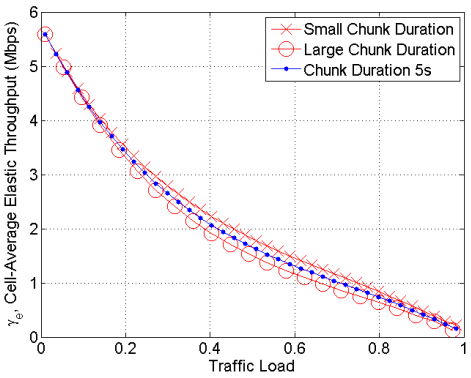
5.2 Impacts de la durée du segments vidéo

Selon le rapport technique d'Akamai [1] et la démonstration empirique de [101], ils sont démontré que le déploiement d'une durée de segment plus courte offre une meilleure fluidité de la vidéo et plus de chances pour les utilisateurs de sélectionner la bonne débit binaire vidéo. Cependant, le déploiement de segments plus courts générera également un grand nombre de segments vidéo et sa signalisation d'HTTP correspondante. Par conséquent, le rapport suggère que les fournisseurs de services vidéo choisissent leurs configurations de durée de segment d'HTTP adaptatif en continu en 10 secondes. Dans cette section, nous effectuons d'abord des simulations pour vérifier ces résultats en définissant les paramètres comme $\mathscr{R} = \{R_C, R_E\} = \{10, 4\}$Mbps, $(p_C, p_E) = (\frac{1}{2}, \frac{1}{2})$ pour les utilisateurs de cell centre et les utilisateurs au bord du cellule. $(p_e, p_s) = (\frac{1}{2}, \frac{1}{2})$ représente les taux d'arrivés des flux élastiques et streaming. $\mathscr{V} = \{\nu_1, \nu_2\} = \{2, 0.5\}$Mbps, $T = 10$s, $\sigma = 5$Mbits et $\lambda_{\max} = 1.14$ flux/s. Les résultats de simulation dans les figures suivantes montrent la performance de toutes les métriques définies par rapport aux paramètres normalisés de la charge de trafic, $(\lambda/\lambda_{\max})$.

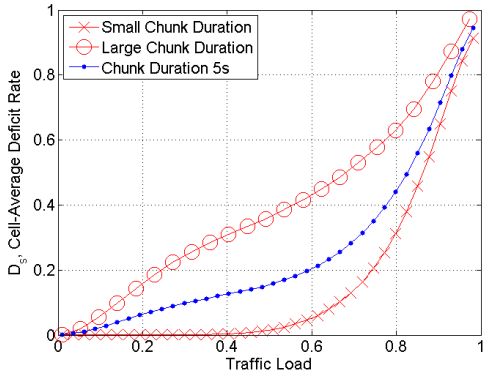
Comme la section 4.1 mentionné, les résultats de simulation montrés dans la figure 10 nous donnent deux limites de performance des deux cas extrêmes, $h = 0$ and $h = \infty$. Ce



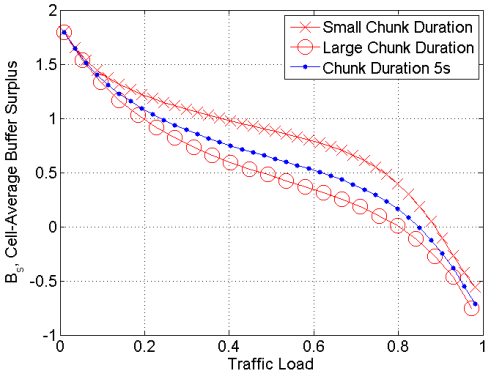
(a) Débit binaire de la vidéo, v_s.



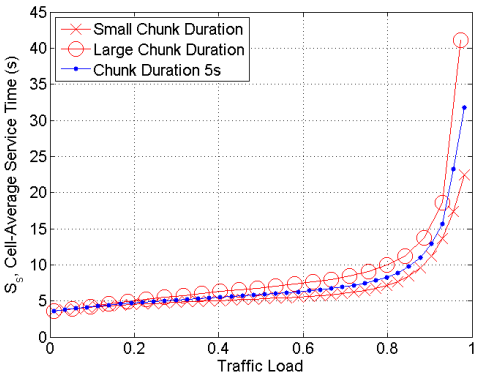
(b) Débit moyenne élastiques, γ_e.



(c) Taux de déficit, D_s.



(d) Surplus de tampon, B_s.



(e) Temps de service, S_s.

Figure 10: Performances de streaming vidéo obtenues par les deux modèles avec deux durées de chunk.

qui nous permet de prédire la performance de la durée intermédiaire du morceau. Dans la simulation, nous démontrons en outre la performance d'une configuration intermédiaire, $h =$

5s entre deux cas extrêmes. Dans les figures 10a et 10b, nous trouvons que la configuration de petits segment a un débit binaire moyen plus faible par rapport au cas d'une grande durée de segment. Figure. 10b nous apprend que le déploiement d'une petite durée peut bénéficier un débit moyen élastique. En termes de performance du tampon, nous observons que le déploiement de la petite quantité de segment entraînera un meilleur taux de déficit dans la Fig. 10c et que le déploiement d'une petite durée de fragmentation résultera en un meilleur excédent de tampon dans la Fig. 10d. Nous obtenons également le même résultat en termes de temps de service moyen dans la Fig. 10e. Ces résultats montrent que la configuration d'une long durée des segments vidéo peut augmenter la satisfaction des utilisateurs dans le sens de la résolution, mais pas dans le sens de la fluidité vidéo. Nous proposons une méthode pour réduire la signalisation HTTP tout en conservant les mêmes performances avec l'idées de transmettre différents nombre de segments dans une requête d'HTTP. Les résultats plus détaillés sont présentés dans la thèse suivante en anglais.

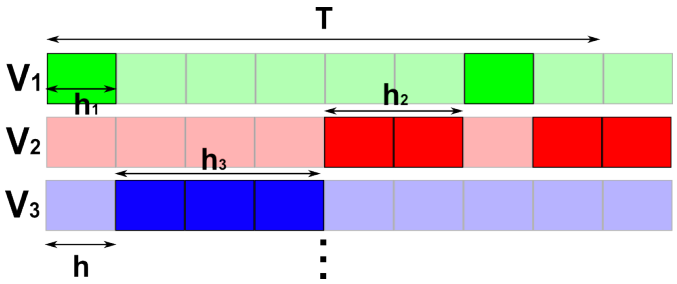


Figure 11: Un exemple démontrant comment télécharger plusieurs segments dans une requête d'HTTP.

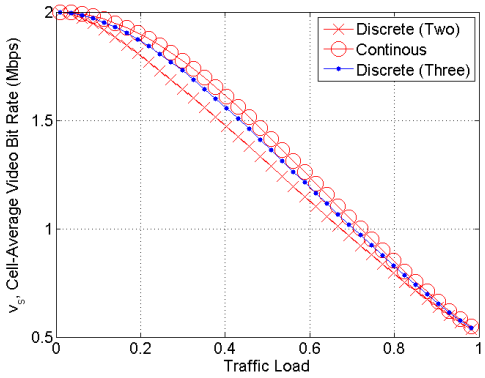
5.3 Impacts du nombre de débits binaires vidéo

Dans la simulation précédente, l'ensemble débit binaire vidéo est configuré comme $\mathcal{V} = \{v_1, v_I\}$. Cependant, une question générale est de savoir quels sont les impacts si plus de débits binaires vidéo sont disponibles pour être choisis dans $\mathcal{V}$? Pour répondre, nous configurons un continu ensemble de débit binaire vidéo $\mathcal{V}_{\text{cont}} = \{v : v_1 \leq v \leq v_I\}$ avec la configuration de petite durée de morceau vidéo. Dans ce cas, $|\mathcal{V}_{\text{cont}}| = \infty$. Différent de l'équation (22), les flux des utilisateurs à la région de capacité k sont supposés pouvoir sélectionner n'importe quel débit binaire vidéo entre v_1 et v_I, exprimé en

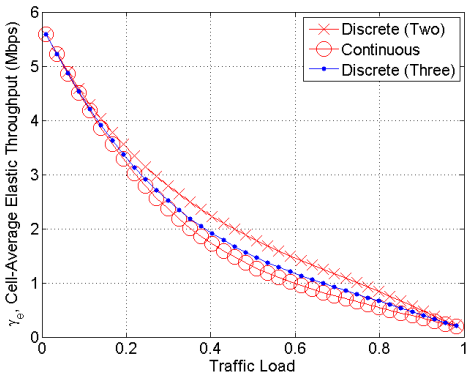
$$v_k(\boldsymbol{x}) = \max\Big(\min\Big(\gamma_k(\boldsymbol{x}), v_1\Big), v_I\Big), \tag{37}$$

où $\gamma_k(\boldsymbol{x})$ est défini comme l'équation (31) et (32). Basé sur la définition de taux de départ, le taux de départ du flux devient

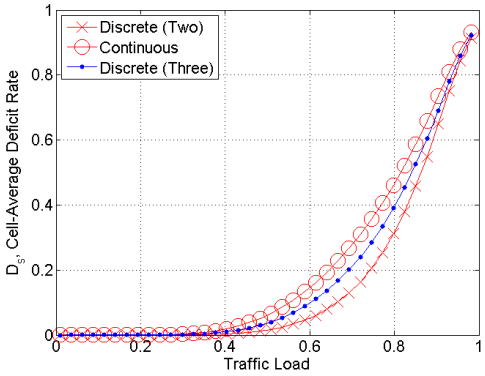
$$\mu_k(\boldsymbol{x}) = \frac{\phi_k(\boldsymbol{x})}{v_k(\boldsymbol{x})T} = \frac{R}{\max\Big(\min\Big(\gamma_k(x), v_1\Big), v_I\Big)T}. \tag{38}$$



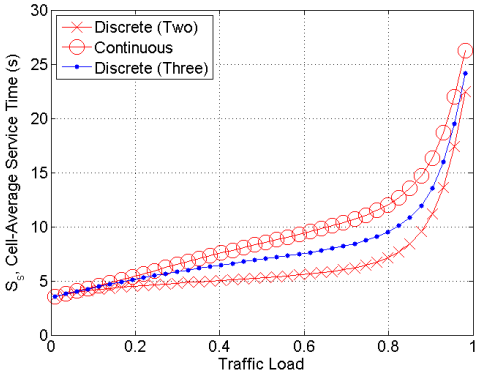
(a) v_s moyenné dans un cellule.



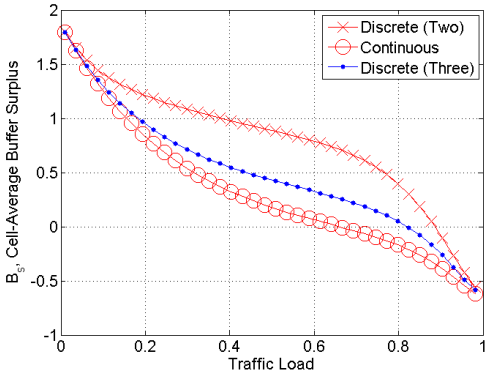
(b) γ_e moyenné dans un cellule.



(c) D_s moyenné dans un cellule.



(d) S_s moyenné dans un cellule.



(e) B_s moyenné dans un cellule.

Figure 12: Performances du streaming adaptatif avec un débit binaire vidéo continu et discret.

Lorsque le débit disponible $\gamma_k(\boldsymbol{x}) = \frac{R_k}{|\boldsymbol{x}|}$ est supérieur à v_1, le tampon commence à remplir sur la mesure que la capacité de téléchargement est plus grande que la vitesse de lecture. Le

flux se comporte donc comme un flux élastique car il n'y a pas de limitation sur la taille du tampon. D'autre part, lorsque $\gamma_k(\boldsymbol{x})$ est inférieur à v_I, le comportement est aussi comme des flux élastique mais avec une faim possible si le tampon se vide en raison du taux de diffusion constant. Entre v_1 et v_I, le flux se comporte comme un temps réel et la taille du tampon reste constante. Résoudre les équations du bilan comme Eq. (20), on obtient les résultats de simulation représentés sur la Fig. 12 avec trois configurations différentes dont 2 débits binaires vidéo, un nombre infini de débits binaires vidéo et un réglage intermédiaire, 3 débits binaires vidéo. On peut observer que les mesures de performance de 3 débits binaires vidéo sont situées entre celles des deux cas extrêmes.

De la Fig. 12a, nous observons que le déploiement de débit vidéo continu, ce qui signifie plus de débits binaires vidéo pour approcher le débit instantané réel des utilisateurs, augmentera la résolution vidéo à long terme mais diminuera le débit élastique moyen. De plus, du reste des figures, nous découvrons que le déploiement de plus de débits binaires vidéo augmentera également la faim des tampons à partir des trois mesures liées au buffer. Par conséquent, le déploiement d'un plus grand nombre de débits binaires vidéo générera un compromis qui peut bénéficier à la résolution vidéo moyenne mais diminue le débit élastique et la lisibilité vidéo.

5.4 Résumé

Dans cette section, les résultats de la simulation montrent que la configuration d'une grande longueur de bloc n'améliore pas la résolution vidéo mais diminue largement la lisibilité de la lecture et que la configuration d'un plus grand nombre de débits vidéo fournira également une meilleure résolution vidéo mais dégradera la fluidité vidéo. Dans la synthèse française nous avons montré l'impacts de la durée de segments vidéo et le nombre de débit binaire vidéo. Dans la thèse anglais nous montrons également le compromis de performance de différents schémas d'ordonnancement avec et sans tenir compte de la mobilité des utilisateurs.

6. Prediction de la QoE de streaming video avec technique apprentissage automatique

6.1 Organization

Dans les sections précédents, nous avons étudié les performances du streaming en temps réel et du streaming d'HTTP. Pour mesurer le QoE du streaming basé sur HTTP, les métriques, la probabilité de *assèchement du tampon* ou les événements de tampon sont adoptés populairement. *Assèchement du tampon* ou les événements de tampon se produisent lorsque le tampon vidéo devient vide et les utilisateurs rencontrent une pause vidéo. Comme il n'est pas facile de développer une forme mathématique et analytique pour les métriques de QoE, en particulier pour le subjectif QoE, des recherches comme [86] appliquent une analyse statistique pour comprendre la corrélation entre QoE et réseau QoS. Dans cette section, nous nous appuyons sur un simulateur qui génère une grande quantité de paires de données (QoE +

caractéristiques réseau) et nous démontrons l'efficacité de la prévision de la QoE, la vacuité de la vidéo, en utilisant les caractéristiques du réseau d'entrée telles que CSI, Le nombre de flux d'utilisateurs et la durée de la vidéo, enregistrés à l'arrivée de chaque utilisateur.

La probabilité de vacuité est étudiée en tant que QoE de service vidéo dans de nombreux travaux. Il est modélisé et calculé analytiquement dans [100]. Cependant, plusieurs contraintes sont nécessaires lors de l'application de ce modèle, par exemple, un débit binaire fixe est requis et le modèle ne peut prendre en compte qu'une seule condition de canal. Dans [98], la forme analytique de la probabilité de vacuité du streaming adaptatif est proposée sans tenir compte de la dynamique du flux. Nous avons utilisé un modèle de niveau d'flux pour étudier les métriques de performance vidéo dans [30] et [66]. Cependant, la relation entre la faim vidéo et les mesures de performance proposées ne sont pas claires. L'apprentissage par machine a été largement utilisé pour étudier à la fois subjective et objective QoE. Notre contribution est de démontrer la corrélation entre la faim vidéo de différemment streaming et les fonctionnalités des utilisateurs enregistrés. Nous analysons également l'importance des fonctionnalités de ces utilisateurs pour la prédiction des événements de privation de la vidéo, y compris le nombre d'utilisateurs vidéo existant dans une cellule, les conditions de canal de l'utilisateur vidéo et la durée de la vidéo enregistrée lorsqu'un flux lance son téléchargement vidéo.

6.2 Description du système

Dans cette section, nous présentons d'abord deux types de flux vidéo. Ensuite, nous introduisons le modèle de niveau d'flux utilisé pour calculer la charge maximale du système. Un simulateur piloté par événement est présenté pour générer des données de la vacuité vidéo pour différentes charges.

Streaming avec un débit binaire vidéo fixé

Les utilisateurs fixent un débit binaire vidéo, v_c, depuis le début jusqu'à la fin du téléchargement vidéo, ce qui est la plus simple implémentation.

Streaming adaptatif

Comme la Fig. 6.1, les services adaptatifs de streaming permettent aux utilisateurs de s'adapter en temps réel à leur débit vidéo. Après avoir fini de télécharger un morceau vidéo avec une durée h, basé sur le débit mesuré, γ_j, les utilisateurs peuvent sélectionner un débit binaire vidéo pour le bloc suivant à partir d'ensemble discret $\mathcal{V} = \{v_1, \cdots, v_I\}$, où $v_1 > \cdots > v_I$. Le débit vidéo sélectionné est donné

$$v_j = \begin{cases} \lfloor \gamma_j \rfloor, & \text{when } \gamma_j \geq v_I, \\ v_I, & \text{when } \gamma_j \leq v_I, \end{cases} \tag{39}$$

où $\lfloor y \rfloor$ est une notation abrégée pour le plus grand débit binaire vidéo en $\mathcal{V}$ mais pas supérieur à y et γ_j représente le débit instantané de l'utilisateur j. Après le téléchargement, les morceaux vidéo seront stockés dans un tampon de lecture. Comme les sections

précédentes, nous supposons encore que la mémoire tampon de lecture des utilisateurs est infinie. Une fois qu'un utilisateur entre dans la cellule, il sera programmé une quantité de ressource jusqu'à la fin du téléchargement de la vidéo.

Réseau d'accès radio et caractéristiques du trafic

Nous considérons un réseau d'accès radio où les utilisateurs ont un débit physique différent en fonction de son emplacement dans la cellule. Nous considérons l'ensemble du débit physique comme $\mathscr{R} = \{R_1, \cdots, R_K\}$ et supposons que le débit physique d'un utilisateur statique reste le même pendant le transfert du flux de données entier (La session vidéo). Pour les utilisateurs mobiles, le débit physique varie entre ces valeurs.

En ce qui concerne les caractéristiques de trafic, nous ne considérons que les services de streaming dans ce modèle et nous faisons l'hypothèse classique que les flux de données ayant un débit physique R_k arrivent comme un processus de Poisson avec le débit $\lambda_k = p_k\lambda$, où λ est le débit global d'arrivée du flux dans la cellule et p_k représente la proportion de trafic avec le débit physique R_k, avec $\sum_k p_k = 1$. La durée des requêtes est supposée indépendante et distribuée de manière exponentielle avec la moyenne T. Le streaming simulé peut être divisé en quatre types suivants et chacun d'eux compte pour w_i proportion d'arrivées avec $\sum_i w_i = 1$, où $i \in$ {I, II, III, IV}.

-Type I: streaming statique et adaptatif.

-Type II: streaming mobile et adaptatif.

-Type III: streaming statique et avec fixe débit binaire vidéo.

-Type IV: streaming mobile et débit fixe. Dans le réseau réel, le taux d'arrivée du trafic, λ, varie le long des heures, généralement plus élevé en jour et plus bas la nuit. Dans les résultats de simulation suivants, les données de privation de la vidéo sont générées avec des taux d'arrivée de trafic différents. La prévision montrée à chaque charge de trafic correspond à la performance potentielle à chaque heure.

Modèle de flux et taux d'arrivée maximal

Le concept de modèle de niveau flux a été utilisé pour obtenir la performance du streaming dans le papier [30] et [66]. Basé sur ce modèle, nous pouvons obtenir le débit maximum d'arrivée de flux qui garantit la stabilité de système. Soit $\boldsymbol{x}(t) = (x_1^1(t), \cdots, x_K^i(t))$ le nombre de flux en continu à l'instant t et $x_k^i(t)$ répresente le nombre de type-i streaming avec R_k à l'instant t. Basé sur la méthode d'ordonnancement à tour de rôle, nous avons le débit instantané calculé comme

$$\gamma_k^i(\boldsymbol{x}) = \frac{R_k}{|\boldsymbol{x}|}, \tag{40}$$

où $|\boldsymbol{x}| = \sum_i \sum_k x_k^i$ est le nombre total de flux en cours. Le débit binaire vidéo du segment vidéo suivant est sélectionné sur la base de l'équation (22) et (40). Lorsque la charge se rapproche de la capacité du système et des caractéristiques de trafic mentionnées, tous les flux adaptatifs sont forcés de s'adapter à v_I, le débit maximum d'arrivée du système peut

être calculé en traitant tous les flux comme un trafic élastique avec le volume vT as

$$\lambda_{\max} = \Big(\frac{w_1 v_I T}{R_s} + \frac{w_2 v_I T}{R_m} + \frac{w_3 v_c T}{R_s} + \frac{w_4 v_c T}{R_m} \Big)^{-1}. \tag{41}$$

$R_s = \Big(\sum_k \frac{p_k}{R_k} \Big)^{-1}$ représente le débit radio équivalent pour les utilisateurs statiques. Comme l'indique le travail [73], $R_m = \sum_k q_k R_k$ représente le débit radio équivalent pour les utilisateurs mobiles avec q_k désigné comme la proportion de temps que les utilisateurs restent avec le débit R_k.

Simulateur événementiel

Dans [30] et [66], par modèle mathématique de niveau flux, nous ne pouvons obtenir que des métriques objectives de QoS au lieu des informations de tampon réel de j-th utilisateur, $b_j(t)$. Par conséquent, nous implémentons un simulateur piloté par événement qui est capable de simuler l'événement de la privation vidéo et d'enregistrer la valeur de la mémoire tampon de tous les utilisateurs. Notre simulateur est implémenté en fonction du concept de niveau d'flux, où chaque session vidéo est considérée comme un flux et chacun d'entre eux peut rencontrer les événements suivants:

- **Événement d'arrivée**: L'arrivée de flux de flux suit la distribution de Poisson. Comme l'utilisateur j arrive à la cellule au moment E_j^a, plusieurs caractéristiques observées, $\boldsymbol{z}_j$, sont enregistrées et utilisées comme données d'entrée pour prédire la vacuité, $y_j \in \{1, -1\}$.

- **Événement de départ**: Les utilisateurs rencontrent un événement de départ au moment E_j^d lorsque la vidéo demandée est complètement téléchargée.

- **Événement Chunk**: Les utilisateurs rencontrent un événement chunk au moment E_j^c lorsque la partie vidéo demandée avec la durée h est téléchargée.

- **Événement Tampon**: Nous classons l'utilisateur de streaming simulé j en trois états, PREFETCH, PLAY et STARVATION(assèchement du tampon), où chacun a un taux de variation de tampon

$$\frac{d\, b_j(t)}{dt} = \begin{cases} \gamma_j, & \text{PREFETCH ou STARVATION,} \\ \gamma_j - 1, & \text{PLAY.} \end{cases}$$

  Lorsque l'utilisateur commence à demander une vidéo, il restera à PREFETCH et passera à STARVATION jusqu'à $b_j(t) \geq B$, où B est le tampon initial. Une fois $b_j(t) = 0$, l'utilisateur entre dans STARVATION, où il est reconnu comme un utilisateur connaissant une vacuité de la vidéo, $y_j = 1$, et il attendra que $b_j(t) \geq B$ pour entrer à nouveau PLAY. E_j^b est le temps des événements de tampon pour l'utilisateur j.

- **Événement de mobilité**: Les utilisateurs avec mobilité changeront le R_j pour le débit adjacent lorsque l'événement de mobilité au moment E_j^m se produit. Dans ce cas, les utilisateurs planifient l'événement de mobilité suivant en fonction de la distribution exponentielle avec le taux ν_j.

Caractéristiques enregistrées

Nous avons mentionné que $\boldsymbol{z}_j$ est enregistré à l'arrivée de j-th utilisateur et va être utilisé pour prédire y_j. Ici, nous présentons les composants de la fonction de l'utilisateur, $\boldsymbol{z}_j$:

R_j	Condition radio(R)	Il est enregistré au début du téléchargement de la vidéo. Si l'utilisateur est statique, R_j est fixe.
T_j	Durée vidéo (T)	Il suit une distribution exponentielle.
$\boldsymbol{x}_j$	Nombre de flux (N)	$\boldsymbol{x}_j = (x_1, \cdots, x_K)_j$ représente le nombre de flux dans chaque région.
$\lvert\boldsymbol{x}_j\rvert$	Nombre total de flux	$\lvert\boldsymbol{x}_j\rvert = \sum_k x_k$ représente le nombre total de flux dans la cellule.
$\boldsymbol{x}_j^s$	Nombre de flux en pause (N_s)	$\boldsymbol{x}_j^s = (x_1^s, \cdots, x_K^s)$ répresente le nombre flux qui subissent la pause vidéo dans chaque région.
$\lvert\boldsymbol{x}_j^s\rvert$	Nombre total de flux en pause	$\lvert\boldsymbol{x}_j^s\rvert = \sum_k x_k^s$ représente le nombre total de flux en pause.

6.3 Outil d'apprentissage automatique

Comme on a déjà introduit les outils d'apprentissage automatique dans la section 2.3, ici nous allons seulement présenter les indicateurs des performances et les libraries relatives.

Indicateurs des performances

Afin de vérifier les performances d'apprentissage de la machine, nous définissons les mesures de performance suivantes pour examiner les performances de prédiction parmi les données de test. Cette probabilité représente la précision moyenne de prédiction parmi tous les échantillons testés.

$$P = P_{\{y_j=-1\}} P_{\{\hat{y}(\boldsymbol{z}_j)=-1 | y_j=-1\}} + P_{\{y_j=1\}} P_{\{\hat{y}(\boldsymbol{z}_j)=1 | y_j=1\}}.$$

Libraries

Pour l 'analyse GLM, nous avons utilisé le paquetage *R* de *stats*, qui a basé son algorithme sur le GLM proposé par [55]. Pour l'analyse numérique de ce travail, nous appliquons l'une des bibliothèques d'apprentissage machine open source SVM les plus populaires, *LIBSVM*,

proposée par [41] pour étudier la performance de prédiction En tenant compte des différents paramètres du réseau.

6.4 Analyse de simulation

Dans cette section, nous présentons d'abord les paramètres généraux du système que nous avons configurés pour les simulateurs. Ensuite, nous analysons les performances de prédiction de l'apprentissage machine parmi les différents types de streaming HTTP avec toutes les fonctionnalités enregistrées. Enfin, nous démontrons la performance de prédiction en considérant seulement certaines caractéristiques.

Configuration de la simulation

Notre simulateur est lancé sur la base des conditions radio réalistes obtenues à partir des données de mesure d'un réseau 4G dans une grande ville européenne, avec un rayon de cellule moyen de 350 mètres. La bande de fréquence concernée est LTE 1800 MHZ. La figure 13 montre la distribution de probabilité mesurée du CQI obtenue à partir de mesures de stations de base collectées à l'aide d'un outil O&M. Chaque CQI est associé à un MCS, en déterminant son efficacité spectrale.

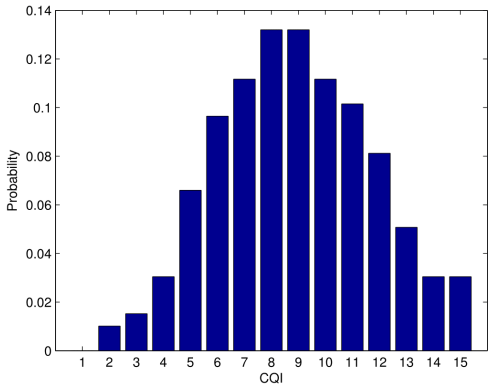


Figure 13: La distribution de la probabilité de CQI mesuré dans un réseau LTE.

En termes de streaming, nous configurons le débit binaire fixe avec $v_c = 1.5$Mbps et le streaming adaptatif avec deux options de débit binaire, $(v_1, v_2) = (2, 1)$Mbps. Les utilisateurs statiques ne bougerent pas et les utilisateurs mobiles auront un taux de changement sur leur mobilité, $v = 1$. En appliquant la distribution CQI précédente, nous simplifions 15CQIs en 5CQIs, $\mathscr{R} = \{49.24, 31.89, 18.84, 9.22, 2.975\}$ Mbps et sa proportion de trafic correspondante est calculée comme $\boldsymbol{p} = (11.1\%, 29.4\%, 37.5\%, 19.2\%, 2.8\%)$. Le reste de la configuration du système est $T = 40$s et $B = 1$s. Avec ces paramètres et sur la base de l'équation (41), nous pouvons obtenir le taux maximum d'arrivée du trafic comme $\lambda_{\max} = \frac{40}{4}\left(\frac{1.5+1}{16.23} + \frac{1.5+1}{21.51}\right)^{-1} = 0.37$.

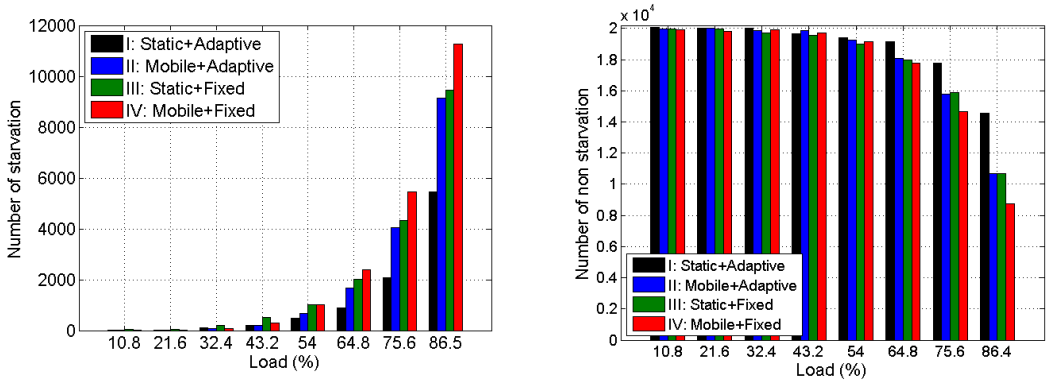
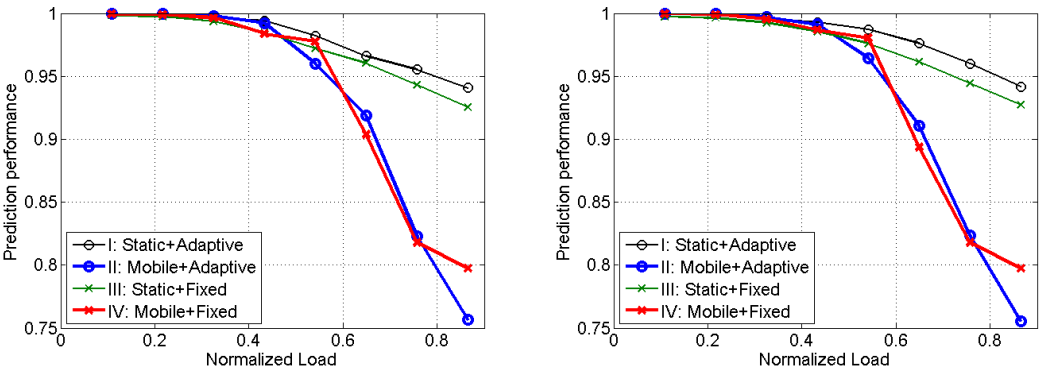


Figure 14: Formation d'échantillons de quatre types de streaming selon des charges.

Comme nous l'avons mentionné, la charge de trafic peut varier en heures, nous avons démontré 8 taux d'arrivée de flux normalisés par la valeur maximale $\lambda_{\max}$. Pour chaque λ, le simulateur génère $l = 10^6$ arrivées en streaming. 80% de données sont sélectionnés au hasard pour la formation et 20% de données pour validation parmi tous les échantillons. Dans la Fig. 6.3, le nombre moyen de privations de la vidéo et de la vacuité de la vidéo utilisées pour la formation est indiqué pour chaque charge. On peut observer que lorsque la charge est faible, la vacuité de la vidéo se produit rarement. Cependant, lorsque la charge approche à $\lambda_{\max}$, plus de flux vidéo éprouvent la pause de la vidéo. Il est également démontré que les utilisateurs statiques et les utilisateurs de flux adaptatif éprouvent moins de vacuité que les utilisateurs mobiles et fixes.

Performances de prédiction de différents flux HTTP



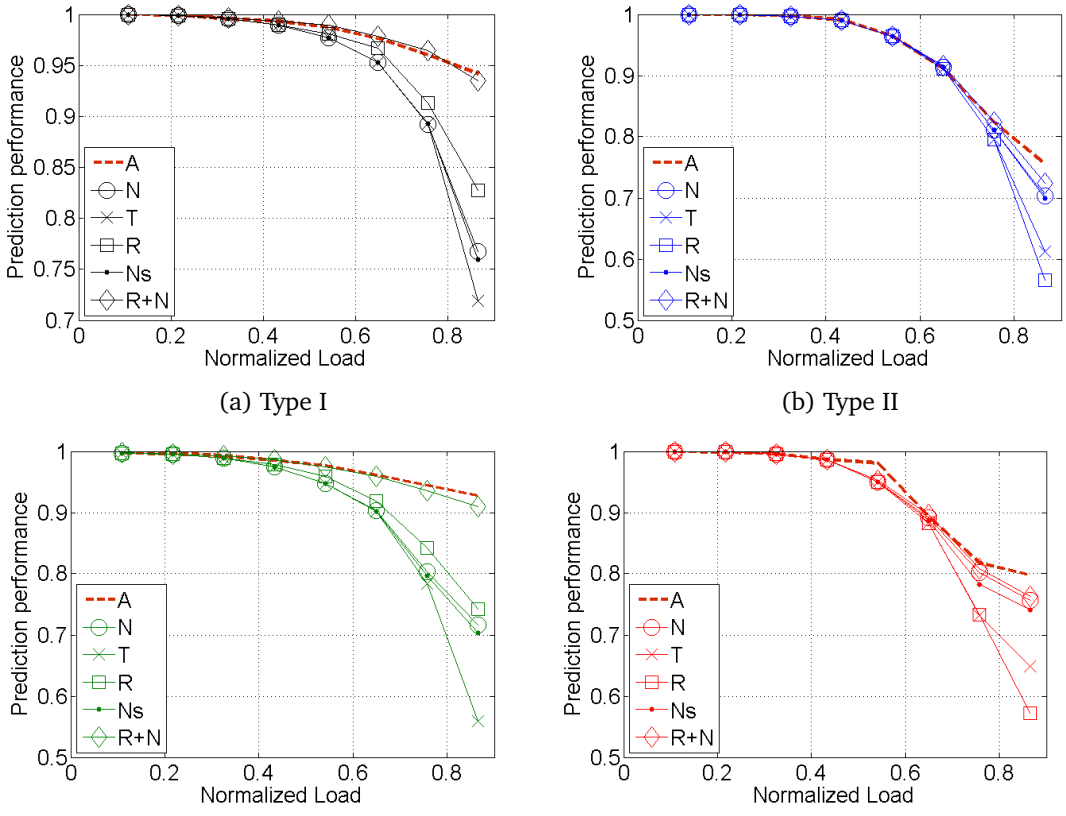
(a) avec GLM (b) avec SVM

Figure 15: Performance moyen de la prédiction

En appliquant les techniques d'apprentissage machine introduites, nous obtenons les deux figures suivantes montrant la performance moyenne de prédiction, P, avec GLM à la

Fig. 15a et avec SVM dans la Fig. 15b. On peut observer que les performances de prédiction de GLM et SVM sont similaires. De plus, quel que soit l'outil d'apprentissage que nous utilisons, nous pouvons observer que lorsque la charge augmente, les performances de prédiction diminueront en raison de l'augmentation de l'incertitude. À partir des résultats de la simulation, nous montrons que la QoE des utilisateurs mobiles est beaucoup plus difficile à prévoir, surtout lorsque la charge est importante. Toutefois, les utilisateurs statiques peuvent atteindre plus de 90% de précision. Il n'existe aucune règle générale disant que le débit fixe est plus facile à prévoir que le streaming adaptatif. Cela dépend de la mobilité. Pour les utilisateurs statiques disposant d'une propriété de diffusion en continu adaptative, la prédiction de QoE est plus précise. Pour les utilisateurs statiques, même près de 95% de précision peut être atteint à haute charge. Cependant, pour les utilisateurs mobiles, on peut dire que les informations initiales ne sont pas suffisantes pour la prédiction.

Performance de prédiction avec différentes caractéristiques



(a) Type I

(b) Type II

(c) Type III

(d) Type IV

Figure 16: Performance de la prédiction envisageant diffèrent caractéristiques du réseau.

Dans cette section, nous vérifions la performance de la prédiction compte tenu de l'accès

limité aux fonctionnalités de certains utilisateurs. Comme nous l'avons montré, GLM et SVM ont des résultats similaires à ceux de la Fig. 15a et 15b. Nous ne démontrons que la performance de prédiction de SVM de quatre types de streaming sur la Fig. 16. Il est montré que considérer toutes les caractéristiques (A) peut toujours effectuer la plus haute précision de prédiction et que N et N_s fournissent des informations très semblables. Il peut être une bonne nouvelle pour les opérateurs qu'ils n'ont pas besoin de connaître plus d'informations d'application du côté des utilisateurs. De plus, nous pouvons également observer que l'information R devient moins importante lorsque les utilisateurs sont mobiles et T est également moins importante. Cela peut être causé par notre hypothèse de mémoire tampon infinie. Dans la Fig. 16, nous démontrons également que considérer à la fois R et N peut fournir de bons résultats dans le cas statique.

7. Conclusion

Mesurer et améliorer la QoE de la vidéo devient de plus en plus important car la vidéo représente plus de 50% du trafic réseau. Dans cette thèse, nous proposons des modèles pour le dimensionnement de différents types de services de streaming, y compris le streaming en temps réel et le streaming adaptatif HTTP à l'intérieur des réseaux sans fils. Les deux sont développés en appliquant les concepts de dynamique de niveau de flux pour modéliser les arrivées et les départs de la demande de trafic, ce qui est très utilisé pour le trafic élastique et le trafic en temps réel dans la littérature.

Dans la section 3, nous développons un modèle de trafic au niveau flux et paquets pour les services de streaming en temps réel. Nous supposons l'existence de propriété quasi-stationnaire et combinons à la fois le débit et les niveaux de paquets pour calculer le taux de panne de paquets. En utilisant notre modèle, les opérateurs pourraient concevoir l'algorithme de contrôle d'admission correspondant pour les services de streaming en temps réel avec un taux de pannes garanti.

Dans la section 4 et 5, nous développons un modèle de trafic au niveau flux pour les services de streaming d'HTTP adaptatif. Le modèle prend en compte la dynamique du niveau flux et vérifie les impacts sur la performance de différents paramètres tels que la durée du morceau vidéo, le nombre de débits binaires vidéo et les schémas d'ordonnancement, etc. Nous abordons ci-dessous les questions potentielles rencontrées par les opérateurs lors de leur déploiement Le service de streaming adaptatif HTTP et souhaitent améliorer la qualité de l'expérience des utilisateurs comme: Impact de la durée du morceau vidéo, Impact du nombre de débits binaires vidéo, propose un nouveau design de la durée du morceau vidéo, Impact des schémas d'ordonnancement sur le streaming adaptatif et aussi impact de la mobilité des utilisateurs.

