



Titre: Admission Regions for DiffServ Network
Title:

Auteur: Ming-Zhang Zhu
Author:

Date: 2010

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Zhu, M.-Z. (2010). Admission Regions for DiffServ Network [Mémoire de maîtrise,
Citation: École Polytechnique de Montréal]. PolyPublie. <https://publications.polymtl.ca/353/>

 Document en libre accès dans PolyPublie
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/353/>
PolyPublie URL:

**Directeurs de
recherche:** Brunilde Sanso, & André Girard
Advisors:

Programme: Génie électrique
Program:

UNIVERSITÉ DE MONTRÉAL

ADMISSION REGIONS FOR DIFFSERV NETWORKS

MING-ZHANG ZHU

DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE ÉLECTRIQUE)

MARCH 2010

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

ADMISSION REGIONS FOR DIFFSERV NETWORKS

présenté par: ZHU Ming-Zhang

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. CARDINAL Christian, Ph.D., président

Mme SANSÒ Brunilde, Ph.D., membre et directrice de recherche

M. GIRARD André, Ph.D., membre et codirecteur de recherche

M. FRIGON Jean-François, Ph.D., membre

”Anyone who has never made a mistake has never tried anything new.”

Albert Einstein

ACKNOWLEDGMENTS

Here, I would like to thank my supervisor professor Brunilde Sanso and her colleague professor Andre Girard during my study, for the support, the consult, the availability and the patience. I would also thank my parents, family and friends, who encouraged me. Here is a list of students in Gerad, who provided me lot of useful information to help me during the study: Armelle Gnassou, Hamza Dahmouni, Cristina Marisol Ortiz De La Roche, Christian Awad and Zhigang Han. At the end, I would like to thank everyone, who ever helped me during this graduate study.

RÉSUMÉ

Dans le monde d'aujourd'hui, le concept de contrôle d'admission pour réseaux DiffServ a été appliqué sur la transmission de données en complément aux réseaux IP (IPv4, IPv6), aux réseaux sans fil (IEEE 802.11) et aux réseaux ATM. De plus en plus d'applications en temps réel s'introduisent aux réseaux internet chaque jour, et la demande de haute qualité de service (QoS) croît exponentiellement. Le contrôle d'admission joue un rôle très important pour garantir la QoS.

Le but de cette recherche est de décider rapidement d'accepter ou de rejeter un nouveau trafic à l'arrivée, donc de trouver une façon de faire le contrôle d'admission sur les réseaux avec DiffServ. Les recherches précédentes indiquent que l'approximation linéaire est suffisante surtout pour la région typique d'opération sans service différentiel (DiffServ). DiffServ garantit la QoS pour chaque application individuellement, selon la notion de priorité. Dans notre recherche, on applique la notion de la bande passante effective sur les réseaux IP qui a inclus DiffServ, pour étudier le contrôle d'admission sur réseaux avec DiffServ à l'aide de graphique sur région d'admission. La majorité de travail porte sur l'étude de la région d'admission à l'aide de simulations utilisant les logiciels de caractérisation de trafics.

En général, la première étape est de programmer un logiciel capable de simuler un lien sur réseaux IP avec DiffServ et de le valider à l'aide de tests, la deuxième étape est d'introduire 2 ou 3 types de trafics à ce lien, et de tracer la région d'admission. On peut varier les paramètres tels que le type de trafic, la priorité, la discipline de files d'attente, etc. La troisième étape est d'essayer d'appliquer l'approximation linéaire.

Pour tracer la région d'admission, il faut répéter une procédure. Pour chaque nombre de trafic (type 1), quel est le maximum nombre de trafic (type 2) en garantissant la QoS. L'ensemble de ces points va définir la région d'admission finale.

La majorité de ces résultats sont clairs et concluants. Les résultats suggèrent que l'approximation linéaire est suffisante et conservatoire pour le contrôle d'admission car les régions d'admission pour les réseaux avec DiffServ est souvent sous forme linéaire ou trapèze (toujours concave).

Entre les différentes disciplines de files d'attente, PQ (file à priorités) a une région d'admission plus grande que RR (round robin) quand les trafics de priorité supérieure ont besoin de QoS beaucoup plus hautes que les trafics de priorité inférieure. Si les deux sortes de trafics ont besoin de même QoS et ont de caractéristiques similaires, la région d'admission pour RR va être plus grande que PQ. WRR (weighted round robin) est toujours entre les courbes de PQ et de RR.

Pour les caractéristiques des trafics, la taille de la région d'admission est directement proportionnelle à la taille des paquets, le temps entre l'arrivée de deux paquets, la capacité de serveur et d'autre paramètres. Pour la majorité des critères de QoS, normalement moins de demande sur QoS augmente la taille de la région d'admission d'une manière non-proportionnelle, il est à noter que ça ne s'applique pas à la gigue (variation de délai d'attente). S'il y a peu de trafics dans le réseau, l'augmentation du trafic peut parfois améliorer la qualité de la gigue.

Le mode de non-préemptif n'a aucun effet sur la région d'admission sauf quand il y a beaucoup plus de trafics de priorités supérieures que le nombre de trafics de priorités inférieures. Dans ce cas extrême, la région d'admission est souvent en forme de trapèze.

À la fin, on peut visionner la région d'admission pour 3 types de trafics sur un graphique en 3 dimensions, ainsi, le contrôle d'admission peut être réalisé même s'il y a plus que 2 niveaux de priorités. Pour quatre types ou plus de type de trafics de données, on ne peut illustrer sous forme graphiques mais la représentation mathématiques est suffisante.

ABSTRACT

This research attempts to determine the possibility of finding "simple" approximated rule(s) to control new incoming traffic flows (accept/reject) while satisfying the Quality-of-Service of all traffic by means of connection admission control (CAC) under differential service (DiffServ) Networks. The study is done in statistical simulations based on a queuing process simulator called CAC_SIM. The current program is only able to simulate quality of service (QoS) of single link network with different traffic sources, capable of supporting DiffServ, and export data for admission region plots.

In general, for every possible number of one type of traffic source, we ask the question of what is the maximum number of a second type of traffic source that would still satisfy the QoS agreement (using CAC_SIM). By connecting these points, we obtain the admission region curve, and the admission region is the area under the curve.

The resulting admission regions are often linear or trapezoidal, and the linear approximations would be conservative. When the higher priority QoS agreements are better than the lower priority QoS agreements, priority queuing (PQ) has larger admission region than round robin (RR). If both have similar QoS agreements and similar traffic characteristics, RR would have larger admission region than PQ. Weighted round robin (WRR) always lies in between PQ and WRR.

For the traffic characteristics, the admission region size is directly proportional to the packet size, the packet inter-arrival time, the server capacity, etc. For the QoS agreements, looser QoS agreements would often increase the admission region size (not directly proportional). However, the jitter behaves differently when there are fewer number of traffic sources in the network. An increase of the number of traffic flows may improve the jitter performance.

In most the time, the non pre-emptive mode would not cause the lower priority packets to affect the higher priority QoS. However, when the higher priority traffic flows outnumber the lower priority traffic flows, the higher priority traffic QoS may become the binding factor(s), thus the admission region forms a trapeze.

At the end, the idea of CAC is extended to 3 different traffic types. The admission region area becomes the admission region space in 3-D.

CONDENSÉ EN FRANÇAIS

Dans les dernières années, de nombreuses applications en temps réel se sont introduites aux réseaux internet. Ces applications peuvent seulement fonctionner avec une haute qualité de service (QoS). En plus, différentes applications ont des besoins différents de QoS. Pour garantir ces QoS, le contrôle d'admission pour réseaux de service différentiels (DiffServ) a été appliqué sur la transmission de données en complément aux réseaux IP.

Le but de cette recherche est de permettre de décider rapidement d'accepter ou de rejeter un nouveau trafic à l'arrivée, c'est à dire de trouver une façon de faire le contrôle d'admission sur les réseaux avec DiffServ. Les recherches précédentes indiquent que l'approximation linéaire est suffisante pour la région typique d'opération sans considérer le service différentielle (DiffServ). Dans notre cas, nous voulons trouver une façon simple (en utilisant l'approximation linéaire si possible) de faire le contrôle d'admission sur les réseaux avec DiffServ. En complément au but principal, nous voulons aussi étudier comment les différents trafics ou files d'attente peuvent influencer sur la région d'admission.

I. Introduction

Ce condensé en français va commencer par les définitions des concepts et des termes de base. Après cela, nous allons décrire toutes les étapes principales de notre expérimentation par simulation. Ensuite, nous allons décrire et analyser les résultats significatifs. À la fin, nous allons faire un résumé de la recherche, en définir des applications possible et donner des recommandations pour les travaux futurs.

II. Les concepts et les termes

Dans cette section, nous allons définir les concepts et les termes importants. Le but est

d'aider les lecteurs à comprendre la base fondamentale de cette recherche.

Le contrôle d'admission est de permettre la prise de décision sur l'acceptation ou le rejet d'un nouveau trafic à l'arrivée basé sur certaines règles simples. Ici, nous utilisons la région d'admission pour prendre cette décision. Le trafic est accepté si le nombre de trafic est à l'intérieur de la région d'admission et il sera rejeté si ce n'est pas le cas.

Une région d'admission est basée sur la notion de la bande passante effective. Si EB est la bande passante effective, N_i est le nombre maximal de trafic en garantissant un certain niveau de QoS, et C est la capacité du serveur, ça donne équation:

$$EB_i = \frac{C}{N_i} \quad (1)$$

Lorsque deux types de trafics ($i = 1, 2$) sont rencontrés, cette équation peut-être montrée sur un graphique où l'on met en x le nombre de trafics de type 1 et en y le nombre de trafic de type 2, la zone sous la courbe sera la région d'admission (voir l'exemple dans figure 1.1). Dans le cas où 3 types de trafic sont rencontrés, une surface avec les axes x , y et z formera un volume qui définit la région d'admission.

Le service différentiel (DiffServ) garantit la QoS pour chaque application individuelle, selon la notion de priorité. En général, EF (expected forwarding) est en priorité la plus haute, BE (best effort) est en priorité la plus basse et AF (assured forwarding) est en milieu.

Il y a plusieurs différentes disciplines de file d'attente avec DiffServ. Dans notre recherche, nous allons utiliser PQ (priority queuing), RR (rounded robin) et WRR (weighted rounded robin). Pour PQ, les paquets de priorités inférieures peuvent être servis seulement s'il ne reste plus de paquet de priorité supérieure. Pour RR, dans un cycle, chaque file d'attente d'une même priorité va envoyer un paquet au serveur. Pour envoyer le deuxième paquet,

il faudra attendre le prochain cycle. Pour WRR, chaque file d'attente d'une priorité a son propre poids (un nombre entier). Dans chaque cycle, un file d'attente peut envoyer un nombre de paquets proportionnels (ou moins) à ce poids.

Il y a un mode non préemptif qui permet au serveur de continuer à servir un paquet de priorité inférieur qui a déjà commencé sa transmission quand un paquet de priorité supérieur arrive. Notez que toutes les simulations de cette recherche utilisent le mode non préemptif car le mode préemptif peut causer un nombre massif de perte des paquets de priorités inférieures.

A ce point, nous avons décrit la notion de bande passante effective, le contrôle de connexion d'admission, la région d'admission, le service différentiel et les disciplines de file d'attente. La prochaine section va décrire tous les étapes principales de cette recherche basée sur les concepts et les termes que nous avons définis.

III. Les étapes de la recherche

Dans cette section, nous allons discuter des étapes principales requises pour obtenir les résultats de simulation. Pour étudier le contrôle d'admission sur réseaux avec DiffServ à l'aide de graphique de région d'admission, la première étape est de créer un logiciel qui simulera les transmissions de paquets dans un lien sur réseaux IP avec DiffServ et de valider les résultats, la deuxième étape est d'introduire 2 ou 3 types de trafics à ce lien et de tracer la région d'admission. La troisième étape est d'essayer d'appliquer l'approximation linéaire.

Le logiciel utilisé pour simuler les transmissions de paquets est programmé en Java à l'aide d'une bibliothèque mathématique qui s'appelle Stochastic Simulation in Java (SSJ), ce logiciel peut classer les différents trafics selon leur priorités, leurs files d'attentes respective, et différents types de générateurs de trafic (voir figure 2.1). Dans cette

recherche, un seul serveur est utilisé comme point de sortie.

Pour simuler des paquets dans un lien de réseaux IP, les générateurs de trafic vont générer des flot de paquets et ils seront ensuite envoyé au programme de classification qui va les classer selon le type d'application des paquets, leur assigner une priorité et l'envoyer aux files d'attente correspondantes. Si le serveur est occupé ou s'il y a déjà des paquets dans la file d'attente, le paquet peut y rester pour une durée indéterminée. Dès que le serveur est libre, il va sortir un paquet à la fois d'un file d'attente selon la discipline de file d'attente (PQ, RR ou WRR).

Cette simulation d'une session peut contenir plus qu'un générateur de trafic ou plusieurs différents types de générateurs de trafics. Pendant la simulation, l'ordinateur peut enregistrer les données de qualité de service (QoS). Donc, nous sommes capables de répondre à une question importante: Pour n_1 nombre de trafic (type 1), n_2 nombre de trafic (type 2), n_3 nombre de trafic (type 3), est-ce que les QoS sont acceptables? ($n_1, n_2, n_3...$) est une configuration.

Pour chaque nombre de trafic (type 1), quels sera le nombre de trafic maximum de (type 2) en garantissant le QoS. Pour tracer la région d'admission, il faut répéter une cette question plusieurs fois. L'ensemble de ces points va définir les frontières de la région d'admission finale.

Prendre en note que pour chaque configuration, il faut répéter la simulation d'une session plusieurs fois afin d'obtenir un résultat convainquant. Dans nos simulation, nous utilisons l'analyse statistique séquentielle pour déterminer combien de répétitions sont nécessaire.

Avant d'appliquer les approximations sur la région d'admission, nous allons étudier les facteurs qui influencent la région d'admission. Nous pouvons varier les paramètres (un

à la fois) tels que le type de trafic, la priorité, la discipline de files d'attente, etc.

À la fin, nous allons essayer d'appliquer l'approximation linéaire aux différentes régions d'admission obtenues dans les simulations. Nous allons aussi essayer d'appliquer d'autre façon d'approximation si possible.

A ce point, nous avons parlés de toutes les étapes majeures de l'expérimentation. Dans la prochaine section, nous allons montrer les résultats principaux et les analyser.

IV. Les résultats et les analyses

Dans cette section, nous allons discuter les résultats principaux. Les résultats suggèrent que l'approximation linéaire est suffisante et conservatoire pour le contrôle d'admission pour les réseaux avec DiffServ. Donc, c'est possible de faire le contrôle d'admission pour les réseaux avec DiffServ. A part de ça, nous avons étudié les régions d'admission en détails et on à trouvés les résultats intéressants. Les résultats sont reliés aux différents facteurs qui influencent la région d'admission.

En général, les régions d'admission pour les réseaux avec DiffServ sont souvent sous forme linéaire, trapèze (toujours convexe) ou en zigzag. La forme zigzag est composée par 3 parties: une ligne linéaire décroissante, une ligne horizontale et une autre ligne linéaire décroissante (voir chapitre 7).

Dans le cas de forme linéaire, il y a seulement la QoS qui est le facteur dominant (souvent une QoS de priorité inférieure). Dans le cas de forme trapèze, il y a 2 QoS qui sont les facteurs dominants. Dans la partie à gauche, le facteur dominant est souvent une QoS de priorité supérieure. Dans la partie à droite, le facteur dominant est souvent une QoS de priorité inférieure. Le cas de forme zigzag apparaissent seulement pour les disciplines de file d'attente de type WRR ou RR, ça ressemble à la forme trapèze, sauf qu'il y a une

zone grise dont les 2 QoS de priorité supérieure et inférieure sont facteurs dominants.

Pour les différentes disciplines de files d'attente, PQ (file à priorités) a une région d'admission plus grande que RR (round robin) quand les trafics de priorité supérieure ont besoin de QoS beaucoup plus hautes que les trafics de priorité inférieure. Si les deux sortes de trafics ont besoin de la même QoS et ont de caractéristiques similaires, la région d'admission pour RR va être plus grande que PQ. WRR est toujours entre le milieu de PQ and RR (voir chapitre 4).

Pour les caractéristiques des trafics, la taille de la région d'admission est directement proportionnelle au débit (throughput ou bitrate) qui est directement proportionnelle à la taille de paquet, le temps entre deux paquets d'arrivés, la capacité de serveur, etc. C'est pourquoi les résultats aussi montrent que la taille de la région d'admission est directement proportionnelle à la taille de paquet, le temps entre deux paquets d'arrivés, la capacité de serveur, etc. Les détails sont dans le chapitre 5.

Pour les QoS en générale, moins de demande sur QoS augmente la taille de la région d'admission (pas directement proportionnelle), sauf la gigue (voir chapitre 5). S'il y a peu de trafics dans le réseau, l'augmentation du nombre de trafics peut améliorer la qualité de la gigue (voir chapitre 6).

Le mode de non préemptif n'a aucun effet sur la région d'admission sauf quand il y a beaucoup plus de trafics de priorités supérieures que le nombre de trafics de priorités inférieures. Dans ce cas extrême, la région d'admission est souvent en forme de trapèze.

À la fin, on visionne la région d'admission pour 3 types de trafics sur 3-D. Donc le contrôle d'admission peut être réalisé même s'il y a plus que 2 niveaux de priorités. La projection sur 2-D (surface en 2 axes) est ressemblé à la région d'admission de ces 2 types de trafic (voir chapitre 8).

Dans cette section, nous avons montrés qu’il est possible de faire le contrôle d’admission dans les réseaux avec DiffServ. Nous avons aussi montrés comment différents facteurs peuvent influencer la région d’admission.

V. Conclusions

Dans notre recherche, nous avons déterminés la possibilité de faire le contrôle d’admission dans les réseaux avec DiffServ, il est aussi possible de le faire en utilisant l’approximation linéaire. En autres mots, un routeur avec DiffServ peut prendre une décision rapide en acceptant ou rejetant un nouveau trafic pour maintenir les QoS de tous les trafics en service basé sur l’approximation linéaire de la région d’admission.

Nous avons aussi montrés comment différents facteurs peuvent influencer la région d’admission. Un jour, ça pourra aider à trouver une formule analytique pour la bande passante effective basée sur les données mesurables.

Toutes les simulations dans cette recherche sont basées sur des réseaux IP, sans considérer la couche MAC ou la couche physique. Dans le futur, nous pouvons vérifier s’il y a des changements en considérant ces 2 couches. Nous pourrons aussi simuler un réseau plus complexe au lieu d’un seul lien.

TABLE OF CONTENTS

DEDICATION	iii
ACKNOWLEDGMENTS	iv
RÉSUMÉ	v
ABSTRACT	vii
CONDENSÉ EN FRANÇAIS	ix
TABLE OF CONTENTS	xvi
LIST OF FIGURES	xxi
LIST OF NOTATIONS AND SYMBOLS	xxv
LIST OF TABLES	xxvii
INTRODUCTION	1
CHAPTER 1 BACKGROUND	4
1.1 Connection Admission Control (CAC)	4
1.1.1 Effective Bandwidth (EB)	5
1.1.2 Admission Region (AR)	7
1.2 Differential Service (DiffServ)	8
1.2.1 Queuing Disciplines	10
1.2.1.1 Virtual Time Concept (VTC)	12
1.2.1.2 Priority Queuing (PQ)	13
1.2.1.3 Fair Queuing (FQ)	14
1.2.1.4 Round Robin (RR)	15
1.2.1.5 Weighted Round Robin (WRR)	16

1.2.1.6	Weighted Interleaving Round Robin (WIRR)	16
1.2.1.7	Other Disciplines and Policies	17
1.2.2	Queuing Policies	17
1.2.2.1	Service Order	17
1.2.2.2	Pre-emptive Mode	17
1.2.2.3	Packet-Drop Policy	18
1.2.2.4	Buffering	19
1.2.3	Differential Service Code Point(DSCP)	19
1.3	Quality of Service Control	20
1.3.1	Delay	20
1.3.2	Jitter	21
1.3.2.1	Definition of Jitter from ITU-T	22
1.3.2.2	Definition of Jitter from IETF	22
1.3.3	Loss Ratio	24
1.3.4	Service Level Agreement	26
1.4	Previous Research	27
1.4.1	Previous Research on Effective Bandwidth	27
1.4.2	Previous Research on DiffServ	28
1.4.3	CAC for PQ Guarantee Delay QoS	30
1.5	Conclusion	31
CHAPTER 2	SIMULATION MODEL	32
2.1	Stochastic Simulation in Java (SSJ)	33
2.2	General Idea of the Simulation	33
2.3	Structure of the CAC_SIM Model	33
2.3.1	Classifier	34
2.3.2	Output Port	36
2.3.2.1	Scheduler	36
2.3.2.2	Queue	37
2.3.2.3	Server	39

2.3.2.4	Statistical Collections	40
2.3.3	Source Generators	42
2.4	CAC_SIM Simulation Process	43
2.5	Simulation Techniques	44
2.5.1	General Techniques	44
2.5.2	Avoiding In Phase Source Generator Problem	44
2.5.3	Acceptance Procedure	46
2.5.4	Discrete Admission Region	48
2.5.5	Precision	49
2.6	Validation of Output	51
2.6.1	Confidence Interval Vs. (Number of Repetition)(Simulation Length)	51
2.6.2	Queuing Simulation Vs. Theoretical Values	52
2.6.2.1	M/G/1 Single Queue	54
2.7	Conclusion	55
CHAPTER 3	TRAFFIC SOURCE GENERATORS	56
3.1	Overview of Traffic Sources	56
3.1.1	Audio Traffic Source	58
3.1.1.1	Audio Codecs	58
3.1.1.2	QoS for Audio Traffic Source	59
3.1.2	Video Traffic Source	59
3.1.2.1	Video Codecs	60
3.1.2.2	Video Frames and Scene	60
3.1.3	Data Transmission Traffic Source	62
3.2	Sources Generators	62
3.2.1	Continuous Traffic Source Generator	63
3.2.2	ON-OFF Traffic Generator	64
3.2.3	Video Traffic Source Generator	65
3.2.3.1	Parameters for Video Traffic Sources from Trace Files	66
3.2.3.2	Lognormal Distribution and Validation Test	67

3.2.3.3	Emulation of Video Traffic Source with Lognormal Distribution	68
3.3	Conclusion	69
CHAPTER 4	ADMISSION REGIONS WITH DIFFSERV	71
4.1	General Parameters and Simulation Procedure	71
4.1.1	Parameters for Each Type of Traffic	71
4.1.2	Queuing Disciplines and Queuing Policies	72
4.1.3	Simulation Procedure Settings	73
4.1.4	Excluding the Configuration with Zero Lower Priority Traffic Flow	73
4.2	Admission Region for VoIP and CBR Data Traffic Sources	75
4.2.1	Admission Region for VoIP and CBR Data Traffic Sources under PQ	75
4.2.2	Admission Region for VoIP and CBR Data Traffic Sources under RR	77
4.2.2.1	Admission Region for VoIP and Audio Traffic Sources	78
4.2.3	Admission Region for VoIP and CBR Data Traffic Sources under WRR	79
4.3	Admission Region for Video and CBR Data Traffic Sources	81
4.4	Admission Region for Data Command File and CBR Data Sessions . .	82
4.5	Conclusion	84
CHAPTER 5	ADMISSION REGION OF DIFFERENT TRAFFIC SOURCES UNDER PRIORITY QUEUEING	85
5.1	General Parameters and Simulation Procedure	85
5.2	CBR Data as the Lower Priority Traffic Source	85
5.2.1	Admission Region for VoIP and Data Traffics	86
5.2.2	Admission Region for LiveTV and Data Transmission	89
5.2.3	Admission Region for Data Command/Control and Data Transmission	92

5.3	VoIP as Highest Priority Traffic	93
5.3.1	Admission Region for VoIP and Other Audio Traffics	93
5.3.2	Admission Region for VoIP and LiveTV	96
5.3.2.1	Admission Region for VoIP and LiveTV (Reverse Priority)	99
5.4	Conclusion	100
CHAPTER 6	INFLUENCE OF JITTER ON THE ADMISSION REGION .	102
6.1	Positive Slope in Admission Region	102
6.2	Causes of the Positive Slope in the Admission Region	103
6.3	Conclusion	106
CHAPTER 7	LINEAR APPROXIMATION FOR ADMISSION REGION .	107
7.1	Linear Approximation for Linear Curves	107
7.2	Other Approximations	110
7.3	Conclusion	111
CHAPTER 8	ADMISSION REGION FOR 3 TRAFFIC TYPES	112
8.1	Admission Region for 3 Traffic Types under Priority Queuing (PQ) . .	112
8.2	Admission Region for 3 Traffic Types under Weighted Round Robin (WRR)	115
8.3	Conclusion	116
CONCLUSION		118
REFERENCES		121

LIST OF FIGURES

Figure 1.1	Illustration of Admission Region	8
Figure 1.2	Illustration of Single Queue System	9
Figure 1.3	Illustration of Multi-Queue System	10
Figure 1.4	Illustration of Fair Fluid Queuing Model	11
Figure 1.5	Admission Region due to Delay for Priority Queuing ATM Network	30
Figure 2.1	Illustration of the output port from CAC_SIM model	34
Figure 2.2	Scheduler Process	43
Figure 2.3	Scheduler Process without QoS Detection	44
Figure 2.4	Illustration of In Phase Problem	45
Figure 2.5	Illustration of Solution to In Phase Problem	46
Figure 2.6	Acceptance Procedure Illustration: It is based on the loss of traffic type 1, where p^- and p^+ being upper and lower bounds respectively.	48
Figure 2.7	Finding the Maximum Value of n_2 (Number of Type 2 Traffic Source) for a Fixed Value of n_1 (Number of Type 1 Traffic Source)	50
Figure 2.8	Delay for MM1 Queuing System	52
Figure 2.9	% Error and Variance for M/M/1 Queuing System	53
Figure 3.1	Audio ON-OFF Traffic Source Generator: x-axis is the time t , y-axis is the real amount of packets being sent divided by the maximum amount of packets possibly being generated if there is no OFF period between time 0 and time t	65
Figure 3.2	Frame I Size Distribution: Generator Outputs (curve) and Trace File (histogram)	69
Figure 3.3	Frame B Size Distribution: Generator Outputs (curve) and Trace File (histogram)	70

Figure 3.4	Frame P Size Distribution: Generator Outputs (curve) and Trace File (histogram)	70
Figure 4.1	Admission Regions (Including Zero Lower Priority Traffic Source): LiveTv-AF11 (StarWars IV MPEG-4 low quality) vs Data-AF41 (1480B Payload) for Different QoS under PQ	74
Figure 4.2	Admission Regions: VoIP-EF (G.711, 160B Payload) vs Data-AF41 (1480B Payload) for Different Queuing Disciplines	75
Figure 4.3	Admission Regions: VoIP-EF (G.711, 160B Payload) vs Audio-AF11 (G.711, 160B Payload) for Different PQ and RR	78
Figure 4.4	Overall Loss Ratios for Both VoIP-EF and Data-AF41 by Adding 1 Extra VoIP-EF Flow to the Curve WRR Weight 8:1 from Figure 4.2	79
Figure 4.5	Admission Regions: LiveTv-AF11 (StarWars IV MPEG-4 low quality) vs Data-AF41 (1480B Payload) for Different Queuing Disciplines	81
Figure 4.6	Admission Regions: Command-AF11 (Command/Control Signals) vs Data-AF41 (1480B Payload) for Different Queuing Disciplines	82
Figure 5.1	Admission Regions: VoIP-EF (G.711, 160B Payload) vs Data-AF41 (64Kbps) for Different CBR Data Packet Size	86
Figure 5.2	Admission Regions: VoIP-EF (G.711, 160B Payload) vs Data-AF41 (1480B Payload) for Different CBR Data Throughput	88
Figure 5.3	Admission Regions: VoIP-EF (G.711, 160B Payload) vs Data-AF41 (64Kbps, 1480B Payload) for Different Server Capacity	88
Figure 5.4	Admission Regions: LiveTV-AF11 (Star Wars IV Low Quality) vs Data-AF41 (64Kbps) for Different CBR Data Packet Size	89
Figure 5.5	Admission Regions: LiveTV-AF11 (Star Wars IV) vs Data-AF41 (1480B) for Doubling Server Capacity and Data-AF41 Throughput)	90

Figure 5.6	Admission Regions: LiveTV-AF11 (Star Wars IV) vs Data-AF41 (128Kbps) for Different Video Qualities, Server Capacity Becomes 4Mbps	91
Figure 5.7	Admission Regions: Command-AF11 (Command/Control Signals) vs Data-AF41 (64Kbps) for Different Data Packet Sizes	92
Figure 5.8	Admission Regions: VoIP-EF (G.711, 160B Payload) vs Audio-AF11 (G.711) for Different Audio-AF11 Payload Size	93
Figure 5.9	Admission Regions: VoIP-EF (G.711) vs Audio-AF11 (G.711, 160B Payload) for Different VoIP-EF Payload Size	94
Figure 5.10	Admission Regions: VoIP-EF (G.711) vs Audio-AF11 (G.711) for Varying Both VoIP-EF and Audio-AF11 Payload Size	95
Figure 5.11	Admission Regions: VoIP-EF (G.711, 160B Payload) vs Audio-AF11 (G.711, 160B Payload) for Different Audio-AF11 Delay Service Level Agreement	96
Figure 5.12	Admission Regions: VoIP-EF (G.711, 160B Payload) vs LiveTV-AF11 (StarWars IV Low Quality) for Different LiveTV-AF11 Delay Service Level Agreement	97
Figure 5.13	Admission Regions: VoIP-EF (G.711, 160B Payload) vs LiveTV-AF11 (StarWars IV Low Quality) for Different LiveTV-AF11 Delay Service Level Agreement	98
Figure 5.14	Admission Regions: LiveTV-EF (StarWars IV Low Quality) vs Audio-AF11 (G.711, 160B Payload)	99
Figure 6.1	Loss due to Delay of Audio-AF41 (low priority)	104
Figure 6.2	Loss due to Delay of Audio-AF41 (low priority)	104
Figure 7.1	Linear Approximation for the Admission Region: LiveTV-AF11 (StarWars IV MPEG-4 low quality) vs Data-AF41 (64Kbps, 1480B Payload) under PQ	108
Figure 7.2	Linear Approximation for the Admission Region: VoIP-EF (G.711, 160B Payload) vs Data-AF4 (64Kbps, 1480B Payload) under PQ	108

Figure 7.3	Linear Approximation for the Admission Region: VoIP-EF (G.711, 160B Payload) vs Data-AF4 (64Kbps, 1480B Payload) under WRR (8:1 Weight Ratio)	109
Figure 7.4	Trapeze Approximation for the Admission Region: VoIP-EF (G.711, 160B Payload) vs Data-AF4 (64Kbps, 1480B Payload) under PQ	110
Figure 7.5	Step Approximation for the Admission Region: VoIP-EF (G.711, 160B Payload) vs Data-AF4 (64Kbps, 1480B Payload) under WRR (8:1 Weight Ratio)	111
Figure 8.1	Admission Regions: VoIP-EF (G.711, 160B Payload) vs Data-AF41 (1480B Payload) vs LiveTV-AF11 (StarWars IV Low Quality) under Priority Queuing	113
Figure 8.2	Admission Regions: VoIP-EF (G.711, 160B Payload) vs Data-AF41 (1480B Payload) for Different Number of LiveTV-AF11 Traffics under Priority Queuing	113
Figure 8.3	Admission Regions: VoIP-EF (G.711, 160B Payload) vs LiveTV-AF11 (StarWars IV Low Quality) for Different Number of Data-AF41 Traffics under Priority Queuing	114
Figure 8.4	Admission Regions: VoIP-EF (G.711, 160B Payload) vs Data-AF41 (1480B Payload) for Different Number of LiveTV-AF11 Traffics under WRR (weight ratio 4 : 2 : 1)	116

LIST OF NOTATIONS AND SYMBOLS

<i>AC</i> :	Admission Control
<i>AR</i> :	Admission Region
<i>ATM</i> :	Asynchronous Transfer Module
<i>C</i> :	Line Capacity
<i>CAC</i> :	Connection Admission Control
<i>CBR</i> :	Constant Bit Rate
<i>DiffServ</i> :	Differential Service
<i>EB</i> :	Effective Bandwidth
<i>FIFO</i> :	First-In-First-Out
<i>FIFS</i> :	First-In-First-Serve
<i>FQ</i> :	Fair Queuing
<i>GOS</i> :	Group of Pictures
<i>GPS</i> :	Generalized Processor Sharing
<i>h</i> :	Bin Width for an Histogram
<i>HTTP</i> :	Hypertext Transfer Protocol
<i>IEEE</i> :	Institute of Electrical and Electronics Engineers
<i>IEFT</i> :	Internet Engineering Task Force
<i>IP</i> :	Internet Protocol
<i>IPDV</i> :	Internet Packet Delay Variation
<i>ISP</i> :	Internet Service Provider
<i>ITU</i> :	International Telecommunication Union
<i>ITU-T</i> :	Telecommunication Standardization Sector from ITU
<i>LIFO</i> :	Last-In-First-Out
<i>LiveTV</i> :	Live Television
<i>MPEG</i> :	Motion Picture Experts Group
<i>pdf</i> :	Probability Density Function
<i>PDV</i> :	Packet Delay Variation

<i>PQ</i> :	Priority Queuing
<i>QoS</i> :	Quality of Service
<i>NaN</i> :	Not a Number
<i>RBD</i> :	Rate-Based Drop
<i>RFC</i> :	Requests for Comment
<i>RR</i> :	Round Robin
<i>s</i> :	Space
<i>SIP</i> :	Signal Initiation Protocol
<i>SLA</i> :	Service Level Agreement
<i>t</i> :	Time
<i>TTL</i> :	Time to Live
<i>UTC</i> :	Virtual Time Concept
<i>VoIP</i> :	Voice over IP
<i>VoD</i> :	Video on Demand
<i>VO</i> :	Video Object
<i>VOP</i> :	Video Object Plane
<i>WIRR</i> :	Weighted Interleaving Round Robin
<i>WRR</i> :	Weighted Round Robin
$\alpha(s, t)$:	See <i>EB</i>
λ :	Packet Arrival Rate
μ :	Packet Service Rate

LIST OF TABLES

Table 1.1	DSCP for AF PHB	20
Table 1.2	Example of QoS Control Factors for Some Traffic Sources . . .	26
Table 3.1	Example of QoS Control Factors for Some Traffic Sources (Ap- plications)	57
Table 3.2	Size and Variance of Frames I, B and P for Different Video Traf- fic Sources (e.g., Size of Frame I = $E(I)$, Variance of Frame I Size = $VAR(I)$)	66
Table 5.1	Size of Frames I, B and P for Star Wars IV (e.g., Size of Frame I = $E(I)$) and the Overall Scene Size for IBBPBBPBBPBB Sequence	91

INTRODUCTION

Nowadays, connection admission control (CAC) concept has been widely used in ATM Network. The concept of CAC has extended its usage to IEEE 802.11 Wireless Network and IP networks (IPv4 and lately IPv6), wherever the communication is connection oriented [1, 2]. The demand on the high Quality-of-Service (QoS) performance for real time applications is growing rapidly. In order to guarantee the QoS of those new applications, CAC mechanisms play an important role. For fast accept/reject decision making on new incoming traffic, the notion of admission region (AR) has been heavily used. Previous studies show that for networks without differential service (DiffServ), linear approximation is sufficient to make admission control decisions. In this thesis, CAC is applied to differential service (DiffServ) IP networks in order to verify if the statement about linear approximation holds as well for these types of networks.