Dans la section finale, comme nous avons constaté qu'il est difficile de trouver une forme analytique exacte pour la QoE vidéo, y compris tous les paramètres possibles du système, nous proposons d'utiliser la technique d'apprentissage automatique pour prédire la qualité vidéo. Les résultats nous aident également à comprendre la corrélation entre la fluidité de la vidéo et les caractéristiques des utilisateurs enregistrées à chaque arrivée. Dans la dernière partie de notre thèse, nous examinons la performance de prédiction en utilisant GLM et SVM

avec différentes charges de système. Nous avons constaté que les précisions de prédiction sont plus de 92% pour les utilisateurs statiques à chaque charge et les paramètres de réseau les plus importants incluent le numéro de la condition radio et des flux, ce qui montre que la prédiction de la fluidité vidéo est faisable pour les utilisateurs statiques. Cependant, plus de fonctionnalités des utilisateurs sont nécessaires pour bien prédire les événements de la pause de vidéo.

7.1 Travaux futurs

Les travaux futurs les plus importants, de notre point de vue, sont liés à la section apprentissage machine. En effet, prédire la QoE vidéo sans avoir complètement la solution analytique est une première étape vers l'exploitation des données du réseau pour prédire et améliorer la qualité de vidéo. Par exemple, une première étape pour améliorer l'erreur de prédiction consiste à essayer d'autres modèles d'apprentissage plus aléatoires comme l'arbre de décision, la forêt aléatoire, le réseau neuronal et les voisins K-voisins. De plus, comme nos données d'entraînement sont générées sur la base des simulateurs d'arrivée de Poisson, il est également préférable de mettre en oeuvre un simulateur avec un profil de trafic plus réaliste ou d'utiliser des mesures réelles du réseau. L'ajout de fonctionnalités liées à la mobilité est également important pour la prévision de QoE. Une fois que la méthodologie de prédiction QoE a été améliorée, la deuxième étape sera de l'utiliser pour améliorer QoE, en proposant des algorithmes avancés d'auto-organisation. Comment mettre en oeuvre des algorithmes efficaces mais simples d'auto-organisation pour améliorer QoE de services de streaming pourrait être un sujet intéressant pour une autre thèse de doctorat.

List of Acronyms

3GPP	3rd Generation Partnership Project
AMC	Adaptive Modulation and Coding
AMPS	Analog Mobile Phone Service
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BS	Base Station
CDMA	Code Division Multiple Access
CQI	Channel Quality Indicator
CSI	Channel State Information
CTMC	Continous-Time Markov Chain
DASH	Dynamic Adaptive Streaming over HTTP
EDGE	Enhanced Data Rates for GSM Evolution
EPC	Evolved Packet Core
E-UTRA	Evolve- Univeral Terrestrial Radio Access
FCFS	First Come First Served
FDD	Frequency-Division Duplex
FDMA	Frequency Division Multiple Access
FTP	File Transport Protocol
GLM	Generalized Linear Model
GPRS	General Packet Radio Services
GSM	Global System for Mobile

HAS	HTTP Adaptive Streaming
HTTP	Hypertext Transfer Protocol
ITU	International Telecommunication Union
LCFS	Last Come First Served
LTE	Long Term Evolution
KPIs	Key Performance Indicators
MCS	Modulation Coding Scheme
MOS	Mean Opinion Score
MMPP	Markov Modulated Poisson Process
MPD	Media Presentation Description
MSE	Mean Square Error
O&M	Observation and Measurements
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
PDN	Packet Data Network
PLR	Packet Loss Rate
PS	Processor Sharing
PSNR	Peak Signal to Noise Ratio
RB	Physical Resource Block
RTP	Real Time Protocol
RTCP	Real Time Control Protocol
RTMP	Real Time Message Protocol
RTSP	Real Time Streaming Protocol
RR	Round Robin
QoE	Quality of Experience
QoS	Quality of Service
S-GW	Service Gatway

SINR Signal to Interference and Noise Ratio

SVM Support Vector Machine

TACS Total Access Communication System

TDMA Time Division Multiple Access

TDD Time-Division Duplex

TTI Transmission Time Interval

UE User Equipments

UMTS Univeral Mobile Telecommunications System

UTRA Universal Terrestrial Radio Access

VoD Video on Demand

VoLTE Voice over LTE

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter introduces the subject of this thesis. We first present the technological context of different video services functioning in telecommunication networks. Then we expose the main objectives and contributions of this thesis. We finally list all the publications made during the thesis at the end of this chapter.

1.1 Context

While 4G technology becomes much more mature, broadband services are easily accessible and affordable for every people. With broadband technology, various types of service which are difficult to support in 2G and 3G have large increasing rate in 4G networks. In the report [4], Cisco gave an interesting forecast reported in Table. 1.1 showing that video traffic has already accounted for more than 50% percent of global mobile data traffic in 2015 and that it has relatively larger increasing rate than the other types of traffic. The phenonemen that video traffic is the most important traffic then becomes unavoidable for the internet service providers.

Traffic Type	2015	2016	2017	2018	2019	2020
Web, Data, and VoIP	1,323	1,968	2,779	3,605	4,427	5,158
Video	2,031	3,643	6,232	9,977	15,410	22,963
Audio streaming	279	462	722	1,034	1,398	1,788
File sharing	51	106	196	317	472	653

Table 1.1: Global Mobile Data Traffic, from 2015 to 2020 (PB per Month) [4]

1.1.1 Video categories

Based on whether the video content are generated at the same time of watching or not, we can simply divide video services into two categories. They are respectively On-Demand

Video Streaming and Real-Time Video [85].

Video on Demand (VoD)

The on-demand video can be abbreviated to VoD, which means broadcasting programs on the basis of user requirements. Different from the traditional TV broadcasting, users can pause/play video anytime as they wish.

The best example of this type of service are provided by YouTube [40], Netflix, Hulu and Dailymotion which have an explosive growth and have become one of the most popular research topics since 2005. Simply speaking, the video content of VoD are stored at the cloud side and users can access anytime they want.

Real-time video

The content of real-time streaming service are generated at the same time of content delivery. Contrary to the VoD streaming service, users of real-time streaming can not replay video as they desired and have to follow the schedule of video content providers.

Real-time video can be easily categorized into two parts. One of them has another well-known name called, Live Streaming. The best examples of this type of service are all the Web TV service like BBC, Orange TV, BFM TV direct, etc. Moreover, real-time video also include the audio conference provided by Skype and other instant messengers like WhatsApp and FB Messengers.

In order to support different types of video services, several communication protocols are proposed and we are going to summarize them here. Some of them are open standards and others are only enclosed for the specified usages.

HTTP progressive download [14]

In progressive download, all the streaming technologies belong to the progressive download, which means that video are downloaded part by part as the word progressive describes. Therefore, users do not need to wait until the end of video download but can start to watch the video while downloading. Hypertext Transfer Protocol (HTTP) is the most popular protocol that supports progressive download and it is currently the most popular technology to deliver on-demand streaming. According to [76], the advantages of using HTTP Web server to deliver video include:

- Broad market adoption of HTTP and TCP/IP protocols; they support the majority of the Internet services today.
- HTTP-based delivery avoids NAT and firewall traversal issues.
- The ability to use standard/existing HTTP servers and caches instead of specialized streaming servers allow reuse of the existing infrastructure.

Moreover, progressive download enables users to start watching their video before the whole video are fully downloaded, as video are divided into small segments and are delivered separately to users. Nevertheless, one drawback of HTTP progressive download is that users are limited to choose only one video quality and video format.

Real Time Protocol (RTP)

RTP is designed for the transport of real-time data including audio and video. It can be used for media-on-demand as well as interactive services such as Internet telephony. RTP usually runs over UDP. Due to the packet loss, it is less reliable compared with HTTP progressive download. RTP is always implemented with Real Time Control Protocol (RTCP) as a control protocol.

For on-demand streaming, Real Time Streaming Protocol (RTSP) [3] is an application protocol like HTTP to deliver streaming and it supports the function of pause and return. For some of live streaming RTP is adopted. Most of video conference services like Skpye, are delivered by RTP/RTCP.

Real Time Message Protocol (RTMP)

RTMP, also known as Flash, is a protocol mainly transmitted based on TCP. More precisely speaking, it is a proprietary protocol for multimedia content transfer between a Flash player and a server. It is also an important video delivery technology but not in the scope of our thesis.

HTTP Adaptive Streaming (HAS)

HAS is an extended feature of HTTP progressive download and it aims to optimize and adapt the video configurations over time in order to deliver the best possible video quality considering changing link or network conditions. Following the property of HTTP progressive download, video are segmented and stored at the video servers. At the same time, these video segments are encoded in more than one version and hosted along with the Media Presentation Description (MPD) [89] as shown in Fig. 1.1. Based on this MPD metadata information in Fig. 1.2 that describes the relation of the segments and how they form a media presentation, clients use HTTP GET request to access the video segment one after another.

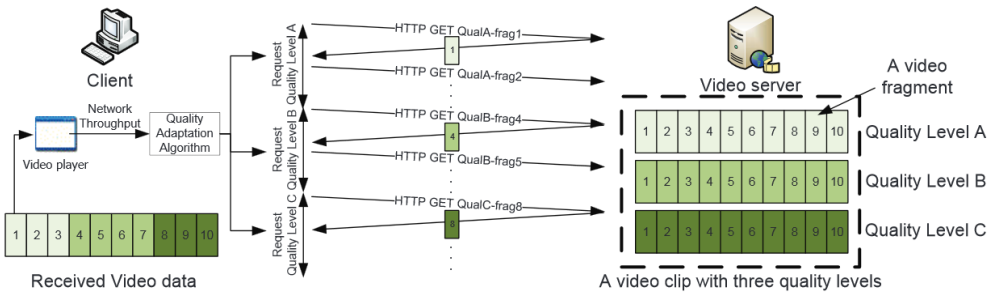


Figure 1.1: A typical HTTP adaptive streaming system [71].

Some industrial solutions are listed here. Some are public and other are private:

- MPEG Dynamic Adaptive Streaming over HTTP (DASH) [88]: MPEG-DASH is the only HTTP adaptive streaming solution that is an international standard. Work on DASH

started on 2010 and the standard was published in 2012 [6]. In addition, the concept of a Media Presentation is introduced in TS 26.234 [7].

- Apple HTTP Live Streaming (HLS): HLS is a communication protocol implemented by Apple as part of QuickTime and iOS. HLS supports both live and video on-demand content.
- Microsoft Smooth Streaming [2]: Smooth streaming is an IIS Media services extension. Microsoft is actively involved with 3GPP, MPEG and DECE for standardization.
- Adobe HTTP Dynamic Streaming

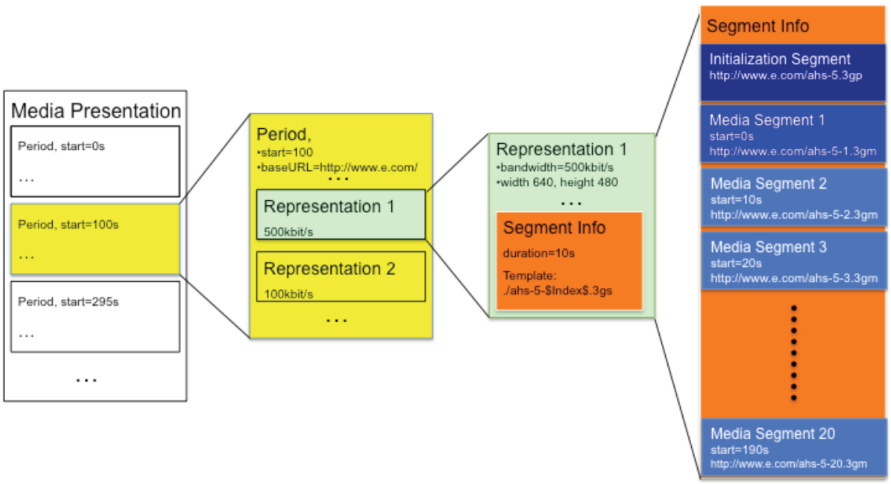


Figure 1.2: Media Presentation Data Model.

On the client side, the most important decisions are which segments to download, when to start with the download, and how to manage the receiver video buffer. The adaptation algorithm should select the appropriate representation in order to maximize the quality of experience [83]. The most common approach is to estimate the instantaneous channel bandwidth and to use it as decision criterion. For channel estimation, authors of [18] reviewed the available bitrate estimation algorithms. Other decision engines based on Markov Decision Process are described in [56]. In addition to the throughput, there are algorithms considering buffer level, e.g., the authors of [69] propose an adaptation engine based on the dynamics of the available throughput in the past and the actual buffer level to select the appropriate representation.

1.1.2 Quality of Experience (QoE)

Since video services became more popular, how to measure video performance has become a hot topic. Video delivery over wireless network has also been studied for more than 15 years. Ways to measure a video quality can be divided into subjective ones and objective one.s The subjective one is common for all types of video. Instead, corresponding to

different types of video services, different types of objective quality measure are defined and proposed.

Subjective quality measure

Mean Opinion Score (MOS)[5] is a measure, which can be either subjective or objective and is originally used for voice quality measurement. Later it is also applied for the usages of all types of video services. Subjective testing for visual assessment has been formalized in ITU-R Rec. BT.500 [51] and ITU-T Rec. P.910[52], which suggest standard viewing conditions, criteria for the selection of observers and test material, assessment procedures, and data analysis methods. MOS quantifies the service qualities into five different levels, from 1, meaning bad quality to 5, meaning excellent quality and the subjective measured method, such as Absolute Category Rating (ACR), Degradation Category Rating (DCR) and Pair Comparison (PC) are standardized in [53]. Several researches also study the video performance by MOS as [86].

Objective quality measure

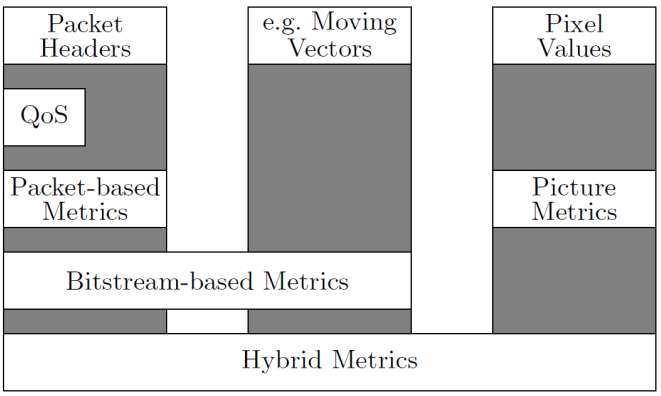


Figure 1.3: Different categories of objective video quality metrics, with QoS added for illustration purposes.[81][95]

Fig. 1.3 attempts to categorize the objective quality measure of video. In addition, it also tries to clarify the relationship between Quality of Service (QoS) and QoE. For QoS measures, the network QoS community has defined simple metrics to quantify transmission errors, such as Bit Error Rate (BER) and Packet Loss Rate (PLR). None of them take into account the content. Secondly, more approaching to the human visual system, Picture metrics treat the video data by pixel unit. The simplest possible metrics, Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR), take into account only signal to noise ratio, although it is also very easy to produce results that deviate form human perception. Packet-or bitstream-based metrics for compressed video delivery over packet networks look at the packet header information and the encoded bitstream directly without fully decoding the video. In paper

[95], authors focus on MPEG-2. To conclude, it is hard to say which metric is better. Hybrid metrics are proposed. Generally speaking, metrics more at the right side of Fig.1.3 can approach more to the user perception.

Quality measure for real-time video

In this section, we only describe the measures for real-time video transmitted by UDP. The one transmitted by TCP can be measured the performance metrics introduced in the following section. Based on the system mechanism that we have mentioned for the real-time video in the previous section 1.1.1 and due to the UDP property, having possibility to loss video packet, real-time video usually experiences video distortion. Therefore, some popular metrics for measuring the real-time video quality transmitted over unreliable protocol.

- PSNR, derived by setting the MSE [94] as

$$MSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\Big(f(i,j)-F(i,j)\Big)^2}{MN}, \tag{1.1}$$

$$PSNR = 20\log_{10}\Big(\frac{255}{\sqrt{MSE}}\Big), \tag{1.2}$$

  where $f(i,j)$ is the original signal at pixel (i,j), $F(i,j)$ is the reconstructed signal, and $M \times N$ is the picture size. The result is a single number in decibels, ranging from 30 to 40 for medium to high quality video. There are other picture metrics, such as VQM, SSIM, etc.

- Packet loss rate (PLR)
  Packet loss will directly influence the video quality. Therefore, several research works analyzed the impact of packet loss to the real-time video quality such as MPEG-2 [92]. In our thesis, when we study the real-time video, we mainly focus on PLR.

- Blocking rate
  In [35], blocking rate is used to study the performance of real-time streaming. Blocking rate is a high-level performance metric. Blocking happens when system capacity can not accept any more new video calls. This metrics is highly utilized in the studies of GSM system capacity, e.g. the Erlang models.

Quality measure for VoD

For the performance metrics of on-demand video, authors in [46] summarize them into five following terms and quantify the impacts of them on user engagement. In Fig. 1.4, we show a simple illustration of a life time of video session. The state of a video session can be mainly categorized into the following three states: Prefetch state, where user fills its buffer without playing, Playing state, where user start to play its video while download its video and Buffering state, where the video stalls until the buffer is filled up to the STARVATION level.

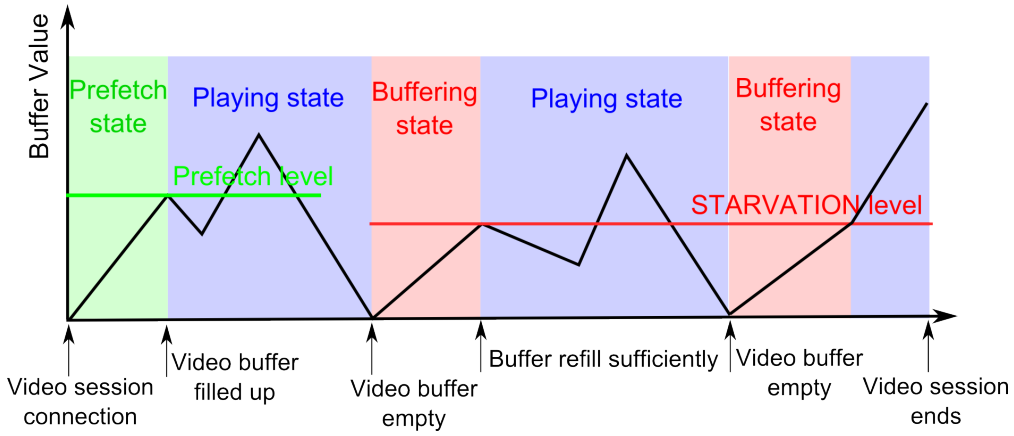


Figure 1.4: An illustration of a video session life time and associated video player events.

- **Join time**: Measured in seconds, this metric represents the duration from the beginning of a session connection until the time sufficient playout buffer has filled up. In Fig. 1.4, it is also called start-up delay.
- **Buffering ratio**: Represented as a percentage, this metric is the fraction of the total session time spent in buffering.
- **Rate of buffering events**: Buffer ratio does not capture the frequency of induced interruptions observed by the user. For example, a user may experience video stuttering where each interruption is small but the total number of interruptions is high.
- **Mean bitrate**: Adaptive streaming allows video player to switch between different bitrate streams. Mean bitrate is the sum of played bitrates weighted by the duration each bitrate is played.
- **Rendering quality**: Rendering rate (frames per second) is central to user's visual perception. Rendering rate may drop due to the CPU overload or due to network congestion.
- **Frequency of bitrate changing**: Once the video bit rate is adapted, users will experience a change of video quality. Therefore, it is better not to change the resolution very frequently.

There are also another performance metrics that is highly examined in some researches

- **Rate of bit rate switching**: When adaptive streaming is introduced, users are allowed to switch among several video bit rates. This metric measures the frequency of rate switching. Studies suggest users are likely to be sensitive to frequent and significant bitrate switches [44][45].

About the importance of these performance metrics, authors of paper [46] show that buffering ratio is the most important metric across all content genres and the bitrate is especially critical for Live (sports) content.

1.2 Objectives

The objectives of this thesis are to build up a traffic model to analyze the objective measure of video inside wireless networks for different types of video services. We begin by taking into account the real-time streaming service. Assuming that real-time streaming has the highest priority, we verify the relationship between

- traffic load and Packet Loss Rate (PLR).

As on-demand video accounts for larger part of network traffic, our thesis mainly focuses on investigating this type of service and especially, the HTTP adaptive streaming, because of the maturity of the technology. Property of adapting video bit rate is supposed to provide a freedom to balance between mean video bit rate and buffer performance. However, it is not clear the performance impacts of parameters both from the wireless networks and video delivery system. Therefore, in this part of research, we focus on the impacts of following network parameters:

- Video chunk duration
- Number of video bit rate
- Scheduling schemes
- Users' mobility

By applying our traffic model, operators understand how to well design the related network and video parameters for providing better adaptive streaming experience. In this thesis, we develop the corresponding traffic model using flow-level dynamics, demonstrate the performance impacts and propose the improving deployment methods.

1.3 Contributions

In this thesis, our main contribution is to propose an analytical model based on flow-level model for evaluation of video performance in different scenarios. Other detailed contributions are specified in the following:

In **chapter 2**, we introduce some background knowledges for this thesis. We begin by describing the basics of wireless cellular system and explain how a single-to-single transmission functions and how to model the capacity of this single-to-single link. Then we introduce the corresponding flow-level traffic model, a well-known method established on queueing theory in order to model wireless system. Finally, we present some popular machine learning techniques such as generalized linear model and support vector machine used for the following studies.

In **chapter 3**, our main contribution is to develop the packet delay distribution of real-time streaming services in a wireless cell. We model a Base Station (BS) by applying queueing theory and based on the quasi-stationary property, where we calculate the packet delay by

combining both packet-level and flow-level dynamics. Under the obtained packet delay distribution, we can then decide the acceptable flow arrival rate as an admission control policy given a packet delay constraint. We show that with some model extension, fast fading effect can be taken into account in the dimensioning problem. Works mentioned in this chapter are published in [C5].

In **chapter 4**, we begin by introducing the state of the art of HTTP adaptive streaming modeling. It is shown that using flow-level model, we consider the impacts of traffic dynamics on the performance of HTTP adaptive streaming. We start with considering significantly small video chunk duration. Then we extend our flow-level traffic model with the configuration of significantly large video chunk duration. Respective models stand for an extreme performance bound for any intermediate chunk duration configurations. These performance impacts have been observed by calculating the Key Performance Indicators (KPIs) as the following, mean video bit rate standing for video resolution and mean deficit rate, mean serving time and mean buffer surplus standing for video smoothness. The adaptive streaming traffic model is also extended to integrate the effects of heterogeneous radio conditions, scheduling schemes and coexistence with elastic traffic as a general one. To make the works complete, we also take into account the intra-cell mobility into our flow-level model. This chapter presents all the contributions which have already been published respectively in [C2-4] and [J1].

In **chapter 5**, our contributions can be divided into two parts. One is to validate our proposed traffic model for the adaptive streaming by simulation. The other is to examine the performance impacts of different system configurations. In order to reduce the complexity of simulation we present an approximation model to simplify the numerical analysis with multiple classes of users flow. Our proposed model can assist service providers to understand the impacts of chunk duration, the impacts of number of video bit rate, the impacts of scheduling schemes and the impacts of mobility. Our results show that smaller chunk duration can offer a better video smoothness with a price to lose little video resolution, vice versa for larger chunk duration configuration. Moreover, our results also show that providing infinite video bit rate may not be a good idea. One of our main contributions is mentioned in this chapter where we propose to deploy the video chunk with same size instead of same duration and we show that this can improve video smoothness. For the publication reference, the results presented in this chapter can be found in [C2-4] and [J1].

In **chapter 6**, we study the video quality of experience by another approach, where we apply machine learning technique to predict one of the important Quality of Experience (QoE) metrics, video starvation. We demonstrate the prediction performance of different HTTP streaming and show that static and adaptive streaming possess the highest prediction accuracy. We also demonstrate that different network parameters have different importances to predict video starvation. By using machine learning technique, we can still understand the relationship between performance metrics and system statistics when no exact mathematical model is available. This gives an access for operators to understand deeply users' QoE. Contributions of this work are submitted in [C1].

1.4 Publications

Conference papers

[C1] (Under Review) **Yu-Ting Lin**, Salaheddine Elayoubi, Eduardo Mucelli and Sana Ben Jemaa, "Predicting the QoE of Video Streaming with Machine Learning in LTE Mobile Networks," submit to *2017 IEEE ICC*, Dec. 2017.

[C2] Nivine Abbas, **Yu-Ting Lin** and Berna Sayrac, "Mobility-driven Scheduler for Mobile Networks Carrying Adaptive Streaming Traffic," in *2016 IEEE PIMRC*, Sep. 2016.

[C3] **Yu-Ting Lin**, Thomas Bonald and Salaheddine Elayoubi, "Impact of Chunk Duration on Adaptive Streaming Performance in Mobile Networks," in *2016 IEEE WCNC*, Apr. 2016.

[C4] Thomas Bonald, Salaheddine Elayoubi and **Yu-Ting Lin**, "A Flow-Level Performance Model for Mobile Networks Carrying Adaptive Streaming Traffic," in *2015 IEEE Globecom*, Dec. 2015.

[C5] **Yu-Ting Lin** and Salaheddine Elayoubi and Ridha Nasri, "Capacity Dimensioning for Real-Time Video Services in Wireless Mobile Networks," in *2015 IEEE VTC Workshop*, Glasgow, Scotland, May 2015.

Journal paper

[J1] (Submitted) **Yu-Ting Lin**, Salaheddine Elayoubi and Thomas Bonald, "Flow-Level Performance Model for Adaptive Streaming Traffic in Mobile Networks," .

Chapter 2

Background

In this chapter, we start by presenting some background knowledges of our thesis, which can be divided into three parties. The background wireless systems will be presented in 2.1. Queueing theory and wireless system modeling are presented in 2.2 and some machine learning techniques are introduced in 2.3.

2.1 Wireless systems

As our thesis is about the video performance in wireless networks, it is important to know the concepts and the characteristics of wireless networks.

2.1.1 Wireless channel characteristics

Characteristic of the mobile wireless channel is the variations of the channel strength over time and over frequency. The variations are generally composed of slow fading and fast fading, as shown in Fig. 2.1 [91]. In this thesis, we mainly focus on the slow fading effect for both real-time streaming and HTTP adaptive streaming. In the case of real-time streaming service, we also try to include the fast fading effect.

Slow fading

Slow-fading is composed of two principle effects, path loss (attenuation) and shadowing, caused by large objects covering such as buildings and hills. The path loss of signal is a function of distance. This occurs as the mobile moves of the order of the cell size, and is typically frequency independent. There are several path loss prediction models that consist in characterizing the propagation medium theoretically and by means of measurements. The empirical models based on statistical analysis over a large number of experimental measures prove to be analytically simple, tractable, and easily extrapolated to other environments with similar propagation conditions as those where measurements were made. The most known examples are HATA and COST-Hata models [12][49]. For urban areas, here we present

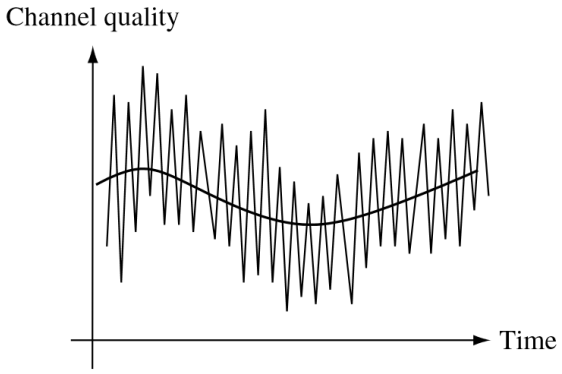


Figure 2.1: Channel quality varies over multiple time-scales. At a slow scale, channel varies due to shadowing, etc. At a fast scale, channel varies due to multi-path effects [91].

the HATA model for calculating the path-loss, $l_{\mathrm{dB}}(d)$ with d denoting for distance between transceiver,

$$l_{\mathrm{dB}}(d) = l_0 + 10\beta \log_{10}(d), \tag{2.1}$$

where β is the path loss exponent and l_0 is a fixed term which depends on the system parameters such as the frequency band and the base station height.

Moreover, the slow-fading channel is also impacted by shadowing, along the propagation path. This random phenomenon has been described so far by a log-normal distribution [49] and is commonly used for network simulation and performance evaluation. The log-normal distribution with parameters (μ_S, σ_S) has the following probabilistic density function:

$$f_S(x) = \frac{1}{x\sqrt{2\pi}\sigma_S} \exp\Big(-\frac{(\log x - \mu_S)^2}{2\sigma_S^2}\Big),\ x \geq 0, \tag{2.2}$$

where μ_S and σ_S denote the mean and the standard deviation in dB, respectively. A example shows that μ_S can be selected as 0 and σ_S is ranging from $3dB$ to $14dB$.

Fast fading

Fast fading is due to the constructive and destructive interference of the multiple signal paths between the transmitter and receiver. This occurs at the scale of the carrier wavelength, and is frequency dependent. Several models are proposed, the Rayleigh propagation model [87] is applicable to environments where there are many different signal paths, none of which can dominate. As a result, the magnitude of the signal has a Rayleigh distribution as:

$$f_R(x) = \frac{x}{\sigma^2} \exp\Big(\frac{-x^2}{2\sigma^2}\Big),\ x \geq 0, \tag{2.3}$$

where $2\sigma^2$ is the average power of the received signal. If there is a dominant line of sight component, Rician fading may be more appropriate [84].

2.1.2 Channel capacity

The channel capacity of a wireless link between a transmitter-receiver pair is constrained by impairments due to the environment e.g. the channel attenuation seen previously and by other simultaneous transmissions on the same or adjacent frequency band which generate interference. We always use Additive White Gaussian Noise (AWGN) to model a wireless link affected by the thermal noise, which is due to the thermal agitation of electrons in electronic devices. With P_u, standing for the received signal and I_u standing for the overall interference perceived from a specific user u, the signal quality is determined by the Signal to Interference and Noise Ratio (SINR) ratio given by:

$$SINR_u = \frac{f_u P_u}{I_u + N_0}, \tag{2.4}$$

where f_u stands for the channel fading effect we have mentioned before, usually it is described by the Channel State Information (CSI). Once we got the fading channel of a signal path, we can calculate the theoretical point-to-point channel capacity using Shannon's formula. And we can express the capacity of a channel as

$$R = W \log(1 + SINR_u), \tag{2.5}$$

where W denotes the system bandwidth. Based on the R value, transmitter will adapt to a proper Modulation Coding Scheme (MCS) for transmission. It is worthy of mentioning that Shannon formula offers us an upper bound for the channel capacity, which is an optimistic result.

2.1.3 Network deployment

A simple diagram of a cellular system is shown in Fig. 2.2. A cellular network is a wireless network that provides services by using a large number of Base Station (BS) with limited power, each of which covers a limited area called a *cell*. These cells provide together the coverage of a wide geographic area. This enables a large number of User Equipments (UE) to communicate with each other and with the fixed infrastructure. When users access video services, the data flow will pass from the servers of video service providers located at Internet through tService Gatway (S-GW) and Packet Data Network (PDN) Gateway to users.

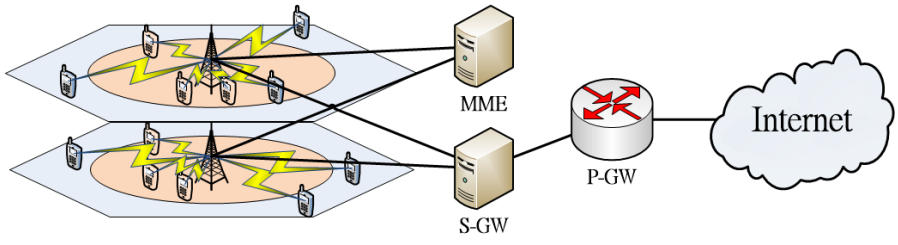


Figure 2.2: Cellular system diagram

When a cellular network is modeled, a cell is usually modeled as an omi-directional cell or a tri-sector cell as shown in Fig. 2.3. In reality, other sectorizations are also possible, i.e. 2, 4, 5 sectors in a cell. The benefit of multiple sectors is to have frequency reuse so as to increase the system capacity. However, the interference between cells will also increase. In the case of tri-sector model, we can regard it as three independent cells with its own BS.

(a) Omi cell (b) Tri-sector cell

Figure 2.3: A typical cell

2.1.4 Cellular system structure and evolution

The mobile cellular network used today has gone through several generations [59][47]. The first generation (1G) appeared in the 1980s which used analog technology to transmit the information. The well-known standard for the 1G system are for example, Analog Mobile Phone Service (AMPS) in American and Total Access Communication System (TACS) in Europe.

In the early 1990's, the Second Generation (2G) starts to emerge using the digital communication and coding technology to guarantee the correctness of data transmission. In terms of services, voice and text message are provided in 2G systems. Although there were several 2G standards, the Global System for Mobile (GSM) is the most successful system and was highly adapted by the operators around the world. This has enabled GSM to be further enhanced and developed so as to support higher data rates. General Packet Radio Services (GPRS) and Enhanced Data Rates for GSM Evolution (EDGE) are the evolutions of GSM with enhanced Adaptive Modulation and Coding (AMC) and some other coding schemes.

The evolution of cellular systems continue to third generation which introduced the packet-switched concept coexisting with the circuit-switched method used in the previous system. The Univeral Mobile Telecommunications System (UMTS) developed within the 3rd Generation Partnership Project (3GPP) is one of the candidates that meet the 3G requirement of International Telecommunication Union (ITU) in terms of performance, service and spectrum efficiency. It provides enhanced radio interface called Universal Terrestrial Radio Access (UTRA) network and a core network evolved from the last generation.

The Long Term Evolution (LTE) of UMTS system is the 3.9th Generation of cellular networks. It was designed with an Evolve- Univeral Terrestrial Radio Access (E-UTRA) network with a full-IP core network called Evolved Packet Core (EPC). Whole information are supposed to be transmitted in packet-switched network and no circuit-switched service will be

offered anymore in 4G. The entire architecture is named Evolved Packet System. Then we have the real 4G system, LTE-Advanced that fulfills the requirement of ITU with the advanced features for both radio and core networks.

Nowadays, discussions are launched for 5G. The ITU requirement of 5G has not been defined yet. However, the most recognized system characteristics of 5G are Massive system capacity, Very high data rate everywhere, Very low latency, Ultra-high reliability and availability, Very low device cost and energy consumption, Energy-efficient networks. 5G also includes a lot of specific topics and technology like virtualization and it can support various types of services such as machine-type communication, automatic car, intelligent factory, etc.

2.1.5 Radio resource management

As radio resources are limited, how to fairly share the wireless resources and how to increase the spectrum efficiency become important issues. In the followings, we introduce two properties concerning the wireless resource management.

Duplexing

The duplexing aims at defining a transmission technique between the downlink and uplink link. There are two common techniques, Time-Division Duplex (TDD) and Frequency-Division Duplex (FDD) as shown in Fig. 2.4. In TDD, the downlink and uplink transmissions are partitioned over time and on the same frequency band. Generally speaking, TDD provides more freedom to the resource allocation. On the other hand, in FDD, the downlink and uplink are allocated to the separated frequency bands during the whole time axis.

(a) Frequency-Division Duplex (FDD)

(b) Time-Division Duplex (TDD)

Figure 2.4: Two ways of duplexing [47]

In this thesis, we focus on the downlink performance, because the video traffic is usually larger on the downlink. Moreover, only FDD is taken into account.

Multiple access

Multiple access is the technique which shares the wireless resource among users. The two basic techniques are Time Division Multiple Access (TDMA) and Frequency Division Multiple Access (FDMA). The 2nd Generation family GSM, GPRS, EDGE was based on TDMA and FDMA. However, starting from the 3rd Generation, UMTS utilized a more advanced multiple access technique, called Code Division Multiple Access (CDMA). This technique

enables multiple mobile stations to communicate simultaneously on the same frequency and at the same time and more spectrum efficiency are exploited. CDMA is implemented based on the spread spectrum technology, where the original data stream is spread with a code over a longer sequence of transmitted bits. The orthogonality among the transmission channels is ensured by the orthogonality between the spreading codes.

In 4G/LTE specifications, Orthogonal Frequency Division Multiple Access (OFDMA) shares the spectrum resource based on OFDM technology, which increases the spectrum efficiency. It consists of splitting each user data stream into several sub-streams, which are sent in parallel on several sub-carriers. These subcarriers provide higher spectrum efficiency because they are orthogonal to each other and less interference are created. In the specification of LTE systems, a Physical Resource Block (RB) is shown in Fig. 2.5, which is composed of 7×12 resource elements in a slot (0.5ms). For a LTE with 20MHz spectrum, 100 RBs are available in one slot.

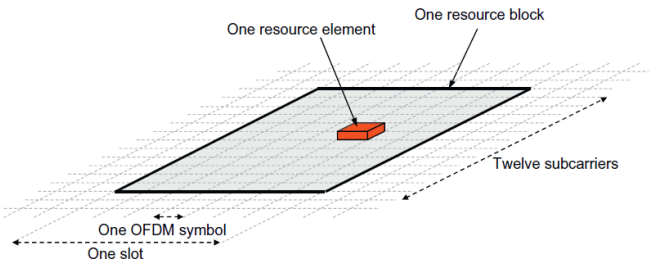


Figure 2.5: The definition of a resource block in OFDM system.

Scheduling

A resource block is the minimum unit for scheduling. As we have shown in the previous section, an LTE system with 20MHz has 100 RBs able to be allocated. The task of a centralized scheduler is to distribute these wireless resource to a group of users. In our thesis, we assume that we can allocate any fraction of time-frequency block, φ_u to a certain user u. To fully utilize the wireless resource, we have

$$\sum_{u \in \mathscr{U}} \varphi_u = 1, \tag{2.6}$$

where $\mathscr{U}$ denotes as the user set. We assume that the fraction, φ_u can be any values in $[0, 1]$. Scheduling algorithm is supposed to improve spectral efficiency of the system. In the thesis, we consider several scheduling schemes starting from Round Robin (RR) scheme, a simple, fair scheduling algorithm that does not exploit fast fading. Other scheduling schemes are introduced in details in section 4.5.

2.2 Queueing theory and traffic modeling

In this section, we introduce the basic knowledge of queueing theory and how to apply queueing theory for traffic modeling. Queueing theory is a popular mathematical tool used to describe a dynamic system with a shared resource. In this section, we only introduce some concepts applied in our thesis. Regarding the details, we refer the readers to [32][22] .

2.2.1 Queue model

We begin by introducing a single station queue as shown in Fig. 2.6. For a queueing network contains more than one station, readers can refer to, [22]. A queue is characterized

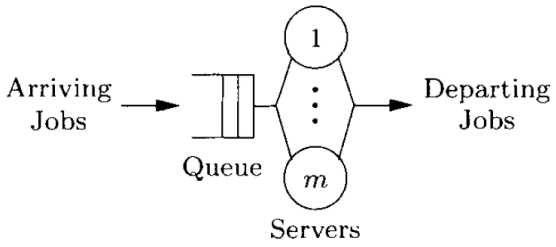


Figure 2.6: Single station queue [22]

by several parameters such as the number of servers, the queue capacity in terms of the number of customers, and the statistical characteristics of customer arrivals and service time. **Kendall's Notation** is invented to describe the characteristics of a queue as

$$A/S/m/[n]/[D] \tag{2.7}$$

where A indicates the distribution of the inter-arrival times, B denotes the distribution of the service times, and m is the number of serves ($m \geq 1$). Moreover, n represents the maximum number of clients allowed in the system and D stands for the service discipline. n and D are not always specified. The following symbols are normally used for A and B, where

- M : Exponential distribution (memoryless property)
- D : Deterministic distribution, the inter-arrival time or service time is constant
- E : Erlang distribution, where it corresponds to a sum of exponentials
- H : Hyper-exponential distribution, where it is a random choice among exponentials
- G : General distribution

There are also some non independent arrival processes including Markov Modulated Poisson Process (MMPP), which is examined in chapter 3 for A. The default discipline for D is typically:

- First Come First Served (FCFS): The jobs are served in the order of their arrivals.

- Last Come First Served (LCFS): The job that arrived last is going to be served first.
- Processor Sharing (PS): This strategy corresponds to round robin with infinitestimally small time slices. It is as if all jobs are served simultaneously and the service time is increased correspondingly.

Queue size, n and service discipline, D are not always specified. By default, n is configured as infinite queue size and FCFS is the default discipline for D.

2.2.2 M/M/1 queue

The simplest example is M/M/1 queue. Recall that in this case, the arrival process is Poisson, the service times are exponentially distributed, and there are a single server and infinite queue size. The system then can be modeled as a birth-death process with arrival rate λ and a constant service rate μ. With the assumption that $\lambda < \mu$, the underlying Continous-Time Markov Chain (CTMC) is ergodic and hence the queueing system is stable. Instead, if the system is not stable, that is $\lambda > \mu$, the number of users inside the system will grow to infinity.

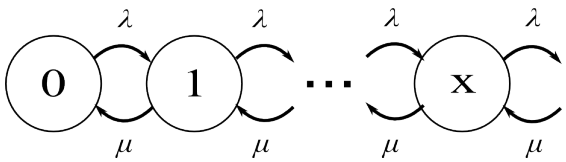


Figure 2.7: Markov chain model for M/M/1 queue.

With the load defined as $\rho = \frac{\lambda}{\mu}$ and assuming $\rho < 1$, the stationary distribution, $\pi(x)$, having x users inside the system is given by:

$$\pi(x) = \pi(0)\rho^x, \; x \in \mathcal{N}, \tag{2.8}$$

with

$$\pi(0) = \Big(\sum_x \rho^x\Big)^{-1} = 1-\rho. \tag{2.9}$$

We can calculate the mean number of users as

$$E(X) = \sum_x x\pi(x) = \frac{\rho}{1-\rho}, \tag{2.10}$$

where we can observe that when $\rho \to 1$, $E(X) \to \infty$. By Little formula [32], we obtain the mean sojourn time of a user, S, as

$$S = \frac{E(X)}{\lambda} = \frac{1}{\mu-\lambda}. \tag{2.11}$$

When$\mu \approx \lambda$, meaning that $\rho \to 1$, the sojourn time of a user goes to infinity.

2.2.3 State dependent queue

In order to better model a general system, it is common to generalize the simple M/M/1 queue. The arrival rate and departure rate of each state depends on the state x, $\phi'(x)$ and $\phi(x)$.

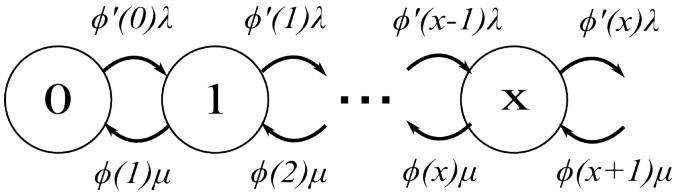


Figure 2.8: Markov chain model for state dependent queue.

Therefore, by the reversibility, the stationary distribution $\pi(x)$ is denoted as

$$\pi(x) = \pi(0)\Big(\frac{\lambda}{\mu}\Big)^x \frac{\phi'(0)\phi'(1)\cdots\phi'(x-1)}{\phi(1)\phi(2)\cdots\phi(x)}, \text{ where } x \in \mathcal{N}. \tag{2.12}$$

In the later modeling of HTTP adaptive streaming, we use the concept of state dependent queue for modeling the parameter, chunk duration and in chapter 5, we use the same concept to develop our approximation model.

2.2.4 Processor sharing discipline

Different from the classical FCFS queues, processor sharing queues provide an interesting property called*insensitivity*, mentioned in [62] and [24]. Assuming Poisson arrivals, the stationary distribution of the number of customers does not depend on the distribution of service times, which is not the case with the FCFS discipline.

In reality, the service time distribution is not always exponential. Therefore, in practice, this property is very useful and has the practical interest in communication networks of allowing the development of engineering rules independently of precise traffic statistics.

2.2.5 Whittle networks

We consider a network of n single-server queues coupled by their service rate. Customers arrive to the system according to a Poisson process of intensity λ_i at queue i, then leave the network after finishing the service. In Whittle networks, the total service rate of each queue depends on the network state. Customers require an exponential service rate with parameter μ_i at queue i is thus equal to $\mu_i\phi_i(\boldsymbol{x})$ in state $\boldsymbol{x}$, where $\boldsymbol{x} = (x_i, \cdots, x_N)$.

The network is said to be a Whittle network if the service capacities satisfy the following balance property given by

$$\phi_i(\boldsymbol{x})\phi_j(\boldsymbol{x}-\boldsymbol{e}_i) = \phi_j(\boldsymbol{x})\phi_i(\boldsymbol{x}-\boldsymbol{e}_j),\ \forall i,j,\forall \boldsymbol{x} : x_i > 0, x_j > 0. \tag{2.13}$$

Seeing in Fig. 2.9, let $(\boldsymbol{x}, \boldsymbol{x}-\boldsymbol{e}_{i_1}, \cdots, \boldsymbol{x}-\boldsymbol{e}_{i_1}\cdots-\boldsymbol{e}_{i_n}, 0)$ be a direct path from state $\boldsymbol{x}$ to state 0. A path of length n where n is the number of customers in state $\boldsymbol{x}$. The balance property implies that the expression

$$\Phi(\boldsymbol{x}) = \frac{1}{\phi_{i_1}(\boldsymbol{x})\phi_{i_2}(\boldsymbol{x}-\boldsymbol{e}_{i_1})\cdots\phi_{i_n}(\boldsymbol{x}-\boldsymbol{e}_{i_1}\cdots-\boldsymbol{e}_{i_n})}, \tag{2.14}$$

is independent of the considered direct path. Therefore, ϕ_i can be uniquely characterized by the function Φ, referred to as the balance function:

$$\phi_i(\boldsymbol{x}) = \frac{\Phi(\boldsymbol{x}-\boldsymbol{e}_i)}{\Phi(\boldsymbol{x})}, \quad i = 1, \cdots, N,\ x_i > 0. \tag{2.15}$$

The Whittle network is stable if and only if:

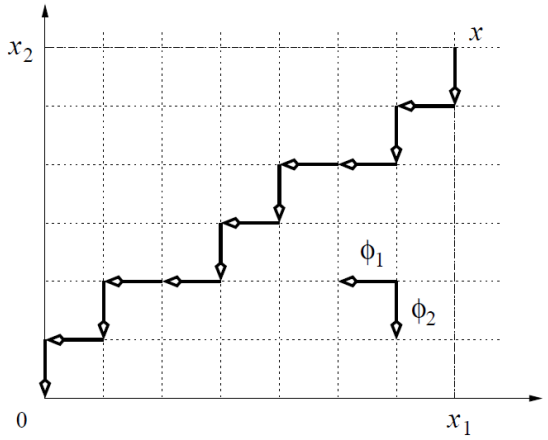


Figure 2.9: The balance function $\Phi(\boldsymbol{x})$ is equal to each weight of any path from state $\boldsymbol{x}$ to state 0 [35].

$$\sum_{\boldsymbol{x}} \Phi(\boldsymbol{x}) \prod_{i=1}^{N} \left(\frac{\lambda_i}{\mu_i}\right)^{x_i} < \infty, \tag{2.16}$$

in which case the stationary distribution is:

$$\pi(\boldsymbol{x}) = \pi(0)\Phi(\boldsymbol{x}) \prod_{i=1}^{N} \left(\frac{\lambda_i}{\mu_i}\right)^{x_i}, \tag{2.17}$$

where the proof can be found in [82] if the processor-sharing property holds. This stationary distribution is insensitive to the service requirements at any node.

2.2.6 Packet-level modeling v.s. Flow-level modeling

Two levels of traffic dynamics can be considered, either at *packet-level* or at *flow-level*. Several studies report that examining the dynamics of IP traffic is difficult (e.g. and references therein) since the statistics of packets arriving at and departing from the network

exhibit self-similar behavior due to the heavy tailed distribution of document size and all the mechanisms of TCP congestion control [48]. As a result, evaluating network performance at packet-level is hardly tractable. In particular, modeling the network using queuing theory with the assumption of Poisson arrival for instance, is not applicable. In [80], the flow-level model have been introduced. The term flow refers to a continuous stream of packets using the same path in a network and characterized by the starting time and the size. In the next section, flow-level modeling and its application for wireless network are introduced.

2.2.7 Flow-level modeling

Paper [68][80] introduced the concept of flow-level dynamics to model the variations of the resource shares. Each flow represents a service request generated by a client and the departures/arrivals of a flow influence the resource shares. Several researches applied this technique for service operators to investigate the performance of wireless network. An important set of works applying flow level modeling in different networks are summarized in [32] and [25] focuses on the insensitive property of flow-level modeling in communication networks. In paper [34], a flow level model has been proposed for studying elastic traffic in cellular networks. Works of [27] extends to the model considering multiple cells. Note that a flow may be subject to different radio conditions due to the position of the user in the cell and to throughput variations due to the dynamics of arrivals and departures of other users. This model has been extended for taking into consideration mobility in [26] and advanced radio features like intra-cell coordination in [60] and inter-cell coordination in [28]. Flow-level model has also been applied in modeling the traffic in fixed network as [33] and in modeling WiFi network in [31].

We present the basic flow-level modeling of elastic traffic in the case of mobile networks here:

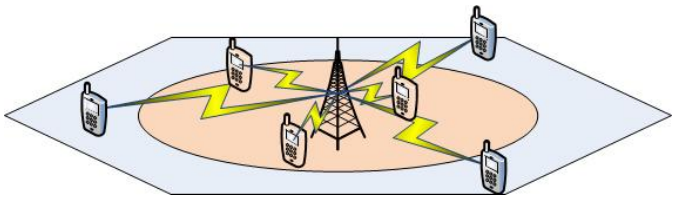


Figure 2.10: A typical flow-level modeling for a typical cell.

We consider an arbitrary set of classes of UEs indexed by $i \in \mathscr{C}$ to reflect the different radio conditions, R_i (i.e., locations) in the considered cell. In practice, the transmission rate depends on the radio environment and varies over time due to user mobility. Unless otherwise specified, we ignore the fast fading effects. Hence, the peak rate R_i depends on the user's position in the cell. We assume that the transmission rate is constant during the data transfer unless user has a large position changes. In each class, we assume that data flows arrive according to a Poisson process with intensity λ in the reference cell. Each flow stays in the system as long as the corresponding data have not been successfully transmitted to UE. Flow sizes are assumed to be independent and exponentially distributed with mean

σ bits, although all our results are approximately insensitive to the distribution. The traffic intensity is $\lambda \times \sigma$ in bit/s. The total arrival rate λ is composed of the arrival rate at each class-i, where $\lambda_i = \lambda p_i$ and

$$\sum_{i \in \mathscr{C}} p_i = 1. \tag{2.18}$$

Let $X_i(t)$ be the number of class-i flows at time t. The vector $X(t) = (X_i)_{i \in \mathscr{C}}$ is an irreducible Markov process whose transition rates depend on the scheduling scheme, which we will discuss this impact in chapter 4. Here, we assume that Round Robin (RR) scheduling scheme. The performance metrics considered is *flow throughput*(in bit/s). Let us say τ_i is the mean duration of class-i flow. According to the Little's formula, $E(X_i) = \lambda_i \tau_i$ and we have

$$\gamma_i = \frac{\sigma}{\tau_i} = \frac{\lambda_i \sigma}{E(X_i)}. \tag{2.19}$$

This is ratio of the traffic intensity of class i to the mean number of class-i flows. This throughput metric reflects user experience, accounting both for the radio conditions and for the random nature of traffic, through the stationary distribution of the Markov process $X(t)$. The mean flow throughput in the cell is given by:

$$\gamma = \frac{\sigma}{\tau}, \tag{2.20}$$

where τ is the mean flow duration of the cell, $\tau = \sum_{i \in \mathscr{C}} p_i \tau_i$. We obtain

$$\gamma = \Big(\sum_{i \in \mathscr{C}} \frac{p_i}{\gamma_i} \Big)^{-1}. \tag{2.21}$$

This is the weighted harmonic mean of the per-class flow throughputs, with weights given by the per-class traffic intensities. The idea of the harmonic average of throughputs was proposed in [34]. Applying the RR scheduling scheme, the balance property, Eq. (2.13) is verified and the queueing system can be viewed as a Whittle network [32]. With the load definition of

$$\rho_i = \frac{\lambda_i \sigma}{R_i}, \ \rho = \sum_{i \in \mathscr{C}} \rho_i = \frac{\lambda \sigma}{\hat{R}}, \tag{2.22}$$

where $\hat{R} = \Big(\sum_{i \in \mathscr{C}} \frac{p_i}{R_i} \Big)^{-1}$. Therefore, the stationary distribution of number of flow in the cell, $\boldsymbol{x}$ is given by:

$$\pi(\boldsymbol{x}) = (1-\rho) \frac{|\boldsymbol{x}|!}{\prod_{i \in \mathscr{C}} x_i!} \prod_{i \in \mathscr{C}} \rho_i^{x_i}, \tag{2.23}$$

where $|\boldsymbol{x}| = \sum_i x_i$.

2.3 Machine learning

Machine learning is a popular tool usually used for making predictions, decisions or classification based on a large amount of data. It is widely applied to pattern recognition and artificial intelligence for instance. It is closely related to the computational statistics. As some QoE metrics in our traffic model might be too complicated to express in an exact mathematical form, we attempt to utilize machine learning to find out the correlation between features and output results, here more specifically the QoE of users. In this section, we present some backgrounds of machine learning useful for chapter 6. Generally speaking, there are two types of machine learning. One is supervised learning. The other is non-supervised learning. In this thesis, we focus on supervised learning.

2.3.1 Supervised learning

A general supervised learning problem is formulated as Fig. 2.11. Assuming there are m training data pairs. For ith training data, we use vector $\boldsymbol{x}_i \in \mathscr{R}^n$ to represent the input variables, also called input features. Here n represents the number of features in $\boldsymbol{x}_i$. y_i is denoted as the output or target variable that we are trying to predict. A pair $(\boldsymbol{x}_i, y_i)$ is called a training example in the dataset that we use to learn is called a training set, $\{(\boldsymbol{x}_i, y_i) : i = 1, \cdots, m\}$. $\mathscr{X} = \{\boldsymbol{x}_i\}_{i=1,\cdots,m}$ denotes the space of input values, and $\mathscr{Y} = \{y_i\}_{i=1,\cdots,m}$ the space of output values. To describe the supervised learning problem slightly more formally,

> **the goal is to learn a function $h : \mathscr{X} \mapsto \mathscr{Y}$ so that $h(\boldsymbol{x}_i)$ is a good predictor for the corresponding value of y_i.**

This function, h is called a hypothesis. When the target variable that we are trying to predict is continuous, e.g. $\boldsymbol{x}_i \in \mathscr{R}^n$ and $\boldsymbol{y} \in \mathscr{R}^m$, we call the learning problem a regression problem. When y can take on a small number of discrete values, e.g. $\boldsymbol{x}_i \in \mathscr{R}^n$ and $\boldsymbol{y} \in \{1, -1\}^m$, then the problem is called as a **classification** problem.

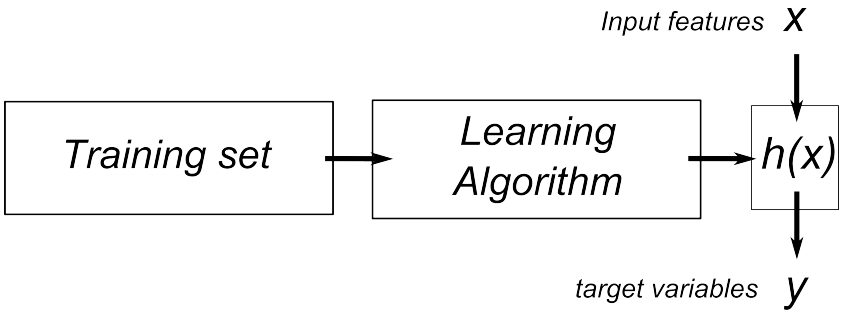


Figure 2.11: Diagram of supervised learning.

2.3.2 Cost function and probabilistic interpretation

We first demonstrate the simplest hypothesis, $h_{\boldsymbol{\theta}}(\boldsymbol{x})$, for both regression and classification problem. Then we formulate it to a more generalized form.

Regression problem

When faced with a regression problem, let us assume that the target variables and the inputs are related via the equation

$$y_i = \boldsymbol{\theta}^T \boldsymbol{x}_i + \epsilon_i, \tag{2.24}$$

where $\boldsymbol{\theta} \in \mathscr{R}^n$ and ϵ_i is an error term that captures either unmodeled effects or random noise. Assuming that ϵ_i are distributed IID according to a Gaussian distribution with mean zero and some variance σ^2. We can write this as $\epsilon_i \sim \mathscr{N}(0, \sigma^2)$. The density function is given by

$$p(\epsilon_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\Big(-\frac{\epsilon_i^2}{2\sigma^2}\Big) \tag{2.25}$$

$$\Rightarrow p(y_i|\boldsymbol{x}_i;\epsilon_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\Big(-\frac{(y_i - \boldsymbol{\theta}^T\boldsymbol{x}_i)^2}{2\sigma^2}\Big), \tag{2.26}$$

where the notation $p(y_i|\boldsymbol{x}_i, \boldsymbol{\theta})$ indicates that this is the distribution of y_i given $\boldsymbol{x}_i$ and parameterized by $\boldsymbol{\theta}$. We define the likelihood function, distribution of $\boldsymbol{y}$ given $\boldsymbol{x}$, for describing the probability as

$$L(\boldsymbol{\theta}) = L(\boldsymbol{\theta}; X, \boldsymbol{y}) = p(\boldsymbol{y}|X;\boldsymbol{\theta}). \tag{2.27}$$

Note that by the assumption of independence on the ϵ_i, likelihood function can also be written as

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{m} p(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}) = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\Big(-\frac{(y_i - \boldsymbol{\theta}^T\boldsymbol{x}_i)^2}{2\sigma^2}\Big). \tag{2.28}$$

In order to maximize the likelihood function, we can also transform the optimization problem to a log likelihood maximization problem as

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \log L(\boldsymbol{\theta}). \tag{2.29}$$

Hence, maximizing $L(\theta)$ gives the same results as minimizing the cost function, $C(\boldsymbol{\theta})$:

$$\theta^* = \arg\max_{\boldsymbol{\theta}} \frac{1}{2}\sum_{i=1}^{m} C(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \frac{1}{2}\sum_{i=1}^{m} (y_i - \boldsymbol{\theta}^T x_i)^2. \tag{2.30}$$

Classification problem

For the classification problem, the target variable y is confined to a set of values as $\{1,-1\}$ and the form of hypothesis $h_{\boldsymbol{\theta}}$ is chosen as

$$h_{\boldsymbol{\theta}}(x) = g(\boldsymbol{\theta}^T \boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{\theta}^T \boldsymbol{x}}}, \tag{2.31}$$

where g function is called logistic function or sigmoid function. Let us assume that

$$p(y=1|x;\boldsymbol{\theta}) = h_{\theta}(\boldsymbol{x}). \tag{2.32}$$

With the property of sigmoid function, we have the likelihood function as

$$p(y=-1|x;\boldsymbol{\theta}) = 1-h_{\boldsymbol{\theta}}(\boldsymbol{x}) = h_{\boldsymbol{\theta}}(-\boldsymbol{x}). \tag{2.33}$$

After summarizing the two previous equations, the likelihood function for classification can be written as

$$p(y|x;\boldsymbol{\theta}) = (h_{\boldsymbol{\theta}}(y\boldsymbol{x})). \tag{2.34}$$

Assuming that m training samples were generated independently, we can then write down the likelihood function as

$$\begin{aligned} L(\boldsymbol{\theta}) =& p(\boldsymbol{y}|X;\boldsymbol{\theta}) && (2.35)\\ =& \prod_{i=1}^{m} p(y_i|\boldsymbol{x}_i;\boldsymbol{\theta}) && (2.36)\\ =& \prod_{i=1}^{m} (h_{\boldsymbol{\theta}}(y_i\boldsymbol{x}_i)). && (2.37) \end{aligned}$$

Same as the last section, optimal $\boldsymbol{\theta}^*$ can be obtained by maximizing the likelihood function, which is equivalent to minimize the following cost function, $C(\theta)$,

$$C(\boldsymbol{\theta}) = \sum_{i=1}^{m} \log(h_{\boldsymbol{\theta}}(y_i\boldsymbol{x}_i)). \tag{2.38}$$

where $\boldsymbol{\theta}$ is a vector of scalars corresponding to each element in $\boldsymbol{x}_i$ and $h_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = (1+e^{\boldsymbol{\theta}^T \boldsymbol{x}_i})^{-1}$. With the obtained $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} C(\boldsymbol{\theta})$, we have

$$\hat{y}(\boldsymbol{x}_i) = \begin{cases} 1, & \text{when } h_{\boldsymbol{\theta}^*}(\boldsymbol{x}_i) \geq 0.5, \\ -1, & \text{when } h_{\boldsymbol{\theta}^*}(\boldsymbol{x}_i) < 0.5. \end{cases} \tag{2.39}$$

2.3.3 Generalized Linear Model (GLM)

In the regression example, we had $y|\boldsymbol{x};\boldsymbol{\theta} \sim \mathcal{N}(\mu,\sigma^2)$, and in the classification one, $y|\boldsymbol{x};\boldsymbol{\theta} \sim$ Bernoulli(ϕ). In this section, both of these methods are special cases of a broader family of models, called Generalized Linear Model (GLM). We say that likelihood function can be written in a more generalized form as

$$p(y;\eta) = b(y_i)\exp(\eta^T T(y) - a(\eta)). \tag{2.40}$$

Here, η is called the natural parameter of the distribution; $T(y)$ is the sufficient statistic and $a(\eta)$ is the log partition function. For regression problem, choosing

$$\begin{aligned} \eta &= \mu = h_{\boldsymbol{\theta}}(\boldsymbol{x}), \\ T(y) &= y, \\ a(\eta) &= \frac{\eta^2}{2} = \frac{\mu^2}{2}, \\ b(y) &= \frac{1}{\sqrt{2\pi}}\exp\Big(\frac{-y^2}{2}\Big) \end{aligned}$$

makes the likelihood function become exactly as Eq. (2.27). Moreover, for classification problem, we can choose the configuration as

$$\begin{aligned} \eta &= \log\Big(\frac{\phi}{1-\phi}\Big) = \log\Big(\frac{h_{\boldsymbol{\theta}}(\boldsymbol{x})}{1-h_{\boldsymbol{\theta}}(\boldsymbol{x})}\Big), \\ T(y) &= y, \\ a(\eta) &= -\log(1-\phi) = \log(1+e^{\eta}), \\ b(y) &= 1, \end{aligned}$$

which makes the likelihood function same as Eq. (2.37). There are many other distributions that are members of the exponential family: The multinomial, the Poisson, the gamma and the exponential, etc. Each member has its corresponding advantages on treating certain type of problems, i.e., Poisson distribution is good for predicting the count-data.

2.3.4 Gradient descent algorithm

In order to maximize the likelihood function of GLM, Eq. (2.40), gradient descent method starts with an initial $\boldsymbol{\theta}$ and repeatedly performs the update as

$$\theta_j := \theta_j - \alpha\frac{\partial}{\partial\theta_i}C(\boldsymbol{\theta}) = \theta_j - \alpha\sum_{i=1}^{m}(y_i - h_\theta(\boldsymbol{x}))x_j, \tag{2.41}$$

where α represents the learning rate, $C(\boldsymbol{\theta}) = \sum_{i=1}^{m}\log p(y_i : \eta(\boldsymbol{\theta},\boldsymbol{x}_i))$ and two examples are shown in Eq. (2.30,2.38). For both cases, it can be proved that the cost function $C(\boldsymbol{\theta})$ is convex, therefore gradient descent will converge to a unique $\boldsymbol{\theta}^*$.

2.3.5 Support Vector Machine (SVM)

Support vector machine introduced in [41][17][13] treat the same question as classification problem with the same training set, $\mathscr{X} = \{\boldsymbol{x}_i \in \mathscr{R}^n\}$ and $\boldsymbol{y} \in \{1,-1\}^m$,

Margins

Assuming there is an hyperplane $\boldsymbol{w}^T\boldsymbol{x} + b = 0$, we want to utilize it for separating a set of data $\{y_i, \boldsymbol{x}_i\}$. Given a $\boldsymbol{x}$, the distance of this point to the plane is

$$\text{distance} = \frac{1}{||\boldsymbol{w}||}|\boldsymbol{w}^T\boldsymbol{x} + b|. \tag{2.42}$$

Therefore, we formulate an optimization problem as follows: to find out the hyperplane parameters, $\boldsymbol{w}, b$, that maximize the minimum distance (margin)

$$\begin{aligned} &\max_{b,\boldsymbol{w}} \min_{i=1,\cdots,m} \frac{1}{||\boldsymbol{w}||} y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b), \\ &\text{s.t. } \forall i,\ y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) > 0. \end{aligned} \tag{2.43}$$

The optimization problem will remain the same if we scale the constraint saying that $\min_{i=1,\cdots,m} y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) = 1$. The optimization problem then becomes

$$\begin{aligned} &\max_{b,\boldsymbol{w}} \frac{1}{||\boldsymbol{w}||}, \\ &\text{s.t. } \forall i,\ y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1. \end{aligned} \quad \Rightarrow \quad \begin{aligned} &\min_{b,\boldsymbol{w}} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}, \\ &\text{s.t. } \forall i,\ y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1. \end{aligned}$$

By solving the dual problem of quadratic optimization problem,

$$\max_{\text{all } \alpha_i \geq 0} \Big[\min_{b,\boldsymbol{w}} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \sum_{i=1}^{M} \alpha_i \Big(1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)\Big)\Big], \tag{2.44}$$

the obtained $\boldsymbol{w} = \sum_{i=1}^{m} \alpha_i(y_i\boldsymbol{x}_i)$ and $b = y_i - \boldsymbol{w}^T\boldsymbol{x}_i$ when $\alpha_i > 0$ define the hypothesis hyperplane as

$$g(\boldsymbol{x}) = \text{sign}\Big[\Big(\sum_i \alpha_i y_i \boldsymbol{x}_i\Big)\boldsymbol{x} + b\Big], \tag{2.45}$$

where α_i is the Lagrange Multiplier. The support vectors on the boundary will satisfy $\alpha_i > 0$. Otherwise, $\alpha_i = 0$.

Kernel

As we can observe in the previous section, $\boldsymbol{w}$ has the same dimension as $\boldsymbol{x}_i$. If we want to increase more VC-dimension to our learning model, we can introduce a non-linear function $\boldsymbol{z}_i = \phi(\boldsymbol{x}_i)$, where $\boldsymbol{z}_i$ could be any element of $\mathscr{R}^d$. Therefore, the optimization becomes

$$\begin{aligned} &\min_{b,\boldsymbol{w}} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}, \\ &\text{s.t. } \forall i,\ y_i(\boldsymbol{w}^T\boldsymbol{z}_i + b) \geq 1. \end{aligned}$$

Like the method that solves the optimization problem in the previous section, we obtain similar results with $\boldsymbol{z}_i = \phi(\boldsymbol{x}_i)$ and Eq. (2.45) becomes

$$g(\boldsymbol{x}) = \text{sign}\Big[\Big(\sum_i \alpha_i y_i \phi(\boldsymbol{x}_i)\Big)\phi(\boldsymbol{x}) + b\Big] = \text{sign}\Big[\sum_i \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b\Big], \tag{2.46}$$

where $K(\boldsymbol{x}_i, \boldsymbol{x})$ is the kernel function. The definitions and the physical meaning of kernel are listed as

- **Linear Kernel:** $K(\boldsymbol{x}_i, \boldsymbol{x}) = \boldsymbol{x}_i^T \boldsymbol{x}$
  Linear kernel makes the question back to the original problem, which has the limited VC-dimension to the dimension of $\boldsymbol{x}$.

- **Polynomial Kernel:** $K(\boldsymbol{x}_i, \boldsymbol{x}) = (\zeta + \gamma \boldsymbol{x}_i^T \boldsymbol{x})^Q$
  Polynomial kernel provides more VC dimension than linear kernel but less than infinity.

- **Gaussian Kernel:** $K(\boldsymbol{x}_i, \boldsymbol{x}) = \exp(-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}\|^2)$
  Gaussian kernel uses Radial basis function (RBF) to make VC-dimension to infinite and also less parameters are needed to control compared to the polynomial kernel.

Soft-margin SVM

Different from the problem formulation introduced in the previous section whose hyperplane can always separate the training set, the soft-margin SVM optimization problem provides the optimal $\boldsymbol{w}$ with a relaxation of margin ξ_i.

$$\begin{aligned} \min_{\boldsymbol{w}, b, \xi} \quad & \frac{1}{2}\boldsymbol{w}^T \boldsymbol{w} + C \sum_{i=1}^{m} \xi_i \\ \text{subject to} \quad & y_i(\boldsymbol{w}^T \phi(\boldsymbol{x}_i) + b) \leq 1 - \xi_i, \\ & \xi_i \geq 0. \end{aligned} \tag{2.47}$$

In reality, soft-margin SVM is common in applications, because always finding out a hyperplane to separate all the positive and negative training sets is difficult. Even though using SVM with Gaussian kernel could also realize a well-separated boundary. Some overfitting could be generated.

2.3.6 Overviews of machine learning techniques

In addition to the machine techniques, GLM and SVM that we applied and introduced in this thesis, we provide an simple overview to some current and popular machine learning techniques, such as Decision Tree, Random Forest, k-Nearest-Neighbor and Neural Network which are not used in this thesis but may be applied in the future works.

Decision tree

Decision tree is a machine technique that mimics the humans' behavior. Here we present a basic decision tree algorithm with original data set $\mathscr{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$. Four configurations need to be made before launching decision tree algorithm: number of branches, branching criteria, termination criteria and base hypothesis $g_t(\boldsymbol{x})$. If termination criteria met, hypothesis function $g_t(\boldsymbol{x})$ will be transmitted back otherwise $\mathscr{D}$ will be split into several $\mathscr{D}_c = \{(\boldsymbol{x}_i, y_i) : b(\boldsymbol{x}_i) = c\}$ by learning the branching function, $b(\boldsymbol{x})$ which is obtained as

$$b(\boldsymbol{x}) = \arg\min_{h(\boldsymbol{x})} \sum_{c=1}^{C} |\mathscr{D}_c \text{with } h| \times \text{impurity}(\mathscr{D}_c \text{with } h), \tag{2.48}$$

Where $C = 2$ is the simplest. Branching is forced to terminate when all $\boldsymbol{y}_n$ are the same or all $\boldsymbol{x}_n$ are the same. For classification, the popular choice of impurity function is

$$\text{impurity}(\mathscr{D}) = \frac{1}{m} \sum_{i=1}^{m} (y_i - \bar{y})^2,$$

and the function for regression is

$$\text{impurity}(\mathscr{D}) = 1 - \sum_{k=1}^{K} \Big(\frac{\sum_{i=1}^{m} [y_i = k]}{m} \Big)^2.$$

By the concept of divide and conquer, the problem is reduced from $G(\boldsymbol{x})$ to several subproblem, $G_c(\boldsymbol{x})$.

$$G(\boldsymbol{x}) = \sum_{c=1}^{C} [b(\boldsymbol{x}) = c] G_c(\boldsymbol{x}). \tag{2.49}$$

For each $G_c(\boldsymbol{x})$ the same process above will be executed another time. The learning process stops when the decision tree becomes fully-grown tree and that $E_{\text{in}}(G) = 0$. However, overfitting happens and a regularizer is needed. A decision tree with regularizer is also called pruned decision tree. The general advantages of Decision Tree learning technique are: **human-explainable**, **multiclass easily**, **categorical features easily**, **missing features easily** and **efficient non-linear training (and testing)**. Several libraries of decision tree are for example C&RT and C4.5.

Random forest

The idea of random forest is to combine bagging and fully-grown decision tree. $\tilde{\mathscr{D}}_t$ are generated by taking out several data from $\mathscr{D}$. The advantage of Random forest include **highly parallel/efficient to learn**, **inherit pros of decision tree** and **eliminate cons of fully-grown tree**. With these advantages, Random forest is more applicable than only one decision tree.

k-nearest-neighbor (KNN)

KNN classifier is one of the most basic classifiers for pattern recognition or data classification. The principle of this method is based on the intuitive concept that data instances of the same class should be closer in the feature space. As a result, for a given data point $\boldsymbol{x}$ of unknown class, we can simply compute the distance between $\boldsymbol{x}$ and all the data points in the training data, and assign the class determined by the K nearest points of $\boldsymbol{x}$.

Neural network

Neural networks is recently very popular in both research and engineering. Inspired from the mechanism of human neurons. In mathematical model of neural network, the number of layers can be configured depends on the need. Pros of neural network is able to approximate anything complex regression and classification problem if enough neurons and the number of layers are configured. Cons of this technique is more about complexity of calculation and overfitting if too many neurons are considered.

Chapter 3

Model of Real-time Streaming Traffic

Nowadays Internet provides a wide range of services and applications. Generally speaking, the traffic can be easily separated into two types, elastic data and non-elastic data. Elastic data are those non-delay-sensitive services, such as email, File Transport Protocol (FTP) and web browsing. Non-elastic data includes voice services and video services such as VoD and real-time streaming service as we have introduced in section 1.1.1. Real-time streaming services including live TV streaming and video conference services become more important according to Cisco forecasts [4]. In this chapter, we study the performance of real-time streaming services and establish a traffic model for it.

3.1 Problem statement and the state of the art

Elastic data services quality mainly evaluated users' average throughput as QoS metrics. As we mentioned in section 1.1.2, many QoE metrics are proposed for real-time streaming. In order to study the performance of real-time streaming, in [35], [21] and [57], authors chose other metrics called flow blocking rate or outage rate as the main performance metrics for real-time streaming. Based on the metric, in [36], the performance of elastic users is evaluated with the presence of streaming users using flow-level model. Here we only consider the ones related to network QoS, packet outage rate, an important QoS metrics when dealing with real-time services as mentioned in [96] and [72]. Different real-time applications have different packet delay constraints. Packets with delay larger than the delay constraint are regarded as useless. Therefore, operators need a good model to predict the packet delay performance under a given traffic intensity so as to deploy proper system capacity and to design the admission control policy accordingly.

Real-time services are specified to generate their packets periodically. Take voice service as an example, each Voice over LTE (VoLTE) user generates their packets every 20ms or longer [79]. For the real-time streaming services, packets are generated periodically as voice services but real-time streaming needs larger bandwidth and this presents some challenges for operators on dimensioning. Usually, real-time services are always considered to have higher priority compared to other services. In [75] and [74], the packet delay is calculated by the quasi-stationary property. However, it is not applied in the wireless scenario.

Contributions

Our contribution is to develop a traffic model for real-time streaming users by assuming that streaming users come to the system independently and that the packets generated by the users are served with a different serving time based on their own channel conditions and their chosen video bit rates. For the real-time streaming service, base station will serve only one user's packet at one time because compared with voice data, streaming packet size is always large enough to occupy all the RB in a Transmission Time Interval (TTI). Considering with the packet delay constraint for different type of streaming services, we calculate within chapter, the maximum capacity of the real-time streaming system under the constraint that 95% of packets have a delay lower than a specific application delay, D. Our other contributions include:

- Development of a model for calculating the capacity of real-time streaming services considering the packet delays performance.
- Proposition a simpler calculation method by using fluid model.
- Extension and validation of our model with fast fading effects.

Chapter organization

Chapter 3, is organized as follows: in section 3.2, we introduce the system model with quasi-stationary regime. We then calculate the packet delay distribution considering the system load described by flow-level dynamics. In section 3.3, we extend the model to multiple classes of users representing users with different channel conditions and video codec usage. In section 3.4, we use the LTE system parameters to simulate the maximum load for different codec configuration and show the applicability of fluid model for LTE real-time services. Finally in section 3.5, model is validated with taking fast fading effects into consideration. It is shown that fast fading can be modeled.

3.2 Flow-level and packet-level model

We utilize flow level dynamics to describe the users dynamics in the system, which has been introduced in section 2.2.

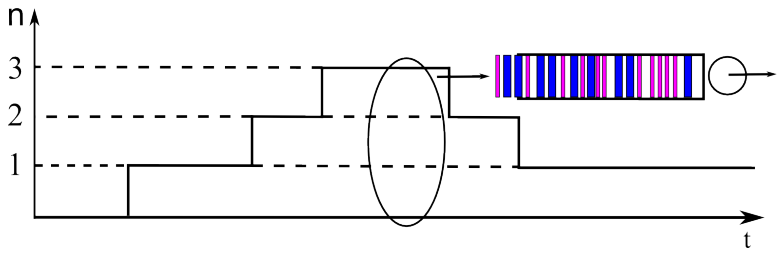


Figure 3.1: Packet arriving scheme with two level modeling.

In order to obtain the maximum system load, we model the system at two levels, flow level and packet level as Fig. 3.1. Instead of using Markov Modulated Poisson Process (MMPP)[65] as our arrival process, we assume that the flow-level dynamics occur on a relatively slow time scale compared to the packet level dynamics, which is referred to the quasi-stationary property shown in [75]. In reality, a streaming service generated by a user will stay in a time scale of seconds and packet service time is always in a time scale of milliseconds. Therefore, the packet level delay performance will approximately reach some sort of steady state in between changes in the population of flow level model. As the packets are generated periodically and each user will generate independently its packets with an average arriving interval we model the packet arriving process as a simple Markov arrival process. We validate on the assumption of quasi-stationary property in section 3.4.1.