Objective and Solution

The objective of this thesis is to find the admission region for different types of traffic source operating under DiffServ IP networks, and study the QoS behavior of these traffic. Yet, although many studies have been done in CAC and DiffServ; very few publications have taken both of them into consideration.

In this thesis, we propose a statistical simulation approach to study the impact of DiffServ on CAC. From the simulation results, we will determine whether the admission region is linear or not, and whether a linear approximation exists or not. We will also study the shape of the admission regions and related inputs (e.g., properties of each traffic source and the network) and outputs (e.g., delay, jitter and loss of each traffic type) from the simulations, and try to learn the correlations between them.

Organization of the Thesis

This thesis is organized as follows. Chapter 1 clarifies some basic terminologies and concepts for CAC, DiffServ, QoS and some previous research.

Chapter 2 has a full description of the simulation model (CAC_SIM queuing model). The structure section will list all the components of the model and their properties/abilities. Later, there will be a detailed description on the queuing process, followed by the different techniques used in the simulations to help getting more reliable results in less time. The end of the chapter presents some validation tests.

Chapter 3 illustrates all the traffic generators, along with some validation tests. There are mainly 2 parts. Part 1 discusses all types of traffic sources (applications). Part 2 discusses all the generators used to emulate these traffic sources.

Chapter 4 mainly shows how different queuing disciplines affect the admission region of different traffic sources. The binding factors (dominant QoS factor) play important rule on shaping the admission regions for different queuing disciplines. In general, the weighted round robin admission curve stays in between the priority queuing (PQ) curve and the round robin (RR) curve.

Chapter 5 illustrates how the traffic source properties (such as packet size) and QoS boundaries (such as upper bound of delay constraint) would affect the admission region under priority queuing. In general, looser QoS constraint, larger packet size, greater throughput and greater server capacity would result larger admission region size.

Chapter 6 attempts to explain why does jitter behave differently from other QoS (results shown in chapter 5). In general, the queuing system acts as a jitter buffer, which would convert some jitter problem into extra delay. This would explain why sometimes having more traffic flows results in lower jitter and higher delay.

Chapter 7 explains the way of doing admission region approximation. The simplest

method is the linear approximation.

Chapter 8 shows some results with 3 different types of traffic source in a network. The resulting admission region would be in 3-D or 2-D projection.

Conclusion will list all the important results and explore some ideas that could be tested someday in the future.

Contributions

In this research, we can show that the admission control can be used in DiffServ network. We can also show that it is possible to be approximated by linear approximations in the admission region plot.

We also study how different traffic flow characteristics, QoS constraints, queuing processes would affect admission regions. We try to generalize some rules though the observations. We learn that changes in binding factor change the slop in admission region; WRR (weighted round robin) curves are always between PQ (priority queuing) and RR (rounded robin) curves; PQ does not always have larger admission region size than RR size; zero higher priority traffic flow may trigger huge increase in lower priority admission; most traffic flow characteristics are directly proportional to admission region size; switching priorities may affect admission region shape and size; jitter behaves different from other QoS criteria. From these findings, hope one day we can obtain an analytical solution.

CHAPTER 1

BACKGROUND

In order to complete the ultimate goal of finding the admission region of different traffic sources in differential service, some terminologies must be well defined. This chapter includes a brief description of Connection Admission Control (CAC) and Differential Service (DiffServ) in section 1.1 and 1.2, followed by a short explanation on Quality-of-Servive (QoS) control. The end of the chapter presents some prior research.

1.1 Connection Admission Control (CAC)

Admission control is a network QoS procedure that consists on deciding whether or not accept a new incoming traffic at call level (establish a new call or drop the call). There are many levels: electromagnetic wave level, bit level, packet level, frame level, call level, application level, etc. At the call level, a new network application requires to establish a connection, which is well known as the connection admission control (CAC) in ATM and 802.11 networks [2]. However, CAC can be implemented where the transmission is connection oriented (e.g. ATM, TCP, etc.). Most of the time, it happens in between the edge and core nodes of a network.

In a connection oriented network, CAC allows a new application initiation only if current network status, including the new admission, can support the minimum QoS requirement for the application. Thus, CAC algorithm has to compute the QoS parameters for all session types before it can accept a new session. This is too hard and the notion of effective bandwidth is used instead (see the section 1.1.1). CAC makes the accept/reject decision based on the characteristics of the incoming traffic and the network bandwidth usage status at that moment.

For a configuration (or a set) $(n_1, n_2, \dots, n_i)$ being authorized, where n_i is the number of type i traffic source, all types of traffic must satisfy their own specific QoS requirements.

The main idea of CAC is to avoid a new incoming traffic causing the QoS of all current active applications from deteriorating, or worse becoming unacceptable. It is similar to integrated service (IntServ) idea, except IntServ using reservation of line capacity (or reserve a fixed/known bandwidth). In the case of CAC, it would have to estimate the Effective Bandwidth of a traffic source. Ideally, if a new incoming source requires an effective bandwidth greater than current available bandwidth, then this new incoming traffic should be rejected. However, the computation of effective bandwidth is too complicated. Thus, an approximation (linear approximation in this thesis) using Admission Region is used. Both Effective Bandwidth and Admission Region will be explained in the sections 1.1.1 and 1.1.2.

1.1.1 Effective Bandwidth (EB)

Network resource is expressed in terms of bandwidth. The effective bandwidth (EB) of an application (or a traffic source) is the bandwidth required in order to have a satisfactory QoS for that application.

Effective bandwidth of a source i (EB_i) is defined based on 2 free parameters, which are space s and time scale t , both positive and finite. For the amount of work $X[0, t]$ (in our case the size of new arrived packets of the arrival process in terms of bits or bytes) that arrives from a source in the interval of time $[0, t]$, Kelly defined the effective bandwidth $\alpha(s, t)$ of a source into a mathematical form (characteristic function) [3]:

$$\alpha(s, t) = \frac{1}{st} \log E[e^{sX[0, t]}] \quad (1.1)$$

From this form of effective bandwidth definition, Kelly has found 4 properties for the

effective bandwidth: time independence, $\alpha(s, t) = f(st)$, additivity and $\alpha(s, t) = g(E[X], VAR[X]) + h(s)$, where $f(\cdot)$, $g(\cdot)$ and $h(\cdot)$ are just some functions, and $E[X]$ and $VAR[X]$ are the expected value and variance of X . The details can be found in Kelly's notes on effective bandwidth [3].

The reason why introducing Kelly's definition in this research is because of his 3rd property, additivity, which says: If $X[0, t] = \sum_i X_i[0, t]$, where $X_i[0, t]$ are independent, then

$$\alpha(s, t) = \sum_i \alpha_i(s, t) \quad (1.2)$$

This shows the property of additivity from the effective bandwidth, which is very useful through out the research. Since different traffic sources are generated independently, the overall effective bandwidth would be the summation of effective bandwidths for all current active traffic sources.

In practice, EB of application type i can be approximated from the bandwidth capacity C of a server and the maximum allowable number $N_i = N^*$ (N^* is the test value) of such application with required QoS. Eq.1.3 is the simpler version of effective bandwidth definition, which can be used in the simulations due to its simplicity.

$$EB_i = \frac{C}{N_i} \quad (1.3)$$

This N_i is the largest number of connections (flows) allowed within the QoS constraints (or boundaries).

1.1.2 Admission Region (AR)

Recall N_i as the maximum number of traffic type i allowed simultaneously on a shared transmission line with a bandwidth capacity C . If there are 2 traffic types, then an admission region graph can be plotted, where the admission region is the area under the curve. The curve is called admission region curve, which is connected through dots $(n_1, n_2) = (n_1, N_2)$. n_1 and n_2 are the number of traffic flows of type 1 and 2 respectively. N_2 is the maximum number of type 2 traffic flows allowed. $N_2 = f(n_1)$ is a function of n_1 , where n_1 can be varied from 0 to N_1 (max value of n_1). Any configuration (dot) (n_1, n_2) appeared above that curve would be rejected due to failed QoS test. Any configuration under the curve are acceptable. The dots on the curve are also acceptable.

Due to limited amount of resources, normally the rule of thumb is that the more type 1 traffic flows is being accepted, the more type 2 traffic flows would be rejected, and vice versa. The idea continues with 3 or more types of traffic, but the admission region would be 3-D space instead of 2-D area. However, sometimes there are exceptions. For example, when jitter becomes the dominant (bending) factor, the rule of thumb may not hold. (See details in later result sections.)

From equation 1.3, we have

$$C = EB_i N_i \quad (1.4)$$

Due to the 3rd effective bandwidth property mentioned above, if more than one type of traffic is present, we obtain

$$C = \sum_i EB_i N_i \quad (1.5)$$

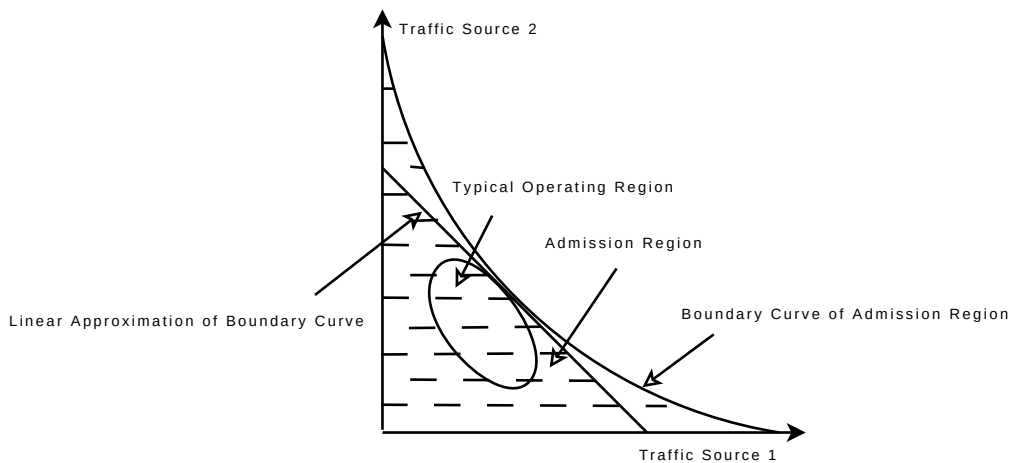


Figure 1.1 Illustration of Admission Region

The admission connection must make decision really fast due the massive amount of traffic flows. Thus simple solutions (less complexity) would be preferred. The solution could be a look-up table (LUT) for the admission region, or better an equation (preferable linear approximation for simplicity) approximating the admission region at its most used area.

Figure 1.1 illustrates an admission region plot for 2 types of traffic. Traffic source 1 and 2 are n_1 and n_2 respectively. The admission region is the area under the curve. Notice that there is often a typical operating region. Thus, even though the admission region curve is not always linear, a linear approximation can still be applied to that typical operating region.

1.2 Differential Service (DiffServ)

There are 2 common ways to guarantee networking services: integrated service (IntServ) and differential service (DiffServ). While integrated service uses resource reservation protocol (RSVP) described in RFC 2205 to reserve bandwidth for a connection, differential service guarantees different traffic sources with different service level agreements [4].

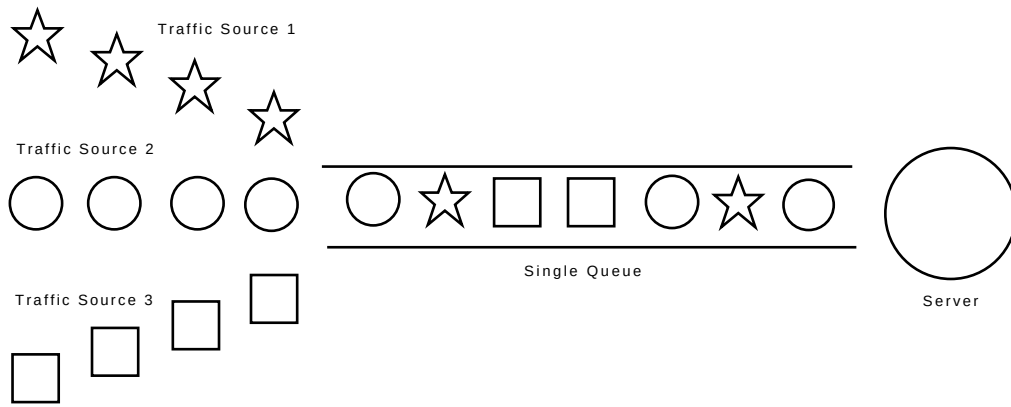


Figure 1.2 Illustration of Single Queue System

The reason why implementing differential service is that some traffic sources require very high quality of service due to their real time application in nature, while some other traffic sources may have no need of such high quality. Moreover, some traffic sources may have greater restriction on one QoS criterion while having much looser constraints on other QoS criteria. DiffServ differentiates these types of traffic sources. It classifies them into different priorities. Higher priority traffic flows would have priority to access resources.

In single queue system, there is only one queue for all the packets, thus all the traffic flows passing through this queue. Differential service however has to be applied on a multi-queue system. In order to activate the queuing process in a multi-queue system, a queuing discipline along with its setting (queuing policies) must be defined. A multi-queue system without taking priorities into account can be seen as single FIFO queue with the potential possibility to differentiate the QoS agreement for each application.

Figure 1.3 illustrates the queuing process in a queuing system at packet level. Once a packet arrives, it will enter a queue corresponding to its class. Notice that each queue corresponds to a priority, but there could be many traffic classes having the same priority, thus corresponding to the same queue. If the buffer of a queue is full, this queue will block new incoming packets of the same class. Some dropping policies would even drop a percentage of packets before the buffer reaches its full capacity. If all queues share

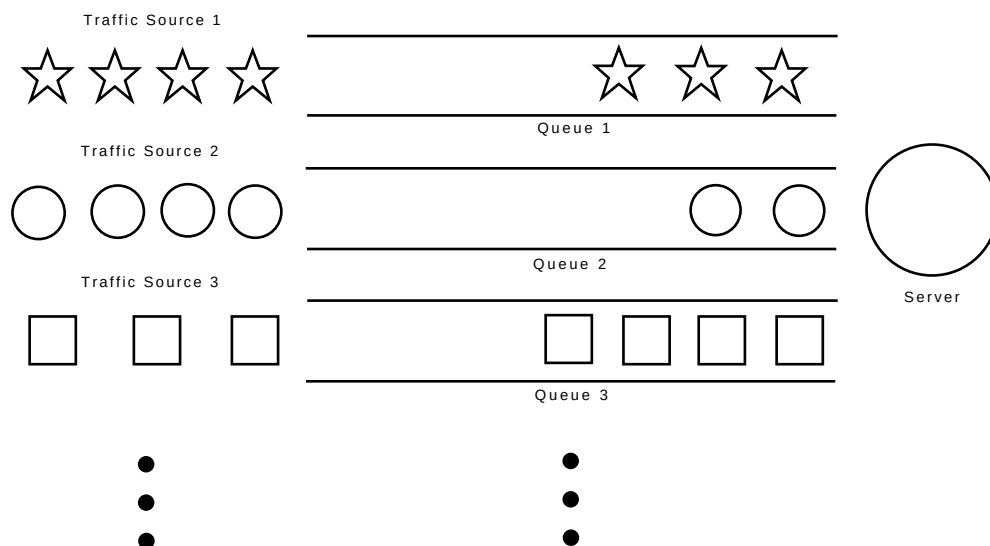


Figure 1.3 Illustration of Multi-Queue System

the same buffer and the overall buffer is full, the lower priority queue will begin to drop some packets.

In this research, we assume that the packet storage process is using fixed buffer size on each queue, i.e. every queue has its own predefined buffer size. Commonly, a higher priority queue has less buffer size than the buffer size of a lower priority queue. The queuing process is set to be FIFO in this research, which means first in, first served by a server.

A packet could be classified by differential service code point (DSCP) for its priority (see section 1.2.3). The queuing disciplines and the queuing policies will be discussed in section 1.2.1 and 1.2.2.

1.2.1 Queuing Disciplines

All queuing disciplines are modeled to approximate the generalized processor sharing (GPS) [5]. Ideally, a traffic flow is broken into tiny little pieces, like a fluid. Think about fluid model: Many heterogeneous liquids (type of traffic) passing through a pipe (queue)

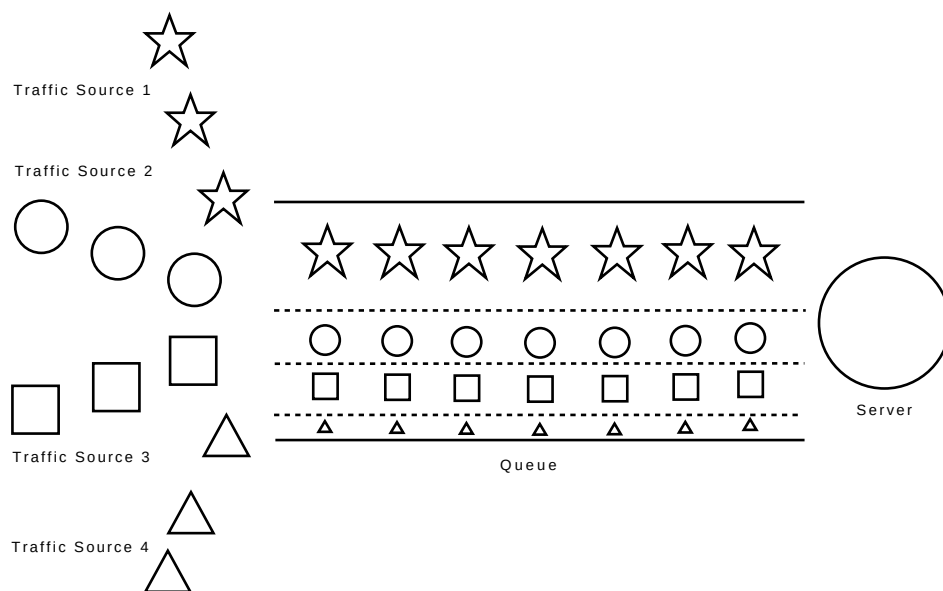


Figure 1.4 Illustration of Fair Fluid Queuing Model

and getting out of a valve (server). At any instant, each type of liquid is using a fixed portion of pipe and water cock in terms of radius. GPS is based on Fair Fluid Model, which combines a number of incoming flows into one combined flow with a density of each type of packets according to the original ratio. It is not practical because in network the transmission is in terms of packet, which can never be divided into infinite small size.

However in reality, a traffic flow can never be broken into infinite small pieces, and the server can never treat different traffic flows at exactly the same moment. Since GPS is an ideal queuing discipline in terms of fairness, which cannot be implemented in reality. However it can be seen as a model, in which all other disciplines are trying to approximate this model. In other words, how fair a queuing discipline can be really dependant on how well it approximates the GPS model. In order to judge the performance on the queuing discipline, there are criteria other than fairness, such as complexity.

Before discussing other queuing disciplines, here is a situation with 4 types of traffic sources (each generates 1 traffic flow in the network). Thus, there are 4 flows, which are F_1 , F_2 , F_3 and F_4 with weight ratios of 4 : 2 : 1 : 1 i.e. 50%, 25%, 12.5% and 12.5% respectively. Normal standard FIFO single queue system will serve any packet

according to its arrival time, regardless its priority.

GPS assumes that each packet of each flow can be divided into infinite small data, and that a single server can serve different flows simultaneously. This means at any given time t_1 , all 4 flows are transmitting data at that exact moment, thus $F1$ will use 50% of the total bandwidth and $F2$, 25% of the bandwidth while $F3$ and $F4$ each is using 25%. However, if at another time t_2 , there is no more data of $F1$ and $F2$ left, then $F3$ and $F4$ each will use 50% of the total bandwidth.

This scenario is illustrated in fluid model diagram (figure 1.4), and it will be later used to describe the small differences between queuing disciplines, such as priority queuing, fair queuing, weighted round robin, etc.

1.2.1.1 Virtual Time Concept (VTC)

Since a server can only serve one packet at a time, GPS (generalized processor sharing) cannot be implemented in reality, and one must decide which packet to be served first. Ideally, the packet requiring less time (waiting time + service time) should be served first. Due to the uncertain waiting time, any predicted (computed) timing is called virtual timing. There are 2 virtual times: virtual starting time and virtual finishing time. There are 2 conventional formula for these 2 virtual time computations.

Here are the 2 equations for computing the virtual starting time $S(k, i)$ and the virtual finishing time $F(k, i)$ of the k^{th} packet in session i :

$$S(k, i) = \max(F(k-1, i), V(a(k, i))) \quad (1.6)$$

$$F(k, i) = S(k, i) + L(k, i)/r(i) \quad (1.7)$$

Notice that the virtual finishing time is zero when there is no packet ($F(0, i) = 0$). For $a(k, i)$ being the arrival time a packet, $L(k, i)$ is the length of that packet.

$V(t)$ is the virtual time function for that $dV(t)/dt = 1/SoASSaT$, where $SoASSaT$ is sum of active sessions shares at time t [5, 6].

There are some new algorithms, such as the new virtual time function used by dubbed weighted fair queuing 2+ (WF2Q+):

$$V(t + T) = \max(V(t) + T, \min(S(h(i, t), i))) \quad (1.8)$$

where $h(i, t)$ is the packet number of the first packet in sessions i^{th} queue at time t . Also WF2Q+ can guarantee delay bounds given a session that is constrained with a leaky bucket [5, 6].

In general, it is very hard to emulate GPS because of the complexity for computing virtual times. Similarly, all queuing disciplines from the fair queuing (FQ) family require computation of virtual time, thus it would be harder to implement than round robin (RR) family.

1.2.1.2 Priority Queuing (PQ)

Priority Queuing (PQ) is the primary choice of queuing discipline in this research. Recall that previous research has applied CAC on networks without DiffServ, and in this thesis, we apply CAC on IP networks with DiffServ. For all queuing disciplines, the strict higher-priority-first PQ provides the largest contrast to simple FIFO queue network (without DiffServ). PQ does not offer any service to lower priority queue until the higher priority queue is empty. There could be many priorities. For instance, Cisco has a four-level prioritizing traffic model. The four levels are high, medium, normal and low.

[4]

Back to the previous GPS example, for $F1$ being priority 1, $F2$ being priority 2, $F3$ and $F4$ both being priority 3, $F1$ in queue 1 will be served first. $F2$ in queue 2 will begin transmission only after all $F1$ packets in queue 1 are being served. Similarly, $F3$ and $F4$ in queue 3 will be served at last, and it is FIFO for any packets in queue 3, regardless the packet is from $F3$ or $F4$.

The advantage of this discipline is that higher priority or more urgent packets has less waiting time. This discipline is also one of the simplest queuing disciplines in terms of complexity. The disadvantage is that the lower priority packets may never have a chance from obtaining service if higher priority packets keep arriving, which means PQ is less in favor, if not, least in favor of fairness.

In implementation, when a higher priority queue is not empty, there is no need to take care of lower priority queues except checking if their buffers are full.

1.2.1.3 Fair Queuing (FQ)

Fair queue (FQ) treats all the classes equally, which means there is no priority involved in the queuing process. Any packet having earliest estimated finishing time would be served first regardless from which traffic source. From the same previous example, it is possible that $F1$, $F2$, $F3$ and $F4$ each take one turn until there is no more packet left, assuming all packets have equal size.

FQ is completely different from FIFO single queue system. Fair queuing is using the virtual time concept (VTC) to compute the virtual finishing (service) time of a packet. Recall the virtual time computation in section 1.2.1.1. The packet that has the earliest of virtual finishing time, serves first. In fair queuing each type of traffic source has the same weight. Thus it often results in rotation for serving one packet of each type of traffic at

a time. In single queue system, it is simply according to the arrival time of each packet regardless its type. Thus fair queuing has greater fairness.

Advantage is that all the packets from different classes will also have a chance to be served, so that the heavy load of one traffic type can only slow other types of traffic flow down to certain degree. The disadvantage is that some more urgent packets may need to wait a relatively long period of time. Moreover, a disadvantage of the whole fair queuing family (FQ, WFQ, WF2Q, etc.) is that they need to compute predicted time (using virtual time concept) based on parameters such as packet size, which leads to highly complicated computations. Hence, rounded robin family (RR, WRR, WIRR, etc.) has been used in our research.

1.2.1.4 Round Robin (RR)

Round Robin (RR) or Packet Based Round Robin (PBRR) has the similar idea as fair queuing, except it considers everything at packet level (i.e. ignore packet size differences). Instead of computing virtual time(s), it serves one packet from each type of traffic at a time. If the packet size is fixed, it will end up with exactly the same result as fair queuing, with much less complexity in implementation. Again, from the same example, $F1$, $F2$, $F3$ and $F4$ transmissions will be equally turn based, regardless the packet sizes.

The advantage is that the complexity of modeling is very simple, and that RR provides fairness. The disadvantage is similar to fair queuing, some real time applications may need to be served at higher priority unless the network resource is so abundant, which is not happening today.

In implementation, making a loop to look over each queue at a turn is not hard.

1.2.1.5 Weighted Round Robin (WRR)

Weighted round robin is the round robin version of weighted fair queuing. Comparing to RR, it simply add a weight to each queue (class of traffic). From the previous example, the weight ratios could be $4 : 2 : 1 : 1$, but also could be $8 : 4 : 2 : 2$ or any $4x : 2x : x : x$ as x being any natural number. If there are always waiting packets in every queue, the service order would be $F1, F1, F1, F1, F2, F2, F3, F4$.

The advantage is still the less amount of complexity. Another advantage is that WRR does act as IntServ in which it reserves a limited minimum amount of bandwidth for lower priority traffic flow regardless the amount of higher priority traffic flow. The disadvantage is similar to Weighted Fair Queuing (WFQ), which is the gaps problem. There is a big gap between servicing two groups of higher priority packets. As the same example illustrated before, the ratio between $F1$ and $F2$ without considering other lower priority traffic flow is $2 : 1$. However, the weight system is set to be $4 : 2$. In which, 2 consecutive $F2$ packets would be served after 4 consecutive $F1$ packets, which is not a good approximation of GPS in terms of fairness. In many applications, especially handshaking communications, this will cause problem. This is why there are other modified versions of WRR, such as weighted interleaving round robin (WIRR).

In implementation, in addition to round robin program, it just requires to add a counter for each queue for its weight.

1.2.1.6 Weighted Interleaving Round Robin (WIRR)

Weighted interleaving round robin is very similar to WRR. The slight difference could be illustrated by the common scenario as previous example, if there are always waiting packets in every queue, the service order would be $F1, F1, F2, F1, F1, F2, F3, F4$. For a 2 priority network, if the weight for the higher and lower priority traffic flow is N and 1 respectively, there is no difference between WIRR and WRR.

The advantage is the slight better fairness comparing to WRR (may reduce the delay). For simplicity, the weight ratio is set to be $N : 1$ in this research.

1.2.1.7 Other Disciplines and Policies

There are many other queuing disciplines. Here is a list of them (there may be more which are not included): Deficit Round Robin (DRR) or Deficit Weighted Round Robin (DWRR), Elastic Round Robin (ERR), Shaped Round Robin (SRR), etc. RR family queuing disciplines. There are also many FQ family queuing disciplines such as WFQ, WF2Q, WF2Q+, SCFQ, SFQ, LLQ, etc., which are beyond the scope of this research.

1.2.2 Queuing Policies

Each queuing discipline may operate differently when some settings are changed. Here are some most important settings: service order, pre-emptive mode, dropping policy, buffering.

1.2.2.1 Service Order

There are many types of service orders, such as first-come-first-serve (FCFS) or first-in-first-out (FIFO), last-in-first-out (LIFO), user defined custom setting, etc. In this research, we only consider FIFO queues.

1.2.2.2 Pre-emptive Mode

Pre-emptive mode is a strictly high priority first execution policy. If a lower priority packet is being hold in service, (i.e. packet in transmission, while started uploading to a transmission line but not finished yet.) and a higher priority packet arrives at that

moment, the lower priority packet would be instantly dropped for granting service to that higher priority packet if pre-emptive mode is on. Because that lower priority packet would be considered as a loss, leaving pre-emptive mode on would introduce significant amount of packet losses to lower priority traffic flows. In this research, pre-emptive mode is always off. This may result slight higher delay for higher priority packets.

In theory, under priority queuing, only if the higher priority traffic flow outnumbers the lower priority traffic flow and that the lower priority packet size is comparable (preferred to be much larger) to the size of higher priority packet, the lower priority traffic flow will then affect QoS of the higher priority traffic flow.

1.2.2.3 Packet-Drop Policy

There are many ways to drop packets when buffer (queue) is full, such as drop tail and rate-based drop (RBD). The simplest is drop tail. It drops the last incoming packet when buffer is full.

Rate-Based-Drop drops packets with a buffer status (empty/full). For example, when buffer is less than 50% full, there is no dropping. When buffer is between 50% and 75% full, there is a probability of dropping some packets to maintain the buffer free space level. When buffer is above 75% full, it will drop at a higher probability. When buffer is nearly full, it will definitely drop some packets [7].

In real life, since the memory space becomes less expensive, there would be no dropped packets most the time. That is why many researchers tends to use unlimited buffer size (or a very large buffer), which means packet losses are often caused by failing QoS test such as delay and jitter tests. Large buffer deviation is not same as infinite buffer, and in practice, infinite buffer may cause memory leakage problem due to computer memory capacity in some cases.

1.2.2.4 Buffering

Buffer size stands for queue buffer size. In this research, each queue that represents a priority for certain traffic classes has its own buffer. Moreover, the lower priority queue has a larger buffer size than a higher priority queue, so that it can reduce packet loss due to buffer limitation as a normal internet service provider (ISP) would do in real world.

1.2.3 Differential Service Code Point(DSCP)

In general, there are 3 types of forwarding in DiffServ: Expected Forwarding (EF), Assured Forwarding (AF) and Best Effort (BE). EF tries to guarantee as much QoS as possible; BE has no QoS guarantee; and AF falls in between. These forwarding are all per-hop-behaviors (PHB). This means for a highest priority packet passing through node 1 (e.g., a router), yet this packet could be considered as medium priority in node 2.

The old IPv4 Type of Service (TOS) byte (8-bits) in an IP header is replaced by the differential service field. The first 6 bits are called Differential Service Codepoint (DSCP), and the last two bits are Currently Unused (CU) [8].

DSCP serves to classify different network traffic applications. Notice that 2 packets with different DSCP may have the same priority at a node due to per-hop-behavior. Based on the used 6 bits, DSCP can define 64 different distinguished types of network services. If all 8 bits are used, DSCP can define up to 256 TOS.

Codepoint for Expedited Forwarding (EF) PHB is $101110_2 = 46$, where PHB stands for per-hop-behavior [9].