3.2.1 Flow-level dynamics

At the flow level, we consider firstly one class of users who have the same channel conditions and we model the number of real-time streaming users n as a continuous-time Markov Chain with arriving rate, $\lambda_f = A_f^{-1}$, inverse of flow arriving interval and serving rate, $\mu_f = S_f^{-1}$, inverse of flow serving interval which are independent from the other user's coming and departure behavior. Based on the Erlang formula [32], we know that the stationary state distribution with infinite and finite users can be expressed like

$$\pi_f(n) = \begin{cases} {\rho_f}^n e^{-\rho_f} \dfrac{}{n!} & \text{, when } n \in [0, \infty] \\ \dfrac{\frac{\rho_f^n}{n!}}{1 + \rho_f + \cdots + \frac{{\rho_f}^m}{m!}} & \text{, when } n \in [0, m] \end{cases} \tag{3.1}$$

where $\rho_f = \frac{\lambda_f}{\mu_f} = \frac{S_f}{A_f}$ represents the flow level load for the real-time streaming user.

3.2.2 Packet-level dynamics

Based on the quasi-stationary regime, each state n, standing for number of users at flow level will correspond to a packet-level regime In the packet queue, we assume that each user will generate its service packets periodically with fixed interval A_p and will be served by the base station with fixed interval S_p. As each user will generate its streaming packets periodically and many users generate the packets at different time. The packet arrival is random and we approximate it to a Poisson process, we use M/D/1 queue to model real-time streaming system at the packet level. At state n, we model the packet arriving behavior as a Poisson process with arriving rate:

$$\lambda_p(n) = \frac{n}{A_p} \tag{3.2}$$

We consider that all the users belong to the same channel condition. The packet departure rate at state n is independent of state n: $\mu_p(n) = {S_p}^{-1}$. With n users in the system, using the

two previous equations, we define the load of packet-level queue as

$$\rho_p(n) = \frac{nS_p}{A_p} = n\rho_p, \quad \text{where } \rho_p = \frac{S_p}{A_p} \tag{3.3}$$

With the detailed derivation of the CDF function in [54], the waiting time distribution is shown in equation (3.4).

$$P_n(T \le x) = \begin{cases} 0 & , \ \rho_p(n) \ge 1 \\ (1-n\rho_p)\sum_{k=0}^{\lfloor x' \rfloor} \frac{(n\rho_p(k-x'))^k}{k!} e^{n\rho_p(k-x')} & , \ \rho_p(n) < 1 \end{cases}$$

where function $\lfloor x \rfloor$ represents the largest integer less than or equal to x variable and $x' = \frac{x}{S_p}$. Because this equation gives us the waiting time distribution, to get the response time distribution we just need to shift the distribution by an S_p. Under the assumption of quasi-stationary two-level system and based on the Bayesian Theorem, the overall delay distribution, $P(T \le x)$ is the average delay of the delay distribution of each state, n. Therefore, with equation (3.1) and (3.4), we obtain

$$P(T \le x) = \sum_n \pi_f(n) P_n(T \le x). \tag{3.4}$$

Because any packet delay larger than a given delay constraint, D, is useless for the delay sensitive service, we can obtain the packet outage rate as

$$\gamma(D) = P(T > D) \tag{3.5}$$

Given certain ρ_p, tolerated packet outage rate ϵ and a certain delay contraint D, we are able to calculate maximum ρ_f, system load, making $\gamma(D) = \epsilon$.

3.2.3 Fluid model approximation

Seeing the packet traffic as fluid, we propose fluid model approximation to simplify packet level model without considering the delay constraint. Because when system enters the overloaded status, delay of packet will become infinity and both the packets out of delay constraint and dropped packets are seen useless for the service, the overall packet outage rate is composed of the indication function expressed as the following equation,

$$\gamma_{\text{fluid}} = \sum_n \pi_f(n) \mathbf{1}_{\{\rho_p(n)>1\}} < \epsilon \tag{3.6}$$

where $\rho_p(n)$ is in equation (3.3) and $\rho_p = S_p/A_p$. Based on the tolerated outage rate, ϵ, we can calculate the acceptable traffic, ρ_f, given a certain resource in the packet level ρ_p by verifying the equation (3.6). Seeing the equation (3.5), we get the following relationship between the two packet outage rate models,

$$\lim_{D\to\infty} \gamma(D) = \gamma_{\text{fluid}} \tag{3.7}$$

which shows that fluid model is a lower bound of two-level model and it is independent of value D.

3.3 Extension to heterogeneous radio conditions

From the point of view of system dimensioning, users might use different codec rate and might have different channel conditions. Therefore, we extend our model to multiple-class users with modified M/D/1 model and fluid model. In the case of multiple classes, we model the system with multiple classes of users having different packet serving times. In addition because of the difficulty to get the closed form of M/D/1 with multiple classes and multiple serving times, fluid model could become a good model to facilitate the calculation.

3.3.1 Flow-level dynamics

In the previous section, we use flow level dynamics to model the number of users in the system. Assuming there are K classes of users which represent the users with different channel conditions and each class $k \in \{1, \cdots, K\}$ has real-time streaming service with Poisson arrival rate λ_k and departure rate μ_k. With the two parameters, we denote processor load of class k by $\rho_k = \frac{\lambda_k}{\mu_k}$. We denote by $n_k(t)$ the number of calls of a given class requesting streaming at time t and $\boldsymbol{n}(t) = (n_1(t), \cdots, n_K(t))$ denotes the number of flows in each class. Based on [32], the stationary distribution of the state $\pi(\boldsymbol{n})$ describing the number of flows of each class is given by

$$\pi(\boldsymbol{n}) = \prod_{k=1}^{K} e^{-\rho_k} \frac{{\rho_k}^{n_k}}{n_k!} \tag{3.8}$$

3.3.2 Packet-level dynamics

Corresponding to different channel conditions, each class has its specific service time $S = \{S_1, S_2, \cdots S_K\}$. As more than one class of users coexist in the system, we modify the M/D/1 outage rate in equation (3.20) with $\boldsymbol{\rho}_f = (\rho_1, \cdots, \rho_K)$.

$$\gamma_{\text{MD1,m}} = \sum_{\boldsymbol{n}} \pi(\boldsymbol{n}) P_{\boldsymbol{n}}(T \leq D, \rho_p) \tag{3.9}$$

by considering more than one serving time, the CDF can be calculated by the numerical result of inverse Laplace transform obtained in equation (3.11), which is also the M/G/1 model shown in [61] with multiple discrete serving time, S_k and corresponding probability $\frac{n_k}{\bar{n}}$.

$$\tilde{P}_{\boldsymbol{n}}(s) = \mathscr{L}\Big\{P_{\boldsymbol{n}}(T \leq x)\Big\} = \frac{\rho - 1}{\Big(\lambda - s - \lambda B(s)\Big)} \tag{3.10}$$

where $B(s)$ function is expressed as

$$B(s) = \sum_{k=1}^{K} \frac{n_k}{\bar{n}} e^{-sS_k} \tag{3.11}$$

and other variables

$$\rho = \lambda\Big(\sum_k \frac{n_k}{\bar{n}} S_k\Big) = \frac{\sum_k n_k S_k}{A_p} \tag{3.12}$$

$$\lambda = \frac{\sum_k n_k}{A_p} \tag{3.13}$$

$$\bar{n} = \sum_k n_k \tag{3.14}$$

The outage rate of fluid model with multiple class of users, $\gamma_{\text{fluid,m}}$, is shown as below using the same logic of equation (3.6).

$$\gamma_{\text{fluid,m}} = \sum_{\boldsymbol{n}} \pi(\boldsymbol{n}) \mathbf{1}_{\{\rho>1\}} \tag{3.15}$$

3.4 Simulation results

In this section, we first show the validation of quasi-stationary regime and then we show the performance of M/D/1 model and fluid model with different services corresponding to different delay constraints configuration. Based on [78], the human tolerant delay for interactive service such as video conference is about 150ms. We configure the delay constraints as 500ms for live TV streaming. We show that the fluid model can be used for the simplification of live TV streaming and that it is better to stay with M/D/1 model in the dimensioning of video conference.

3.4.1 Quasi-stationary regime

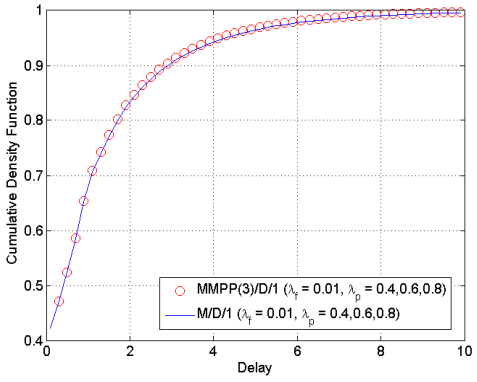


Figure 3.2: Performance comparison with $\frac{\lambda_f}{\lambda_p} \leq 0.025$

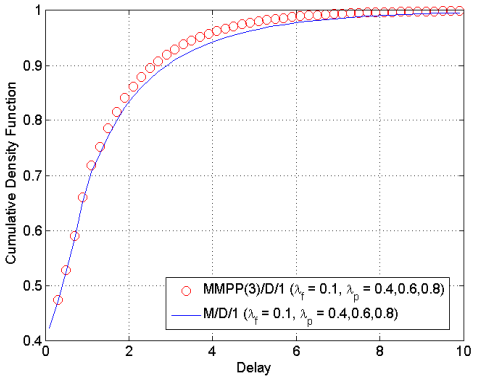


Figure 3.3: Performance comparison with $\frac{\lambda_f}{\lambda_p} \leq 0.25$

We show that the quasi-stationary regime is valid by showing the comparison between the CDF delay distribution of Markov Modulated Poisson Process (MMPP)/D/1 queue with

only three states in the upper markov chain and the CDF delay distribution of M/D/1 queue with quasi-stationary regime. Based on the definition of MMPP/D/1[65], the simulation in Fig.3.2 and Fig.3.3 show that quasi-stationary regime is valid when $\frac{\lambda_f}{\lambda_p} < 0.25$, which covers all our simulation configurations.

3.4.2 Single class model validation

In the Table. 3.1, we assume that the mean flow arriving time is $S_f = 10$s which is one hundred time larger than the mean packet arriving time $A_p = 100$ms. Based on the signal and interference noise ratio (SINR) distribution obtained in [21] and the configuration of 3GPP specification [9][8], the LTE average throughput is calculated as $\tau = 9.4$Mbps and based on different codec settings, different S_p settings are shown in the Table. 3.1, with c denoting chosen codec rate.

$$S_p = \frac{c \times A_p}{\tau} \tag{3.16}$$

It can be observed that $S_f, A_f >> S_p, A_p$, which follows the quasi-stationary regime we assume.

Parameter	Sym	Value
Mean flow arriving time (s)	A_f	$[4, 20]$
Mean flow serving time (s)	S_f	10
Mean packet arrival time (ms)	A_p	100
Mean packet serving time (ms)	S_p	2Mbps codec → 21.3 1Mbps codec → 10.6 512kbps codec → 5.45 256kbps codec → 2.72
Maximum user number	m	100

Table 3.1: Simulation configuration for single class users

In Fig.3.4 and Fig.3.5, we show that configuration with 2Mbps and 512kbps codec rate and users have the same service time $S_p = 21.3$ms and $S_p = 5.45$ms respectively, the red curve stands for the packet outage rate obtained by fluid model and the other blue curves are the results obtained from M/D/1 model with different delay constraint $D = 50\text{ms}, 150\text{ms}$ and 500ms. It can be seen that the packet outage rates are lower bounded by fluid model and when delay constraint approaches to 500ms, the performance of fluid model and M/D/1 are the same. Therefore, we say that fluid model is enough to describe the packet-level performance of streaming services for service like live TV streaming. For the interactive services like video conference, it is better to use M/D/1 model. In Table. 3.2, we show the maximum flow-level load obtained by the simulation results for different types of services and different codec configurations which limit the packet outage rate under 5%.

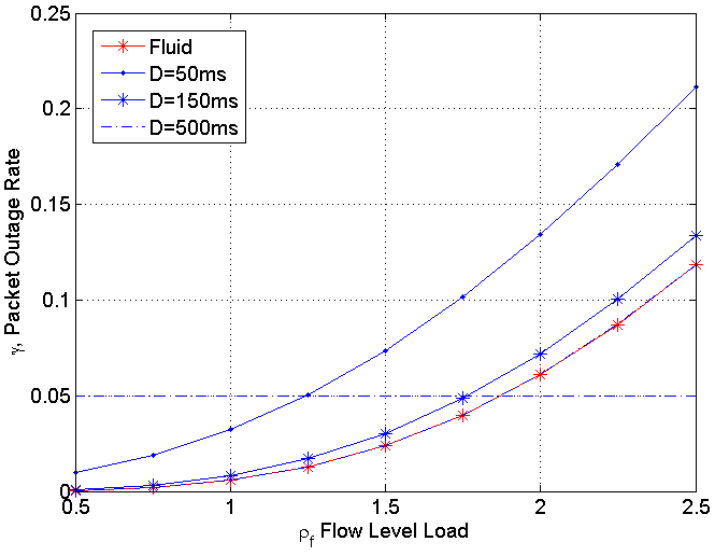


Figure 3.4: Packet outage rate with 2Mbps codec.

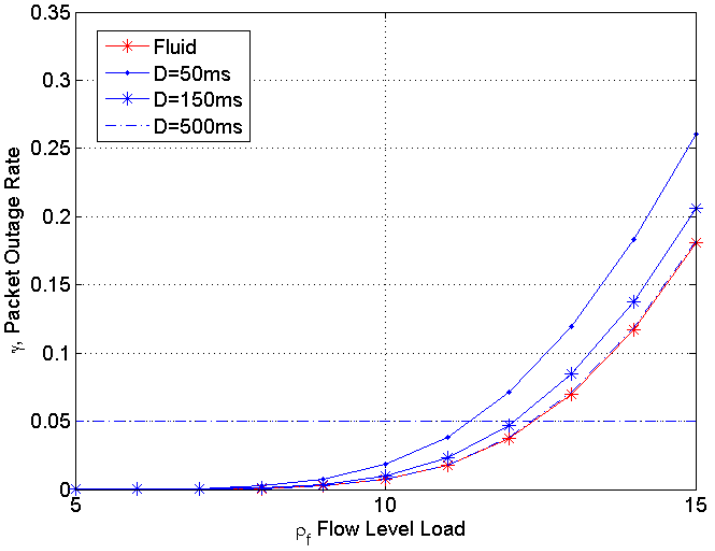


Figure 3.5: Packet outage rate with 512kbps codec.

3.4.3 Multiple class model validation

To validate the extension of our models to the multiple class scenario, we take an example of users with two classes, $S = \{S_c, S_e\}$, representing cell edge and cell center users respectively. In the validation, we assume that users utilize the codec with coding rate 512kbps. Based on the same SINR distribution and equation (3.16) in section 3.4, we calculated the

Codec Configuration	Max Flow-level Load for Live TV Streaming	Max Flow-level Load for Video Conference
2Mbps	1.9	1.75
1Mbps	5.5	5
512kbps	12.4	12
256kpbs	27.6	27.4

Table 3.2: Maximum flow-level load with different codec

average throughput and one-packet serving time of cell center and cell edge users as Table. 3.3.

Users' Class	Rate	Serving time
Cell center	14.63 Mbps	$S_c = 3.50$ms
Cell edge	3.73 Mbps	$S_e = 13.73$ms

Table 3.3: Serving time of cell edge and cell center users

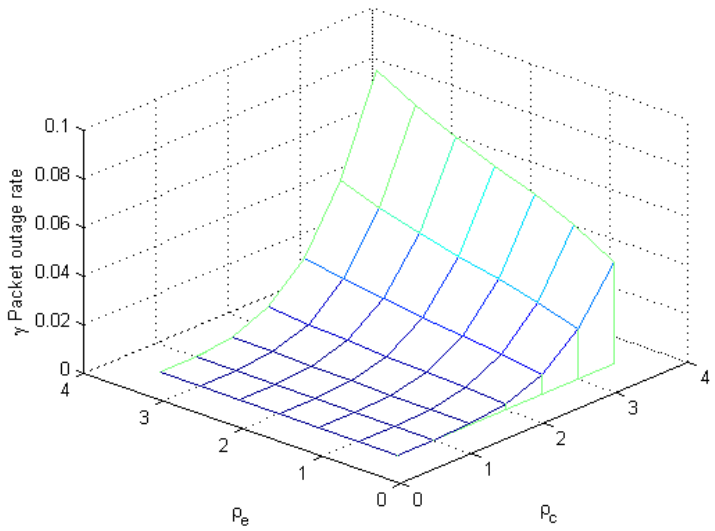
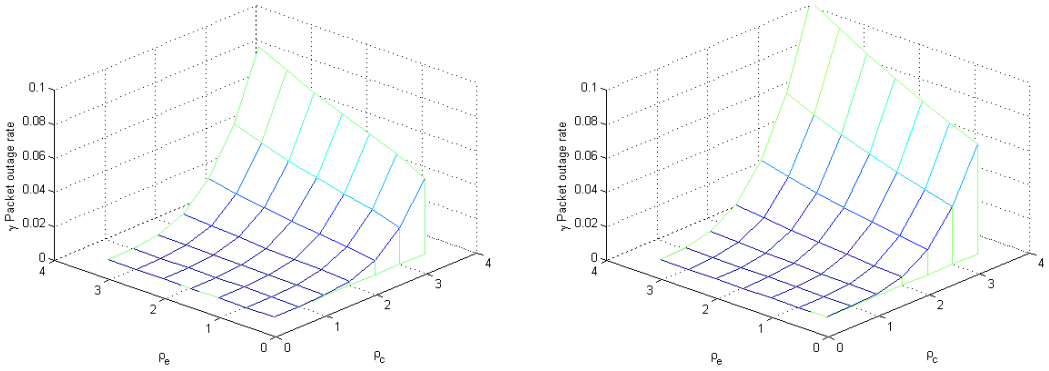


Figure 3.6: Packet outage rate calculated by fluid model with multiple class of users.

In Fig. 3.6, with ρ_e standing for the flow-level load of cell edge users and ρ_c standing for the flow-level load of cell center users, we obtain the fluid model outage value based on equation (3.15). In Fig. 3.7, it can be observed that the packet outage rate obtained



(a) Live TV Streaming (b) Video Conference

Figure 3.7: Packets outage rate for different applications.

by fluid model has a 2% difference with obtained with delay constraint, 500ms and there is a 25% difference with delay constraint 150ms. Therefore, we say that fluid model is enough to describe the delay performance with multi-class of users for the service like live TV streaming.

3.5 Validation with fading effect

For the completeness of the study, we consider the fast fading effects described by using Rayleigh distribution. Considering a Rayleigh fading with $\sigma = \sqrt{2/\pi}$ and coherent time, 1ms, we show the simulation configuration of five different serving times for each user class in Table 3.4. We use the same average serving time as previous simulation, 3.5ms, 13.73ms.

User Class	Portion	S_p	User Class	Portion	S_p
cell center	3.51%	1.7ms	cell edge	2.19%	9.7ms
	31.89%	2.6ms		20.66%	11.5ms
	36.89%	3.5ms		39.85%	13.3ms
	25.39%	4.6ms		26.73%	15.2ms
	2.29%	6.4ms		10.57%	17.5ms

Table 3.4: Serving time and probability distribution of two classes of users with fading effect consideration.

We then show the outage rate of fluid model as

$$\gamma_{\text{fluid,fading}} = \sum_{\boldsymbol{n}} \pi(\boldsymbol{n}) \mathbf{1}_{\rho>1} \tag{3.17}$$

where

$$\rho = \frac{\sum_k n_k \sum_i p_{k,i} S_{k,i}}{A_p} \tag{3.18}$$

and the outage rate of exact two-level model can be obtained by adjusting the equation (3.11) to

$$\tilde{P}_{\boldsymbol{n}}(s) = \mathscr{L}\left\{P_{\boldsymbol{n}}(T \le x)\right\} = (\rho - 1)\Big(\lambda - s - \lambda \sum_{k=1}^{K} \frac{n_k}{\bar{n}} \sum_i p_{k,i} e^{-sS_{k,i}}\Big)^{-1} \tag{3.19}$$

$$\gamma_{\mathrm{MD1,m}} = \sum_{\boldsymbol{n}} \pi(\boldsymbol{n}) P_{\boldsymbol{n}}(T \le D) \tag{3.20}$$

where $p_{k,i}$ stands for the portion of different channel conditions in the same class of users and $S_{k,i}$ stands for the serving time in the Table. 3.4. In Fig. 3.8, we show the packet outage rate obtained by fluid model with consideration of fast fading. It also shows that the fast fading effect will have the same outage rate as the one without fading effect in Fig. 3.6. In Fig. 3.9, we show the packet outage rate of modified M/D/1 model with different D values. Based on the simulation results, we conclude the fluid model is useful for live TV streaming with $D \geq$ 500ms and when considering service as video conference, it is better to utilize modified M/D/1 model. In addition, we can also show the same results for slow fading with log-normal distribution, $\sigma = 3$dB, by the same method.

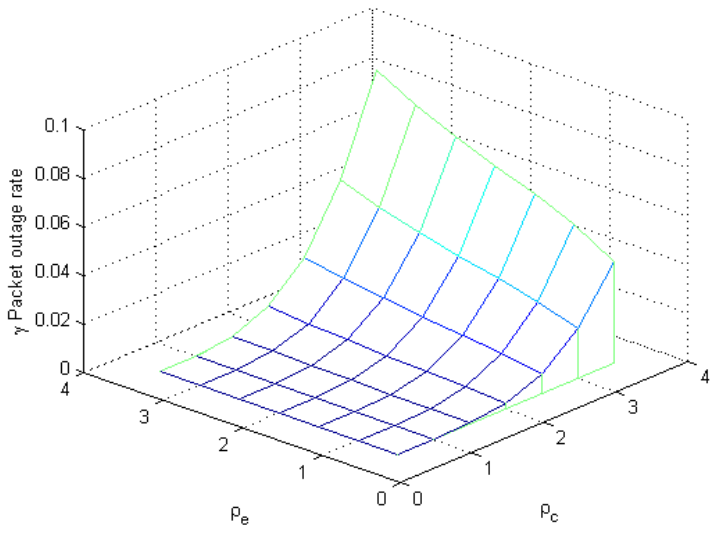
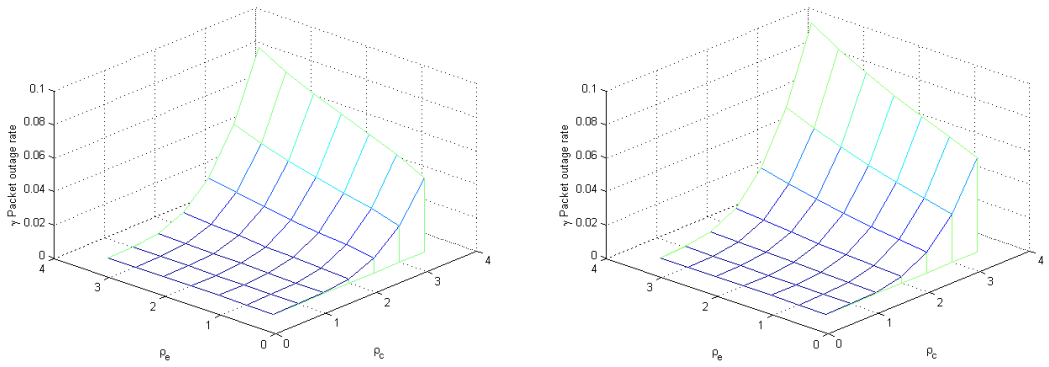


Figure 3.8: Packets outage rate of fluid model with fast fading channel effect.



(a) Live TV Streaming (b) Video Conference

Figure 3.9: Packets outage rate for different applications with fast fading channel effect.

3.6 Summary

In this chapter, we include the impact of flow-level dynamics into the calculation of the packet delay of real-time video service. Under the quasi-stationary property, we calculate the packet delay performance with M/D/1 queue at packet level and combine it with the stationary distribution of flow-level dynamics. To facilitate the calculation, we propose a substitution of M/D/1 packet-level model by fluid model. In the simulation, it is shown that the fluid model approximates well the performance of M/D/1 queue under the LTE networks configuration for the delay tolerant of application like live TV video. On the other hand, for services like video conference which require smaller packet delay, an obvious difference between M/D/1 and fluid model can be observed. Thus we conclude that it is better to use the M/D/1 for packet level modeling. Model extension with multiple classes case are validated both for M/D/1 and fluid model, also considering with the fast fading effects.

Chapter 4

Model of Adaptive Streaming Traffic

In the previous chapter, we have presented a model of real-time streaming traffic. As we know that on-demand video account for larger proportion of traffic than the real-time streaming and the fact that services like YouTube and Netflix [4] becomes very popular, a traffic model to analyze the impact of different configuration of HTTP streaming is needed, especially HTTP Adaptive Streaming (HAS) becomes a mature and popular technical solution [71][69][44][15][45]. As we mentioned in the introduction chapter, the biggest difference between real-time streaming and HTTP streaming is the existence of a video playout buffer. Moreover, TCP is the transport layer protocol used for HTTP streaming. In this chapter, we develop a general traffic model that aims at helping operators to assess the quality-of-service perceived by their users and properly dimension their networks. We apply the well-known flow-level model, where a flow can represent either a video streaming session or a elastic session.

4.1 Problem statement and state of the art

Flow-level model is widely used for evaluating the impacts of traffic for elastic traffic and real-time adaptive streaming. In chapter 2, we introduce how flow-level model can be used for evaluating the performance of elastic data. It is also applied for real-time streaming. For instance, authors of [35] investigated the integration of elastic and streaming services by modeling the real-time adaptive streaming as a flow and only provided the performance bound because the insensitivity property does not hold. However, the considered streaming services are modeled as a specific type of streaming, real-time adaptive streaming. Paper [36] examined the video performance with different scheduling methods with the video QoS expressed by a blocking rate with the same assumption of real-time adaptive streaming. Another work also considers the performance of real time streaming services on the flow level [21]. Moreover, it also focuses on the same metrics, flow blocking rate. Compared to the real-time streaming modeling, modeling for the HTTP streaming is still at the early stage.

Most of the existing works focus on the evaluation of HTTP streaming. There is no mature traffic model for HTTP adaptive streaming and its performance trade-offs.

The above mentioned flow level models focused on classical elastic and real-time streaming services and does not consider the impacts of buffer. Moreover, classical performance models developed for real-time adaptive streaming and elastic services are not suitable for HAS as the latter service has similarities with both types of traffic. Flow-level model has been applied in studies of HTTP streaming [100][99] with infinite video buffers. The KPIs like starvation probability have been computed using a detailed buffer analysis. However the mentioned model is not suitable to be adapted for evaluating the performance of HTTP adaptive streaming because of the lack of consideration for rate adaptivity. Indeed, the works that considered real-time streaming traffics were limited to real time streaming and works about HTTP streaming did not consider video bit rate adaptivity. Works of HAS include [102], where authors propose an analytical framework for HTTP adaptive streaming under the assumption of fix frames arrival rate for different video bit rates and [98], where authors model the frame arrival as Markov modulated fluid arrival. Both do not consider the impacts of other traffic and overall system loads. For further studies, it is better to combine the impacts of other traffic to the packet arrival rate, which is the most difficult part.

How to allocate resources for both streaming and elastic services becomes a question for operators. It has been well-known that wireless system capacity can be enhanced with multi-user diversity using opportunistic schedulers from [90][10][19]. Slow channel variations due to the mobility can be exploited as well even under a blind fair scheduling strategy like round-robin [11][29][38][58][37]. Applying flow-level dynamics, authors of [11] analyze the performance impacts of mobility in the presence of elastic traffic and they suggest that operators deploy opportunistic schedulers for the elastic data in order to profit from the multi-user diversity generated by the users' mobility. However, as video streaming service like YouTube and Netflix account for larger part of system traffic, if there is a model for HTTP adaptive streaming, it will facilitate operators to understand whether these suggestions are still valid for streaming services.

Main contributions

Our first contribution is to develop a flow-level model for the adaptive streaming traffic taking into account the main characteristics of this service, the presence of playout buffer, the different configurations of video chunk duration, video bit rates, scheduling schemes and users' mobility. Our second contribution is to extend this model to consider heterogeneous radio conditions and a mixed service between streaming and elastic traffic. As of measures, we believe that users are satisfied if they watch the video with a high video bit rate and if the video play back is smooth, i.e. the playout buffer never gets empty. We look at the KPIs that directly influence the Quality of Experience (QoE) of users such as the average video bit rate observed during the video session and the starvation probability, i.e. the probability that the video buffer becomes empty. Although our flow-level model is able to compute the QoS such as average video bit rate, the computation of starvation probability needs a packet-level analysis as it depends on the behavior of the player in terms of prefetching policy and the detailed buffer state. As our objective is to provide simple models that can be used for mobile network dimensioning purposes, we examine several KPIs related to the

starvation probability and they can be computed using the flow-level modeling: the deficit rate expressed as the probability that the instantaneous user throughput is lower than the chosen video bitrate, the buffer surplus representing the average buffer variation during the video download and the mean service time representing the average time to transmit a video session.

The model with the consideration of different network parameters are published in three scientific papers. We first introduce a specific flow-level model for adaptive streaming considering with buffer in [30], then we examine the impacts of the system parameters, video chunk duration, in [66] and in [73], model considering the mobility and different scheduling schemes is introduced.

Chapter organization

The remainder of this chapter is organized as follows. Section 4.2 describes the key components for deliver HAS and introduces the important configuration, video chunk duration, that we are going to examine. In section 4.3, we develop the flow-level model for adaptive streaming traffic with small and large chunk duration respectively and we define the performance metrics in section 4.4. Then we introduce different heterogeneous radio conditions and different scheduling schemes into our system 4.5 and model are extended to consider the integration of elastic and streaming services in section 4.6. Finally, the users' mobility between different capacity regions are modeled and considered in section 4.7.

4.2 System description

This section introduces two key aspects that influences the performance of HTTP adaptive streaming services delivered in wireless networks: video content configuration and wireless access network.

4.2.1 Video content configuration

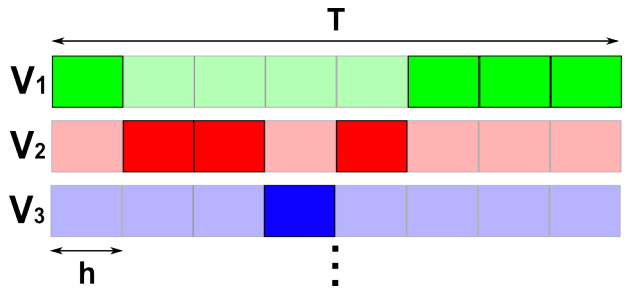


Figure 4.1: An example of video chunks delivery for a user.

According to the mechanism of HTTP adaptive streaming we have introduced in chapter **??**, a video is separated by several video chunks (segments) as Fig. 1.1 and they are requested

one after another by HTTP requests. The proper and corresponding video bit rate is selected at the beginning of each chunk download. Fig. 4.1 gives an example showing how a video download is composed by a bunch of chunks. The chunk duration, h, is a system parameters that service providers can control.

Intuitively speaking, by selecting a shorter duration, users have more chances to adapt its video bit rate. In this chapter, we are going to study the analytical results of two extreme chunk duration configurations shown in the following Table 4.1.

Configurations	Value of h	Section	Chances of bitrate transition
Significantly small chunk duration	$h \approx 0$	1	Many
Significantly large chunk duration	$h \approx \infty$	4.3.2	Zero

Table 4.1: Two extreme video chunk durations.

4.2.2 Wireless access network

In this section, we focus on the performance of adaptive streaming delivered in a typical cell as Fig. 4.2. Mobile users in the cell download the streaming traffic to they buffer by the allocated resources of the cell. We begin, for the ease of understanding of the model, by a homogeneous radio condition where all users are supposed to see a capacity R equal to the average radio condition over the cell. Section 4.5 and 4.6 will show how to extend these models to multiple radio conditions and how to integrate other services like elastic traffic.

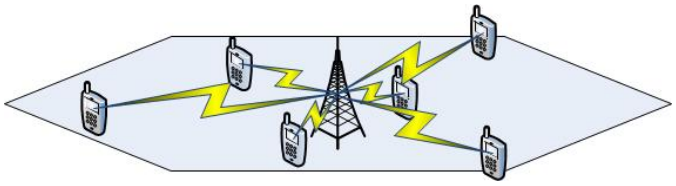


Figure 4.2: Wireless system with single capacity region

Moreover, to facilitate the flow level modeling, for the streaming system configuration, we assume that each flow has an infinite playout buffer so that each will continuously download until its corresponding video is fully downloaded. During the transmission, each flow requests the sequential video chunks according to the allocated resource. As we mention before, the streaming chunk duration impacts how users adapt their video bit rates.

4.3 System model with flow-level dynamics

In this section, we first consider adaptive streaming services, the mix with other services is considered later. In addition, we make a classical assumption that traffic flows arrive

according to a Poisson process with intensity, λ and the video duration of a flow is assumed to be independent and exponentially distributed with mean T.

4.3.1 Small chunk duration model

• Markov model

To model the dynamics of flow number, $X(t)$, we utilize the model of processor-sharing queue. With the assumption of homogeneous radio condition and small chunk duration, we model all the flows in one class. As $X(t) = x$, meaning that x flows are served in the system, the flow departure rate, $\mu(x)$, can be expressed as

$$\mu(x) = \frac{\phi(x)}{\sigma(x)}, \tag{4.1}$$

where $\phi(x)$ stands for the physical throughput allocated to all UEs of the cell at state x and $\sigma(x)$ stands for the remaining flow size at state x.

As we consider an infinite buffer size at users' side, users can profit at the maximum from the throughput allocated to them, occupying thus the whole scheduling time of cell, leading to an allocation $\phi(x) = R$ in this case. As of the remaining flow size, $\sigma(x)$, as we consider a small chunk duration leading to an instantaneous adaptation, the video bit rate, $v(x)$, at state x depends only on the number of flows x and not on the history of the system. Furthermore, as the video duration is assumed to be exponential, the memoryless property implies that the flow size at state x is also exponentially distributed with its mean

$$\sigma(x) = v(x)T. \tag{4.2}$$

$X(t)$ is thus a Markov process whose departure rates depend on the video bit rate selection. We show in the following sections how this bit rate is computed.

With Eq. (4.1) and (4.2), flow departure rate then is obtained as $\mu(x) = \frac{\phi(x)}{v(x)T}$ and the system can be easily shown to have a product-form stationary distribution computed as

$$\pi(x) = \pi(0) \prod_{n=1}^{x} \frac{\lambda}{\mu(n)}, \tag{4.3}$$

where $\pi(0) = \left(1 + \sum_{x=0}^{\infty} \prod_{n=1}^{x} \frac{\lambda}{\mu(n)}\right)^{-1}$.

• Video bit rate selection

Video bit rate chosen by each flow is decided based on the instantaneous throughput, $\gamma(x)$, that flow observes. In flow-level model, by applying round-robin scheduling policy, the instantaneous throughput can be calculated as

$$\gamma(x) = \frac{R}{x}. \tag{4.4}$$

In reality, video bit rates are not continuous. Instead, flows will select a specific video bit rate from a set of discrete video bit rates $\mathcal{V} = \{v_1, \cdots, v_I\}$, where we assume $v_1 > \cdots > v_I$. Knowing the throughput, users will select the video bit rate as

$$v(x) = \begin{cases} \lfloor \gamma(x) \rfloor, & \text{when } \gamma(x) > v_I, \\ v_I, & \text{when } \gamma(x) \leq v_I, \end{cases} \tag{4.5}$$

where $\lfloor z \rfloor$ stands for a function selecting a maximum value in $\mathcal{V}$ but lower than z. It can also be observed that $\gamma(x)$ is always equal to $v(x)$ or larger than $v(x)$ only when $\gamma(x) < v_I$. With the defined video bit rate selection mechanism, when $\gamma(x) \geq v_I$, users' video buffer will increase or remain the same. Instead, video buffer will decrease only when $\gamma(x) < v_I$.

4.3.2 Large chunk duration model

With the same system characteristic mentioned in section 1, here, we consider the streaming system with infinitely large chunk duration, where flows choose their video bit rate at the beginning of their arrivals and keep the one until the end of download. Different from the case of small chunk duration, where we model the system with one class of user who select the same video bit rate at the same time. For configuration of infinite chunk duration, multiple classes of queue are needed to describe the number of flows choosing different video bit rates at given time. Same as previous section, the discrete set of video bit rates is denoted as $\mathcal{V} = \{v_1, \cdots, v_I\}$, where $|\mathcal{V}| = I$.

Because flows with different video bit rate possess different arriving and departure rate, we denote the state of system as $\boldsymbol{x} = (x_1, \cdots, x_I) \in N^I$, where x_i represents the number of flows that chose video bit rate v_i at its arrival. The flow arrival rate of queue-i depends on the total number of flows in the system, $|\boldsymbol{x}| = \sum_i x_i$. Given a state $\boldsymbol{x}$, all the flow arrival will only select video bit rate v_i. Therefore, we have

$$\forall i,\ \lambda_i(\boldsymbol{x}) = \begin{cases} \lambda, & \text{if } v(\boldsymbol{x} + \boldsymbol{e}_i) = v_i, \\ 0, & \text{otherwise}, \end{cases} \tag{4.6}$$

where $v(\boldsymbol{x})$ is defined as Eq.(4.5) and $\boldsymbol{e}_i$ represents a vector with an unit value at class i. Applying the same concept of Eq. (4.1), flow departure rate and the allocated resource of class i is denoted as

$$\mu_i(\boldsymbol{x}) = \frac{\phi_i(\boldsymbol{x})}{v_i T}, \tag{4.7}$$

$$\phi_i(\boldsymbol{x}) = \frac{x_i R}{|\boldsymbol{x}|}, \tag{4.8}$$

with the same setting of Round-Robin scheduling. We then can calculate the stationary distribution $\pi(\boldsymbol{x})$ by using the arrival and departure rate of both cases in Eqs. (4.6-4.7) and

solving the balance equations denoted as

$$\forall \boldsymbol{x},\ \sum_i \Big(\lambda_i(\boldsymbol{x}) + \mu_i(\boldsymbol{x})\Big)\pi(\boldsymbol{x}) \\ = \sum_i \lambda_i(\boldsymbol{x} - \boldsymbol{e}_i)\pi(\boldsymbol{x} - \boldsymbol{e}_i) + \mu_i(\boldsymbol{x} + \boldsymbol{e}_i)\pi(\boldsymbol{x} + \boldsymbol{e}_i). \tag{4.9}$$

With the stationary distribution, $\pi(\boldsymbol{x})$ calculated in the case of small chunk duration and large chunk duration, we then define the key performance indicators in section 4.4.

4.3.3 Stability condition

The flow arrival rate should be smaller than the maximum flow departure rate. In the case that v_n is the video bit rate selection of n-th flow, the maximum flow departure rate implies that $\forall n, v_n = v_I$, leading to the following stability condition for both chunk duration configuations and maximum flow arrival rate, $\lambda_{\max}$:

$$\lambda < \frac{R}{v_I T} \Rightarrow \lambda_{\max} = \frac{R}{v_I T}. \tag{4.10}$$

If $v_I = 0$, the system is always stable. However, when $v_I > 0$, the system becomes unstable with a large number of arrivals.

4.4 KPIs definition

To evaluate the QoE of adaptive streaming service, we propose four key performance indicators, mean video bit rate, mean service time, deficit rate and buffer surplus. All of them are defined based on the stationary distribution $\pi(\boldsymbol{x})$.

4.4.1 Video bit rate

Mean video bit rate stands for the average video bit rate that a flow experiences during playing its video. When mean video bit rate is high, user has a better video experience. We calculate the cell-average video bit rate using the following concept,

$$\text{Video Bit Rate} = \frac{\text{Allocated Resource}}{\#\text{Flows Served}}, \tag{4.11}$$

where we devide all the allocated resource on the number of flow multiplied the mean video duration to calculate mean video bit rate. Then we define the overall mean video bit rate, $\bar{v}$ for both small and large chunk duration,

$$\bar{v} = \frac{\sum_{x:x>0} \pi(x)\phi(x)}{\lambda T}, \quad \bar{v} = \frac{\sum_{\boldsymbol{x}:|\boldsymbol{x}|>0} \pi(\boldsymbol{x}) \sum_i \phi_i(\boldsymbol{x})}{\lambda T}, \tag{4.12}$$

where $|\boldsymbol{x}| = \sum_i x_i$.

A popular QoE indicator used to evaluate streaming performance is the starvation probability [100]. Even if starvation happens only when the video bit rate is larger than the instantaneous throughput, the latter condition is not a sufficient condition for starvation as the buffer may counteract the impact of short periods of low throughput. The computation of the starvation probability has to take into account the memory of the system by introducing the buffer size in the Markovian analysis as in [100]. Here, we introduce and examine three KPIs called service time, deficit rate and buffer surplus.

4.4.2 Service time

To evaluate the starvation probability, we propose another KPIs, mean service time of a video flow, which is calculated by the Little's formula as

$$S = \frac{L}{\lambda} = \frac{\bar{x}}{\lambda}, \quad S = \frac{L}{\lambda} = \frac{\bar{\boldsymbol{x}}}{\lambda}, \tag{4.13}$$

with $\bar{x} = \sum_{x>0} x\pi(x)$, $\bar{\boldsymbol{x}} = \sum_{|\boldsymbol{x}|>0} |\boldsymbol{x}|\pi(\boldsymbol{x})$ and that $L = \bar{x}$ represents the mean number of flow. By observing S, we can imply the starvation probability, which is positively related with S. Saying that smaller S could imply smaller starvation probability.

4.4.3 Deficit rate

As [32] mentioned, flows have higher probability to stay in the state for downloading $v(x)$ because x users exist. Therefore, by weighting the corresponding metrics at state x by the number of flows, x, also called as size-biased distribution we define the following metrics. The deficit rate is equal to the probability that an ongoing flow sees its instantaneous throughput lower than its chosen video bit rate. As the adaptation of video bit rate is assumed to occur instantaneously in reaction to the variations of the observed throughput, the deficit rate is defined by the probability that the instantaneous throughput,

$$\gamma(x) = \frac{\phi(x)}{x},$$

is smaller than $v(x)$ in the case of small video chunk duration or

$$\gamma_i(x) = \frac{\phi_i(x)}{\sum_i x_i},$$

is smaller than v_i in the case of large video chunk duration. Note that for small chunk duration configuration deficit happens only when $\gamma(x) < v_I$, based on the selection mechanism, Eq. (4.5). The overall deficit rate is defined by weighting the stationary distribution at different state x with the number of flows:

$$D = \Pr\{\gamma(x) < v(x)\} = \sum_{x:x>0} \frac{x\pi(x)}{\bar{x}} \mathbf{1}_{\{\gamma(x)<v(x)\}}, \tag{4.14}$$

where **1** stands for indicator function. For the case configuring large video chunk duration,

$$D = \Pr\{\gamma_i(\boldsymbol{x}) < v_i(\boldsymbol{x})\} = \sum_{\boldsymbol{x}:|\boldsymbol{x}|>0} \frac{\pi(\boldsymbol{x})}{\bar{\boldsymbol{x}}} \sum_i x_i \mathbf{1}_{\{\gamma_i(\boldsymbol{x})<v_i\}}. \tag{4.15}$$

Starvation probability is also positively related to the deficit rate. Therefore, larger deficit rate will cause larger starvation probability.

4.4.4 Buffer surplus

We also introduce another performance metric called buffer surplus, which represents the average buffer variation of each flow in a second. It is calculated by weighting all the buffer variation, $\frac{\gamma(x)-v(x)}{v(x)}$, at each state x as

$$B = \sum_{x:x>0} \frac{x\pi(x)}{\bar{x}} \Big(\frac{\gamma(x)-v(x)}{v(x)} \Big), \tag{4.16}$$

for the small video chunk duration configuration. When $\gamma(x) > v(x)$, users' buffer accumulates certain amount of duration of video. When $\gamma(x) < v(x)$, then user starts to consume the video packets stored in the buffer, which reduces the values of average buffer surplus. For the case of large chunk duration, buffer surplus is calculated as

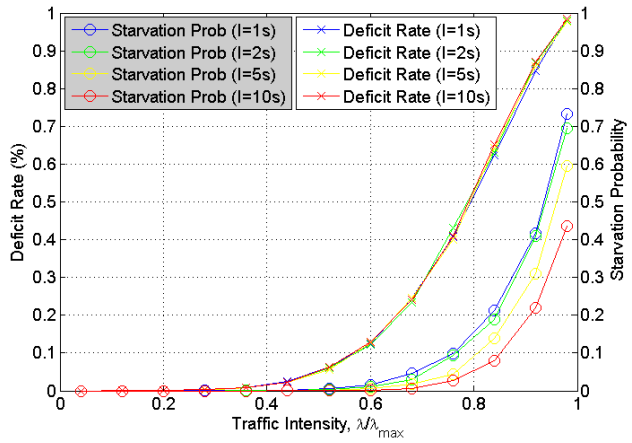
$$B = \sum_{\boldsymbol{x}:|\boldsymbol{x}|>0} \frac{\pi(\boldsymbol{x})}{\bar{\boldsymbol{x}}} \sum_i x_i \Big(\frac{\gamma_i(\boldsymbol{x})-v_i}{v_i} \Big). \tag{4.17}$$

Larger buffer surplus will decrease the starvation probability. Therefore, it is negatively related to the starvation probability.

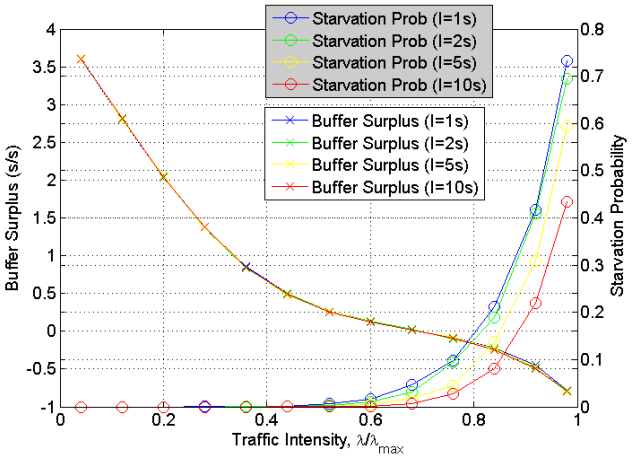
4.4.5 Performance of Different KPIs v.s. Starvation Probability

In order to demonstrate the relationship between desired starvation probability and our introduced KPIs, we simulate the real starvation probability with event-based simulation considering the initial buffer, I, which is a parameter not included in the flow-level model. For the simulation we set up the following configurations, $R = 10$Mbps, $T = 20$s, and $(v_1, v_2) = (2, 1)$Mbps. In Fig. 4.3, we can observe the correlation of starvation probability and our defined KPIs and that we can first observe that three defined KPIs are not dependent on the initial buffer values, I. Moreover, we can also observe some simple correlation between our proposed KPIs and the stationary probability. For example, deficit rate is always the upper bound of starvation probability. Second, when buffer surplus value is equal to zero, starvation probability is around $10-20\%$. Third, mean service time should be smaller than the whole video duration. Otherwise, the starvation probability becomes high.

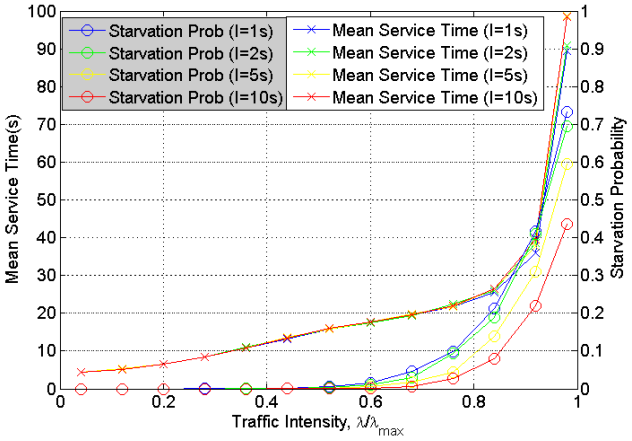
In chapter 6, that works that we predict the QoE of video streaming using each users' features is inspired by the insufficiency of correlation between QoE and our proposed KPIs.



(a) Performance of Deficit Rate



(b) Performance of Buffer Surplus



(c) Performance of Mean Service Time

Figure 4.3: Performance comparison between proposed QoS metrics and starvation probability

4.5 Scheduling schemes

Scheduling is an important topic discussed a lot in the system design of LTE. It is also one of the most important parameters that internet service providers can control. In this section, we first introduce the model considering heterogeneous channel condition and we present several scheduling schemes implemented knowing the channel information.

4.5.1 Heterogeneous radio conditions

Based on the 3GPP LTE-A standards [43], users with various positions have different discrete Channel Quality Indicator (CQI), for example from CQI-1 to CQI-15. Therefore, traffic flows can be separated into several classes having a radio condition R_k, $R_k \in \mathscr{R} = \{R_1, \cdots, R_K\}$, where $R_1 \geq \cdots \geq R_K$. Each class accounts for a proportion of flow arrivals, p_k, with $\sum_{k=1}^{K} p_k = 1$. Then we present the model with corresponding performance results

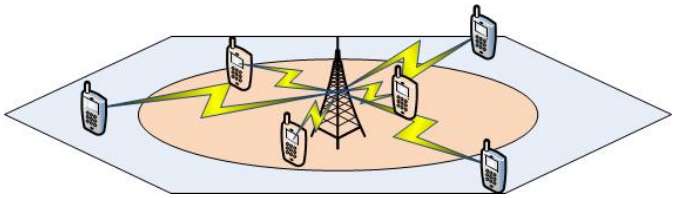


Figure 4.4: Wireless system with multiple capacity region.

under round-robin, max C/I [63] , max-min [23] and opportunistic scheduling schemes. As we model each cell by a set of K regions. In each region, radio conditions are supposed to be homogeneous and thus users are served at the same physical data rate on the downlink. In the simple case of two regions illustrated by Figure 4.5, users may be close to the cell center and experience good radio conditions (light gray) or close to the cell edge and suffer from bad radio conditions (dark gray). We model each region by a queue with a specific service rate corresponding to the physical data rate in this region; since all users in the cell share the same radio resources, each cell can be viewed as a set of K parallel queues with coupled processors. The precise coupling depends on the scheduling policy, as explained below.

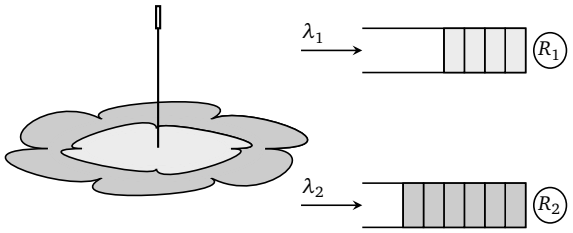


Figure 4.5: A simple model with two cell regions.

4.5.2 Round-robin scheme

Under the round-robin policy, users share the radio resources equally, independently to their radio conditions. Thus users in region k are allocated a fraction of radio resources in state $\boldsymbol{x}$ as

$$\varphi_k(\boldsymbol{x}) = \frac{x_k}{\sum_j x_j}. \tag{4.18}$$

Then the radio resource allocated to users in region k is calculated as

$$\phi_k(\boldsymbol{x}) = \varphi_k(\boldsymbol{x}) R_k. \tag{4.19}$$

At high load all users in all regions select $v_{\min} = v_I$ as their video bit rate under the round robin policy. Thus the stability condition follows from (4.10):

$$\rho = \sum_k \frac{p_k \lambda}{\frac{R_k}{v_{\min} T}} < 1 \;\rightarrow\; \lambda_{\max} = \Big(\sum_k \frac{p_k v_{\min} T}{R_k} \Big)^{-1}, \tag{4.20}$$

where $\lambda_{\max}$ is the maximum arrival rate that the system can handle. Observe that this corresponds to the less restrictive stability condition that can be obtained from (4.10).

4.5.3 Max C/I scheme

The max C/I policy is an extreme case of opportunistic scheduling strategies that prioritizes those users with the best radio conditions. For two regions for instance, cell-center users are scheduled first and are allocated all the resources whenever active; cell-edge users are served only when there are no active cell-center users.

Stability condition approximation: Under the max C/I policy, base station first allocate its resources to the cell-center flows. Then it allocates the rest of resources to users having lower physical throughput. The maximum traffic intensity happens when the rest of resource allocated to the cell-edge users can only support for the selection of $v_{\min}$, which is shown as

$$\lambda_{\max} = \Big(\sum_{k \neq K} \frac{p_i \bar{v}_k T}{R_k} + \frac{p_K v_{\min} T}{R_K} \Big)^{-1}, \tag{4.21}$$

where $\bar{v}_k$ is also a function of $\lambda_{\max}$. Therefore, $\lambda_{\max}$ of max C/I policy can be solved as a fixed-point solution of Eq.(4.21) and $\bar{v}_k$ is calculated as

$$\bar{v}_k = \sum_{n=0}^{\infty} \pi_k(n) v_k(n), \tag{4.22}$$

with

$$\pi_k(n) = \pi_i(0) \prod_{j=1}^{n} \frac{p_k \lambda_{\max} R'_k}{v_k(j) T}, \tag{4.23}$$

$$v_k(n) = \max\Big(\min\Big(\frac{R'_k}{n}, v_{\max} \Big), v_{\min} \Big), \tag{4.24}$$

where $R'_k = \prod_{j<k} \pi_j(0) R_k$ and $\pi_k(0)$ is denoted as the probability that there is no flows in class-k.

4.5.4 Max-min scheme

The max-min policy achieves fairness through users throughput equalization. Users in good radio conditions are allocated fewer resources while users in bad radio conditions get the largest share, so that all users get the same throughput:

$$\frac{\varphi_1(\boldsymbol{x})R_1}{x_1} = \frac{\varphi_2(\boldsymbol{x})R_2}{x_2} = \dots \frac{\varphi_K(\boldsymbol{x})R_K}{x_K}.$$

Thus users in region k are allocated a fraction

$$\varphi_k(\boldsymbol{x}) = \frac{x_k/R_k}{\sum_j x_j/R_j} \tag{4.25}$$

of radio resources in state $\boldsymbol{x}$. Similarly to the round robin policy, all users get $\nu_{\min}$ at high load and the stability condition is given by (4.20).

4.5.5 Opportunistic scheduling scheme

The capacity shares in equation (4.18) are computed supposing a round robin scheduling. However, we consider a channel aware scheduling. A proportional fair scheduler that operates at the fast fading time scale, as that of [90][64], is assumed. In this case, when there are $|\boldsymbol{x}|$ flows that are active in the LTE cell, the throughput of a user of radio condition R_k that gets a proportion φ_k of the cell resources is equal to $\varphi_k R_k G(|\boldsymbol{x}|)$, where $G(|\boldsymbol{x}|)$ is the opportunistic scheduling gain that depends on many parameters such as the channel model, the receiver and the Multiple Input Multiple Output (MIMO) scheme [42]. Note that, contrary to real time streaming that does not profit from the opportunistic scheduling gain due to its stringent delay constraints, http streaming with buffering profit from this type of scheduling like elastic traffic. The performance model proposed in 4.3 can thus be extended to the opportunistic scheduling case by introducing $G(|\boldsymbol{x}|)$ into the formulation of allocated resource in Eq. (4.1) and (4.8) and video bit rate selected in either discrete or continuous set. Note that, $G(|\boldsymbol{x}|) = 1$, when round-robin scheduling is applied.

4.5.6 KPIs definition for heterogeneous radio condition

Here we extend the KPIs definition with heterogeneous radio conditions and different scheduling schemes, where we only demonstrate the case configured by small chunk duration.

Video bit rate

Under the assumption that users watch all the downloaded video.The first performance metrics we measure is the mean video bit rate, which is the average video resolution that a user experiences while watching the video. Noting that the expectation is calculated based

on the stationary distribution $\pi(\boldsymbol{x})$, in the static case, we can compute the mean video bit rate of users in each region i as follows:

$$\bar{v}_k = \frac{E\left(\varphi_k(X)R_k\right)}{\lambda_k T}. \tag{4.26}$$

The cell mean video bit rate in both scenarios (without and with mobility) is given by:

$$\bar{v} = \frac{E\left(\sum_k \varphi_k(X)R_k\right)}{\sum_k \lambda_k T}. \tag{4.27}$$

Intuitively speaking, the mean video bit rate is obtained by dividing the average wireless resource allocated to different classes by the average number of arriving flows. In the case with mobility, only the cell mean video bit rate $\bar{v}$ can be calculated. However, in the absence of mobility it is observed that $\bar{v}$ is nothing then the arithmetic mean of $\bar{v}_i$ weighed by the probabilities p_k:

$$\bar{v} = \sum_k p_k \bar{v}_k. \tag{4.28}$$

Buffer surplus

As we have introduced in section 4.4, the distribution seen by users in region k is the size-biased distribution. In the following discussion, we denote by E_i the expectation using corresponding biased distribution:

$$\pi_k(\boldsymbol{x}) \propto x_k \pi(\boldsymbol{x}).$$

In addition to the mean video resolution, the quality of experience is also influenced by the video smoothness measured in terms of buffer surplus. By calculating the buffer surplus as Eq. 4.29, this performance metrics reflects the average relative buffer variation of each flow.

$$\begin{aligned} B_k &= E_k\left(\frac{R_k\varphi_k(X)/X_k - v_k(X)}{v_k(X)}\right) \\ &= \sum_{\boldsymbol{x}} \pi_k(\boldsymbol{x}) \frac{\frac{R_i\varphi_k}{x_k} - v_k(\boldsymbol{x})}{v_k(\boldsymbol{x})} = \sum_{\boldsymbol{x}} \pi(\boldsymbol{x}) \frac{\frac{R_k\varphi_k}{v_k(\boldsymbol{x})} - x_k}{\sum_{\boldsymbol{x}} x_k \pi(\boldsymbol{x})}. \end{aligned} \tag{4.29}$$

That is:

$$\bar{B}_k = E\left(\frac{R_k\varphi_k(X) - X_k v_k(X)}{v_k(X)}\right) / E\left(X_k\right).$$

Similarly, the mean buffer surplus over the cell is:

$$B = E\left(\sum_k \frac{R_i\varphi_k(X) - X_k v_k(X)}{v_k(X)}\right) / E\left(\sum_k X_k\right). \tag{4.30}$$

Note that

$$B = \sum_k p'_k B_k, \text{ with } p'_k = \frac{E(X_k)}{\sum_j E(X_j)}$$

where p'_k represents the probability that an active user is in region k. When load becomes large, both B and B_k values approach -1.

4.6 Integration of elastic services

Section 4.3 proposed performance metrics for adaptive streaming and showed how to compute them when only adaptive streaming flows share the capacity with both cases of small and large chunk duration. Here, we are going to generalize the model to heterogeneous radio conditions and discuss the performance of these flows in the presence of elastic traffic. Here we take round robin scheduling as the example. For other scheduling schemes we can simply change the shared resources in Eq. (4.33).

We also consider an extended setting where streaming flows share the cell capacity with elastic flows. Therefore, the system can be represented with two groups of coupled processor-sharing queues. Flows in queue e,k correspond to the elastic traffic with radio condition, R_k, and flows in queue s,k,i correspond to the streaming ones with R_k and selecting v_i. To remind, for small chunk duration, there is only one video bit rate i corresponds to a specific radio condition R_k at a given time, therefore, queue s,k,i can be simplified as s,k. However, we show the general case in the following.

With the proportion of elastic and streaming denoted as p_e, p_s and $p_e + p_s = 1$, the flow arrival rates at each queue are calculated as $\lambda^e_k = p_e p_k \lambda$ and

$$\begin{aligned} &\text{Small Chunk,} \quad \lambda^s_k = p_s p_k \lambda, \\ &\text{Large Chunk,} \quad \lambda^s_{ki}(\boldsymbol{x}) = \begin{cases} p_s p_k \lambda, & \text{as } v_k(\boldsymbol{x} + \boldsymbol{e}^s_{ki}) = v_i \ , \\ 0, & \text{otherwise,} \end{cases} \end{aligned} \tag{4.31}$$

where $\boldsymbol{x} = (x^e_1, \cdots, x^e_k, x^s_{11}, \cdots, x^s_{ki})$ represents the number of flows in each queue. Applying the same concept to formulate the departure rate as equation (4.1), the flow departure rate of elastic and adaptive streaming services can be respectively expressed as

$$\mu^e_k(\boldsymbol{x}) = \frac{\phi^e_k(\boldsymbol{x})}{\sigma}, \quad \mu^s_{ki}(\boldsymbol{x}) = \frac{\phi^s_{ki}(\boldsymbol{x})}{v_{ki}(\boldsymbol{x})T}, \tag{4.32}$$

where σ is the mean flow size of elastic data. Besides, $\phi^e_k(\boldsymbol{x})$ and $\phi^s_{ki}(\boldsymbol{x})$ stand for the allocated wireless resources for each class. These capacity shares can be computed as:

$$\phi^e_k(\boldsymbol{x}) = \frac{x^e_k R_k}{|\boldsymbol{x}|}, \quad \phi^s_{ki}(\boldsymbol{x}) = \frac{x^s_{ki} R_k}{|\boldsymbol{x}|}, \tag{4.33}$$

where $|\boldsymbol{x}| = \sum_k (x_k^e + \sum_i x_{ki}^s)$. Moreover, for small chunk case, the video bit rate, $v_{ki}(\boldsymbol{x}) = v_k(\boldsymbol{x})$, is calculated based on the mechanism (4.5) and the instantaneous throughput,

$$\gamma_{ki}^s(\boldsymbol{x}) = \frac{\phi_{ki}^s(\boldsymbol{x})}{x_{ki}^s} = \frac{R_k}{|\boldsymbol{x}|}. \tag{4.34}$$

Moreover, for large chunk case, $v_{ki}(\boldsymbol{x}) = v_i$. We can obtain departure rate values as Eq.(4.32). With the formulation above, note that the balance property introduced in [35] is not valid. The Markov chain is not reversible and we have to solve the balance equations (by a matrix inversion) for the stationary distribution $\pi(\boldsymbol{x})$:

$$\begin{aligned} &\forall \boldsymbol{x}, \Big(\sum_{j=e,s} \sum_k \sum_i \lambda_{ki}^j + \mu_{ki}^j(\boldsymbol{x}) \Big) \pi(\boldsymbol{x}) \\ &= \sum_{j=e,s} \sum_k \sum_i \Big(\lambda_{ki}^j(\boldsymbol{x}) \pi(\boldsymbol{x} - \boldsymbol{e}_{ki}^j) + \mu_{ki}^j(\boldsymbol{x} + \boldsymbol{e}_{ki}^j) \pi(\boldsymbol{x} + \boldsymbol{e}_{ki}^j) \Big). \end{aligned} \tag{4.35}$$

4.6.1 Stability condition

We begin by assessing the stability region of the system. Applying the same implication of section 4.3.3, when system approaches the stability limit, the video bit rate of streaming users decreases to v_I. Both streaming flows with small chunk and large chunk behave like elastic traffic. Stability holds only when the sum of offered loads for both services is less than 1, where we obtain the max system traffic intensity, $\lambda_{\max}$, as

$$\begin{aligned} &\sum_k \Big(\frac{\lambda p_e p_k \sigma}{R_k} + \frac{\lambda p_s p_k v_I T}{R_k} \Big) \leq 1 \\ &\Rightarrow \lambda_{\max} = \Big(\sum_k \frac{R_k}{p_e p_k \sigma + p_s p_k v_I T} \Big)^{-1}. \end{aligned} \tag{4.36}$$

4.6.2 KPIs definition for integrating elastic traffic

We extend the adaptive streaming related KPIs to general case and define a KPI for the performance of elastic traffic.

•Video bit rate

We define the overall video bit rate for the general model, $\bar{v}$, by summing up all the class and weighting by the flow number of each class at state $\boldsymbol{x}$ as

$$\bar{v} = \frac{\sum_{\boldsymbol{x}:|\boldsymbol{x}^s|>0} \pi(\boldsymbol{x}) \sum_k \sum_i \phi_{ki}(\boldsymbol{x})}{\lambda p_s T} \tag{4.37}$$

and the average video bit rate of class-k with R_k is denoted as

$$\bar{v}_k = \frac{\sum_{\boldsymbol{x}:x_{ki}^s>0} \pi(\boldsymbol{x}) \sum_i \phi_{ki}(\boldsymbol{x})}{\lambda p_s p_k T} \tag{4.38}$$

where $|\boldsymbol{x}^s| = \sum_k \sum_i x_{ki}^s$.

•Deficit rate

The overall deficit rate of multiple class model, D and the deficit rate of each class with R_k, D_k are defined as

$$D = \sum_{\boldsymbol{x}:|\boldsymbol{x}^s|>0} \frac{\pi(\boldsymbol{x})}{\bar{\boldsymbol{x}}^s} \sum_k \sum_i x_{ki}^s \mathbf{1}_{\{\gamma_{ki}^s(\boldsymbol{x})<\nu_{ki}(\boldsymbol{x})\}}, \tag{4.39}$$

$$D_k = \sum_{\boldsymbol{x}:x_{ki}^s>0} \frac{\pi(\boldsymbol{x})}{\bar{x}_{ki}^s} \sum_i x_{ki}^s \mathbf{1}_{\{\gamma_{ki}^s(\boldsymbol{x})<\nu_{ki}(\boldsymbol{x})\}}, \tag{4.40}$$

where $\mathbf{1}$ is the indicator function equal to 1 when the condition is satisfied, otherwise the indicator function will become 0. In addition, $\bar{\boldsymbol{x}}^s = \sum_{\boldsymbol{x}} \pi(\boldsymbol{x})|\boldsymbol{x}^s|$, $\bar{x}_k^s = \sum_{\boldsymbol{x}} \pi(\boldsymbol{x}) \sum_i x_{ki}^s$.

•Buffer surplus

We then define the overall buffer surplus of multiple class model, B and the buffer surplus of each class with R_k, B_k as

$$B = \sum_{\boldsymbol{x}:|\boldsymbol{x}^s|>0} \frac{\pi(\boldsymbol{x})}{\bar{\boldsymbol{x}}^s} \sum_k \sum_i x_{ki}^s \Big(\frac{\gamma_{ki}^s(\boldsymbol{x}) - \nu_{ki}(\boldsymbol{x})}{\nu_{ki}(\boldsymbol{x})} \Big), \tag{4.41}$$

$$B_k = \sum_{\boldsymbol{x}:x_{ki}^s>0} \frac{\pi(\boldsymbol{x})}{\bar{x}_k^s} \sum_i x_{ki}^s \Big(\frac{\gamma_{ki}^s(\boldsymbol{x}) - \nu_{ki}(\boldsymbol{x})}{\nu_{ki}(\boldsymbol{x})} \Big). \tag{4.42}$$

•Service time

Same as the previous section, the overall mean service time and mean service time of class-k are denoted as

$$S = \frac{\bar{\boldsymbol{x}}^s}{\lambda}, \quad S_k = \frac{\bar{x}_k^s}{\lambda_k}. \tag{4.43}$$

•Average elastic throughput

The average elastic throughput is chosen as the performance metric for elastic flows, with $\bar{\gamma}^e$ and $\bar{\gamma}_k^e$ standing for the overall metric and mean throughput for each class with R_k,

$$\bar{\gamma}^e = \sum_{\boldsymbol{x}:|\boldsymbol{x}^e|>0} \frac{\pi(\boldsymbol{x})}{\bar{\boldsymbol{x}}^e} \sum_k x_k^e \frac{\phi_k^e(\boldsymbol{x})}{x_k^e}, \tag{4.44}$$

$$\bar{\gamma}_k^e = \sum_{\boldsymbol{x}:x_k^e>0} \frac{\pi(\boldsymbol{x})}{\bar{\boldsymbol{x}}^e} \frac{x_k^e \phi_k^e(\boldsymbol{x})}{x_k^e}, \tag{4.45}$$

where $\bar{\boldsymbol{x}}^e = \sum_{\boldsymbol{x}} \pi(\boldsymbol{x})|\boldsymbol{x}^e|$, $\bar{x}_k^e = \sum_{\boldsymbol{x}} x_k^e \pi(\boldsymbol{x})$ and $|\boldsymbol{x}^e| = \sum_k x_k^e$.

4.7 Mobility model

As Fig. 4.6 shows, we assume that each user in region k moves to region $k-1$ (for $k > 1$) and to region $k+1$ (for $k < N$) after exponential durations, at respective rates $\nu_{k,k+1}$ and $\nu_{k,k-1}$. The probability that a user is in region i then satisfies:

$$q_k \propto \prod_{j=1}^{k-1} \frac{\nu_{j,j+1}}{\nu_{j+1,j}}.$$

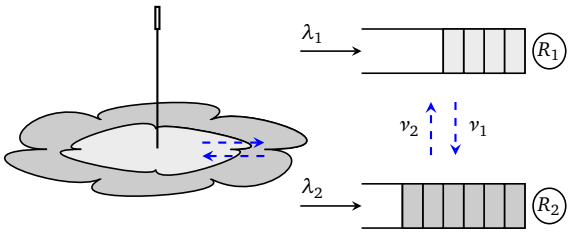


Figure 4.6: Queuing model for two regions with intra-cell mobility.

Note that this is not the probability that an active user is in region k, which is given by p'_k. In addition, based on the mobility rates, the stationary distribution, $\pi(\boldsymbol{x})$, of the Markov process is the solution of the following balance equations:

$$\sum_k (\lambda_k + \mu_k(\boldsymbol{x}) + \nu_{k,k+1} + \nu_{k,k-1})\pi(\boldsymbol{x}) = \sum_k \Big(\lambda_k \pi(\boldsymbol{x} - \boldsymbol{e}_k) + \mu_k(\boldsymbol{x} + \boldsymbol{e}_k)\pi(\boldsymbol{x} + \boldsymbol{e}_k)\Big)$$
$$+ \sum_k \Big(x_{k+1}\nu_{k+1,k}\pi(\boldsymbol{x} - \boldsymbol{e}_k + \boldsymbol{e}_{k+1}) + x_{k-1}\nu_{k-1,k}\pi(\boldsymbol{x} - \boldsymbol{e}_k + \boldsymbol{e}_{k-1})\Big).$$

4.7.1 Stability condition

In this section, we show how to calculate the $\lambda_{\max}$ obtained at the stability condition, which follows from the limiting regime of infinite mobility where $\nu_{k,k+1}, \nu_{k+1,k} \to \infty$ and is given by:

$$\lambda/\bar{\mu} < 1,$$

where $\bar{\mu}$ is the mean service rate at high load. We shall see that this service rate depends only on the scheduling strategy and is independent from the mobility rate in the presence of mobility. The $\lambda_{\max}$ is defined as

$$\lambda_{\max} = \bar{u}.$$

round-robin policy

The mean service rate at high load under the round robin strategy is given by:

$$\bar{\mu} = \sum_k q_k \frac{R_k}{\nu_{\min} T}.$$

max C/I policy

Under the max C/I policy all mobile users are served in the first region (that of the best physical rate R_1) at high load. It follows that:

$$\bar{\mu} = \frac{R_1}{\nu_{\min} T}.$$

max-min policy

Under the max-min policy, the mean service rate at high load follows from (4.25):

$$\bar{\mu} = \frac{1}{\nu_{\min} T} \sum_k \frac{q_k}{\sum_j q_j / R_j}.$$

Observe that in the presence of mobility the less restrictive stability condition is obtained under the max C/I policy. However in the absence of mobility this strategy engender the most restrictive stability condition compared to more fair allocation strategies (round robin, max-min).

4.7.2 Performance in light traffic

The performance in light traffic (that is, when $\rho \to 0$) is independent of the scheduling policy. Indeed, a user when alone in the system is always allocated all radio resources. As Eq. (4.5), the video bit rate in region k can be extended to a continuous set as $c_k = \max(\min(R_k, \nu_{\max}), \nu_{\min})$, that is $\nu_{\max}$ in all regions k where $R_k > \nu_{\max}$. The mean buffer surplus in region k is then:

$$b_k = \frac{R_k - c_k}{c_k}.$$

Thus the mean buffer surplus in the cell in light traffic is given by:

$$B = \sum_k p'_k b_k.$$

When $\nu_k \to 0$, $\forall k \leq K$ (no mobility) the mean buffer surplus in the cell in light traffic can be written as:

$$B = \sum_k \frac{p_k/\mu_k}{\sum_j p_j/\mu_j} b_k, \text{ where } \mu_k = \frac{R_k}{c_k T}.$$

However, in the limiting regime of infinite mobility $\nu_k \to \infty$ ($\forall k \leq K$) the cell mean buffer surplus is given by:

$$B = \sum_k q_k b_k.$$

For two regions for instance, explicit expressions of the cell buffer surplus in light traffic as a function of the mobility rates can be written as:

$$B = \frac{(\nu_2 + p_1\mu_2)b_1 + (\nu_1 + p_2\mu_1)b_2}{\nu_1 + \nu_2 + p_1\mu_2 + p_2\mu_1}.$$

4.8 Summary

In this chapter, we present the analytical model for HTTP adaptive streaming considering the impacts of flow-level dynamics. The model takes into account the system parameters like video chunk duration, video bit rate configuration, scheduling schemes and users' mobility. Based on the system stationary distribution, we also define the KPIs or the QoS that we are going to observe in two different senses, video resolution and playback smoothness. Because the complexity of solving balanced equations is too large, there is no exact form for the KPIs that we want to observe. We can only obtain these values by numerical analysis and launching simulation. The performance impacts of these system parameters and the trade-offs of KPIs will be presented and discussed in the next chapter.

Chapter 5

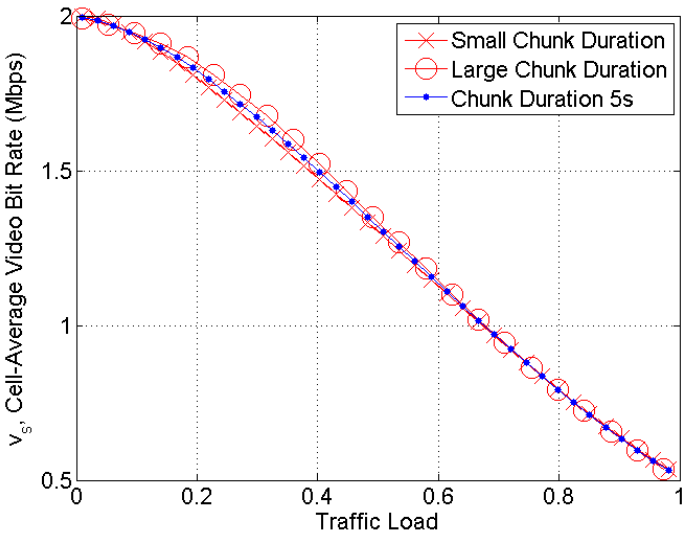
Simulation of Adaptive Streaming and Approximation Model

After having presented the model for HTTP-based adaptive streaming in the previous chapter, in this chapter, we begin by showing impacts of different system configurations in the video delivery system or wireless access networks including video chunk duration, number of available video bit rates and scheduling schemes. We also demonstrate the performance impact of users' mobility. Then we present an approximation model which can reduce the complexity of calculation. We apply the approximation model for illustrating how to use our model for network dimensioning and studying the impacts of the video bit rate limitation on streaming performance.

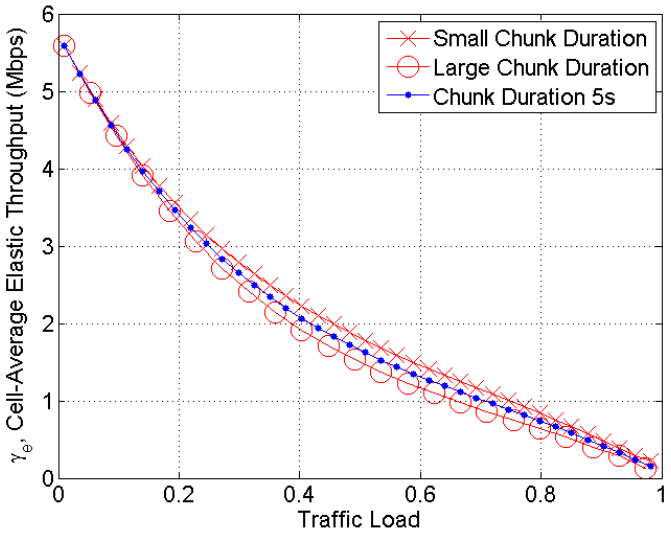
5.1 Impacts of chunk duration

According to the technical report of Akamai [1] and the empirical demonstration of [101], it is shown that deploying shorter chunk duration provides better video smoothness and more chances for users to select the proper video bit rate. However, deploying shorter chunks will also generate a large number of video chunks and its corresponding HTTP signaling. Therefore, the report suggests that video service providers choose their chunk duration configurations of HTTP adaptive streaming as 10 seconds. In this section, we first perform simulations for checking these findings by setting the parameters as $\mathscr{R} = \{R_C, R_E\} = \{10, 4\}$Mbps, $(p_C, p_E) = (\frac{1}{2}, \frac{1}{2})$ for cell center and cell edge, $(p_e, p_s) = (\frac{1}{2}, \frac{1}{2})$ for the elastic and streaming flows, $\mathscr{V} = \{v_1, v_2\} = \{2, 0.5\}$Mbps, $T = 10$s, $\sigma = 5$Mbits and $\lambda_{\max} = 1.14$ users/s. The simulation results in figs. 5.1 and 5.2 show the performance of all defined metrics with respect to the normalized settings of traffic load, $(\lambda/\lambda_{\max})$.