For Assured Forwarding (AF), there are 4 different subclasses each having 3 different dropping precedence. Codepoints for AF subclasses are shown in table 1.1. Overall, there are 12 subclasses in AF. For AFXY, smaller X represents higher priority, while Y

Drop Precedence	Low	Medium	High
Class 1	AF11 001010 ₂ = 10	AF12 001100 ₂ = 12	AF13 001110 ₂ = 14
Class 2	AF21 010010 ₂ = 18	AF22 010100 ₂ = 20	AF23 010110 ₂ = 22
Class 3	AF31 011010 ₂ = 26	AF32 011100 ₂ = 28	AF33 011110 ₂ = 30
Class 4	AF41 100010 ₂ = 34	AF42 100100 ₂ = 36	AF43 100110 ₂ = 38

Table 1.1 DSCP for AF PHB

value only affects the dropping process [10].

Codepoint for Best Effort (BF) PHB is 000000₂ = 0, which is the default PHB [10].

1.3 Quality of Service Control

There are mainly two type of quality of service control: Quality of Service (QoS) and Quality of Experience (QoE) [11]. QoE is measured as how many customers are satisfied at the application level. QoS focuses more on some performance measurements such as delay, jitter, loss ratio, throughput, etc., which is more at lower level (e.g. packet, frame, session). In this research, mainly 3 QoS criteria are taking into account: delay, jitter and loss. In the following sections, there will be detailed descriptions on these QoS factors and the way of measuring them.

1.3.1 Delay

The End-to-End delay is defined as the time required for a packet to travel from the source of origin to its destination [12]. For simplicity in this research, only the main components of the delay are considered, e.g. propagation time is considered to be zero. In this research, the delay of a packet is the time accounted from the instant when the packet enters its corresponding queue until it is sent, which mainly includes the time waiting in the queue and the transmission time.

Different types of application have different degrees of demand about the maximum end-to-end delay. VoIP for example, low delay is really important for a high quality audio stream during real time conversation. Although company Avaya has proposed 80-100ms in order to provide better quality than cell phones years ago, the maximum delay people using today is between 150 – 200ms for a relatively good quality VoIP. ITU-T G.114 suggests for highly interactive networking tasks, the delay must be below $< 150ms$ (although sometimes, below 100ms delay may still affect the performance), and this maximum delay would provide good service for most applications. (Highly interactive applications include VoIP, video conferencing, interactive data applications, etc.) Based on their simulations for delay below 500ms, they suggest $> 400ms$ to be unacceptable. Thus there is a gray area in between 150ms and 400ms [13, 14]. In this research, the maximum delay for high quality VoIP is set to be 150ms. 400ms delay upper bound was also used for few low quality VoIP.

The delay value usually increases very fast as the number of traffic flow increases. Similarly, the loss from delay conversion also increases exponentially. The details will be described in chapter 2 validation section.

1.3.2 Jitter

The jitter is defined as the delay variation. More specifically, it is the difference between the delay of a packet and a referenced delay value. There are 2 definitions of jitter, thus 2 different ways to measure the jitter proposed by International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) and the Internet Engineering Task Force (IETF) respectively.

1.3.2.1 Definition of Jitter from ITU-T

ITU-T names jitter as packet delay variation (PDV). It uses $Delay_{min}$ as the reference value for delay variation. As $Jitter_i$, $Delay_i$, $Delay_{min}$ stand for the jitter for packet i , the delay for packet i and the minimum delay value for the packets belonging to the same type of traffic respectively, then the following equation defines the ITU-T version of jitter [15, 16, 17].

$$Jitter_i = |Delay_i - Delay_{min}| \quad (1.9)$$

The implementation would be easy if the $Delay_{min}$ value is known in advance. However it is not in our case, where there is no way of telling how well the estimation of the $Delay_{min}$ could be for some types of traffic (e.g. video traffic source).

Notice that sometimes the difference between $Delay_i$ and the reference delay value (in this case $Delay_{min}$) may be negative. Most researchers use absolute values for the jitter computation in order to avoid the trouble with the positive/negative signs.

1.3.2.2 Definition of Jitter from IETF

IETF, on the other hand, recommends using the previous packet delay as the reference value. In IETF, jitter is called IP delay variation (IPDV), except in RFC 3550, where the name jitter was introduced. $Jitter_i$, $Delay_i$, $Delay_{i-1}$ stand for jitter of packet i , delay of packet i and delay of packet $i - 1$, respectively. Then the following equation defines the IETF version of jitter [12, 18, 19].

$$Jitter_i = |Delay_i - Delay_{i-1}| \quad (1.10)$$

The IETF version of jitter is easier to implement because there is no need to know any mean/min/max values in advance. However, if a packet has been lost or duplicated between the time that the packet has been sent and the time that packet has been received or not, there could be situations where $Jitter_i$ is infinite or become undefined. If the packet does not arrive to its destination within a reasonable time, its delay and jitter would also be considered undefined. RFC 3393 recommends to throw out these invalid samples in order to proceed any statistical collection process and analysis [18].

Hence, in this research, we have applied the IETF version of jitter. Moreover, each time $Jitter_i$ is being computed, while packet i may not arrive destination, packet $i - 1$ must be the previous packet of the same traffic type that has reached its destination. In other words, the reference delay value would be $Delay_{i-2}$ if packet $i - 1$ failed any QoS criterion and was considered as a loss.

If a network application requires high QoS in terms of jitter performance (e.g., less than 1ms), most often there would be a jitter buffer to decrease the variation of delay, which is jitter, by increasing average delay. Any new arrival packets would first pass through jitter buffer (a simple FIFO queue). These packets would exit the jitter buffer in a constant rate if there is always some packets in the buffer. The size of the jitter buffer is in terms of average waiting time in the buffer.

The final jitter measured from the output of a jitter buffer would mostly depend on the jitter buffer size instead of the original jitter. Larger jitter buffer size would cover greater jitter, but also increase the average delay. In general, the larger jitter buffer size, the better jitter QoS, as long as the new average delay would not exceed delay boundary. For the buffer size, some researchers use 60ms [20]. A typical jitter buffer size is 30ms $\sim$ 50ms, maximum 100ms $\sim$ 200ms for VoIP. However, over 100ms jitter buffer would cause more problem on the delay side. For high quality video traffic source, jitter below 20ms is excellent, and above 50ms is unacceptable. Cisco suggests 30ms for both audio and video traffic source [21].

In this research, there is no jitter buffer, and the maximum jitter is set to be 20ms for audio and 50ms for video traffic source. For some audio traffic source related simulations, 50ms jitter has also been tested.

1.3.3 Loss Ratio

The loss ratio for a traffic source is the percentage of packets lost somewhere in between the source and the destination, or did not reach the destination within a predefined period of time T [22]. In real implementation, that time T is not a constant time in terms of seconds. It is a discrete count down number, $T, T - 1, T - 2, \dots, 0$. This number is called time to live (TTL). Each time the packet i is passing through a node, TTL decreases by factor of 1. A number of ways can cause packet i to be lost between the source and the destination.

- Packet i passed through too many intermediate nodes, and could not reach the destination before TTL reaches 0;
- Packet i arrives to the destination long after other packets in the same burst, in which the entire stream had already been declared as corrupted at higher layer;
- Packet i has been dropped due to buffer limitation dropping policy at a node between source and destination, and there is no other copies available in the network;
- Packet i has been corrupted;
- Packet i fails some QoS criteria test, during a transmission, and has been dropped (most likely as edge node in a network);
- Packet i could encounter a collision with another packet during transmission.
- other reasons.

In this research, the simulation focuses on a single link. Therefore, the main concern is when there is an overwhelming number of packets that arrive to a queue for certain traffic classes, which may cause overloading, thus forcing the system to drop some packets according to some predefined dropping policies. For this kind of loss, both audio and video traffic sources can accept a fair amount of loss (e.g. 1%). For audio traffic source (e.g. VoIP), the acceptable loss ratio is under 3%, and the favorable loss ratio is under 1%. Video traffic source has the same loss ratio requirement. Moreover, video traffic source is more sensitive to loss ratio than any other QoS criteria, especially for the packets belonging to frame I for MPEG-4 codecs (or any other reference frame for other codecs). However, data transmission would not accept this loss. If there is any packet lost or corrupted, the transmission must be restarted over at session level.

On the other side, those packets with delay or jitter value higher than the predefined acceptable range for the traffic source, would also be considered lost and be dropped. Again, for audio and video traffic sources, it is possible to simply consider that any packet dropped due to delay or jitter as part of packet loss, and set the maximum overall packet loss ratio to be less than 1%. However, for data transmission there is no room for error. If a data file has been corrupted, it must be resend. In this research, we simply assume that there is a recovery mechanism, which is able to recover loss less than 1% by resending that corrupted chunk (data is sent in many small pieces).

In the simulations, data traffic source packet loss ratio can never be caused by jitter, because jitter has no importance to any data transmissions. Normally, this loss ratio would not be caused by dropping policy either, because of large buffer settings for lower priority traffic source. Moreover, if there is one packet being dropped, very often more packets will be dropped as well. The number of packets being dropped can climb rapidly (either $<< 1\%$ loss or $> 99\%$). Most of the time (happens in all the simulations in this research where data traffic source is in present), the only reason why a data traffic source packet is considered as a loss, is when that packet has a huge delay.

QoS	Audio	Video	Data
Delay	75ms	100ms	2s
Jitter	20ms	20ms	NaN
Loss Ratio	1%	1%	1%

Table 1.2 Example of QoS Control Factors for Some Traffic Sources

The best way to test the QoS for data traffic source is to test the loss ratio at session level. However, since the simulations in this research are done at the packet level, the data traffic source loss ratio boundary has to be set to some value greater than 0%. This value has been set too be 1%. In other words, for data traffic source loss ratio (caused by delay constraint) to be less than 1%, CAC mechanism allows the transmission to be established and accepts the risk of being asked for retransmission (which is beyond the scope of packet level).

1.3.4 Service Level Agreement

The call acceptance is based on quality of service (QoS) control. If and only if all the QoS outputs are within its boundary, the QoS would be said to be “satisfied” and the call would be accepted. The boundary of each QoS is defined based on a service level agreement, which is individually predefined for each traffic class.

Table 1.2 shows some QoS boundaries for the 3 typical types of traffic (audio, video and data). This is just a general example. More detailed QoS service level agreement are used in the simulations. They will be shown in Chapter 3. For example, VoIP in single direction requires a delay less then 150ms. Voice message, on the other hand, has a maximum delay of 1s and 2s for playback and record respectively.

The delay and jitter are converted into loss ratio. When a packet exceeds its delay or jitter limit, it would be considered as a loss. The limit is set to be 1% for good service performance. In reality, up to 3% loss ratio is acceptable (but may not be desirable). For

some very low quality services, 10% may not be desirable but might still be considered acceptable.

For the QoS criteria, there are some records that can help understand how good is the internet performance in current time. For the Month April 2009 (August 2008 in brackets), the global response time has a mean of $132ms$ ($132ms$) and average peak value of $143ms$ ($144ms$). The global packet loss ratio has a mean of 6% (2% in August 2008) and peak value of 8% (4% in August 2008) due to recent 20% poor performance in Australia. Between the continents, North America has the lowest response time, the second highest packet loss ratio, and the second best overall performance [23].

1.4 Previous Research

Both effective bandwidth and differential service have been studied by many researchers in past several years, but very few studied them together. Most of these studies are solely related to admission control, i.e. how to control the networking traffic flow based on the admission region. In some papers, a traffic source can be called as a call, a connection or an application. The traffic type can be also called as the class, traffic class or call class.

1.4.1 Previous Research on Effective Bandwidth

In late 80's and early 90's, J.Y. Hui first introduced effective bandwidth into a network where a call's resource requirements vary the lifetime of a call. In a bufferless multi-class circuit-switch network (without a queuing system) before Hui's proposal, a call of traffic type i must know its bandwidth requirement in advance in order for equation 1.11 to be hold [24]. This is a deterministic QoS guarantee. The capacity of link k is also predefined. α_{ik} is the bandwidth required for type i traffic source (call) at link k .

$$\sum_i n_i \alpha_{ik} \leq C_k \quad (1.11)$$

Hui states that in a buffer free network, even if the bandwidth requirement (or so called effective bandwidth) α_{ik} varies in time t , the equation 1.11 can still be used to maintain a low probability of overload. This is a statistical QoS guarantee. In other words, providing a minimum required QoS for a type i traffic source, the equation 1.11 still holds. However, the effective bandwidth α_{ik} would be changed for different minimum required QoS [24, 25].

Later, also in early-mid 90's, Kelly proposed another version of effective bandwidth, in which queuing process has been involved. As mentioned in section 1.1.1, Kelly shows that the notion of effective bandwidth can be applied not only to unbuffered networks but also to networks with FIFO queuing systems.[26, 3].

Since then, many researches have been done to study the effective bandwidth for different traffic sources in different networks, especially for ATM network, IP network and wireless network. The study area covers admission region, congestion, protection, connection admission control protocol, etc. In the past few years, LORLAB has studied the admission region for different traffic sources in wireless network. Without DiffServ, the resulting admission region curves are mostly linear, so that the admission regions are often in triangular forms [1]. Thus, the admission region curve can easily be approximately by a linear equation. In our thesis, we study the admission region in DiffServ IP network.

1.4.2 Previous Research on DiffServ

Differential service (DiffServ) is more a definition or a concept. It is defined in several RFC's. Here is a list of currently available RFC's related to DiffServ.

- RFC 2474 Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers
- RFC 2475 An Architecture for Differentiated Services
- RFC 2597 Assured Forwarding PHB Group
- RFC 3140 Per Hop Behavior Identification Codes (Obsoletes RFC 2836)
- RFC 3246 An Expedited Forwarding PHB (Obsoletes RFC 2598)
- RFC 3260 New Terminology and Clarifications for DiffServ
- RFC 4594 Configuration Guidelines for DiffServ Service Classes

In general, DiffServ can guarantee different QoS demand for each class (type of traffic). It can also aggregate traffic flows to different classes. In other words, classify different applications to a number of predefined classes. The QoS guarantee implementations are per-hop-behaviors, which means the DiffServ is applied at each node (e.g., switch or router) separately instead of reserving a path from end to end. Each traffic flow obtain its QoS statistically.

DiffServ working group in the IETF was formed in 1997 due to scalability concerns of integrated service (IntServ). A year later, the first two RFC's, RFC 2474 and RFC 2475 were published. RFC 2474 obsoleted the old 8-bit type of service (TOS) field in IPv4 packet. A new 8-bit DiffServ or DS Field was taking in place [8, 27].

Before 1998, the study focus was on ATM QoS and IntServ. After 1998, the focus shifted to DiffServ. In year 2000, the market began to sell routers support DiffServ. In our thesis, we are applying the notion of effective bandwidth to DiffServ network.

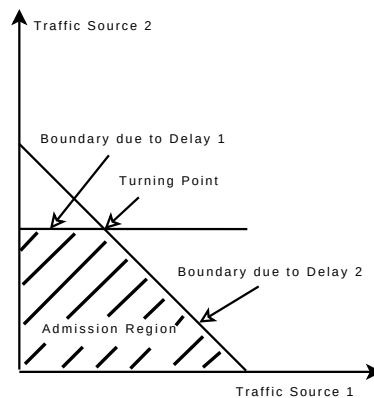


Figure 1.5 Admission Region due to Delay for Priority Queuing ATM Network

1.4.3 CAC for PQ Guarantee Delay QoS

There is only one paper that has done some research on admission region in DiffServ network, by Berger & Ward [28]. Their 2 priority system under priority queuing results in a trapezoid form of admission region due to the fact that it uses large buffer approximation and considers the delay performance as the only QoS factor for effective bandwidth by converting delay performance to cell loss ratio. Moreover, all cell (ATM packets) sizes are always equal and very small. In that case, there is already a mathematical formula for the admission region curve (geometry equation of trapeze).

To better understand Berger & Ward's model, figure 1.5 illustrates an admission region plot for 2 types of traffic in a priority queuing ATM network, where delay is the only QoS criterion taken into consideration. The trapeze form of admission region (AR) is formed by 2 straight lines. The horizontal line is due to the delay constraint on traffic type 1 (higher priority). The slope is due to the delay constraint on traffic type 2 (lower priority). Yet, the higher priority traffic flow QoS is the binding factor for the left part of the AR, and the lower priority traffic flow QoS dominates the right part of the AR.

In our research, jitter has been introduced, and the queue buffer is not infinitely large. Thus the admission regions may no longer be a trapeze and may vary depends on the type of traffic. Beside the shape of the admission region, the idea of converting other

QoS to cell loss ratio has been borrowed.

1.5 Conclusion

In this chapter, we have shown the related background terminologies and concepts that will support the later experiments in this thesis. We have also discussed some previous researches. In chapter 2, we will describe our simulation model using these terminologies and concepts.

CHAPTER 2

SIMULATION MODEL

In networking, there are many problems that cannot be solved analytically. Recall the goal is to compute the boundary of the admission region and check if this is sufficiently linear to use an effective bandwidth technique for admission control. The computation is too difficult for an analytical approach because of non Poisson traffic, complex queuing and scheduling, multiple QoS measures, dependence of one QoS measure on the other, etc. Thus, software simulations become handy.

There is one well known software that can test network in many different situations, “ns2”. It is free and open source. However, it does not have DiffServ implemented. Beside ns2, other software such as OPNET are not free in charge. There are also many small programs, designed specifically for some study purposes, for example a specific queuing discipline. Therefore, the first step of this research is to design a software, which is capable of simulating the networking environment.

Recall the ultimate goal of plotting the admission region for different network applications under DiffServ. In order to do so, a queuing model and several traffic source generators must be made. This queuing model is built into a program called CAC_SIM (stands for connection admission control simulation). This program CAC_SIM must be able to input data from selected traffic source generators, to simulate queuing behavior, and to output the Quality of service (QoS) measurements. CAC_SIM is programmed under java environment because it is using a java math library called SSJ.

2.1 Stochastic Simulation in Java (SSJ)

For QoS simulations, there is an available tool called Stochastic Simulation in Java (SSJ) [29]. It is a MATH tool, which allows us to generate random variables and/or random streams. It also provides some observers for the purpose of statistical collections. Since SSJ was developed under Java environment, our program is therefore also using the Java language. Moreover, we use simulation by “events” technique. i.e. the simulation program ONLY updates when an event occurs. An event could be the arrival of a new packet, the buffer becoming full (blocking), the server becoming free, etc.

2.2 General Idea of the Simulation

The problem of how to obtain the admission region can be translated into a smaller problem. Assuming there are 2 types of traffic, and we want to plot the admission region on a plane. Given a point (n_1, n_2) in the plane, where n_1 and n_2 are the number of type 1 and 2 traffic flows respectively, we can ask whether all QoS are satisfied. The QoS satisfied points will form the admission region. The core part of CAC_SIM Model is made to answer this question.

Overall, in order to find the boundary of the admission region in a systematical way, n_1 can be looped from 0 to the maximum number of traffic type 1 allowed (N_1). For every single value of n_1 , find the maximum number of traffic type 2 allowed ($N_2 = f(n_1)$) using CAC_SIM.

2.3 Structure of the CAC_SIM Model

Currently CAC_SIM simulate the queuing process at the link layer. It has not yet considered either the MAC nor the physical layer into account. Basically it simulates an output

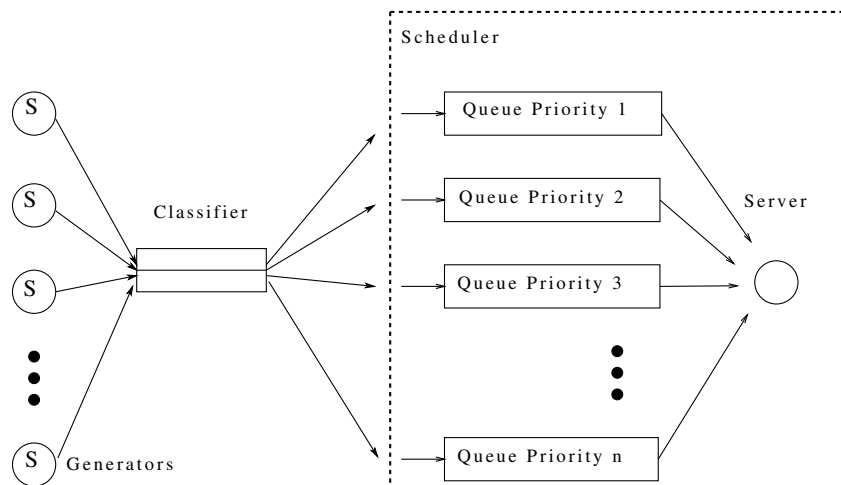


Figure 2.1 Illustration of the output port from CAC_SIM model

port, which includes several queues and a single server. In order for a traffic source generators to be able to feed packets into the corresponding queues, a small module called classifier has been created. The basic components of CAC_SIM is illustrated in figure 2.1.

A number of traffic source generators feed packets into the classifier, which would transfer each packet into its corresponding queue according to the packet type and the priority setting. The scheduler would drive one packet at a time to the server according to the queuing discipline.

2.3.1 Classifier

There are thousands of networks applications. It is therefore impossible to assign each application a different priority. While ATM only allows maximum 2 different priorities, normal IPv4 packet allows maximum $2^6 = 64$ different priorities (recall DSCP from chapter 1). In other words, each type of traffic source $n \in N$ can be classified into $m \in M \leq 64$ priorities. Today, mostly 3 or 4 priorities have been used. Thus the applications must be categorized into a fewer number of classes.

Normally audio source such as VoIP is classified as EF (DSCP= 46), video source classified as AF41 (DSCP= 34) and data transmission such as email or web surfing is classified as AF11 (DSCP= 10) or BE (DSCP= 0) depends on network settings [21]. However there are some exceptions. For example, a video traffic source could be classified as EF if it is used for video conferencing. Video-on-demand (VOD) could as well be considered as priority AF11 because it is pure data burst transmission in some sense, yet VOD is also a video traffic source.

The classifier considers EF as the highest priority (priority 1), and the priorities for AF41, AF11 and BE are considered as medium high (priority 2), medium low (priority 3) and low (priority 4) respectively. Here, from the highest priority to the lowest priority, the ranking system is priority 1, priority 2, priority 3 and priority 4. For differential service code point (DSCP), the ranking system is EF, AF41, AF11 and BE. The detailed DSCP system description can be found in section 1.2.3.

The way of classifying priorities is defined by Internet Service Providers (ISP's) based on their own rules because there is no standardization yet. There are still some general guidelines to follow. For example, real time applications usually end up to become higher priorities. This is the main reason why VoIP is always at top of the priority list. Of course the greater a customer is paying to ISP would result in better QoS due to higher priority settings. For example, IPTV companies usually own the transmission lines or has a deal with the local ISP because they require considerable amount of bandwidth.

In CAC_SIM model, the classifier receives incoming packets possibly from several different traffic source generators. Based on the information related to the type of traffic source (application) stored in each packet, it will assign a priority index to the packet, and send the packet to the output port. This priority index is same as the queue index in the queues inside of the output port.

2.3.2 Output Port

An output port includes a number of queues and one server. Its main purpose is to upload packets to the transmission line plugged into the output port, in an order according to the queuing discipline and the queuing policies (which were discussed in chapter 1). The whole mechanism is controlled by the core of the program, the scheduler.

2.3.2.1 Scheduler

The scheduler is responsible for all packet behaviors from entering the queuing system until exiting the server. It is also responsible for recording all the time, space (size in kB) and QoS related statistics.

Once the classifier transfers a packet with a queue index into the output port, the scheduler will store the packet arrival time in this packet type object and put this packet into the according queue if this new arrival packet does not cause the buffer of that queue to overflow. As long as none of the queues are empty, the scheduler will remove a packet from a queue based on the predefined queuing discipline. During this process, the scheduler will record the delay and jitter of that packet. For each packet i from traffic type n ,

$$Delay_i^n = DepartureTime_i^n - ArrivalTime_i^n \quad (2.1)$$

$$Jitter_i^n = |Delay_i^n - Delay_{i-1}^n| \quad (2.2)$$

If the delay and jitter (notice that $i \geq 1$ and $Jitter_0^n$ does not exist) exceeds the QoS limit, the overall loss (loss due to any failed QoS test) would be increased by 1. In other words, we translate delay or jitter problem into loss problem. In reality, this packet

would carry on toward the destination, although the destination may ask the source to resend the packet. We keep records on the overall loss as well the loss due to drop, the loss due to delay, the loss due to jitter, the loss due to delay or jitter, the average delay, the average jitter, etc. Notice for the delay equation, the delay includes the time a packet waited in the corresponding queue and the time required for the server to upload it onto the transmission line. The time needed for a packet after exiting the output port and entering next node is considered as 0.

For the jitter computation, the previous packet $i - 1$ must be stored as a referencing point because the jitter is the variation of the end-to-end delays. Another adaptation in practice is that there is no existing $Delay_{i-1}$ values for $Jitter_0$ to $Jitter_k$, where k^{th} packet is the first packet that satisfies all other QoS criteria except jitter which is yet unknown but assumed to be satisfied. Moreover, the last packet $i - 1$ delay value must be a packet from the same traffic type n . Hence, the new $Delay_{i-1}^n = NaN$ until k^{th} packet, in which the scheduler will assign $Delay_k^n$ to new $Delay_{i-1}^n$.

$$Delay_{i-1}^n = Delay_i^n \quad (2.3)$$

Therefore, the first referencing delay value would be $Delay_k^n$, and the first jitter value would be $Jitter_{k+1}^n$ for the traffic type n .

2.3.2.2 Queue

For X number of priorities, X number of queues are created (often 3 to 4, and 64 at most). Each single FIFO queue has some status variables for its properties in which the program can initiate, read and rewrite these records.

- boolean priority: the priority of the queue in the system

- double maxBufferSize: maximum allowed storage size for the buffer of the queue (in [kB]); the maximum buffer for the highest priority (priority 1) queue is 4096kB, and it doubles for each lower rank of priority.
- double bufferSize: current used buffer size (in [kB])
- int numPacketInQueue: current number of packets in this queue
- int blockPackets: (true/false) for blocking a new arrival packet
- boolean blockModeOn: (true/false) for simulation to take the buffer size limitation into account, set to be false stands for infinite buffer.
- int numBlocks: number of blocks in this queue until this moment

Beside these status variables, there are several methods (functions) being used to allow other classes to have indirect access (read or write) the objects inside of class QueueSingle:

- public double bufferSizeUpdate(double totalLength, boolean entrance): an internal method to update the current buffer size when a packet is arriving/leaving.
- public boolean addPacket(Packet newPacket): If the block mode is OFF, it will simply call bufferSizeUpdate() to update the current buffer size, add the packet to the end of the queue, add 1 to the status variable numPacketInQueue, and return boolean false (i.e. the packet is NOT blocked).

If the block mode is ON, and the latest bufferSize plus new packet size all together is greater then the maximum buffer size, then the variable numBlocks will be updated and the method will return true (i.e. the packet is blocked).

- public Packet remove(): It updates the buffer size and number of packet in the queue, returns and deletes the first packet in the queue.

- `public Packet read()`: It returns the first packet in the queue without any modification to the queue, such as delete the returned packet or change amount of used buffer size.
- `public double getNumBlocks()`: It returns the accumulated number of blocks in this queue.
- `public int getNumPacketInQueue()`: It returns the current number of packets in the queue.
- `public double getUsedBufferSize()`: It returns the current amount of current used buffer (in [kB]).
- `public void setMaxBuffer(double maxBufferSize)`: It will set a new user defined maximum allowed buffer size for the queue (in [kB])

2.3.2.3 Server

Similar to single queue, this class is designed to create a server with certain properties (some properties are user defined). It has a number of variables showing its status:

- `servRate`: service rate of the server (in [kB/s])
- `inServ`: true/false the server is in service (i.e. currently being used)

Beside these status variables, there are several methods being used to allow other classes to have indirect access (read or write) to the objects inside of class `ServerSingle`:

- `public void setServRate(double serviceRate)`: It allows user to set a new service rate in [kB/s].
- `public double getServRate()`: It returns the service rate in [kB/s] for this server.

- `public void beginService()`: It set the variable `inServ` to be true.
- `public void endService()`: It set the variable `inServ` to be false.
- `public boolean isInService()`: It return the variable `inServ` in true/false.

2.3.2.4 Statistical Collections

All the statistics for each repeated test are recorded by the scheduler. All the data are collected using Tally or Accumulator. Tally is a class from SSJ library to create objects for observations. It allows people to compute average, maximum, minimum, confidence interval, etc. Accumulator is similar to Tally, except that it simply accumulates values as a counter with more flexibilities and functionalities. For example, accumulator could be used to measure the number of blocked packets.

Once a packet has been created, any measurable values for this packet (e.g. arrival time) will be stored in this packet type object. In the scheduler, each Tally object is only responsible for the information from one traffic type. Therefore, even if 2 packets were stored in the same queue because of having the same priority, their status may be recorded separately.

Notice that since there are hundreds of repeated test for each testing scenario, for each type of information, an additional Tally object must be created at the highest level. At the end of each repeated test, the average value of the Tally object inside the scheduler will be stored in the Tally object at the highest level. Then the Tally object inside the scheduler has to be initialized before the beginning of next test. In other words, the final raw data is the average of mean values (e.g. the average of the average packet losses).

Primary Data:

- `int nbSi`: the maximum number of connections (flows) for traffic type i while there may be other types of traffic in presence;

- double $loss_i$: the overall packet loss ratio of traffic type i ;

Secondary Data:

- double $loss_i$: the overall loss for traffic type i due to the failure of any QoS test, which includes delay, jitter and drop;
- double $lossBlock_i$: the loss for traffic type i due to the corresponding queue buffer capacity;
- double $lossDelay_i$: the loss for traffic type i due to delay exceeding the service level agreement;
- double $lossJitter_i$: the loss for traffic type i due to jitter exceeding the service level agreement;
- double $lossDelayJitter_i$: the loss for traffic type i due to delay and/or jitter exceeding boundary limitation;
- int arr_i : total number of type i packets arrived to the queuing system;
- double $delayT_i$: the average end-to-end delay value for traffic type i (excluding dropped packets);
- double $jitterT_i$: the average end-to-end jitter value for traffic type i (excluding dropped packets);
- double $delayW_i$: the average delay value (only consider the time for a packet waiting in a queue) for traffic type i (excluding dropped packets);
- double $jitterW_i$: the average jitter value (only consider the time for a packet waiting in a queue) for traffic type i (excluding dropped packets);
- int $nbPackQueue_i$: average number of packet in the queue corresponding to traffic type i ;

- double *bufferUsedQueue_i*: average used buffer size of the queue corresponding to traffic type *i* in terms of Byte;
- int *arrFree_i*: whenever a packet from traffic type *i* is being transmitted without waiting in a queue, *arrFree_i* increment by 1;

Other Data:

- double *arrFreeJitterT_i*: the average end-to-end jitter value for arrFree packets from traffic type *i* (excluding dropped packets);
- double *arrFreeJitterW_i*: the average jitter (only consider the time for a packet waited in a queue) value for arrFree packets from traffic type *i* (excluding dropped packets);
- double *var(loss_i)*: the variance of *loss_i*;
- double *var(delayT_i)*: the variance of *delayT_i*;
- double *var(jitterT_i)*: the variance of *jitterT_i*;
- double *var(delayW_i)*: the variance of *delayW_i*;
- double *var(jitterW_i)*: the variance of *jitterW_i*;
- other double *var()* data.

2.3.3 Source Generators

There are mainly 3 types of generators used in this research: ON-OFF generator, continuous generator and video generator. Each of these traffic source generators is able to generate several different types of traffic sources. CAC_SIM allows multiple generators of same/different type to generate multiple traffic flows simultaneously. The generated

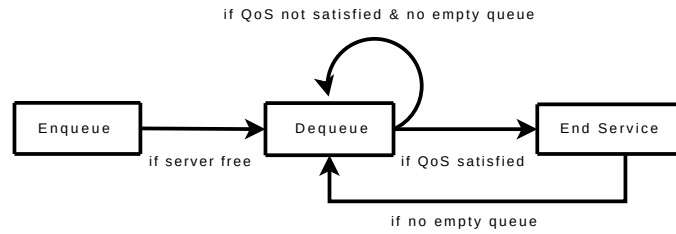


Figure 2.2 Scheduler Process

packets would be sent to classifier with its properties such as creation time, packet size, packet type, etc. (for details about different source generators please see chapter 3)

Despite the differences of each source generator, there are some common points. In order to make the simulation more realistic, when a source generator is launched, it would be assigned with a random phase shift to avoid in phase problem (details in the section of Simulation Techniques later in this chapter).

2.4 CAC_SIM Simulation Process

Two mechanisms have been used in the simulations. Figure 2.2 shows the process used in most simulations through the research. Basically, this mechanism has a QoS detector, in which it will drop packets that fail any service level agreement on any QoS criterion. This type of scheduling process often happens at the edge nodes.

In this thesis, the process in figure 2.3 is only for analyzing some scenarios where the node is a simple switch (no QoS control). Normally, this type of scheduling process would happen at non-edge nodes. In practice, although the process without QoS detection (figure 2.3) would result slightly less admission region area comparing the case in figure 2.2, there should not be too much differences between both process outputs, because the difference would only be applied to a maximum of 1% packets. 1% of total amount of packet can hardly affect the outcome at grand scheme.

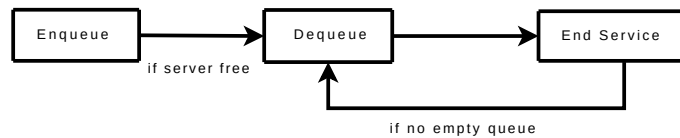


Figure 2.3 Scheduler Process without QoS Detection

2.5 Simulation Techniques

In order to understand how the final results are obtained and how well the results stand for, the techniques used in the simulation would be introduced in this section. The goal of using these techniques is to make the simulation faster in terms of CPU run time, more reliable in terms of emulating real networking problems and more convincing in terms of % error.

2.5.1 General Techniques

The overall simulation is at the packet level. Thus everything below this level would either be emulated by the properties of packet level or be neglected. Similarly for the levels above, CAC_SIM would break them into a streams of packets, e.g. CAC_SIM takes frames into account but in terms of packet burst. Each session is assumed to be infinite length or long enough so that it won't end before the end of each repeated test.

2.5.2 Avoiding In Phase Source Generator Problem

The final result, each point on the admission region boundary curve, requires multiple traffic sources being generated simultaneously. If 2 traffic source generators begin to generate packets at the exact same moment, the high probability of collision between the first few packets may slightly increase the packet loss ratio and the average end-to-end delay. Moreover, if the traffic source generator is a deterministic source (e.g. CBR data traffic source), which have fixed packet arrival time, the collisions are certainly going

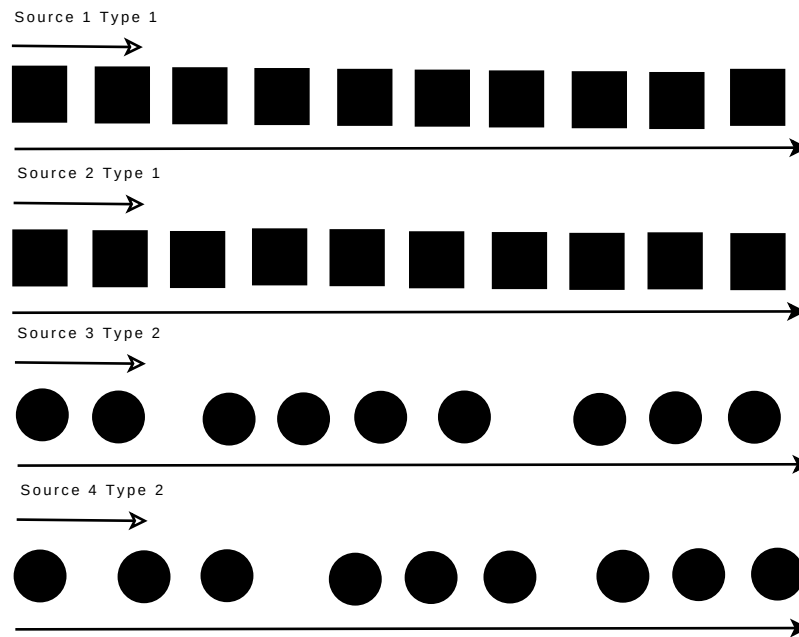


Figure 2.4 Illustration of In Phase Problem

to be a matter as the number of traffic flows increases. Notice that the video source generator has a fixed frame rate, which would produce packet burst at the beginning of each frame. This would also result in higher collision probability.

In Figure 2.4, source 1 and 2 (traffic source generator 1 and 2) would send type 1 packets at a fixed rate, and this would cause multiple packets created at exact the same moment. The collision problem would be worse if the number of any type 1 sources increases. However, this problem only has small influence on the first few packets of type 2 sources, where it is illustrated between source 3 and 4.

Since the problem is caused by the same starting time, a slight phase shift can easily solve the problem. This phase shift is an uniform random generator with a boundary of $[0, Period]$. The *Period* can be the average packet arrival time or average frame rate (See Figure 2.5).

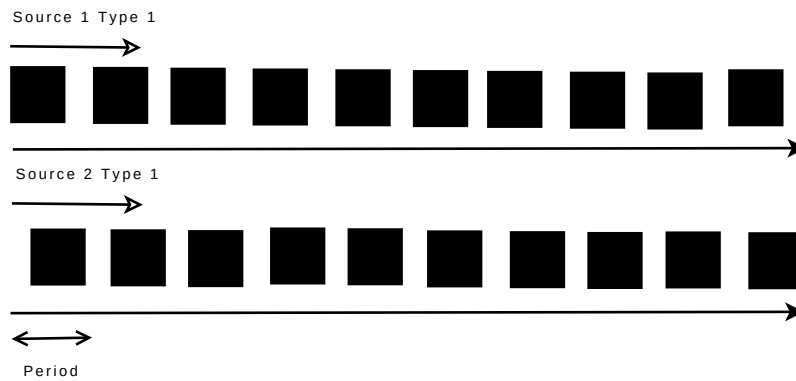


Figure 2.5 Illustration of Solution to In Phase Problem

2.5.3 Acceptance Procedure

For a given configuration (n_1, n_2) , where n_1 and n_2 are the number of type 1 and 2 traffic flows respectively, we can ask the question whether all QoS are satisfied. Since simulation cannot keep running forever, another question would be how many times to repeat the test for a good approximation. This is where the acceptance procedure steps in.

The acceptance procedure is based on the idea of sequential analysis (or sequential hypothesis testing), first introduced by Wald [30]. The general idea is that for unfixed sample size, continue to sample until certain conditions met based on some predefined stopping rules. In other words, the number of samples for each configuration depends on a DO-WHILE-LOOP, which only stop when reaching certain condition. In our case, the conditions would be based on the log-likelihood ratio.

This procedure requires to set few user defined parameters (could be changed, but must be consistent throughout the research). First, a configuration (or a set) can be defined as $[s_1, s_2]$, where s_1 being the number of traffic type 1 (higher priority) and s_2 being the number of traffic type 2 (lower priority); and the other main parameters are:

- I^i : number of simulations to obtain the approximated variance, for traffic type i ;

- α^i : Probability of error in accepting a configuration for traffic type i ;
- β^i : Probability of error in rejecting a configuration for traffic type i ;
- p_0^i : Lower bound of the configuration for traffic type i ;
- p_1^i : Upper bound of the configuration for traffic type i ;

For example, a primary test of 32 repeated tests ($I = 32$) give a variance of σ^2 . For a maximum loss/arrival percentage of 1%, set α to be 5%, β to be 10%, p_0 to be 0.9% and p_1 to be 1.1%. The acceptance procedure is based on the following 3 equations, where n is the number of iterations (repeated test), x_i is a single experimental result, $\bar{X}_n$ is the accumulated experimental results from 1 to n :

$$\bar{X}_n = \sum_{i=1}^n x_i \quad (2.4)$$

$$p^- = \frac{\sigma^2}{p_1 - p_0} \log \frac{\beta}{1 - \alpha} + \frac{n}{2}(p_0 + p_1) \quad (2.5)$$

$$p^+ = \frac{\sigma^2}{p_1 - p_0} \log \frac{1 - \beta}{\alpha} + \frac{n}{2}(p_0 + p_1) \quad (2.6)$$

The decision making about whether to accept or reject the configuration can only be done outside of the gray area ($p^- < \bar{X}_n < p^+$). After a predefined number of iteration, if $\bar{X}_n$ is inside of that gray area, where no decision yet can be made, then the next iteration must be computed until $\bar{X}_n$ reaches either the upper or lower bound. Since there are already 32 repeated tests done to compute the variance; the minimum number of repeated test is 32 in this example. For the variance values when $n > 32$, σ^2 is recalculated ($\sigma^2 = \sigma_n^2$) [30].

Figure 2.6 illustrate the way the acceptance procedure can be applied on the simulation.

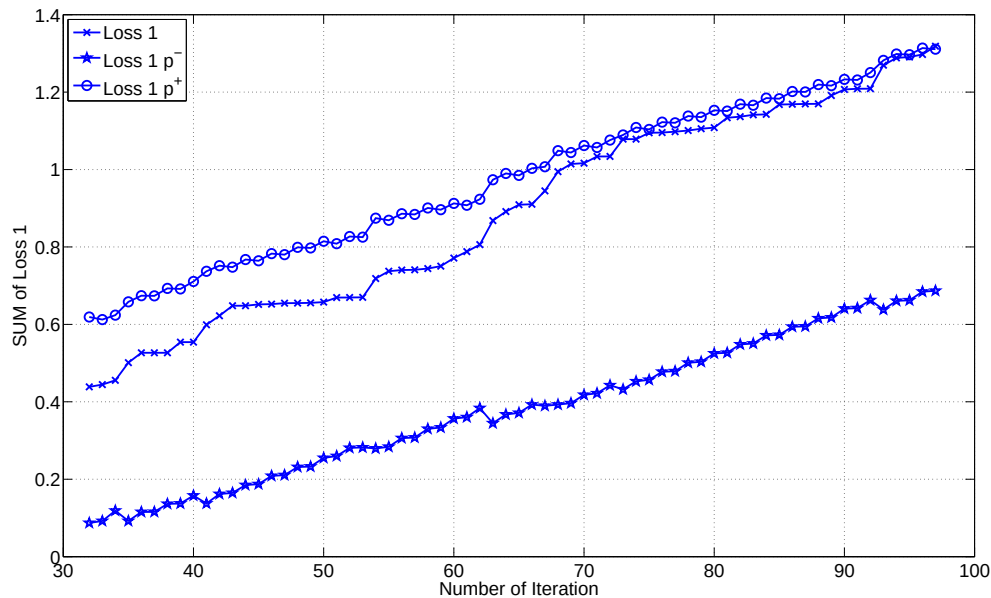


Figure 2.6 Acceptance Procedure Illustration: It is based on the loss of traffic type 1, where p^- and p^+ being upper and lower bounds respectively.

The same configuration would continue as every new iteration restart the same simulation over and over again, until the final result reaches either the upper bound or the lower bound. In this case, the simulation hits the boundary at the 97th iteration. (The data were extracted from VoIP-EF vs LiveTV-AF41 under WRR.)

Another comment is that if one type of traffic has a QoS outside the gray area and it fails the service level agreement, there is no need to continue the simulation even if other type of traffic having QoS are still within the gray area.

2.5.4 Discrete Admission Region

The admission region (AR) is in the discrete environment (see Figure 1.1). Recall that for every value of n_1 , we have to find the maximum n_2 allowed, where n_1 and n_2 are the number of type 1 and 2 traffic flows respectively. The old fashion way for an AR test is for a fixed number of type 1 traffic flows (n_1), begins with 0 type 2 traffic

flows (n_2), and increment by factor of 1 each cycle until QoS fails. This requires around $(\max(n_1)\max(n_2))/2$ times of computations. It should be done in a much more efficient way.

Knowing the curve is decremental, a modified version is to use M/G/1 model computing an approximated value of n_2 when $n_1 = 0$. Begin with this value, test if it satisfies the QoS. If it satisfies, increment n_2 by 1 unit, else reduce by 1 unit. The final value $n_2 = f(n_1)$ can only occur in 2 occasions during the simulation, where n_1 would be incremented by 1:

- When $n_2 = b$ satisfied QoS and $n_2 = b + 1$ unsatisfied QoS, record $n_2 = b$ for $n_1 = a$ and let $n_1 = a + 1$
- When $n_2 = b$ unsatisfied QoS and $n_2 = b - 1$ satisfied QoS, record $n_2 = b - 1$ for $n_1 = a$ and let $n_1 = a + 1$

This way, ideally the computation complexity would be around $\max(n_1) + \max(n_2)$. a and b are just some integer values. In figure 2.7 (flowchart), for a fixed number of type 1 traffic source, we want to find the maximum number of type 2 traffic source allowed while satisfying the QoS for both type of traffic sources. When the simulation reaches the END of the flowchart, n_1 can be increased by 1 and restart the whole simulation again unless QoS test fails even if there is 0 type 2 traffic source.

2.5.5 Precision

Each traffic source (type i) requires certain amount of effective bandwidth. Both the throughput of each traffic source and the line capacity would affect the maximum number of allowed type i traffic source (N_i). Greater this maximum (N_i), longer the simulation would run. The goal is to adjust this number to be not too large (long simulation run time) and not too small (low precision). In general, we want N_i to be around 50.

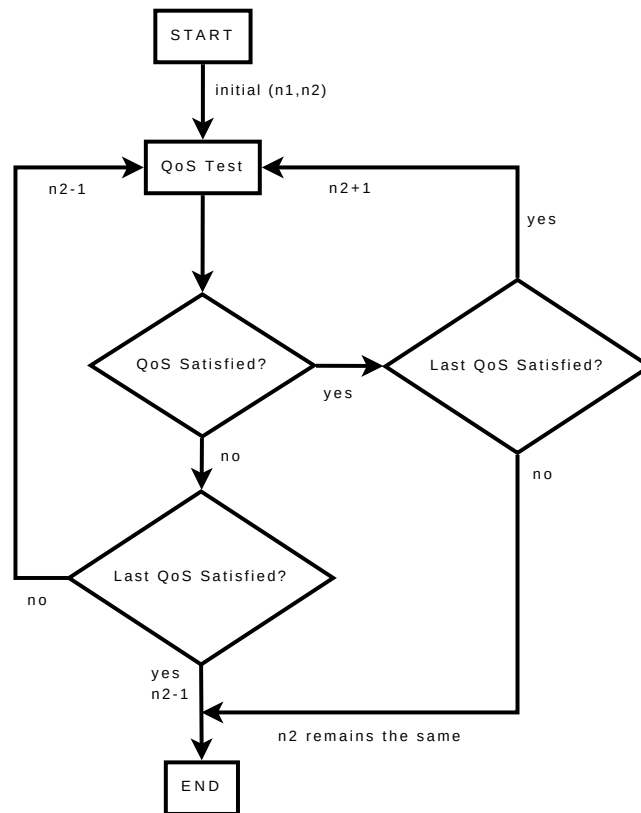


Figure 2.7 Finding the Maximum Value of n_2 (Number of Type 2 Traffic Source) for a Fixed Value of n_1 (Number of Type 1 Traffic Source)

The throughput of both audio and video traffic sources cannot be changed because we are emulating real network traffic sources. By adjusting the line capacity setting to 2Mbps, most of the N_i values are between 30 and 60. For this fixed line capacity 2Mbps, the CBR (constant bit rate) data traffic source bitrate can be adjusted for the maximum allowed CBR data traffic source to be around 30-40. We can play with these numbers because of the CBR data traffic source in its nature and the additivity property of effective bandwidth mentioned in chapter 1.

2.6 Validation of Output

Although it is really hard to use mathematical model to describe a queuing system, for some simple cases, there are existing analytical formula. This section is going to show that CAC_SIM performs correctly, i.e. simulation output is similar to theoretical values (based on formula) or practical data (e.g. video trace files).

2.6.1 Confidence Interval Vs. (Number of Repetition)(Simulation Length)

Theoretically, increasing the number of repetitions would introduce a smaller confidence interval (CI), thus better precision [31]. In other words, the problem of what is the satisfactory size of the CI region has been transformed into another problem, which is to find how many repetitions (samples) are enough for a satisfactory precision. This problem is resolved using the acceptance procedure based on Wald sequential test (see section 2.5.3 on the Acceptance Procedure). For all the simulations in this research, all the variance values are based on 95% confidence interval.

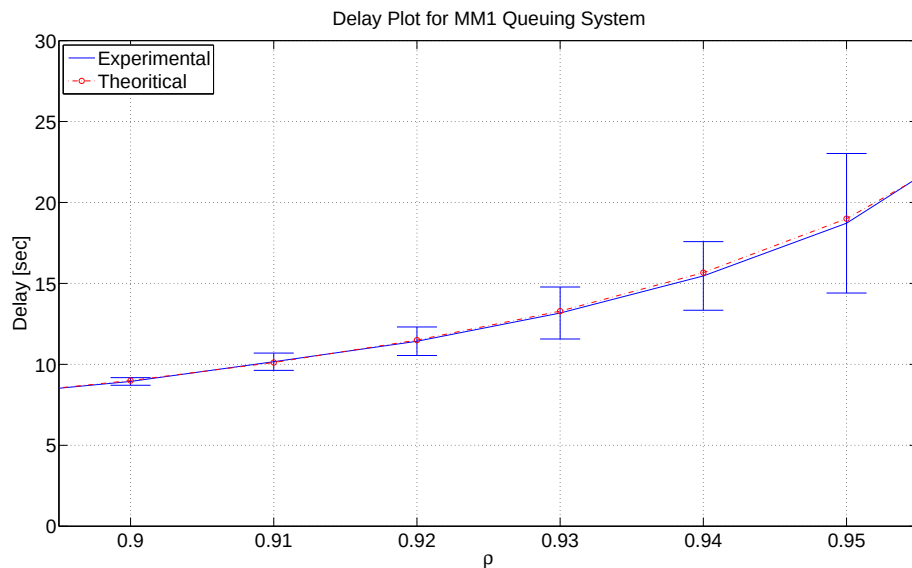


Figure 2.8 Delay for MM1 Queuing System

2.6.2 Queuing Simulation Vs. Theoretical Values

The objective of this section is to show that CAC_SIM provides valid data. Mathematical models are available for certain simple queuing simulations. For example, in a queuing system, delay can be computed if it is the only QoS criteria. Packet loss can be computed for bufferless system (not for queuing system) and there is no existing formula for jitter.

M/G/1 Queue: all M/G/1 that has mean $E(X)$ and variance $Var(X)$ has mathematical model for delay (details in section 2.6.2.1). Thus these equations can be used to test CAC_SIM. The simplest model would be the M/M/1 queue. Both arrival rate and service rate are exponential distribution, and there is only 1 server.

Figure 2.8 is a simple M/M/1 queuing system simulation. The x-axis in the plot is ρ ($\lambda/\mu = arrivalrate/servicerate$), and the y-axis is delay in second. Each dot in the plot is the mean value for 100 repetition of 10000 second length simulation. Both figure 2.8 and 2.9 show that the closer ρ approaches the value 1, the greater variance value will be, and this huge variance does only occur after ρ is larger than 0.95. Figure 2.9 also shows that the closer ρ approaches the value 1, the larger differences between theoretical

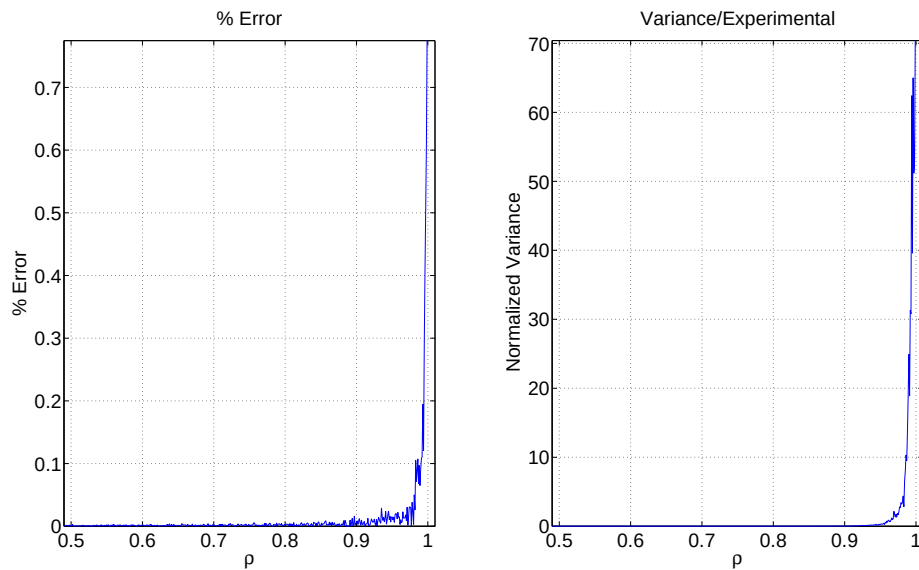


Figure 2.9 % Error and Variance for M/M/1 Queuing System

and simulation values will occur, and this also only occurs for ρ greater than 0.95.

In order to compensate this error, when ρ approaches the value 1, the simulator must either extend the simulation run time or increase the number of repetition. Good news is that the greater the error, the greater the variance based on the result from figure 2.9. Thus, the simulator can use the variance to verify the need of running extra simulations.

Another useful information is that this sort of large error only occurs when ρ is getting really closer to 1. Recall from chapter 1, in queueing theory, each traffic flow is independent and additive. In all the admission region simulations, the summation of all λ from every traffic flow would still be far smaller than the value μ as long as all QoS criteria are within their boundary; therefore, the ρ value can never reach any point closer to 1 for a stable queueing system. In our simulations, the maximum value of ρ is often between 0.55 and 0.85 (typically 0.7-0.8) at the edge of admission region (right on the admission region curve), which all are below 0.95.

2.6.2.1 M/G/1 Single Queue

In this section, we will describe simple M/G/1 single queue system and its simplest variation M/M/1 queue, which was used to validate the delay output from the simulation [32]. In the simulation, many parameters could be adjusted in the “MAIN” testing program without changing the rest of the codes. The 3 most important adjustable parameters are the arrival rate (λ) in [packet/sec], average packet length (l) in [kB] and the average servicing rate ($\tilde{\mu}$) for a server in [kB/sec]. Thus, the service rate $\mu = \frac{\tilde{\mu}}{l}$, and the unit will be [packet/sec]. Let ρ be the ratio between λ and μ , then we obtain the following:

$$\rho = \frac{\lambda}{\mu} \quad (2.7)$$

We only take waiting time (W) and the transmission time (T_{trans}) into account. Other delays, such as propagation time, will be neglected. Thus, the total delay $T = W + T_{trans}$. Since $T_{trans} = \frac{1}{\mu}$, the total delay would be:

$$T = W + \frac{1}{\mu} \quad (2.8)$$

Let R to be the remaining servicing time for a customer, then

$$W = \frac{R}{1 - \rho} \quad (2.9)$$

Knowing $R = \frac{1}{2}\lambda E[X^2]$ and $E[X_i^2]$ is the 2^{nd} moment of the random variable distribution $f(X)$, where X is the interval of time for service, we obtain the Pollaczek-Khintchine mean formula:

$$W = \frac{\frac{1}{2}\lambda E[X^2]}{1 - \rho} \quad (2.10)$$

Based on variance of X ($VAR[X]$) and expected value of X ($E[X]$), we can compute $E[X^2]$:

$$E[X^2] = VAR[X] + E[X]^2 \quad (2.11)$$

For a single source, single priority single server queuing system with arrival rate to be exponential, as long as the 2^{nd} moment of the service time could be computed, there will always be an analytical solution to the delay. For example, M/M/1, since the 2^{nd} moment is $\frac{2}{\mu^2}$, the waiting time and total delay will be:

$$W = \frac{\lambda \frac{1}{2} \frac{2}{\mu^2}}{1 - \rho} = \frac{\lambda}{\mu(\mu - \lambda)} \quad (2.12)$$

$$T = \frac{\lambda}{\mu(\mu - \lambda)} + \frac{1}{\mu} = \frac{1}{\mu - \lambda} \quad (2.13)$$

2.7 Conclusion

In this chapter, we have illustrated our simulation model CAC_SIM and the simulation process. We have also described the problems encountered during the simulation and the techniques to solve them. At the end, we validated one of the QoS criteria (delay), which is the only QoS that there is analytical formula for buffered system. The inputting packets for CAC_SIM would be generated from specific source generators, which will be discussed in chapter 3.

CHAPTER 3

TRAFFIC SOURCE GENERATORS

This chapter will fully describe different network traffic source generators used in the research, and what type of traffic these generators can generate. There are mainly 4 types of traffic used in this research: audio traffic source, video traffic source, data traffic source and simple form testing traffic source (e.g. M/M/1 traffic source). Some variations of these traffic sources will also be introduced. In order to simulate these traffic sources, 3 traffic source generators were created: continuous traffic source generator, ON-OFF traffic source generator and video traffic source generator. In this chapter, first, there will be an overview about different types of traffic, describing the characteristics of each traffic type. Followed up by different type of traffic source generators. In which, there would be detailed descriptions on the mechanism of each generator, how to use each generator to create related types of traffic, and generator validation by comparing the generated traffic sources and theoretical values or trace files.

3.1 Overview of Traffic Sources

There are 3 large network traffic classes: audio, video and data. Each traffic source has its own properties at the packet level, such as packet size (header+payload) and packet arrival rate. Each traffic source also has its own service level agreements for each QoS criteria (delay, jitter and loss). Understanding the characteristics of each type of traffic would help in making better programs (traffic source generators) to emulate the network traffic flows. On the other hand, knowing the service level agreements (SLA) of each type of traffic would help in making accept/reject incoming traffic decisions during a simulation.

Application	Class	Priority	Delay	Jitter	Loss	Payload
High Quality VoIP	Audio	EF	150ms	20ms	1%	10-240B
Low Quality VoIP	Audio	EF	400ms	20ms	1%	10-240B
Voice Message (Playback)	Audio	AF	1s	20ms	1%	10-240B
Voice Message (Record)	Audio	AF	2s	20ms	1%	10-240B
Streaming Audio	Audio	AF	10s	20ms	1%	10-240B
High Quality Videophone	Video	AF1	150ms	20ms	1%	Vary
Low Quality Videophone	Video	AF1	400ms	20ms	1%	Vary
High Quality LiveTV	Video	AF1	150ms	20ms	1%	Vary
Low Quality LiveTV	Video	AF1	400ms	20ms	1%	Vary
VoD	Video	AF	1s	20ms	1%	Vary
VoD	Video	AF	2s	20ms	1%	Vary
VoD	Video	AF	10s	20ms	1%	Vary
Command/Control	Data	AF	250m	NaN	1%	1000B
Interactive Game	Data	AF	200ms	NaN	1%	<1000B
General Data Transfer	Data	AF4	2s	NaN	1%	<1480B

Table 3.1 Example of QoS Control Factors for Some Traffic Sources (Applications)

In the research, there is a more detailed way of grouping traffic groups from ITU recommendations [13, 14, 15, 16, 17]. Column 1 of the table 3.1 lists the traffic groups studied in the simulations, along with some the characteristics of each traffic source (such as packet payload size and common priority setting) and the corresponding service level agreements that have been discussed in chapter 1, especially regarding the maximum values of delay, jitter and loss.

One comment is that video packet size depends on video frame size, and the size of each frame has a different values. Thus the average payload size is not available. The average frame sizes are however known for different type of video files, which will be discussed in section 3.2.3.1.

3.1.1 Audio Traffic Source

There are many audio applications via Internet connections. They can be classified into 3 categories: conversational voice, voice messaging and streaming audio. Here is some descriptions of each category [13].

For conversational voice, the best known application is Voice over IP (VoIP). It requires low delay, low jitter, low loss QoS, which makes it always a high priority, if not the highest priority application in a differential service network.

As for voice messaging, for example, it could be playing or recording a voice mail. the QoS required is a little lower than conversational voice. Moreover, recording an audio message is less sensitive to delay than playback, because only one human being is sensing the QoS during the playback in a one-way transmission.

Streaming audio requires a low QoS in terms of delay, but it is still sensitive to jitter and loss.

3.1.1.1 Audio Codecs

Most simulations use *G.711* Codec, which has a bit rate of $64Kbps$ and a packet payload size between $10B$ to $230B$, with an increment of $10B$ (B is byte). Adding $20B$ header, the resulting packet size would be ranged from $30B$ to $250B$. In this research, the payload size of $10B$, $20B$, $40B$, $80B$ and $160B$ have been simulated. Sometimes, in order to see how the trend goes with larger packet size, $240B$ and $1480B$ (not used in real implementation) have also been tested. Since both packet size and traffic flow bit rate are known, the packet arrival rate and the inter-arrival time can be easily computed [33].

For a speech, the ON time is when a person speaks, and the OFF time is when the person

is muted (in silence). Each ON period of time is followed by an OFF period, and each OFF period is followed by an ON period. Each ON/OFF period may have different length in time. Hence, random variables are involved. From the studies, the length of these periods are exponential distributions[34].

3.1.1.2 QoS for Audio Traffic Source

In terms of QoS, normal speech does not require very high audio quality. However, if someone plays musics on one side of a phone conversation, the other side would hear a strange sound due to the low bit rate originally designed for conversational purposes only. For some applications such as online virtual music theater, not only the traffic flow bit rate would be greatly increased (greater throughput), the demand on higher QoS performance would also be very different from the simple VoIP.

Back to VoIP, in most cases, there is no packet during the off period. Except when using ATM network, there would be 53Byte size cell transmission even during the idle time. Since all simulations are done in the IP network, the ON-OFF generator is being used to simulate VoIP. The service level agreement of different speech applications are listed in table 3.1.

3.1.2 Video Traffic Source

Video traffic flow bit rate is much larger than the audio traffic flow bit rate, especially for high resolution images. There are many factors influencing the characteristics of video traffic source, such as resolution, codec compression, type of movie, etc. Sometimes, the type of movie (action movie or lecture room) may have a huge differences on the throughput and the variation of throughput, especially for some high compression ratio codecs. For example, many codecs use one picture from a scene as a reference of several following up pictures. An action movie would have a much larger average frame size

compared to a slow motion lecture room video clip or a surveillance camera recording video of the same resolution.

3.1.2.1 Video Codecs

The goal of using video codec is to compress larger video files to a smaller size, so that less amount of data required for network transmission while still getting original resolution. In the simulations, MPEG-4 codec is being used. It belongs to Motion Picture Experts Group (MPEG) family, MPEG-1 as VCD using Layer 3 MP3, MPEG-2 as most DVD and broadcast-quality TV, MPEG-7 as formal system for describing multimedia content. MPEG-4 is an extension of MPEG-1, which allows low bit rate encoding. Nowadays, MPEG-4, along with H.263, RealMedia Variable Bitrate (.rmvb) and blue-ray (.mkv) cover most video files on the internet.

The color space (e.g., RGB, YUV) has no big influence on the traffic source generations at the packet level. However, video codec compression will. For example, some high compression ratio codecs may cause video traffic flows more bursty.

The main challenge for designing a video source generator is to generate frames. After this challenge is being solved, breaking each frame into packets would be much simpler.

3.1.2.2 Video Frames and Scene

A video file is literally a series of pictures shown on a screen at a very higher rate so that naked human eyes cannot catch the transition of 2 consecutive images. Each of these pictures is called Video Frame or simply Frame. A scene is composed by a number of sequential frames.

Similar to other types of video codecs, MPEG-4 compresses a video file by referring some video frames to some other frames belonging to the same scene, in which only

1 frame from each scene has the native resolution (without compression). The rest of the frames from the same scene only have to record the pixels that are different from the reference frame. Of course referring similar images would save more recording space. If 2 pictures are identical, the second picture may require nearly zero space. MPEG-4 has a highly efficient compression while having good quality video, thus less overall data required for transmission. MPEG-4 treats each scene as an independent video object (VO). It also treats each frame as a video object plane (VOP). There are 3 types of frames being used [35].

- Frame I (Intra-Coded Frame): It contains all information for every pixel in the picture corresponding to this frame.
- Frame P (Predictive-Coded Frame): It records the differences between the corresponding picture and the previous frames I or P.
- Frame B (Bidirectional-Predictive-Coded Frame): It records the differences between the corresponding picture and both previous and future frames I and/or P.

Thus, frame I would have very large size according to the original picture. Frame P would have a relative smaller frame size, while frame B would be even smaller.

The frames in each scene is followed one after another in a sequence. Some commonly used frame sequences are III..., IPIPI... and IBBPBBPBB...IBBPBBP.... I, B and P are the symbols for 3 types of frames. For fixed scene length, each periodic scene with specific frame sequence is called group of pictures (GOP). In this research, the sequence being used is a fixed sequence with scene length being 12: IBBPBBPBBPBB. The frame inter-arrival time is set to be 40ms. These settings are based on the video trace files from real MPEG-4 videos [36].

3.1.3 Data Transmission Traffic Source

Here, there are mainly 2 data traffic sources: data transmission and data command file. The data transmission traffic source is the transmission of data files after initialization protocols (e.g., HTTP, SIP) are established. For example, once a P2P connection is established between 2 ends, data are transmitted in a constant rate. On the other hand, data command file are meant for signalizations. For example, it could be used for initialization protocol, computer long distance remote control, online multiplayer gaming, or any online interactive activities.

The data transmission traffic source is simulated using continuous source generators, which could be a constant bit rate (CBR) traffic source. The data command file has a variable bit rate (VBR). Most often, the arrival rate is exponential. For instance, HTTP traffic can be simulated by VBR source using exponential arrival distribution [37].

In this research, all data transmission traffic sources are simulated by CBR, and all data command file traffic sources (such as HTTP) are simulated by VBR with exponential arrival distribution. Both CBR and VBR traffic sources are continuous source generators. We use the word “continuous” to differentiate from ON-OFF source generators. Basically, continuous source generators never stop sending packets (i.e., there is no OFF period).

3.2 Sources Generators

Most traffic sources are emulated by heavy-tail distributions for the arrival time and packet size (if not fixed). Since the simulation is done in IP networks, the maximum packet payload is 1480B for IPv4. With the addition of a 20B header, the maximum packet size is 1500B for IPv4.