As section 4.2.1 mentioned, the simulation results shown in figs. 5.1 and 5.2 give us two performance bounds of the two extreme cases, $h = 0$ and $h = \infty$ which enable us to predict the performance of intermediate chunk duration. In the simulation, we further demonstrates the performance of an intermediate configuration, $h = 5$s between two extreme cases. In Fig. 5.1a, we find that configuring small chunk has smaller mean video bit rate compared to



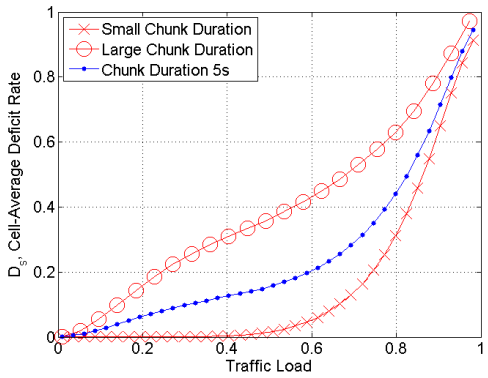
(a) Mean video bit rate v_s.



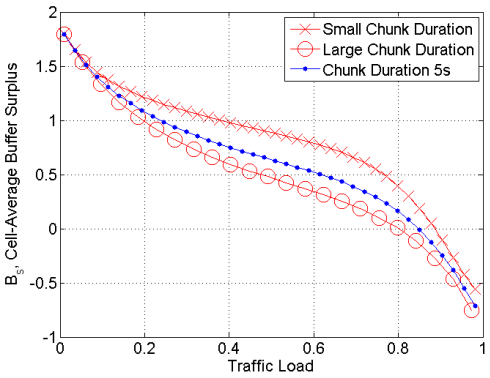
(b) Mean elastic throughput γ_e.

Figure 5.1: Performance of video resolution and elastic traffic obtained by both models with small and large chunk duration.

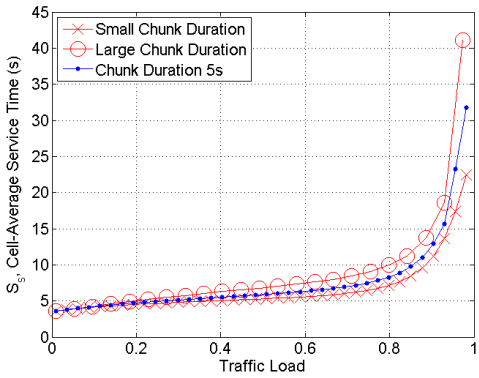
the case of large chunk duration. Fig. 5.1b tells us that deploying small chunk duration can be beneficial for mean elastic throughput. In terms of buffer performance, we observe that deploying small chunk duration will result in better deficit rate in Fig. 5.2a and that deploying small chunk duration will result in a better buffer surplus in Fig. 5.2b. We also get the similar result in terms of mean service time in Fig. 5.2c. These results show that configuring



(a) Deficit rate D_s.



(b) Buffer surplus B_s.



(c) Mean service time S_s.

Figure 5.2: Performance of video smoothness obtained by both models with two of chunk durations.

large chunk duration may increase the users' satisfaction in the sense of resolution but not in the sense of video smoothness. We proposes a method to reduce HTTP signaling while keeping the same performance in the following section.

5.2 Chunk duration design

Based on the analysis of previous section, a shorter chunk duration will slightly deteriorate the mean video bit rate but improve the mean service time, buffer surplus and deficit rate without taking HTTP signaling into consideration. Therefore, in this section, we propose and investigate a new mechanism allowing users to request multiple video chunks in an HTTP requests which can reduce the HTTP request signaling. We show that if we design the number of video chunk transmitted in an HTTP request following the rule to have same data volume in a single HTTP request, we can offer a chance for operators to achieve the system performance as the same aspect of video smoothness but with less number of HTTP requests.

5.2.1 One chunk per HTTP request

Assuming that the video chunk duration is configured as h for all video bit rates, $v_i \in \mathcal{V}$ and that a general video has duration T, users are going to download their video with a total number of video chunks, $N_h^{(1)}$, calculated as

$$N_h^{(1)} = \left\lfloor \frac{T}{h} \right\rfloor, \tag{5.1}$$

where function $\lfloor x \rfloor$ gives a largest integer value less than x. In this case, all the video bit rates adopt the same value of chunk duration, which has the same configuration as what we have discussed in the previous sections.

5.2.2 Multiple chunks per HTTP request following same size of requests

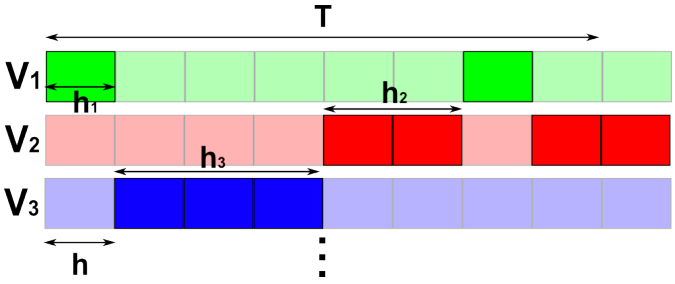


Figure 5.3: An example showing the mechanism of multiple chunks downloaded in an HTTP request.

Here, instead of choosing only one video chunk duration, h, for all the video bit rates, we assume that for different video bit rates, v_i, a specific number of chunks a_i, where $a_i \in \mathcal{N}$ and $a_i \geq 1$, will be downloaded in an HTTP request. Therefore, it seems that for video bit rate, v_i, we have a longer chunk duration $h_i = a_i h$ as shown in Fig. 5.3. Assuming that each video bit rate is selected equally, we have the number of HTTP requests calculated as

$$N_h^{(2)} = \sum_{v_i \in \mathcal{V}} \left\lfloor \frac{T}{h_i I} \right\rfloor \leq \left\lfloor \frac{T}{h} \right\rfloor \sum_{v_i \in \mathcal{V}} \frac{1}{a_i I} \leq \left\lfloor \frac{T}{h} \right\rfloor = N_h^{(1)}. \tag{5.2}$$

With the fact that $a_i \geq 1$, we can show that $N_h^{(2)} \leq N_h^{(1)}$. In order to facilitate the design of $\{a_i\}$, we propose to select the set, $\{a_i\}$, as follows:

$$a_i = \left\lfloor \frac{v_1}{v_i} \right\rfloor. \tag{5.3}$$

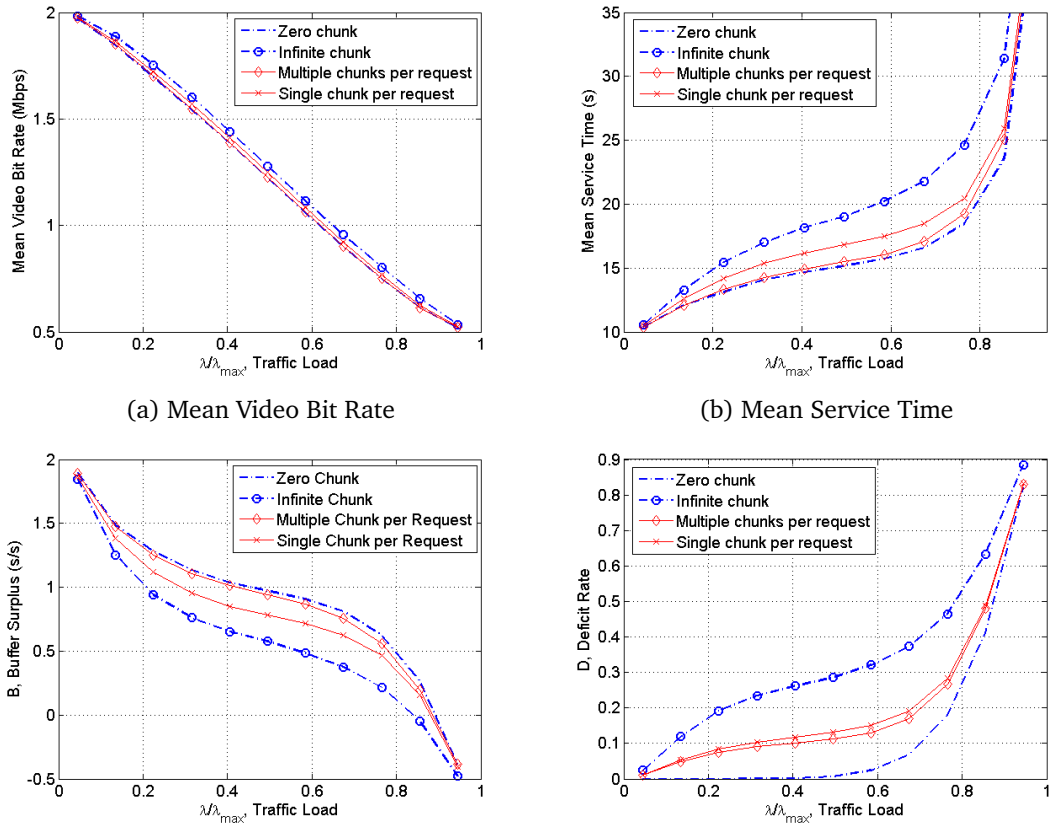
This means that for a low video bit rate, more video chunks will be transmitted in an HTTP request. An example is given as Fig. 5.3.

Several solution sets of chunk durations, $\{a_i\}$, are feasible. In this thesis, we only examine the proposed design guideline figuring out the number of video chunks requested in a HTTP

request for a certain video bit rate. There might be other ways to design $\{a_i\}$, for example to design chunk duration proportional to their video bit rate. These alternatives are not studied in this thesis but can be part of the future works.

5.2.3 Performance comparison

In order to analyze the performance of proposed policies, with the system configuration where $\mathscr{R} = \{R_1, R_2\} = \{10, 5\}$Mbps, $(p_1, p_2) = (0.5, 0.5)$, $T = 30$s, $\mathscr{V} = \{v_1, v_2\} = \{2, 0.5\}$Mbps, we compare two different ways to download video chunks, one chunk per HTTP request and multiple chunks per HTTP request. We simulate the HTTP adaptive streaming with the mentioned system configuration but with different numbers of chunks requested for two different video bit rates shown in Table 5.1. For the proposed policy, we serve the smaller video bit rate v_2 with a longer chunk duration, a_2, compared to the first policy.



(a) Mean Video Bit Rate

(b) Mean Service Time

(c) Mean Buffer Surplus

(d) Mean Deficit Rate

Figure 5.4: Peformance of transmitting multiple video chunks in a HTTP request.

In Fig. 5.4, we demonstrate the performance of two policies on the four overall performance metrics we have introduced in section 4.4. The simulation results show that compared

	a_1	a_2	h_1	h_2
One chunk per request	1	1	10s	10s
Multiple chunks per request	1	$\lfloor \frac{2}{0.5} \rfloor = 4$	10s	40s

Table 5.1: Chunk Configuration for Two Policies with $h = 10$s

with the first policy, our proposed policy offers slightly worse performance for the mean video bit rate but better performance for the rest of metrics. The degradation of mean video bit rate is not very significant. However, the improvements of other smoothness-related metrics are large. By comparing the simulation results between Fig. 5.1, 5.2 and Fig. 5.4, it can also be concluded that our proposed policy has similar effects as one chunk per HTTP request policy with shorter chunk duration, e.g. $h = 9$s in this case, which means that the proposed policy needs less video chunks stored at servers while provides the same performance. In addition, adaptive streaming systems that deploy multiple chunk per request policy will also generate less HTTP requests compared to the one chunk per HTTP request policy with the same video chunk duration.

5.3 Impacts of the number of video bit rates

In the previous simulation, video bit rate set is configured as $\mathcal{V} = \{v_1, v_I\}$. However, a general question is what are the impacts if more video bit rates are available to be chosen in $\mathcal{V}$?

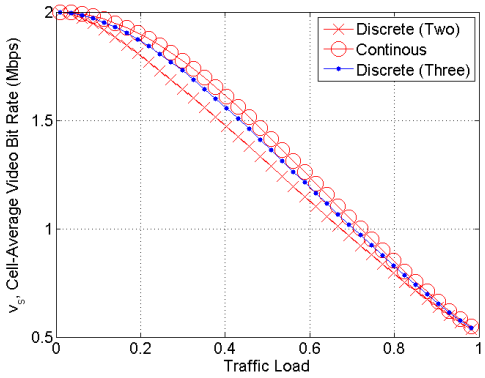
In order to answer, we configure a continuous video bit rate set $\mathcal{V}_{\text{cont}} = \{v : v_1 \leq v \leq v_I\}$ with configuring small video chunk duration. In this case, $|\mathcal{V}_{\text{cont}}| = \infty$. Different from Eq. (4.5), users' flows at capacity region k are assumed to be able to select any video bit rate between v_1 and v_I, expressed as

$$v_k(\boldsymbol{x}) = \max\Big(\min\Big(\gamma_k(\boldsymbol{x}), v_1\Big), v_I\Big), \tag{5.4}$$

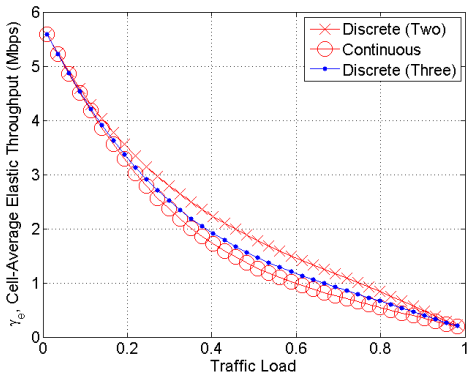
where $\gamma_k(\boldsymbol{x})$ can be found in Eq. (4.34) and based on Eq. (4.32), the flow departure rate becomes

$$\mu_k(\boldsymbol{x}) = \frac{\phi_k(\boldsymbol{x})}{v_k(\boldsymbol{x})T} = \frac{R}{\max\Big(\min\Big(\gamma_k(x), v_1\Big), v_I\Big)T}. \tag{5.5}$$

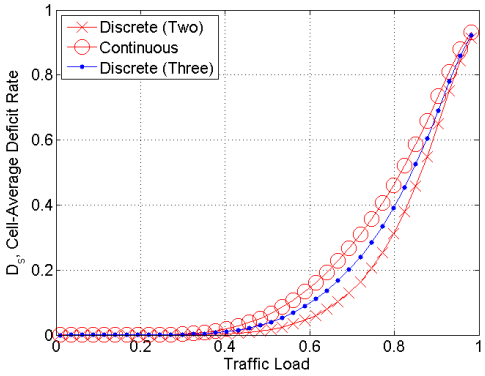
When the available throughput $\gamma_k(\boldsymbol{x}) = \frac{R_k}{|\boldsymbol{x}|}$ is larger than v_1, the buffer starts filling as the download capacity is larger than the playout rate and the flow behaves thus as an elastic one as there is no limitation on the buffer size. On the other hand, when $\gamma_k(\boldsymbol{x})$ is smaller than v_I, the behavior is also elastic but with possible starvation if the buffer empties due to the constant playout rate. Between v_1 and v_I, the flow behaves as a real time one and the buffer size remains constant. Solving the balance equations as Eq. (4.35), we obtain the



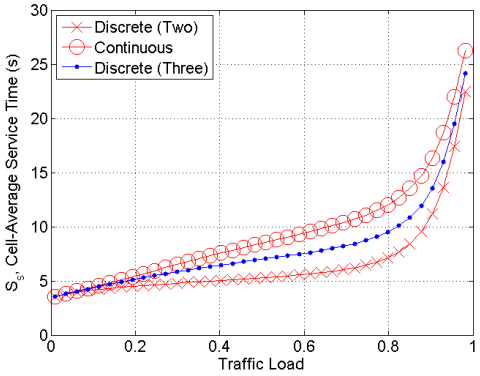
(a) Cell-average video bit rate v_s.



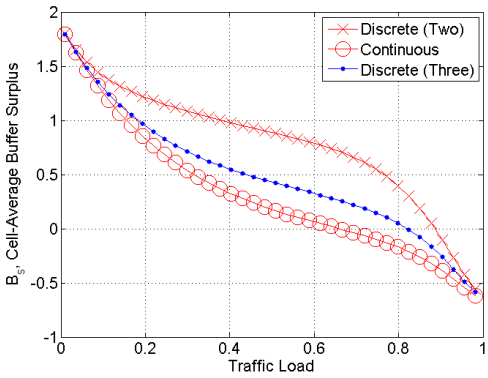
(b) Cell-average elastic throughput γ_e.



(c) Cell-average deficit rate D_s.



(d) Cell-average service time S_s.



(e) Cell-average buffer surplus B_s.

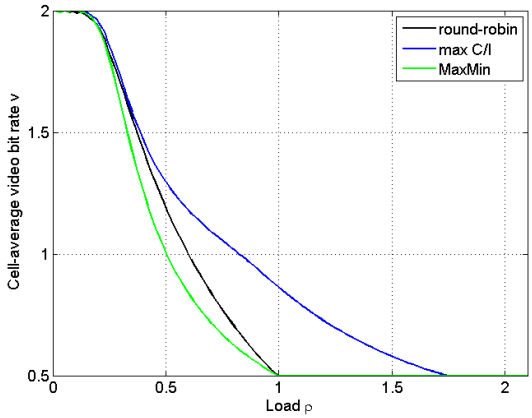
Figure 5.5: Performance of adaptive streaming with continuous and discrete video bit rate set.

simulation results shown in Fig. 5.5 with three different configurations including 2 video bit rates, infinite number of video bit rates and an intermediate setting, 3 video bit rates. It can

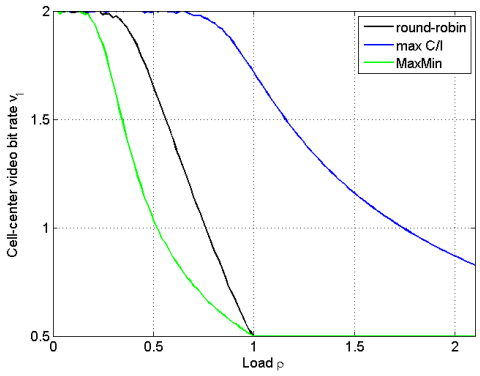
be observed that performance metrics of 3 video bit rates are located between the ones of the two extreme cases.

From Fig. 5.5a, we observe that deploying continuous video bit rate, meaning more video bit rates to approach users' real instantaneous throughput, will increase the long-term video resolution but decrease mean elastic throughput. In addition, from the rest of figures, we discover that deploying more video bit rates will also increase the buffer starvation from the three buffer-related metrics. Therefore, deploying larger number of video bit rates will generate a trade-off which can benefit the mean video resolution but decrease the elastic throughput and video smoothness.

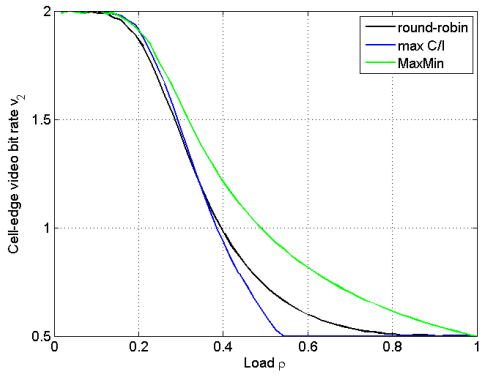
5.4 Impacts of scheduling schemes



(a) Cell-average video bit rate.



(b) Cell-center video bit rate.

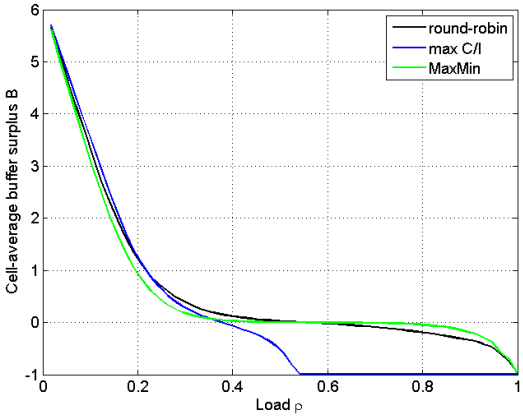


(c) Cell-edge video bit rate.

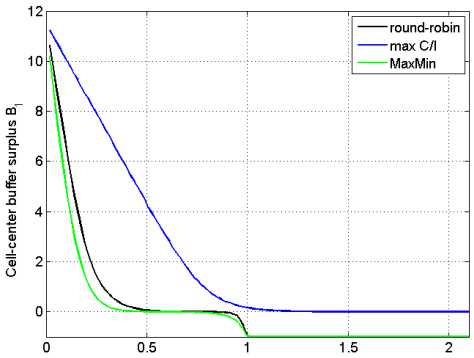
Figure 5.6: Video bit rate performance under different scheduling schemes.

In order to examine the performance impacts of scheduling schemes, with the considera-

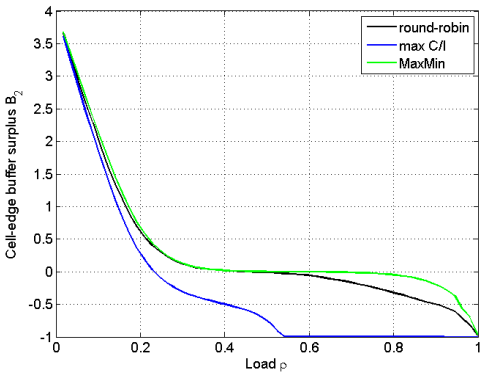
tion of two capacity regions standing for cell-center and cell-edge respectively, the simulation results are shown in Fig. 5.6 and Fig. 5.7 which compare the performance of video bit rate and the buffer surplus with different policies for two regions, where the configurations are set up as $p_1 = p_2 = 1/2$, $R_1 = 25$ Mbps, $R_2 = 10$ Mbps, $v_{\min} = 0.5$ Mbps, $v_{\max} = 2$ Mbps and $T = 10$s. These results are obtained by the numerical evaluation of the stationary distribution of the Markov process $X(t)$.



(a) Cell-average buffer surplus



(b) Cell-center buffer surplus



(c) Cell-edge buffer surplus.

Figure 5.7: Buffer surplus performance under different scheduling schemes.

Fig. 5.6 shows that under the max C/I policy, we can obtain better cell-average video bit rate compared to the other policies. Observe that the cell-average video bit rate is simply $\bar{v} = p_1\bar{v}_1 + p_2\bar{v}_2$ in accordance with (4.28). However, in terms of stability condition, we observe that max C/I policy provides lower stability condition, $\lambda_{\max}$, than other two policies. Based on Eq. (4.20) and (4.21), we obtain $\lambda_{\max,\mathrm{RR}} = 2.85$, and $\lambda_{\max,\max\ \mathrm{C/I}} = 0.54 \times \lambda_{\max,\mathrm{RR}}$. It is shown that the analytic calculation give the same results as we obtain in simulation, where $\lambda_{\max,\mathrm{RR}}$ and $\lambda_{\max,\max\ \mathrm{C/I}}$ are the minimum flow arrival rate that makes $\bar{v}_c$ or $\bar{v}_e$ equal

to $\nu_{\min}$ for respective scheduling policy.

Moreover, in Fig. 5.7, bad buffer surplus shows that the starvation event will strongly happen for the cell-edge users under the max C/I policy. In [11], it is concluded that there is no difference to deploy either round-robin or max C/I policy in the absence of mobility. However, for adaptive streaming services, it is showed that a trade-off exists between the two performance metrics and max C/I policy degrades system stability conditions. With little improvement of mean video bit rate and large degradation of system stability, operators are suggested to deploy round-robin policy for adaptive streaming in the static case.

5.5 Impacts of intra-cell mobility

In this section, we discuss the impacts of intra-cell mobility introduced in Fig. 4.6. We suppose that there is only two regions and users move from the center of the cell to the edge and vice versa. We still assume that there is no fast fading. Considering two capacity regions stand for cell-center and cell-edge with the same traffic and system configurations as before, we suppose that mobility rates are symmetric, that is $\nu_1 = \nu_2 = \nu$. We consider the static case $\nu = 0$ and a case with a mobility rate $\nu = 1$. The results shown in Figure 5.8 and 5.9 are obtained by the numerical evaluation of the stationary distribution $\pi(\boldsymbol{x})$ of the Markov process $X(t)$.

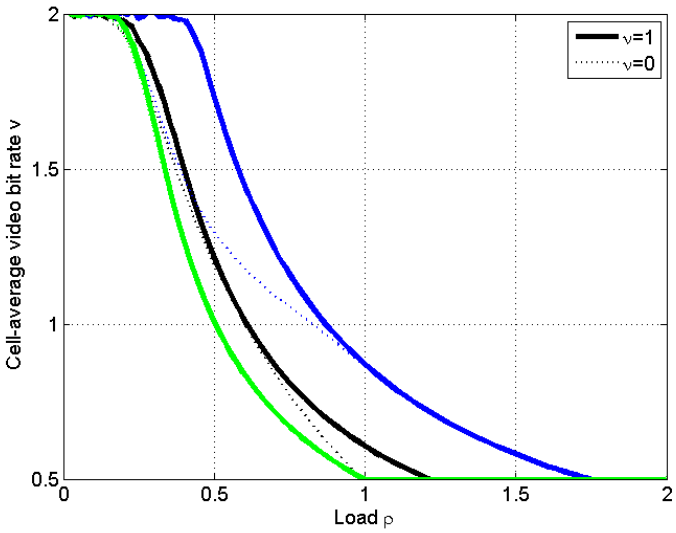
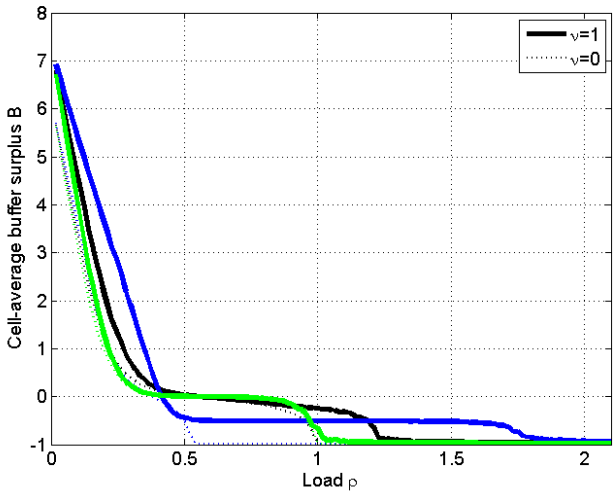
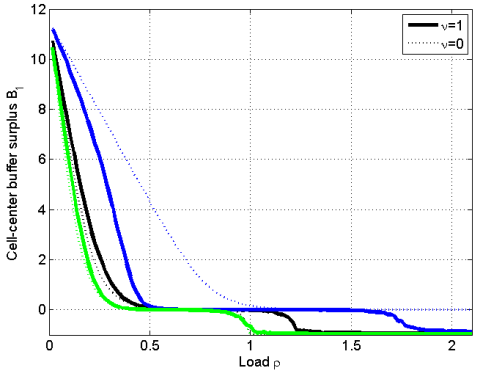


Figure 5.8: Mean video bit rate with and w/o mobility, under round-robin(black), max C/I(blue) and maxmin(green).

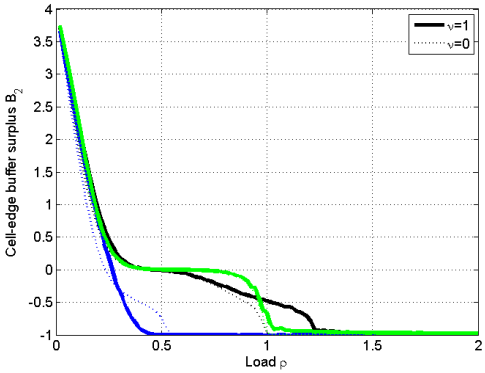
For the video bit rate, we can only obtain the average all over the cell, because when users' mobility is considered, from the flow-level model, no flows belong to only one class. In addition, it can be shown that max C/I policy provides better mean video bit rate and higher system stability. On the other hand, for the mean buffer surplus, max C/I policy



(a) Cell-average buffer surplus



(b) Cell-center buffer surplus



(c) Cell-edge buffer surplus.

Figure 5.9: Buffer surplus with and w/o mobility, under round-robin(black), max C/I(blue) and maxmin(green).

performs better at low load and also at the high load and round-robin performs better at medium load. Generally speaking, mobility provides more opportunities for users to exploit diversity gain. However, high video bit rate may degrade the buffer surplus.

5.6 Discriminatory scheduling scheme

In this section, we summarize the suggested scheduling policy for different services in Table. 5.2. In the static case, all scheduling policies have the same performance for elastic traffic. However, for adaptive streaming round-robin policy is suggested as the max C/I degrades the stability condition. In the presence of mobility, max C/I is recommended for the

elastic data. However, for the adaptive streaming suggested policy depends on the desired optimized performance metric.

Services / Scenario	Elastic	Adaptive Streaming
Static	Same for all policies	RR: Better $\bar{B}$, stability condition max C/I: Minor improve $\bar{v}$ RR is suggested
Mobile	max C/I	RR: Better $\bar{B}$ at medium load max C/I: Better $\bar{v}$, stability condition max C/I: Better $\bar{B}$ at low and high load Depend on the needs

Table 5.2: Policies recommended for different cases and services.

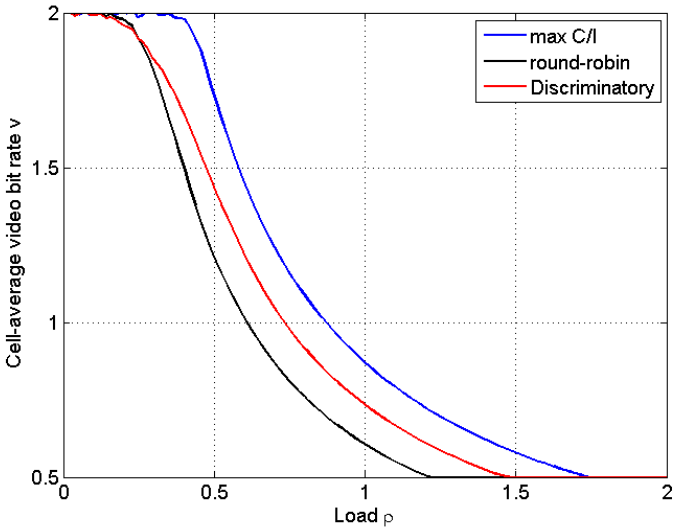


Figure 5.10: Mean video bit rate with mobility under round-robin (blue), max C/I (black) and discriminatory policy(green).

In the case of mobile users, there is a trade-off to deploy either round-robin or max C/I policy. Therefore, we examine the performance of discriminatory scheduler [16] to provide some intermediate results between the two scheduling policies. The resource allocation of users in region k under the discriminatory scheduler is calculated as follows:

$$\phi_k(\boldsymbol{x}) = \frac{w_k x_k}{\sum_j w_j x_j} R_k, \tag{5.6}$$

where w_k is the weight value of users in region k.

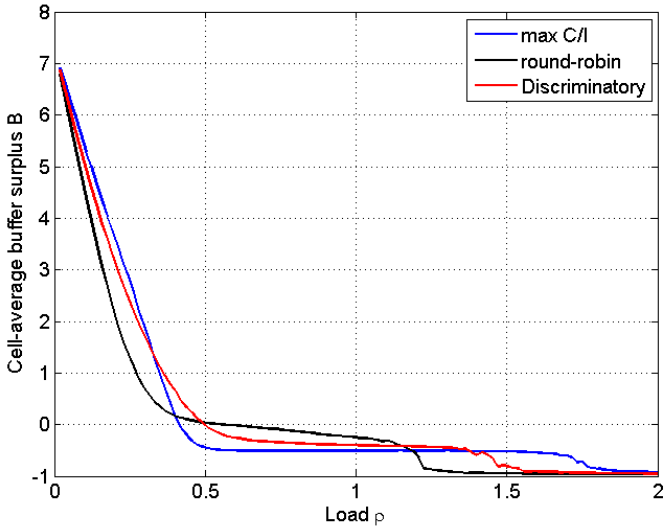


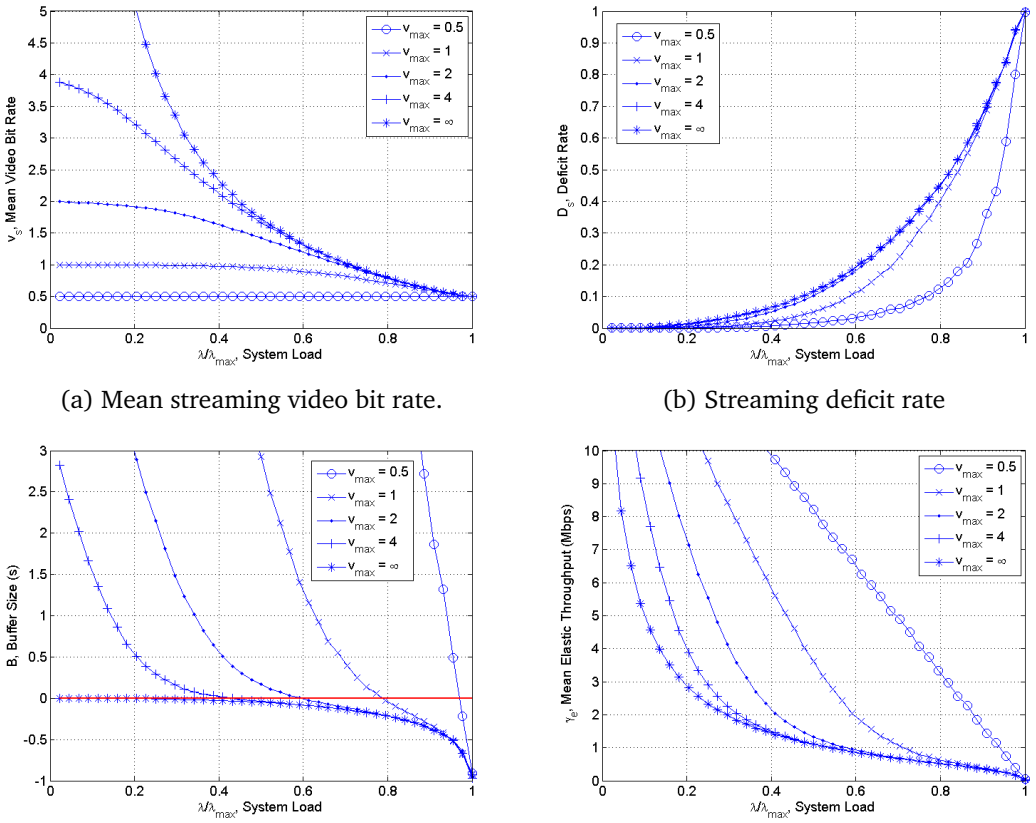
Figure 5.11: Mean buffer surplus with mobility under round-robin(blue), max C/I(black) and discriminatory policy(green).

Fig. 5.10 and 5.11 shows the simulation results using the same system configuration as previous where users are all mobile. Moreover weighting value are configured as $(w_1, w_2) = (1000, 1)$. Simulation results give an intermediate performance between the round-robin, where $(w_1, w_2) = (1, 1)$ and the max C/I policy, where $(w_1, w_2) = (\infty, 1)$.

5.7 Impacts of largest video bit rate

With the configuration of continuous video bit rate shown in Eq. (5.4), in this section, we demonstrate the impacts of maximum video bit rate, $v_I = v_{\max}$ on the four performance metrics.

Under the same system configuration as previous section but continuous video bit rate set, four performance metrics are shown with different $v_{\max}$ configurations in Fig. 5.12. It can be observed that when $v_{\max}$ increase, $\bar{v}_s$ and D_s increase, but B_s and $\bar{\gamma}_e$ decreases, meaning that in the case of $v_{\min} = 0.5$Mbps, decreasing $v_{\max}$ can benefit $\bar{v}_s$, D_s and B_s regardless of the trade-offs of $\bar{\gamma}_e$ reduction. We can also observe that the deficit rate is not highly influenced when $v_{\max}$ is large. Therefore, we believe that compared with the deficit rate buffer surplus has more information than deficit rate which better represents the starvation performance of streaming users.



(a) Mean streaming video bit rate.

(b) Streaming deficit rate

(c) Streaming buffer surplus

(d) Mean elastic throughput.

Figure 5.12: Performance with different $v_{\max}$.

5.8 Approximation model

Section 4.3 introduces the general model considering the heterogeneous radio conditions with small and large chunk duration. However, the complexity of solving the balance equations mentioned in Eq. (4.35) increases polymonially with the number of classes, making the computation of the stationary distribution very difficult. In this section, we propose an approximation model which can reduce the number of class while keeping the access to the defined KPIs. Here the approximation model is proposed for adaptive streaming with consideration of elastic traffic.

5.8.1 Approximation model for significantly small chunk duration

Here, we approximate the multiple classes of users to the processor-sharing model carrying only one equivalent class of elastic and adaptive streaming, where the system state is

denoted as $\boldsymbol{x} = (x^e, x^s)$. The equivalent flow arrivals of these two queues are expressed as:

$$\hat{\lambda}_e(\boldsymbol{x}) = p_e \lambda \sum_k \frac{p_k}{\eta_k}, \quad \hat{\lambda}_s(\boldsymbol{x}) = p_s \lambda \sum_k \frac{p_k \alpha_k(\boldsymbol{x})}{\eta_k}, \tag{5.7}$$

where

$$\eta_k = \frac{R_k}{R_K}. \tag{5.8}$$

p_k and η_k^{-1} weights the contribution of the k-th capacity region to the overall arrived rate. Therefore, classes with large R_k have reduced impact. $\alpha_k(\boldsymbol{x})$ is another factor that impacts the equivalent arrival rate, which represents the video bit rate ratio depending on the definitions of continuous video bit rate, Eq. (5.4) or discrete one, Eq. (4.5): Here, we give an example with continuous video bit rate selection,

$$\alpha_k(\boldsymbol{x}) = \frac{v_k(\boldsymbol{x})}{v_{\min}} = \frac{\max\Big(\min\Big(\gamma_k^s(\boldsymbol{x}), v_{\max}\Big), v_{\min}\Big)}{v_{\min}}. \tag{5.9}$$

On the other hand, the equivalent flow departure rates with the assumption of round-robin scheduling scheme are formulated as

$$\hat{\mu}_e = \frac{x^e}{|\boldsymbol{x}|}\frac{R_K}{\sigma}, \ \hat{\mu}_s = \frac{x^s}{|\boldsymbol{x}|}\frac{R_K}{v_{\min} T}. \tag{5.10}$$

With the flow departure rate and the flow arriving rate mentioned above, the approximated stationary distribution of $\hat{\pi}(x)$ can be obtained using the same concept of balance equations shown in Eq. (4.35). However, the complexity to solve the stationary distribution with one class is much lower. Here, we succeed to reduce the number of users class impacted by the radio conditions. Note that the stability condition of this approximation is same as Eq.(4.36) and that this approximation also holds when applying opportunistic scheduling; it is thus sufficient to multiply the throughput by the state dependent scheduling gain $G(|\boldsymbol{x}|)$. However, for the other scheduling schemes and consideration of users' mobility, more efforts for extension are needed.

Calculation of approximated KPIs of small chunk duration

As of the approximated performance metrics, the mean video bit rate, $\hat{v}_s$, the deficit rate, $\hat{D}_s$ and the buffer surplus, $\hat{B}_s$ for adaptive streaming traffic and the mean throughput for

elastic traffic, $\hat{\gamma}_e$, can be computed with the newly calculated stationary distribution $\hat{\pi}(\boldsymbol{x})$ as

$$\hat{v}_s = \frac{1}{\lambda p_s T} \sum_{\boldsymbol{x}:x_s>0} \frac{x_s \hat{\pi}(\boldsymbol{x})}{|\boldsymbol{x}|} \sum_k \frac{\beta_k(\boldsymbol{x})}{\bar{\beta}(\boldsymbol{x})} R_k, \tag{5.11}$$

$$\hat{D}_s = \sum_{\boldsymbol{x}:x_s>0} \frac{x_s \hat{\pi}(\boldsymbol{x})}{\bar{x}_s} \sum_k \frac{\beta_k(\boldsymbol{x})}{\bar{\beta}(\boldsymbol{x})} \mathbf{1}_{\{\gamma_k^s(\boldsymbol{x})<v_{\min}\}}, \tag{5.12}$$

$$\hat{B}_s = \sum_{\boldsymbol{x}:x_s>0} \frac{x_s \hat{\pi}(\boldsymbol{x})}{\bar{x}_s} \sum_k \frac{\beta_k(\boldsymbol{x})}{\bar{\beta}(\boldsymbol{x})} \Big(\frac{\gamma_k^s(\boldsymbol{x}) - v_k(\boldsymbol{x})}{v_k(\boldsymbol{x})} \Big), \tag{5.13}$$

$$\hat{\gamma}_e = \sum_{\boldsymbol{x}:x_e>0} \frac{x_e \hat{\pi}(\boldsymbol{x})}{\bar{x}_e} \frac{\phi_e(\boldsymbol{x})}{x_e} = \sum_{\boldsymbol{x}} \frac{\hat{\pi}(\boldsymbol{x}) \phi_e(\boldsymbol{x})}{\bar{x}_e}, \tag{5.14}$$

where

$$\beta_k(\boldsymbol{x}) = \frac{p_k \alpha_k(\boldsymbol{x})}{R_k},\ \bar{\beta}(\boldsymbol{x}) = \sum_k \beta_k(\boldsymbol{x}),\ \gamma_k^s(\boldsymbol{x}) = \frac{R_k}{|\boldsymbol{x}|}, \tag{5.15}$$

and $\frac{\beta_k(\boldsymbol{x})}{\bar{\beta}(\boldsymbol{x})}$ represents the fraction of load volume of class-k streaming users when there are in total $\boldsymbol{x}$ users in the system. In addition, $\bar{x}_s = \sum_{\boldsymbol{x}:x_s>0} x_s \hat{\pi}(\boldsymbol{x})$ and $\bar{x}_e = \sum_{\boldsymbol{x}:x_e>0} x_e \hat{\pi}(\boldsymbol{x})$ stand for the average number of streaming and elastic calls in the cell. It is also worth of mentioning that the approximation model can also predict all the metrics for each R_k as

$$\hat{v}_k^s = \frac{1}{\lambda p_s p_k T} \sum_{\boldsymbol{x}:x_s>0} \frac{x_s \hat{\pi}(\boldsymbol{x})}{|\boldsymbol{x}|} \frac{\beta_k(\boldsymbol{x})}{\bar{\beta}(\boldsymbol{x})} R_k, \tag{5.16}$$

$$\hat{D}_k^s = \sum_{\boldsymbol{x}:x_s>0} \frac{x_s \hat{\pi}(\boldsymbol{x})}{\bar{x}_s} \frac{\beta_k(\boldsymbol{x})}{\bar{\beta}(\boldsymbol{x})} \mathbf{1}_{\{\gamma_k^s(\boldsymbol{x})<v_{\min}\}}, \tag{5.17}$$

$$\hat{B}_k^s = \sum_{\boldsymbol{x}:x_s>0} \frac{x_s \hat{\pi}(\boldsymbol{x})}{\bar{x}_s} \frac{\beta_k(\boldsymbol{x})}{\bar{\beta}(\boldsymbol{x})} \Big(\frac{\gamma_k^s(\boldsymbol{x}) - v_k(\boldsymbol{x})}{v_k(\boldsymbol{x})} \Big), \tag{5.18}$$

where all the summations are taken off compared to the Eqs. 5.11 and elastic throughput can be obtained by

$$\hat{\gamma}_k^e = \frac{R_k}{R_{\text{eq}}} \hat{\gamma}_e, \quad \text{with } R_{\text{eq}} = \sum_k \frac{p_k}{R_k}.$$

5.8.2 Approximation model for significantly large chunk duration

In the case of large chunk duration, we propose an approximation model to reduce the system complexity. We can reduce the classes of users from both elastic and streaming services considering different radio conditions and video bit rates into a model classified by different video bit rate as $\boldsymbol{x} = (x^e, x_{v_1}^s, \cdots, x_{v_i}^s)$. We reformulate the arriving rate as

$$\hat{\lambda}^e = p_e \lambda \sum_k \frac{p_k}{\eta_k},\ \hat{\lambda}_i^s(\boldsymbol{x}) = \sum_k \frac{p_k \lambda_{k,i}(\boldsymbol{x})}{\eta_k}, \tag{5.19}$$

where $\eta_k = \frac{R_k}{R_K}$ has the same meaning as in previous section. Moreover, the corresponding flow departure rates are formulated as

$$\hat{\mu}^e = \frac{x^e}{|\boldsymbol{x}|}\frac{R_K}{\sigma},\ \hat{\mu}_i^s(\boldsymbol{x}) = \frac{x_i^s}{|\boldsymbol{x}|}\frac{R_K}{v_i T}. \tag{5.20}$$

Based on the arrival and departure rate, we get the stationary distribution $\pi(\boldsymbol{x})$ and we calculate the following KPIs using $\pi(\boldsymbol{x})$.

Calculation of approximated KPIs for large chunk duration

To calculate the KPIs, we need to know $\beta_{k,i}$, the probability that a flow of v_i belongs to radio condition R_k. Knowing the $\pi(\boldsymbol{x})$, we can obtain the probability that flows of R_k will choose v_i, calculated as

$$a_{k,i} = \sum_{\boldsymbol{x}} \pi(\boldsymbol{x}) \infty_{\{\lfloor \gamma_k^s(\boldsymbol{x}+\boldsymbol{e}_{k,i})\rfloor = v_i\}}, \tag{5.21}$$

and we then can calculate the value $\beta_{k,i}$ as

$$\beta_{k,i} = \frac{p_k a_{k,i}/R_k}{\sum_k p_k a_{k,i}/R_k}. \tag{5.22}$$

With $\beta_{k,i}$, we can calculate the performance metrics as the followings:

$$\hat{\bar{v}}_s = \frac{1}{\lambda p_s T}\sum_{\boldsymbol{x}} \frac{\hat{\pi}(\boldsymbol{x})}{|\boldsymbol{x}|}\sum_i\sum_k x_i^s \beta_{k,i} R_k, \tag{5.23}$$

$$\hat{D}_s = \sum_{\boldsymbol{x}} \frac{\hat{\pi}(\boldsymbol{x})}{\bar{\boldsymbol{x}}^s}\sum_i\sum_k x_i^s \beta_{k,i} \mathbf{1}_{\{\gamma_k^s(\boldsymbol{x}) < v_i\}}, \tag{5.24}$$

$$\hat{B}_s = \sum_{\boldsymbol{x}} \frac{\hat{\pi}(\boldsymbol{x})}{\bar{\boldsymbol{x}}^s}\sum_i\sum_k x_i^s \beta_{k,i}\Big(\frac{\gamma_k^s(\boldsymbol{x}) - v_i}{v_i}\Big), \tag{5.25}$$

$$\hat{\gamma}_e = \sum_{\boldsymbol{x}} \frac{\hat{\pi}(\boldsymbol{x})}{\bar{\boldsymbol{x}}^e} x^e \gamma^e(\boldsymbol{x}). \tag{5.26}$$

To conclude the two previous subsections 5.8.1 and 5.8.2, with the approximation model, we can reduce the number of classes as shown in Table. 5.3, where I is the number of video bit rate and K is the number of capacity regions. We can observe that large chunk duration always needs to have more classes than cases with small chunk duration.

5.8.3 Performance of approximation model

To validate the performance of approximation model, we follow the previous configurations in section 5.1 with two video bit rates and two radio conditions representing cell center and cell edge respectively. We set the parameters as $\mathscr{R} = \{R_C, R_E\} = \{10, 4\}$Mbps, $(p_C, p_E) = (\frac{1}{2}, \frac{1}{2})$ for cell center and cell edge, $(p_e, p_s) = (\frac{1}{2}, \frac{1}{2})$ for the elastic and streaming

Case	Chunk	# of Class(Exact)	# of Class(Approximated)
1	Large	$I*K+K$	$I+1$
2	Small	$1*K+K$	$1+1$

Table 5.3: System configuration of examined scenarios.

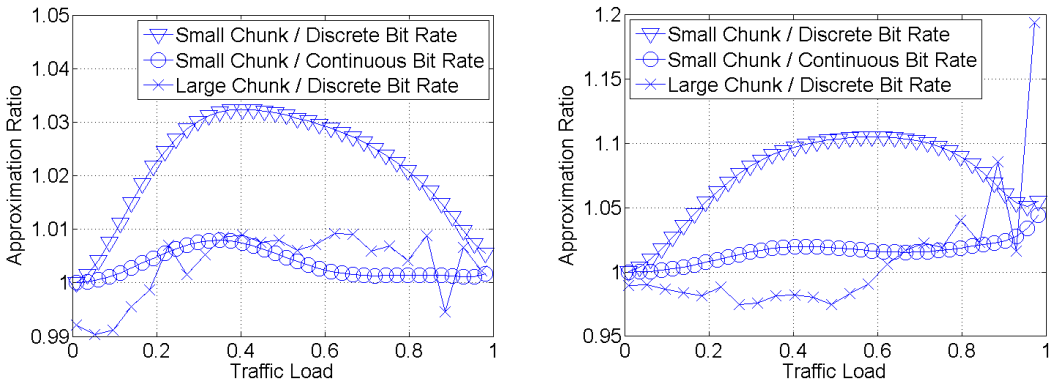
Chunk	# of Class(Exact)	# of Class(Approximated)
Large	6	3
Small	4	2

Table 5.4: System configuration of examined scenarios.

flows, $\mathcal{V} = \{v_1, v_2\} = \{2, 0.5\}$Mbps, $T = 10$s, $\sigma = 5$Mbits and $\lambda_{\max} = 1.14$ users/s. We demonstrate the number of classes that we are going to simulate in Table. 5.4 with $I = 2$ and $K = 2$.

In Fig. 5.13 and 5.14, we show the performance of approximation model by looking at two different metrics:

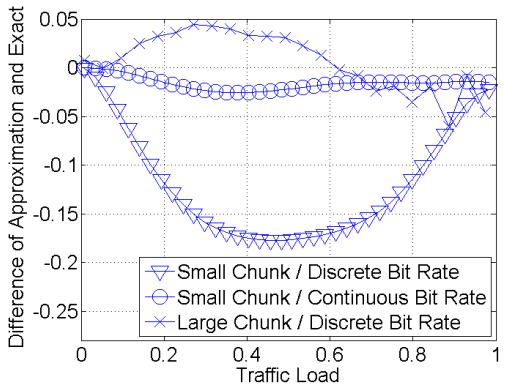
$$\text{Approximation Ratio} = \frac{\text{Approximation}}{\text{Exact}}, \quad \text{Approximation Difference} = \text{Approximation} - \text{Exact},$$



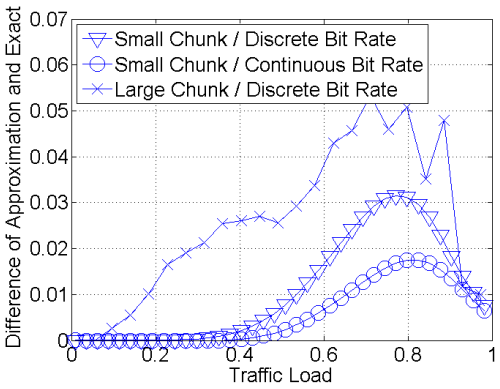
(a) Video Bit Rate (b) Service Time

Figure 5.13: Approximation ratio of video bit rate and service time.

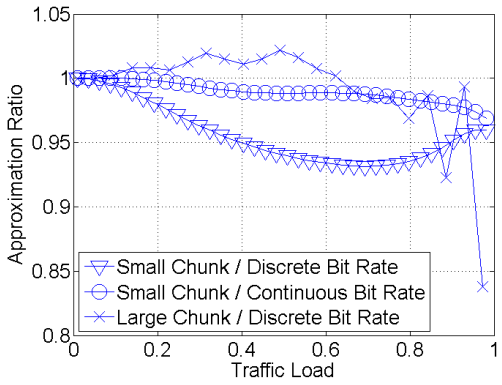
where approximation ratio is frequently used in studying the performance of approximation as [97]. Approximation difference is extended to avoid the case when the exact value of buffer surplus and deficit rate can approach 0. Compared to the maximum value, 2 for buffer surplus and 1 for deficit rate, the difference remains small and acceptable. It can also be concluded that the predictions of approximation model perform well generally.



(a) Buffer Surplus



(b) Deficit Rate



(c) Elastic Throughput

Figure 5.14: Approximation ratio and approximation difference.

Approximation model performs the best in the case with configuration of small chunk duration and continuous video bit rate. There are some larger differences between the exact and approximation results with discrete video bit rate configured, but they remain acceptable.

5.9 System dimensioning

Based on our models, operators can design their dimensioning algorithm allowing a certain traffic intensity obtained under some QoS constraints, which might be any combination of performance metrics we defined.

Here, we demonstrate an example utilizing realistic radio conditions based on measurement data from a 4G network in a large European city, with an average cell radius of 350 meters. The concerned frequency band is LTE 1800 MHZ. Figure 6.2 shows the measured probability distribution function of the Channel Quality Indicator (CQI) obtained from base station measurements collected using an Observation and Measurements (O&M) tool. Each CQI is associate to an MCS, determining its spectral efficiency. Using the CQI-MCS associ-

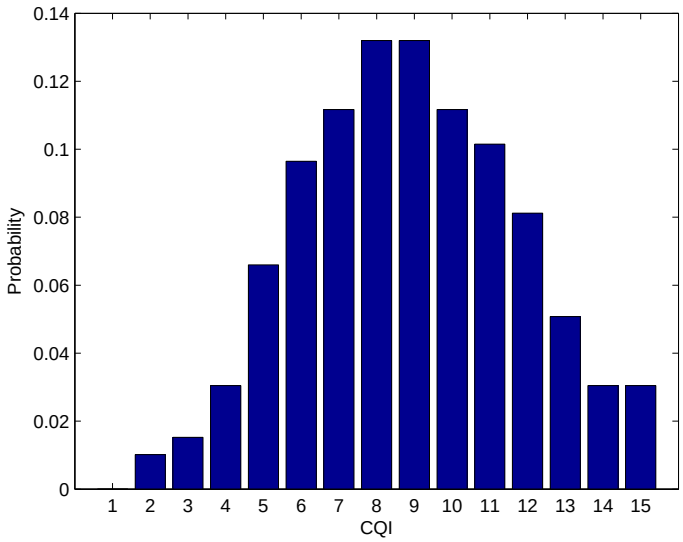


Figure 5.15: Measured CQI probability distribution function on a live LTE network.

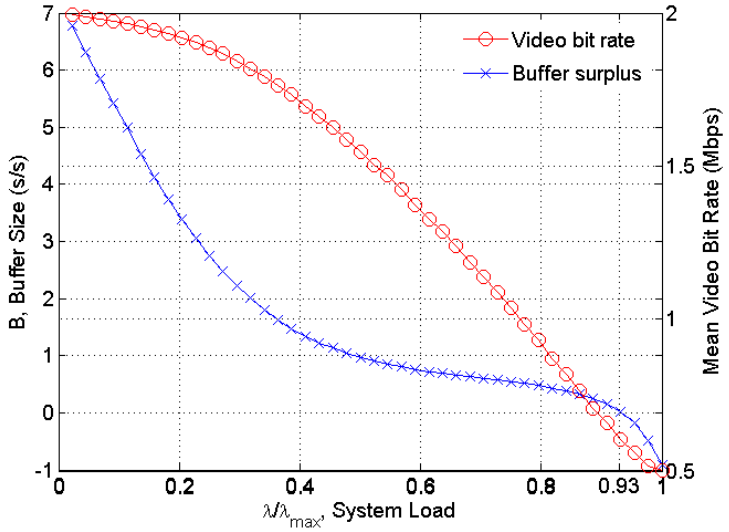


Figure 5.16: Dimensioning with real LTE system configuration.

ation figures of [39] and considering a bandwidth of 10 MHZ, the corresponding harmonic capacity of an LTE cell is computed as equal to $R_e = 16.82$ Mbps. As of the opportunistic scheduling gain, we make use of the scheduling gain calculated in [93] for a MIMO 2×2 LTE system and an AWGN channel and that converges to $G(\infty) = 1.7$ starting from a number of active users in the LTE cell equal to 15. Other traffic-related parameters are configured as $T = 20$s, $\sigma = 5$Mbits. We consider a discrete video bit rate with $(v_1, v_2) = (2, 0.5)$Mbps.

In Fig. 5.16, we show the simulation results of two metrics, video bit rate and buffer surplus. Based on the stability condition, we have $\lambda_{\max} = 2.189$ users/s and take $B = 0$ as an example of QoS constraint, the traffic intensity should be lower than $0.93 * 2.189 = 2.01$ users/s. Otherwise, the mean buffer surplus is smaller than zero, meaning that the starvation events happen very often. With the same concept, more QoS constraints can be considered together such as taking service time constraint as the average video duration as $S = T$.

5.10 Summary

In this chapter, simulation results show that configuring large chunk duration littlely improves video resolution but largely degrades the playback smoothness and that configuring larger number of video bit rate will also provide better video resolution but degrade the video smoothness. We also show the performance trade-off of different scheduling schemes with and without considering the users' mobility.

Moreover, we propose an approximation model in order to reduce the complexity of simulating a system with multiple classes. The numerical analysis validates that the approximation performs well compared to the exact solution. By applying the approximation model, ways of dimensioning is demonstrated. Different metrics, mean codec rate, deficit rate, buffer surplus and elastic mean throughput are taken as the main KPIs.

Chapter 6

Predicting QoE of Video Streaming with Machine Learning Technique

In the previous chapters, we studied the performance of both real-time streaming and HTTP-based streaming. To measure the QoE of HTTP-based streaming, metrics, the probability of *starvation events* or so-called buffer events are adopted popularly. *Starvation* or buffer events occur when the video buffer becomes empty and users encounter a video pause. As it is not easy to develop an analytical form for QoE metrics, especially for the subjective QoE, therefore, researches like [86] apply statistical analysis to understand the correlation between QoE and network QoS. In this chapter, we rely on a simulator which generates a big amount of data pairs (QoE+network features) and we demonstrate the efficiency of predicting QoE, video starvation, using the input network features such as CSI, the number of users' flows and video duration, recorded at the arrival of each user.

6.1 Problem statement and state-of-the-art

Different from the traditional metrics for measuring the quality of real-time video as we introduced in chapter 1, it is more reasonable to have other metrics for evaluating the Quality of Experience (QoE) [77] of progressive downloaded video. Regarding to the QoE of video users, objective performance metrics, such as starvation probability, rebuffering rate or mean duration of a rebuffering event are summarized in [46] and they are highly studied in researches like [100] and [44]. In fact, the mentioned performance metrics can be easily obtained by setting up a client-server testbed and measuring those video buffer statistics at TCP session level. Authors of [70] showed a correlation of users' QoE and network features by collecting those data from a simple testbed. Nevertheless, when it comes to the impact of wireless networks to the video performance, it is a challenging task to correlate the radio-related parameters like Channel State Information (CSI) to QoE metrics. Can we predict the video starvation using channel state information and other network features? The main difficulty on finding out this correlation lies on the distance (and consequent lack of cross-layer information) between where video application and radio information can be accessed.

Information of video services are monitored at higher level like application layer, which is not accessible for operators. Information of transport layer is recorded at the PDN Gateway but not sufficient for studying the QoE, video starvation. On the other hand, radio-related parameters are recorded at lower layer such as data link and physical layer. In order to understand this correlation, we establish a simulator which generate all the correlated data for streaming users to predict their video starvation.

Starvation probability is studied as QoE of video service in many works. It is modeled and calculated analytically in [100]. However, several constraints are needed when applying that model, e.g., fixed video bitrate is required and the model can only take a single channel condition into account. In [98], analytical form of starvation probability of adaptive streaming is proposed however without consideration of flow dynamics. We utilized flow-level model to investigate the video performance metrics in [30] and [66]. However, the relationship between video starvation and the proposed performance metrics are not clear. Machine learning has been widely used to study both subjective and objective QoE. [20] and [103] use machine learning to study the correlation between objective users' satisfaction and application metrics such as buffer times. Authors of [86] studied the QoE with TCP information. However, as aforementioned, the correlation between QoE and users' radio information are not considered.

Contributions

Our contribution is to demonstrate the correlation between the video starvation of different streaming and the recorded users' features. We also analyze the importance of these users' features to the prediction of video starvation events, including the number of video users existing in a cell, channel conditions of video user and video duration recorded when a flow starts its video download. To do so, we develop a C++ event-driven simulator that generates statistics per video streaming. In order to identify the most important performance features that predict the video starvation event, we apply two typical machine learning models for analysis. We show that generally prediction accuracy decreases as the system load increases and that users' mobility and having fixed bitrate will cause more video starvations. Considering both number of flows and radio condition is sufficient to achieve more than 92% of prediction accuracy for the static users but not sufficient for the mobile users. More features are needed to improve the prediction accuracy. We also show that video duration is not that important for predicting a video starvation and that number of flows and number of flows in starvation are similarly important to the prediction.

Organizations

The rest of chapter is organized as follows. In Section 6.2 we introduce two types of video streaming delivery and present the flow-level concept used to calculate the maximum flow arrival rate. Moreover, in the same Section, we also describe the structure of our simulator and the users' features that we examine for prediction performance. Section 6.3 introduces the two evaluated machine learning tools, Generalized Linear Model (GLM) and Support Vector Machine (SVM). Prediction performance of four different types of video streaming

are shown in Section 6.4 with access of all features and also limited access of all users' features are considered. Section 6.5 concludes the chapter and discusses the future works.

6.2 System Description

In this section, we first present two types of video streamings. Then we introduce the flow-level model used for calculating the maximum system load. Event-driven simulator is presented to generate data of video starvation for different loads.

6.2.1 Video streaming

According to the mechanism of HTTP streaming services as we mentioned in chapter 1, users request one video chunk after another. Generally, they can be categorized into fixed video bitrate streaming and adaptive streaming as following:

Fixed video bitrate streaming

Users fix a video bitrate, v_c, from the beginning till the end of video download, which is the simplest implementation.

Adaptive streaming

As Fig. 6.1, adaptive streaming services allow users to adapt to their video bitrate in real-time. After finishing downloading a video chunk with duration h, based on the measured throughput, γ_j, users can select a video bitrate for the next chunk from the discrete set $\mathcal{V} = \{v_1, \cdots, v_I\}$, where $v_1 > \cdots > v_I$. The selected video bitrate is given as

$$v_j = \begin{cases} \lfloor \gamma_j \rfloor, & \text{when } \gamma_j \geq v_I, \\ v_I, & \text{when } \gamma_j \leq v_I, \end{cases} \tag{6.1}$$

where $\lfloor y \rfloor$ is a shorthand notation for the largest video bitrate in $\mathcal{V}$ but not greater than y and γ_j stands for the instantaneous throughput of user j.

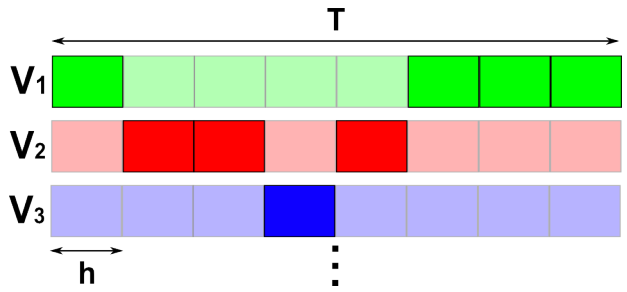


Figure 6.1: Video chunk selected by the adaptive streaming.

After downloaded, video chunks will be stored at a playout buffer. Like the previous sections, we still assume that the playout buffer of users is infinite. Once a user enters into the cell, there will be scheduled an amount of resource until the end of video download.

6.2.2 Radio access network and traffic characteristics

We consider a radio access network where users have different physical throughput depending on its location in the cell. We denote the set of physical throughput as $\mathscr{R} = \{R_1, \cdots, R_K\}$ and assume that the physical throughput of a static user remains the same during the transfer of the whole data flow (i.e., the video session). For mobile users, the physical throughput varies among these values.

In terms of traffic characteristics, we only consider streaming services in this model and we make the classical assumption that data flows having physical throughput R_k arrive as a Poisson process with rate $\lambda_k = p_k\lambda$, where λ is the overall flow arrival rate in the cell and p_k stands for the traffic proportion with physical throughput R_k, with $\sum_k p_k = 1$. The duration of the requests are assumed to be independent and exponentially distributed with mean T. The simulated streaming can be divided into four following types and each of them accounts for w_i proportion of arrivals with $\sum_i w_i = 1$, where $i \in$ {I, II, III, IV}

- Type I: static and adaptive streaming.
- Type II: mobile and adaptive streaming.
- Type III: static and fixed bitrate streaming.
- Type IV: mobile and fixed bitrate streaming.

In the real network, traffic arrival rate, λ, varies along hours, usually higher in day and lower at night. In the following simulation results, video starvation data are generated with different traffic arrival rates. Prediction shown at each traffic load corresponds to the potential performance at each hour.

6.2.3 Flow-level model and maximum arrival rate

The concept of flow-level model has been utilized to obtain the performance of streaming in paper [30] and [66]. Based on it, we can obtain the maximum flow arrival rate that guarantees the system stability. Let $\boldsymbol{x}(t) = (x_1^1(t), \cdots, x_K^i(t))$ be the number of streaming flows at time t and $x_k^i(t)$ be the number of type-i streaming with R_k at time t. Based on the round-robin scheduling method, we have the instantaneous throughput calculated as

$$\gamma_k^i(\boldsymbol{x}) = \frac{R_k}{|\boldsymbol{x}|}, \tag{6.2}$$

where $|\boldsymbol{x}| = \sum_i \sum_k x_k^i$ is the total number of ongoing flows. Video bitrate of the next video chunk is selected based on Eq.(6.1) and (6.2). When load approaches to system capacity and the mentioned traffic characteristics, all the adaptive streaming are forced to adapt to

v_I, the maximum system arrival rate can be calculated treating all the flows as an elastic traffic with volume vT as

$$\lambda_{\max} = \Big(\frac{w_1 v_I T}{R_s} + \frac{w_2 v_I T}{R_m} + \frac{w_3 v_c T}{R_s} + \frac{w_4 v_c T}{R_m}\Big)^{-1}. \tag{6.3}$$

$R_s = \Big(\sum_k \frac{p_k}{R_k}\Big)^{-1}$ stands for the equivalent radio throughput for static users. As work [73] mentioned, $R_m = \sum_k q_k R_k$ stands for the equivalent radio throughput for mobile users with q_k denoted as the proportion of time users stay with throughput R_k .

6.2.4 Event-driven simulator

In [30] and [66], by mathematical flow-level model, we can only obtain some objective QoS metrics instead of the real buffer information of j-th user, $b_j(t)$. Therefore, we implement an event-driven simulator which is able to simulate the video starvation event and record the buffer value of all users. Our simulator is implemented based on flow-level concept, where each video session is regarded as a flow and each of them may encounter the following events:

- **Arrival Event**: Flow arrival of streaming follows Poisson distribution. As user j arrives to the cell at time E_j^a, several observed features, $\boldsymbol{z}_j$, are recorded and used as the input data for predicting starvation, $y_j \in \{1, -1\}$.
- **Departure Event**: Users encounter a departure event at time E_j^d when its requested video is fully downloaded.
- **Chunk Event**: Users encounter a chunk event at time E_j^c when its requested video chunk with duration h is downloaded.
- **Buffer Event**: We classify the simulated streaming user j into three states, PREFETCH, PLAY and STARVATION, where each has a buffer variation rate

  $$\frac{db_j(t)}{dt} = \begin{cases} \gamma_j, & \text{PREFETCH or STARVATION,} \\ \gamma_j - 1, & \text{PLAY.} \end{cases}$$

  When user starts to request a video, it will stay at PREFETCH and switch to STARVATION until $b_j(t) \geq B$, where B is the initial buffer. Once $b_j(t) = 0$, user enters into STARVATION, where it is recognized as a user experiencing a video starvation, $y_j = 1$, and it will wait until $b_j(t) \geq B$ to enter again PLAY state. E_j^b is the time of buffer events for user j.
- **Mobility Event**: Users with mobility will change the R_j to the adjacent throughput when mobility event at time E_j^m occurs. In this event, users schedule the next mobility event based on the exponential distribution with rate ν_j.

Algorithm 1 summarizes the mechanism of the event-driven simulator with input parameters and output resutls listed in Table 6.1, where $\mathscr{E}$ is the set of all next event time and $\mathscr{X}$ is the set of all users in the cell.

Algorithm 1: Event-driven simulation

Input: $\mathscr{R}, \boldsymbol{p}, \lambda, l, T, B$ and h.
Output: $\boldsymbol{z} = (\boldsymbol{z}_1, \cdots, \boldsymbol{z}_l)$ and $\boldsymbol{y} = (y_1, \cdots, y_l)$
Initialize $\boldsymbol{y} = -\mathbf{1}$, $\mathscr{E} = \{E_1^a\}$ and $\mathscr{X} = \emptyset$;
while $j = 1; j \leq l$ **do**
    $E = \min_{e \in \mathscr{E}}(e)$;
    //**Arrival Event:**
    if $E == E_j^a$ **then**
        record $\boldsymbol{z}_j$, $j = j + 1$;
        update new E_j^a and put it in $\mathscr{E}$, put user j in $\mathscr{X}$;
    //**Departure Event:**
    else if $E == E_i^d$ **then**
        remove user i from $\mathscr{X}$;
    //**Chunk Event:**
    else if $E == E_i^c$ **then**
        if *user i is adaptive streaming user* **then**
            user i chooses a v in $\mathscr{V}$ based on its γ_i;
        else
            $E_i^c = E_i^d$;
    //**Buffer Event:**
    else if $E == E_i^b$ **then**
        if *next state is STARVATION* **then**
            change $\frac{db_i}{dt} = \gamma_i$ and $y_i = 1$;
        else if *next state is PLAY* **then**
            change $\frac{db_i}{dt} = \gamma_i - 1$;
    //**Mobility Event:**
    else if $E == E_i^m$ **then**
        change R_k of user h to R_{k+1} or R_{k-1};
    else
        Error happens;
    update all γ_i based on new $|\mathscr{X}|$;
    recalculate $E_i^d, E_i^c, E_i^d, \forall i \in \mathscr{X}$ and put them in $\mathscr{E}$;
obtain pairs $(\boldsymbol{z}, \boldsymbol{y})$;

Symbol	Notation	Unit
$\mathscr{R}$	Set of physical throughput	Mbps
$\boldsymbol{p}$	Proportion of traffic to each R	null
λ	Total arrival rate	1/s
l	Number of user arrivals	null
T	Streaming duration	s
B	Intial buffer for playing	s
h	Chunk duration	s
y_j	Starvation metrics of user j	binary
$\boldsymbol{z}_j$	Features recorded at user j's arrival	vector

Table 6.1: Notations of input parameters and output results.

6.2.5 Recorded features

In section 6.2.4, we have mentioned that $\boldsymbol{z}_j$ is recorded at the arrival of j-th user and is going to be used to predict y_j. Here, we introduce the components of user's feature, $\boldsymbol{z}_j$:

- R_j, Radio condition (R): It is recorded at the beginning of video download. If user is static, R_j is fixed.
- T_j, Video duration (T): It follows an exponential distribution.
- $\boldsymbol{x}_j$, Number of flows (N): $\boldsymbol{x}_j = (x_1, \cdots, x_K)_j$ stands for the number of flows in each region.
- $|\boldsymbol{x}_j|$, Total number of flows: $|\boldsymbol{x}_j| = \sum_k x_k$ represents the total number of flows in the cell.
- $\boldsymbol{x}_j^s$, Number of flows in starvation (N_s): $\boldsymbol{x}_j^s = (x_1^s, \cdots, x_K^s)_j$ stands for the number of flows experiencing video starvation in each region.
- $|\boldsymbol{x}_j^s|$, Total number of flows in starvation: $|\boldsymbol{x}_j^s| = \sum_k x_k^s$ represents the total number of flows experiencing starvation.

6.3 Machine Learning Tool

In this section, we introduce two efficient and widely used machine learning tools, GLM and SVM, which were used in the herein presented analysis. The performance metrics and reference the libraries of algorithms are also shown here.

6.3.1 Generalized Linear Model (GLM)

Given a training set of instance-label pairs $(\boldsymbol{z}_j, y_j)$, where $j = 1, \cdots, l$, $\boldsymbol{z}_j \in \mathscr{R}^n$ and $\boldsymbol{y} \in \{1, -1\}^l$, GLM with logistic regression model tries to minimize the following cost function

$$C(\boldsymbol{\theta}) = \frac{1}{l} \sum_{j=1}^{l} \log(g(-y_j \boldsymbol{\theta}^T \boldsymbol{z}_j)), \tag{6.4}$$

where $\boldsymbol{\theta} \in \mathscr{R}^n$ is a vector having the same dimention as $\boldsymbol{z}_j$ and $g(y_j \boldsymbol{\theta}^T \boldsymbol{z}_j) = (1 + e^{y_j \boldsymbol{\theta}^T \boldsymbol{z}_j})^{-1}$, known as logistic function or sigmoid function. With the obtained $\boldsymbol{\theta}_{\text{opt}} = \arg\min_{\boldsymbol{\theta}} C(\boldsymbol{\theta})$, we have the prediction function as

$$\hat{y}(\boldsymbol{z}) = \begin{cases} 1, & \text{when } g(\boldsymbol{\theta}_{\text{opt}}^T \boldsymbol{z}) \geq 0.5, \\ -1, & \text{when } g(\boldsymbol{\theta}_{\text{opt}}^T \boldsymbol{z}) < 0.5. \end{cases} \tag{6.5}$$

Compared with SVM, GLM technique needs less calculation time and it is easier to implement.

6.3.2 Support Vector Machine (SVM)

In [41], with the same training set as previous subsection but with $\boldsymbol{y} \in \{1, -1\}^l$, SVM solves the following optimization problem to obtain the optimal $\boldsymbol{w}$.

$$\begin{aligned} \min_{\boldsymbol{w}, b, \xi} \quad & \frac{1}{2} \boldsymbol{w}^T \boldsymbol{w} + C \sum_{j=1}^{l} \xi_j \\ \text{subject to} \quad & y_j (\boldsymbol{w}^T \phi(\boldsymbol{z}_j) + b) \leq 1 - \xi_j, \\ & \xi_j \geq 0. \end{aligned} \tag{6.6}$$

Based on different data characteristics, SVM provides us a different solution $\phi(\boldsymbol{z}_j)$ to choose. If data can not be easily separated by a hyperplane then other types of $\phi(\boldsymbol{z}_j)$ need to be considered. By solving the optimization problem using Lagarange multiplier, α_j^*, we can rewrite the decision function, $g(\boldsymbol{z})$ as

$$\begin{aligned} \hat{y}(\boldsymbol{z}) = g(\boldsymbol{z}) &= \text{sign}\Big(\sum_j y_j \alpha_j^* \phi^T(\boldsymbol{z}_j) \phi(\boldsymbol{z}) + b^* \Big) \\ &= \text{sign}\Big(\sum_j y_j \alpha_j^* K(\boldsymbol{z}_j, \boldsymbol{z}) + b^* \Big) \end{aligned} \tag{6.7}$$

We can choose a linear kernel as $K(\boldsymbol{z}, \boldsymbol{z}_j) = \boldsymbol{x}^T \boldsymbol{z}_j$, or other more elaborated kernel such as radial basis function (RBF): $K(\boldsymbol{z}, \boldsymbol{z}_j) = \exp(-\gamma \|\boldsymbol{z} - \boldsymbol{z}_j\|^2)$, where we do not optimize γ, but choose a default value $\gamma = 1$.

6.3.3 Performance metrics

In order to verify the machine learning performance, we define the following performance metrics to examine prediction performance among testing data.

$$P = P_{\{y_j=-1\}} P_{\{\hat{y}(\mathbf{z}_j)=-1|y_j=-1\}} + P_{\{y_j=1\}} P_{\{\hat{y}(\mathbf{z}_j)=1|y_j=1\}}.$$

This probability represents the average prediction accuracy among all testing samples.

6.3.4 Libraries

For the GLM analysis, we have used *R*'s *stats* package, which based its algorithm on the GLM's proposed by [55]. For the SVM numerical analysis of this works, we apply one of the most popular open-source SVM machine learning library, *LIBSVM*, proposed by [41] to investigate the prediction performance considering different network parameters.

6.4 Simulation Analysis

In this section, we first introduce the general system parameters that we configured for the simulators. Then we analyze the prediction performance of machine learning among different types of HTTP streaming with all recorded features. Finally we demonstrate the prediction performance considering only certain features.

6.4.1 Simulation configuration

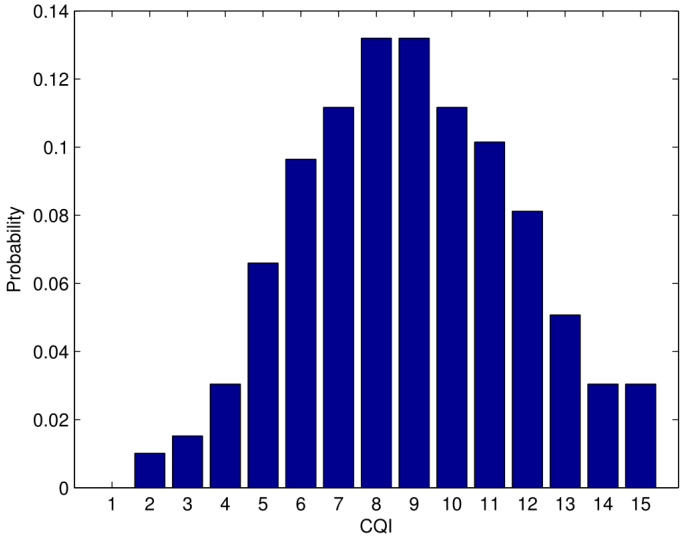


Figure 6.2: Measured CQI probability distribution function on a live LTE network.

Our simulator is launched based on the realistic radio conditions obtained from the measurement data of a 4G network in a large European city, with an average cell radius of 350 meters. The concerned frequency band is LTE 1800 MHZ. Figure 6.2 shows the measured probability distribution function of the Channel Quality Indicator (CQI) obtained from base station measurements collected using an Observation and Measurements (O&M) tool. Each CQI is associate to an Modulation Coding Scheme (MCS), determining its spectral efficiency.