In order to simulate different types of traffic sources such as audio, video, data and test

(e.g. M/M/1 traffic source), several types of source generator were created. There are 3 types of generator used in this research. The first one is the continuous traffic source generator used for validation testing purpose and to create CBR data traffic flows. It can also serve as background traffic flows. Another type of generator is the ON-OFF traffic source generator, which is mainly responsible for generating audio traffic flows such as VoIP. Finally the video traffic source generator is solely designed for MPEG-4 video traffic sources.

3.2.1 Continuous Traffic Source Generator

The continuous traffic source generator generates packets from the beginning until the end of each simulation session, without any idle time. It can generate any types basic traffic sources such as CBR, VBR and M/M/1, with given average packet size, average arrival rate, packet size distribution, arrival rate distribution, type of traffic source and DSCP.

In this research, the continuous traffic source generator is used to create data traffic flows and testing traffic flows. Data transmission (e.g. downloading a large file) traffic source can be simulated by a CBR source. Data command file (e.g. HTTP, remote control, online gaming) traffic source uses VBR source with exponential distribution. Testing traffic sources could be any traffic with simple distributions such as M/M/1 or M/D/1.

CBR source only emits a constant size packet every constant period of time through out the entire session. For example, normal data transfer would be sending packets with 1480B maximum payload at a constant rate. M/D/1 source would be sending constant size packets in variable rate. The packet inter-arrival time would be exponential. M/M/1 source would also have the packet inter-arrival time to be exponential, and the packet size distribution to be exponential as well.

Thanks to the SSJ library, it is not hard to generate random variables in most well known

distributions such as exponential distribution, lognormal distribution, pareto distribution, normal distribution, etc. We are not going to show the plots for every single distribution because SSJ has already verified all these random variable generators [29].

3.2.2 ON-OFF Traffic Generator

This generator is mainly used to simulate audio, especially for voice (speech). It requires extra amount of information regarding the durations of on and off periods. During an on period, the constant packet transmission rate is called peak rate. It generates constant packet size and constant packet arrival rate (peak rate) for each on period, just like a CBR traffic flow. However, the length of each ON period (busy time) and OFF period (idle time) is exponential.

The ON-OFF traffic generator is mainly designed for voice traffic sources. The goal is to be able to emulate normal human speeches. A normal human speech has an average of 0.352s ON time and 0.65s OFF time. Therefore, theoretically a normal voice conversation would have 0.352s (35.1%) average on period and 0.65s (64.9%) average off period.

Again, we are not going to test the random variable generations because SSJ has already tested its exponential variable generator [29]. Here, we are going to test whether the generator can generate ON-OFF traffic sources with the average length of ON and OFF time within the normal speech range during a 10000s long simulation session.

In our test, a single audio ON-OFF traffic source has created 11800 packets during a 669.6s length simulation. For a 64Kbps bit rate, 160B payload size and using G.711 audio codec, the audio traffic flow would have a packet inter-arrival time of 20ms for 8kHz speech signal during the on period. We can see here, $11800 / (669.6s / 0.020s) = 35.2\%$, which is a match to the 35.1% on period. For a longer simulation of 5000s, figure 3.1 shows that after a certain amount of time (around 400s), the percentage of “ON” time

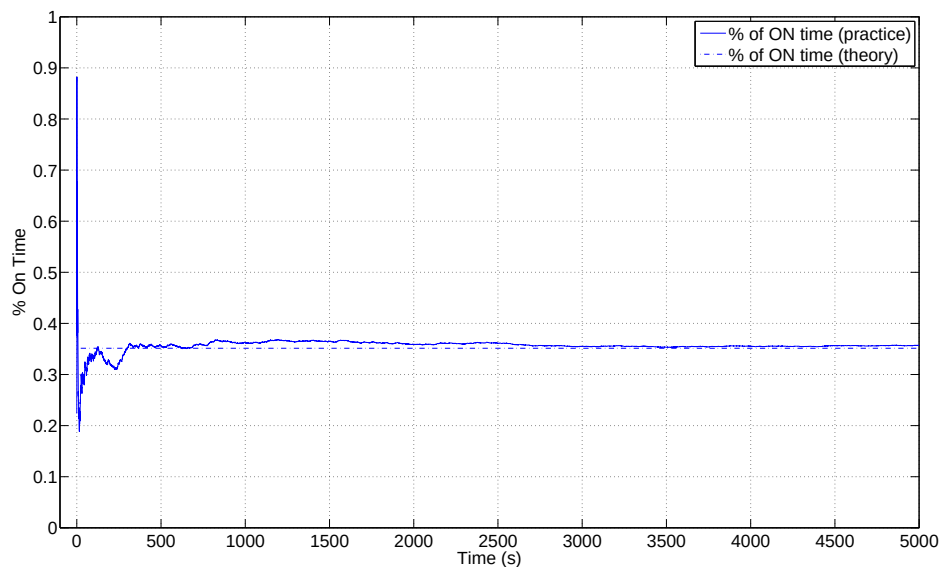


Figure 3.1 Audio ON-OFF Traffic Source Generator: x-axis is the time t , y-axis is the real amount of packets being sent divided by the maximum amount of packets possibly being generated if there is no OFF period between time 0 and time t

would reach a steady state, which is close to the theoretical value 35.1%. Notice in this research, each simulation session has a duration of 10000s, which is much longer than 400s.

3.2.3 Video Traffic Source Generator

The video traffic source generator used in this research is specifically designed to generate MPEG-4 video traffic flows. Video traffic source frame sizes are often pareto or lognormal distribution. Here, the lognormal distribution has been used [38, 39]. The distribution parameters are based on some real video trace file measurements [40, 41, 36].

Video File	Quality	E(I)	VAR(I)	E(B)	VAR(B)	E(P)	VAR(P)
Lecture Room	Low	1.7446	0.0690	0.0485	0.0007	0.1160	0.0048
Lecture Room	Medium	1.7446	0.0690	0.1260	0.0048	0.2089	0.0124
Lecture Room	High	3.6453	0.3859	0.7210	0.1178	0.9550	0.1800
Die Hard III	Low	2.9573	1.5760	0.3890	0.1056	0.9774	0.4660
Die Hard III	Medium	2.9573	1.5760	0.8970	0.4913	1.4364	0.9570
Die Hard III	High	6.1305	5.2056	2.8600	3.0962	3.9445	5.0100
Parking Cam	Low	6.8176	0.4431	0.0527	0.0005	0.1877	0.0055
Office Cam	Low	4.6311	0.0304	0.0433	0.0009	0.1046	0.0077
Simpsons	Low	5.4933	1.7124	0.5080	0.3557	1.3678	0.9870
Star Wars IV	Low	1.5158	0.2407	0.0952	0.0077	0.2828	0.0669
Star Wars IV	Medium	1.5158	0.2407	0.2189	0.0296	0.4290	0.1231
Star Wars IV	High	3.3443	0.9715	1.0301	0.2134	1.5149	0.5952

Table 3.2 Size and Variance of Frames I, B and P for Different Video Traffic Sources (e.g., Size of Frame I = E(I), Variance of Frame I Size = VAR(I))

3.2.3.1 Parameters for Video Traffic Sources from Trace Files

There are many ways for video compression, and they do not have to be fixed scene length. However, from all the video trace files we have found, the scene length is fixed to be 12 frames long, and the frame sequence is IBBPBBPBBPBB and repeating [36].

The inter arrival time between each two frames is set to be 40ms for a frame rate of 25frame/s. However, due to the maximum packet size being 1500B for a IPv4 packet (1480B payload size), a frame could be divided into multiple packets all being 1500B, except that the last packet may be smaller. The inter arrival time between 2 packets in the same frame is set to be 1ms. Therefore, a frame should consist less than 40 packets or 0.4736Mbits, thus the maximum throughput is 11.84Mbps. This would not create much interference problem between 2 frames because the simulations were not running for 1080p resolution (may go beyond 100Mbps). For high definition video, a normal digital MPEG-2, HD video would require 30Mbps, but better codecs such as *MPEG – 4* or *H.264* only requires 6-9Mbps [42].

The statistical values that can be extracted from the trace files are mean, variance, minimum, maximum, etc. Table 3.2 lists all the parameters measured from the video trace files. This table also shows that both MPEG-4 low and medium quality have the same I frames (reference frames). Moreover, the I frames under high quality MPEG-4 are not compressed [36].

The video traffic source generator can generate video traffic sources similar to these trace files solely based on these measurable parameters. In section 3.2.3.2 and 3.2.3.3, we will explain how to import these parameters into the lognormal distribution equation and plot the trace file frame size distribution vs the output from the video traffic generators.

3.2.3.2 Lognormal Distribution and Validation Test

Lognormal distribution has been used for generating video frame sizes, and it has its own specific parameters: μ and σ . These 2 parameters cannot be directly measured from the video trace files [36]. The only measurable variables are mean, variance, minimum, maximum, etc. Thus, we must convert the probability density function (pdf) for lognormal distribution into another format, which only depends on the measurable variables. Here is the probability density function for lognormal distribution (3.1), followed by the formula for expected value and variance.

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma x} e^{-(\ln(x)-\mu)^2/(2\sigma^2)} \quad (3.1)$$

$$E(X) = e^{\mu+\sigma^2/2} \quad (3.2)$$

$$Var(X) = (e^{\sigma^2} - 1)e^{2\mu+\sigma^2} \quad (3.3)$$

The expected value (mean) and the variance can be measured, but μ and σ cannot. In order to obtain the parameters μ and σ based on mean and variance, the equations 3.2 and 3.3 can be derived into forms shown in equations 3.4 and 3.5:

$$\mu = -\frac{1}{2}\ln\left(\frac{Var(X) + E(X)^2}{E(X)^2}\right) + \ln(E(X)) \quad (3.4)$$

$$\sigma = \sqrt{-2\mu + 2\ln(E(X))} \quad (3.5)$$

3.2.3.3 Emulation of Video Traffic Source with Lognormal Distribution

Currently lognormal is the most used distribution for generating video frame size. The goal in this section is to test how well can lognormal distribution emulates the video traffic source frame size generation.

Figures 3.2, 3.3 and 3.4 are using histogram plots to show the slight differences between the video traffic source generator outputs and the real trace file data. Both of them are using the same number of bins and bin width on the histograms. Knowing a bin of a histogram is the non-overlapping interval. For example, 0-30 can be divided into 3 bins of identical size, which would be 0-10, 10-20 and 20-30.)

The number of bins for the histogram is chosen to be 100; therefore, the bin width (h) is the difference between the maximum and the minimum frame size divided into 100 pieces, i.e. each two values next to each other in the histogram x-axis have an equal difference h .

The parameters in figures 3.2, 3.3 and 3.4 are based on a video trace file “Star Wars IV MPEG 4 low quality”. The histograms are based on the raw data extracted from the trace file, and the curves are the simulation outputs using lognormal distribution. Although

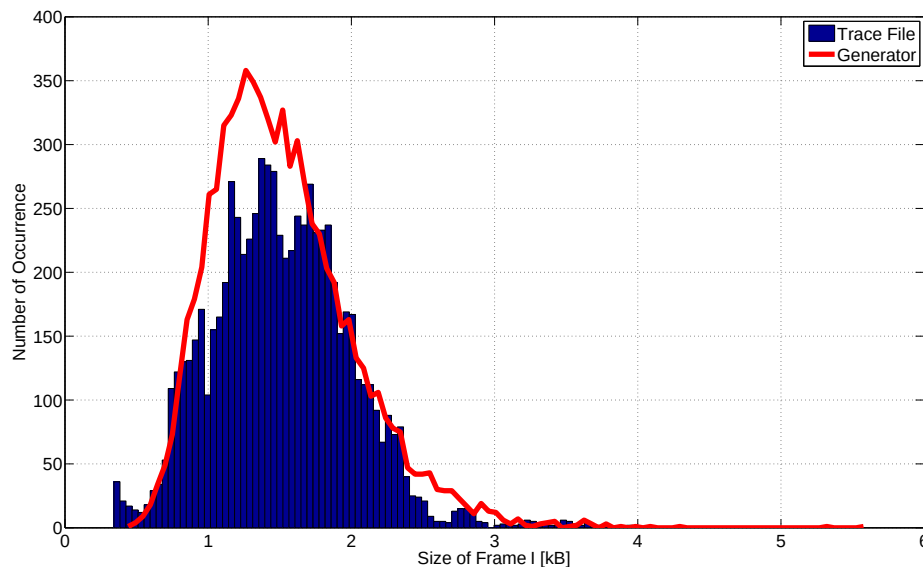


Figure 3.2 Frame I Size Distribution: Generator Outputs (curve) and Trace File (histogram)

there is still a little difference between simulated values and video trace file, especially for each frame peak values, in general, lognormal distribution can be an approximation for video traffic source frame size generation.

3.3 Conclusion

In this chapter, we have listed the traffic sources that would be used in the simulations. We have also described the source generators in terms of structure, functionalities and parameters (either extracted from real trace files or suggested by other researchers). For some traffic generators that are not simple in design, we have also validated the generator outputs by comparing to known measurable parameters or to existing trace files found in other research papers. Starting from chapter 4, we will begin to show results regarding the admission region in DiffServ network.

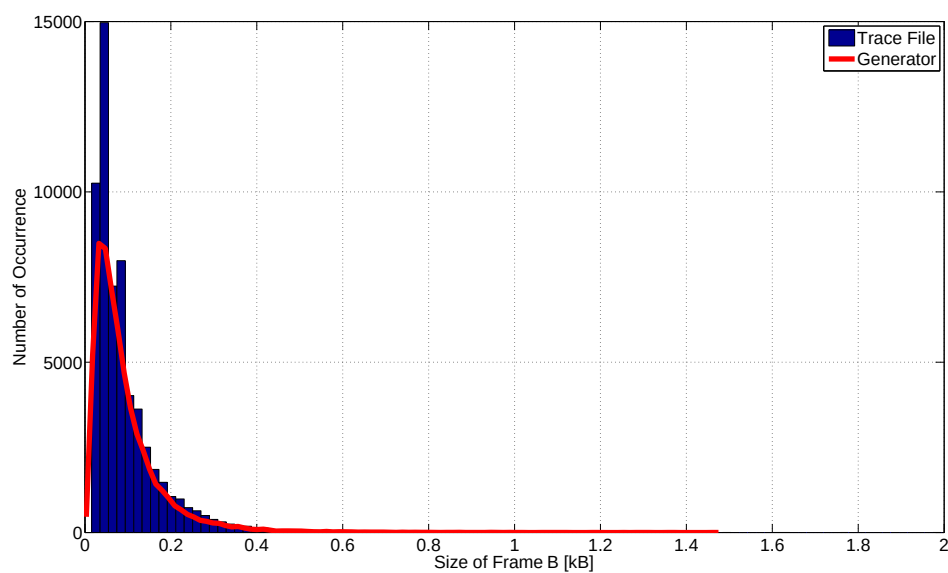


Figure 3.3 Frame B Size Distribution: Generator Outputs (curve) and Trace File (histogram)

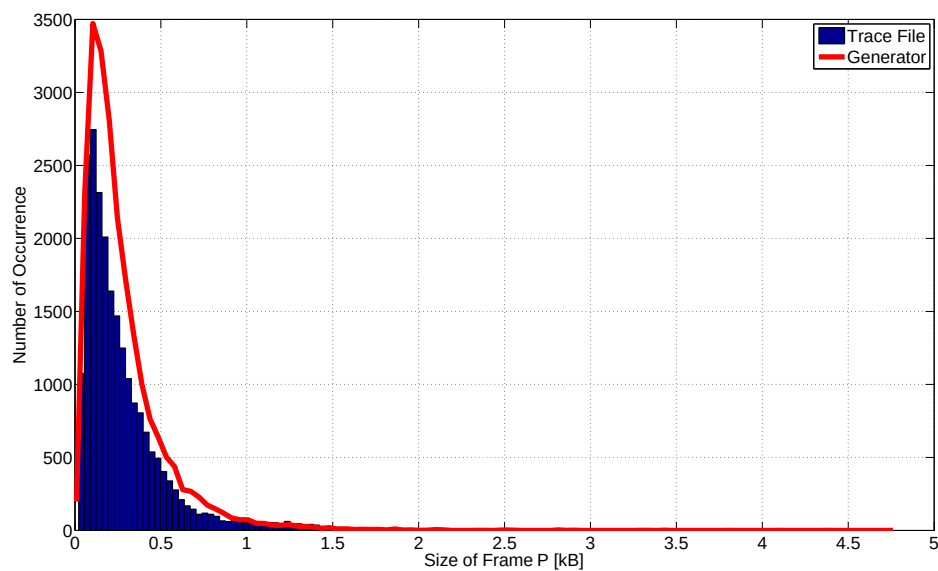


Figure 3.4 Frame P Size Distribution: Generator Outputs (curve) and Trace File (histogram)

CHAPTER 4

ADMISSION REGIONS WITH DIFFSERV

Recall that admission region (AR) is an area with satisfactory quality-of-service (QoS) for all types of traffic, and it is used for purpose of connection admission control (CAC). This chapter will concentrate on how different queuing disciplines affect the admission region. For each pair of traffic types, there will be admission region plots based on different queuing disciplines. There are mainly 3 queuing disciplines used in this chapter: priority queuing (PQ), round robin (RR) and weighted round robin (WRR).

4.1 General Parameters and Simulation Procedure

This section will first define the parameters of all traffic types used in this chapter. Then, it will list all the different queuing disciplines simulated in this chapter. At the end, there will be a brief description of all the general settings for the simulation process.

4.1.1 Parameters for Each Type of Traffic

The highest priority audio traffic source is VoIP-EF, which is generated by the on-off traffic source generator. Assuming the average of 0.352s ON time, the audio traffic source is encoded using 64Kbps bit rate G.711 audio codec with packet size of 180B (160B payload and 20B header) and 0.65s OFF time, and the packet inter-arrived time is 20ms. The QoS agreement is 150ms delay, 20ms jitter and 1% overall loss.

The second highest priority video traffic source is LiveTV-AF41, which is the movie StarWars IV, encoded in MPEG-4 low quality. The details about I, B and P frames were discussed in chapter 3. The QoS agreement is 400ms delay, 50ms jitter and 1% overall

loss.

The lowest priority data traffic source is Data-AF41, which is generated by the continuous traffic source generator. It is simulated by a $64Kbps$ CBR traffic source with a fixed packet size $1500B$ ($1480B$ payload and $20B$ header), and the packet inter-arrival time is $185ms$. The QoS agreement is $2.0s$ delay and 1% overall loss. There is no QoS requirement for jitter.

Data command/control file traffic source DC-AF11 is also considered as second highest priority. Recall that data command/control file traffic source (initialization protocol, computer long distance remote control, online multiplayer gaming, etc.) is used for signalizations. It has an average packet size of $1000B$. In order to keep the throughput to be $64Kbps$, the sending rate is assumed to be 8 commands (signals) per second. The QoS agreement is $250ms$ delay and 1% overall loss. There is no QoS requirement for jitter.

4.1.2 Queuing Disciplines and Queuing Policies

3 types of queuing disciplines have been applied. They are priority queuing (PQ), round robin (RR) and weighted round robin (WRR). For WRR, we will test 3 different weight ratios, 8:1, 4:1 and 2:1. Therefore a total of 5 different queuing disciplines (5 curves) will be shown on the admission region plot.

Recall that all the simulations were run under the non pre-emptive mode. Each single queue is FIFO with drop tail as the dropping policy. Lower priority queue has larger buffer size (2 times the buffer size for each priority lower).

4.1.3 Simulation Procedure Settings

The simulation length is $10000s$ for each repetition. On the admission region plot, each point or configuration (n_1, n_2) has the meaning of having n_1 traffic type 1 and n_2 traffic type 2 in the network. Recall from chapter 2, if either type of traffic fails its QoS test, the configuration would be rejected, and the simulation would be restarted with less number of traffic type 2 and the same number of traffic type 1. For the simulations nearby the admission region boundary, each configuration at least repeat 32 times (minimum number of iteration to obtain a reliable variance value for the Wald sequential hypothesis testing discussed in chapter 2), and at most 1000 times (although it does not happen often, but this maximum can avoid certain computer deadlock run time errors). Using the sequential hypothesis test, the number of repetitions is often around 100 in our simulations.

4.1.4 Excluding the Configuration with Zero Lower Priority Traffic Flow

The configurations with either $n_1 = 0$ or $n_2 = 0$ would not appear on the admission region plots shown in the future, where n_1 and n_2 are the numbers of higher and lower priority flows respectively. The reason is because the zero values are often the peak values. Here is an example illustrated in figure 4.1, which is same as figure 5.12, VoIP-EF vs LiveTV-AF11, except it includes the initial zero value. Other comments beside the zero value will be discussed in section 5.3.2.

In the example shown in figure 4.1, for $n_2 = f(n_1)$, when $n_2 = 1 \rightarrow 0$ (cursor 1), n_1 increased about 10 units. This means that between zero lower priority flow and one lower priority flow, the maximum higher priority flow allowed could be dramatically changed.

The main reason for this to happen is because when DiffServ has been introduced into the network simulation, higher priority applications would be grant much more service proportion to lower priority applications, especially under priority queuing. Thus lower

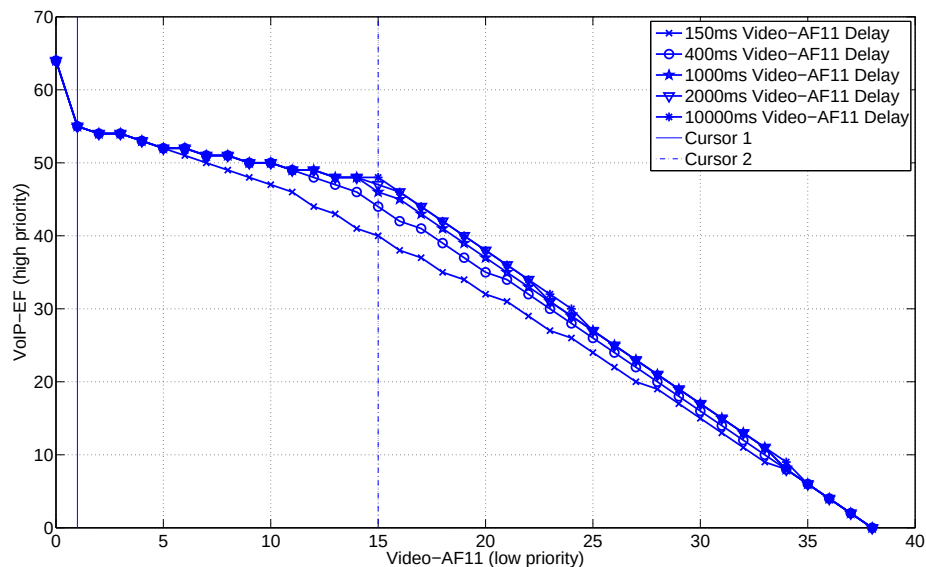


Figure 4.1 Admission Regions (Including Zero Lower Priority Traffic Source): LiveTv-AF11 (StarWars IV MPEG-4 low quality) vs Data-AF41 (1480B Payload) for Different QoS under PQ

priority QoS would be more likely to be the binding factors that cause a configuration to fail the QoS test. When there are many higher priority flows and only 1 lower priority flow in presence, it is still very likely that the lower priority flow fails the QoS test before any higher priority flow failed their QoS test. However, when there is no lower priority flow, suddenly only the high priority traffic QoS matters. In other words, the sudden change in binding factor causes the spike.

Going a little further, even if there is no lower priority flow, there would still be a QoS for lower priority traffic flow. In other words, if introducing a very small amount (ϵ) of lower priority packets into the network, these packets would experience the lower priority QoS. However, in practice, this cannot be done in a simulation because fewer packets would cause much greater error in statistics. Therefore, for simplicity, this point has been removed from all the admission region plots.

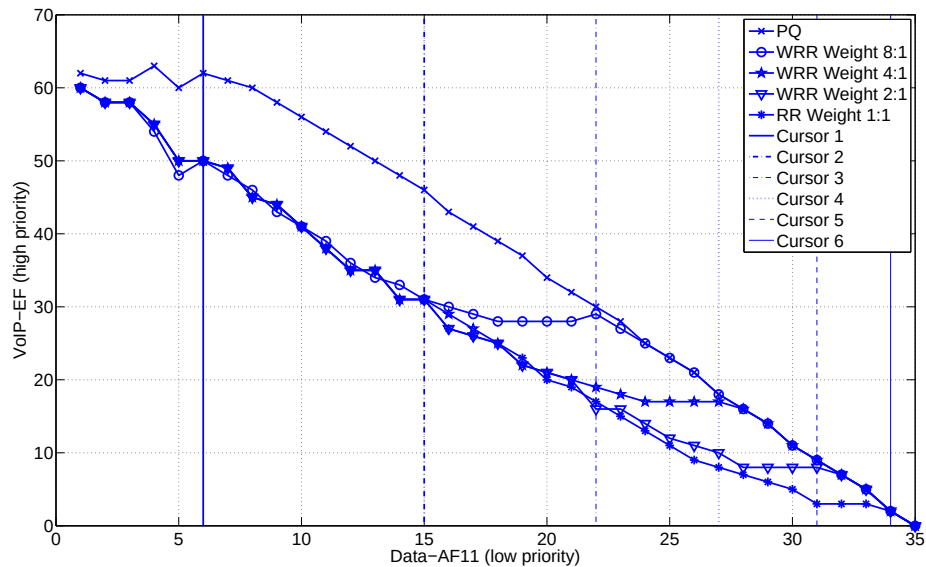


Figure 4.2 Admission Regions: VoIP-EF (G.711, 160B Payload) vs Data-AF41 (1480B Payload) for Different Queuing Disciplines

4.2 Admission Region for VoIP and CBR Data Traffic Sources

For higher priority VoIP-EF traffic source vs lower priority CBR Data-AF41 traffic source, figure 4.2 illustrates the differences between 2 extremely different queuing disciplines in DiffServ networks. While priority queuing (PQ) has strict higher priority serve first rule, round robin (RR) provides fairness. Weighted round robin (WRR) on the other hand is in between PQ and RR. It uses a weight system to balance the priority system and the fairness requirement. Thus, as expected, the WRR admission region curves are in between PQ and RR curves.

4.2.1 Admission Region for VoIP and CBR Data Traffic Sources under PQ

From figure 4.2 PQ curve, in the section at left of cursor 1, where there are less than 6 lower priority CBR Data-AF41 sessions, the higher priority VoIP-EF sessions experience delay, jitter and loss (drop). Both delay and jitter for the VoIP-EF traffic flows are the

binding factors for this small area of admission region. The amount of packets being dropped is less than 10% of the packets experiencing high delay/jitter. Moreover, if a packet experienced delay greater than the service level agreement, this packet would also have a jitter greater than the service level agreement based on the fact that the loss due to delay, the loss due to jitter and the loss due to either delay or jitter have the same value. In other words, if we consider the delay as the only QoS requirement, we would still obtain the same admission region for this section of the entire admission region plot.

In the section at right of cursor 1, where there are greater than 6 lower priority CBR Data-AF41 sessions, all binding factors are from lower priority traffic flows (i.e. only lower priority traffic source service level agreement would affect the admission region). More specifically, the only binding factor in this area is the lower priority traffic source delay, which is shaping the admission region. Again, recall that the lower priority queue buffer is much larger than the higher priority queue buffer (almost 0 packet drop), and that CBR data traffic source does not care about jitter.

One more comment is that this horizontal line (in the section at left of cursor 1) would not appear if the higher priority traffic QoS is the binding factor when there are very few lower priority flows. There are 2 possible explanations. First, lower priority flows may affect higher priority traffic flows due to non pre-emptive mode, if lower priority traffic flows have much greater packet size than higher priority packet size. Recall that non pre-emptive mode allows server to continue servicing a lower priority packet if the already started the transmission, even if there is a higher priority packet arrives during this period of service time. Thus, non pre-emptive mode would let lower priority traffic flows to create a small influence on the higher priority traffic flows. Yet, this effect seems to be too small to be visible on the admission region in most cases.

Another explanation is that there are simply too many higher priority flows compared to the number of lower priority flows, so that there are so many holes in the timeline where there are no higher priority packets in the queue. Thus the lower priority traffic flows can

have acceptable QoS if only a small number of lower priority flows in presence. When this is happening, the left section of the admission region would often be a horizontal line. If the QoS agreement for higher priority flows is too strict compared to the lower priority QoS agreement, it may also result in the same way (horizontal line).

In general the entire admission region is mainly based on the delay constraints. The flat region (at left of cursor 2) is mainly due to VoIP-EF delay and the negative slope region (at right of cursor 2) is due to Data-AF41 delay. These 2 regions together form a trapeze, which supports Arthur W. Berger & Ward Whitt's research [28]. Recall that their experiment for PQ was based on the fact that the delay is the only QoS factor and that the packet size is constant (constant cell size in ATM network). Yet VoIP-EF and Data-AF41 packets are fixed 180B and 1500B respectively, and the delay is the binding factor. Although there are some off periods for VoIP-EF traffic flows, in general it is similar to the ATM network with delay as the only QoS factor. The pivot point on the trapezoid is where the binding factors changes hands. This will be explained in section 4.2.3 and figure 4.4.

4.2.2 Admission Region for VoIP and CBR Data Traffic Sources under RR

The same figure 4.2 shows that RR is more a linear shape. The binding factor is the VoIP-EF delay at the left section of cursor 6 (nearly the entire admission region). At right side of cursor 6, CBR Data-AF41 delay is the binding factor, and RR curve joins PQ curve. This will be explained in section 4.2.3. For other QoS, there is no significant amount of drops for audio packets, nor any unacceptable jitter behavior before delay breaks the service level agreement.

RR is like WRR but each type of traffic has equal weight 1. RR has less admission area than PQ if higher priority QoS has greater restrictions on the admission region. This is because if both types of traffic having similar throughput, when higher priority traffic flows demand much higher QoS than lower priority traffic flows, priority queuing would

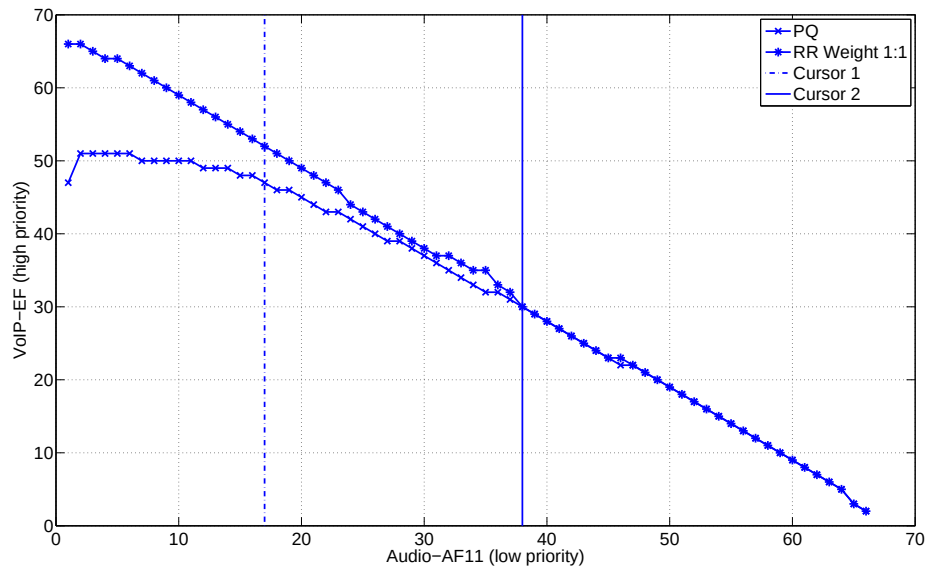


Figure 4.3 Admission Regions: VoIP-EF (G.711, 160B Payload) vs Audio-AF11 (G.711, 160B Payload) for Different PQ and RR

be in favor. Thus, the RR admission region area would be smaller than the PQ admission region in this kind of situation, which is the same situation as in figure 4.2.

On the contrary, if both traffic types have similar service level agreements, RR would be in favor, and it would have greater admission region area than PQ in that case. This would be illustrated by figure 4.3, where both higher and lower priority traffic sources are audios with same codec, same QoS agreement and same properties. For similar service level agreements, if a traffic flow has greater throughput, it has the same effect as having tighter QoS boundaries. It would be hard to make any prediction when both QoS and throughput varies.

4.2.2.1 Admission Region for VoIP and Audio Traffic Sources

Figure 4.3 shows the admission region for higher priority VoIP-EF traffic source vs lower priority Audio-AF11 traffic source under priority queuing and round robin. The RR admission region has a larger area than PQ admission region. This is because both

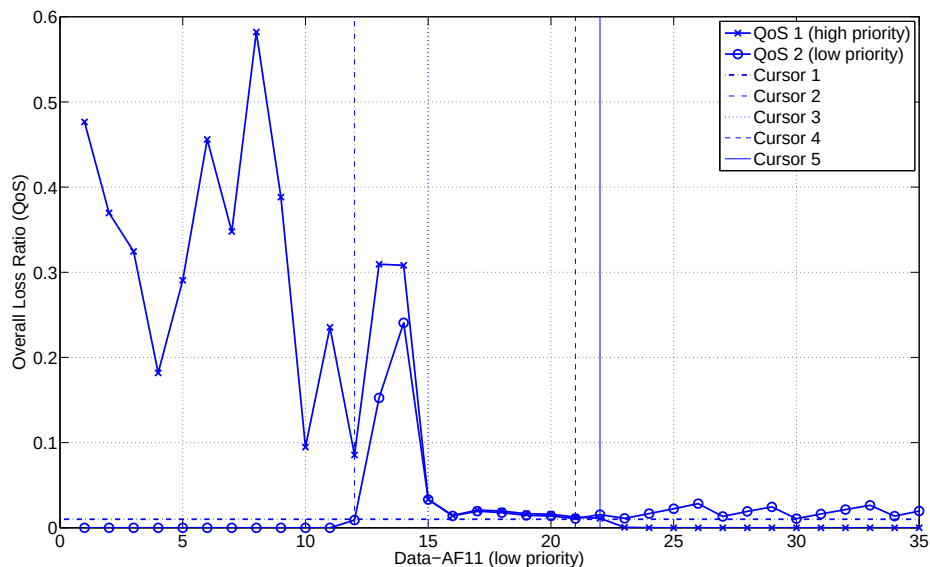


Figure 4.4 Overall Loss Ratios for Both VoIP-EF and Data-AF41 by Adding 1 Extra VoIP-EF Flow to the Curve WRR Weight 8:1 from Figure 4.2

VoIP-EF and Audio-AF11 have the same QoS agreement. In this case, fairness becomes the most important factor in choosing the queuing discipline. Recall from figure 4.2 in section 4.2, where the lower priority Data-AF41 traffic source had much lower QoS agreement than the higher priority VoIP-EF traffic source, and the RR admission region was larger than the PQ admission region.

4.2.3 Admission Region for VoIP and CBR Data Traffic Sources under WRR

The same figure 4.2 shows that WRR curves have certain transition period between the RR curve and PQ curve. This is due to the change in binding factors. When there are much greater number of higher priority sessions than lower priority ones, the higher priority QoS would be the binding factor. There is always a threshold somewhere where low priority traffic QoS becomes the binding factor. This threshold point can vary due to the variations of weight ratios (for WRR), traffic source properties (e.g. throughput) and traffic QoS limitations (e.g. delay). The grouping the points around this threshold

point would be the transition period, where both higher and lower priority flows experience certain QoS failures.

Figure 4.4 illustrates how does the 8:1 weight ratio WRR curve make transition from RR curve to PQ curve. The only horizontal cursor 1 stands for the maximum 1% loss ratio. Anywhere above cursor 1 means failed QoS test. Cursor 2 to 5 divide the plot into 5 sections. Recall the delay is the only binding factor. At left of cursor 2, the VoIP-EF delay is the binding factor, and that the loss due to Data-AF41 delay is 0. In between cursor 2 and 3, the loss due to Data-AF41 delay is above 0 but below 1%. In between cursor 3 and 4, the losses due to VoIP-EF delay and Data-AF41 delay are both above 1%. In other words, both VoIP-EF and Data-AF41 delays are the binding factors. In between cursor 4 and 5, the loss due to VoIP-EF delay is above 0 but below 1%, and that Data-AF41 delay becomes the only binding factor. At right of cursor 5, the loss due to VoIP-EF delay is 0.

The corresponding 8:1 weight ratio WRR curve on figure 4.2 joins the RR curve when there are 15 or less Data-AF41 sessions. This is when VoIP-EF delay is the only binding factor. This WRR curve also joins the PQ curve when there are 22 or more Data-AF41 sessions. This is when Data-AF41 delay is the only binding factor. In between 15 and 22 Data-AF41 sessions, both VoIP-EF delay and Data-AF41 delay are the binding factors.

In general, the WRR curve joins RR or PQ curve due to the change of binding factors, especially from high priority QoS to low priority QoS. Notice that not only WRR curves but the RR curve would also join PQ curve when the lower priority QoS becomes the only binding factor. Another comment is that not only each WRR curve would join the RR curve, each smaller weight ratio WRR curve would also join the larger weight ratio WRR curves when the higher priority QoS becomes the only binding factor.

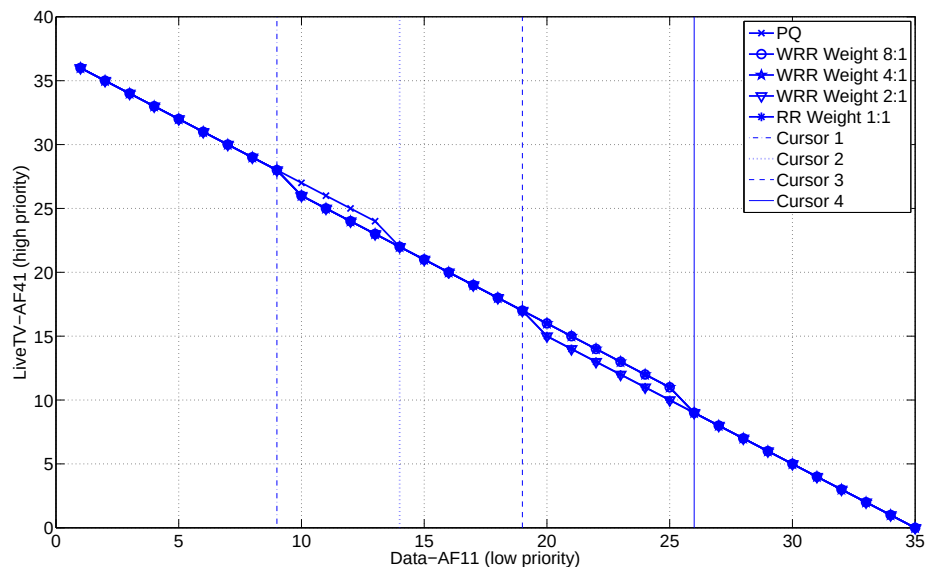


Figure 4.5 Admission Regions: LiveTV-AF11 (StarWars IV MPEG-4 low quality) vs Data-AF41 (1480B Payload) for Different Queuing Disciplines

4.3 Admission Region for Video and CBR Data Traffic Sources

For LiveTV-AF11 vs Data-AF41 in general, PQ and RR make no difference on the admission region plot shown in figure 4.5. Thus, WRR would also have the same admission region because it is in between PQ and RR.

For PQ, the binding factor is always the CBR data traffic source delay because the lower priority traffic flows have very little influence on the higher priority traffic flows. Remind that non pre-emptive has negligible effect on the admission region. Yet delay is still the only binding factor, the video stream packet size has much greater variations than other traffic sources (such as VoIP), and the arrival rate becomes more bursty. Although the delay is the only binding factor, the resulting admission region is linear instead of trapeze for PQ.

For RR and WRR, the CBR data delay is also the binding factor at the right section of the admission region. However, on the left section of the admission regions, the binding

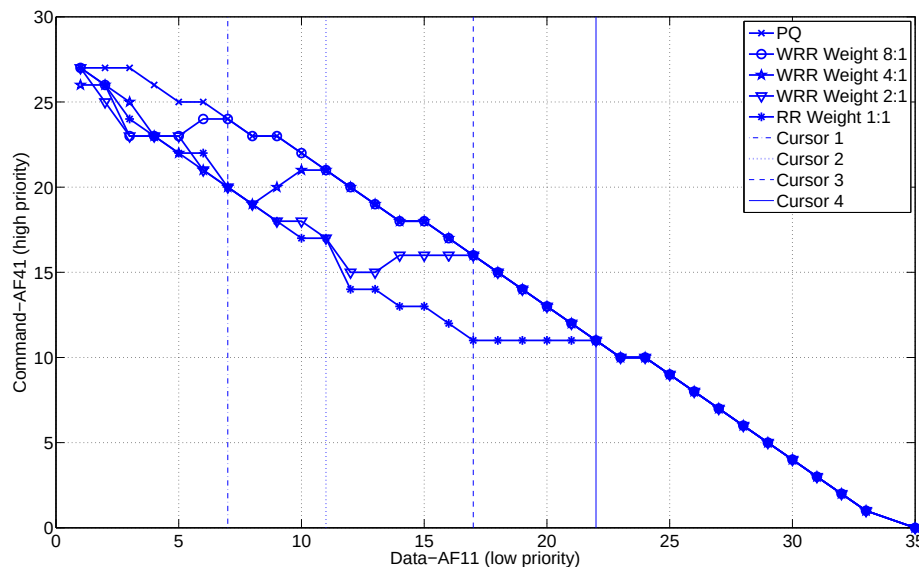


Figure 4.6 Admission Regions: Command-AF11 (Command/Control Signals) vs Data-AF11 (1480B Payload) for Different Queuing Disciplines

factor is both delay and loss (drop) for the higher priority VoIP-EF traffic flows. The buffer for the highest priority queue is $4MB$. If this queue buffer size is increased, the admission region would still be the similar shape, except the binding factor for the left section would be solely the VoIP-EF delay.

4.4 Admission Region for Data Command File and CBR Data Sessions

Figure 4.6 illustrates the admission regions for Command-AF11 vs Data-AF11. Recall that Command control data files are for the purpose of signalization (e.g., SIP, remote control, online gaming), and that data transmission files are for pure data transmission after signalizations (e.g. file downloading). In the figure, WRR curves are still in between RR and PQ curves. RR admission region is still smaller than PQ admission region because both traffic types have the same bit rate, and that the higher priority Command-AF11 demands lower delay. Thus, it favors PQ.

For PQ, although it may appear that the first 2 or 3 points at the left section forms a horizontal line, the binding is still the CBR data traffic source delay. The higher priority Command-AF11 does have some packets exceeding delay boundary, but there is just not significant amount of them. There is not many packets being dropped for both type of traffic flows. Recall that both Command-AF11 and Data-AF41 have no interest in the jitter QoS.

The lower priority CBR data traffic source delay constraint is also the binding factor for the right section of RR and WRR admission region, where RR and WRR curves joins the PQ curve. By right section, we mean the right section of the corresponding cursor. For example, cursor 1 is at the point when the 8:1 weight ratio WRR curve joins the PQ curve. In the left section of these 2 types of admission regions is dominated by the higher priority Command-AF11 delay. Although there is a visible amount of packet drops for Command-AF11 traffic flows, it is nothing compared to the amount of packets violating the delay limitation. Generally speaking, similar to idea we have explored in section 4.2.3, each WRR curve joins the RR curve when the higher priority Command-AF11 QoS is the only binding factor, and each WRR curve (including the RR curve) joins the PQ curve when the lower priority Data-AF41 QoS becomes the only binding factor. In between these 2 regions, both higher and lower priority QoS are the binding factors.

On top of the binding factor discussion, RR and PQ curves overlap each other on about 37% of the right section of figure 4.6, where this number is only 6% for VoIP-EF vs Data-AF41 from figure 4.2. Although the higher priority Command-AF11 delay still has greater influences on the admission regions of RR and small weight ratio WRR (smaller than 2 : 1), the lower priority service level agreement has greatly extended its influence on the admission region, on the contrast of other scenarios in this chapter. This is because the command/control data traffic source is very similar to CBR data traffic source. Although it is still in favor of PQ, it has slightly more demand on the fairness compared to the case of VoIP-EF vs Data-AF41.

4.5 Conclusion

In this chapter, we have shown the results regarding the admission regions for different types of traffic sources under different queuing disciplines (PQ, RR and WRR). We have concluded that when delay is the only binding factor, it is possible to have a trapeze form admission region under PQ or a linear admission region (when the higher priority QoS never becomes the binding factor). WRR curves are always in between or overlapped with the PQ and RR curves due to the change in binding factors from higher priority QoS to lower priority QoS. WRR curves overlaps with the RR curve at the left section of the admission region when the higher priority QoS is the binding factor. Both the WRR curves and the RR curve joins the PQ curve at the right section of the admission region when the lower priority QoS becomes the binding factor. Another interesting result is that PQ has greater admission region area than RR when the higher priority traffic source demands much higher QoS than lower priority QoS agreement, the RR admission region would be greater when both higher and lower priority traffic sources have similar QoS agreements. In chapter 5, we will focus more on the variation of traffic sources.

CHAPTER 5

ADMISSION REGION OF DIFFERENT TRAFFIC SOURCES UNDER PRIORITY QUEUING

This chapter will concentrate on the admission regions for different types of traffic under priority queuing. The goal is to study how the characteristics of each traffic source can affect the admission region under priority queuing, by varying one parameter at a time. These parameters could be related to the packet size, the quality of a video file, the server capacity, the QoS constraints (being tighter or looser), etc.

5.1 General Parameters and Simulation Procedure

Most of the traffic sources used in simulations from chapter 4 will be used again, with variation of one parameter at a time. For example, the highest priority audio traffic VoIP-EF is still encoded in $64Kbps$ bit rate G.711 audio codec. Although most of the time, the payload size is $160B$, sometimes in order to find the influence on admission region for varying the payload size, the audio packet size can be varied. Notice that G.711 audio codec allows a payload size range from $10B$ to $230B$, with increment of $10B$. Sometimes, $240B$ and $1480B$ are also being tested only for the purpose of seeing some trend.

5.2 CBR Data as the Lower Priority Traffic Source

In this section, the CBR Data-AF41 has been used as the lower priority traffic source. CBR data traffics are often best-effort (BE) traffics. Here, since there is still some QoS provided for the CBR data traffic, this type of traffic can no longer be considered as pure BE traffic, and it is set to be the lowest AF priority traffic AF41.

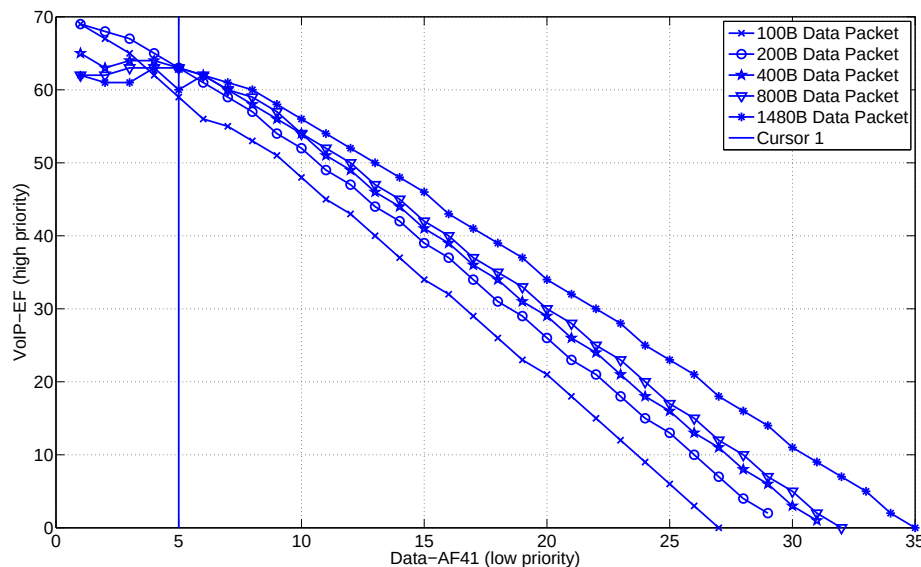


Figure 5.1 Admission Regions: VoIP-EF (G.711, 160B Payload) vs Data-AF41 (64Kbps) for Different CBR Data Packet Size

5.2.1 Admission Region for VoIP and Data Traffics

Figure 5.1 plots higher priority VoIP-EF traffic vs lower priority CBR Data-AF41 traffic admission regions for varying of data packet sizes. The Data-AF11 payload size varies from 100B to 1480B with fixed 64Kbps data bit rate, which means the packet sending rate decreases when packet payload size increases. The packet size is the payload size plus 20B header.

In general, the admission region plot shows that larger packet size would improve QoS, thus result in larger admission region area. This is because the 20B header is relatively small for larger packet. For 100B payload packet, the header represent 17% of the overall packet size, while the header of 1480B payload only represents 1.3%.

However, there is little inconsistency for the small left section of cursor 1 on the admission region plot. When the lower priority Data-AF41 payload size is $\leq 200B$, the admission region is linear. This is because the delay of lower priority CBR Data-AF41

traffic is the only binding factor for the entire admission region. Although there is some higher priority VoIP-EF packet failed QoS test when VoIP-EF sessions outnumber Data-AF41 sessions, the loss due to these QoS failures is traceable (≥ 0) but $\leq 1\%$.

When the Data-AF41 payload is $\geq 400\text{B}$, the admission region becomes trapeze. The horizontal section (at left of cursor 1 is formed due to the delay and jitter of higher priority VoIP-EF traffic being the binding factors. The loss due to delay is the same as the loss due to jitter and the loss due to delay or jitter. The loss due to lower priority Data-AF41 delay is also non-zero and $\leq 1\%$.

For the left section of cursor 1, both VoIP-EF and Data-AF41 QoS are competing with each other, and the binding factor is not always the same QoS criteria. It seems that the relative Data-AF41 packet size to the VoIP-EF packet size plays an important role. For smaller Data-AF41 payload size (around $100 - 200\text{B}$), which is comparable to the VoIP-EF payload size (160B), Data-AF41 delay becomes the binding factor. For larger Data-AF41 payload size ($\geq 400\text{B}$), VoIP-EF QoS dominates.

We can conclude that larger lower priority Data-AF41 packets have larger influence on the higher priority VoIP-EF QoS. This could be the result of using non pre-emptive mode, and it only happens when the VoIP-EF sessions outnumber the Data-AF41 sessions.

Figure 5.2 shows that for the same VoIP-EF traffic, if the Data-AF41 throughput (in terms of packet transmission bit rate) increases, the admission region would be reduced. Because of the CBR traffic nature, the maximum number of Data-AF41 is almost halved when its throughput doubles. Moreover, changing the Data-AF41 throughput does not affect the y-axis (number of VoIP-EF traffics). Therefore, the admission region size is halved as well.

If the server capacity is halved without changing the throughput, for the same VoIP-EF and Data-AF41 traffic sources, figure 5.3 shows that the maximum number of both

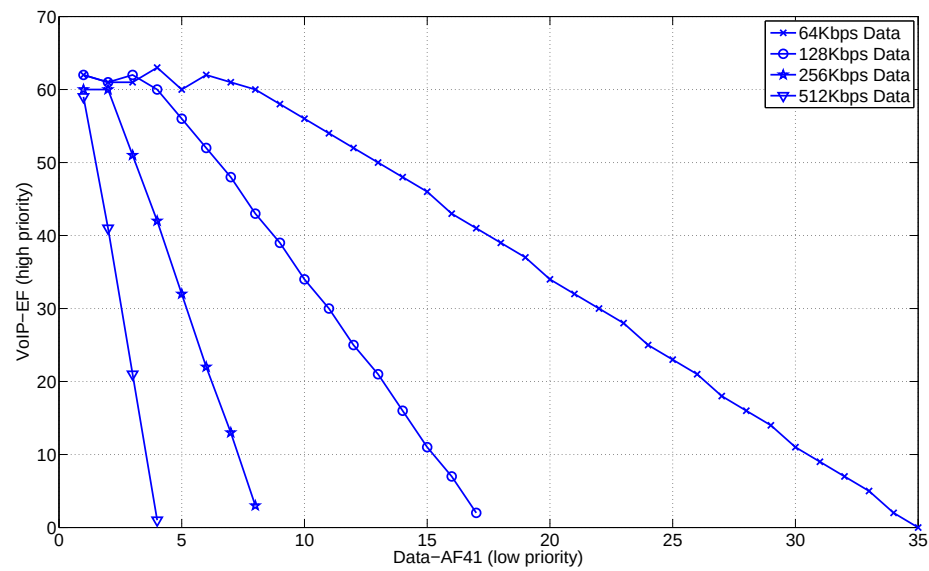


Figure 5.2 Admission Regions: VoIP-EF (G.711, 160B Payload) vs Data-AF41 (1480B Payload) for Different CBR Data Throughput

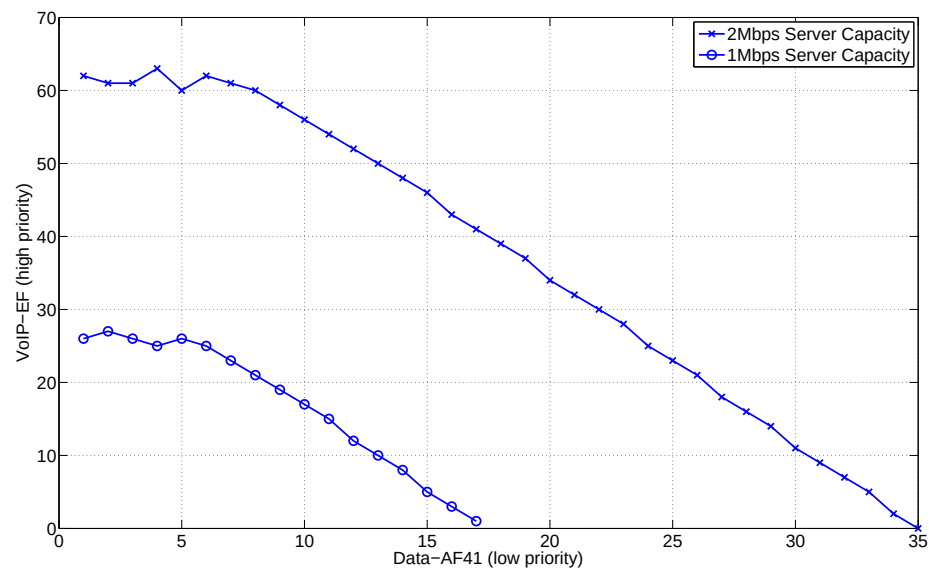


Figure 5.3 Admission Regions: VoIP-EF (G.711, 160B Payload) vs Data-AF41 (64Kbps, 1480B Payload) for Different Server Capacity

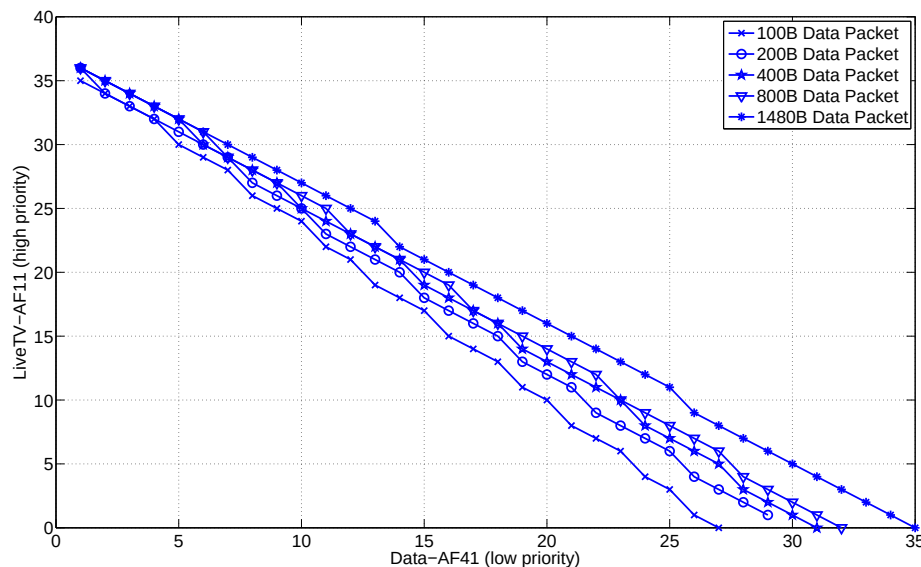


Figure 5.4 Admission Regions: LiveTV-AF11 (Star Wars IV Low Quality) vs Data-AF41 (64Kbps) for Different CBR Data Packet Size

VoIP-EF and Data-AF41 sessions would be halved. Thus, the overall admission region is reduced to $1/4$ of the original size.

5.2.2 Admission Region for LiveTV and Data Transmission

Figure 5.4 plots higher priority LiveTV-AF11 traffic vs lower priority CBR Data-AF41 traffic admission regions for various data packet sizes. Again, the Data-AF11 payload size varies from 100B to 1480B, but the throughput (in terms of packet transmission bit rate) stays the same.

Similar to VoIP-EF vs Data-AF41, larger Data-AF41 payload would expand the admission region. However, the lower priority Data-AF41 delay is the only binding factor for the entire admission region, which is already linear.

Figure 5.5 shows that the admission region size is halved for doubling both transmission line capacity and Data-AF41 throughput. Double server capacity would cause the

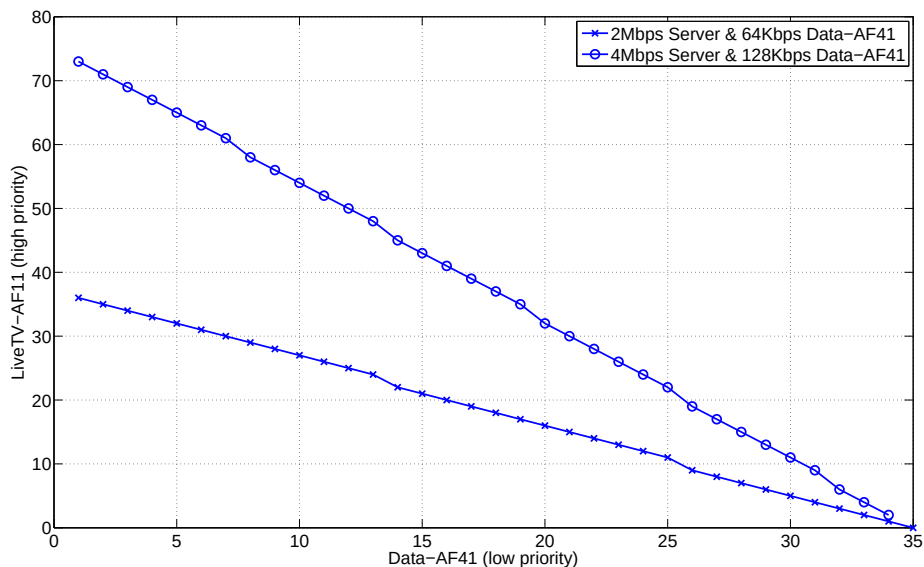


Figure 5.5 Admission Regions: LiveTV-AF11 (Star Wars IV) vs Data-AF41 (1480B) for Doubling Server Capacity and Data-AF41 Throughput)

maximum number of both LiveTV-AF11 and Data-AF41 sessions to be doubled as well. However, since the lower priority Data-AF41 throughput also doubled, the resulting new admission region would be doubled instead of quadrupled.

For the same movie Star Wars IV, figure 5.6 shows that different coding quality strongly affect the admission region. Obviously, greater the quality, smaller the admission region. However, there is a much greater reduction between high and medium quality comparing to the difference between medium and low quality. From basic area under the curve computation, low quality video admission region (AR) size is 1.46 and 5.21 times of the medium and high quality video AR sizes respectively.

Table 5.1 is extracted from table 3.2. Because the scene length is 12 and the frame sequence is IBBPBBPBBPBB, each scene has 1 frame I, 8 frame B and 3 frame P. The scene size in table 5.1 is total average size of all these 12 frames in a scene. We can see that the low quality video scene size is 1.46 and 5.16 times of the medium and high quality video scene sizes, which matches to the ratio of admission region sizes. Hence

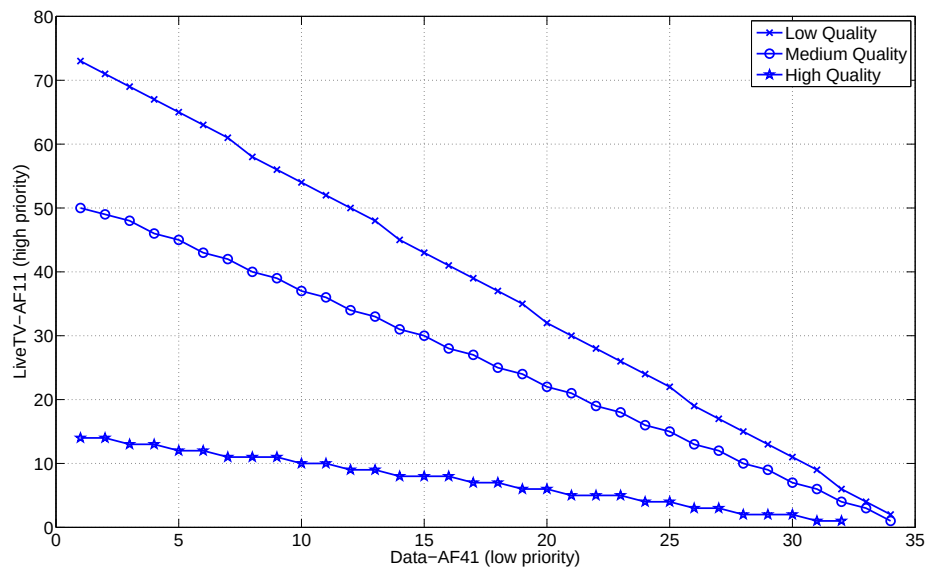


Figure 5.6 Admission Regions: LiveTV-AF11 (Star Wars IV) vs Data-AF41 (128Kbps) for Different Video Qualities, Server Capacity Becomes 4Mbps

Quality	E(I)	E(B)	E(P)	Scene Size
Low	1.5158	0.0952	0.2828	3.1258
Medium	1.5158	0.2189	0.4290	4.5540
High	3.3443	1.0301	1.5149	16.1298

Table 5.1 Size of Frames I, B and P for Star Wars IV (e.g., Size of Frame I = E(I)) and the Overall Scene Size for IBBPBBPBBPBB Sequence

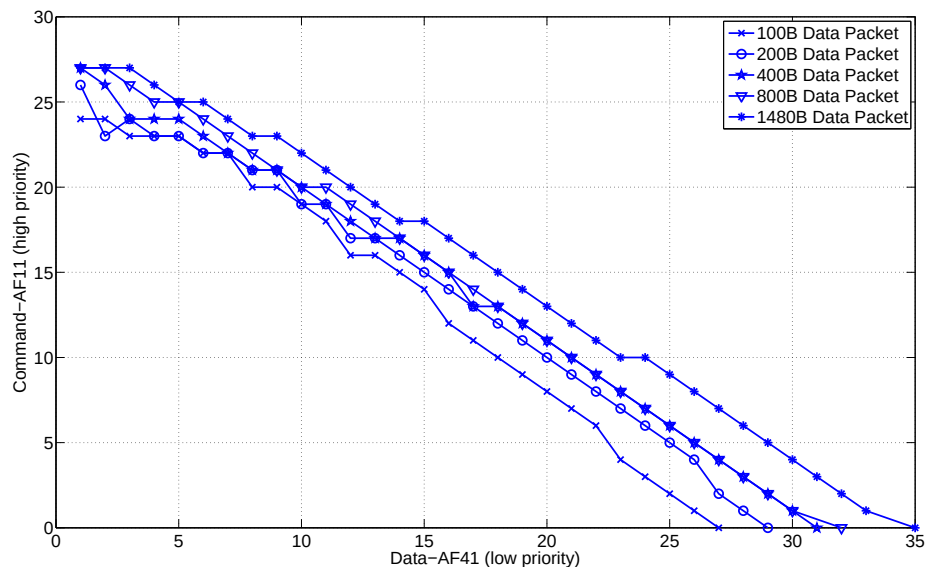


Figure 5.7 Admission Regions: Command-AF11 (Command/Control Signals) vs Data-AF41 (64Kbps) for Different Data Packet Sizes

we can conclude that the overall scene size is directly proportional to the admission region size in 1:1 ratio.

5.2.3 Admission Region for Data Command/Control and Data Transmission

Figure 5.7 plots higher priority Command-AF11 (signalization packets) vs lower priority CBR Data-AF41 admission regions for different data packet sizes. Again, the Data-AF11 payload size varies from 100B to 1480B for the same throughput.

Similarly to the previous simulations, larger Data-AF41 payload size provides better result in the admission. The binding factor is always the lower priority Data-AF41 delay. Although there is some loss due to higher priority Command-AF11 delay, the amount is just not significant.

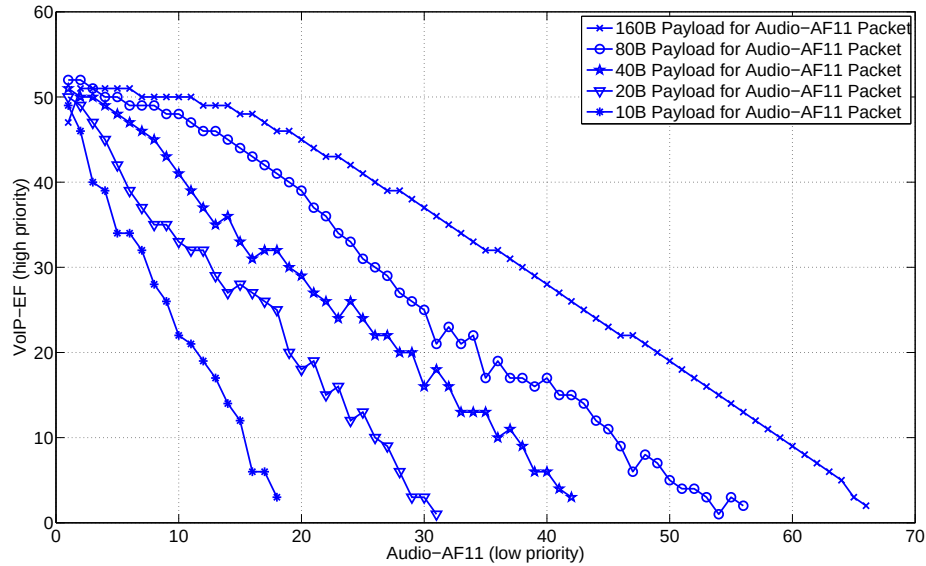


Figure 5.8 Admission Regions: VoIP-EF (G.711, 160B Payload) vs Audio-AF11 (G.711) for Different Audio-AF11 Payload Size

5.3 VoIP as Highest Priority Traffic

VoIP is always among the top priority in networking. Since VoIP-EF vs CBR Data-AF41 has already been shown in the section 5.2.1, in this section, Audio-AF11 and VOD-AF21 will be the 2 lower priority traffics in 2 different scenarios to study how VoIP-EF affect other types of lower priority traffic sources.

5.3.1 Admission Region for VoIP and Other Audio Traffics

There are audio applications other than the most popular VoIP, such as voice mail. In this section, the lower priority traffic source is Audio-AF11, which is also encoded in 64Kbps G.711 audio codec. Figures 5.8, 5.9 and 5.10 illustrate how different audio packets affect the admission regions.

Figure 5.8 shows that larger lower priority Audio-AF11 packet size would increase the

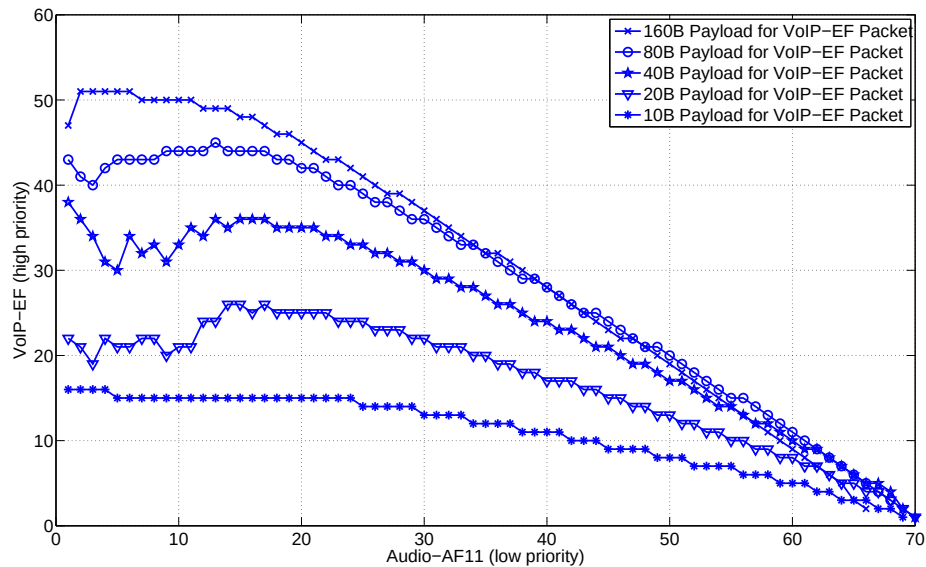


Figure 5.9 Admission Regions: VoIP-EF (G.711) vs Audio-AF11 (G.711, 160B Payload) for Different VoIP-EF Payload Size

admission region. However, double the packet size would not double nor quadruple the admission region area.

Both lower priority Audio-AF11 delay and jitter are the binding factors for the entire admission region when Audio-AF11 packet size is very small (e.g., 10B). As Audio-AF11 packet size increases, although the binding factors are still both the delay and the jitter on the right side of the admission region. However, the jitter will start to dominate over the delay in the left section of the admission region. In this left section, although the loss due to delay is visible ($\geq 0\%$), it is insignificant to the loss due to jitter.

Figures 5.9 and 5.10 illustrate the same phenomena as previous figures: larger Audio-AF11 packet size leads to larger admission region. For the right section of each admission region (the negative slope), the binding factor is both Audio-AF11 delay and jitter. The interesting part is that there are some horizontal lines or even some positive slopes in the left section of the admission region from these 2 plots. From internal data, we know that the lower priority Audio-AF11 jitter is the only binding factor in the left section of

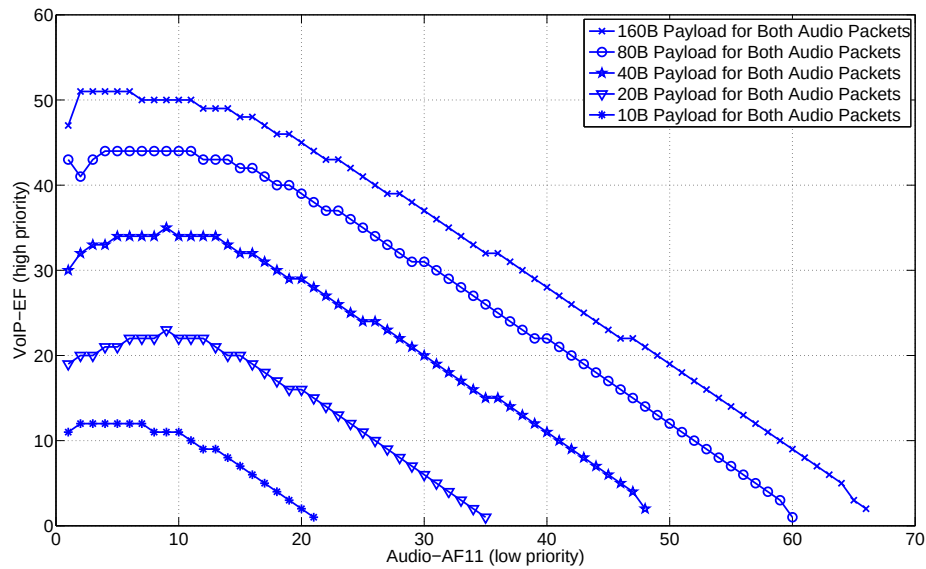


Figure 5.10 Admission Regions: VoIP-EF (G.711) vs Audio-AF11 (G.711) for Varying Both VoIP-EF and Audio-AF11 Payload Size

each admission region plotted in figures 5.9 and 5.10.

The positive slope violates a norm in admission region, where normally the more traffic flows would result worse QoS and smaller admission region. Since the jitter is the only binding factor in this left section of the admission region, it is very likely that the jitter causes the positive slope. The effect of jitter on the admission region is quite complex and will be the subject of chapter 6.

An interesting phenomena from these 3 plots is that when the lower priority Audio-AF11 packet size is relatively smaller to the higher priority VoIP-EF packet size, both Audio-AF11 delay and jitter are the binding factor for the entire admission region. When this happens, the loss due to delay, the loss due to jitter and the loss due to delay or jitter are about the same. In other words, if a packet exceeds the delay service level agreement (SLA), it would exceed the jitter SLA as well, and vice versa.

Once the VoIP-EF packet size is greater or equal to the Audio-AF11 packet size, there would be a left section from the admission region, where only jitter is the binding factor.

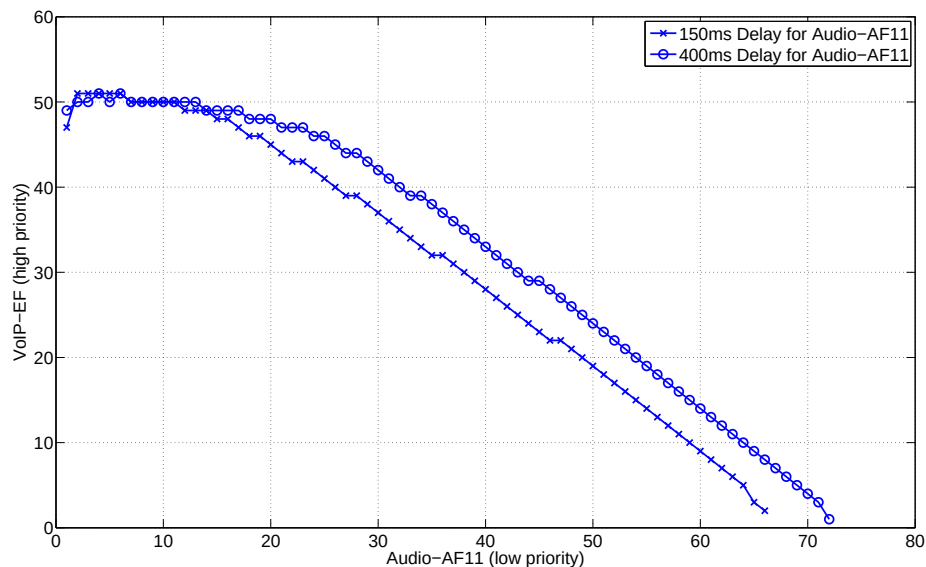


Figure 5.11 Admission Regions: VoIP-EF (G.711, 160B Payload) vs Audio-AF11 (G.711, 160B Payload) for Different Audio-AF11 Delay Service Level Agreement

The binding factors for the right section are still both the delay and the jitter.

In figure 5.11, both audio traffics have the same properties and service level agreements for most QoS criteria, except the lower priority Audio-AF11 delay varies between 150ms and 400ms. The results show that changing delay boundary has no effect in the left section of the admission region, and that looser delay boundary would increase the admission region in the right section. This is because the jitter is the only binding factor in the left section of the admission region, and naturally varying the delay service level agreement would not affect the jitter QoS. In general, looser delay boundary would result larger admission region area.

5.3.2 Admission Region for VoIP and LiveTV

In figure 5.12, both VoIP-EF and LiveTV-AF11 traffic sources stay the same, and most the QoS constraints as well, except the LiveTV-AF11 delay varies between 150ms and 10s. The resulting admission region size increases while the delay boundary getting

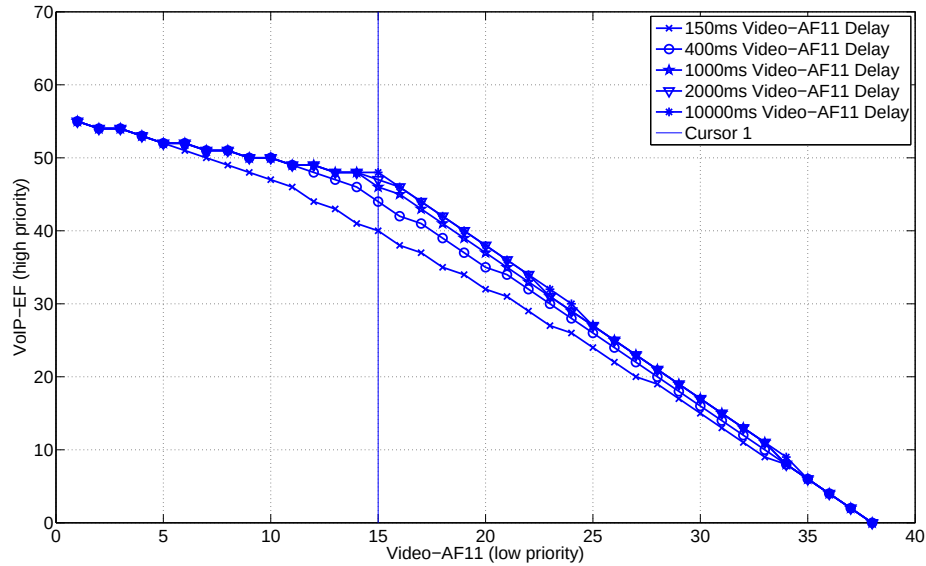


Figure 5.12 Admission Regions: VoIP-EF (G.711, 160B Payload) vs LiveTV-AF11 (StarWars IV Low Quality) for Different LiveTV-AF11 Delay Service Level Agreement

looser.

In general, the binding factor for the left section of the admission region is the LiveTV-AF11 jitter. On the other side, the LiveTV-AF11 delay becomes the dominant factor in the right section. In between, there is a visible pivot point, where the domination changes hands.

For very tight LiveTV-AF11 delay constraint ($150ms$), although the LiveTV-AF11 jitter is still the binding factor for the left section, there are a significant amount of LiveTV-AF11 packets exceeding the delay constraints as well. There is a factor (2 : 1) in this section of the admission region, which means for every 2 packets violating the jitter limitation, there is 1 packet violating the delay limitation. That is why for the $150ms$ curve, the pivot point is not as obvious as in the other curves.

In order to learn more about which QoS criteria has greater effect on the admission region, the QoS criteria are treated separately in figure 5.13. First, it is not easy to isolate the loss (drop) due to buffer being full from the rest QoS criteria because if considers

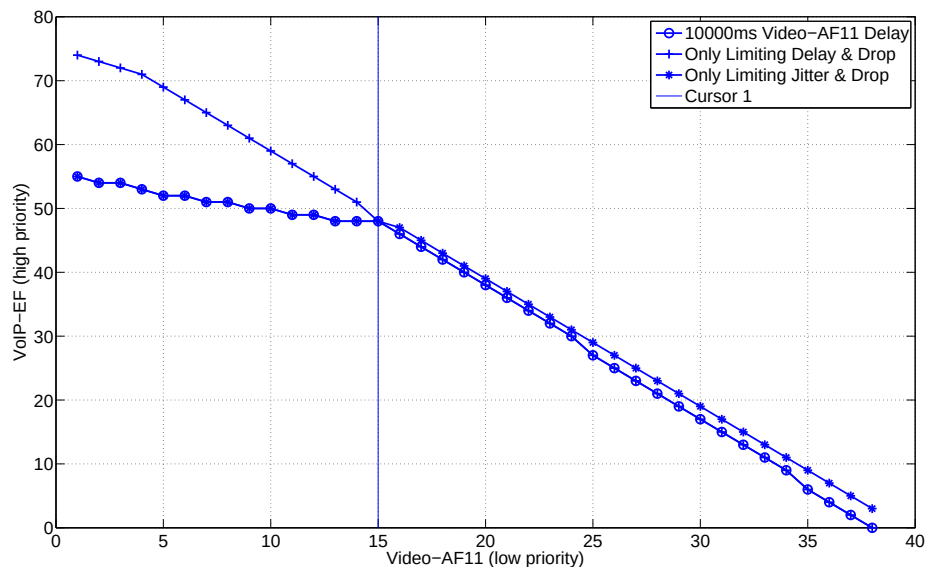


Figure 5.13 Admission Regions: VoIP-EF (G.711, 160B Payload) vs LiveTV-AF11 (StarWars IV Low Quality) for Different LiveTV-AF11 Delay Service Level Agreement

infinite buffer size for the queues, it would often cause a buffer leakage problem. Buffer leakage means buffer overflow (i.e. simulation require too much buffer during the process). Therefore, the plots of limiting delay only (or jitter only) would also include the loss due to dropping, although normally the loss due to drop is always 0 in almost all the simulations in this thesis.

The admission region due to delay limitation is not exactly the trapeze shape proposed by Arthur W. Berger & Ward Whitt because the delay is not the only binding factor. Recall that for delay being the only QoS criteria in an ATM network under priority queuing, the admission region forms a trapeze [28].

Although the loss (drop) has not been separated from other QoS, since there is no loss, it will not affect the admission region at all. The jitter dominance in the left section may be caused by the nature of the video traffic source (sudden change for the size of bursts due to relative large frame I compared to the rest of frames).

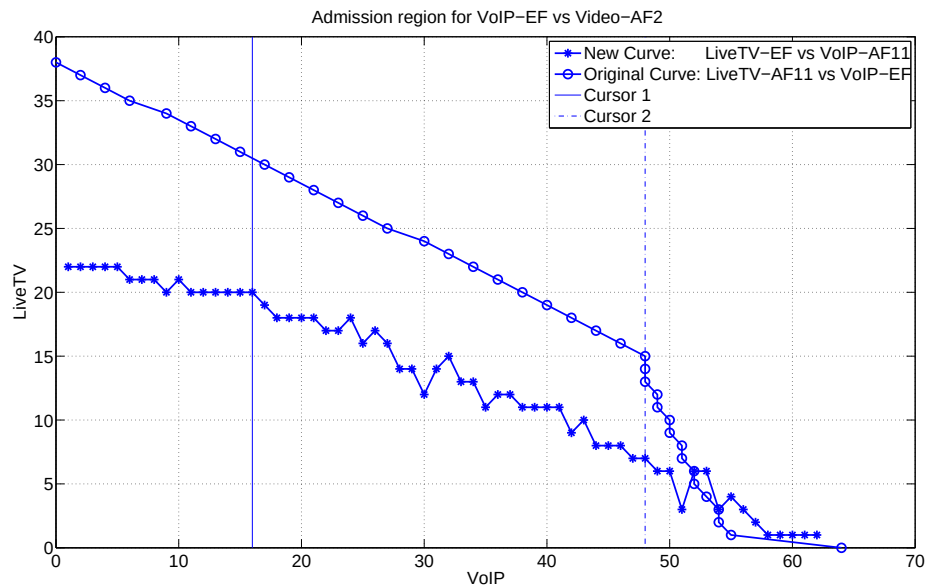


Figure 5.14 Admission Regions: LiveTV-EF (StarWars IV Low Quality) vs Audio-AF11 (G.711, 160B Payload)

5.3.2.1 Admission Region for VoIP and LiveTV (Reverse Priority)

Currently, the common usage is to put voice on the EF class and real-time video in the AF classes (usually the highest) [21]. However, both types of traffic have some strong QoS requirements (typically the Delay for VoIP and the jitter for both). This section shows results for the case where voice is in the AF class and LiveTV in the EF class and compares to the previous results where VoIP has higher priority than LiveTV shown in section 5.3.2.

In figure 5.14, the video traffic source (Live-TV) is StarWars IV video, and the audio traffic source (VoIP) is coding in the standard G.711 audio codec with 160B payload. The original curve was extracted from figure 5.12 and transposed (x-axis $\leftrightarrow$ y-axis), where LiveTV was the lower priority LiveTV-AF11, and VoIP was the higher priority VoIP-EF. The new curve is simulated by switching VoIP to AF class and LiveTV to EF class.

For the original curve (LiveTV-AF11 vs VoIP-EF), the curve 2 is the same as the curve 1

in figure 5.12 but has been transposed. For the new admission region curve (LiveTV-EF vs VoIP-AF11), the curve 1 in figure 5.14 separates the admission region into 2 sections. Based on the intermediate data, in the left section, the binding factor is the lower priority VoIP-AF11 jitter. In the right section, the binding factor becomes both the delay and the jitter of VoIP-AF11.

In general, the admission region is trapeze like, but the left section is not perfectly horizontal. VoIP-EF vs LiveTV-AF11 produces better results than LiveTV-EF vs VoIP-AF11. The VoIP-EF vs LiveTV-AF11 case almost has double the admission region size compared to the LiveTV-EF vs VoIP-AF11 case. This explains why most companies choose VoIP-EF vs LiveTV-AF11 over LiveTV-EF vs VoIP-AF11. As for why, a possible explanation could be that the audio traffic source requires better QoS than the video traffic source (e.g. the delay). Another possible explanation is that the video traffic source requires much more bandwidth than an audio traffic source, and that if the video traffic source becomes the highest priority, it could cause the audio traffic source QoS to downgrade very fast.

5.4 Conclusion

In this chapter, we have shown how each traffic source parameter affects the admission region under priority queuing. In general, the throughput (packet bit rate) is inversely proportional to the admission region size. Therefore any parameter that is directly proportional to the throughput is also inversely proportional to the admission region size. These parameters could be the packet size, the packet arrival rate, the average frame size (or the scene size) for video, the server capacity, etc. Notice that this only applies to the case(s) when only varying one parameter. An example of varying 2 parameters in the same time, sending the same file (audio, video or data) with larger packet payload size would reduce the number of packets (each packet has a header). Thus, the reduced amount of headers would increase the admission region size. For the QoS service level

agreement parameters (such as delay), the looser QoS constraints would lead to larger admission region (not directly proportional). Another interesting discovery is that when the lower priority packet size is much larger than the higher priority packet size, the lower priority packets may influence the higher priority QoS due to non pre-emptive mode if the higher priority sessions outnumber the lower priority sessions. The VoIP-EF vs LiveTV-AF11 admission region has almost double the size compared to the case of LiveTV-EF vs VoIP-AF11. At the end, there is one important result showing that jitter affects the admission region differently than the rest of QoS measures. It is possible that more traffic flows would improve the quality of jitter and cause positive slope in the admission region. Chapter 6 will explain this in details.

CHAPTER 6

INFLUENCE OF JITTER ON THE ADMISSION REGION

Recall the positive slope admission region curve problem from section 5.3.1, which is against the normal admission region behavior that increasing the number of traffic flows would normally result in worse QoS (section 5.3.1 has a reversed result). In this chapter, we will first show what the positive slope is. Then, we will show that the jitter is the cause. At the end, we will discuss how the jitter affects the admission region differently than the rest of QoS measures and show why more traffic flows may improve the quality of jitter.

6.1 Positive Slope in Admission Region

With or without DiffServ, normally the more traffic flows presence in a network would result worse QoS. Thus, under normal circumstances, there should not exist any positive slope on an admission region plot. However, some results show otherwise. Recall the example from figure 5.11, VoIP-EF (high priority) vs Audio-AF41 (low priority), there is a positive slope for the left section of each admission region curve, the jitter is the only binding factor.

It is clear that when there are very few lower priority Audio-AF41 sessions presence in the network (left section of the admission region), there are obviously some positive slopes, especially for 20B and 40B G.711 audio packets, which illustrates the irony that more traffic flows results in better QoS and larger admission region.

Recently, a researcher has done some research on the jitter behaviors for M/M/1 and M/G/1 single queue system. Based on his research, the jitter value should be decreas-

ing when the packet arrival rate increases if the network is not heavily loaded. This is because when the network is less loaded, the main factor to cause the variation of delay (jitter) is the fluctuation of the packet inter-arrival time [43, 44].

In the example from figure 5.11, both traffic sources have fixed packet sizes, the service time would also be fixed because the server capacity is a fixed number and that $(\text{Server Capacity})/(\text{Packet Size}) = (\text{Service Time})$. Thus in this case, only the waiting time in the queue (W) can affect the variation of the delays (jitter).

6.2 Causes of the Positive Slope in the Admission Region

In order to understand what are the factors triggering the positive slope to appear in the simulations under DiffServ network, each QoS criterion is viewed separately for the left section of the admission region.

From the example shown in figure 5.10 (20B payload curve), based on some intermediate data, there is no loss due to any higher priority VoIP-EF QoS constraints (delay, jitter and drop) within the admission region. The loss due to drop for Audio-AF41 is also zero. The only two QoS criteria affecting the admission region is the delay and the jitter from Audio-AF41. Since the positive slope in figure 5.10 only occurs for ≤ 9 lower priority Audio-AF41 sessions, we only plot the loss due to delay and jitter separately for fixed numbers higher priority VoIP-EF sessions and varying the number of Audio-AF41 sessions between 1 and 15, see figures 6.1 and 6.2.

Figure 6.1 shows that the Audio-AF41 delay has strong effect on the admission region when there are ≥ 12 Audio-AF41 (low priority) sessions. It also shows that more sessions causes greater delay (worse QoS).

On the other hand, figure 6.2 shows that the jitter decreases as the number of low priority Audio-AF41 sessions increases when there are very few number of Audio-AF41

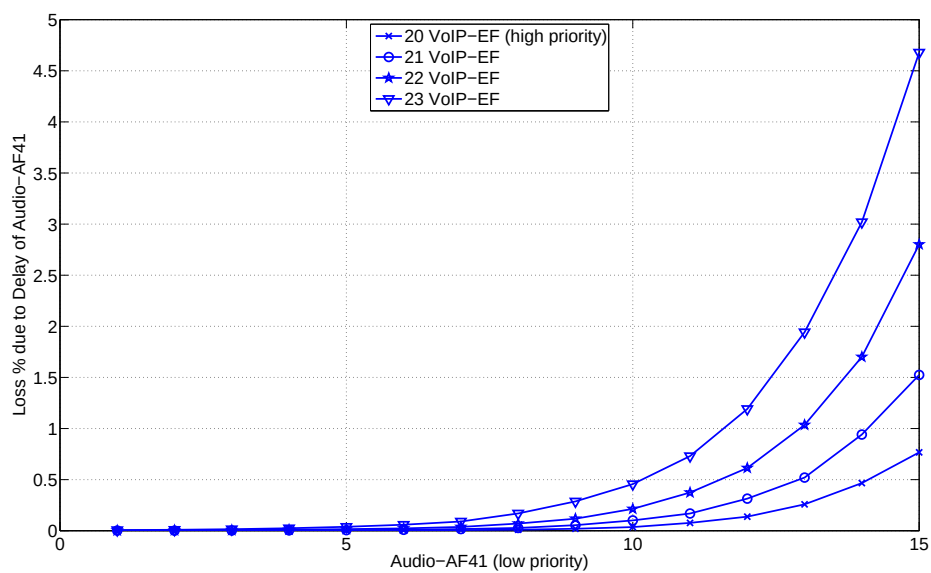


Figure 6.1 Loss due to Delay of Audio-AF41 (low priority)

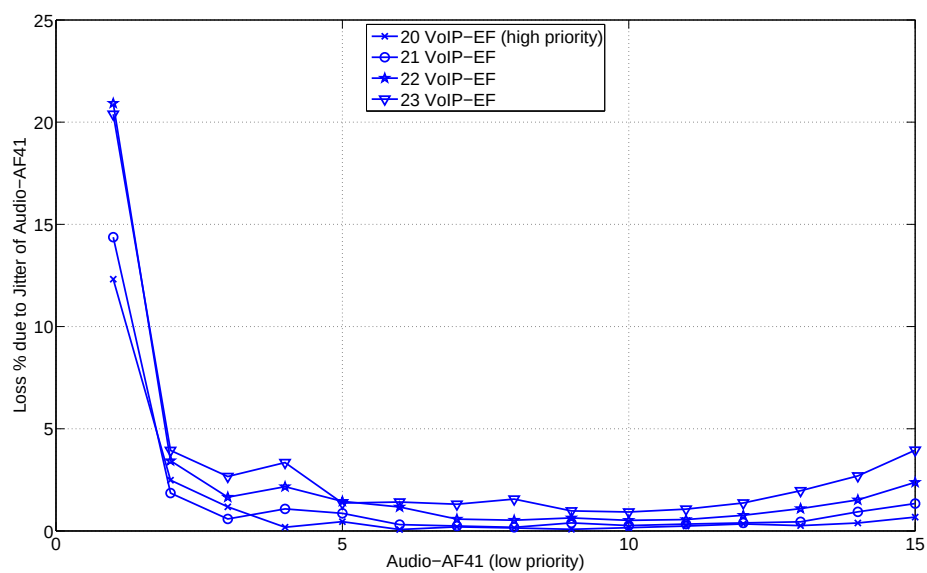


Figure 6.2 Loss due to Delay of Audio-AF41 (low priority)

sessions. This is due to the similarity between queue buffer and jitter buffer. The jitter would eventually increase again, when there are too many Audio-AF41 sessions. The reason might be that the queue buffer has a limited capability to convert jitter to delay.

The queue buffer is not exactly the same as a jitter buffer. A normal jitter buffer tries to output packets at a constant rate by introducing extra delay. If the delay of a traffic flow is far from the service level agreement (SLA), and that the jitter exceeds the SLA, the jitter buffer can simply do trade-off between delay and jitter and increase the maximum number of sessions allowed.

On the other hand, queue buffer simply send packets to the server according to a queuing discipline. At first when there are very few lower priority Audio-AF41 sessions, the queue buffer does not emulate well jitter buffer due to highly random packet arrival rate. Notice that the jitter is the variation of delay. The variation of delay can be caused by the fluctuation of the packet inter-arrival time and the variation of waiting time in the queue.

The variation of waiting time can be caused by the number of packets needed to be served first and the packet service time. In our case, since the packet size is constant, the packet service time would also be constant. For fixed number of higher priority VoIP-EF sessions in figure 6.2, the VoIP-EF jitter stays the same because lower priority Audio-AF41 packets have very little influence on the VoIP-EF QoS. Since there are very few lower priority Audio-AF41 sessions in the left sections of figure 5.10, often there is no packet in the Audio-AF41 queue when a Audio-AF41 packet just arrived to the queue. Thus, the variation of the waiting time does not have significant influence on the overall jitter. Therefore, we can conclude that only the fluctuation of the inter-arrival time of each lower priority Audio-AF41 packet would influence the jitter in the left sections of figure 5.10, where the positive slope occurs.

When the number of lower priority Audio-AF41 sessions increases, the packet inter-arrival time becomes more constant. Thus the jitter decreases for an increase of Audio-AF41 sessions. However, when there are many lower priority Audio-AF41 sessions in

the network, although there is less fluctuation for the inter-arrival time, the variation of the waiting time in the queue begin to cause jitter to slowly increase again.

6.3 Conclusion

In this chapter, we have shown that jitter behaves differently than other QoS (e.g., delay), where the more traffic flows should result worse QoS. For jitter, increasing the number of traffic flows may result better jitter quality if there are only very few traffic flows in the network. This is because the variation of the packet inter-arrival time is the main factor to cause jitter to change. In chapter 7, we will show how to approximate different admission regions that we have seen so far in this thesis.

CHAPTER 7

LINEAR APPROXIMATION FOR ADMISSION REGION

Recall the admission regions are used for doing connection admission control (CAC), and that a linear approximation could make CAC much simpler. In this chapter, we will show that most admission regions can be approximated by a linear equation. Some admission regions could be better approximated by a trapeze form (2 linear equations) or a twisted trapeze form (3 linear equations).

7.1 Linear Approximation for Linear Curves

If the admission region is already in linear form, it would be very easy to compute the linear equation with 2 dots from the figure: $(n_1 = 1, n_2)$ and $(n_1, n_2 = 0)$, where n_1 and n_2 are the number of higher and lower priority traffic flows respectively. Figure 7.1 is extracted from figure 5.4, where the Data-AF41 packet payload size is 1480B. The linear approximation is real close to the simulated admission region.

Linear form admission region happens when one of the lower priority QoS is the only binding factor for priority queuing. It is possible to be a result of loose higher priority QoS constraints. In this case, even if the lower priority traffic flows outnumber the higher priority traffic flows, the higher priority traffic flows would still suffer minimum loss (due to delay, jitter and drop). Round robin would also cause the admission region to be linear form. In general, linear shape admission region often means that there is only one binding factor. If there are more than one binding factor, the admission region would often be non-linear form.

When the admission region is not linear but convex, the approximated linear equation

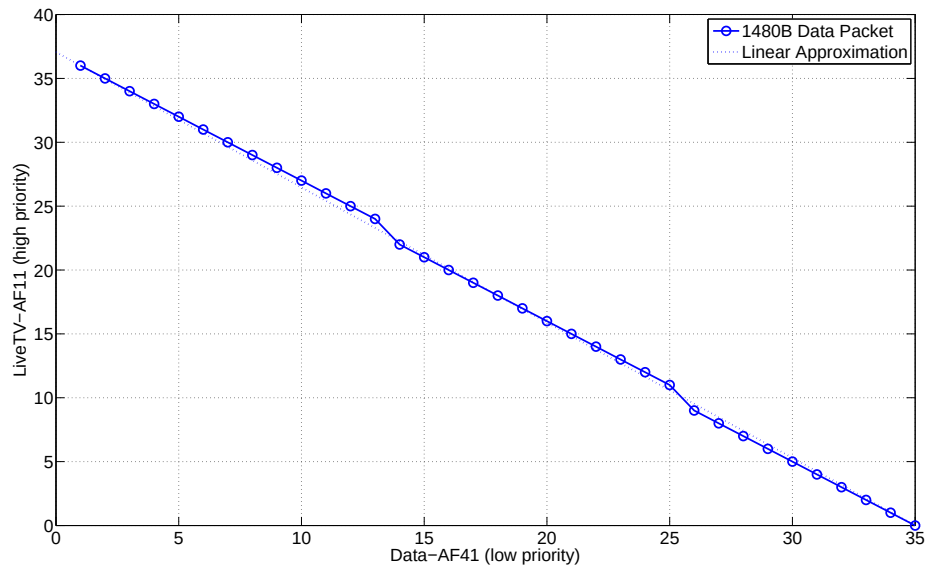


Figure 7.1 Linear Approximation for the Admission Region: LiveTV-AF11 (StarWars IV MPEG-4 low quality) vs Data-AF41 (64Kbps, 1480B Payload) under PQ

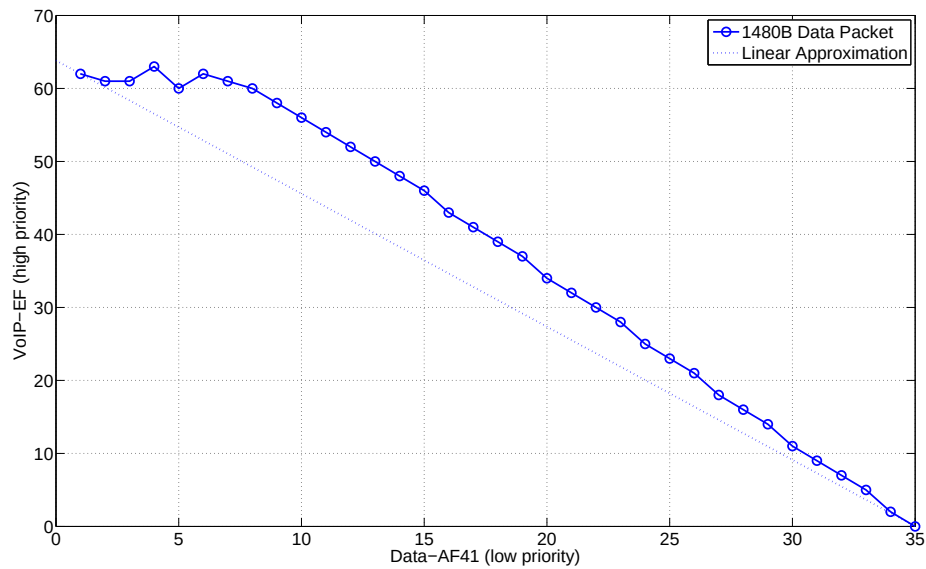


Figure 7.2 Linear Approximation for the Admission Region: VoIP-EF (G.711, 160B Payload) vs Data-AF4 (64Kbps, 1480B Payload) under PQ

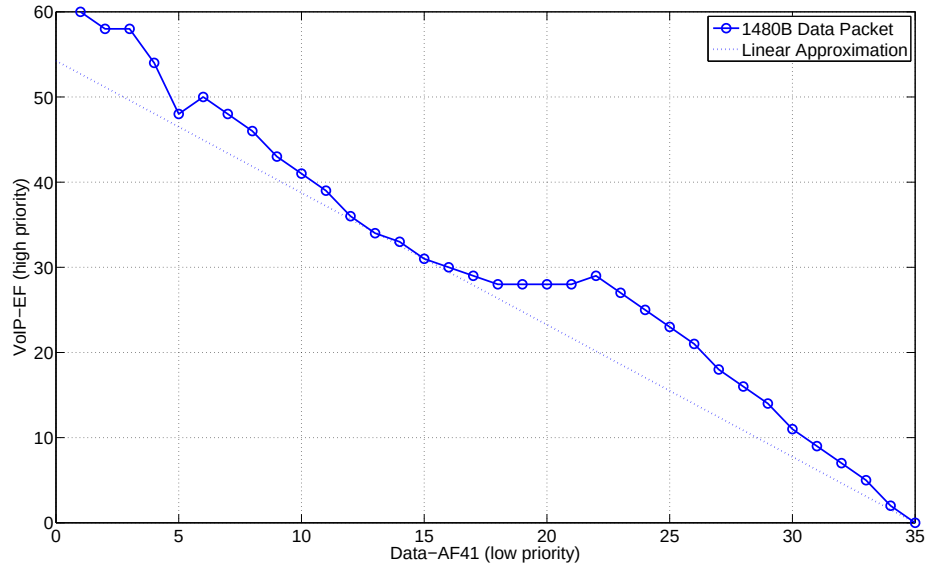


Figure 7.3 Linear Approximation for the Admission Region: VoIP-EF (G.711, 160B Payload) vs Data-AF4 (64Kbps, 1480B Payload) under WRR (8:1 Weight Ratio)

can still be computed in the same way. Figure 7.2 is extracted from figure 5.1, where the Data-AF41 packet payload size is 1480B. The linear approximation has sacrificed a little admission region in order to make faster admission decisions.

Many admission regions for IP network under DiffServ shown in this thesis are concave. This means the linear approximation is conservative. It would produce better value of QoS than what is required.

When the delay is the only binding factor and that higher priority delay also exceeds the service level agreement when there are many higher priority traffic flows in the network, the admission region is often in form of a trapeze.

Another possible shape is twisted trapeze shown in figure 7.3, extracted from figure 4.2. The linear approximation shown in the figure 7.3 is conservative. In other words, it would provide better QoS than the service level agreement. The admission region is in form of a twisted trapeze only when the queuing discipline is weighted round robin.

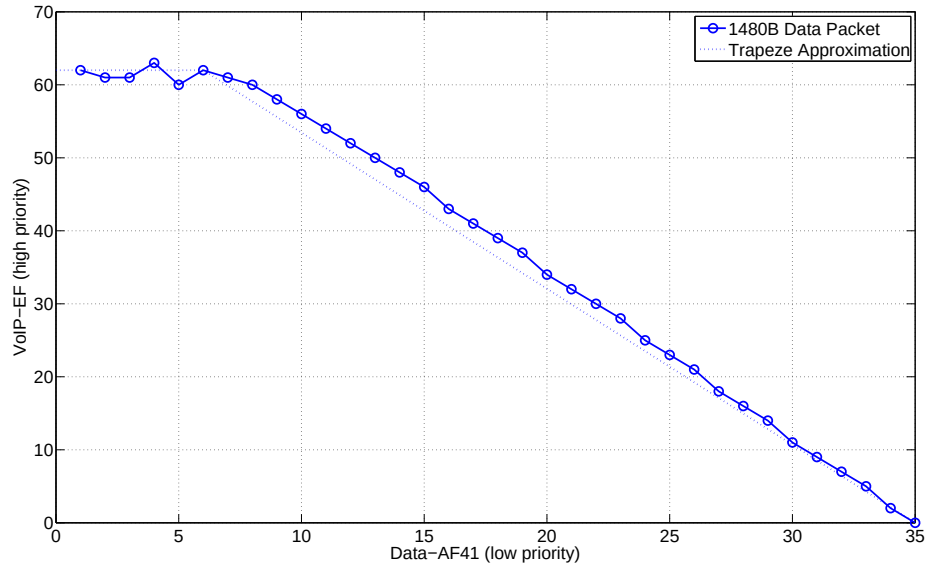


Figure 7.4 Trapezoidal Approximation for the Admission Region: VoIP-EF (G.711, 160B Payload) vs Data-AF4 (64Kbps, 1480B Payload) under PQ

7.2 Other Approximations

It is possible to approximate some trapezoidal admission regions, where the lower priority traffic delay is the binding factor for the entire admission region. Same as figure 7.2, figure 7.4 is also extracted from figure 5.1. The trapezoidal approximation would allow more traffic flows than linear approximation.

For twisted trapezoidal admission regions, another approximation could be composed by 3 linear equations: a negative slope straight line, an horizontal line and another decreasing straight line. The example shown in figure 7.5 illustrates well this approximation. We call it linear-horizon-linear approximation, and it allows more traffic flows than conservative linear approximation.

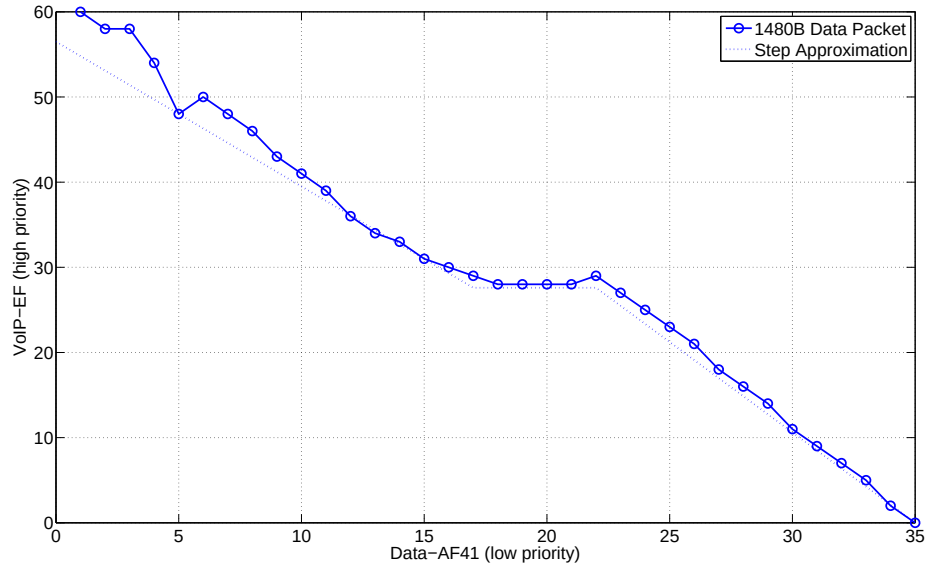


Figure 7.5 Step Approximation for the Admission Region: VoIP-EF (G.711, 160B Payload) vs Data-AF4 (64Kbps, 1480B Payload) under WRR (8:1 Weight Ratio)

7.3 Conclusion

In this chapter, we have shown that it is possible to use linear approximation admission region to make CAC simpler. However, the linear approximation are often conservative because many admission regions are concave. In the cases when the non-linear shape admission region is caused by changing binding factor(s), it is possible to use several linear equations to better approximate the admission region (often 2 or 3 linear equations would be enough). Every time there is a change in binding factor(s), an additional linear equation may be added. In chapter 8, we will show that the idea of admission region in DiffServ network can be extended to the case of more than 2 types of traffic sources.

CHAPTER 8

ADMISSION REGION FOR 3 TRAFFIC TYPES

In the previous simulation scenarios, there are always 2 types of traffics to create an admission region area. In this chapter, 3 types of traffics are introduced into the network at the same time, thus resulting an admission region space in 3-D.

8.1 Admission Region for 3 Traffic Types under Priority Queuing (PQ)

Figure 8.1 illustrate a 3-D admission region for 3 traffic types under priority queuing. The 3 traffics are: Data-AF41 (lowest priority) on the x-axis, LiveTV-AF11 (medium priority) on the y-axis and VoIP-EF (highest priority) on the z-axis. In general, the admission region looks like an inclined plane with a small plateau, and there are some sudden increase at the edge of the plateau.

On the small plateau, intermediate data show that the binding factor(s) are always highest priority VoIP-EF QoS. It is similar to the horizontal line in the left section of a 2-D admission region (e.g., figure 5.3). The sudden increase at the edge of the plateau is due to the zero value that has been discussed in section 4.1.4.

Figure 8.2 is a projection of figure 8.1 on the XZ-plane. For each number of medium priority LiveTV-AF11 sessions range from 0 to 35 with an increment of 5, we plot the admission region curve (VoIP-EF vs Data-AF41). Figure 8.2 illustrates that greater the number of LiveTV-AF11 sessions, the smaller the admission region projection on the plane of VoIP-EF vs Data-AF41.

Recall that for VoIP-EF vs Data-AF41, there is a left section where both the VoIP-EF delay and jitter are the binding factors. The Data-AF41 delay would gradually dominate

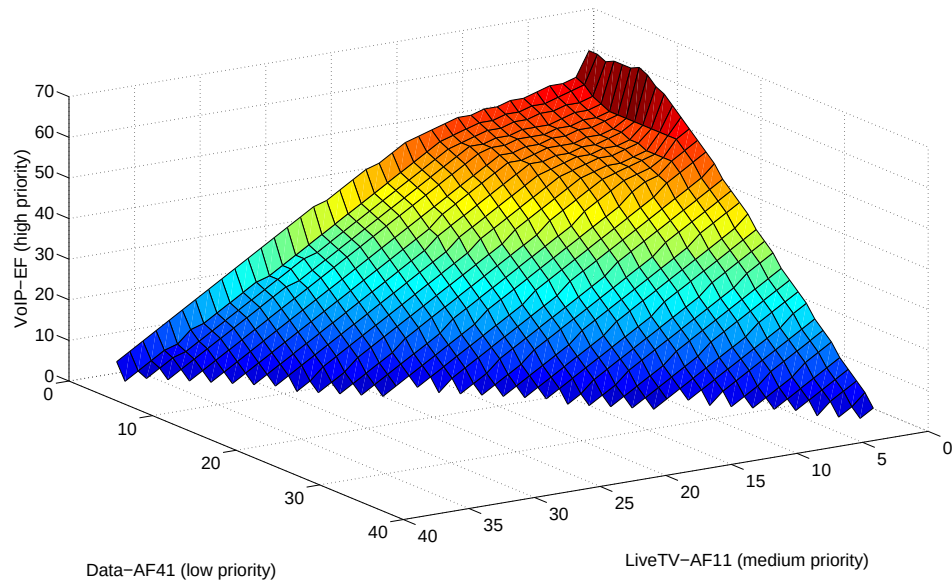


Figure 8.1 Admission Regions: VoIP-EF (G.711, 160B Payload) vs Data-AF41 (1480B Payload) vs LiveTV-AF11 (StarWars IV Low Quality) under Priority Queuing

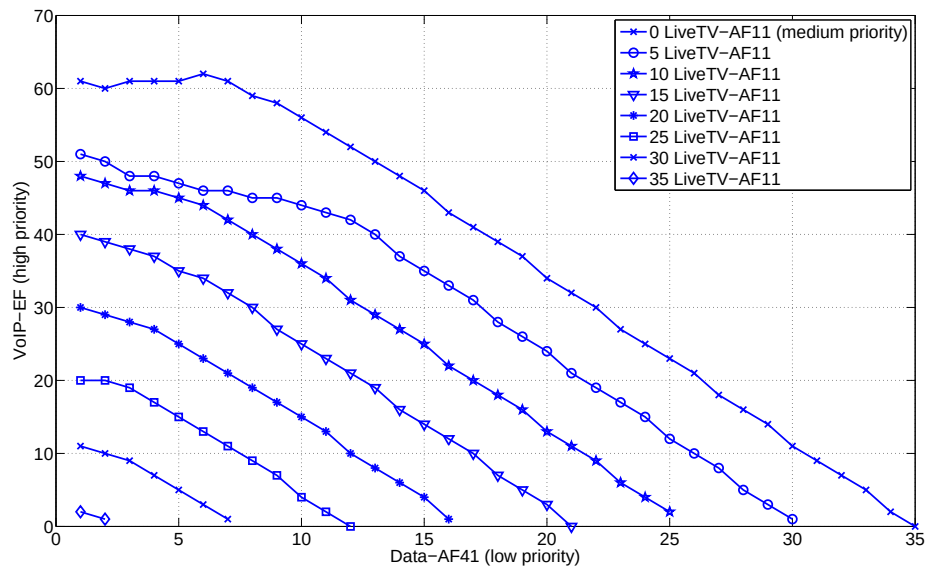


Figure 8.2 Admission Regions: VoIP-EF (G.711, 160B Payload) vs Data-AF41 (1480B Payload) for Different Number of LiveTV-AF11 Traffics under Priority Queuing

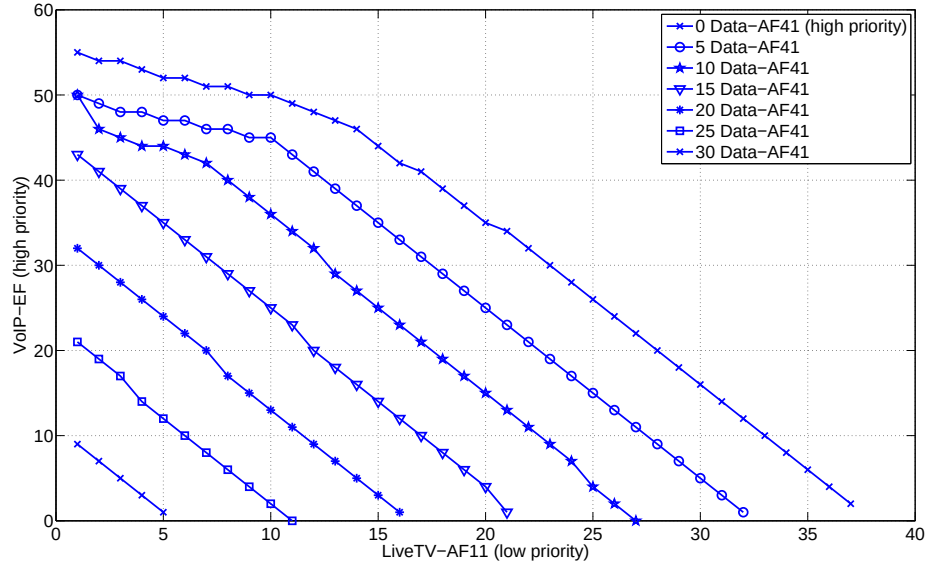


Figure 8.3 Admission Regions: VoIP-EF (G.711, 160B Payload) vs LiveTV-AF11 (Star Wars IV Low Quality) for Different Number of Data-AF41 Traffics under Priority Queuing

over VoIP-EF QoS as the number of Data-AF41 sessions increases.

Here, as the number of LiveTV-AF11 sessions increases, for the left section mentioned above, the higher priority VoIP-EF delay and jitter would have less effect on the admission. If the number of LiveTV-AF11 sessions is greater than a certain value, the higher priority VoIP-EF delay and jitter would have no effect on the admission region projection. However the binding factor in the right section of the admission region will not be the Data-AF41 delay but rather be the LiveTV-AF11 delay and jitter.

Figure 8.3 is a projection of figure 8.1 on the YZ-plane. For each number of low priority Data-AF41 sessions range from 0 to 30 with an increment of 5, we plot the admission region curve (VoIP-EF vs LiveTV-AF11). Figure 8.3 illustrates that greater the number of Data-AF41 sessions, the smaller the admission region projection on the plane of VoIP-EF vs LiveTV-AF11.

Recall that for for VoIP-EF vs LiveTV-AF11, the binding factor in the left section of the

admission region is the LiveTV-AF11 jitter, and the binding factor in the right section is the LiveTV-11 delay.

Here, similar to the VoIP-EF vs Data-AF41 projection, as the number of Data-AF41 sessions increases, the admission region becomes smaller and linear. This is because VoIP-EF QoS would not become the binding factor if there are more Data-AF41 sessions, even if only few Live-TV sessions present in the network.

For the projection plots in general, whenever there are very few 3^{rd} type (not on the projection plane) of traffic has very few sessions, the projection has the similar shape to the admission regions shown in chapters 4 and 5. Whenever there is one type of traffic missing (zero value), there may be a sudden increase for other type(s) of traffic in terms of the maximum number of sessions. Whenever none of any type of traffic has too few sessions, the admission region is a flat plane (linear for the projections).

8.2 Admission Region for 3 Traffic Types under Weighted Round Robin (WRR)

Figure 8.4 is the same scenario as figure 8.2, except it is under WRR instead of priority queuing (PQ). In figure 8.4, the weight ratio for the WRR queuing discipline is 4 : 2 : 1, in which the highest priority VoIP-EF, medium priority LiveTV-AF11 and lowest priority Data-AF41 have a weight of 4, 2 and 1 respectively. Again, the admission region projection size decreases as the number of the 3^{rd} traffic type (LiveTV-AF11) sessions increases.

Recall that for WRR admission regions, there is always a left section where the higher priority QoS becomes the binding factor, regardless which type of traffics. The only difference would be the size of this left section. Here, similar to the case of PQ, as the number of LiveTV-AF11 sessions increases, the highest priority VoIP-EF QoS would no longer be the binding factor at the left section of the admission region projection, although there are very few number of Data-AF41 traffics in that area. When there are

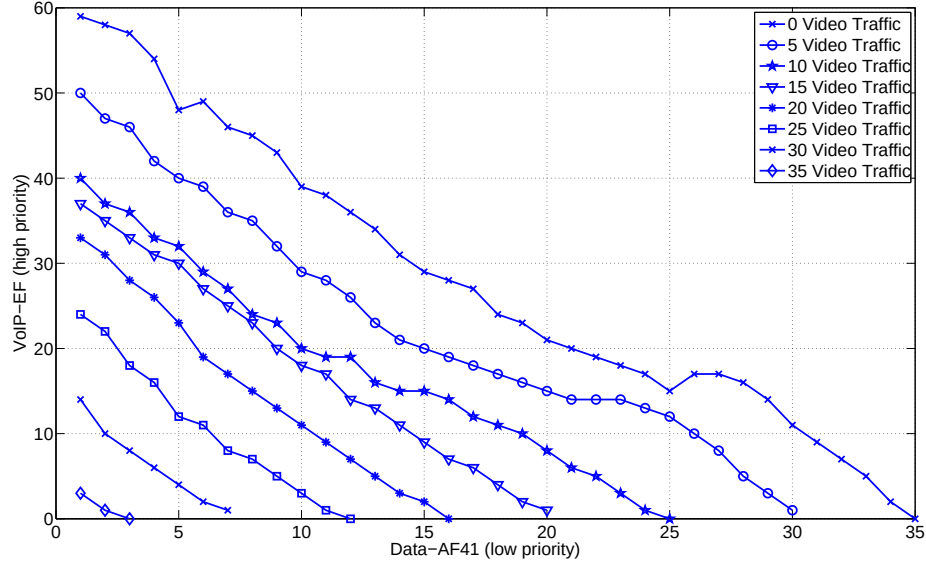


Figure 8.4 Admission Regions: VoIP-EF (G.711, 160B Payload) vs Data-AF41 (1480B Payload) for Different Number of LiveTV-AF11 Traffics under WRR (weight ratio 4 : 2 : 1)

very few LiveTV-AF11 sessions, the projected admission region would be just like the admission regions shown in chapters 4 and 5.

Due to limited time, we did not run all $70 \times 40 \times 40 = 112000$ simulations for the WRR. It would be hard to plot the 3-D graph and the projections on other planes. However, in general, we would expect an inclined plane (not perfectly flat).

8.3 Conclusion

In this chapter, we have shown that the idea of connection admission control (CAC) is not limited to 2 types of traffic. It could be 3, 4 or greater. However, the admission region space can no longer be approximated by a straight line. Instead, a plain surface might do the job.

The overall result shows that when one type of traffic outnumbered all other types of

traffic under priority queuing, its QoS would may become the binding factor. The 3-D admission region plot is more or less an inclined plane. There could be a small plateau if the highest priority QoS can sometimes be the binding factor.

Whenever a 3^{rd} type of traffic has very few sessions, the projection on the first 2 traffic type plane would have similar shape to its original admission region (where there is no other type of traffic). The difference between the curves when there is no 3^{rd} type of traffic and the time when there is 1 3^{rd} type of traffic can be huge. Although the shape of the admission region projection might be similar, the maximum number of sessions could be increased dramatically. Another comment is when the 3^{rd} type of traffic has many sessions in the network, the admission region projection shape is often linear, and the binding factor is often the lowest priority QoS.

CONCLUSION

We have investigated the issue of how to apply connection admission control (CAC) to differential service (DiffServ) networks. In general, although the admission regions are not always linear (always convex) under DiffServ, a linear approximation can still be applied, hence we can do the admission control under DiffServ in a conservative way. This statement is true for all different traffic types (Audio, Video and Data) under different queuing disciplines, such as priority queuing (PQ), round robin (RR) and weighted round robin (WRR). It is also possible to better approximate the non-linear admission regions with ≥ 2 linear equations. Every time there is a change in the binding factor, an extra linear equation may be required.

The shapes of the admission regions under DiffServ are often linear (e.g., when the lower priority delay is the only binding factor), trapeze (e.g., when the higher priority QoS or the lower priority jitter becomes the binding factor in the left section of the admission region) and twisted trapeze (e.g., when the queuing discipline is WRR). Recall that the twisted trapeze is composed by a negative slope straight line, an horizontal line and another decreasing straight line.

The queuing disciplines have strong influences on the admission regions. If the higher priority traffic sources require better QoS than the lower priority traffic sources, priority queuing (PQ) would have larger admission region size than rounded robin (RR). Otherwise, if both higher and lower priority traffic sources have the similar characteristics and QoS service level agreements, RR would have larger admission region size than PQ. The weighted rounded robin (WRR) admission region curves are always in between the PQ curves and the RR curves. Moreover, WRR would join the RR curve in the left section of the admission region and join the PQ curve in the right section. In between these 2 sections, there is an horizontal region connecting the 2.

The throughput (packet bit rate) of a traffic source is inversely proportional to the size of

the admission region at 1:1 ratio. Thus, the related parameters such as the packet size, the packet inter-arrival time and the server capacity are also inversely proportional to the size of the admission region, if we only vary one parameter at a time. For increasing the packet payload size while keeping the packet payload sending rate constant, the admission region would be reduced. For the QoS service level agreement parameters (such as delay), the looser QoS constraints would increase the size of the admission region (may not be directly proportional).

There is an interest finding about the jitter. Normally, the more traffic flows would cause worse QoS. For jitter, the more traffic flows may improve the jitter quality if there are only very few traffic flows in the network. This is because the variation of the packet inter-arrival time is the main factor to cause jitter to change.

The non pre-emptive mode has effect on the admission region only when the higher priority sessions outnumber the lower priority sessions and that the lower priority packet size is much larger than the higher priority packet size. If this happens, the admission region would be a trapeze, and that the higher priority QoS would be the binding factor in the left section of the admission region.

Between having 0 or 1 of a type of traffic flow (especially higher priority traffic flow) in the network, the maximum number of another type of traffic flows could be changed dramatically. The change could be huge if lower priority QoS is the binding factor for the entire admission region except when there is no lower priority traffic flow in the network.

The idea of CAC can be extended to 3 different traffics, where the admission region area would become a space. In general, when one type of traffic flows outnumber the others, its QoS would become the binding factor. The overall shape is more or less an inclined plane. Based on the projection plots, when a 3rd type of traffic has very few sessions, the projection on the first 2 traffic type plane would be very similar to the normal 2-D plot where there are only 2 traffic types. However, when the first 2 types of traffic sources do

not outnumber the 3^{rd} type of traffic sources, the lowest priority QoS is often the only binding factor, and that the shape of this section of the admission region is flat.

All simulations in this thesis were done in IP networks, without considering the MAC nor the physical layer. In the future, it would be nice to verify CAC under DiffServ taking these lower layers into account. Another possible exploration is to study the problem for a whole network (for certain topology). We also hope one day, we can find an analytical solution to the problem (a math equation to approximate the admission region without running simulations).

REFERENCES

- [1] Cristina Ortiz, “Impact de la couche MAC des réseaux sans fil locaux IEEE 802.11 sur la bande passante effective des applications IP”, jun 2007.
- [2] Kunyan Liu and Hongbo Zhu and David W. Petr and Victor S. Frost and Cameron Braun and William L. Edwards, “Design and Analysis of a Bandwidth Management Framework for ATM-Based Broadband ISDN”, *IEEE Communications Magazine*, vol. 35, pp. 1712–1716, 1997.
- [3] F.P. Kelly, “Notes on Effective Bandwidths”, in *Stochastic Networks: Theory and Applications*, F. Kelly and S. Zachary and I. Ziedins, Ed., vol. 4 of *Royal Statistical Society Lecture Notes Series*, pp. 141–168. Oxford University Press, 1996.
- [4] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, and S. Jamin, “RFC 2205: Resource ReSerVation Protocol (RSVP) — version 1 functional specification”, sep 1997.
- [5] Robbins Ethan and Sim Yong, “Weighted Fair Queuing”, 2003.
- [6] Ian Marsh, “Weighted Fair Queuing (Theory and Implementation)”, oct 1998.
- [7] Saran H. and Bruno J. and zden B. and Silberschatz A., “Early fair drop : A new buffer management policy”, *SPIE proceedings series*, vol. 3654, pp. 148–161, jan 1999.
- [8] K. Nichols and S. Blake and F. Baker and D. Black, “RFC 2474: Definition of the Differentiated Services Field in the IPv4 and IPv6 Headers”, dec 1998.
- [9] V. Jacobson and K. Nichols and K. Poduri, “RFC 2598: A Two-bit Differentiated Services Architecture for the Internet”, jun 1999.
- [10] J. Heinanen and F. Baker and W. Weiss and J. WroclawskiV, “RFC 2597: Assured Forwarding PHB Group”, jun 1999.

- [11] Timothy M. O’Neil, “Polycom White Paper: Quality of Experience and Quality of Service for IP Video Conferencing”, Tech. Rep., Polycom, 2002.
- [12] G. Almes and S. Kalidindi and M. Zekauskas, “RFC 2679: A One-way Delay Metric for IPPM”, sep 1999.
- [13] Telecommunication Standarization Sector of ITU, “ITU-T Recommendation G.1010, End-user multimedia QoS categories”, Tech. Rep., ITU, nov 2001.
- [14] Telecommunication Standarization Sector of ITU, “ITU-T Recommendation G.114, One-way transmission time”, Tech. Rep., ITU, may 2003.
- [15] Telecommunication Standarization Sector of ITU, “ITU-T Recommendation Y.1540, Internet protocol data communication service - IP packet transfer and availability performance parameters”, Tech. Rep., ITU, nov 2007.
- [16] Telecommunication Standarization Sector of ITU, “ITU-T Recommendation Y.1540, Network performance objectives for IP-based services — Version 3 Functional Specification”, Tech. Rep., ITU, aug 2008.
- [17] Telecommunication Standarization Sector of ITU, “ITU-T Recommendation G.1010, End-user multimedia QoS categories”, Tech. Rep., ITU, jul 2006.
- [18] C. Demichelis and P. Chimento, “RFC 3393: IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)”, nov 2002.
- [19] H. Schulzrinne and S. Casner and R. Frederick and V. Jacobson, “RFC 3550: RTP: A Transport Protocol for Real-Time Applications”, jul 2003.
- [20] Miroslav Voznak, “Voice over IP and Jitter Avoidance on Low Speed Links”, 2002.
- [21] Tim Szigeti and Christina Hattingh, “Quality of Service Design Overview”, Tech. Rep., Cisco, dec 2004.
- [22] G. Almes and S. Kalidindi and M. Zekauskas, “RFC 2680: A One-way Packet Loss Metric for IPPM”, September 1999.

- [23] “Internet Traffic Report: Global Traffic”, Website, April 2009, <http://www.internettrafficreport.com/main.htm>.
- [24] J.Y. Hui, “Resource allocation for broadband networks”, *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1598–1608, dec 1988.
- [25] Joseph yu Ngai Hui and Robert G. Gallager, *Switching and Traffic Theory for Integrated Broadband Networks*, The Kluwer International Series in Engineering and Computer Science. Kluwer Academic Publishers (Boston), 1990.
- [26] F. P. Kelly, “Effective bandwidths at multi-class queues”, *Queuing Systems*, vol. 9, pp. 5–16, 1991.
- [27] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “RFC 2475: An architecture for differentiated services”, dec 1998, Status: PROPOSED STANDARD.
- [28] Berger Arthur W. and Whitt Ward, “Effective bandwidths with priorities”, *IEEE/ACM Trans. Netw.*, vol. 6, no. 4, pp. 447–460, 1998.
- [29] P. L’Ecuyer and E. Buist, “Simulation in Java with SSJ”, *Proceedings of the 2005 Winter Simulation Conference (IEEE Cat. No.05CH37732C)*, p. 10, 2005.
- [30] Georgios Fellouris and George V. Moustakides, “Decentralized Sequential Hypothesis Testing using Asynchronous Communication”, aug 2009.
- [31] Douglas Altman and David Machin and Trevor Bryant and Stephen Gardner, *Statistics with Confidence: Confidence Intervals and Statistical Guidelines*, BMJ Books, may 2000.
- [32] Cheng-Shang Chang, *Telecommunication Networks and Computer Systems*, Springer, apr 2000.
- [33] Cisco, “Voice Over IP - Per Call Bandwidth Consumption”, Tech. Rep., Cisco, feb 2006.

- [34] Michael Menth and Andreas Binzenhöfer and Stefan Mühleck, “A note on source models for speech traffic”, *SIGMETRICS Perform. Eval. Rev.*, vol. 35, no. 3, pp. 22–24, 2007.
- [35] Pinnacle technical documentation, “MPEG-2 White Paper”, Tech. Rep., Pinnacle System, feb 2000.
- [36] TU Berlin Telecommunication Networks Group TKN Project, “MPEG-4 and H.263 Video Traces for Network Performance Evaluation: Video Trace Files”.
- [37] Vladimir Deart and Vladimir Mankov and Alexander Pilugin, “HTTP Traffic Measurements on Access Networks, Analysis of Results and Simulation”, in *Smart Spaces and Next Generation Wired/Wireless Networking*, Lecture Notes in Computer Science, pp. 180–190. Springer Berlin / Heidelberg, 2009.
- [38] Ba Tu Truong and Svetha Venkatesh, “Finding the Optimal Temporal Partitioning of Video Sequences”, 2005.
- [39] Lopez-Ardao Jos C. and Argibay-Losada Pablo and Rordiguez-Rubio Raul F., “On modeling MPEG video at the frame level using self-similar processes”, nov 2004.
- [40] Frank H.P. Fitzek and Martin Reisslein, “TKN Technical Report Series (TKN-00-06): MPEG-4 and H.263 Video Traces for Network Performance Evaluation (Extended Version)”, Tech. Rep., Technical University Berlin, Dept. of Electrical Engineering, Telecommunication Networks Group, oct 2000.
- [41] Frank H.P. Fitzek and Martin Reisslein, “MPEG-4 and H.263 Video Traces for Network Performance Evaluation”, Tech. Rep. 6, Technical University Berlin, Dept. of Electrical Engineering, Telecommunication Networks Group, nov 2001.
- [42] Exterity, “White Paper: An Introduction to IPTV”, Tech. Rep., Exterity, 2008.
- [43] B. Sansò and A. Girard and J.F. Frigon and H. Dahmouni, “Sympatico traffic characteristics and jitter performance model”, Tech. Rep., GERAD, mar 2008, CRD contract report for Bell Canada.

- [44] Hamza Dahmouni, “Modlisation de la gigue pour les rseaux tout IP”, apr 2009.