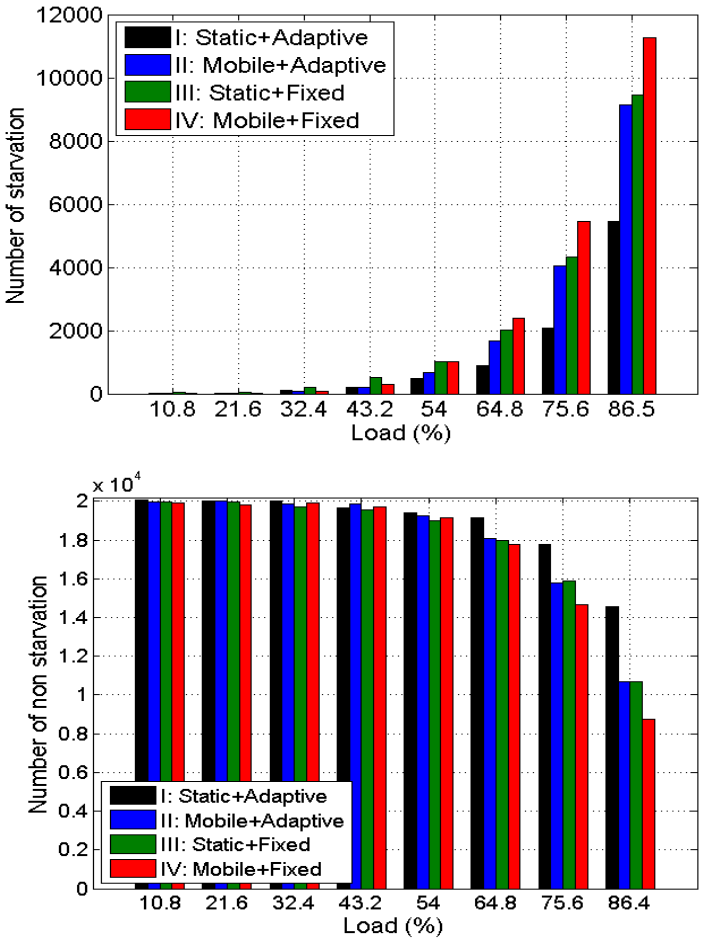


Figure 6.3: Training samples of four types of streaming along loads

In terms of streaming, we configure fixed video bitrate with $v_c = 1.5$Mbps and adaptive streaming with two bitrate options, $(v_1, v_2) = (2, 1)$Mbps. Static users will not move and mobile users have mobility rate as $v = 1$. Applying the previous CQI distribution, we simplify 15 CQIs into 5 CQIs as $\mathscr{R} = \{49.24, 31.89, 18.84, 9.22, 2.975\}$ Mbps and its corresponding traffic proportion are calculated as $\boldsymbol{p} = (11.1\%, 29.4\%, 37.5\%, 19.2\%, 2.8\%)$. The rest of system configuration are $T = 40$s and $B = 1$s. With these settings and based on Eq. (6.3), we can obtain the maximum traffic arrival rate as $\lambda_{\max} = \frac{40}{4}\left(\frac{1.5+1}{16.23} + \frac{1.5+1}{21.51}\right)^{-1} = 0.37$.

As we have mentioned that traffic load might vary along hours, we demonstrated eight flow arrival rate normalized by the maximum value λ_{max}. For each λ, simulator generates $l = 10^6$ streaming arrivals. 80% of data are randomly selected for training and 20% of data for validation among all the samples. In Fig. 6.3, the average number of video starvation and non-starvation used for training is listed for each load. It can be observed that when load is small, starvation seldomly happens. However, when load approaches to λ_{max}, more video flows experience video starvation. It is also shown that static users and adaptive streaming users experience less video starvation than mobile and fixed bitrate users.

6.4.2 Prediction performance of different HTTP streaming

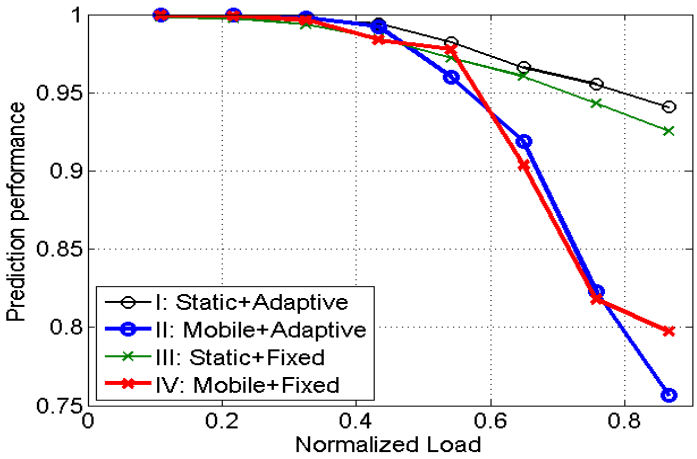


Figure 6.4: Average prediction performance with GLM

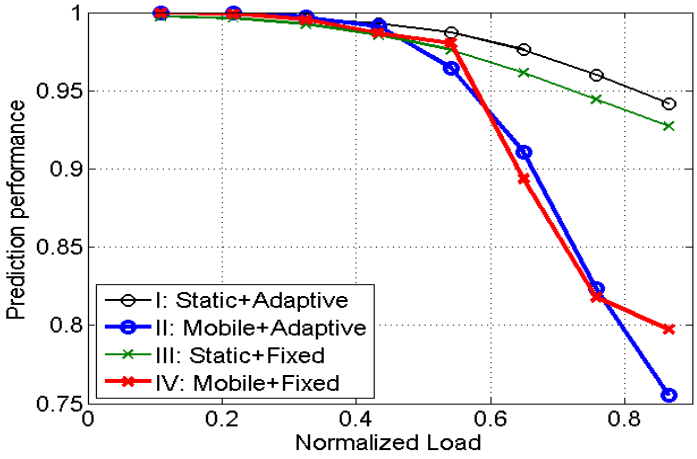
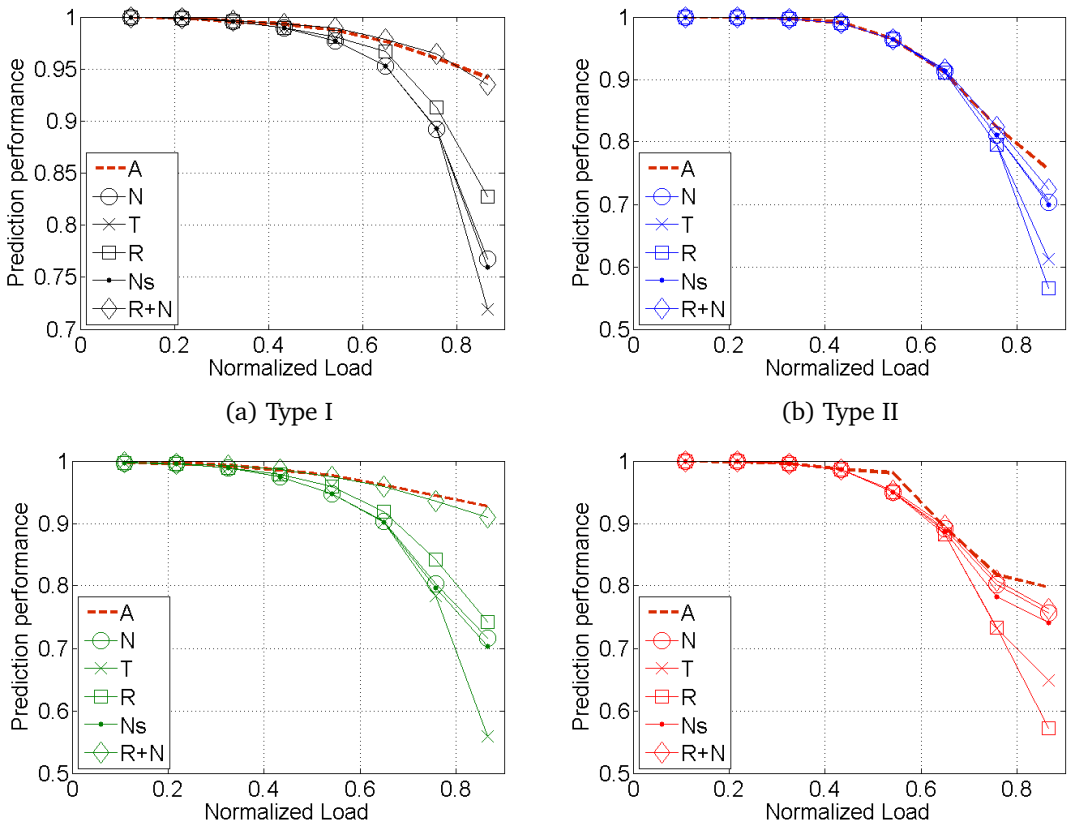


Figure 6.5: Average prediction performance with SVM

Applying the introduced machine learning techniques, we obtain the two following figures showing the average prediction performance, P, with GLM in Fig. 6.4 and with SVM in Fig. 6.5. We can observe that the prediction performance of GLM and SVM are similar. In addition, no matter which machine learning tool we use, we can observe that as load increases, prediction performance will decrease because of the increase of uncertainty. From the simulation results, we show that QoE of mobile users are much more difficult to predict, especially when load is large. However, static users can achieve more than 90% of accuracy. There is no general rule saying that fixed bitrate is easier to predict than adaptive streaming. It depends on mobility. For static users with adaptive streaming property, prediction of QoE is more accurate. For static users, even almost 95% of accuracy can be achieved at high load. However, for mobile users, it can be said that the initial information are not sufficient for prediction.

6.4.3 Prediction performance of different features



(a) Type I

(b) Type II

(c) Type III

(d) Type IV

Figure 6.6: Prediction performance considering different network features.

In this section, we verify the prediction performance considering limited access to certain

users' features. As we have shown that GLM and SVM performs similar results in Fig. 6.4 and 6.5. We only demonstrate the SVM prediction performance of four types of streaming in Fig. 6.6. It is shown that considering all features (A) can always perform the highest prediction accuracy and that N and N_s provide very similar information. It may be a good news for operators that they do not need to know more application information from users' side. Moreover, we can also observe that R information becomes less important when users are mobile and T is also less important. This may be caused by our infinite buffer assumption. In Fig. 6.6, we also demonstrate that considering both R and N can provide good results in static case.

6.5 Summary

In this chapter, by applying the concept of flow-level dynamics, we examine the prediction performance of video starvation of different video streaming by looking at different users' features. We develop an event-driven simulator in C++ and we have used supervised machine learning techniques, GLM and SVM, to obtain our training model along the system loads. We correlate the users' QoE and features, and simulation results show that the prediction performs better when users are static and adaptive streaming. For mobile users, prediction accuracy degrades. In terms of users' features, it can be observed that users' features such as number of flows and radio conditions lead to better prediction of starvation event than others. With the two users' features, we can obtain more than 92% of prediction accuracy in the static cases. In addition, we also observe that GLM provides similar performance with regards to SVM. Future works will consider other QoE metrics like rebuffering time and the evaluation with other machine models such as Random Forests, Neural Networks, Naive Bayes, and K-Nearest Neighbors to widen the view on other techniques which may improve the prediction accuracy. In addition, prediction accuracy of other configurations needs to be checked.

Chapter 7

Conclusions and Future Works

Measuring and improving the QoE of video become more and more important as video accounts for more than 50% of network traffic. In this thesis, we propose models for dimensioning different types of streaming services, including real-time streaming and HTTP adaptive streaming inside wireless networks. Both of them are developed by applying the concepts of flow-level dynamics for modeling the arrivals and the departures of the traffic demand, which is highly used for both elastic traffic and real-time streaming traffic in the literature.

In chapter 3, we develop a flow-level traffic model for real-time streaming services. We assume the existence of quasi-stationary property and combine both the flow and packet levels to calculate the packet outage rate. In addition, we show that fluid approximation can be adapted to apply at the packet level based on different delay constraints for different types of real-time streaming. Using our model, operators could design the corresponding admission control algorithm for real-time streaming services with a guaranteed packet outage rate.

In chapter 4, we develop a flow-level traffic model for HTTP streaming services. The model takes the flow-level dynamics into account and verify the performance impacts of different parameters such as video chunk duration, number of video bit rates and scheduling schemes, etc. We address in the following are the potential questions encountered by operators when they deploy the HTTP adaptive streaming service and want to improve the users' quality-of-experience:

1. **Impact of video chunk duration:** Deploying larger video chunk duration will decrease video smoothness. However, it gives relatively little gain concerning the video resolution.

2. **Impact of number of video bit rates:** Configuring more options of video bit rates provides better mean video resolution but decreases the video smoothness in the case of small chunk duration.

3. **Design of video chunk duration:** Designing the number of video chunk transmitted in a HTTP request based on providing the same video volume for each video bit rate could offer a similar video performance as deploying smaller video chunk duration, which offer another option of deployment for video service providers.

4. **Impact of scheduling schemes to adaptive streaming:** In the static case, we suggest to deploy round-robin scheduling or opportunistic scheduling schemes but not max-C/I and max-min scheme. This conclusion is different from the one for elastic traffic saying there is no difference among all scheduling schemes.

5. **Impact of users' mobility:** When users' mobility is considered inside the cell, deploying RR has better video smoothness but less video resolution. Vice versa for the max C/I scheduling scheme. We show that implementing discriminatory scheduling can achieve an intermediate performance.

In addition to the qualitative conclusions given above, our other contribution is to provide the quantitative results, which could assist both the service providers and networks operators to well dimension their systems given certain performance constraints.

In chapter 6, as we found that it is difficult to find out an exact analytical form for video QoE including all the possible system parameters, we propose to utilize the machine learning technique to predict the video quality. The results also help us to understand the correlation between the video starvation and the users features recorded at each arrival. In the last part of our thesis, we examine the prediction performance using GLM and SVM with different system loads. We found out that the prediction accuracies are more than 92% for the static users at each load and the most important network parameters include the radio condition and flows number, which shows that prediction video starvation is feasible for static users. However, more users' features are needed to well predict the starvation events.

Future works

For real-time streaming services, an important future work would be the extension of the model to consider real time streaming with adaptive video bit rate, on both packet and flow levels.

As of HTTP adaptive streaming traffic, although we have computed important KPIs for video smoothness that are correlated with one of the common video QoE indicator, starvation probability, obtaining the analytical form of starvation probability is still a challenging future work. We have started a set of works on this that did not lead to concrete results, but that may be good guidance for future works. More in details, in order to obtain an analytical approximation of starvation probability, we have tried to model the metrics by applying the concepts developed in [67] and [50]: Assuming that elastic traffic dynamics are fast enough, the packet arrival process of streaming service can be modeled as a Brownian motion and important metrics can be computed. However, this assumption for elastic traffic dynamics being unrealistic, the approximation did not seem to work well. More investigations may be needed in the future.

Concerning the QoE metrics, future works may study other video QoE metrics as those mentioned in the introduction section, such as join time, buffer ratio, etc [46], in addition to starvation probability considered in this thesis. The performance impact of buffer configuration is also a good direction for research while we assume that playout buffer is infinite.

The most important works, from our perspective, are related to machine learning chapter. Indeed, predicting the video QoE without having to rely completely on analytical solution is a first step towards the exploitation of the network data for predicting and enhancing video QoE. For example, a first step for improving the prediction error is to try other learning models with more randomness like Decision Tree, Random Forest, Neural Network and K-Nearest Neighbors. In addition, because our training data are generated based on the simulators of Poisson arrival, it is also better to implement a simulator with a more realistic traffic profile or to use real network measurements. Adding features related to mobility is also important for QoE prediction. Once the QoE prediction methodology has been improved, the second step would be to use it for enhancing QoE, by proposing advanced self-organizing algorithms. How to implement efficient yet simple self-organizing algorithms for enhancing QoE of streaming services could be an interesting topic for another Ph.D thesis.

Bibliography

[1] "Akamai HD for iPhone Encoding Best Practices - Akamai HD Network," Tech. Rep.

[2] "Microsoft Corporation: IIS Smooth Streaming Technical Overview," Tech. Rep.

[3] "Real-Time Streaming Protocol (RTSP)," IETF, RFC 2326, Apr. 1998.

[4] "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015 to 2020 White Paper," CISCO, TR, Feb. 2016.

[5] "ITU-T recommendation p.800, Methods for subjective determination of transmission quality," Tech. Rep., August 1996.

[6] I. D. 23009-1, "Information Technology - Dynamic Adaptive Streaming Over HTTP (DASH) - part 1: Media Presentation Description and Segment Formats," Tech. Rep., 2011.

[7] 3GPP, "Transparent end-to-end packet switched streaming service (pss); protocols and codec, TS 36.942," Tech. Rep.

[8] ——, "Evolved Universal Terrestial Radio Access (E-UTRA) further advancements for E-UTRA physical layer aspects TR 36.814," Tech. Rep., 2012.

[9] ——, "Evolved Universal Terrestial Radio Access (E-UTRA) radio frequency RF system scenarios TR 36.942," Tech. Rep., 2012.

[10] S. Aalto, A. Penttinen, P. Lassila, and P. Osti, "On the optimal trade-off between srpt and opportunistic scheduling," in *Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems*, ser. SIGMETRICS '11. New York, NY, USA: ACM, 2011, pp. 185–196. [Online]. Available: http://doi.acm.org/10.1145/1993744.1993761

[11] N. Abbas, T. Bonald, and B. Sayrac, "Opportunistic gains of mobility in cellular data networks," in *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2015 13th International Symposium on*, May 2015, pp. 315–322.

[12] V. S. Abhayawardhana, I. J. Wassell, D. Crosby, M. P. Sellars, and M. G. Brown, "Comparison of empirical propagation path loss models for fixed wireless access systems,"

in *2005 IEEE 61st Vehicular Technology Conference*, vol. 1, May 2005, pp. 73–77 Vol. 1.

[13] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning From Data*. AMLBook, 2012.

[14] Adobe., "Video technology center dlivery, Progressive download ." Tech. Rep., August 1996. [Online]. Available: http://www.adobe.com/devnet/video/progressive.html

[15] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http," in *Proceedings of the second annual ACM conference on Multimedia systems*. ACM, 2011, pp. 157–168.

[16] E. Altman, C. Barakat, E. Laborde, P. Brown, and D. Collange, "Fairness analysis of TCP/IP," in *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, vol. 1. IEEE, 2000, pp. 61–66.

[17] N. Andrew, "CS229 lecture notes: Part V Support Vector Machines," University Lecture, 2016.

[18] T. Arsan, "Review of bandwidth estimation tools and application to bandwidth adaptive video streaming," in *High Capacity Optical Networks and Emerging/Enabling Technologies*, Dec 2012, pp. 152–156.

[19] U. Ayesta, M. Erausquin, and P. Jacko, "A modeling framework for optimizing the flow-level scheduling with time-varying channels," *Performance Evaluation*, vol. 67, no. 11, pp. 1014 – 1029, 2010, performance 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0166531610001124

[20] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang, "Developing a predictive model of quality of experience for internet video," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 339–350, Aug. 2013. [Online]. Available: http://doi.acm.org/10.1145/2534169.2486025

[21] B. Blaszczyszyn, M. Jovanovic, and M. Kadhem Karray, "Quality of Real-Time Streaming in Wireless Cellular Networks - Stochastic Modeling and Analysis," *ArXiv e-prints*, Apr. 2013.

[22] Bolch and G. et al., *Queueing Networks and Markov Chains*. Wiley, 2000.

[23] T. Bonald, "A score-based opportunistic scheduler for fading radio channels," in *Proceedings of European Wireless*, 2004.

[24] T. Bonald and A. ProutiÃ¨re, "Insensitivity in processor-sharing networks," *Performance Evaluation*, vol. 49, no. 1-4, pp. 193 – 209, 2002, performance 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0166531602001104

[25] T. Bonald, "Insensitive traffic models for communication networks," *Discrete Event Dynamic Systems*, vol. 17, no. 3, pp. 405–421, 2007.

[26] T. Bonald, S. Borst, N. Hegde, M. Jonckheere, and A. Proutiere, "Flow-level performance and capacity of wireless networks with user mobility," *Queueing Systems*, vol. 63, no. 1-4, pp. 131–164, 2009.

[27] T. Bonald, S. Borst, N. Hegde, and A. Proutière, "Wireless data performance in multi-cell scenarios," *ACM SIGMETRICS Performance Evaluation Review*, vol. 32, no. 1, pp. 378–380, 2004.

[28] T. Bonald, S. Borst, and A. Proutiere, "Inter-cell coordination in wireless data networks," *European Transactions on Telecommunications*, vol. 17, no. 3, pp. 303–312, 2006.

[29] T. Bonald, S. C. Borst, and A. Proutière, "How mobility impacts the flow-level performance of wireless data systems," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 3. IEEE, 2004, pp. 1872–1881.

[30] T. Bonald, S. Elayoubi, and Y.-T. Lin, "A flow-level performance model for mobile networks carrying adaptive streaming traffic," *IEEE Globecom*, 2015.

[31] T. Bonald and M. Feuillet, "On the stability of flow-aware CSMA," *CoRR*, vol. abs/1003.5068, 2010. [Online]. Available: http://arxiv.org/abs/1003.5068

[32] ——, *Network Performance Analysis*. Wiley, 2011.

[33] T. Bonald, P. Olivier, and J. Roberts, "Dimensioning high speed ip access networks," in *proceedings of the 8th International Teletraffic Congress (ITC)*, 2003, pp. 241–251.

[34] T. Bonald and A. Proutière, "Wireless downlink data channels: user performance and cell dimensioning," in *Proceedings of the 9th annual international conference on Mobile computing and networking*. ACM, 2003, pp. 339–352.

[35] T. Bonald and A. Proutière, "On performance bounds for the integration of elastic and adaptive streaming flows," in *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems*, ser. SIGMETRICS '04/Performance '04. New York, NY, USA: ACM, 2004, pp. 235–245. [Online]. Available: http://doi.acm.org/10.1145/1005686.1005716

[36] S. Borst and N. Hegde, "Integration of streaming and elastic traffic in wireless networks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, May 2007, pp. 1884–1892.

[37] S. Borst, N. Hegde, and A. Proutiere, "Mobility-driven scheduling in wireless networks," in *INFOCOM 2009, IEEE*. IEEE, 2009, pp. 1260–1268.

[38] S. C. Borst, A. Proutiere, and N. Hegde, "Capacity of wireless data networks with intra-and inter-cell mobility." in *INFOCOM*, 2006.

[39] Y. Bouguen, E. Hardouin, and F.-X. Wolff, *LTE et les réseaux 4G*. Editions Eyrolles, 2012.

[40] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '07. New York, NY, USA: ACM, 2007, pp. 1–14. [Online]. Available: http://doi.acm.org/10.1145/1298306.1298309

[41] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[42] R. Combes, S.-E. Elayoubi, and Z. Altman, "Cross-layer analysis of scheduling gains: Application to lmmse receivers in frequency-selective rayleigh-fading channels," in *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2011 International Symposium on*. IEEE, 2011, pp. 133–139.

[43] E. Dahlman, S. Parkvall, and J. Skold, *4G: LTE/LTE-Advanced for Mobile Broadband*, 1st ed. Academic Press, 2011.

[44] L. De Cicco and S. Mascolo, "An experimental investigation of the akamai adaptive video streaming," in *Proceedings of the 6th International Conference on HCI in Work and Learning, Life and Leisure: Workgroup Human-computer Interaction and Usability Engineering*, ser. USAB'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 447–464. [Online]. Available: http://dl.acm.org/citation.cfm?id=1947789.1947828

[45] L. De Cicco, S. Mascolo, and V. Palmisano, "Feedback control for adaptive live video streaming," in *Proceedings of the Second Annual ACM Conference on Multimedia Systems*, ser. MMSys '11. New York, NY, USA: ACM, 2011, pp. 145–156. [Online]. Available: http://doi.acm.org/10.1145/1943552.1943573

[46] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang, "Understanding the impact of video quality on user engagement," in *Proceedings of the ACM SIGCOMM 2011 Conference*, ser. SIGCOMM '11. New York, NY, USA: ACM, 2011, pp. 362–373. [Online]. Available: http://doi.acm.org/10.1145/2018436.2018478

[47] S. P. Erik Dahlman and J. Skold, *4G LTE/LTE-Advanced for Mobile Broadband*. Elsevier, 2011.

[48] A. Feldmann, A. Gilbert, P. Huang, and W. Willinger, "Dynamics of ip traffic: A study of the role of variability and the impact of control," 1999, pp. 301–313.

[49] L. Fenton, "The sum of log-normal probability distributions in scatter transmission systems," *IRE Transactions on Communications Systems*, vol. 8, no. 1, pp. 57–67, March 1960.

[50] I.-H. Hou and P.-C. Hsieh, "Qoe-optimal scheduling for on-demand video streams over unreliable wireless networks," in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '15. New York, NY, USA: ACM, 2015, pp. 207–216. [Online]. Available: http://doi.acm.org/10.1145/2746285.2746288

[51] ITU-R Recommendation BT.500, "Methodology for the subjective assessment of the quality of television pictures," International Telecommunication Union , Geneva, Tech. Rep., 2002.

[52] ITU-T Recommendation P.910, "Subjective video quality assessment methods for multimedia applications," International Telecommunication Union , Geneva, Tech. Rep., 2008.

[53] ITU-T Recommendation P.911, "Methods for subjective determination of transmission quality," Tech. Rep., August 1996.

[54] V. Iversen and L. Staalhagen, "Waiting time distribution in M/D/1 queueing systems," *IEE Electronics Letters*, vol. 35, no. 25, pp. 2184–2185, 1999.

[55] R. W. M. W. J. A. Nelder, "Generalized linear models," *Journal of the Royal Statistical Society. Series A (General)*, vol. 135, no. 3, pp. 370–384, 1972. [Online]. Available: http://www.jstor.org/stable/2344614

[56] D. Jarnikov and T. Ozcelebi, "Client intelligence for adaptive streaming solutions," in *Multimedia and Expo (ICME), 2010 IEEE International Conference on*, July 2010, pp. 1499–1504.

[57] M. K. Karray and Y. Khan, "Evaluation and comparison of resource allocation strategies for new streaming services in wireless cellular networks," in *Third International Conference on Communications and Networking*, March 2012, pp. 1–7.

[58] M. K. Karray, "User's mobility effect on the performance of wireless cellular networks serving elastic traffic," *Wireless Networks*, vol. 17, no. 1, pp. 247–262, 2011.

[59] F. Khan, *LTE for 4G Mobile Broadband: Air Interface Technologies and Performance*, 1st ed. New York, NY, USA: Cambridge University Press, 2009.

[60] A. Khlass, T. Bonald, and S. Elayoubi, "Flow-level performance of intra-site coordination in cellular networks," in *Modeling Optimization in Mobile, Ad Hoc Wireless Networks (WiOpt), 2013 11th International Symposium on*, May 2013, pp. 216–223.

[61] L. Kleinrock, *Queueing Systems: Theory*, ser. A Wiley-Interscience publication. Wiley, 1976, no. 1.

[62] ——, *Queueing Systems*. Wiley Interscience, 1976, vol. II: Computer Applications, (Published in Russian, 1979. Published in Japanese, 1979.).

[63] T. E. Kolding, K. I. Pedersen, J. Wigard, F. Frederiksen, and P. E. Mogensen, "High speed downlink packet access: Wcdma evolution," *IEEE Vehicular Technology Society News*, vol. 50, no. 1, pp. 4–10, 2003.

[64] H. J. Kushner and P. A. Whiting, "Convergence of proportional-fair sharing algorithms under general conditions," *IEEE Transactions on Wireless Communications*, vol. 3, no. 4, pp. 1250–1259, July 2004.

[65] X. Li, *Radio Access Network Dimensioning for 3G UMTS*. Vieweg+Teubner Verlag, 2011.

[66] Y.-T. Lin, T. Bonald, and S. Elayoubi, "Impact of chunk duration on adaptive streaming performance in mobile networks," *IEEE WCNC*, 2016.

[67] T. H. Luan, L. X. Cai, and X. Shen, "Impact of network dynamics on user's video quality: analytical framework and qos provision," *IEEE Transactions on Multimedia*, vol. 12, no. 1, pp. 64–78, 2010.

[68] L. MassouliÃ© and J. Roberts, "Bandwidth sharing and admission control for elastic traffic," *Telecommunication Systems*, vol. 15, no. 1-2, pp. 185–201, 2000.

[69] K. Miller, E. Quacchio, G. Gennari, and A. Wolisz, "Adaptation algorithm for adaptive streaming over http," in *2012 19th International Packet Video Workshop (PV)*, May 2012, pp. 173–178.

[70] R. K. P. Mok, E. W. W. Chan, and R. K. C. Chang, "Measuring the quality of experience of http video streaming," in *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*, May 2011, pp. 485–492.

[71] R. K. Mok, X. Luo, E. W. Chan, and R. K. Chang, "QDASH: a QoE-aware DASH system," in *Proceedings of the 3rd Multimedia Systems Conference*. ACM, 2012, pp. 11–22.

[72] R. Mugisha and N. Ventura, "Packet scheduling for VoIP over LTE-A," in *AFRICON, 2013*, Sept 2013, pp. 1–6.

[73] Y.-T. L. Nivine Abbas and B. Sayrac, "Mobility-driven scheduler for mobile networks carrying adaptive streaming traffic," *IEEE PIMRC*, 2016.

[74] P. Olivier, *Internet Data Flow Characterization and Bandwidth Sharing Modelling*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 986–997. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-72990-7_85

[75] ——, "Performance evaluation of multi-rate streaming traffic by quasi-stationary modelling," vol. 5894, pp. 30–44, 2009.

[76] O. Oyman and S. Singh, "Quality of experience for http adaptive streaming services," *IEEE Communications Magazine*, vol. 50, no. 4, pp. 20–27, April 2012.

[77] S. M. Patrick Le Callet and A. Perkis, "Qualinet White Paper on Definitions of Quality of Experience (2012)," Tech. Rep. Switzerland, Version 1.2., March 2013.

[78] C. Perkins, *RTP: Audio and Video for the Internet*, 1st ed. Addison-Wesley Professional, 2003.

[79] M. Poikselkä, H. Holma, J. Hongisto, J. Kallio, and A. Toskala, *Voice over LTE (VoLTE)*. Wiley, 2012. [Online]. Available: https://books.google.fr/books?id=baCd1wIsyLsC

[80] J. Roberts and S. Oueslati-Boulahia, "Quality of service by flow–aware networking," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 358, no. 1773, pp. 2197–2207, 2000.

[81] R. Schatz, T. Hossfeld, L. Janowski, and S. Egger, "Datatraffic monitoring and analysis," E. Biersack, C. Callegari, and M. Matijasevic, Eds. Berlin, Heidelberg: Springer-Verlag, 2013, ch. From Packets to People: Quality of Experience As a New Measurement Challenge, pp. 219–263. [Online]. Available: http://dl.acm.org/citation.cfm?id=2555672.2555685

[82] R. Serfozo, *Introduction to Stochastic Networks*, ser. Stochastic Modelling and Applied Probability. Springer New York, 1999. [Online]. Available: https://books.google.ca/books?id=9IuVTP_GpfgC

[83] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hossfeld, and P. Tran-Gia, "A survey on quality of experience of http adaptive streaming," *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 469–492, Firstquarter 2015.

[84] M. K. Simon and M.-S. Alouini, *Digital communication over fading channels*. John Wiley & Sons, 2005, vol. 95.

[85] W. Simpson and H. Greenfield, *IPTV and Internet Video: Expanding the Reach of Television Broadcasting*. Taylor & Francis, 2012. [Online]. Available: https://books.google.com.tw/books?id=QK7Vwkzl0cIC

[86] K. D. Singh, Y. Hadjadj-Aoul, and G. Rubino, "Quality of experience estimation for adaptive http/tcp video streaming using h.264/avc," in *2012 IEEE Consumer Communications and Networking Conference (CCNC)*, Jan 2012, pp. 127–131.

[87] B. Sklar, "Rayleigh fading channels in mobile digital communication systems part ii: Mitigation," *IEEE Communications Magazine*, vol. 35, no. 9, pp. 148–155, Sept 1997.

[88] I. Sodagar, "The MPEG-DASH standard for multimedia streaming over the internet," *MultiMedia, IEEE*, vol. 18, no. 4, pp. 62–67, April 2011.

[89] T. Stockhammer, "Dynamic adaptive streaming over http –: Standards and design principles," in *Proceedings of the Second Annual ACM Conference on Multimedia Systems*, ser. MMSys '11. New York, NY, USA: ACM, 2011, pp. 133–144. [Online]. Available: http://doi.acm.org/10.1145/1943552.1943572

[90] D. Tse, "Multiuser diversity in wireless networks," in *Wireless Communications Seminar, Standford University*, 2001.

[91] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[92] O. Verscheure, P. Frossard, and M. Hamdi, "User-oriented QoS analysis in MPEG-2 video delivery," *Real-Time Imaging*, vol. 5, no. 5, pp. 305–314, Oct. 1999. [Online]. Available: http://dx.doi.org/10.1006/rtim.1999.0175

[93] Y. Wang, "System level analysis of lte-advanced: with emphasis on multi-component carrier management," Ph.D. dissertation, Department of Electronic Systems, Aalborg University, 2010.

[94] ——, "Survey of objective video quality measurements," 2006.

[95] S. Winkler and P. Mohandas, "The evolution of video quality measurement: From PSNR to hybrid metrics," *IEEE Transactions on Broadcasting*, vol. 54, no. 3, pp. 660–668, Sept 2008.

[96] D. Wu, Y. T. Hou, and Y.-Q. Zhang, "Transporting real-time video over the internet: challenges and approaches," *Proceedings of the IEEE*, vol. 88, no. 12, pp. 1855–1877, Dec 2000.

[97] J. Wu, C. Yuen, and J. Chen, "Leveraging the delay-friendliness of tcp with fec coding in real-time video communication," *IEEE Transactions on Communications*, vol. 63, no. 10, pp. 3584–3599, Oct 2015.

[98] Y. Xu, Y. Zhou, and D. M. Chiu, "Analytical QoE models for bit-rate switching in dynamic adaptive streaming systems," *IEEE Transactions on Mobile Computing*, vol. 13, no. 12, pp. 2734–2748, Dec 2014.

[99] Y. Xu, E. Altman, R. El-Azouzi, M. Haddad, S. Elayoubi, and T. Jimenez, "Analysis of buffer starvation with application to objective QoE optimization of streaming services," *IEEE Transactions on Multimedia*, vol. 16, no. 3, pp. 813–827, 2014.

[100] Y. Xu, S. Elayoubi, E. Altman, and R. El-Azouzi, "Impact of Flow-level Dynamics on QoE of Video Streaming in Wireless Networks," in *INFOCOM, 2013 Proceedings IEEE*, April 2013, pp. 2715–2723.

[101] J. Yao, S. S. Kanhere, I. Hossain, and M. Hassan, "Empirical evaluation of http adaptive streaming under vehicular mobility," in *Proceedings of the 10th International IFIP TC 6 Conference on Networking - Volume Part I*, ser. NETWORKING'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 92–105. [Online]. Available: http://dl.acm.org/citation.cfm?id=2008780.2008790

[102] Z. Ye, E.-A. Rachid, and T. Jimenez, "Analysis and modelling quality of experience of video streaming under time-varying bandwidth," in *2016 9th IFIP Wireless and Mobile Networking Conference (WMNC)*. IEEE, 2016, pp. 145–152.

[103] Y. Zhang, T. Yue, H. Wang, and A. Wei, "Predicting the quality of experience for internet video with fuzzy decision tree," in *Computational Science and Engineering (CSE), 2014 IEEE 17th International Conference on*, Dec 2014, pp. 1181–1187.

Analyse de performance des services de vidéo streaming dans les réseaux mobiles

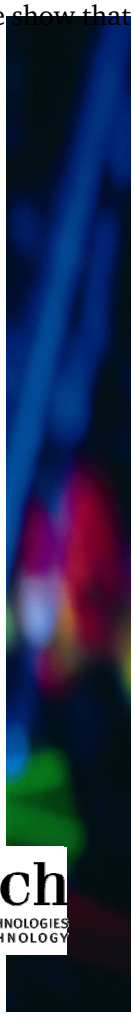
Yu-Ting LIN

RESUME : Le trafic de vidéo streaming étant en très forte augmentation dans les réseaux mobiles, il devient essentiel pour les opérateurs de tenir compte des spécificités de ce trafic pour bien dimensionner et configurer le réseau. Dans cette thèse, nous nous intéressons à la modélisation du trafic de vidéo streaming dans les réseaux mobiles. Pour le trafic de vidéo streaming en temps-réel, nous obtenons une forme analytique pour une métrique de qualité de service (QoS) importante, le taux de perte de paquets, et utilisons ce modèle à faire du dimensionnement. Pour le trafic de vidéo streaming de type HTTP adaptatif, nous proposons et analysons d'autres métriques de QoS comme le *bitrate* moyen, le taux de déficit vidéo et le surplus de *buffer*, afin de trouver le bon compromis entre résolution de la vidéo et fluidité de la diffusion vidéo. Nous étudions par simulation l'impact de quelque paramètres clés du systéme. Nous montrons que l'utilisation de segments de vidéo courts, d'un nombre réduit d'encodages vidéos et de l'ordonnancement de type *round robin* améliore la fluidité de la vidéo tout en diminuant sa résolution. Nous proposons par ailleurs d'adapter le nombre des segments téléchargés dans une requête HTTP de sorte que chaque requête corresponde au même volume de données. Enfin, nous appliquons les techniques de l'apprentissage automatique pour analyser la corrélation entre les caracteristiques du système et la qualité d'expérience (QoE) des utilisateurs.

MOTS-CLEFS: Streaming Vidéo, Streaming Temps Réel, Streaming Adaptatif, Qualité de l'Expérience, Segment Vidéo, Modèle Niveau Flow, L'intelligence Artificielle.

ABSTRACT: As the traffic of video streaming increases significantly in mobile networks, it is essential for operators to account for the features of this traffic when dimensioning and configuring the network. The focus of this thesis is on traffic models of video streaming in mobile networks. For real-time video streaming traffic, we derive an analytical form for an important Quality-of-Service (QoS) metric, the packet outage rate, and utilize the model for dimensioning. For HTTP adaptive video streaming traffic, we propose and evaluate other QoS metrics such as the mean video bit rate, the deficit rate and the buffer surplus, so as to find the good trade-off between video resolution and playback smoothness. We study by simulation the impacts of some key parameters of the system. We show that using smaller chunk durations, fewer video coding rates and round-robin scheduling may provide a smoother video playback but decrease the mean video resolution. We also propose to adapt the number of chunks downloaded in an HTTP request so that each HTTP request has the same data volume. Finally, we apply machine learning techniques to analyze the correlation between system characteristics and the quality of experience (QoE) of users.

KEY-WORDS: Streaming Video, Real-Time Streaming, Adaptive Streaming, Quality of Experience, Video Chunk, Flow-Level Model, Machine Learning.



TELECOM ParisTech

EDiTE

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY