AGH University of Science and Technology
Faculty of Electrical Engineering, Automatics, Computer Science and Electronics

Ph.D. Thesis

Robert Wójcik

Net Neutral Quality of Service Differentiation in Flow-Aware Networks

Supervisor:

Prof. dr hab. inż. Andrzej Jajszczyk

AGH University of Science and Technology
Faculty of Electrical Engineering, Automatics, Computer Science and Electronics
Department of Telecommunications

Al. Mickiewicza 30, 30-059 Kraków, Poland
tel. +48 12 634 55 82
fax. +48 12 634 23 72

www.agh.edu.pl
www.eaie.agh.edu.pl
www.kt.agh.edu.pl

Reviewers:

prof. dr hab. inż. Wojciech Kabaciński[1]
prof. dr hab. inż. Zdzisław Papir[2]

[1] Poznań University of Technology
[2] AGH University of Science and Technology

ISBN 978-83-88309-97-7

Cover and layout design by Rafał Stankiewicz

Printed in Poland
by *Drukarnia Cyfrowa EIKON PLUS, ul. Wybickiego 46, Kraków*

To my loving wife, Izabela

Acknowledgements

Many people have helped me throughout the course of my work on this dissertation over the past four years. I would like to express my gratitude to all of them and to a few in particular. First of all, I would like to thank my supervisor, Professor Andrzej Jajszczyk, for his invaluable comments, advice and constant support during my whole research. I am sure that without his patience, broad vision and motivation, completing this dissertation would not have been possible.

I have been fortunate to meet James Roberts, the founder of the original concept of Flow-Aware Networking, and discuss several issues with. Our joint work on the project concerning FAN architecture was a milestone in my vision of the future Internet and strongly contributed to my understanding of the ideas and problems related to admission control, scheduling and network design.

Much of my experience has been formed through the collaboration with Jerzy Domżał, my friend and collegue. His insight and remarks concerning various issues have contributed significantly to the improvement of my results. It has been a pleasure to work with him and I look forward to many fruitful discussions in the future.

Last, but not least, I would like to express my deepest gratitude towards my family for their continuing love and support. I would like to thank my parents for creating the perfect conditions for me to learn and to pursue my PhD studies. I am also deeply indebted to my dear wife Izabela for her love and understanding through my graduate years. She has always stood behind me and given me unconditional support, even if that has meant sacrificing some of the time we spend together.

Abstract

Flow-Aware Networking (FAN) architecture is evaluated in this dissertation with regard to its service differentiation and quality of service (QoS) assurance capabilities. The network neutrality debate is presented first, and it is shown that the potential resolutions will have a strong impact on QoS architectures. It is also shown that FAN, as well as all the proposed mechanisms, is perfectly suited to the future Internet and in-line with the network neutrality principle. Secondly, the detailed concept of FAN is presented and compared with other flow-based QoS architectures designed for IP networks. It is argued that all the solutions have their advantages and disadvantages, however, in FAN, the pros outnumber the cons in comparison with other architectures. Not only is it net neutrality compliant, but also efficient and scalable.

The main goal of the dissertation is to propose and evaluate new mechanisms to enhance service differentiation capabilities of FAN architecture. The waiting times phenomenon as a result of admission control functionality is documented. Next, differentiated blocking and differentiated queuing mechanisms are proposed. Those mechanisms offer improved prospects of providing differentiated treatment for end-user flows. The Static Router Configuration approach is also presented, as a feasible method of implementing the new mechanisms. Finally, Class of Service on Demand is shown as the ultimate method of providing rich service differentiation without violating the network neutrality principle.

Although FAN offers QoS protection, the basic method is inefficient. This leads to fair rate degradations shown and analyzed in-depth. Several solutions to the problem are presented in the dissertation. The static limitation mechanism is proposed as a simple, yet efficient way of improving service assurance. It is shown that the limitation mechanism significantly contributes to FAN's scalability and yields great performance benefits. The static mechanism can be enhanced by the dynamic limitation mechanism which offers better results, although, only provided that the mechanism is properly set up, which is not a trivial task. To overcome this drawback, an automatic intelligent limitation mechanism is proposed which can adjust to current network conditions and is not dependent on the proper setup.

Finally, the predictive approach is presented, which changes the functioning of the admission control block in FAN. Instead of waiting for congestion to appear and only then blocking new connections, the mechanism takes a pro-active approach and starts to act on the basis of the predicted values of the congestion indicators. This enables the admission control block to react appropriately even before congestion occurs. It is shown that the best results are obtained when the predictive approach is combined with the limitation mechanism.

Keywords: Flow-Aware Networks, FAN, service differentiation, admission control, quality of service, QoS, net neutrality

Streszczenie

Tematem rozprawy jest architektura sieci zorientowanych na przepływy FAN (*Flow-Aware Networks*) ze szczególnym uwzględnieniem jej możliwości różnicowania oraz gwarantowania jakości obsługi. Na wstępie przedstawiona jest debata dotycząca neutralności sieci. Przedstawione są różne wizje związane z neutralnością oraz jest pokazane jak ewentualny wynik dyskusji może wpłynąć na architektury gwarantowania jakości obsługi. Wykazano również, że sieci FAN, jak również wszystkie zaproponowane w tej rozprawie mechanizmy, są zgodne z zaleceniami neutralności sieci.

Następnie przedstawiona jest architektura sieci FAN i porównana z innymi architekturami zapewniania jakości usług w sieciach opartych na protokole IP, które, podobnie jak FAN, za jednostkę różnicowania jakości przyjmują przepływ (ang. *flow*). Pokazane jest, że wszystkie rozwiązania posiadają swoje mocne i słabe strony. Jednakże, w przypadku sieci FAN, zestaw plusów wyraźnie dominuje nad minusami. Sieci FAN są nie tylko zgodne ze standardami neutralności sieci, ale również stanowią propozycję o dobrej wydajności i skalowalności.

Głównym celem rozprawy jest zaproponowanie i ocena nowych mechanizmów, które rozszerzą możliwości różnicowania jakości obsługi oraz poprawią gwarantowanie jakości w sieciach FAN. Efekt oczekiwania na transmisję jako wynik pracy bloku sterowania dostępem jest dokładnie opisany. Następnie, zaproponowano mechanizmy zróżnicowanego blokowania oraz kolejkowania. Mechanizmy te wprowadzają nowe możliwości różnicowania jakości usług w sieciach FAN. Dodatkowo, zaproponowano podejście statycznej konfiguracji usług w ruterach FAN, jako wydajnego i wystarczającego rozwiązania do implementacji nowych mechanizmów. Na koniec, wprowadzono klasę usług na żądanie (*Class of Service on Demand*), jako skuteczną metodę różnicowania jakości bez naruszania zasad neutralności sieci.

Mimo że sieci FAN oferują protekcję aktywnych przepływów, gwarantując im pewną minimalną jakość obsługi, oryginalnie zaproponowana realizacja tej funkcjonalności jest mało wydajna. Prowadzi to często do obniżenia przepływności sprawiedliwej (ang. *fair rate*) do poziomu znacznie niższego niż poziom gwarantowany. Ta wada jest udokumentowana i dokładnie zbadana. Następnie zaproponowano szereg usprawnień wspomnianej funkcjonalności, w tym mechanizm statycznego ograniczania liczby przepływów. Pokazano, że stosowanie tego mechanizmu znacząco przyczynia się do poprawienia skalowalności architektury sieci FAN, jednocześnie wprowadzając znaczącą poprawę wydajności. Mechanizm statyczny może być rozszerzony do dynamicznego mechanizmu ograniczeń, który daje lepsze wyniki, jednakże tylko wtedy, gdy jest bardzo precyzyjnie skonfigurowany, co nie jest zadaniem łatwym. By ominąć tę niedogodność, zaproponowano również mechanizm automatycznego, inteligentnego doboru ograniczenia, który potrafi dostosować się do panujących warunków w sieci. W tym podejściu nie jest konieczna konfiguracja ograniczeń przez operatora, co znacząco ułatwia poprawne zainstalowanie mechanizmu.

Zaproponowane jest również podejście predykcyjne. Ten mechanizm zmienia działanie bloku sterowania dostępem w sieciach FAN. W normalnych warunkach blok ten podejmuje odpowiednie działania dopiero w chwili stwierdzenia przeciążenia na łączu wyjściowym. Zaproponowane podejście zmienia działanie bloku na aktywne, tj., takie, w którym działania są podejmowane nie na podstawie aktualnego wyniku pomiaru obciążenia łącza, ale na podstawie analizy trendu i wyznaczenia najbliższej wartości oczekiwanej. Pozwala to zareagować ruterowi jeszcze zanim nastąpi przeciążenie. Wyniki symulacji pokazują, że najlepsze rezultaty można osiągnąć gdy podejście predykcyjne jest połączone z mechanizmami ograniczania liczby przepływów.

Słowa kluczowe: *Flow-Aware Networks*, FAN, różnicowanie jakości usług, sterowanie dostępem, jakość usług, QoS, neutralność sieci

Contents

List of Figures

List of Tables

List of symbols

ABR	Available Bit Rate
ACM	Association for Computing Machinery
AF	Assured Forwarding
AFAN	Approximate Flow-Aware Networking
AFD	Approximate Fair Dropping
AFL	Active Flow List
AR	Available Rate
ARS	Available Rate Service
ATM	Asynchronous Transfer Mode
BE	Best Effort
BTC	Behavioral Traffic Control
CAM	Content-Addressable Memory
CJVC	Core-Jitter-VC
CoS	Class of Service
CPU	Central Processing Unit
DiffServ	Differentiated Services
DPS	Dynamic Packet State

DRR	Deficit Round Robin
DS field	Differentiated Services field
DSCP	Differentiated Services CodePoint
EF	Expedited Forwarding
EHOT	Enhanced Hold-Off Timer
F&D	Feedback and Distribution
FAbS	Flow-Aggregate-Based Services
FAN	Flow-Aware Networking
FIFO	First-In, First-Out
FQ	Fair Queuing
FR	Fair Rate
FSA	Flow-State-Aware Transport
FTTH	Fiber-To-The-Home
GR	Guaranteed Rate
GRS	Guaranteed Rate Service
GS	Guaranteed Service
IDFA	Inter-Domain Flow Aggregation
IETF	The Internet Engineering Task Force
IFD	Intelligent Flow Discard
IntServ	Intergrated Services
IP	Internet Protocol
ISP	Internet Service Provider
ITU-T	International Telecommunication Union — Telecommunication Standardization Sector
maxPL	Maximum Priority Load Threshold
MBAC	Measurement Based Admission Control

MFAN	Multilayer FAN
minFR	Minimum Fair Rate Threshold
MPLS	Multi-Protocol Label Switching
MR	Maximum Rate
MRS	Maximum Rate Service
MTU	Maximum Transfer Unit
NANO	Network Access Neutrality Observatory
NAT	Network Address Translation
NGN	Next Generation Networks
ns-2	Network Simulator version 2
PDRR	Priority Deficit Round Robin
PFL	Protected Flow List
PFQ	Priority Fair Queuing
PIFO	Push-In, First-Out
PL	Priority Load
PON	Passive Optical Networks
PS	Predictive Service
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RFC	Request for Comments
RNG	Random Number Generator
RPR	Resilient Packet Rings
RSVP	Resource Reservation Protocol
RTP	Real-Time Transport Protocol
SCORE	Stateless Core Architecture

SFQ	Start-Time Fair Queuing
SRC	Static Router Configuration
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TIA	Telecommunications Industry Association
ToS	Type of Service
UDP	User Datagram Protocol
VC	Virtual Clock
VoIP	Voice over IP
VPN	Virtual Private Network
VRS	Variable Rate Service
WFQ	Weighted Fair Queuing
XP	Cross-Protect

Part I

Introduction and background

1 Introduction

The purpose of data networks is to satisfy human impatience.

— Andrew Odlyzko

The rapid growth and the popularity of the Internet has exceeded even the wildest expectations of its founders. In the beginning, only simple file transfers were envisioned, therefore, the IP protocol with its best effort packet delivery was introduced. The operation of the IP protocol is well suited and sufficient for these kinds of transfers; however, more demanding services have appeared over time. They include live conferencing with voice and video connections, television broadcasts, online gaming, and other delay-sensitive applications. It soon became clear that the IP protocol must be enhanced so that the network could fully support new types of services.

Since then, introducing an architecture that could guarantee quality of service (QoS) differentiation has been a hot research topic. The Internet Engineering Task Force (IETF), a key player in the Internet standardization market, has been contributing to the research. Its two flagship QoS architectures, Integrated Services (IntServ) and Differentiated Services (DiffServ), are still the most recognizable solutions providing QoS in the IP networks. Unfortunately, they have significant drawbacks and have not been deployed on a large scale in the networks. As a result, people and organizations around the world continue to devote their research to service quality, and the effects of their work are visible. Many network protocols and architectures providing service differentiation are now available, and more are in development. Architectures such as Flow-Aware Networking, Flow-State-Aware Transport and Dynamic Packet State have all emerged in recent years.

Flow-Aware Networking (FAN) differs from other QoS architectures in that it is designed to be simple, yet efficient. The idea behind such an approach is that the success of the Internet with the best effort packet delivery lies in its simplicity. In FAN, nodes do not need to exchange any explicit information between themselves; in fact they do not even require any information about the flows. All data is gathered by performing certain local measurements. The absence of any kind of signaling and the unique Cross-Protect mechanism render FAN the ultra-scalable solution, and therefore particularly well suited to the Internet.

As a result of the lack of signaling, the QoS differentiation in FAN tends to be weak. This dissertation shows how much service differentiation can be provided in FAN and what the capabilities of the architecture are. To enhance service differentiation offered by FAN, certain new mechanisms are proposed. They show that it is possible to provide rich service differentiation with the simplest means possible, even without signaling.

FAN intends to provide a minimum level of service for each active flow. It does that by blocking new flows when congestion indicators exceed their fixed thresholds. It is assumed that those thresholds define the minimum level of assured service in each FAN link. However, this dissertation shows that this assumption is incorrect, since the thresholds are significantly exceeded when many new flows arrive at the same instant. To eliminate the problem, I propose several mechanisms which alter the admission control block functionality. As the results show, the solutions are efficient and viable, and they improve the service assurance capabilities of the architecture.

The global pursuit of scalable and efficient QoS architecture for the future Internet gained a new development path a few years ago due to the emergence of the net neutrality debate. It is generally considered to be a hot topic and is widely covered in technical, economical and legal literature. The outcome of the debate is important for network engineers as it will impact QoS architectures, since certain differentiation actions work against net neutrality. However, FAN is a QoS architecture which perfectly fits into net neutrality boundaries while still providing QoS awareness. The main advantage of FAN in this context is that it provides service differentiation, taking into account just the traffic characteristics of the ongoing transmissions. As a result, it is not possible to discriminate against certain applications or end-users. Moreover, instead of providing differentiated treatment, FAN introduces fairness, which actually enhances the existing IP networks' equality. With that in mind, the proposed new mechanisms are evaluated with respect to their conformity with the net neutrality principle.

1.1 Scope and thesis

This dissertation proposes new service differentiation and quality assurance mechanisms for Flow-Aware Networking. Terms 'service differentiation' and 'QoS differentiation' are used interchangeably in the thesis and they refer to every action which alters the regular best-effort traffic delivery observed in today's IP networks. In this context, each admission control block and each queueing mechanism other than FIFO is a mean of service differentiation. Therefore, a service differentiation mechanism is a set of actions which results in providing a varied treatment to different flows. Quality assurance, on the other hand, refers to mechanisms which aim at ensuring that ceratin quality of service is provided.

All the solutions proposed in the dissertation are described in detail, implemented in the ns-2 network simulator, and thoroughly tested. The simulation analysis shows their usefulness, as well as their advantages and disadvantages. Additionally, all the presented mechanisms are evaluated with respect to the network neutrality principles.

The following thesis is proposed and proved in this dissertation:

> ***It is possible to provide Quality of Service differentiation mechanisms in Flow-Aware Networks which follow the Net Neutrality concept.***

All the proposed mechanisms are intended to be very simple. The reason behind that is twofold. Firstly, a simple mechanism is easy and therefore inexpensive to implement. However, more importantly, as FAN is a proven scalable architecture, any complicated mechanism would greatly reduce FAN's scalability. The results show that despite the proposed mechanisms' simplicity, the benefits are remarkable. One group of mechanisms improves the service differentiation capabilities of FAN, whereas the other substantially improves service assurance.

1.2 Publications

Some of the results presented in this dissertation were published in the following papers:

[113] R. Wojcik and A. Jajszczyk. Flow oriented approaches to QoS assurance. *ACM Computing Surveys (to be published)*, 2011.

[53] A. Jajszczyk and R. Wojcik. Emergency Calls in Flow-Aware Networks. *Communications Letters, IEEE*, 11:753–755, September 2007.

[112] R. Wojcik, J. Domzal, and A. Jajszczyk. Fair Rate Degradation in Flow-Aware Networks. In *Proc. IEEE International Conference on Communications ICC 2010*, pages 1 –5, May 2010.

[30] J. Domzal, R. Wojcik, and A. Jajszczyk. QoS-Aware Net Neutrality. In *Proc. The First International Conference on Evolving Internet, INTERNET 2009,*, pages 147 –152, Cannes, France, August 2009.

Paper [113] presents a survey on the QoS architectures designed for flow-based IP networks. An understanding of the ideas, advantages and disadvantages of previous concepts is vital when designing a new solution. In the paper, nine architectures are presented and compared in many aspects, including: flow definition, classes of service, proposed admission control and scheduling blocks, signaling, etc. Chapter 4 of this dissertation is a condensed version of this paper.

In [53], the concept of differentiated blocking in FAN is presented alongside the Static Router Configuration approach. The proposed solution aims to provide better performance for emergency VoIP-based connections. The simulation analysis shows that the time needed for a new connection to start depends on the amount of the offered traffic in the network, and that time can be reduced to zero by applying the differentiated blocking approach. The notion of differentiated blocking is expanded in this dissertation.

Fair rate degradation, as an effect of too much traffic in FAN, is presented in [112]. The issue is investigated through simulations. This is followed by the description of the limitation mechanism, a simple yet very efficient method of mitigating the problem. Only the static limitations are proposed in the paper. In this dissertation the concept of limiting the number of flows is presented more extensively, and some new approaches are proposed.

All the proposed mechanisms in this dissertation are analyzed with respect to their network neutrality compatibility. In [30], it is shown that FAN is a concept which fits perfectly into network neutrality boundaries, with the statement also explained in the dissertation. The analysis is extended, and covers not only the original concept of FAN, but all the proposed new mechanisms. The assessment is based on the current, most common understanding of the net neutrality principle.

The following conference papers, co-authored by R. Wójcik, concern the Flow-Aware Networking architecture, yet their scope is outside of this dissertation.

[31] J. Domzal, R. Wojcik, and A. Jajszczyk. Reliable Transmission in Flow-Aware Networks. In *Proc. IEEE Global Communications Conference GLOBECOM 2009*, pages 1 –6, Honolulu, USA, December 2009.

[33] J. Domzal, R. Wojcik, K. Wajda, A. Jajszczyk, V. López, J.A. Hernandez, J. Aracil, C. Cardenas, and M. Gagnaire. A multi-layer recovery strategy in FAN over WDM architectures. In *Proc. 7th International Workshop on*

Design of Reliable Communication Networks, DRCN 2009, pages 160 –167, Washington, USA, October 2009.

[29] J. Domzal, R. Wojcik, and A. Jajszczyk. The Impact of Congestion Control Mechanisms on Network Performance after Failure in Flow-Aware Networks. In *Proc. International Workshop on Traffic Management and Traffic Engineering for the Future Internet, FITraMEn 2008, Book: Traffic Management and Traffic Engineering for the Future Internet, Lecture Notes on Computer Science 2009*, Porto, Portugal, December 2008.

[32] J. Domzal, R. Wojcik, A. Jajszczyk, V. López, J.A. Hernandez, and J. Aracil. Admission control policies in Flow-Aware Networks. In *Proc. 11th International Conference on Transparent Optical Networks, ICTON 2009*, pages 1 –4, Azores, Portugal, July 2009.

Paper [31] shows that it is possible to assure reliable transmission in FAN. A new congestion control mechanism is proposed and evaluated through simulations. The mechanism ensures fast acceptance times of streaming flows and good transmission performance for elastic flows. The presented solution is promising and may be used in the future Internet.

In [33], a cross-layer recovery strategy for FAN built over WDM architectures is presented. The use of the Enhanced Hold-Off Timer (EHOT) algorithm [23], known from RPR networks, to control network operation after link or node failure is envisaged. Network performance after failures is also presented in [29] where the impact of proposed congestion control mechanisms in case of network overload is assessed. The results show that the acceptance times of streaming flows are relatively low even with the presence of network failures, provided that proper congestion control mechanisms are used. Both papers essentially show that FAN has great resilience capabilities.

Admission control policies proposed for Multilayer Flow-Aware Networking (MFAN) are compared in [32]. As a result, a new admission control strategy is proposed. The solution inherits advantages from established admission control proposals while ensuring fast acceptance times of new streaming flows.

1.3 Structure of the dissertation

The dissertation is divided into four parts. The first (Chapters 1, 2 and 3) provides the theoretical background for the research. Chapter 1 serves as a general introduction to the topic. Chapter 2 presents the ongoing public discussion on network neutrality. The arguments from all sides of the dispute are outlined and discussed. Furthermore, the impact of this debate on QoS architectures is presented. Chapter 3 shows, in detail, the architecture of FAN with a special focus on the admission control and scheduling mechanisms.

The second part of the dissertation contains of one chapter, Chapter 4, which surveys existing QoS architectures designed for IP networks and allowing service differentiation based on individual flows. Nine architectures are presented and aligned along the time axis. Subsequent sections compare and contrast the architectures in different aspects. Finally, the comparison is summarized and the future of those architectures is subjectively discussed. Parts 1 and 2 of the dissertation, especially Chapters 2, 3 and 4, show the state of the art and related works in QoS architectures and the network neutrality debate.

The third part of the dissertation includes two chapters. In Chapter 5, new QoS differentiation techniques in FAN are proposed. First, the notion of implicit admission control is described in Section 5.1. Then, Section 5.2 documents the waiting times phenomenon in FAN. Next, mechanisms such as differentiation blocking (Section 5.3), differentiated queuing (Section 5.4), Static Router Configuration (Section 5.5) and Class of Service on Demand (Section 5.6) are presented and evaluated. The chapter ends with an assessment of the proposed mechanisms relating to the network neutrality concepts, followed by concluding remarks. The second chapter in this part, Chapter 6, shows new service assurance mechanisms proposed for FAN. The chapter opens with Section 6.1 explaining what fair rate degradations are and why they occur. In the subsequent sections, new mechanisms to mitigate the problem are proposed, including the static limitation mechanism (Section 6.2), dynamic limitation mechanism (Section 6.3), the predictive approach (Section 6.4) and the automatic intelligent limitation mechanism (Section 6.5). Similarly to the previous chapter, this one also closes with an assessment of the proposed mechanisms relating to the network neutrality concept, followed by concluding remarks.

The fourth part of the dissertation includes just one short chapter. Chapter 7 summarizes the research and the achievements presented in the dissertation.

The attached appendix describes the procedure of conducting the experiments and presents the techniques used by the author to assure the credibility of the obtained results.

2 Net Neutrality

The fantastic advances in the field of electronic communication constitute a greater danger to the privacy of the individual.

— Earl Warren

This chapter discusses the notion of network neutrality, i.e., a concept so vastly covered in the literature, that it lost some of its meaning and gained some new. Although network neutrality is usually referred to as net neutrality, for short, both terms convey exactly the same meaning. Throughout this dissertation both terms will be used interchangeably.

2.1 Introduction

The net neutrality debate attracted an enormous amount of attention over the last few years. It may definitively be considered as a hot topic and is widely covered in the technical, economical and legal literature. One part of the attraction is, surely, due to its controversial nature, i.e., there are several parties involved and each have its own view on the matter. Unfortunately, it is not the whole story. One of the big attendees of the debate is the telecom operators, a major companies with substantial market power and visibility. They chose to actively participate in the dispute, often not to present their 'objective' opinions, but to protect their interest, as the possible legal outcome of the debate would introduce certain new regulations and restrictions aimed directly at them. As a result, the literature on net neutrality must be read with caution, as there are positions which do not present objective statements and conclusions, but rather, formulate false claims for the benefit of the authors' employer. G. Goth in [39] says “However passionate the public discussion might be, bandwidth providers and content providers will be dancing an elaborate minuet to maximize both camps' market opportunities”.

The literature is, therefore, complex to read, to say the least. After all, as the subject concerns us all, everyone is bound to produce its own opinion on the matter, which does not contribute much to objectivity. The only approach to discuss the problem is to show it from the very wide perspective, presenting the ideas and opinions from all the parties involved. That is why, in this chapter I survey the existing literature to present the most comprehensive view on the network neutrality debate, as of today. Most parts of the discussion concern the debate carried out in the United States of America, where the debate is the loudest. The rest of the World carefully monitors that dispute and tries to participate. Nevertheless, the values conveyed by network neutrality apply to all the networks worldwide.

The remainder of this chapter is organized as follows. Section 2.2 presents all the parties involved in the debate and their view on what net neutrality is. The answer to that question is not trivial, despite what may seem. This section also shows the rationale of the parties to participate in the discussion. Section 2.3 shows two issues: how the regulations were enforced in the past, and how have the telecom companies, what would now be called, violated the net neutrality principles. Section 2.4 explains the merit of the debate by showing both arguments and counterarguments of all sides of the discussion. In Section 2.5, I discuss the relationship between network neutrality and QoS architectures in IP networks. This relationship is interesting, as it shows whether or not, it is possible to use service differentiation mechanisms without violating the net neutrality principles. Finally, Section 2.6 shows some currently ongoing actions with the focus on preserving the neutral Internet. Also, the possible future of the debate is discussed.

2.2 The definition of net neutrality

Gilbert Held in [48] opens the discussion with the sentence: “*Net neutrality represents one of a few telecommunications terms that, while very difficult to precisely define, can cause a large amount of conversation on both sides of the issue*”. This observation is obviously true. Most of us associate ‘neutral’ as a positive term, and agree that the Internet should be neutral of some sort. However, ask various parties about the meaning of ‘the neutral Internet’ and you are likely to receive different answers. This ambiguity (or the lack of precision) started many unnecessary discussions and accusations. Pierre Larouche says that "*«Network neutrality» has become a slogan of sorts, which covers a more complex reality than either side of the U.S. debate is willing to admit*"[1] [69].

The discussion has also acquired major political attention in the U.S., mostly because the issue is already well-known and controversial, but also because initiatives such as savetheinternet.com [107] or similar urge people to contact with their representatives to act on their behalf. Senator Barack Obama in one of the presidential campaign televised political discussions said: "*I am a strong supporter of net neutrality. (...) What you've been seeing is some lobbying that says that the servers and the various portals through which you're getting information over the Internet should be able to be gatekeepers and to charge different rates to different Web sites... And that I think destroys one of the best things about the Internet.*"

The most common understanding of the phrase net neutrality can be found on the savetheinternet.com website [107]. It is an American web page which consolidates the nationwide movement to legalize the neutral Internet. Their definition is as follows: "*Net Neutrality is the guiding principle that preserves the free and open Internet. Net Neutrality means that Internet service providers may not discriminate between different kinds of content and applications online. It guarantees a level playing field for all Web sites and Internet technologies*". The website authors' say that the Internet should remain open, meaning that it is available to every user or company, and free, i.e., everybody is free to use it however she/he likes. It does not mean, and has never had, that users should not pay for the access to the Internet. It is only natural that people pay fees to their Internet Service Providers (ISPs) to gain the access with the quality proportional to what they pay. This might seem obvious, yet in certain publications, it can be found that net neutrality proponents demand for the Internet access to be delivered to every home for free.

The second part of the definition from [107] is more important. It says that the telecom operators should not be allowed to differentiate the traffic based on its content, application, source or destination. In other words, the operators should be prohibited to:

1. provide better QoS for certain applications or users,
2. charge more for using certain applications.

The reasons for such statements lie in the fact that telecom operators might manipulate the traffic in their networks for their own benefit. In net neutral reality, the network's only function is to transmit data — not to choose which

[1] Although the author refers to the U.S. debate, the statement is more general and concerns net neutrality worldwide.

Table 2.1: The definition of network neutrality by all the involved parties

Debate side	How do they see net neutrality?	Vote
Telecom operators	Unnecessary regulations (market rules are sufficient)	NO
Content providers	Fair competition, no double payments	YES
Technically aware users	Challenging, yet important step	YES/NO
Technically unaware users	Freedom	YES

data should be privileged with higher service quality. Net neutrality wants the operators to be only 'carriers' of data, and their sole responsibility should be to get data from one side of the globe to the other, without caring what is inside the packets. This proposed regulations are based on the loud examples from the past when Internet providers tampered with the users' traffic to obtain monetary benefits by blocking or restricting the access to some services otherwise publicly available. The list of well-known violations of the net neutrality principles is presented in Section 2.3.

Judging by the already mentioned parties involved, the debate might seem two-sided: the offense against the telecom operators by everybody else. Of course, the issue is much more complicated and, therefore, leaves room for the most elaborate discussions. In its course, more groups heavily interested in their outcome have emerged. To simplify, the defending side is represented by large nationwide telecom operators (e.g., AT&T, Verizon), Internet service providers and other network traffic carriers, whereas, the other side consists of content providers (e.g., Google, Skype) and regular users of the Internet. The defending side is mostly unanimous in their views. Unfortunately, the other side is not. Partly due to the fact, that it is represented by many groups: standard Internet users, networking specialists, politicians, lawyers, economists, businessmen and small to large companies. Furthermore, among the groups, there are people who are aware of how networking works and those who propose sound solutions but with no possible implementation in reality. Jon Crowcroft says that "Much of what I have read on the subject of net neutrality by economists is technically naive and simplistic" [19].

The definition of the term net neutrality differs among sides. Tables 2.1 and 2.2 summarize the most common perceptions of each involved party. The telecom operators, obviously, are against any resolutions which would enforce new regulations upon them. They feel that the free market mechanisms are sufficient guardians of the existing status quo, and new regulations would only hinder further development. On that grounds, telecom operators are against putting the network neutrality principles into law. Content providers are on the

Table 2.2: The expected effects of possible network neutrality law enforcements through the eyes of all the involved parties

Debate side	How do they see the effects?
Telecom operators	The regulations will lessen the revenues, therefore, hindering the development of the networks
Content providers	The regulations will ensure fairness and promote development
Technically aware users	Regulations will promote fairness but may make network management more challenging
Technically unaware users	The regulations will enforce free and fair Internet

completely opposite side of the debate. They argue that only by law, can those large influential companies be forced to provide fair competition among them. Large content providers fear that they might be extorted to share significant amount of revenues just to be able to exist in the Internet. Small, innovative, start-up companies are afraid that they might not be able to effectively sell their ideas (and hence develop them) if they are forced to pay substantial fees from the start. The general public opinion on that matter believes that the possible unfair behavior of the telecom industries may stop the development of small Internet-based businesses worldwide.

The last side of the discussion are the users of the Internet. Those with no backgrounds in networking feel that the neutral Internet is the only fair solution and it should be preserved. People associate net neutrality with freedom of speech, freedom of choice of application or service. They also reckon that if the Internet works just fine now, there should not be any changes in the future, maybe apart from the possible speed increase. To some extent that line of reasoning seems viable. However, there are certain aspects to which the proposed network neutrality demands are risky. Specialists in networking argue that e.g., only by looking into payloads of the carried packets can the operators protect the users from certain attacks. Therefore, while the group's standing is divided, most technically aware users feel that the network neutrality principles are valid and important, however, their enforcement must be carried out with utmost caution and rationality.

2.3 The history

Georges Santayana, an American philosopher once said that "Those who do not study history are doomed to repeat it". Although net neutrality might seem like a relatively new concept, due to its possible legislative restrictions, it is important to be aware of how similar regulations have impacted the companies in the past.

Also, to fully protect the users against unwanted practices from the telecom operators, we need to know what sorts of abuse happened in the past. Both issues are dealt with in the following sections.

2.3.1 Regulations in the past

The debate over network neutrality and the possible upcoming regulations are often compared to previous legislative motions in the U.S., i.e., the regulations of the postal service and the telephone companies. Fred Schneider in [100] says that "The 1984 breakup of AT&T radically changed the telephone business in the U.S. More than a quarter-century later, the action has shifted from telephone voice networks to wireless networks and the Internet." The reason behind such a comparison is twofold: firstly, the regulations concern large companies with substantial market power, usually with monopolistic (or close to monopolistic) inclinations, and, secondly, the proposed resolution revolves around the 'common carrier' approach which now governs the telephone companies.

In [89], the reader can find a comparison of the current network neutrality debate to previous attempts to regulate commerce and the telephone companies. The author tries to point out the failures of the previous legislative motions and shows their consequences. He says that in the past: "In many cases, consumers would have been better off without regulation. The starkest evidence: deregulation of airlines, trucking and most rail rates actually produces lower prices" [89].

It is also argued, that regulated commerce is much less innovative than the monopolistic one. The author claims that "Bell Labs was a famous source of invention, but AT&T was a ponderous and reluctant innovator" [89]. To some extent it may be true, but many would disagree. It is true that monopolists, due to their almost infinite funding, can conduct research also on technologies or services which have small chance of success. In the competitive market, only the well promising research is conducted, if any. However, the competitive market develops much faster and constantly probes the market to find new solutions because companies feel the breath of their competition. Apart from that, the monopolists globally fail to satisfy consumers on other levels of their operation. The final thought, on which everybody agrees, is that when creating new laws for the preservation of the neutral Internet, the experiences from the past must be carefully considered.

2.3.2 Net neutrality violations

The fact that network neutrality has become such a widely discussed topic has its roots in the past, when certain telecom companies violated the net neutrality principle. Table 2.3 shows only the best known instances of acts against the

Table 2.3: Documented violations of network neutrality principles [37], [109]

Operator Content Provider	Year	Discrimination
Madison River Vonage	2004	Vonage and other rival VoIP services were blocked
Telus TWU	2005	website sympathetic with TWU was blocked
AOL www.dearaol.com	2006	website was blocked for critiquing AOL's pay-to-send email scheme
Comcast P2P, Vuze	2007	all P2P connections shut down or severely degraded
Verizon Wireless NARAL	2007	denied to be able to send text messages through the network
AT&T Pearl Jam	2007	deleted words criticizing the American president

TWU: Telecommunications Workers Union
NARAL: NARAL Pro-Choice America

neutrality, however, such violations happen every day, everywhere, only they are either not exposed, or publicly recognized.

The first loud dispute happened in 2004, when Madison River, the telephone company and an ISP from North Carolina, U.S., blocked the Vonage VoIP service from their customers on the DSL lines. The Vonage service competed with the standard PSTN telephone service offered by the operator and was constantly stealing some of the revenues. One year later, the Canadian second largest ISP, Telus, blocked the access for its users to the website run by a member of the Telecommunications Workers Union (TWU). At that time, Telus and TWU were engaged in a harsh labor dispute.

Probably, one of the most recognized disputes was held between Comcast, the second largest Internet provider in the U.S. and the P2P environment, represented by Vuze, the Bittorrent application. Beginning around May 2007, Comcast began to block certain Internet communication protocols, including P2P protocols: Bittorrent and Gnutella. Comcast did not deny the blockings, but instead justified them. They claimed that their networks were not designed to provide Bittorrent service and such a service deteriorates other services in the network. Instead of investing in the development of their network, a simpler solution was to block the unwanted protocol. However, there is more to it than meets the eye. By blocking Bittorrent, Comcast got rid of Vuze, the application which legally delivered television content to end users based on the peer-to-peer protocol and threatened Comcast's traditional cable-based content delivery. More on the Comcast case can be read in [110].

In 2006, America On-Line (AOL) blocked the website that put some negative words about the AOL's new pay-to-send email scheme, thereby, discouraging users against this new service. Similarly, in 2007, Verizon Wireless denied certain messages to be rightfully forwarded through their network, and AT&T deleted words criticizing J. W. Bush, said by a singer of the Pearl Jam band at a transmitted concert. Both those censorship acts by the telecom companies were conducted for political judgement or personal beliefs.

Those mentioned violations of the network neutrality principles happened in the past and acquired enough public attention to be recognized worldwide. However, such malpractices happen more often than we can imagine. The fact that an ISP favors its own service over the competing services is not uncommon. For example, one of the Polish ISPs favors its VoIP connections over all the other kinds of traffic, which in terms of congestion, results in better quality of their service. The possibilities are boundless. In many publications, e.g., in [43] or [48] the reader can find more examples of possible violations of the network neutrality principle.

2.4 The debate

In this section, I present in more details the merits of the network neutrality debate. I try not to take any side and be objective as far as possible. Therefore, the arguments and counterarguments are shown, so that the reader may form his/her own opinion.

Section 2.3 showed what were the reasons behind the discussion and how the debate started. From what can be observed, the large body of Internet users, despite being strongly interested, are underrepresented in the discussion. This is because operators and large content providers like Google or Yahoo are able to make their voices heard. For example, in 2006, Ebay.com emailed over 1 million of their customers urging them to support the legislation. Similarly, Google CEO Eric Schmidt wrote an open letter to Google users asking them to take active steps to protect the Internet freedom. However, at the same time, it is estimated that telecom and cable companies in the U.S. have been spending 1 million dollars per week on advertisements the oppose to network neutrality legislation steps [72]. On that background, the users voice is presented only on websites such as [107].

2.4.1 The proponents perspective

Net neutrality is the reason the Internet has driven economic innovation, democratic participation and free speech online. It protects the consumer's right to use any equipment, content, application or service without interference from the network provider. The proponents feel that they need strict regulations to protect them, because:

1. violations of the net neutrality principles have already happened in the past and are likely to happen in the future,
2. most homes have little or no choice between broadband Internet access providers,
3. if the Internet users want to use all the possible applications and services, the operators should not decide for them,
4. network access providers should not be allowed to inspect the content of the transmission for the sake of privacy.

Although the exact demands of various network neutrality proponents are not identical, the Internet Freedom Preservation Act from January 2007, introduced by eight U.S. senators, including senators Barack Obama and Hillary Clinton, enumerates many of them. The restrictions are summarized in [20]. According to the document, a broadband service provider:

1. may not block, impede, discriminate or degrade the ability of any person to use a broadband service to access, use, send, post or offer any lawful content, application or service available on the Internet,
2. cannot prevent users from attaching a physical device on the network, as long as the device does not degrade or damage the network,
3. must provide clear terms of service to their subscribers, explaining the access type, speed and limitations applied,
4. cannot impose a charge on the basis of the usage of the network,
5. cannot charge for prioritization of traffic,
6. cannot require a subscriber to purchase additional services to receive some content.

The telecom companies refuse to meet those demands explaining that the fact that Internet works so fine is a result of unregulated competition on the market. The laws of the market, in the eyes of telecom syndicates, will be sufficient to protect consumer rights. AT&T chairman Ed Whitacre in March 2006 said that: "Any provider that blocks access to content is inviting customers to find another provider. And that's just bad business" [1].

Unfortunately, as is well shown in [109], for a number of reasons that is not the case. Firstly, if all network providers block the same applications, there will be no one to switch to, and the choice is not that great to begin with. Secondly, customers do not have an incentive to switch if they do not realize that their operator interferes with the traffic. Thirdly, switching to another ISP requires significant time, effort and money, as most consumers signed timed contracts. Finally, if tampering with the users traffic is such a 'bad business', why do operators argue that they need to do it to develop the infrastructure with the earned money? This argument, essentially concedes that ISPs have incentives to discriminate in order to increase their profits.

The economists Economides and Tag in [35] presented a two-sided market analysis, a mathematical model to assess the network neutrality. Even though certain assumptions were made, the authors claim that net neutrality is good for total welfare. More and most recent information about the proponents perspective can be found at the savetheinternet website [107].

2.4.2 The opponents perspective

As any possible regulations related to the network neutrality will be inconvenient, to say the least, for network operators, they are against them. Some say that the values of net neutrality are worth respecting, however, to put them into law will break the Internet [86]. There are also numerous works by economists or law professors with little background on networking which state that for the number of reasons, the legislative approach is unnecessary or even harmful [45], [74].

The telecom operators explain that network neutrality should not be legalized because:

1. broadband service providers should be allowed to control traffic inside their own network as they want for the benefit of the users,

2. the Internet is not neutral today: quality of service is and needs to be applied for certain applications to work,

3. the additional stream of revenues from providing differentiated treatment will allow for more investments in the infrastructure which, in turn, will result in a better overall quality for all the users,

4. broadband competition is increasing and users are free to switch to another operator if they are not satisfied with the enforced traffic policies,

5. network administrators need to be able to inspect packet payloads in order to defend their networks against certain attacks,
6. the market competition is sufficient for the operators to refrain from any bad behavior.

Most of the arguments mentioned above do not convince the users and content providers. A. Odlyzko in [85] criticizes telecom operators for convincing people that they need additional revenues to build the future Internet and that those funds will not come if the network neutrality principle is enforced. The argument that the market rules are sufficient to maintain fairness can be simply overruled by the examples of network neutrality violations from the past. If the market laws did not apply then, we should not hope for them to apply in the future.

Very often the position of the proponents is displayed as they ask for total net neutrality, i.e., lack of possibility to even discard viruses, SPAM or denial of service attacks traffic. This is an attempt to make net neutrality look absurd. Network neutrality proponents never proposed that. Instead, users fear that ISPs will gain the power to completely model everybody based on his/her behavior in the Internet by collecting information stored in our transmissions [55].

More and most recent information about the opponents views can be found at the NETCompetition website [81].

2.5 How does net neutrality impact QoS?

The lack of commercial revenue prospects inhibits the development of QoS architectures. Moreover, the network neutrality debate will turn decisive for the future of QoS. If network neutrality is the vision in which the network operator is not allowed to discriminate traffic of certain users or applications and favor the others, many QoS architectures are simply unusable. Although the outcome of the debate is unclear, most definitely, it will impact the future QoS development. Therefore, even now, QoS architectures are assessed with respect to their neutrality.

The technical aspects of the debate revolve around how to provide service differentiation in a neutral way. XP. Xiao in [115] proposes a new business model, in which QoS is not sold explicitly, but rather it is put into the services and sold as a package. He proposes that service providers sell their services in the form of bandwidth blocks with embedded QoS price. Each block has its own amount of bandwidth and a set of QoS policies to enable certain applications. Therefore, ISPs do not discriminate users, as nobody pays extra. With regards to network neutrality, the author defends his proposal. He says that network neutrality opposes traffic discrimination against different application providers depending on whether they pay a QoS fee. However, net neutrality does not oppose network service providers from raising or lowering their service price as long as it applies uniformly to all businesses and people. Having that in mind, the proposed business model would not cause controversy in that field.

Common QoS architectures, including IntServ and DiffServ, provide means for network operators to differentiate the service without any limitations. It is possible to discriminate traffic based on virtually anything the operator decides such as: the application type, source or destination addresses, traffic volume, etc. It is also possible to implement a Deep Packet Inspection mechanisms [22] and police the traffic based on its mother application or content. However, since most of the differentiation actions are against the net neutrality, choosing such a powerful and complex solution is neither useful, nor cheap. The real goal, therefore, is to find a solution which could be used with the IP protocol, would be simple, efficient, scalable, and in conformity with the network neutrality rules.

An example of such an architecture is FAN [87]. In [30], it is shown that FAN is a QoS architecture which perfectly fits into the mentioned neutrality boundaries while providing QoS awareness. The main advantage of FAN, with respect to the net neutrality issue, is that it provides service differentiation, taking into account only the traffic characteristics of the ongoing transmissions. Therefore, it is not possible to discriminate certain applications or end-users. Moreover, instead of providing differentiated treatment, FAN introduces fairness, which even enhances the current IP network equality.

2.6 The future of net neutrality

If there were definite and ultimate answers, the debate over network neutrality would not have been so difficult. Americans are working intensely to come up with a resolution, whereas the rest of the world carefully observes. However, we should realize that the broadband market situation is not the same in every country. The authors of [19], [40], [60] and [69] show the differences in telecommunications regulations and local broadband markets in Korea, UK, and European Union, respectively. In [69], we read that: "... the landscape in Europe looks different than in the U.S. and is likely to remain so in the foreseeable future: fewer competing infrastructures, but more market players (...)". While it is true that we will not be able to produce a 'one fits all' resolution to the network neutrality problem, we can always learn from predecessors' mistakes and only slightly adjust our solutions. Some scholars say that current approach to network neutrality problem is not right, and they propose a new approach, one that allows nondiscriminatory network management and QoS provisioning but prohibits discriminatory use of the network infrastructure. Such an approach is presented in [56].

Although the debate has been active for a couple of years now, and no resolutions have been enforced, it has not reached stalemate. Even more than ever, the new proposals related to the debate emerge. In [106], the authors present Network Access Neutrality Observatory (NANO), a system which detects network neutrality violations. The system discovers when an ISP applies policies that discriminate against specific classes of applications, users or destinations. A product compares the performance of a particular service to the performance of the same service through other ISPs. The authors claim that NANO can detect violations very effectively. Such initiatives put pressure on the network operators, as they can no longer hope to hide their malpractices and discriminations. Moreover, people have more tools to check if their operator applies any traffic policies to their transmissions.

There is also a sign that operators started to care enough about network neutrality, or fear the consequences. In August 2010, Google and Verizon have publicly announced their joint policy proposal for an open Internet [38]. Those companies see their proposal as a compromise. The plan is to preserve the open Internet while allowing network operators the flexibility and freedom to effectively manage their networks. A broadband Internet access provider would be prohibited from preventing users from:

1. sending and receiving lawful content of their choice,
2. running lawful applications and using lawful services of their choice,
3. connecting their choice of legal devices that do not harm the network or service, facilitate theft of service, or harm other users of the service.

This framework proposal may result in nothing concrete, however, it shows, to the rest of the community, that the network neutrality should be considered seriously. I strongly believe it is possible that someday we will reach the free and completely neutral Internet, yet only if enough effort will be put into the design. [19] ends with the following statement: “We never had network neutrality in the past, and I do not believe we should engineer for it in the future either”. While the first part is simply true, I cannot agree with the second.

3 Flow-Aware Networking

QoS is ... Quite often Stupid!

— James Roberts

The success of the Internet lies in its simplicity, however, it comes with the cost of only best effort non-differentiated service. For years, institutions like IETF, tried to introduce a QoS architecture to the current IP network. Unfortunately, the proposed QoS models, i.e., Intergrated Services [11] (IntServ) and Differentiated Services [8], [83] (DiffServ) are not suitable for the whole Internet. To provide a service at a reasonable level, under the terms of congestion, some priorities and discriminations must be imposed. The mentioned architectures proposed the use of a reservation protocol and packet marking scheme, respectively, however, these solutions require proper inter-domain agreements, complex router implementations, and most of all, the end-user compliance. Beside IntServ and DiffServ, many other QoS architectures have been proposed for the IP networks. They are reviewed in this chapter.

The efficient and robust QoS architecture for the IP networks requires that the user-network interface remains the same as today, no signaling protocol or packet marking is introduced, no new user-operator or operator-operator agreements are signed. These constraints are very strict, yet they have been met. This chapter introduces a novel approach to achieve QoS guarantees in the Internet — Flow-Aware Networking, or FAN for short.

The description of FAN starts with Section 3.1 which shows why a new QoS architecture is needed. Section 3.2 describes the basic concepts of FAN. Sections

3.3 and 3.4 introduce the flow-aware approach and Cross-Protect (XP) router, respectively. Section 3.5 describes one of the FAN-specific mechanisms i.e., measurement based admission control, while fair queuing algorithms are presented and compared in Section 3.6. Finally, Section 3.7 shows what other mechanisms and architectures were proposed for Flow-Aware Networks.

3.1 The need for a new QoS architecture

IETF introduced two ideas on how to assure QoS. Chronologically, the first was Integrated Services. IntServ has many advantages, like: real (in opposed to statistical) assurances, easy controlling in nodes, using the reservation protocol, possibility to create various traffic profiles. However, there are certain disadvantages, which make IntServ unsuitable for larger networks. These include, e.g., keeping information about all flows in every node, demanding from end-users to explicitly define required transmission parameters. These pros and cons make IntServ a good solution when dealing with a small network, where all end-users are known, traffic is mostly defined, and every router in the network can be easily configured by one network operator.

To overcome the scalability issue, IETF introduced a different idea — the Differentiated Services. At the cost of certain constraints, DiffServ omits problems that have eventually stopped the development of its predecessor. That is the reason, why in DiffServ the assurances are statistical and the admission control blocks are placed only at the borders of each DiffServ domain. Moreover, the inner nodes do not keep the flow information, which suits it better for larger networks, however, the scalability issue is not completely overcome. Still, all routers in a domain must be pre-configured, so the per-hop behavior would match the actual classes of service, which are provided inside the domain. As DiffServ is more flexible and scalable than its predecessor, it still holds features, which make it unsuited for extra large networks, like the Internet.

In [111], an opinion, that IntServ and DiffServ represent a trade-off between fine service granularity and scalability, and therefore the trade-off between scalability and QoS exists, is expressed. Over the years, many attempts to alleviate this strict relationship have been proposed, including: combined use of IntServ and DiffServ or new and better congestion control mechanisms cooperating with service isolation provided by DiffServ.

3.2 Basic concepts of FAN

Flow-Aware Networking is a new direction of the QoS assurance in IP networks. The original idea was initially introduced by J. Roberts et al. in [10], [95] and, then, presented as a complete system in 2004 [67], [94]. Their intention was to design a novel QoS architecture, so it would be possible to use it in networks of all sizes, including the global IP network — the Internet. In [87] the belief, that an adequate performance can be assured much more simply than in classical QoS architectures, and more reliably than in over-provisioned best effort networks, is expressed.

The goal of FAN is to enhance the current IP network by improving its performance under heavy congestion. To achieve that, certain traffic management mechanisms to control link sharing are proposed, namely: measurement-based admission control [87] and fair scheduling with priorities [67], [66]. The former is used to keep the flow rates sufficiently high, to provide a minimal level of performance for each flow in case of overload. The latter realizes fair sharing of link bandwidth, while ensuring negligible packet latency for flows emitting at lower rates. All the new functionalities are performed by a unique router, named: Cross-Protect router. This device alone is responsible for providing admission control and fair queuing.

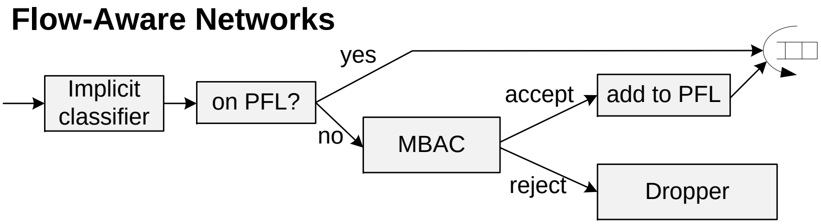


Figure 3.1: Operation of FAN

Figure 3.1 illustrates the operation of FAN. All incoming packets are, firstly, classified into flows. The flow identification process is implicit and its goal is not to divide flows into different classes, but only to create an instance on which the service differentiation will be performed. Then, all the flows that are currently in progress, i.e., are present on the Protected Flow List (PFL) are forwarded unconditionally, whereas all new flows are subject to admission control. The admission control in FAN is measurement based (MBAC) which implies that the accept/reject decisions are based only on the current link congestion status. If a new flow is accepted, it is put onto the PFL list and then all forthcoming packets of this flow are forwarded without checking the status of the outgoing link by MBAC.

In FAN, admission control and service differentiation are implicit. In classic explicit service differentiation architectures, user requirements are signaled to the network and the nodes perform differentiation actions based on the obtained information. For example, the way of providing better quality of service results from explicit identification of a certain transmission. On the contrary, implicit service differentiation performs differentiation actions based on traffic characteristics and network measurements.

In FAN, there is no need for *a priori* traffic specification, as well as there is no class of service distinction. Both streaming and elastic flows achieve a necessary QoS without any mutual detrimental effect. Nevertheless, streaming and elastic flows are implicitly identified inside the FAN network. This classification, however, is based solely on the current flow peak rate. All flows emitting at lower rates than the current *fair-rate* are referred to as streaming flows, and packets of those flows are prioritized. The remaining flows are referred to as elastic flows.

FAN is supposed to be suited even for the whole Internet. This is due to some constraints, that were imposed by the designers. First of all, nodes do not need to exchange any information between themselves, they even do not need any explicit information about the flows. All information is gathered through local measurements. The Flow-Aware Networking is based on the XP mechanism, which enhances the current IP router functionality. The XP mechanism is presented in Section 3.4.

One of the most important aspect of FAN is that it is only an enhancement of a currently existing IP network. Both networks can easily coexist. Moreover, the advantages of FAN can be seen even if not all nodes are FAN based. That means that it is possible (and advised) to gradually improve the network by replacing nodes, starting from the ones that are attached to the most heavily congested links.

3.3 Flow-aware approach

FAN is flow oriented. It means that traffic management is based on user-defined flows. The definition of a flow in Flow-Aware Networking comes from [87]: “By flow we mean a flight of datagrams, localized in time and space and having the same unique identifier”. The datagrams are localized in space, as they are observed at a certain interface, (e.g., on a router) and in time, as they must be spaced by no more than a certain interval, which is usually a few seconds. The space localization causes that a typical flow has many instances, one at every interface on its path.

The identifier is obtained from certain IP header fields, including IP addresses and some other fields, e.g. IPv6 *flow label*. One idea is to allow users to freely define flows, that correspond to a particular instance of some application. The intention is to allow users as much flexibility as possible in defining what the network should consider as a single flow. Such an approach is surely beneficial for the user, however, it always introduces the possibility of malicious behavior. A flow label may also be deduced from IPv4 header fields. Typically, it could be a standard 5-tuple, though, this approach limits the flexibility, allowing users no control in defining their flows.

All flows in FAN are divided into either a streaming or elastic type, hence two classes of service. The distinction is implicit, which means that the system categorizes the flows based on their current behavior. There is no need for *a priori* traffic specification as the classification is based solely on the current flow peak rate. All flows emitting at lower rates than the current fair rate[2] are referred to as streaming flows, and packets of those flows are prioritized. The remaining flows are referred to as elastic flows. The association of a flow with a certain class is not permanent. If a flow, initially classified as streaming, surpasses the current fair rate value, it is degraded to the elastic flow category. Analogously, a flow is promoted to streaming, if, at any time, it emits at lower rate than the current fair rate. Note that both factors, i.e., flow's bitrate and current fair rate can change.

The assumption of FAN was to provide two classes of service. For low-rate flows which are typically associated with streaming transmissions, the streaming type is considered. All flows classified as streaming receive prioritized treatment — packets of those flows are sent through priority queues, hence, little delays and delay variations. For the rest of the flows, proper fair queuing algorithms provide fair bandwidth sharing which cannot be assured with standard FIFO-based queueing disciplines. Finally, the distinctive advantage of FAN is that both streaming and elastic flows achieve sufficiently good QoS without any mutual detrimental effect.

3.4 Cross-Protect mechanism

To install FAN in a network, an upgrade of current IP routers is required. Figure 3.2 shows a concept diagram of an XP router, the standard interconnecting device in FAN. FAN adds only two blocks to the standard IP router, namely the admission control and scheduling blocks. The former is placed in the incoming line cards of the router, whereas the latter is situated in the outgoing line cards.

Admission control is responsible for accepting or rejecting the incoming packets, based on the current congestion status. The purpose of scheduling is twofold: it provides prioritized forwarding of streaming flows and assures fair sharing of the residual bandwidth by all elastic flows. If a packet is allowed, the flow associated with it may be added to the PFL, and then all forthcoming packets of this flow will be accepted. The admission control block realizes the measurement based admission control functionality, which is described in Section 3.5. The scheduler is responsible for queue management. It is a very important block, as

[2] fair rate is one of the measured indicators of the link condition and is defined in Section 3.6

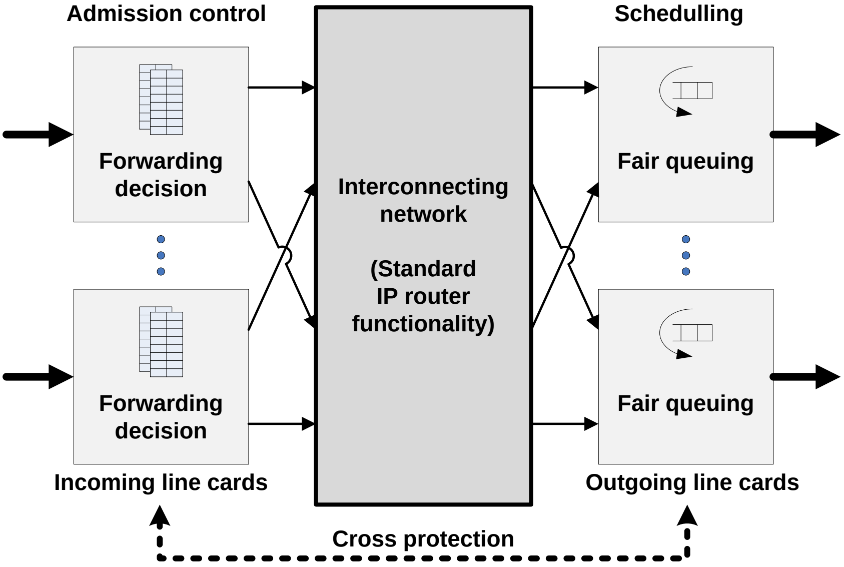


Figure 3.2: Concept diagram of a Cross-Protect router [67]

it have to ascertain that all flows are equally treated. All flows that currently have at least a packet in a queue, are added to the Active Flow List (AFL). The detailed information on scheduling algorithms is provided in Section 3.6.

Naming FAN devices as "Cross-Protect routers" is a result of mutual cooperation and protection, which exists between both extra blocks. The admission control block limits the number of active flows in a router, which essentially improves the queuing algorithm functionality, and reduces its performance requirements. It is vital that queuing mechanisms operate quickly, as for extremely high speed links the available processing time is strictly limited. On the other hand, the scheduling block provides admission control with the information on congestion status on the outgoing interfaces. The information is derived based on, for example, current occupancy of the queues. The mutual protection contributes to smaller protected flow list and active flow list sizes, which significantly improves FAN's scalability.

The advantage of XP routers is that they may be introduced progressively, starting from most heavily loaded links. In such a scenario, the overall network efficiency will gradually improve, however, obviously, for the best performance, all nodes in a network should be FAN aware. The incremental replacement is possible, because each XP router operates independently and transparently to other standard IP routers. There is no need for a signaling protocol, end-user compliance or any inter-network agreements. Moreover, in [87], a belief that in FAN there is “virtually no requirement for standardization” with exception only for “agreed convention for defining the flow identifier”, is expressed. The lack of standardization is possible, because of the local nature of the XP router functionality. As long as nodes perform well and maintain their functions, the exact method of their operation is insignificant. Lastly, once developed and implemented, the proposed mechanisms are supposed to be particularly inexpensive.

3.5 Measurement based admission control

Admission control is a mechanism, which allows blocking of some portion of traffic, should congestion occurs. This ensures, that the quality of currently realized transmissions will not deteriorate below a certain threshold. The benefits of using admission control in the IP networks were presented in [6], [78] and [96].

In FAN, admission control is used to keep the maximum flow rates at a reasonable level, while ensuring negligible latency for low-rate flows.

In FAN, the admission control block implements the measurement based admission control (MBAC) functionality [54], and is designed to protect both streaming [65] and elastic flows [36]. Measurement-based means, that the admission decision relies solely on the measurements of the outgoing link congestion. Therefore, MBAC is local to a particular network link. Since no signaling is used in FAN networks, MBAC must be performed with the minimal knowledge about the ongoing traffic. All of the above renders FAN admission control implicit, as it does not rely on any explicit user-network signaling. In [7], it is shown that MBAC in FAN, is able to protect both streaming and elastic flows.

MBAC is not class-based or user-based: each new flow obtains the same treatment, and in case of congestion, all new flows are blocked. Such an approach may be considered as “unfair” service differentiation since in congestion, some flows are admitted and some are blocked. However, MBAC treats all the flows equally, i.e., a) the decision of accepting or rejecting the traffic affects all new incoming flows, not just some of them, b) admission decisions are implicit, based only on internal measurements.

MBAC performs actions based on the information derived from the scheduling algorithms. Two parameters are constantly measured, i.e., fair rate (FR) and priority load (PL). Following [67], “fair rate is an estimation of the rate currently realized by backlogged flows”, and represents the amount of link's bandwidth, which is guaranteed to be available for a single flow, should it be necessary. Similarly, “priority load is the sum of the lengths of priority packets transmitted in a certain interval divided by the duration of that interval”, and shows the amount of data that is prioritized. The manner of calculating both indicators is a feature of the scheduling algorithm, and is presented in Sections 3.6.1 and 3.6.2 where fair queuing algorithms, suited for FAN, are described.

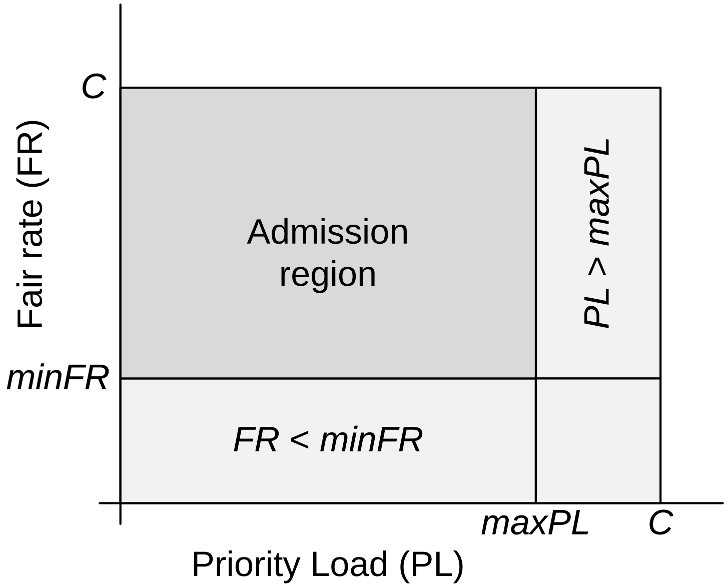


Figure 3.3: Admission region in FAN

Figure 3.3 illustrates the admission decision in MBAC. Each router has two pre-defined threshold values of FR and PL which are to be maintained, namely: the minimum value of FR (minFR) and the maximum value of PL (maxPL). If the current FR is lower then minFR or if the current PL is greater than maxPL, the incoming flow is blocked. Otherwise, when in the admission region, the new flow is admitted.

The user-defined flows discussed in Section 3.3 appear as the most appropriate entity, on which the admission control should be performed. Admitted flows and those currently in progress are registered in PFL. If the flow identity of a newly arriving packet is already on the PFL, the packet is forwarded. If not, the flow is subject to admission control. If the outgoing link is congested, the packet is simply discarded. In the absence of congestion, the packet is forwarded, and its flow may be added to the PFL. This decision, on whether to include the flow on the PFL or not, is probabilistic. The flow is added with the probability p, e.g. $p = \frac{1}{10}$. This procedure aims in decreasing the size of the PFL, as with high probabilities, very short flows (a few packets) will not be added to the PFL. Flows heaving tens of packets and more, will be added to the PFL eventually.

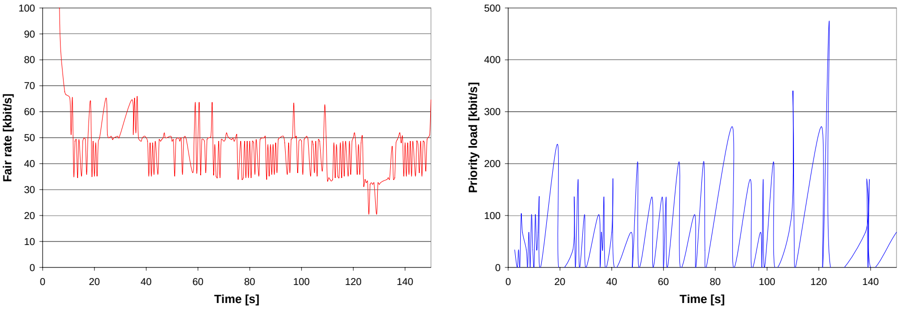


Figure 3.4: FR and PL measurements; no exponential smoothing

In the simplest example, the admission criteria could rely only on PL and FR measurements. For instance, FR and PL thresholds could be set to 0.1 and 0.7, respectively. It means, that packets of flows not registered in the PFL, are discarded, if the current FR is lower than 10% of the link capacity or the amount of priority traffic exceeds 70% of the link capacity. However, the MBAC algorithm may be based on some additional factors. In [65], such an algorithm is proposed. The algorithm takes also into account the measured aggregate load, and it should be less than a predefined threshold, for new flows to be admitted. The ns-2 [82] simulations have shown that a high link utilization can be achieved while maintaining a low packet delay and loss rate.

Figure 3.4 shows the examples of FR and PL measurements in FAN-based routers, over time, using the Priority Fair Queuing scheduling algorithm. The experiment is performed over 1 Mbit/s link, with the offered load exceeding the link capacity almost twice. Strong variations in measured values can easily be observed. They appear due to the extremely dynamic nature of packet-based transmission. In order to make the measurements more reliable, the notion of exponential smoothing was proposed in [64] and [65]. The smoothing formula is presented in Equation 3.1, in which α represents the smoothing parameter.

$$\text{new value} = \alpha \times \text{old value} + (1 - \alpha) \times \text{new measurement} \tag{3.1}$$

Figures 3.5 and 3.6 show the same measurements as in Figure 3.4, only with use of the smoothing parameter of 0.5 and 0.9, respectively. The measurements seem to be stabilized and more reliable, as the amplitude differences between consecutive measurements are significantly decreased. However, there is also a drawback of using the smoothing. The smoothed system reacts much more slowly to changes. It is extremely important when currently measured values of the fair rate drop below the threshold and the system should start to block incoming new

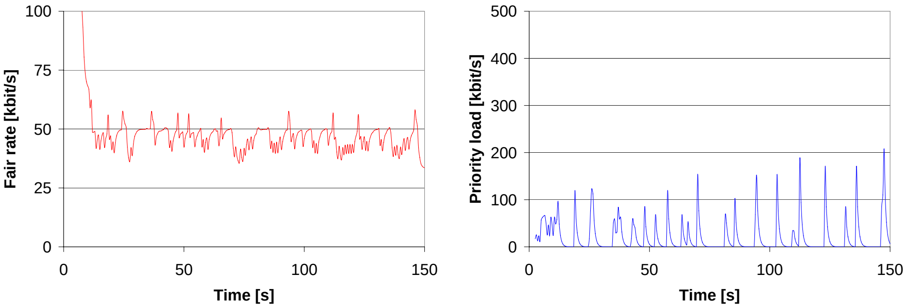


Figure 3.5: FR and PL measurements; exponential smoothing $\alpha = 0.5$

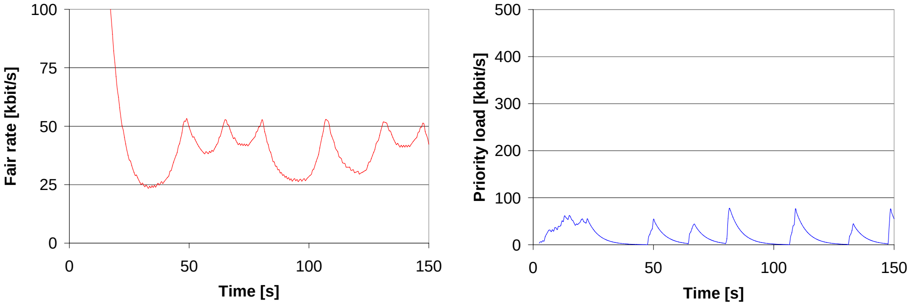


Figure 3.6: FR and PL measurements; exponential smoothing $\alpha = 0.9$

connections. When the smoothing parameter is set to a high value, the system is not able to respond for quite a time, which leads to fair rate degradations. The issue of slow response to the network condition is shown in Section 6.1 of this dissertation along with means to solve the problem.

3.6 Fair queuing with priority

The queue management in FAN is realized in the scheduling block of an XP router. Fair queuing ensures that link bandwidth is shared equally, without relying on the cooperative behavior of end users. This is a different approach than in currently used IP routers, where, usually, the FIFO queue is implemented. The FIFO queuing discipline does not ensure fair sharing of the link bandwidth. Instead, all flows are limited to the same percentage of their nominal rates. Figure

3.7 shows the behavior of FIFO and FQ queues, when two flows struggle for link resources.

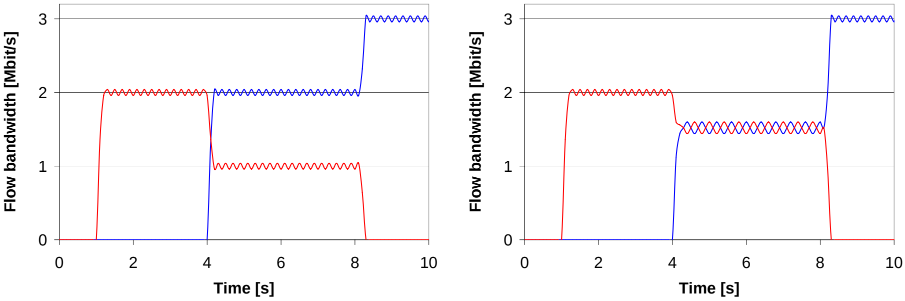


Figure 3.7: FIFO (left) and FQ (right) scheduling comparison

In this scenario, the bottleneck link capacity is 3 Mbit/s, the red and blue UDP flows [3] have nominal rates of 2 Mbit/s and 4 Mbit/s respectively. Between 4th and 8th second of the simulation time, both flows are in progress. The FIFO queue limits the rates of both flows to 50% of their desired rates, allowing the faster flow to utilize twice more bandwidth than its competitor. The FQ discipline, which is represented in this example by Priority Fair Queuing , limits both flows to the same rate: 1.5 Mbit/s, which is exactly half of the total link capacity. It should be noted that the order in which flows appear on the link is irrelevant to the outcome of fair queueing.

There are two per-flow fair queuing algorithms proposed for FAN: Priority Fair Queuing (PFQ) and Priority Deficit Round Robin (PDRR). Both algorithms have, logically, one priority queue and a secondary queuing system. They are intended to realize fair sharing of link bandwidth to elastic flows and priority service to streaming flows. The latter (PDRR) was primarily suggested to speed up commercial adoption since it improves the algorithm complexity from O(log(N)) to O(1); where N is the number of currently active flows. However, it has been shown that both scheduling algorithms achieve a similar performance [63].

Regardless of the algorithm used, it has been proved in [62] and [63] that fair queuing in FAN is scalable since the complexity does not increase with the link capacity. Moreover, fair queuing is feasible, as long as link loads are not allowed to attain saturation levels, which is asserted by admission control. Most recently, another FAN architecture has been proposed in [28], based on the Approximate Fair Dropping [93] queuing algorithm. The new architecture is referred to as

[3] UDP flows were chosen, as they are shaped only by the queuing algorithms, and not by the protocol behavior. This allows for presenting the features of solely the queuing disciplines.

Approximate Flow-Aware Networking or AFAN for short, and aims at further simplifying the queuing processes. As shown in [28], the enqueue and dequeue operations in AFAN are realized in a simpler way than in the previous proposals of PFQ and PDRR.

In the forthcoming sections, two originally proposed queuing disciplines for FAN are presented. Sections 3.6.1 and 3.6.2 describe PFQ and PDRR, respectively. To be suited for FAN, both queuing disciplines are modifications of well known algorithms.

3.6.1 Priority Fair Queuing

A large number of queuing algorithms have been proposed in the literature. The Start-time Fair Queuing (SFQ) [41] is particularly well suited for the FAN architecture. However, in [67], an enhancement of the SFQ algorithm is proposed. This modified queuing discipline is referred to as Priority Fair Queuing. PFQ differs from SFQ by the fact, that it gives head of line priority to packets of flows, whose rate is lower than the current fair rate. Therefore, PFQ implicitly prioritize packets of low-rate flows, which are streaming flows in FAN.

```
if PIFO congested, reject packet at head of longest backlog
if $F \in flow\_list$
begin
    $backlog(F) += L$
    if $bytes \geq MTU$
        push $\{packet,\ flow\_time\_stamp\}$ to PIFO
    else begin
        push $\{packet,\ virtual\_time\}$ to PIFO behind $P$; update $P$
        (counter of priority bytes $+= L$)
        $bytes(F) += L$
    end
    $flow\_time\_stamp(F) += L$
end
else begin
    push $\{packet,\ virtual\_time\}$ to PIFO behind $P$; update $P$
    (counter of priority $bytes += L$)
    if $flow\_list$ is not saturated
    begin
        add flow $F$
        $flow\_time\_stamp(F) = virtual\_time + L$
        $backlog(F) = L$
        $bytes(F) = L$
    end
end
```

Figure 3.8: PFQ packet arrival operations [67]

The PFQ algorithm is based on the Push-In, First-Out (PIFO) queue. PIFO is the shorthand for the sorting algorithm that allows a packet to join the queue at any position, and serves always the packet at the head of the line. A position that a packet is inserted is determined by a time stamp, according to which packets are sorted within the queue. Therefore, every element in the PIFO queue has the form of *packet, time stamp*, where *packet* represents the data relating to the packet (e.g., a memory location pointer), and *time stamp* is a packet "start tag" determined by the PFQ algorithm.

Figure 3.8 shows the pseudocode that is executed on each packet arrival, as proposed in [67]. First, it is necessary to test, whether the queue is congested, and if so, which packet should be dropped. Dropping the packet at the head of the longest backlog is a proposition, however different criteria are possible. If a flow is active (line 2), its *backlog* is increased by the size of the packet (L). The packet is given a priority when the cumulative volume of transmitted bytes is lower than the maximum transfer unit (MTU) (lines 7–11). Then, the packet is enqueued with the *time stamp* of *virtual time*, which is essentially the head of the queue. When the transmitted bytes count is greater than MTU, the packet is placed in a PIFO queue, according to its nominal place. Lines 14–24 represent a situation in which the arriving flow is not active. Then, the packet is given a priority (line 15), and providing that the *flow list* is not saturated (line 17), the flow is added to the *flow list* (lines 18–23).

```
 1   if PIFO is now empty
 2       remove all flows from flow_list
 3   else begin
 4       $backlog(F) -= L$
 5       serve packet at head of line
 6       next_time_stamp designates time stamp of this packet
 7       if $next\_time\_stamp \neq virtual\_time$
 8       begin
 9           $virtual\_time = next\_time\_stamp$
10           for all flows $f \in flow\_list$
11           begin
12               if $flow\_time\_stamp(f) \leq virtual\_time$
13                   remove f from flow_list
14           end
15       end
16   end
```

Figure 3.9: PFQ packet departure operations [67]

Figure 3.9 shows the operations performed after each packet departure. If the PIFO queue is empty, obviously all flows must be removed from the *flow list* (lines 1–2). Otherwise, the next packet in the queue is prepared (lines 4–6):

the flow backlog is reduced and the *next time stamp* is set to this packet's *time stamp*. If *virtual time* is equal to the *next time stamp* (line 7) no further operations are required, as *virtual time* has not changed since the last packet departure. If it did change (lines 8–15), the *virtual time* is updated, and flows that become inactive (i.e., their *flow time stamp* is less than or equal to the new value of the *virtual time*), are removed from the *flow list*.

To provide the admission control block with a proper congestion status, *priority load* and *fair rate* indicators are measured periodically. An estimation of the priority load is derived from Equation 3.2. Variables $pb(t)$ represent the values of a counter, incremented on the arrival of each priority packet by its length in bytes, at time t. (t_1, t_2) is a measured time interval (in seconds), and C is the link bit rate. Priority load, therefore, represents the sum of the lengths of priority packets transmitted in a certain time interval, divided by the duration of that interval, and normalized with respect to the link capacity.

$$priority\_load = \frac{[pb(t_2) - pb(t_1)] \times 8}{C(t_2 - t_1)} \tag{3.2}$$

Equation 3.3 is used to calculate the fair rate, which is an estimation of the rate currently realized by backlogged flows. To estimate the fair rate, a fictitious flow emitting single byte packets is considered. In an idle period, the fictitious flow could transmit at the link rate. Otherwise, the number of bytes that could have been transmitted, is given directly by the evolution of virtual time. In Equation 3.3, $vt(t)$ is the value of virtual time at time t, (t_1, t_2) is the measurement interval, S is the total idle time during the interval, and C is the link bit rate.

$$fair\_rate = \frac{\max\{S \times C, [vt(t_2) - vt(t_1)] \times 8\}}{(t_2 - t_1)} \tag{3.3}$$

When the measured link is lightly loaded, the first term of the $\max\{\dots\}$ formula is significant, as the fictitious flow uses all residual link capacity. When the link is busy, the second term becomes important, as it approximately measures the throughput achieved by any flow that is continuously backlogged in this time interval.

As mentioned, both congestion indicators are calculated periodically. Considering the extremely dynamic variations of priority packets occurrence, the period between two consecutive measurements of the priority load is advised to be several milliseconds, whereas a several hundreds of milliseconds period is sufficient for estimating the fair rate. Regarding the frequent measurements of the congestion indicators in PFQ, the complexity may be an issue, especially in the core. However, all the mathematical calculations in Equations 3.2 and 3.3 are very simple, and not time consuming. Moreover, the complexity of enqueuing and dequeuing operations of PFQ, like SFQ, is logarithmic with the respect to the number of active flows, which is essentially limited by the admission control block.

3.6.2 Priority Deficit Round Robin

PFQ was the first queuing algorithms proposed to be suited for the FAN architecture. In fact, simulations have shown, that PFQ performs well, and cooperates with the admission control block correctly [67]. Furthermore, the scalability of PFQ has been demonstrated by means of trace driven simulations and analytical modeling in [62] and [63]. However, PFQ can be advantageously replaced by an adaptation of the Deficit Round Robin (DRR) [101] algorithm. An enhancement to DRR, called Priority Deficit Round Robin is presented in [66].

PDRR retains the low complexity of DRR, at the same time, providing low latency for streaming flows. PDRR complexity is constant (O(1)), therefore, it does not increase with the growing number of active flows (PFQ complexity was logarithmic with respect to the number of active flows). PDRR enhances the DRR algorithm, in that it introduces the priority queue, which is used for low rate flows. Figure 3.10 shows the operations performed by PDRR on each packet arrival.

```
1    on arrival of packet $P$
2    if no free buffers left then
3        FreeBuffer()
4    $i$ = ExtractFlow($P$)
5    if ($i \notin AFL$)
6    begin
7        add $i$ to $AFL$
8        $DC\_i = 0$
9        $ByteCount\_i$ = Size($P$)
10       Enqueue($PQ$, $P$)
11   end
12   else begin
13       $ByteCount\_i$ + = Size($P$)
14       if ($ByteCount\_i \leq Q\_i$) then
15           Enqueue($PQ$, $P$)
16       else
17           Enqueue($Queue\_i$, $P$)
18   end
```

Figure 3.10: PDRR packet arrival operations [66]

Initially, if the buffer for incoming packets is full, a certain packet must be selected for dropping (lines 1–3). PDRR does not specify which dropping mechanism should be used. One policy would be to drop packets at the head of the flow with the longest backlog, however, this approach is not mandatory. If packet P does not belong to an active flow, a new flow is added to AFL, the Deficit Count (DC) and Byte Count counters are properly initiated, and the packet is forwarded to the priority queue (PQ) (lines 5–11).

If an arriving packet belongs to a flow currently on the flow list, it may be placed at the end of his flow queue (line 17) or in the priority queue, providing that $ByteCount\_i \leq Q\_i$ (lines 14–15). The variable $ByteCount\_i$ holds the number of bytes inserted in the priority queue for flow i, while Q_i represents flow quantum: the cumulative number of bytes allowed for transmission after every cycle of the algorithm. Although DRR allows for resource allocation differentiation, by means of assigning different quanta Q_i for different flows, the FAN fairness concept implies that the same quanta should be used for each flow.

```
1   while TRUE do
2   begin
3       while $PQ$ not empty do
4       begin
5           $P$ = Dequeue($PQ$)
6           $i$ = ExtractFlow($P$)
7           Send($P$)
8           $DC\_i -= $ Size($P$)
9       end
10      if $AFL$ is not empty then
11      begin
12          get head of $AFL$, say flow $i$
13          $DC\_i += Q\_i$
14          while ($DC\_i \geq 0$) and ($Queue\_i$ not empty) do
15          begin
16              $PacketSize$ = Size(Head($Queue\_i$))
17              if ($PacketSize \leq DC\_i$) then
18              begin
19                  Send(Dequeue($Queue\_i$))
20                  $DC\_i -= PacketSize$
21              end
22              else
23                  break; (*skip while loop*)
24          end
25          RemoveActiveList($i$)
26          if $Queue\_i$ is not empty then
27              add $i$ to $AFL$
28      end
29  end
```

Figure 3.11: PDRR packet departure operations [66]

Figure 3.11 shows the dequeue operations in the PDRR algorithm. The priority queue is served, whenever it is not empty (lines 3–9). When a packet is sent through the priority queue, the deficit counter of its flow is decreased by the size of the packet. This operation prevents serving more than one quantum in a single round. When there are no packets in the priority queue, and AFL contains some flows, the flow at the current head of the AFL cycle is selected for service (line 12). The deficit counter of this flow is incremented by one quantum (line 13), and packets at the head of this flow's queue are prepared for being serviced (lines 14–24). The flow may emit up to *DC_i* bytes. At the end of the cycle, the AFL is rebuilt, i.e., completely erased (line 25) and re-created (lines 26–27).

The congestion indicators are measured differently than in PFQ. To measure the fair rate, we count the number of bytes that a fictitious and permanently backlogged flow could emit in a certain time interval and divide that value by the duration of the interval. This procedure is extremely easy to implement. The algorithm maintains one fictitious flow and treats it as a normal transmission, except, it does not transmit any packets. Therefore, the value of the deficit counter, which is regularly increased by the quantum, represents the theoretical amount of data that could be emitted by that flow. Since this flow is permanently backlogged, it does not disappear from the AFL, and thus its *DC_i* value is sustained. The priority load measurements are even simpler and are performed by averaging the emitted bit rate from a priority queue over a suitable time interval.

The introduction of a priority queue in PDRR results in situations in which flows with empty queues, i.e. flows whose packets are forwarded only via the priority queue, may exist. This implies that the dequeue procedure complexity is not strictly O(1). However, according to [66], this can be corrected by modifying AFL, as a list for non-empty queues only. This list would be updated, whenever a new flow receives more that its quantum in the initial round.

The enqueuing module complexity depends on the speed of detecting the presence of a flow in the AFL (line 5 on Figure 3.10). In order to maintain the O(1) complexity, the Content-Addressable Memory (CAM) must be used. CAM is a special kind of memory, designed to search its entire contents in a single operation, but its hardware implementation issues require that the size of AFL be small enough. However, the ns-2 based simulations have shown [66] that the required size of AFL is relatively small and, most importantly, does not increase with the link speed.

3.6.3 PFQ and PDRR comparison

PDRR and PFQ are similar queuing algorithms. Although their operation is quite different, they realize the same objectives. The only advantage of PDRR over PFQ is simplicity. As mentioned before, the complexity of queuing disciplines in

PFQ is logarithmic with respect to the number of active flows, whereas the same complexity is constant, and does not depend on the AFL size in PDRR.

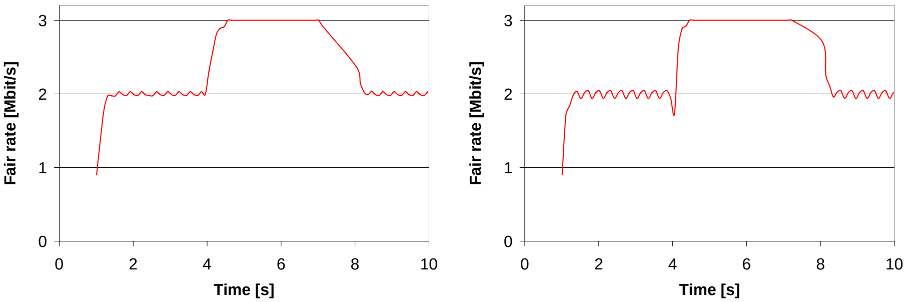


Figure 3.12: Fair rate measurements; PFQ (on the left) and PDRR (on the right)

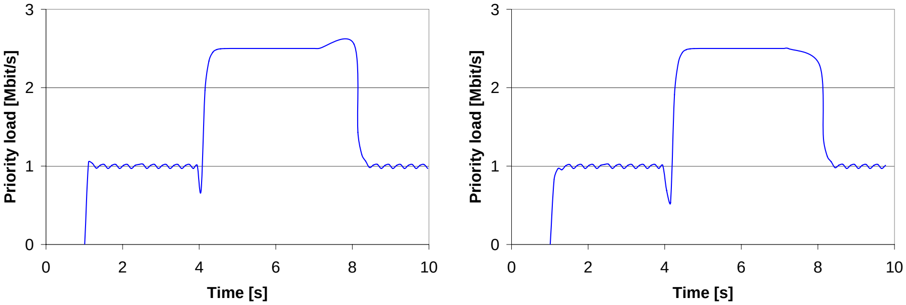


Figure 3.13: Priority load measurements; PFQ (on the left) and PDRR (on the right)

Figures 3.12 and 3.13 show the comparison between measured congestion indicators by PFQ and PDRR algorithms, namely: fair rate and priority load, respectively. The ns-2 simulation scenario was identical for both scheduling disciplines. Two UDP flows of 1 Mbit/s and 2.5 Mbit/s nominal bit rates struggled to utilize the 3 Mbit/s bottleneck link. Between 4th and 7th second, only the latter flow was active.

According to the fair rate definition (see Section 3.5), its value should be 100% of the link capacity when only one flow is active, as that flow could emit at the link maximum bitrate, should it be necessary. When both flows are in progress, 2 Mbit/s FR value is also expected. 1 Mbit/s flow is not backlogged, as it emits through the priority queue constantly (its rate is always below FR). Therefore, 2

Mbit/s is left for backlogged flows, but since only one more flow is in progress, 2 Mbit/s is the value of FR.

Priority load is the amount of data that is prioritized by the schedulers. When both flows are in progress, as mentioned previously, FR is equal to 2 Mbit/s. Therefore, 1 Mbit/s flow is below FR, and its packets constantly go through the priority queue, whereas 2.5 Mbit/s flow is above FR and always utilizes normal queuing. However, when only 2.5 Mbit/s flow is in service, FR is equal to 100% of link capacity and the flow's packets use the priority queue, as the flow's rate is below the current FR. Therefore, measured PL values are 1 Mbit/s in the former case, and 2.5 Mbit/s in the latter.

As expected, no major differences between the measured values by both scheduling disciplines exist. This confirms the previous thesis, that the difference between PFQ and PDRR lies only in the complexity issues. Although minor differences can be observed, they are insignificant to the overall behavior of admission control, for which these indicators are used.

3.7 Additional FAN architectures and mechanisms

Up to this section, the original concept of FAN was presented. The architecture attracted some worldwide attention which resulted in many more studies. Over the years, numerous additional mechanisms were proposed for FAN. They were the answer to certain technical problems with the implementation or performance. New mechanisms evaluate the possibilities of using FAN in certain scenarios or just improve its functioning.

When the link is congested, XP routers do not allow any network connections which increases the time needed for new flows to begin their transmission. There are two approaches to solve the problem. One is to use the scheme of Static Router Configuration to help with the transmission of the Emergency Calls. This method was presented in [53] and is described in details in Section 5.5. The second approach is based on periodic partial or total clearing of the PFL list of an XP router. Various modifications of this method were presented in [25], [26], [27], [31] and then gathered in the PhD thesis of J. Domżał [24]. Additionally, [31] presents the most mature flushing mechanism and evaluates its robustness and reliability.

The notion of Multilayer Flow-Aware Networking (MFAN) was introduced in [76], and later presented as a complete approach in the PhD dissertation of V. López in 2010 [75]. In those works, it is shown that FAN can be extended by including an optical layer to be considered by the system. The idea is that a FAN router can request additional optical resources once the standard IP link is congested. Under normal circumstances, upon the congestion of the outgoing link, FAN starts to block new incoming connections. In MFAN, the router is able to utilize additional resources and redirect flows to that resource, creating space for new flows to be admitted. There are three admission control policies deciding which flows to be redirected to the optical link, i.e., Newest Flow Policy, Oldest Flow Policy and Most-Active Flow Policy. In [76], those policies are compared and their performance is evaluated.

In [32], the authors compare admission control policies proposed for MFAN. As a result of the comparison, a new admission control strategy is proposed. The solution inherits the advantages from already established admission control proposals while ensuring fast acceptance times of new streaming flows. It is also possible to combine the advantages on MFAN with those of flushing mechanisms. That work is continued under the BONE EU project, where the authors show the differences between admission control strategies proposed for IP-level FAN and MFAN.

In [33], a multi-layer recovery strategy for the MFAN nodes is presented. The authors propose using the Enhanced Hold-Off Timer (EHOT) algorithm [23], known from RPR networks, to control network operation after link or node failure. Network performance after failures is also presented in [29] where the authors measure the impact of proposed congestion control mechanisms in case of network overload. The results show that the acceptance times of streaming flows are acceptable even with the presence of network failures, provided that proper congestion control mechanisms are used. Both papers essentially show that FAN networks have great resilience capabilities.

Originally, FAN was developed to work with the PFQ queuing algorithm. Later, it was proposed to exchange PFQ with PDRR, just to decrease the complexity. However, in [28] a new architecture is proposed. Approximate Flow-Aware Networking (AFAN) is a new method to realize the queuing procedures based on Approximate Fair Dropping algorithm [93]. The AFAN is simple and ensures implicit service differentiation, fairness for elastic and high priority for streaming flows. The comparison of AFAN with two other FAN architectures (with PFQ and PDRR scheduling algorithms) shows that the enqueue and dequeue operations are realized in a simpler way.

FAN was also extensively tested in the Grid environment. In [15], the authors show the impact of DiffServ and FAN on the grid traffic, and compare the efficiency of those architectures in providing QoS assurances. Further, in [16], [14] and [18], the performance of FAN in the Grid environment is evaluated. It is shown that FAN outperforms DiffServ in the average GridFTP session delay and the average GridFTP session goodput under increasing offered load.

FAN does not interfere with the IP protocol functionality, including the routing procedures. However, it is possible to introduce a new routing scheme, one which would cooperate with FAN. In [88], such a scheme is proposed. The authors conclude that adaptive routing clearly improves network performance especially in overload and failure conditions. FAN, based on fair queueing and admission control mechanisms, is considered as a pragmatic implementation of an optimal fluid model which provides ideal performance. In 2007, a FAN simulator was developed [34] working at the flow-level basis, as opposed to ns-2 [82] which works on the packet-level basis. At the cost of certain constraints, flow-based simulator runs much faster than ns-2, however, it is much less accurate. A new simulator was tested and the results suggest that this tool is sufficient for evaluation of routing algorithms, to which ns-2 was simply too slow.

In [59], a traffic control algorithm which performs traffic control on flow level was proposed for FAN. Using the proposed FAN node model, the simulation analysis proved that FAN can be a new approach for realizing QoS guarantees in the IP networks.

3.8 Net neutrality with respect to Flow-Aware Networking

The idea of net neutrality is that a user traffic is not discriminated at all in relation to a traffic generated by other network users. In the Internet, it is possible to guarantee different QoS based on, e.g., source or destination addresses or network device port. Internet Service Providers may use this possibility to prioritize some network applications, therefore, assuring better QoS to selected traffic. Common QoS architectures provide means for the network operators to differentiate the service without any limitations. However, since most of the differentiation actions are against the net neutrality, choosing such a powerful and complex solution is neither useful, nor cheap.

In [30], it is shown that FAN is a QoS architecture which perfectly fits into the net neutrality boundaries while still providing QoS awareness. The main advantage of FAN, with respect to the net neutrality issue, is that it provides service differentiation, taking into account only the traffic characteristics of the ongoing transmissions. Therefore, it is not possible to discriminate certain applications or end-users. Moreover, instead of providing differentiated treatment, FAN introduces fairness, which even enhances the current IP networks equality.

FAN, as opposed to IntServ and DiffServ, does not allow to provide an explicit differentiation by the ISPs. It is a very important advantage of this technique. Of course, ISPs may try to change router's software and provide a traffic classification which allows for packet queuing and servicing according to their rules. However, such behavior is opposed to FAN principles and, as so, it is more difficult to introduce than in, e.g., DiffServ.

The significance of the net neutrality problem forces the researchers to propose and develop new solutions for QoS guarantees. The Flow-Aware Networking is a solution that meets net neutrality assumptions and allows for implicit service differentiation. Using this architecture, ISPs will not have to implement any traffic policies or explicit QoS mechanisms to guarantee proper traffic performance. Moreover, they will not be able to do it, and in consequence to charge extra money from Internet users. FAN, originally simple, is a viable proposal for the future Internet. It perfectly fits into both the followers and opponents of the net neutrality concept. The simulation results presented in this dissertation confirm the usefulness of FAN in this context. I am convinced that the solutions proposed in this thesis will contribute to solving the net neutrality problem with satisfaction to any side.

Part II

Quality of Service in IP networks

4 Flow-oriented approaches to QoS assurance

Adding bandwidth, processing power, and routes to stupid networks helps, but only to a point.

— Lawrence Roberts

The first significant attempt to introduce QoS to networks based on the IP protocol took place in June 1994 when the IETF group published RFC 1633 [11], thereby introducing Integrated Services (IntServ). This was the first complete approach to differentiate traffic between end users. Unfortunately, very soon this solution was announced to have severe problems with scalability and complexity [44]. Since then, the development of QoS architectures has gone in two ways. One of them is to retain the flow-based approach of IntServ while improving the scalability. The other, pursued mainly by IETF itself, focuses on decreasing the granularity of flows (therefore, reducing the amount of the required flow state information in routers) and dealing with aggregates rather than single instances. The IETF's second approach is known as Differentiated Services (DiffServ) [8].

This chapter presents the first mentioned line of QoS differentiation architectures, which are flow-aware. All the presented approaches try to provide the differentiation of service for every flow individually. The definition of a 'flow' may be slightly different in each of them, however, they still see a 'flow' as a single 'connection' between end users. A flow should be seen as a whole, neither as a set of packets that need a preferential treatment, nor as a group of connections classified into one aggregate. The extended version of this chapter is published as [113].

There are numerous studies on QoS assurance techniques based on individual flows. The fact that flow-awareness can be found in technical papers, patents, recommendations, standards, and commercial products proves its viability and importance. In this chapter, I present and compare all significant contributions to flow-aware QoS guarantees, assessed either by the originality of the approach or by the common recognition. Thereby, I discuss such architectures as: Integrated Services, Connectionless approach to QoS guarantees, Dynamic Packet State, Caspian Networks and Anagran, the Feedback and Distribution method, Flow-Based Differentiated Services, the Flow-State-Aware transport and Flow-Aggregate-Based Services. All those architectures are compared with each other and with Flow-Aware Networking which was presented in Chapter 3.

The descriptions of the technical side of all the solutions are widely accessible, yet the current literature regarding the comparison of these approaches is scarce, to say the least. This survey, therefore, attempts to compare and contrast the most promising or relevant solutions proposed up-to-date. Section 4.1 introduces the reader to flow-aware architectures, presenting their common goal, main similarities, the development time frame and a short description of each. Then, instead of presenting every architecture one by one, Sections 4.3 through 4.6 deal with certain aspects of every proposal. Namely, I describe how the flows are defined in each architecture, what the classes of service are, how the admission control and scheduling are realized and how the signaling problem is resolved. Section 4.7 summarizes all presented approaches, identifies their pros and cons, and shows my opinion and forecast for the future QoS architectures in the Internet.

4.1 Background and development history

The necessity to introduce quality of service to the Internet was not noticed in the beginnings of the IP networks. Initially, the network was used only for simple file transfers for which the IP protocol was perfectly sufficient. The popularization of the Internet, growing offered capacities and emergence of multimedia applications rendered the existing IP protocol unfit. Over time, numerous QoS architectures have been proposed. Figure 4.1 presents the timeline for the flow-aware QoS architectures that are compared here. The time frames show the development of the particular architectures. Wherever there are ambiguities in determining the exact dates, the time bars are faded into background while the relevant explanation is in the text.

The QoS issue was firstly addressed by Internet Engineering Task Force (IETF) in 1994 when the Integrated Services (or IntServ for short) model was

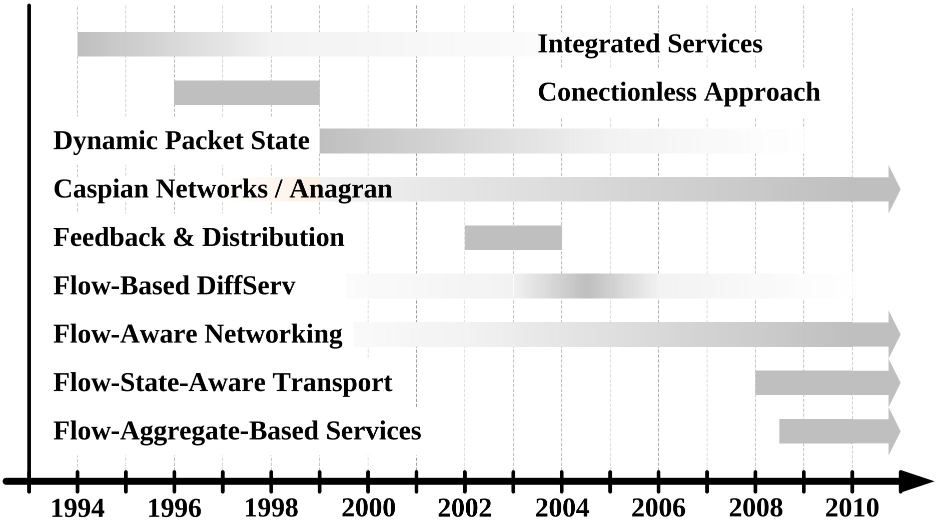


Figure 4.1: QoS Architectures: development history

introduced in RFC 1633 [11]. Almost immediately the problem with IntServ's scalability was widely recognized. Nevertheless, numerous papers have appeared ever since, either mitigating some of the disadvantages or providing new functionalities. Especially, after the advent of Differentiated Services (DiffServ) in 1998 [8], several approaches to combine both solutions have been proposed.

The Connectionless Approach to providing QoS in the Internet was presented in 1998 in [80], although it was based on the automatic QoS method from 1997 [17]. The Stateless Core architecture (SCORE) [104] provided the foundation for Dynamic Packet State [105]. All the results were, then, discussed in the PhD dissertation of I. Stoica in 2000 [103].

The beginnings of Caspian Networks, Inc. go back to 1998. In 2000, a patent on micro-flow management was filed to United States Patent and Trademark Office. Currently, Anagran, Inc. continues the main line of Caspian Networks providing a real commercial product: a flow-aware network router. In [68], a flow-based QoS architecture for large-scale networks, named the Feedback and Distribution method was presented. To the author's knowledge, the proposal was not pursuit any further, yet it demonstrates an interesting and original approach. The Flow-based DiffServ architecture is an enhancement to plain DiffServ which introduces flow-awareness. As it is based on DiffServ, the natural origins of this method go back to 1998, still the method was presented in 2004 [73]. Since

DiffServ is a very popular approach, it is difficult to specify clear boundaries as the flow-aware line of DiffServ may be pursued in future.

Flow-Aware Networking was initially introduced in 2000 [95], and presented as a complete system in 2004 [67]. Since then, many papers have appeared regarding new mechanisms for the architecture which improve the performance of the solution. Flow-State-Aware Transport is a proposition for the NGN networks and appeared as an ITU-T recommendation in 2008 [50]. In the same year, the Flow-Aggregate-Based Services architecture that is based on Flow-State-Aware Transport was proposed [58]. The last three architectures, i.e., FAN, Flow-State-Aware Transport, and Flow-Aggregate-Based Services are currently under further development and specifications.

4.2 Flow-based architectures at a glance

Sections 4.3 to 4.6 discuss the main aspects of the QoS architectures, such as: flow definition, provided classes of service, admission control mechanisms, queue management and signaling. In this section all the discussed architectures are briefly presented.

Integrated Services IntServ enumerates services which are vulnerable to end-to-end packet delay, namely: remote video, multimedia conferencing, visualization and virtual reality, and presents a method to care for them. The term Integrated Services represents the Internet service model, which includes best effort traffic, real-time traffic and controlled link sharing.

The Integrated Services model's design process was based on certain key assumptions. Since it was the pioneering endeavor, these assumptions were mostly theoretical and not based on previous experiences or simulations. They were expressed as follows:

1. a reservation protocol is necessary,
2. simply applying a higher priority to real-time traffic is not sufficient,
3. the Internet should be a common infrastructure for both real-time and elastic traffic,
4. the TCP/IP protocol stack should still be used (without major changes) in the new architecture,
5. the Internet Protocol robustness should be preserved,
6. there should be a single service model for the whole Internet.

To allow reservations, the authors designed a new protocol: Resource Reservation Protocol (RSVP). This protocol is used in IntServ for setting-up and tearing-down connections. Each QoS-enabled connection must be pre-established and maintained by the RSVP protocol.

Connectionless Approach Although the complexity and scalability issues of IntServ were quickly recognized, the foundation for QoS architectures had been established and this allowed many researchers to follow the trails. Some tried to go in a different direction and introduce class-based differentiation, like IETF in Differentiated Services. The team from Computing Technology Lab in Nortel, however, decided not to abandon the flow-awareness and to propose their idea of service differentiation in the IP networks, based on the already established IntServ [80]. Their intention was to address the scalability issues by removing the need for the connection-oriented reservation protocol. Instead, they propose an automatic detection of the QoS requirements by network devices.

As mentioned, the proposed approach is based on IntServ but does not use any signaling protocol. The architecture of this proposal consists of the Traffic Conditioner and a connectionless mechanism for ensuring consistent end-to-end delivery of QoS based on application requirements. The traffic conditioner is based on the reference implementation framework of IntServ [11]. It contains three elements necessary to manage bandwidth on a router, namely: the classifier, admission controller and scheduler. Instead of RSVP, the Automatic QoS mechanism [17] is used to discover the quality requirements on-the-fly and service them accordingly. The detection is based on measuring the traffic pattern of the incoming flow: the transport protocol, the size of the packets and their interarrival times. Unfortunately, the mentioned connectionless mechanism for ensuring consistent end-to-end delivery is not addressed. The authors express, however, their opinion about the need for such a mechanism, and point at the possibility of using the DiffServ marking scheme for that purpose.

Dynamic Packet State Another approach to eliminate the scalability problem of IntServ (and per-flow mechanisms in general) is Dynamic Packet State (DPS). DPS is a technique that does not require per-flow management at core routers, but can implement service differentiation with levels of flexibility, utilization and assurance similar to those that can be provided with per-flow-mechanisms.

DPS was introduced in 1999 [105] by I. Stoica and H. Zhang. In 2000, the former presented his Ph.D. dissertation on the Stateless Core (SCORE[4]) approach [103], which received ACM Best Dissertation Award in 2001.

In DPS (and SCORE for that matter), only edge routers perform per-flow management while core routers do not. The information that is required for flow-based guarantees is carried in the packet headers. Edge nodes inject the

[4] SCORE is sometimes also referred to as ‘Scalable Core’

information and each router along the path updates it accordingly. There are, therefore, per-packet regulations in every core node, however, flow-awareness is maintained due to ingress nodes' proper packet header inclusions. Although core nodes do not see real flows, the information carried in the packet headers enables the packets to be served in a way which provides end-to-end flow-based guarantees. Finally, DPS provides QoS scheduling and admission control without per-flow states in core routers.

Caspian Networks / Anagran One of the fathers of the Internet, Larry Roberts, previously of Caspian Networks, currently in Anagran, pursues the notion of intelligent networks. In [97], he presented his opinion about the today's Internet drawbacks, which can be solved by injecting intelligence into the networks. Having realized those issues, L. Roberts and Anagran proposed an optimized flow-based approach to IP traffic management.

Anagran went a bit further, as along with the proposal of the novel QoS architecture, they also built a device which puts their ideas into practice. This device, Anagran FR-1000, is a layer 3 interconnecting device or simply: an enhanced IP router. Unfortunately, all the knowledge about this device and the technology it implements can be obtained only from the company documents, which, naturally, are marketing oriented. Nevertheless, the idea must be solid and mature enough to have been implemented.

The Anagran approach to QoS in IP networks is based on flow-awareness. FR-1000 uses the *Fast Flow Technology* [3] to maintain constant state information for every active flow. By using flows rather than single packets as the entity on which the traffic management is performed, the insight into traffic dynamics and behavior over time can be gained. The Anagran's product automatically sorts a diverse traffic mix into dynamic 'virtual channels'. This allows for the coexistence of various traffic types in the same pipe. The only concern could be the potential difficulty to constantly maintain the per-flow state for all active connections, as this solution is not considered scalable. However, Anagran defends this concept by stating that "rapid decline in memory cost over the past decade has actually made keeping flow state virtually insignificant from a cost standpoint".

Feedback and Distribution The problem of scalability has been haunting IntServ since its beginnings. Therefore, the team from NTT Access Service Systems Laboratories in Japan proposed an architecture that would be suitable for large-scale networks. Their approach is referred to as *Feedback and Distribution Method* and is presented in [68]. This method provides per-flow QoS differentiation for large-scale networks. The operability is based on measuring traffic in the access system, where traffic is divided for each user, and these measurements are fed into the network.

The proposed method is very simple and efficient. The whole idea is to keep inner-network devices as simple as possible, while performing all the required operations at the edges. Although being similar to DiffServ, Feedback and Distribution approach retains flow-awareness.

Profile meters are located at network boundaries, as close to the end user as possible, for example, at the termination point of every access line. Markers are also put at the network boundaries but on the side of the servers. The role of a profile meter is to constantly measure the individual user traffic and send that data to the markers. A marker is responsible for setting the priority for packets, according to the data obtained from the profile meter. The only role of a network router, playing also the role of a dropper, is to forward packets according to their priority: packets are dropped in the priority order when congestion occurs. There are only two possible priority indicators, i.e., high and low. The high priority is assigned to packets whose traffic rate is lower than the guaranteed rate, and the low priority for the rest. Packets with low priority are more likely to be dropped in the network in case of congestion. Dropping packets effectively reduces the rate at which the flow transmits, and this reduction process may continue until the measured rate becomes lower than the guaranteed bandwidth, in which case the flow is prioritized again. This mechanism, therefore, under severe congestion, shapes each transmission to the guaranteed rate.

Flow-Based Differentiated Services As this survey focuses on flow-aware approaches, the Differentiated Services (DiffServ) model [8] is outside of its scope as it does not support flow-based differentiation. Still, it is possible to enhance the architecture so that it could provide flow-based proportional differentiated services in the class-based environment. This novel scheme with an estimator for the number of active flows, a dynamic Weighted Fair Queuing (WFQ) [21] scheduler and a queue management mechanism was proposed by J.-S. Li and C.-S. Mao in [73].

The authors present their approach as a result of the observation that DiffServ does not guarantee that flows classified with a higher priority will really observe a better QoS than lower priority ones, due to the fact that the distribution of active flows in individual classes might be different. In other words, even larger bandwidth allocated to the higher class with a greater number of active flows may not provide better QoS than a lower class with fewer active flows. Therefore, this model proposes a flow-based proportional QoS scheme which always provides a better quality for flows with a higher class. In general, supposing that a network provides N classes, the following equation should always be true: $q_i/q_j = \delta_i/\delta_j$, where q_i is the service level obtained by class i and δ_i is the fixed differentiation parameter of class i, $i = 1, 2, ..., N$. The actual QoS for an individual flow will depend on the number of currently active flows in its class, however, the quality ratio between classes should remain constant. Therefore, the purpose of this model is, in fact, to ensure the constant proportion between perceived QoS by flows in different classes, regardless of the current class loads.

Flow-State-Aware Transport As the concept of the Next Generation Networks (NGN) developed, Flow-State-Aware Transport technology (FSA) [50] was presented as a method to provide QoS in these networks. FSA was proposed by British Telecom, Anagran, and the Electronics and Telecommunications Research Institute (ETRI), and was endorsed in ITU-T Recomendation Y.2121 in January 2008. In general, ITU-T proposes that the target of differentiation should be single flows, however, the QoS architecture, especially at the core, should be based on DiffServ. Therefore, the aggregation of the flows is inevitable.

The idea behind FSA is to provide a robust QoS architecture, with differentiation capabilities matching those of IntServ, yet scalable. To achieve scalability, certain flow aggregation was unavoidable, however, the QoS controls operate on a per-flow basis, not on per-aggregate basis, as in DiffServ. Additionally, FSA is not to provide strict assurances. ITU-T recommends [50] that "it is not necessary for FSA nodes to guarantee support under all possible conditions, only that they have high confidence that they can support the resulting request under normal operating conditions". Such an approach, i.e., statistical assurances, became practically a mainline for all QoS architectures since the IntServ's utter lack of scalability had been proved.

The FSA QoS controls are designed to be agnostic to the underlying transport technology. This goal is in line with NGN's general trends to separate service functions from transport functions and to be able to provide QoS over various underlying architectures. The last assumption covers interoperability and sharing the common network resources among different transport technologies. Any network link may not be dedicated to carrying FSA traffic only. However, when a link is used for transporting a mixture of traffic, the FSA node needs to assume that a certain part of the link capacity is guaranteed to be available solely for the Flow-State-Aware traffic. There is a set of recommendations in [50] on how to manage and limit the capacity provided for each traffic.

Flow-Aggregate-Based Services Soon after the disclosure of FSA, the researchers from Electronics and Telecommunications Research Institute (ETRI), proposed their solution to QoS provisioning in packet networks: the Flow-Aggregate-Based Services (FAbS) [58]. FAbS origins from FSA and aims at resolving the FSA issues, mainly by introducing two novel building blocks: inter-domain flow aggregation and endpoint implicit admission control.

FAbS focuses on three aspects of congestion, i.e., instantaneous congestion, sustainable congestion and congestion avoidance. The distinction between instantaneous and sustainable congestion lies in the observation spectrum. The former refers to packet or burst level congestion (when occasional bursts cause congestion), while the latter refers to flow level congestion (when there are more flows than a network can handle). Instantaneous congestion is mitigated, in FSA, through proper flow aggregations and packet discards. Sustainable congestion is resolved by admission control, rate limiting and flow discards. Finally, for congestion avoidance, a protection switching mechanism is proposed.

4.3 Flow definition

Flow-aware QoS architectures, as the name implies, aim at providing guarantees and service differentiation based on transmission of flows. Such architectures recognize that *a flow* is the most proper entity to which QoS mechanisms should be applied. In general, *a flow* is associated with a single logical connection, e.g., a single VoIP transmission between any two end users. Every application can simultaneously create and maintain many flows and each one of them is subject to separate treatment by QoS mechanisms. Flow-aware architectures refrain from assigning flows to aggregates, however, if it is necessary to aggregate traffic, the QoS differentiation still remains on a per-flow basis.

All presented architectures perform service differentiation based on individual flows. However, “a flow” is not understood exactly the same in each of them. The most common identification is the, so-called, 5-tuple, i.e., source and destination IPv4 addresses, source and destination IPv4 port numbers and the transport protocol used for transmission. In case of IPv6, 5-tuple changes into 3-tuple: source and destination IPv6 addresses and the flow-label field. This means, that “a flow” is considered as a set of packets that have the same values in the mentioned 5-tuple (or 3-tuple in case of IPv6). Now, I elaborate on the differences in flow perception.

In the basic Internet architecture all packets receive the same QoS; usually they are forwarded in each router according to the First In, First Out (FIFO) queuing discipline. For Integrated Services, every router must implement an appropriate QoS for each flow. The flow is defined as a stream of related datagrams from a single user activity (e.g., a single voice transmission or video stream). Because every flow is related to a single service, all packets within the flow must be treated equally, with the same QoS. In Connectionless approach, a flow is defined as a stream of packets between two applications, in a client and in a server. Flows are uniquely identified by a 5-tuple. Therefore, the same set of end users may easily create many flow instances in the network and each of them is treated individually.

The exact definition of a flow does not appear in the available descriptions of the Dynamic Packet State approach. However, the author in [103] states that the applications that share a link create separate flows, e.g., a file-transfer application and an audio application create two flows. Hence, the approach is similar to the common understanding of a flow. Caspian Networks and Anagran adopt the flow recognition by the 5-tuple of IPv4 header fields or 3-tuple in case of IPv6. Similarly to DPS, the authors of Feedback and Distribution do not specify exactly what a flow means. From the analysis, however, it can be deduced that a flow is associated with a single transmission. Given that there is a clear distinction between TCP and UDP flows, the standard 5-tuple recognition can be applied to this model.

In Flow-Based Differentiated Services, the distinction between the flows is based on the source-and-destination (S-D) pair of the IP addresses and a value of the DiffServ field (DS field) [83]. Therefore, all transmissions between the same end users, and classified to the same class of service, are regarded as a single flow. This approach is not perfect as, for example, an S-D pair may be identical for all connections between two networks hidden under NAT (Network Address Translation), which in extreme cases may even exacerbate the QoS for high priority flows with respect to the original DiffServ.

The important contribution of the Flow-State-Aware transport is the elaborate description of flows, their parameters and classes of service to which they may belong. The ITU-T Recommendation [50] defines a flow as: ‘a unidirectional sequence of packets with the property that, along any given network link, a flow identifier has the same value for every packet’. A flow identifier is recommended to be derived from the standard 5-tuple of the IP header as well as the value of the DS field. However, it may also be defined by the multi-protocol label switching (MPLS) label. Therefore, the term flow in FSA may indicate either IP 5-tuple flows or aggregates of them. Flow-Aggregate-Based Services is derived from the FSA transport technology and as such uses the same definition of the flows.

4.4 Classes of service

A class of service (CoS) is one of the fundamental aspect of every QoS architecture, not merely a means to provide service differentiation. In general, CoS represents a group of flows that are treated following class-specific rules. Depending on the proposed solution, CoS may be defined thoroughly, presented as a set of rules or left entirely for the operator to define and implement.

Integrated Services The Integrated Services model has three explicitly defined classes of service. They are as follows: *Guaranteed Service* (GS), *Predictive Service* (PS), *Best effort Service* (BE).

The *Guaranteed Service* class provides a certain amount of ensured bandwidth, absolutely no losses of packets due to buffer overloads and a perfectly reliable upper bound on delay of packets in the end-to-end relation. This service is designed for delay-intolerant real-time applications.

In the *Predictive Service* (also named in [42], [2], [46] as *Controlled Load*) the flow does not obtain strict QoS guarantees. Instead, the application receives a constant level of service equivalent to that obtained with the *Best effort Service* at light loads. It means that even under severe congestion, the quality of transmission should not degrade. This class of service was planned for real-time applications, which tolerate occasional loss of packets (e.g., VoIP) or which may adjust to the level of service that is offered at the moment.

The third service level (de facto unclassified) is oriented towards classical data transmission, without any QoS guarantees. When congestion occurs, the quality of transmission degrades. This service is designed for elastic applications, due to their adaptability to the offered level of service.

Connectionless Approach The flow classification strategy in connectionless approach is based on the method introduced in [17]. The classification for both TCP and UDP flows is performed on the basis of different treatment that is required by different traffic types. Therefore, all the applications with similar service requirements are likely to fall under the same class. The classification process can also be enhanced by the port number information. However, it is not considered to be sufficient source of information, but rather as an addition to the data gathered dynamically. In [17], six traffic classes are proposed, three for TCP flows, and three for UDP. The main idea behind dividing flows into these categories is to separate flows which require fast response times from those which are delay insensitive.

TCP flows may be classified as: *Interactive*, *Bulk Transfer With Reserved Bandwidth*, *Bulk Transfer With Best Effort*. The *Interactive* class is suited for applications which require short round trip time, like: Telnet, X-Windows or web browsing. These applications may last for very long time, but they predominantly use short packets for transmission. If the TCP flow is not interactive (too many long packets arrive), it is classified as a bulk transfer. If some portion of the reserved bandwidth is available, flows are moved to the *Bulk Transfer With Reserved Bandwidth* class, otherwise, the *Bulk Transfer With Best Effort* class is their last choice. Whenever the bandwidth becomes available, these flows may be moved to the reserved bandwidth class.

UDP flows may belong to the following classes: *Low Latency*, *Real Time*, and *Bulk Best Effort* with *Low Latency* being the default class. This class contains flows of very low bandwidth, for example, voice transfers, network control packets, etc. If the flow exceeds the threshold bandwidth it is moved to the *Bulk Best Effort* class. The *Real Time* class is designed for applications which cannot fit into the *Low Latency* class but are delay sensitive. These applications include: high quality voice connection, video streaming and conferencing, etc.

Dynamic Packet State DPS, as announced in [103] was developed “to bridge this long-standing gap between stateless and stateful solutions in packet switched networks”. DPS does not really define any classes of service on its own. The method focuses on providing *Guaranteed Service* (GS), a CoS known from IntServ, but without using flow-state information in the core network.

The name GS is only used in DPS to provide an analogy to stateful solutions. Although one CoS is identified, service differentiation is still possible, as certain flow requirements are associated with each flow, namely: the reserved rate and the deadline of the last packet that was served by a node. These parameters come from the Jitter Virtual Clock algorithm which is described in Section 4.5. Therefore, by changing the values of such parameters, DPS provides differentiated treatment of flows, though only one CoS is mentioned.

Caspian Networks / Anagran Caspian Networks in [98] proposed three types of service, namely: *Available Rate* (AR), *Maximum Rate* (MR), and *Guaranteed Rate* (GR). AR traffic does not have real-time requirements associated with the flow. Therefore, AR flows have very loose delay and jitter characteristics as well as relatively relaxed discard (loss) requirements. MR traffic, on the other hand, requires more rigid delay and jitter assurances and is more sensitive to traffic loss. Typically, MR flows will correspond to UDP-based real-time transmissions, such as voice or video.

GR traffic is similar to MR traffic with regard to its characteristics. It also has strict delay, jitter and loss requirements, however, the rate of the flow which is desirable by the end user is fed to the network prior to transmission, either by explicit signaling, examining the Real-Time Transport Protocol (RTP) type or by user-defined traffic profiles.

It needs to be noted, however, that these three classes of service are merely coarse characterizations of quantified state information that is associated with different types of transmission. Within each CoS, multiple flows may receive similar, yet differential treatment, including differences in delay variations and delay characteristics.

The above characteristics of CoS derived from Caspian Networks find their place in Anagran as well. Anagran supports AR and GR classes, however, new classes can be defined and created by network administrators.

Feedback and Distribution The Feedback and Distribution method does not specify any CoS. There is, however, distinction between low-priority and high-priority traffic in the network. Profile meters measure each flow’s rate and send those measurements to markers. A marker sets a high priority to packets of flows whose rate is lower than the guaranteed rate and low priority to all other flows. When congestion occurs, surplus packets from the low priority flows are dropped, therefore, the flow is shaped to the guaranteed rate. Although simple, this approach proves to be effective.

Flow-Based Differentiated Services Flow-Based Differentiated Services operates on classes that were defined by DiffServ itself, just provides flow-awareness. DiffServ envisages using the following classes of service: *Expedited Forwarding* (EF), *Assured Forwarding* (AF) and unclassified service.

The EF class ensures low packet delays and low packet latency variations. Additionally, traffic belonging to this class has certain amount of link's bandwidth reserved. The guaranteed EF rate must be settable by the network administrator. The AF class does not impose any guarantees. Instead, the AF traffic is to be delivered with a probability no less than a certain threshold. AF provides forwarding of IP packets in four independent AF subclasses. Each subclass consists of three levels the of packet drop precedence. Best effort traffic is covered under the unclassified service. A more detailed description of the EF and AF classes can be found in [52] and [47], respectively.

Flow-State-Aware Transport In FSA, the following four classes of service, referred to as 'service contexts', are defined: *Available Rate Service* (ARS), *Guaranteed Rate Service* (GRS), *Maximum Rate Service* (MRS), and *Variable Rate Service* (VRS).

ARS is similar to the ATM's available bit rate (ABR) and is, typically, used for data traffic flows. GRS is similar to the guaranteed service class in IntServ and is designed for applications that require guaranteed bandwidth for the entire duration of the flow. MRS is designed for video, voice, or other streaming media. The distinctive difference between GRS and MRS is that MRS flows have the option of 'immediate transmission', i.e., they do not need to wait for the network response and can send traffic immediately after the request. VRS is a combination of MRS and ARS and is designed for obtaining a minimum response time for a transaction.

FSA carefully defines four service contexts and specifies the signaling messages used to set up the connection and inform the nodes about the requirements. Information such as the requested and offered rates is exchanged and negotiated if necessary. For each CoS these messages have their own own meaning and the respective exchange processes are different. The specification is detailed and seems to cover all the possible application requirements.

Flow-Aggregate-Based Services As stated by the authors in [58], one of the most significant contributions of FSA was its elaborate description of flows and provided classes of service. As such, FAbS also adopts all the classes of service proposed by FSA.

4.5 Architecture

This section describes the means to provide QoS in each architecture. Therefore, all the blocks that contribute to QoS provisioning, e.g., admission control, scheduling, meters, markers, etc., are mentioned herein. Additionally, some method-specific solutions are also presented in this section.

Integrated Services Every router in conformity with the Integrated Services model, must implement the following mechanisms of traffic control: packet scheduling, packet classification, and admission control. The packet scheduler is responsible for altering the order of datagrams in the outgoing queues. This procedure is necessary when congestion occurs, and it allows certain flows to be treated with a higher priority, ascertaining a proper QoS to them. The packet scheduler is also responsible for dropping packets if necessary.

Each incoming packet must be mapped into some class of service. This is the role of the classifier module. The mapping may be performed based on certain packets' header fields or some additional information. All packets in the same class are treated equally by the packet scheduler. The admission control module is responsible for admitting or rejecting new flows. The decision is made based on whether admitting a new flow would negatively impact earlier guarantees. The admission control block operates on each node, and takes local accept/reject decisions. A new flow must find a path along which every node will be able to accept its requirements.

Connectionless Approach Connectionless Approach defines the traffic conditioner, a part of a router which performs flow-level service differentiation. Traffic conditioning is performed on flows, rather than on individual packets. Flows are maintained in a flow-list. If the flow associated with an arriving packet is not on the flow-list, a new flow entry is attached to the list.

The functions of the router are divided into real-time and background operations. The real-time data path has two major functions, i.e., to identify the flow for the input packet and to schedule the packet to the output. The background functions are supposed to: classify the flows, perform admission control and estimate the bandwidth of different classes of traffic. The classification is performed on-the-fly for each flow, based on its traffic characteristics.

Scheduler is a key component in bandwidth management. Scheduling of previously classified flows is performed with two priorities: high and low. High priority traffic is scheduled without any delay and limited by the admission control mechanism. Low priority traffic, on the other hand, is scheduled according to the set of rules. Flows in the *Real Time* and *Low Latency* classes are handled with high priority, whereas the rest are scheduled with low priority. Six general rules of packet scheduling are enumerated in [80].

Additionally, to effectively manage queue lengths, a congestion control mechanism, similar to the drop tail, is suggested to restrict the excessive traffic for best effort classes. Again, similarly to IntServ, Connectionless Approach does not define exact algorithms. Instead, an extensive list of guidelines is provided.

Dynamic Packet State In DPS, only edge routers perform per-flow management, whereas core routers do not. However, DPS provides end-to-end per-flow delay and bandwidth guarantees as defined in IntServ. To achieve that, two algorithms are proposed: one for the data plane to schedule packets, the other for the control plane to perform admission control.

Scheduling in DPS is based on the Jitter Virtual Clock (Jitter-VC) algorithm, which is a non-work-conserving version of the Virtual Clock algorithm [116]. In Jitter-VC, each packet is assigned en eligible time and a deadline, upon its arrival. The packet is held in the system until it becomes eligible, i.e., the system current time exceeds the packet's eligible time. Then, the scheduler orders transmission of eligible packets according to their deadlines, starting from the one which has the closest deadline. It is claimed that Jitter-VC servers can provide the same guaranteed service as a network of Weighted Fair Queuing (WFQ) [21] servers [105].

For the purpose of realizing scheduling in the core, which is to be stateless, a variant of Jitter-VC, called Core-Jitter-VC (CJVC), which does not require flow state at core nodes was proposed in [105]. It was shown that CJVC can provide the same guarantees as a network of Jitter-VC servers, hence the same as WFQ servers. The key idea behind CJVS is to have ingress nodes encode scheduling parameters in each packet's header.

In DPS, core nodes do not maintain any per-flow state. It is therefore difficult to decide whether a new flow may be admitted or not. To cope with this issue, each node in DPS keeps an aggregate reservation rate parameter for each outgoing link. The most straightforward condition that has to be met in order to admit a new flow is as follows: $R + r \leq C$, where: R is the aggregate reservation rate, r is the rate reservation request and C denotes the link capacity. Unfortunately, due to: partial reservation failures, packet losses, and link and node failures, the above admission control scheme is not robust. Therefore, DPS uses a more sophisticated approach, the upper bound of R is estimated and periodically recalibrated. Similarly, as in the case of the scheduling algorithm, the exact mathematical formulas for implementing the admission control in DPS are presented in [105] and [103].

Caspian Networks / Anagran Caspian Networks, in [98], presented a model of a network router which consists of an ingress micro-flow manager, an egress micro-flow manager and a memory. Although many blocks are mentioned in the document, their operation is quite straightforward and nothing solution-specific is there. Concerning the scheduling, Caspian Networks recommends using the WFQ algorithm. It is argued that with so rich state information, the WFQ scheduler can be efficient.

As for Anagran, the situation is similar, yet the information is even less precise. That is due to the fact that Anagran is an actual living technology as the device that incorporates the features exists. This device is the FR-1000 router. The router provides the Intelligent Flow Discard (IFD) technology. The idea behind this name is, essentially, just the flow-based admission control. Behavioral Traffic Control (BTC) is responsible for constantly monitoring all active flows, and comparing their behavior against a simple set of operator-defined rules per flow class. BTC can identify 'suspect' flows based on the following criteria: their duration, byte count, source/destination addresses, or other criteria. For the flows which require some form of corrective or policing action, BTC can: reduce the allowed maximum rate of the flow, change the class of the flow (to lower or higher), or forward the flow over different output port.

Unfortunately, available patents, i.e., [99] and [98] say little about the mentioned technologies. Rather, they present a set of rules that should be followed.

Feedback and Distribution The Feedback and Distribution architecture defines a scheduling discipline which distinguishes high and low priority traffic. Packets that are treated as low priority have a higher probability of being dropped in case of congestion. The solution does not specify any admission control mechanism. It is assumed that local profile meters, present at each end user node, are able to perform admission decisions, however, it is not explicitly mentioned.

The simplest queuing discipline in this architecture is presented in Figure 4.2(a). The scheduler contains two separate queues, one for the high priority packets, and the other for low priority ones. The preference is given to the flows from the high priority queue. Due to the nature of TCP traffic the presented queuing fashion cannot guarantee low jitter for UDP flows. This issue is considered significant, as the authors of [68] notice that the jitter vulnerability for the interactive applications increases along with their transfer rate and due to that it may become a bigger problem in the future. Therefore, Figure 4.2(b) shows the revised version of the scheduling mechanism. Jitter is made lower by the use of the third queue and by classifying traffic into TCP and UDP. Two of the queues serve high priority traffic (separate for TCP and UDP) and one serves low priority traffic (for the rest of the packets). In this method, the burst-like TCP traffic does not interfere with the UDP flows. This method combines the low delay UDP service and the burst-like TCP with bandwidth controlled for each flow.

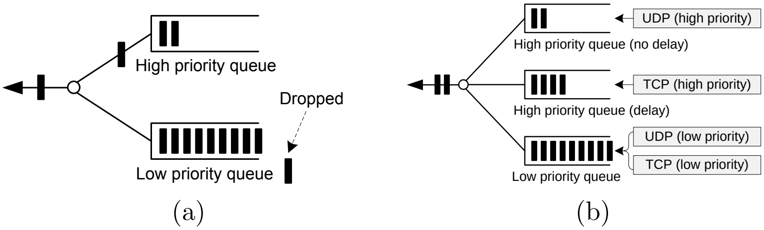


Figure 4.2: Scheduling in the Feedback and Distribution method

The scalability of the architecture is not a problem, since traffic measurement is performed only near the end users. A busy server does not need to provide measurements for multiple flows simultaneously. Instead, it only receives the information from traffic meters and mark the packets accordingly. Additionally, routers do not need to maintain per-flow state for each transmission, which simplifies their operation. Finally, the queuing discipline is particularly easy, but still efficient.

Flow-Based Differentiated Services In order to change class based DiffServ into a flow-based architecture, an estimation of the number of the currently active flows in each class is needed. This is the role of the flow number estimator. Given that, Flow-Based DiffServ presents a method for proportional bandwidth and delay differentiation.

On entering the router, packets are assigned to an appropriate class by the *classifier* and then forwarded to the associated queue. Packets belonging to different flows in the same class are dispatched to the same FIFO queue, as only one instance of queue for each class exists. The number of active flows in each class may be different and time-varying. This is an important indicator for handling traffic. To provide flow-based differentiation, the number of active flows in each class is constantly monitored by the *flow number estimators* which work based on the Bloom filters [9]. According to these estimations the weights for each class in the WFQ algorithm can be dynamically adapted. Additionally, the flow number estimator feeds the *queue management* blocks which, consequently, dynamically allocate buffers and control their queues. The dynamically adjusted *WFQ scheduler* is responsible for proportional flow-based bandwidth allocation among the classes, whereas queue management is employed to achieve proportional delay differentiation. The presented blocks altogether provide flow-based proportional bandwidth and delay differentiation without maintaining per-flow state in the router.

To achieve proportional QoS differentiation between classes (see page 53) the actual bandwidth and delay must be adjusted proportionally to the defined quality ratio and the number of active flows in each class. Bandwidth allocation is performed by dynamically changing the weight of each class in a WFQ scheduler. The ratio of weights for two different classes is presented in the following equation: $w_i/w_j = N_{act,i}/N_{act,j} \times \alpha_i/\alpha_j$, where w_i, $N_{act,i}$ and α_i are the WFQ weight, the estimated number of active flows and the bandwidth differentiation parameter for class i, respectively. Delay adjustment is handled by the queue management block which obtains information from the flow number estimator and from the queue. Having that data it is possible to adapt drop probabilities in the Random Early Detection (RED) mechanism in order to hold the average queue length at a target value. By combining these two mechanisms, the proportional delay differentiation can be achieved.

Flow-State-Aware Transport The recommendation [50] forms a set of rules that an FSA architecture should follow. The guidelines include: the parameters that should be taken into account, the required decision time, possible treatment of flows, etc. In FSA, flows are treated differently, according to flow specification parameters, such as: flow identity, class of service, requested rate, preference priority indicator, packet discard priority, burst tolerance, delay priority. Flow identity is derived automatically from each packet. The rest of the parameters must be signaled prior to transmission, except for the packet discard priority which is not included in the signaling information.

The requested rate parameter has different meanings for different classes of service, however, it is required for each of them. The preference priority indicates the priority for admission decision, i.e., flows with a greater preference priority will be accepted first. The packet discard priority, on the other hand, is required to distinguish between at least two values, namely: ‘discard first’ and ‘discard last’. It is used for packet discard decisions upon congestion.

Typically, video and voice transmissions require lower delay variance than file transfers. To encompass a wide range of existing and future application needs, the delay priority parameter has been proposed. This may give some additional information for the queuing disciplines in the FSA nodes on how to manage the packet scheduling process. Moreover, due to queuing procedures and the nature of the Internet traffic, packets often arrive in bursts. It is, therefore, vital for the FSA nodes to apply a level of tolerance (burst tolerance parameter) to rates that exceed the requested rate for a short duration.

Ingress nodes may aggregate selected flows into fewer aggregates which facilities the flow treatment in the core. Flow aggregation plays an important role in FSA. Despite aggregation, the network maintains per-flow treatment, as when single flows are aggregated into one instance, that new instance is subject to appropriate QoS constraints. For example, in case of aggregated flows with a guaranteed rate, the reserved rate for an aggregate will be the sum of the reservations for each flow individually. The ITU-T recommendation [51] provides a set of rules for the exchange of information on flow aggregates between domains.

Flow-Aggregate-Based Services As mentioned in Section 4.2, FAbS focuses on dealing with instantaneous congestion, sustainable congestion and providing congestion avoidance. The resolution of instantaneous congestion is based on Inter-Domain Flow Aggregation (IDFA), while the sustainable congestion incorporates endpoint implicit admission control and endpoint rate limiting with DiffProbe delay measurement. The congestion avoidance is, basically, an MPLS traffic engineering and is left for future studies.

FAbS flow aggregation is similar to that proposed by FSA. However, considering the fact that a flow passing through the aggregation-deaggregation process can exhibit inferior performance than if it has not been put through it [57], flow aggregation should be executed only if it is possible to carry the aggregates across the network domain. This mechanism is referred to as Inter-Domain Flow Aggregation (IDFA). The domain is defined as a single administrative network domain in which flow aggregation policy remains the same. The main idea of IDFA is that a flow membership should remain unchanged as much as possible.

The suggested admission control in FAbS checks the congestion status of the network using end-to-end delay measurement by DiffProbe [102]. DiffProbe measures one-way delay of the target class of service using the interarrival time between the supreme class and the target class packets. The greater the difference between both packet arrivals, the greater congestion is assumed along the path. For DiffProbe to work, it needs to be assured that both DiffProbe packets as well as the new transmission will follow the same path. Such a behavior can be assured by MPLS.

4.6 Signaling

The essence of signaling is grounded on the following question: how to feed the network with the information on specific treatment of flows? In other words, how to inform the nodes that a new flow should be treated, e.g., with priority. The task is not trivial and numerous approaches to the problem are known.

Integrated Services For the purpose of realizing Integrated Services, IETF specified the Resource Reservation Protocol (RSVP) [12], [13], [77], described its interoperation with the IntServ networks [114], extensions [49], [92], and additional procedures [61]. The IntServ model is not strictly associated with the RSVP protocol. IntServ may interoperate with various reservation protocols, and RSVP is an instance of such a protocol (although it is practically the sole example).

RSVP is receiver oriented, i.e., the receiver (all of them in the case of multicast transmission) of the transmission initiates and maintains the resource reservation procedures. The parameters of the transmission are stored at each device along the path with the so-called *soft state* approach which means that periodic refresh messages are sent to maintain the state along the reserved path. In the absence of this refresh messages, the state is automatically deleted.

It is obvious that the RSVP refresh messages increase the traffic in the network. This growth is strictly proportional to the number of existing paths, and therefore, becomes a significant problem while dealing with multicast transmissions. The soft state approach also increases router overloading, decreasing the standard CPU time available for basic routers' actions. Moreover, every router needs to store and process great amounts of information. All the above necessities, unfortunately, render the RSVP protocol unscalable.

Connectionless Approach In Connectionless Approach, all the decisions in the nodes are taken based on current flow characteristics. Therefore, they operate independently, and as such the solution does not require any kind of signaling. This is a great advantage of the architecture as, usually, signaling involves scalability problems.

Dynamic Packet State DPS encapsulates flow-state information within packets. This information is then used by core routers to treat flows according to their requirements. As core nodes read state information from packet headers, they do not need to remember it, hence the stateless core paradigm. For carrying state information, 4 bits from the Type of Service (ToS) byte (or DS field) reserved for local and experimental purposes, and up to 13 bits from the Fragment Offset of IPv4 header fields are used.

State information is injected into each packet upon arrival in the ingress node. Each subsequent node along the path processes the packet state and eventually updates both its internal state (general, not specific to each flow) and the packet state before forwarding it. The egress node removes the state from the packet header. Although the end users do not observe any difference in the protocol stack, the network devices have to deal with minimum incompatibility with IPv4 due to imposed changes in understanding certain packet header fields. Unfortunately, this triggers another problem with the implementation, as in order for DPS to operate, all nodes in the network must be DPS aware.

Additionally, for the purpose of admission control, DPS considers a lightweight signaling protocol to be used within the domain, such as RSVP. The utilization of RSVP is, however, different than in case of IntServ, as here, nodes do not need to keep per-flow states. Instead, only the aggregate reservation rate for each outgoing link is maintained. The use of an explicit signaling protocol is another reason why all nodes within the network must be DPS aware.

Caspian Networks / Anagran Neither Caspian Networks nor Anagran specify the signaling method to be used. Most of the work is done by the device which measures the flow's traffic and assigns it to a certain CoS. However, in case of the Guaranteed Rate class, the rate needs to be fed to the network prior to transmission. In [98], it is stated that any kind of explicit signaling can be used, e.g., RSVP or ATM/Frame Relay signaling. Additionally, some information can be derived from examining the RTP protocol type, or user-defined traffic policies.

Although the exact signaling method is not directly specified, both Caspian Networks and Anagran propose in-band signaling. The former suggests including the QoS requirements within the first packet of each new flow. Routers need to remember this information until the flow lasts. In case of Anagran, in FR-1000, the commercially available router, the TIA-1039 [4] signaling protocol is used. TIA-1039 is an in-band signaling protocol which adds a small amount of extra information within each TCP header. This allows a sender to communicate what rate traffic can be sent over the incoming TCP connection, and also allows the requestor to either accept that rate or request a lower rate.

Feedback and Distribution Profile meters measure the traffic for each end user and then feed the information to markers. Having that information, markers can assign certain packets to high or low priority. High, when a flow does not exceed the guaranteed rate, and low, otherwise. In case of congestion, packets with low priority are dropped first.

Given that the authors do not specify the way certain blocks communicate with each other, it can easily be assumed that the communication is based on regular TCP/IP transmissions. It is true that putting profile meters close to the end users make them feasible, however, the amount of signaling data that need to be sent through the network is significant. Therefore, Feedback and Distribution reduces the requirement to measure and maintain flow-state information on core routers at the cost of increased network load due to extensive signaling associated with each flow.

Flow-Based Differentiated Services In Flow-Based Differentiated Services, much like in plain DiffServ, the signaling is embedded into packet headers. For this purpose, the Type of Service (ToS) field in the IPv4 packet header has been transformed into a DiffServ field (or DS field) [83]. Into this field, a fixed number identifying a certain class is inserted. It needs to be noted that flows do not communicate their specific QoS requirements. Rather, they chose one of the predefined classes (or the operator makes the choice) which most closely suit their needs.

In original DiffServ, all packets having the same DS field number are treated as one instance, therefore, receive identical treatment. In Flow-Based DiffServ, additionally, the amount of reserved resources varies according to the number of active flows in one class, hence the flow-awareness.

Flow-State-Aware Transport ITU-T allows for the use of in-band or out-of-band signaling in FSA, however, wherever possible, in-band signaling is strongly recommended. In-band signaling means that the messages are within the flow of the data packets and follow the path that is tied to the data packets. They are routed only through nodes that are in the data path. Out-of-band signaling, on the other hand, is when messages are not in the same flow of data packets and may follow a different path. Usually they will visit other nodes in the network, either deliberately or not. Signaling packets and data packets must be recognizable by each FSA node, however, the exact method has not been specified yet.

Recommendation [50] specifies the following five types of in-band signaling packets: *request, response, confirm, renegotiate, close.* Each CoS in FSA requires specific signaling messages. Depending on the class, some messages are needed before transmission, whereas some are used during it.

Flow-Aggregate-Based Services FAbS, generally, adopts the signaling approach from FSA, i.e., in-band and out-of-band signaling. Moreover, as information on flow aggregates has to be transferred from one domain to the other, FAbS uses flow aggregate information exchange signaling, as presented in [51]. The authors of [58] claim that the signaling complexity of their solution, FAbS, is better than that of FSA, however, the arguments towards this statement are unclear. The foundations may lie in the concept of DiffProbe signaling and in IDFA which by smart aggregations can reduce the amount of exchanged information. Nevertheless, being a technology at its infancy, further analysis is needed.

4.7 Summary

Flow-awareness as a way to provide QoS to the IP networks has become a hot topic since the well known pioneering IntServ architecture proposed by IETF. Since then, several new approaches have been developed. By using different methods, they all try to enable the possibility to differentiate traffic based on a flow entity. The urge to do so is supported by the requirements of new applications, which become more and more demanding, not only in terms of bandwidth.

This chapter presented the most important and original propositions and the key features are summarized and compared in this section. Issues such as packet delay, jitter or the degree of losses associated with each architecture were not evaluated, as they depend mostly on factors which are not solution-specific. For instance, to evaluate packet delays, we need to assume a certain queuing algorithm, whereas most presented architectures can operate with more than one.

4.7.1 Pros and Cons

IntServ is the first QoS architecture for the IP networks, and its developers focused on providing diverse service differentiation possibilities. This makes IntServ a model architecture in terms of QoS guarantees. However, strong assurances came with the price of utterly low scalability. Connectionless Approach went in the opposite direction. There is no signaling in the architecture and the nodes try to recognize flows on the fly. The lack of signaling, however, precludes on-demand service differentiation. In other words, users are not able to pay for a better service. Additionally, automatic flow recognition does not always work. The architecture is not robust, as users may try to imitate other traffic types to get a better treatment.

DPS relieves core routers from maintaining flow-state information which significantly contributes to the solution's scalability. However, the data handling which involves modifications of the packet header in each node (even in the core), the CJVC algorithm, distributed admission control with a relevant signaling protocol, is quite complex. Additionally, the architecture cannot be installed gradually in the network as it requires slight changes in the IP protocol functionality.

Anagran is a working technology, yet the technical side lacks detailed descriptions. On the negative side, routers need to maintain and constantly monitor flow-state of each instance. This might be an easy task for small networks, however, such an approach does not seem suitable for high-speed core networks.

The Feedback and Distribution method simplifies core nodes operation, as they do not need to perform measurements. Instead, they read the information provided within the marked packets and perform actions accordingly. Profile measurements are performed only at the edge, which is feasible, however, this information must be constantly fed to the network for each flow, which does not scale well. Additionally, only two classes of service are proposed, low priority and high priority, which does not provide service differentiation possibilities such as those of IntServ or FSA.

Flow-Based Differentiated Services maintain the scalability of the original DiffServ, which is a great advantage. Despite the fact that flow-based treatment is retained even inside fixed classes of service, there are still DiffServ specific difficulties, such as: limited number of CoS, difficulties in carrying service across domains, admission control issues and complex operation.

Flow-Aware Networking is a solution with many advantages. First of all, it does not require signaling, is simple, efficient and can be installed gradually. Additionally, neither inter-operator nor user-operator agreements are needed. It provides differentiation based on the flow current peak rate and protects low-rate flows. However, in terms of congestion, the admission control block may force new flows (even those that should be prioritized) to wait for the network resources to be available again. Certain mechanisms to mitigate this issue were presented in [25], [26] and [53]. One weakness, that is an effect of the lack of signaling, is poor service differentiation. Flows are divided only on low-rate and high-rate flows and treated accordingly.

Both FSA and FAbS provide great service differentiation. There is plenty of parameters to be assigned to each flow and multiple classes of service. The signaling, however, is quite complex which limits the scalability. Fortunately, due to flow aggregations these architectures seem more scalable than IntServ.

The last issue, that should be mentioned, is the vulnerability to user misbehavior. In particular, since users can create many flows, they may try to game the system by dividing the original single transmission into several flows, hoping to gain an advantage. When a flow distinction is based on the 5-tuple, an end user can create multiple connections on different TCP or UDP ports, therefore, creating multiple flows. The Flow-Based Diffserv architecture seems to be especially prone to such a malicious behavior, as the estimated number of flows directly impacts the system operation. However, the flow distinction in Flow-Based DiffServ is not based on the 5-tuple (port numbers are not taken into account) which means that end users are not able to manipulate with the number of flows. In the approaches such as Connectionless Approach, Caspian Networks and Anagran, Flow-Aware Networking, Flow-State-Aware Transport and Flow-Aggregate-Based Services, the flow division is a real issue and needs to be taken into account.

4.7.2 Perspectives

All the presented flow-aware architectures are well thought, obviously have their pros and cons, however, their future is unclear. Neither of them has been widely implemented in the Internet. What is, then, the reason why each of them fails to become the dominant QoS architecture? To answer this question we need to look at the big picture of QoS. Xiao, in [115], shows that it is commercially difficult to install QoS in the network which, mainly due to over-provisioning, works satisfactorily. Currently, even highly demanding applications can achieve sufficiently good QoS, provided that access networks are not congested (core networks are never congested according to most major networks operators).

This situation has two consequences. First of all, it does not put the pressure on telecom operators to provide any differentiation mechanisms whatsoever. They argue that, when a network works fine, it is best to leave it at that, and occasionally throw in some bandwidth. Such an approach is typical for most operators that believe that "bandwidth is infinite" and more capacity can always be provided. Secondly, in an uncongested network it is, at the very least, difficult to convince users to buy supreme services while the standard service works just fine.

Does it mean that QoS architectures do not have a future? Not necessarily. We can note that the progress in access network capacities is far greater than in the core networks. The proposals of Fiber-To-The-Home (FTTH), Passive Optical Networks (PON) and other broadband access technologies provide more and more bandwidth to the end users. And the bandwidth is always consumed. In the end of 1980's when 155 Mbit/s link was introduced, the operators wondered if they will ever need such a capacity in the core. Today, we can see how wrong they were. In light of these facts, there may be a time when networks start to be congested on a regular basis and the efficient and feasible QoS technology might be needed then.

Based on the information showed in this chapter, I would like to conclude that it is difficult to point out a single solution and claim that it suits the current and the future networks best, as all proposals have their strong and weak sides. However, in FAN, the pros outnumber the cons in comparison with other architectures. Not only the solution is net neutrality compliant, but also very efficient, scalable and easy to implement.

Part III

Quality of Service in FAN

5 Quality of Service differentiation in FAN

It is the quality rather than the quantity that matters.

— Lucius Annaeus Seneca

Quality of service differentiation in FAN is definitively on the weak side. This is due to the fact that FAN does not use any kind of signaling which makes the process of informing the nodes about the incoming transmissions challenging. This chapter shows how much of service differentiation can be provided in FAN networks and what are the capabilities of the architecture. To enhance the service differentiation offered by FAN, I propose some new mechanisms. Some of the results shown in this chapter have been published in [53].

In this chapter I present the following, new mechanisms:

- differentiation blocking approach,
- differentiated queuing approach,
  - bit rate differentiation,
  - fair rate ignoring scheme,
- Static Router Configuration approach,
- Class of Service on Demand approach

The chapter is organized as follows. I start with Section 5.1, explaining how the implicit service differentiation works in FAN. Afterwards, in Section 5.2, I show why new flows may wait for a long time before they are admitted on a FAN link. Those two sections are crucial to understand the operation of the mechanisms proposed later. Section 5.3 presents the differentiated blocking approach. Although differentiated blocking offers great possibilities, using this scheme might have a negative impact on the network performance. This issue is documented in Sections 5.3.1 and 5.3.2. The following section, Section 5.4 shows the differentiated queuing scheme with its two variations, i.e., bit rate differentiation and fair rate ignoring. Section 5.5 presets the Static Router Configuration approach: a method to efficiently realize differentiated blocking in FAN. In Section 5.6, Class of Service on Demand as an approach which combines the possibilities provided by differentiated blocking and differentiated queuing is presented. All the mechanisms proposed in this chapter are discussed with relation to the network neutrality principle in Section 5.7. Finally, Section 5.8 concludes the chapter.

5.1 Implicit service differentiation

To understand how important are the proposed service differentiation mechanisms, first I present the concept of implicit service differentiation in FAN. The general objective of FAN is to ensure low packet latency for streaming flows, while utilizing all residual bandwidth to provide maximum throughput to elastic flows. Figure 5.1 explains how this scheme works by showing the possible scenario on a 3 Mbit/s FAN link. Until 4th second of the simulation, flows 1 and 2 realize their desired bit rates. It may be assumed that flow 1, emitting at approximately 50 kbit/s comes from a streaming application, whereas flow 2 is probably elastic. Nevertheless, both flows emit at a lower rate than the current fair rate and, therefore, are treated with priority. However, in the 4th second, the congestion occurs in the link, as many new transmissions appear. This situation causes fair rate to drop.

In such a situation on a today's classic IP link, the rate of all the existing flows would have been reduced. However, due to FAN's implicit service differentiation scheme, flow 1 remains untouched and its service is preserved. Flow number 2 bit rate was reduced to the level of the fair rate, as this is effectively the maximum value that each flow could realize at the moment.

This procedure is very useful, as it protects low rate flows from degradation, should congestion occurs. This is extremely important because most streaming applications, though operating at low bit rates, cannot function when these bit rates are not provided. If flow number 1 represented VoIP transmission, in the currently existing IP network, the carried voice could have been unrecognizable and, therefore, the connection must have been terminated. As observed in Figure

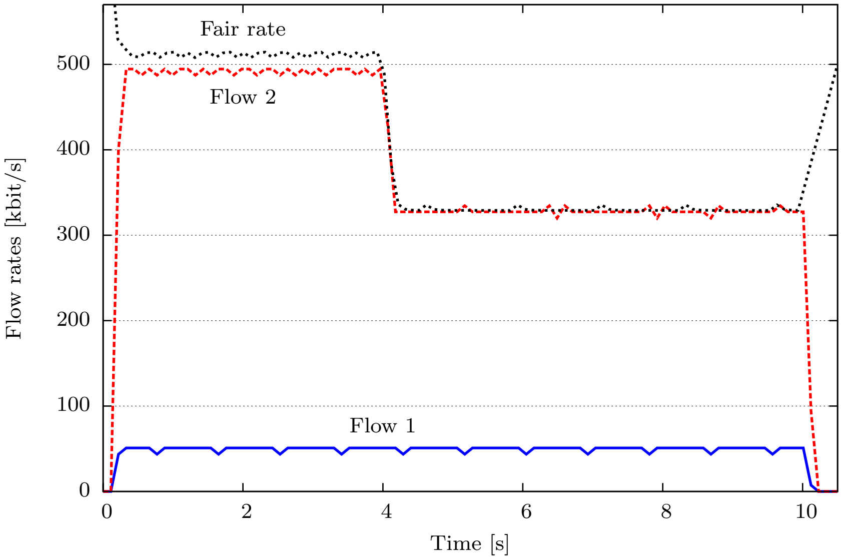


Figure 5.1: Implicit service differentiation in FAN; flow rates and fair rate measurements

5.1, FAN manages to protect this service. The service associated with flow 2 must be degraded, however, the bit rate reduction in the elastic applications has milder consequences.

Under congestion, FAN performance may be considered superior to the behavior of the classic IP network. It is due to the fact, that only a limited number of flows may be simultaneously admitted on a link. Such an approach virtually guarantees that once a flow is admitted, it will perceive at least a decent QoS. Considering the VoIP technology, any accepted flow is bound to obtain a good enough QoS level, which is not necessarily true in case of the current, congested IP network. To demonstrate the difference between the behavior of classic IP and FAN links, a simple simulation was performed. The scenario in which 300 TCP-based elastic flows and 25 UDP-based VoIP flows compete for resources of a 1 Mbit/s link was identical for both cases. Figure 5.2 compares the results by showing the measured fair rate values over time. As can be observed, these values constantly fluctuate and the oscillations are caused by the high frequency of the measurements.

On the classic IP link (lower line), all flows are admitted, once they appear. Since their number is significant, the rate at which they can transmit quickly drops down below 10 kbit/s. On the other hand, FAN (upper line) preserves the fair rate on a level of approximately 40-50 kbit/s. Unfortunately, in order

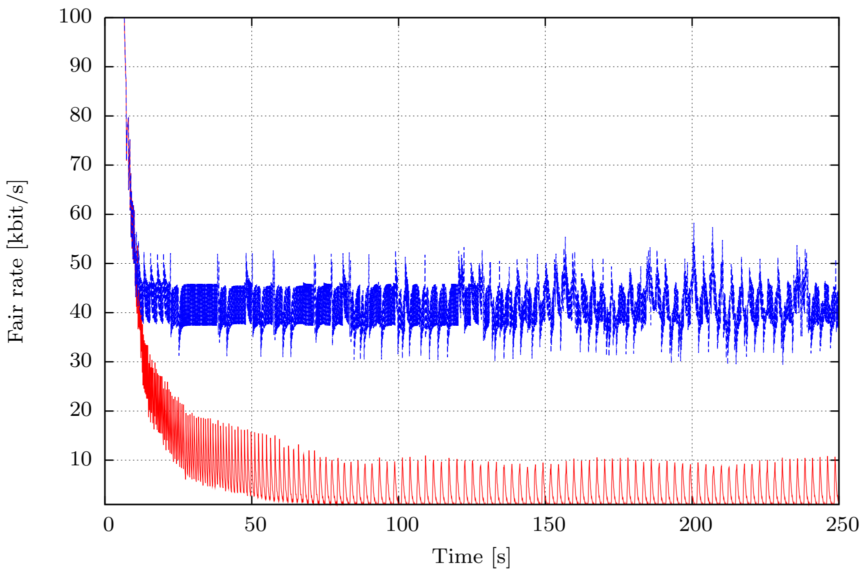


Figure 5.2: Performance under congestion of a classic IP link (lower line) and a FAN link (upper line)

to achieve this goal, some flows must be temporarily blocked. This process is documented in the next section.

5.2 Waiting times

A classic FAN thinking includes a general rule to limit the number of active flows, so that the transmissions currently in progress could always obtain at least a decent QoS level. Although such a behavior is considered beneficial for low-rate streaming applications (like VoIP), in some cases it may be unsatisfactory, due to the admission control flow blocking phenomenon. The purpose of this section is to expose and document that negative aspect of FAN, i.e., blocking the incoming connections upon congestion. Subsequently, a feasible solution to overcome this negative effect is proposed and objectively evaluated.

As explained in Sections 3.5 and 3.6 two congestion indicators are calculated periodically in a FAN router. Fair rate is used to differentiate between streaming and elastic flows within the XP router. Additionally, along with priority load, these indicators are used by the admission control to selectively block new incoming flows, providing the congestion state is detected. If the measured FR is currently below a certain pre-set minimum fair rate value (FR_{min}), or the PL exceeds its maximum threshold (PL_{max}), all new incoming flows are blocked. This routine is presented in Figure 5.3. The values of FR_{min} and PL_{max} must be carefully chosen by the network administrators and this should be done for each link individually.

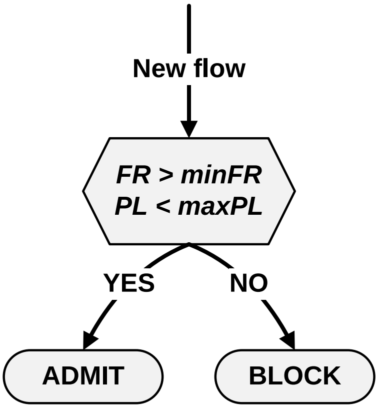


Figure 5.3: Admission control routine in FAN

The waiting time, which is not observed in the current IP network, must be taken into account while assessing the performance of any service, especially the Internet telephony. The availability to make a phone call is a very important factor for the end-users. Additionally, as the Internet becomes part of everyday's life, more and more customers use the VoIP technology, instead of the ordinary PSTN telephone service. This means that in case of emergency, the ability to contact with emergency services depends on the current congestion status in the network. For these customers, the availability to make a phone call is much more important than its quality.

Figure 5.4 presents waiting times for VoIP flows while they compete for network resources with other TCP flows during the scenario, when the 1 Mbit/s link is FAN-aware . In this case, there were 300 background flows, on average having 500 kbits to transmit. On top of that, 25 VoIP flows (20 kbit/s each) wanted to begin their transmission, starting from the 50th second of the simulation run, and continued trying until they were finally admitted. As can be observed, FAN admission control block forced flows to wait until the congestion ended. For some flows, the waiting time was short and nearly unnoticeable. Unfortunately, some of them had to wait for more than 200 seconds before their first packet could be transmitted. Such a situation is very inconvenient for the realization of VoIP connections, especially for the emergency calls. It needs to be mentioned that the

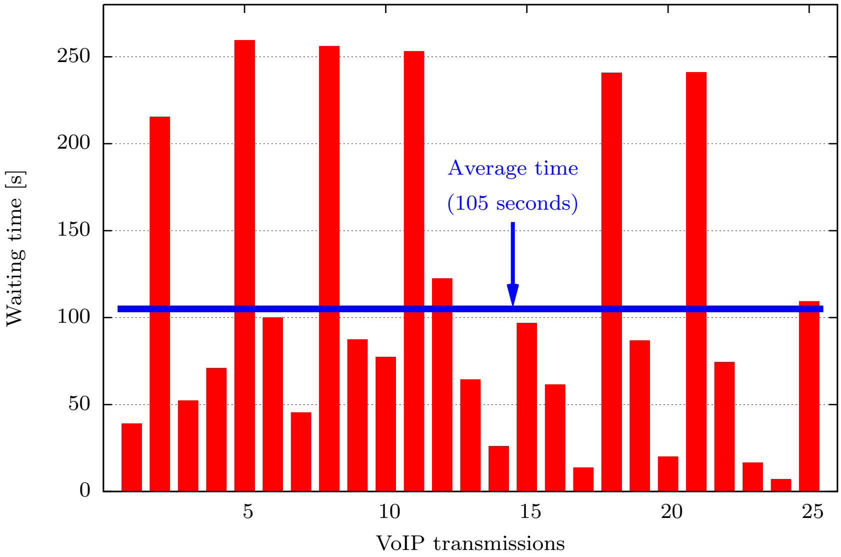
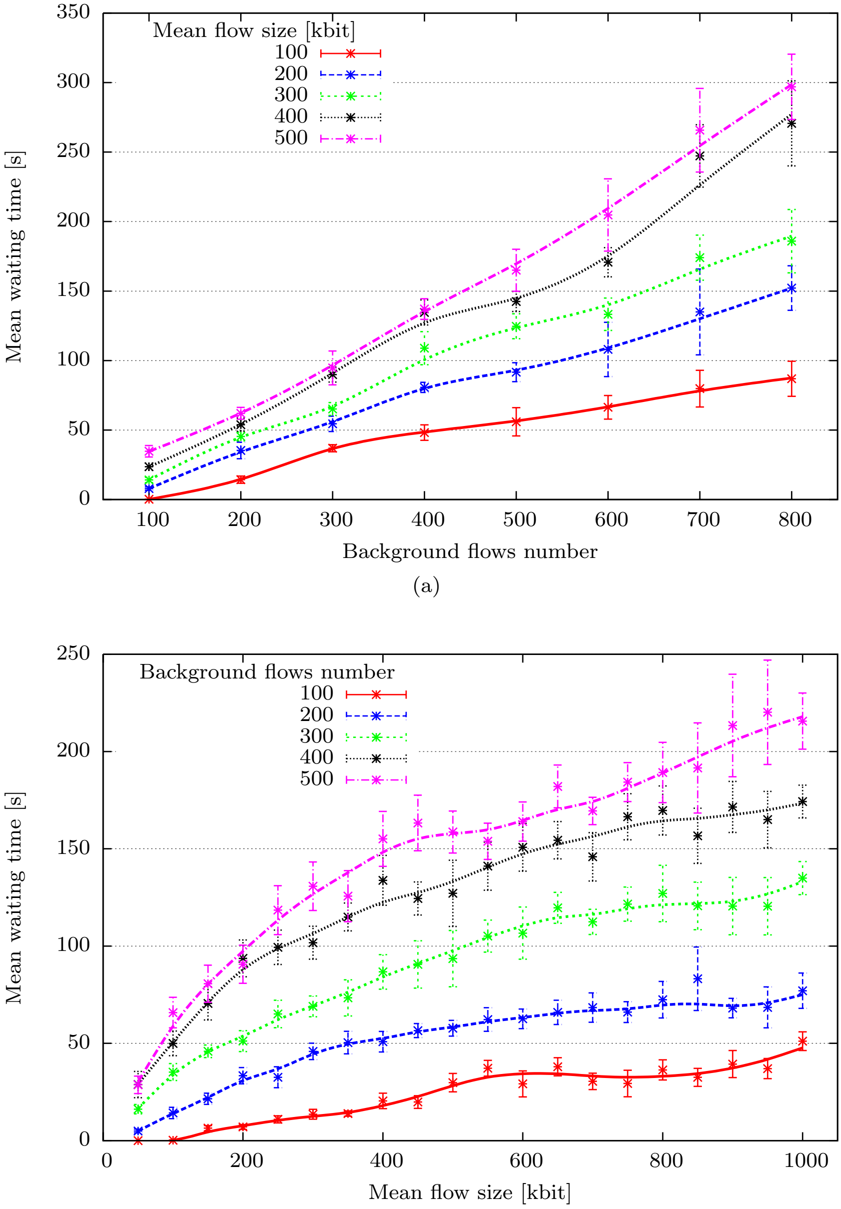


Figure 5.4: Exemplary VoIP connection waiting times

situation presented in Figure 5.4 is only exemplary, yet it illustrates the problem well. The most important lesson from this example is that it is possible that, in FAN, any new flow may be forced to wait for an unreasonable amount of time should congestion occur.

To understand how certain amount of traffic impacts this waiting time, several simulations were performed, each time with various background traffic characteristics. Again, the absolute waiting time values are not as important, as the general dependency and influence of the background traffic on the waiting times. Figure 5.5 presents the mean VoIP flow waiting times with respect to the number of background flows and the mean background flow size. At least 10 simulations were performed to obtain each value: from them the average as well as the 95% confidence intervals (using a Student's t distribution) were calculated.

In Figure 5.5(a), the link congestion rises along with the number of background flows, while in Figure 5.5(b) the increased congestion is caused by varying the mean background flow size. As seen in both parts of the figure, the mean waiting time for transmission grows along with the offered load, which is expected. The greater the flow number, the lower the chance of a particular VoIP flow to be admitted, and therefore, the longer waiting time. On the other hand, when the flow number is constant, but their mean size grows, the more rarely a



(a)

(b)

Figure 5.5: Mean VoIP flow waiting time with respect to the number of background flows (BFN) (a) and the mean background flow size (MFS) (b)

flow ends, hence, a new one may be admitted with a lower frequency, which also increases the average waiting time.

The values presented in Figure 5.5 are averaged. In fact, having in mind the exemplary situation (Figure 5.4), during the simulations, certain amount of VoIP flows observed very short and absolutely acceptable waiting times. However, for the rest of them that period was excessively long and simply much too long for life-saving emergency connections.

It is worth mentioning that FAN does not degrade the performance of streaming flows in comparison to the classic best effort transmissions. In case of today's IP networks, the emergency connections do not observe excessive waiting times, however, they are endangered by congestion, as the low transmission rates may render voice imperceivable. FAN networks, although providing superior transmission quality, may force us to wait for the network resources. Fortunately, both these disadvantages may be overcome by introducing differentiated blocking into FAN networks.

5.3 Differentiated blocking

The differentiated (selective) blocking aims at applying different blocking criteria to newly arriving flows. The standard FAN routine causes the admission control block to make the decision based on currently measured values of the fair rate and priority load (see Figure 5.3). To eliminate long waiting times for certain flows, I propose the differentiated blocking approach, i.e., applying different blocking criteria for priority flows.

In the simplest example, the differentiated blocking scenario includes two classes of service, namely: the standard class and the premium class. The admission control procedure in such a situation is presented in Figure 5.6. The role of the class selector is to recognize which blocking criteria should be applied to the incoming flow. Flows belonging to the standard class are subject to admission control under the rules of the original classless FAN, whereas the premium class flows are always admitted. It is also possible to introduce additional classes of service, however, for the purpose of realizing the emergency calls, the premium class is sufficient.

Differentiated blocking operates only when congestion occurs, as in the other cases, there is obviously no need for blocking the arriving flows. Additionally, this mechanism does not interfere with protected flows. Any flow that is already placed in the protected flow list is always forwarded. Furthermore, differentiated blocking does not prioritize flows that are in progress. In other words, all flows receive the same treatment from the scheduling algorithm once they are admitted.

The procedure presented in Figure 5.6 is well suited for the emergency VoIP connections. All flows related to the VoIP emergency call would belong to the

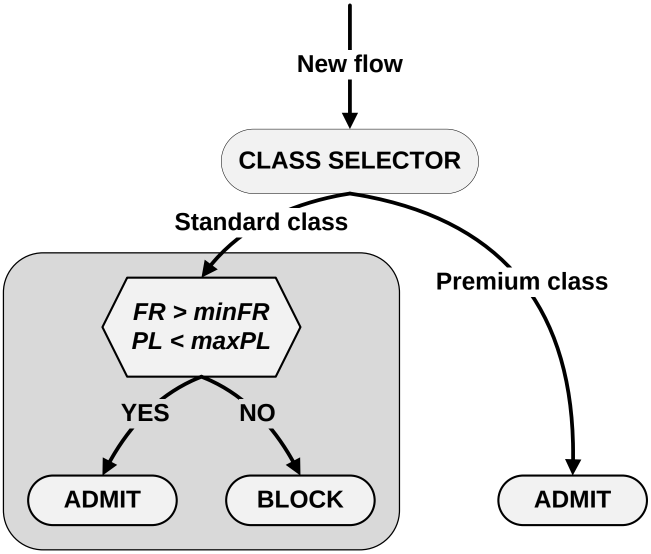


Figure 5.6: Admission control routine of FAN with premium class of flows. The grey area presents the original FAN routine.

premium class, i.e. they would never be blocked by admission control in a FAN router, and therefore, the connection waiting times would always be unnoticeable. In this way, the quality of a voice call is kept high by the FAN's implicit service differentiation scheme, whereas the availability to make a call is protected by the differentiated blocking approach.

This scheme, however, introduces a certain drawback. As we interfere with the admission control mechanism, we may observe the performance degradation. This is due to the fact that prioritized flows are admitted on the link, even under the circumstances in which, to protect the ongoing flows, they normally would not be admitted.

Fortunately, in case of VoIP connections, this behavior has limited impact on the overall link performance for two reasons. Firstly, the required bit rate of a single internet telephony connection is relatively low, especially compared to the core link capacities, and therefore, admitting even a few additional flows should not degrade the quality of the remaining transmissions significantly. Secondly, the fair rate degradation is a temporal process. It is temporal due to the fact that while active flows terminate naturally, new ones are not admitted until the fair rate returns to its desired value.

Although introducing differentiation mechanisms to FAN routers is very simple, the signaling issue remains. As the experience of IntServ and DiffServ has shown, every method of introducing the knowledge about the treatment of particular flows to the network, is inevitably associated with a major increase of complexity or severe scalability reduction. Therefore, each explicit service differentiation mechanism should not rely on any signaling or packet marking procedure, as the IP's and FAN's original simplicity and scalability are to be preserved. To cope with this issue I propose a Static Router Configuration approach, and present it in Section 5.5.

5.3.1 Fair rate degradation

Introducing the differentiated blocking or differentiated queuing is beneficial for some services. However, manipulating with the blocking criteria brings a new problem. It is due to the fact that prioritized flows are admitted on the link, even under the conditions in which they normally would not be. This section documents the negative impact of prioritized flows on the fair rate.

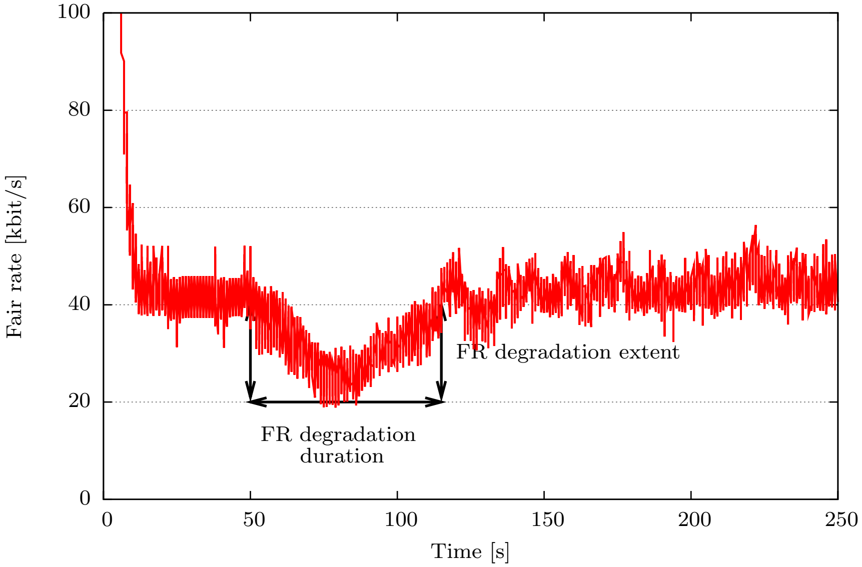


Figure 5.7: Fair rate degradation with differentiated blocking

Considering the scenario described in previous sections, i.e.: 1 Mbit/s link, 300 background flows and 25 VoIP flows starting their transmission after the 50th second, the fair rate measurements, when the VoIP flows are assigned to the premium class, are shown in Figure 5.7. When prioritized flows appear, they are admitted instantly. As a consequence, the fair rate degrades, which is natural, as the link has to concurrently serve more flows than it normally would. 25 VoIP flows start their transmission from 50th second of the simulation with 1 second interval. As may be observed, fair rate continuously drops until approximately 75th second. From then on, each time a background flow ends, FR raises, until it reaches its nominal value (close to the minimum FR value).

It would be natural to try to mitigate this degradation, by dropping some active flows from the protected flow list. However, I argue that the pre-emption process is not necessary when dealing with the Internet telephony. Not introducing preemption seems to be a more adequate solution for the following reasons. Firstly, the required bit rate of a single internet telephony connection is relatively low, especially compared to the core link capacities, and therefore, admitting even a few additional flows should not degrade the quality of the remaining transmissions significantly. Secondly, the fair rate degradation is a temporal process. It is temporal due to the fact that while active flows terminate naturally, new ones are not admitted until the fair rate returns to its desired value. And finally, the FAN architecture does not become more complicated, which is an obvious advantage.

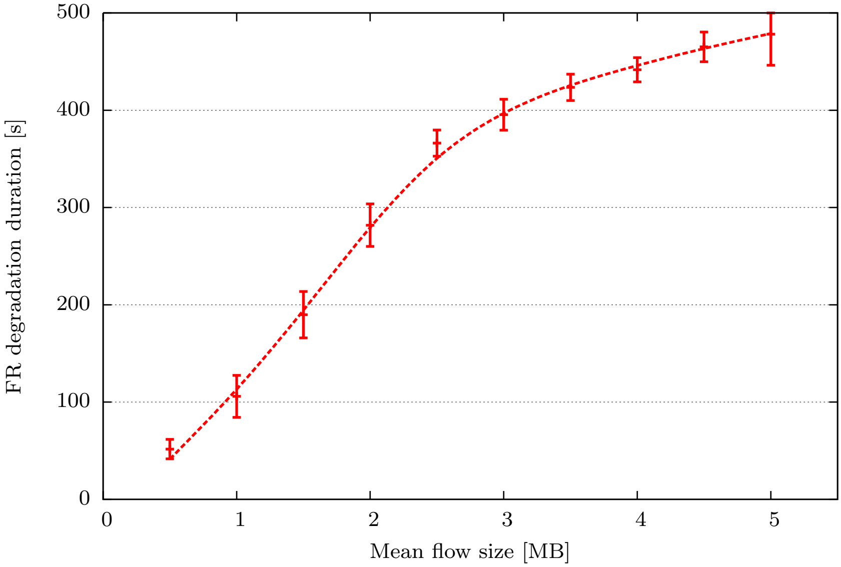


Figure 5.8: Duration of the FR degradation with respect to mean flow size

To support the presented arguments and to evaluate the extent of FR degradation and its length, several simulations were performed. Figure 5.8 shows the length of FR degradation with respect to the various mean background flow sizes, ranged from 500 kB to 5 MB. Apart from the mean background flow size, all the scenarios were identical as before. The minimum FR value was set to 5% of the link capacity, i.e., to 50 kbit/s and under normal circumstances this value does not drop below 45 kbit/s. Therefore, the period of FR degradation was defined as the amount of time when the FR was below 45 kbit/s, due to the appearance of the prioritized traffic. The length of the FR degradation process, as may be seen in this experiment, is strictly dependent of the mean background flow size. The longer the flows, the longer this FR degradation process lasts. This is easily explainable, as when flows are shorter, they end more frequently, therefore, FR raises more rapidly.

Moreover, there is a second factor that contributes to the length of the FR degradation process, and that is the number of active background flows[5]. Obviously, the greater this number is, the more chances that an active flow naturally ends, and consequently, the FR growth becomes faster. This number depends on the link capacity, the minimal FR value and the traffic characteristics.

The temporality of the FR degradation process is one issue which should be accounted. The other is the extent of this degradation. Figures 5.9 and 5.10 show the impact of prioritized VoIP flows on the FR measurements on links with different capacities. Again, the setup was similar to the previous experiment with the difference that flow average size was set to 1 MB and the link capacity was changing from 1 Mbit/s to 5 Mbit/s. In each case, the minimum FR value was set accordingly, so that it would correspond to 50 kbit/s.

As can be observed, greater capacity links suffer less from the prioritized traffic, for two reasons. Firstly, the same amount of the prioritized traffic is less significant on, say a 5 Mbit/s link than on a 1 Mbit/s link. It may be observed in Figure 5.10 as the FR degradation extent is smaller on higher capacity links. Secondly, greater capacity links may serve more flows simultaneously, and, therefore, the FR degradation process is shorter (Figure 5.9).

The experiments presented in this section show that for the realization of emergency calls, the FR degradation should not be considered as a problem. The analysis was performed on low capacity links. However, the simulation results (especially those presented in Figures 5.9 and 5.10) show the general tendency that high capacity links are even less vulnerable to this negative effect of the differentiated blocking procedure. Therefore, it is believed that not caring about the FR degradation for the purpose of realization of the Internet telephony (especially the emergency calls) is the correct and adequate approach.

Although for the emergency calls the FR degradation process is not significant, the differentiated blocking might also be used for much more bandwidth consuming services. In such a case, in order to obtain real prioritization, the pre-emption procedures might be inevitable. Pre-emption, as considered, is a mechanism to delete one (or some) active flow(s) from PFL, when a prioritized

[5] By 'active flow' we mean a flow which *flow id* is on the PFL list.

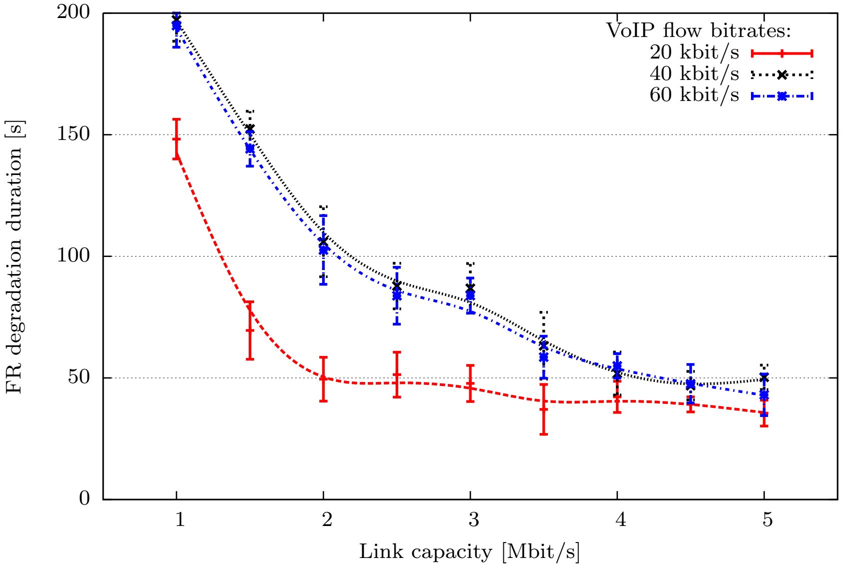


Figure 5.9: FR degradation duration with respect to link capacity

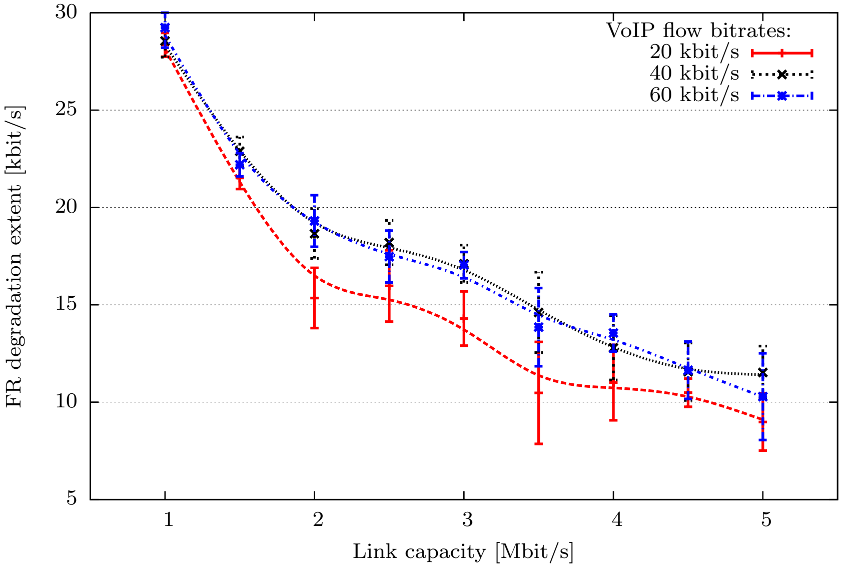


Figure 5.10: FR degradation extent with respect to link capacity

flow appears. In this way, even under severe congestion, the FR values would not be degraded, and the overall performance would be preserved. It needs to be mentioned that one of the FAN principles is that already active flows are guaranteed to be forwarded even under the most severe network conditions. The introduction of pre-emption mechanisms will violate this principle.

However, if pre-emption mechanisms were to be enforced, certain issues need to be resolved first. For instance, which active flow should be deleted from the PFL list? Should it be a randomly selected flow or maybe the one with the longest backlog? The answer to these questions is vital if we consider the fact that potentially to-be-deleted flow can consume less or more bandwidth than the newly arriving prioritized one. Perhaps deleting one flow may not be enough, and we should think on erasing a few small flows in order to admit one big prioritized.

These questions are not easy to answer, but the response is required for the pre-emption mechanism to operate correctly. Therefore, the pre-emption mechanism needs to be examined when possible prioritized services are defined. For the purpose of realizing the emergency calls, there seem to be no need for this mechanism, however, as soon as other usages are identified, this proposition should be reevaluated.

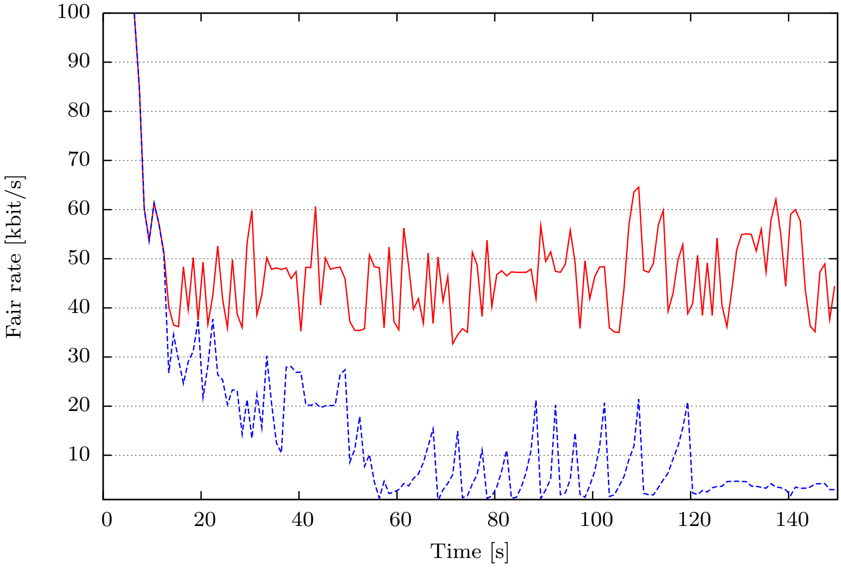
5.3.2 Network failures and differentiated blocking

FAN manages to operate well in terms of congestion. However, the introduction of premium class flows, insignificantly, but still impacts the fair rate. As congestions in a network may also be related to link failures, this scenario should be investigated. Figure 5.11 shows the fair rate measurements on a saturated link. There are 300 TCP flows, 10% of which belong to a premium class. In 50th second of the simulation time, due to a possible failure, additional traffic of identical characteristics is transferred from another link. Otherwise, the simulation setup is identical to the previous experiment. Figure 5.11(a) presents the behavior of the classic IP network (bottom line) and the original FAN without differentiated blocking (upper line).

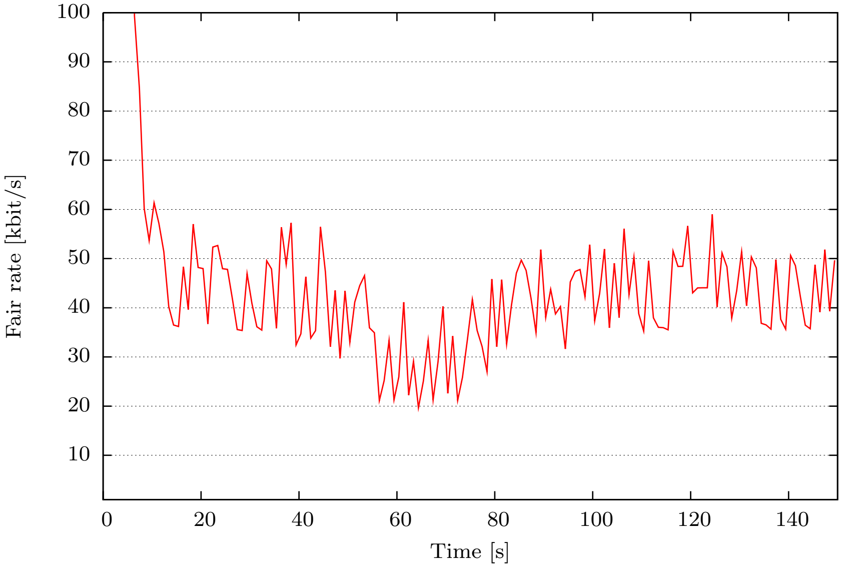
The results obtained on a classic IP link are disastrous. After the 50th second, the fair rate[6] drops from very low to a completely unsatisfactory level. On the contrary, the classic FAN link operates indifferently to any network failure. The 50 kbit/s is the minimum fair rate threshold, and it is kept untouched, when new flows appear in 50th second. This approach is beneficial to the currently realized flows, as they do not suffer from any service degradation.

Figure 5.11(b) illustrates the behavior of a FAN link with the differentiated

[6] Although the method of estimating the fair rate in FAN cannot be applied to the original IP networks, the fair rate in this case has identical meaning, i.e., the bitrate available to each flow.



(a)



(b)

Figure 5.11: Performance during network failures; (a) classic IP link (bottom line) and FAN link (upper line), (b) FAN with differentiated blocking

blocking scheme. Here, 10% of the total traffic is of premium class. Up to the network failure, the fair rate is very similar to the case with FAN link and do differentiated blocking, only a little bit lower. In 50th second, however, FR drops about 15 kbit/s, due to premium class flows that are present in the transferred traffic. These flows are admitted despite the fact that the minimum FR threshold is exceeded. Fortunately, the fair rate degradation is only temporary as FR grows, and quickly achieves its desired values. This behavior is caused by the fact that certain flows naturally terminate their transmission while, at the same time, no new flows are admitted to the link.

FAN with the differentiated blocking performs well, even in terms of network failures. The temporal FR degradation is a small drawback, yet, the premium class flows from a broken link are sustained, which is the obvious advantage. Additionally, as described in Section 5.3.1, to avoid FR reduction, the notion of dropping currently active flows of the lowest priority might be used. However, provided that the amount of the prioritized traffic is kept in reasonable boundaries, the FR degradation extent is acceptable and FR recovers relatively quickly.

5.4 Differentiated queuing

Applying differentiation blocking is one way to provide better service differentiation possibilities to FAN networks. This approach emerges from the necessity to alleviate the long waiting times phenomenon. Although FAN provides basic service level guarantees by prioritizing low-rate flows and keeping fair rate sufficiently high for the rest of the flows, this scheme can also be extended. However, altering the scheduling mechanisms to provide service differentiation is strictly related to resignation from the FAN main capability to provide fairness, i.e., to ascertain that each admitted flow may emit at the same bit rate.

In this section, two main methods of applying differentiated queuing are presented. One of them aims at assuring more or less than the current fair rate to a certain class of flows, e.g. a priority flow might utilize twice as much bandwidth as any other normal class flow. The second approach forces scheduling algorithms to treat packets of a certain flow as if they emit at a rate lower than the current fair rate, even though they emit faster. In other words, packets would be forwarded through priority queues, even if they transmit with a rate greater than the current fair rate. Both approaches are described with the required changes to PFQ and PDRR queuing algorithms, as they are both capable of implementing these features. It is worth mentioning that differentiated queuing aims at improving or degrading the transmission quality of certain flows, once they are admitted. Unless the differentiated blocking approach is used, the admission control block treats each flow equally. Therefore, in terms of congestion, flows are blocked regardless of their differentiated queuing class.

5.4.1 Bitrate differentiation

Bit rate differentiation improves or degrades bitrates of certain flows under the terms of congestion. This method is not able to provide certain bandwidth assurances. Instead, better or worse QoS may be imposed only with respect to the normal class of flows. However, if we consider the minimum level of service assured by the minimum fair rate value, we can easily provide, e.g., twice the minimum fair rate, or half of the minimum fair rate to certain flows.

To realize bitrate differentiation, parameter *differentiation factor* needs to be introduced. The differentiation factor represents the portion of the FR which is provided to a flow. For example, differentiation factor of 2 means that twice the bitrate of the fair rate is provided to a flow.

Figure 5.12 presents the pseudocodes' fragments of the PFQ and PDRR queuing algorithms, that are to be changed, so that these schedulers could realize bit rate differentiation. For the full pseudocode listings of PFQ and PDRR operations, see Section 3.6 on page 32.

```
11    (...)
12    $flow\_time\_stamp(F) += L$
13    (...)
```

(a)

```
11    (...)
12    get head of $AFL$, say flow $i$
13    $DC\_i += Q\_i$
14    (...)
```

(b)

Figure 5.12: PFQ (a) and PDRR (b) pseudocodes' fragments to be changed to provide bit rate differentiation

The PFQ algorithm organizes its queue by inserting new packets in a proper place. Each backlogged flow is described by certain variables, one of which is *flow time stamp*. This indicator describes the time in which the last packet of this flow will be transmitted. Normally, when a new packet is inserted into the queue, this variable is increased by the packet's length (L) (Figure 5.12 (a), line 12). Such a functionality provides fairness. In order to support bit rate differentiation, *flow time stamp* must be increased by values different than the incoming packet length, e.g., by its fraction. For instance, to achieve the bit rate twice as high as the fair rate, only $L/2$ should be added, while to achieve three times less than the fair rate, *flow time stamp* should be increased by as much as $3L$.

Modifying PDRR is more straightforward and simpler, as this algorithm itself was designed to provide the differentiation. In each cycle, the deficit counter of every flow (*DC_i*) is incremented by a proper value, referred to as quantum (*Q_i*) (Figure 5.12 (b), line 13). PDRR in FAN is supposed to provide fairness, therefore, the quantum variable is equal for every flow. However, the algorithm is capable of using different quanta. The more a certain flow receives, the more bandwidth will it be able to consume. For example, incrementing the deficit counter with 2 quanta instead of 1 results in achieving the bit rate twice as high as the current fair rate.

The idea of bit rate differentiating with respect to the currently realized fair rate is interesting, due to the FAN admission control functionality. In a classical IP network the assurance of achieving twice the current fair rate would not be of a great value, as on a heavily congested link, realizing, e.g. 0.2 kbit/s instead of 0.1 kbit/s is still unsatisfactory. Fortunately, FAN preserves the minimum fair rate threshold, therefore, ascertaining more than the current fair rate results in keeping the prioritized flows on better than the rest, and always reasonable, level of the QoS.

5.4.2 Fair rate ignoring

While the bit rate differentiation is probably sufficient for introducing differentiated queuing, the fair rate ignoring scheme aims at achieving the same goals, yet differently. As described in Section 3.6 on page 32, the SFQ and DRR algorithms were enhanced to be suited for FAN by implementing priority mechanisms, to support the better treating of streaming applications. These mechanisms are based on priority processing of flows which emit at a lower rate than the current fair rate. The fair rate ignoring scheme forces the queuing algorithms to treat a certain flow with priority even if it transmits faster than the current fair rate. Figure 5.13 presents the pseudocodes' fragments of PFQ and PDRR queuing algorithms that must be changed, so that these schedulers could realize the fair rate ignoring.

The fair rate ignoring procedure is based on not taking into account the FR measurements for certain flows. Fragments of codes presented in part (a) and (b) of Figure 5.13 concern PFQ and PDRR, respectively, but are identical in functionality. First, a proper condition is checked and based on this result the packet is either prioritized or not.

Both algorithms compare the number of the transmitted bytes in an active cycle (*bytes* in PFQ and *ByteCount_i* in PDRR) with the maximum number

```
4     (...)
5     if bytes ≥ MTU
6         push {packet, flow_time_stamp} to PIFO
7     else begin
8         push {packet, virtual_time} to PIFO behind P; update P
9     (...)
```

(a)

```
13    (...)
14    if ByteCount_i ≤ Q_i
15        Enqueue(PQ, P)
16        else
17            Enqueue(Queue_i, P)
18    end
```

(b)

Figure 5.13: PFQ (a) and PDRR (b) pseudocodes' fragments to be changed to provide fair rate ignoring

of bytes that may be transmitted in a single cycle (*MTU* in PFQ, and *Q_i* in PDRR). If less than possible bytes were transmitted the packet is prioritized, i.e., inserted at the head of the PIFO queue in PFQ (Figure 5.13 (a), line 8), or in case of PDRR, it is forwarded through the priority queue (Figure 5.13 (b), line 15). If the packet is not to be prioritized, PFQ inserts it to the queue according to its flow time stamp (Figure 5.13 (a), line 6), while PDRR forwards it to its own queue (Figure 5.13 (b), line 17).

In order to introduce the fair rate ignoring scheme, the comparisons between already transmitted and maximum possible bytes need to be changed. Analogously as in case of the bit rate differentiation, the variables: *MTU* in PFQ and *Q_i* in PDRR may be increased or reduced. Greater values assure that more bytes from a certain flow may be prioritized in an algorithm cycle and, therefore, even high bit rate flows may experience the lowest possible packet latency and jitter.

Both methods of providing differentiated queuing are simple to implement, however, they pose some concerns. When a number of prioritized flows appear on a link and they consume more bandwidth than normal class flows, the measured fair rate degrades. This happens, as FR is the estimation of the rates currently realized by backlogged flows and since some of them utilize more bandwidth than they should, other flows have less resources to share. Additionally, an excessive utilization of priority queues by the fair rate ignoring scheme may cause standard streaming flows to observe a greater latency or jitter.

5.4.3 Feasibility study

This section shows that by applying the differentiation factor for certain flows, the total number of active flows in the XP router changes. This is a natural consequence of the fact that we allow some flows to achieve greater or lower bit rates than the current fair rate. A flow with the differentiation factor of 2 is able to consume twice the fair rate at any time and can, therefore, take place of two regular flows.

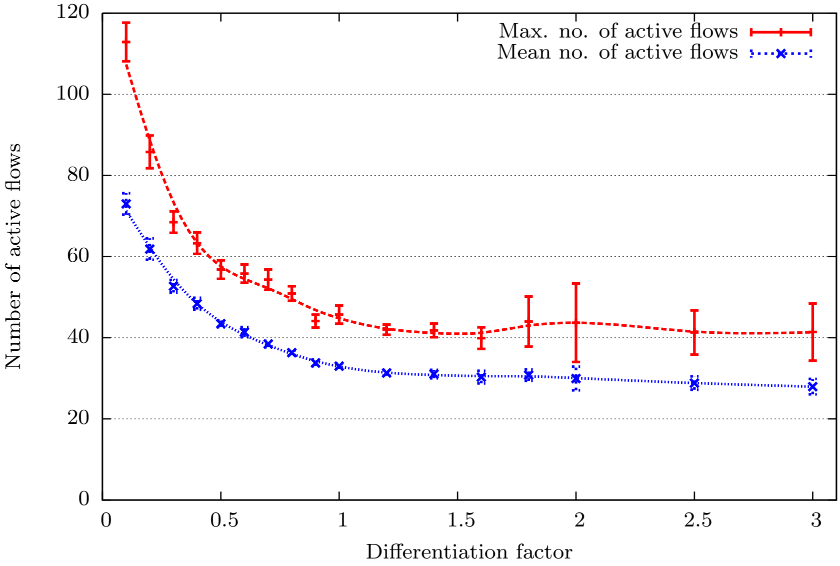


Figure 5.14: The number of active flows with respect to the differentiation factor

Figure 5.14 illustrates this situation, as it shows the number of active flows with respect to the differentiation factor. In this scenario, 300 TCP flows, having on average 2.5 MB of data to send (Pareto distribution with the shape factor of 1.5) start the transmission following the exponential distribution with the mean value of 0.3 seconds. The link capacity is 5 Mbit/s and the minimum FR value is set to 5% of the link capacity, i.e., to 250 kbit/s. The differentiated factor of 1 means that all flows receive the same treatment. In other cases, approximately half of the flows are differentiated with the corresponding differentiation factor.

As can be seen in Figure 5.14, the number of active flows (both the mean number and the maximum number) rises when the differentiation factor is smaller than 1, and decreases when it is greater than 1. The operator must be aware that admitting traffic with various differentiation factors may change the number of active flows, however, the minimum FR value is still preserved by the admission control block and the computation process of the fair rate is unaffected by the differentiated queuing mechanism.

5.4.4 Usage cases

Differentiated queuing has many possible realizations. The idea is that we can assure more or less than a current fair rate. Since FR changes dynamically, depending on the volume of traffic that is carried in the link, providing, say, twice the FR for certain flows might seem as not so great assurance. However, in FAN, FR is not allowed to drop below a certain threshold, and therefore, the assurance of twice the FR, is really the assurance of twice the minimum FR value in the worst case.

The differentiated queuing scheme might be offered to anyone who wants better treatment of his/her traffic in the network, particularly for:

- video conferencing,
- Virtual Private Networks,
- premium customers, etc.

These examples show the instances in which flows would benefit from being provided with better performance. As presented in Section 5.1, VoIP flows do not need more bandwidth, as the bitrate associated with a single flow is, typically, far below the minimum FR value, and is, therefore, always assured. However, video conferences consume much more bitrate, especially those with high video quality. For those applications, the minimum FR threshold might not be sufficient. In such a case, a video conferencing application might be provided with the differentiation factor greater than 1, depending on the requirements and the network link capacities.

Similarly, a Virtual Private Network (VPN) might be established. A consumer might request that his VPN traffic can utilize as much bandwidth as it is available at the moment, however, during congestion periods, the bitrate is not allowed to fall down below, say 5 Mbit/s. To achieve that, an operator can set the differentiation factor on each link in the VPN network such that the following formula is met:

$$differentiation\_factor \cdot minFR = 5 \text{ Mbit/s} \tag{5.1}$$

Figure 5.15 explains how this service differentiation scheme works in practice, as it shows the bitrate obtained by a flow exemplifying a VPN connection to which the differentiation factor was set to 2. There are 300 TCP flows, having on

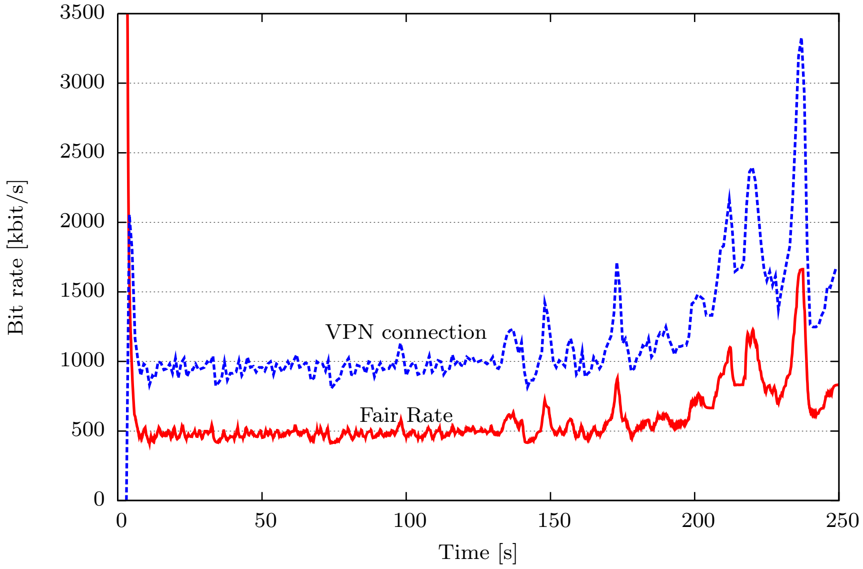


Figure 5.15: Differentiated queuing in practice

average 1 MB of data to send (Pareto distribution with the shape factor of 1.5), and starting following the exponential distribution with the mean value of 0.3 seconds. Also, there is one TCP flow with the preferential treatment: its bitrate is doubled. The link capacity is 10 Mbit/s and the minimum FR value is set to 5% of the link capacity, i.e., to 500 kbit/s. This means, that when the current fair rate drops below 500 kbit/s, new elastic flows are not admitted on the link.

As seen in Figure 5.15, the VPN connection obtains exactly twice the current fair rate. When the link is congested, FR oscillates around the minimum FR threshold (500 kbit/s), and therefore, the VPN connection is guaranteed at least twice the minimum fair rate bitrate. However, when the congestion ends and the current FR rises (after 150 seconds of the simulation) the bitrate obtained by the VPN connection also rises. This means that the VPN connection is always able to consume twice the current fair rate, irrespectively of the actual value of this parameter.

5.5 Static Router Configuration

The proposed mechanisms of explicit service differentiation are easy to implement, does not require any new functionalities and hardly complicate the existing ones. However, the signaling remains an important issue. It is very difficult to inform the nodes which flows should be discriminated, without reducing the scalability of the architecture. Implicit service differentiation works well in FAN because it does not rely on any network signaling. Flows are prioritized or discriminated based on their performance which is internally measured by proper XP mechanisms. However, to implement differentiated blocking, routers must be somehow informed which flows should be treated differently.

The IntServ and DiffServ experiences have shown that introducing explicit service differentiation is difficult, due to the signaling problems and the required inter-domain agreements. Therefore, it seems that it is impossible to introduce differentiated blocking into FAN networks globally. However, for a limited scope, the explicit service differentiation procedures may be used in FAN. To achieve that I propose the Static Router Configuration approach.

Static Router Configuration (SRC) is a strategy of manually defining classes of flows and their treatment by network administrators. This approach, obviously, cannot be used globally, yet it is the easiest way to provide explicit service differentiation without any network complication or modification. SRC seems to be an adequate and simplest solution for introducing differentiated blocking to FAN networks.

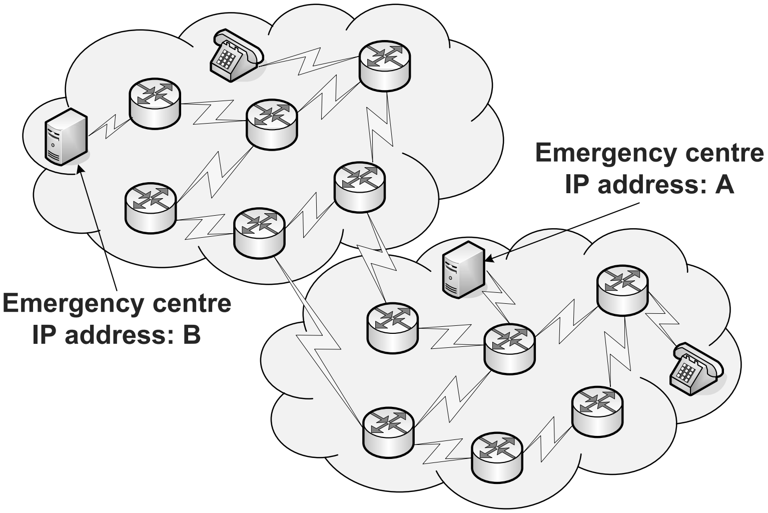


Figure 5.16: Emergency connections scope

It is particularly suited for emergency calls. Because emergency calling is a local matter (always to the nearest emergency center), the SRC approach may be used. An emergency center is responsible for a certain geographical region.

For the differentiated blocking scheme to be used, all nodes in the region must recognize and prioritize flows with the source or destination IP address equal to the address of the proper emergency center. Provided that the emergency center's IP address is static (does not change over time), all routers in the region must be configured only once.

The SRC strategy is the only solution that does not interfere with FAN's superior scalability. Obviously, this approach is not sufficient for many services, however, it is perfectly suited for VoIP emergency connections. Moreover, with SRC, the differentiated blocking scheme may be used for any other local scope service.

When global services are required, and the SRC scheme cannot be used, there is also an option of external signaling protocol. Although IntServ's experience with the RSVP protocol showed that such an approach is highly unscalable, the mentioned configuration signaling protocol for FAN could be quite different. It is different mainly because its operation is not associated with each single flow. Once a node is configured to treat certain group of flows with priority, the signaling protocol is not needed, unless a change is required. Considering the limited required functionality of a signaling protocol, this might be a real alternative to SRC for global services.

5.6 Class of Service on Demand

Arming FAN with differentiated blocking and differentiated queuing greatly increases the service differentiation capabilities of this QoS architecture. Unfortunately, as was explained, the issue related to signaling still remains. We can either stick to the local nature of the traffic and use the SRC approach, or we can apply a simple signaling protocol to inform the nodes of the preferential treatment for certain flows.

However, there is a third option. In this section, I propose using the Class of Service of Demand method in FAN, the method which combines both differentiated blocking and differentiated queuing. Here, a user decides to which class of service his/her packets belongs. There are many possibilities on how to transmit and realize an end-user class selection. The easiest one would be to set a certain value in the IP packet headers, e.g., to use the ToS field in IPv4 (also known as DSCP field) or flow label in IPv6.

The most important issue in this approach is the proper design of the classes. Classes should be designed in such a way that one class is not generally better than the other. For example, if two classes were proposed as in Figure 5.6 and each user is able to choose freely between them, everyone is bound to use the premium class, just because it is better. To make the scheme reasonable, the *quid pro quo*[7] approach must be applied. Classes should be designed for certain applications, but no class should be objectively better than the other.

One possible realization of Class of Service on Demand in FAN is as follows. We provide two classes of service:

1. elastic: admission controlled by MBAC, unlimited bitrate

2. streaming: no admission control, bitrate limited to 50 kbit/s

Additionally, due to the possibility of malicious behavior, the number of streaming flows must be limited for any pair of source-destination addresses. The purpose of this limitation is that an end-user may create many flows and he/she could use the streaming class with all its benefits and not care about the bitrate limitation.

The trick is how to efficiently impose the 50 kbit/s bitrate limitation to flows, given that algorithms such as PFQ and PDRR do not provide such functionality. Although in PFQ and PDRR, it is not possible to set strict bitrate limits, as shown in Section 5.4, we can provide better or worse treatment with respect to the current fair rate. In other words, it is possible to limit the bitrate of a flow to a certain amount of the current fair rate.

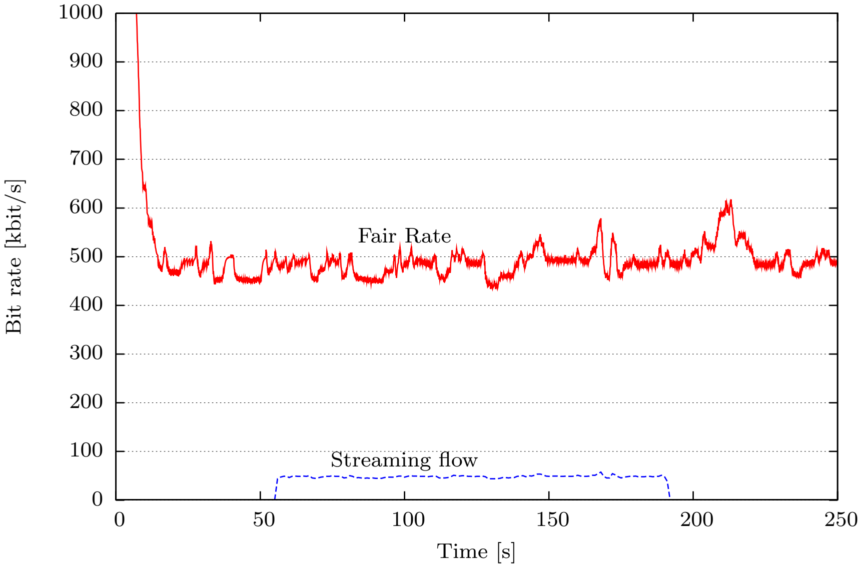


Figure 5.17: Streaming flow's achieved bitrate

[7] from the Latin meaning "this for that"

Figure 5.17 presents the feasibility of the presented scenario. There are 300 TCP flows, having on average 2.5 MB to send (Pareto distribution with the shape factor of 1.5), and starting following the exponential distribution with the mean value of 0.3 seconds. Also, there is one streaming flow: UDP transmission, constant bitrate of 200 kbit/s which starts the connection in 55th second of the simulation. The link capacity is 10 Mbit/s and the minimum FR value is set to 5% of the link capacity, i.e., to 500 kbit/s. This means, that when the current fair rate drops below 500 kbit/s, new elastic flows are not admitted. Streaming flows, however, are never blocked. To achieve the limit of 50 kbit/s for streaming flows, the differentiated queuing mechanism is set so that they can realize up to 10% of the current FR. When the link is congested, this effectively restricts the bitrate of streaming flows to roughly 50 kbit/s. When the link is not congested the limit also applies, however, since FR is greater than its minimum threshold, the streaming flows can also obtain a greater bitrate. In Figure 5.17 we can see that the streaming flow is admitted on the link instantaneously and it transmits with roughly 50 kbit/s bitrate even though its desired speed is set to 200 kbit/s.

There is plenty of potential configurations of the Class of Service on Demand in FAN. In this section I presented only one possible realization. The most important benefit of this approach is that it does not need any kind of signaling to operate. Obviously this method can be combined with SRC to provide even greater service differentiation. For example, to the presented scenario, one might add the possibility to protect emergency connections and prioritize traffic related to virtual private networks. In this way, FAN's service differentiation offer is significantly enriched.

5.7 Service differentiation and network neutrality

It was shown in Section 3.8 and [30] that FAN is a QoS architecture which provides service differentiation in a neutral way. The neutrality comes from the fact that differentiation in FAN is performed based on each flow's current bitrate, not taking into account flow's source, destination or the application that generates it. The extensions to the architecture proposed in this chapter improve the service differentiation capabilities of FAN routers, and therefore, their conformance to net neutrality rules must be discussed separately.

Differentiated blocking is a mechanism which allows for applying different blocking criteria to different new flows. Therefore, the net neutrality conformance depends on the usage patterns of this mechanism. The operator is able to violate the net neutrality principles by providing better blocking criteria for certain flows, e.g., for applications from which the operator generates revenue. The method, however, was designed to provide service differentiation equally. The intended usage is twofold: 1) to provide different blocking criteria for different kinds of traffic, 2) to provide better admission chances of the emergency services. The former usage originates from one of the net neutrality principles, i.e., an operator is able to provide service differentiation, yet it has to be applied to all the applications related to a certain service. In other words, if an operator wants to provide better blocking criteria for the VoIP flows, all VoIP flows (from all the applications) must be treated the same way. The latter usage, i.e., providing emergency services, does not violate the net neutrality principle, as both sides of the debate agree that network operators can prioritize emergency services. Therefore, although giving priorities based on the traffic origin or destination is against net neutrality, in case of emergency services, it is allowable.

The differentiated queuing mechanisms observe a similar story. When they are applied to network management, emergency services or other life-saving actions, their use is permitted. When they are used to prioritize or deteriorate traffic related to a certain service in general, and not to a certain application, they will be considered as in line with net neutrality. However, the operator might want to use this scheme to prioritize only ceratin traffic (for his/her benefit), and that is, obviously, against the neutral Internet principles.

The proposed scheme of Class of Service on Demand combines the achievements of differentiated blocking and differentiated queuing, however, here, the story is different. The difference is that it is the user who makes the decision to which class his/her flow should belong. The decision has a significant impact on the treatment his/her flow will obtain in the network, yet there is no discrimination. All streaming flows, and all elastic flows observe the same QoS level. This is a perfect example of how service differentiation can be provided in the network in a net neutral way.

5.8 Conclusion

Admission control and scheduling blocks of a FAN's XP router are the key components responsible for improving network performance in case of overload. The active flows may perceive a sufficiently good QoS, if only a certain number of flows is simultaneously admitted on a link. Unfortunately, this mechanism may be dangerous for the Internet telephony, especially for emergency connections.

To overcome the described negative behavior, I proposed the differentiated blocking scheme, and make all flows related to realizing emergency connections unblockable by admission control blocks. To achieve this goal, the Static Router Configuration, as a way to inform all the nodes which flows should be prioritized, is also proposed. Considering significant benefits, along with a reasonably low cost associated with the proposition, I believe that introducing differentiated blocking along with the SRC approach will greatly improve the end-user perception of the FAN architecture. Lastly, it has been evaluated that for the purpose of the Internet telephony, the proposed solutions do not interfere with the overall performance of the architecture significantly.

Differentiated queuing is also possible in FAN. Bitrate differentiation enables FAN networks to provide guarantees on a different level than the minimum fair rate threshold. Moreover, to implement differentiated queueing, only cosmetic alterations to the FAN's queuing disciplines are required.

The proposed mechanisms interfere with the admission control and scheduling blocks of the XP router, possibly resulting in a temporal performance degradation of the carried traffic. This issue was thoroughly documented and proved to be insignificant to the overall performance of the FAN architecture, provided that the amount of prioritized traffic remains within reasonable boundaries. Otherwise, the pre-emption-based methods need to be applied.

Finally, the Class of Service on Demand approach was presented. This scheme utilizes the possibilities that are provided by both differentiated blocking and differentiated queuing. This way the service differentiation possibilities offered by the FAN architecture are greatly enhanced. Moreover, this approach proves that it is possible to provide service differentiation in a net neutral way.

6 Quality of Service assurance mechanisms in Flow-Aware Networks

Stop thinking in terms of limitations and start thinking in terms of possibilities.

— Terry Josephson

To assure a ceratin level of guaranteed bandwidth some admission control procedures must be applied. In FAN, admission control is measurement-based. Moreover, as FAN does not use any kind if signaling, network routers are not aware of the incoming flow characteristics. This fact makes the admission decisions more challenging than in case of, e.g., IntServ supported IP or ATM, where transmission parameters are more or less known *a priori*.

FAN intends to provide a minimum level of resources for each active flow. It does that by blocking new flows when congestion indicators exceed their fixed thresholds. It is assumed that those thresholds define the minimum level of assured service on each FAN link. However, as shown in this chapter, this assumption cannot be made, as when many new flows arrive at the same instant, the thresholds are significantly exceeded. To eliminate the problem, I propose using a limitation mechanism which not only improves QoS assurance capabilities, but also enhances the scalability of the FAN architecture. The aim of the mechanism is to limit the maximum number of new flows that may be admitted on a link between any two consecutive network's auto-measurements. The solution is efficient, viable and dramatically reduces the fair rate degradation, thereby improving the service assurance capabilities of the architecture.

In this chapter I present the following, new mechanisms:

- static limitation mechanism,
- dynamic limitation mechanism,
- predictive approach,
- automatic limitation mechanism.

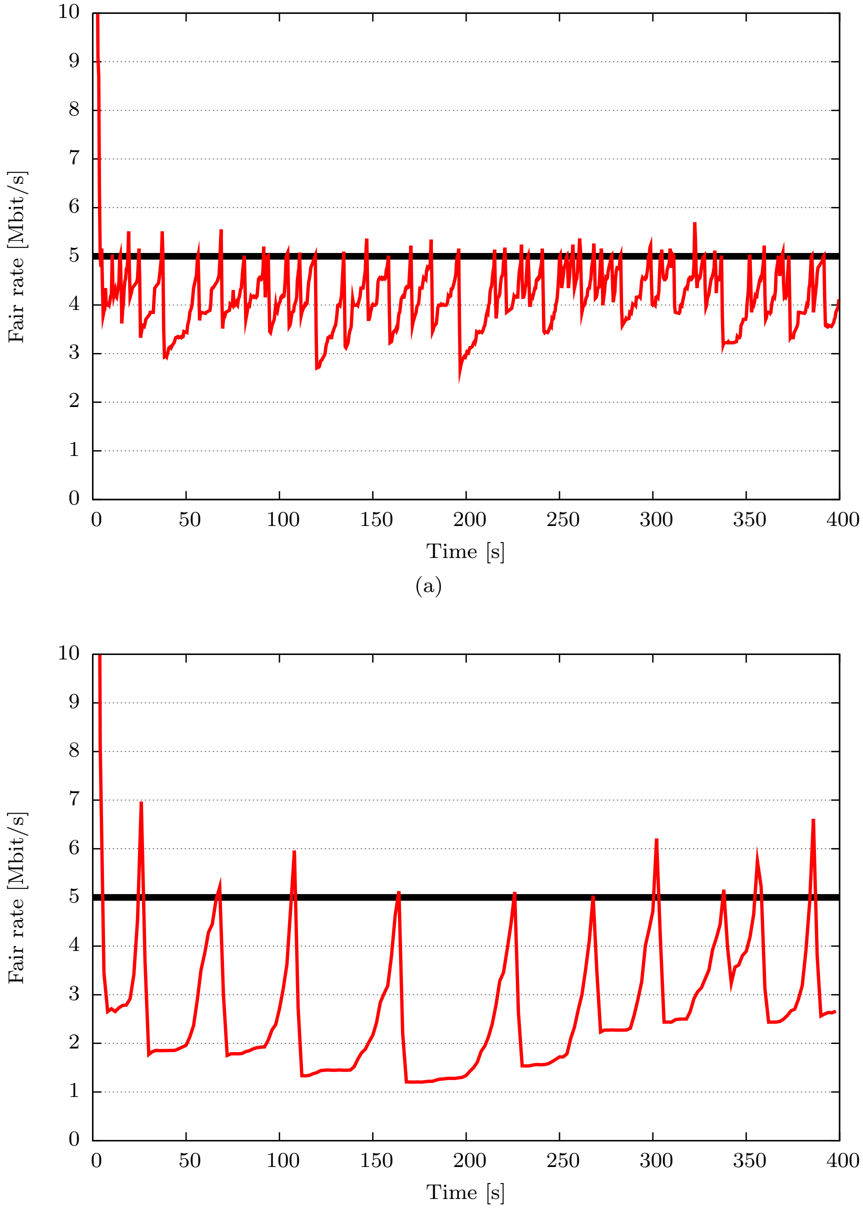
The chapter is organized as follows. I start with Section 6.1 which exposes a fair rate degradation problem of FAN networks, i.e., the inability to ascertain assumed QoS when the number of incoming connections is significant. Subsequent sections provide solutions to the presented problem. Section 6.2 shows the easiest, yet very efficient approach to mitigate the problem, i.e., the static limitation mechanism. Section 6.3 shows an enhancement to the static limitations and proves its superiority in certain cases. Section 6.4 proposes a different approach to the realization of the admission control block in FAN networks, namely the predictive approach. The automatic mechanism which facilitates the limit choosing process is proposed in Section 6.5. The proposed mechanisms are discussed with relation to the network neutrality principles in Section 6.6. Finally, Section 6.7 concludes the chapter.

6.1 Fair rate Degradation

The occurrence of fair rate degradations were presented in Section 5.3.1 as a consequence of admitting priority flows under the conditions in which a regular flow would not have been accepted. This section shows that FR degradations also happen as a natural effect of the admission control routine designed for FAN. The XP mechanism, in FAN, is supposed to provide at least a minimum fair transmission rate to all the active flows. To achieve that, each time the measured FR drops below the minFR threshold, the admission control starts blocking all new connections. Therefore, in fact, this procedure does not guarantee to maintain the minFR value under congestion since proper actions are undertaken only after the minFR boundary is crossed. A similar situation concerns the second congestion parameter, i.e., the priority load.

In theory, the fair rate should be allowed to drop below the threshold only slightly before the admission control block starts functioning. Unfortunately, in practice, the FR drops might be significant.

Figure 6.1 demonstrates the problem as it shows the measured fair rate values over time on severely congested FAN links with 1000 flows arriving with the intensity of, on average, 5 flows per second. In this scenario, a 100 Mbit/s FAN link was analyzed. The minFR parameter was set to 5% of the link capacity (5 Mbit/s) and was measured every 0.4 second in (a) and every 2 seconds in (b).



(a)

(b)

Figure 6.1: Measured FR values over time on a congested FAN link: FR measured once every (a) 0.4 s, (b) 2 s.

The volume of traffic to be sent by each flow was generated following the Pareto distribution (15 MB on average, shape factor: 1.5). The exponential distribution for generating the time intervals between the beginnings of the transmissions of the flows was used. The duration of each simulation run was set to 400 s.

As can be seen, in both cases presented in Figure 6.1, FR drops well below the minFR threshold (5% of the link capacity, marked with solid flat lines). In case (a) the degradations are shorter and reach up to 2 Mbit/s, whereas in case (b) degradations are much longer and more intense (even up to 4 Mbit/s). Such situations occur because between two consecutive FR measurements many new flows arrive and are admitted before the router realizes that the admission control should be in the blocking state. The ever repeating routine shown in Figure 6.1 comprises the following four steps:

1. FR drops below minFR threshold and no new flows are admitted,
2. existing flows naturally end their transmission and FR slowly rises,
3. FR rises above minFR,
4. the admission control block starts accepting all new flows until FR drops below minFR.

The unfortunate behavior is a consequence of the step number 4. As the admission control block relies on the data delivered by the scheduling blocks, and the fact that the scheduler performs measurements periodically, only after the next measurement can the admission control block start to block new flows. Since the frequency of measurements directly contributes to the extent and the duration of degradations, it can be easily explained why FR degradations are less significant in case (a) of Figure 6.1 than in case (b). Nevertheless, in both cases, these FR drops are unwanted as they are dangerous to streaming applications which require a certain available bandwidth. The whole concept of FAN is that this bandwidth (minFR) can be provided for such flows. However, Figure 6.1 shows that FAN fails in providing this key quality.

To understand how the length of the FR measurement interval impacts the FR degradation process, several simulations were performed. The scenario was similar as before, with only the following differences: the length of the FR measurement interval varied from 0.2 second to 3 seconds. Three sets of experiments were performed with the number of active flows equal to 1000, 2000 and 3000. The number of active flows was related with the intensity of their arrival by the following formula:

$$\frac{N}{\lambda} = T = const \tag{6.1}$$

where: N is the number of active flows, λ is the intensity of their arrival and T is the simulation time. The intensity was set in the simulator by changing the mean interval between the beginnings of the transmissions of new flows, according to the following formula:

$$\lambda = \frac{1}{t_e} \tag{6.2}$$

where t_e is the interval to the beginning of the next flow's transmission obtained from the exponential distribution. Several simulations were performed for each case, to calculate the 95% confidence intervals using the Student's t-distribution.

To present the problem numerically, a mean deviation from the minFR threshold was defined as follows:

$$\frac{1}{n}\sum_{i=1}^{n}\frac{|minFR - FR_i|}{minFR} \cdot 100\% \tag{6.3}$$

where FR_i are the measured FR values over time. This parameter shows how much the measured FR values differ from the minFR during the total measurement time (simulation time). As, in all cases, we simulate only the overloaded links, the ideal FR values should oscillate around the threshold and the deviation should be very low.

Figure 6.2 shows how the length of the FR measurement interval impacts the mean deviation from the minFR, as defined by formula 6.3. Two trends can be observed: the deviation from the minFR grows with the increased FR measurement interval and with the number of active flows. The reason behind the first trend was explained earlier. When the duration between two consecutive measurement is longer, statistically more flows will appear and be admitted, which results in significant over-admission. The second trend impacts the degradations in a similar manner, i.e., when there is more active flows, more of them will appear during a certain period of time which increases the over-admission.

The mean deviation from the minFR is the indicator which compares the performance of the system under different setups. For the individual flows, more important is the amount of time during which certain bitrates are not assured. Figure 6.3 shows the amount of time in which FR drops below (a) 90% and (b) 80% of the minimum FR from the previous experiment. This characteristic is very important for streaming applications which require a certain amount of bandwidth to be available. Being aware of the fact that FR boundaries can be and are constantly crossed, the network administrator might want to set the threshold a bit higher, e.g., to provide a guaranteed level of 5% of the link's bandwidth, a minFR value could be set to, say 7%. However, as observed in Figure 6.3, this approach may be deceiving, as FR degradations are uncontrollable and unforeseeable.

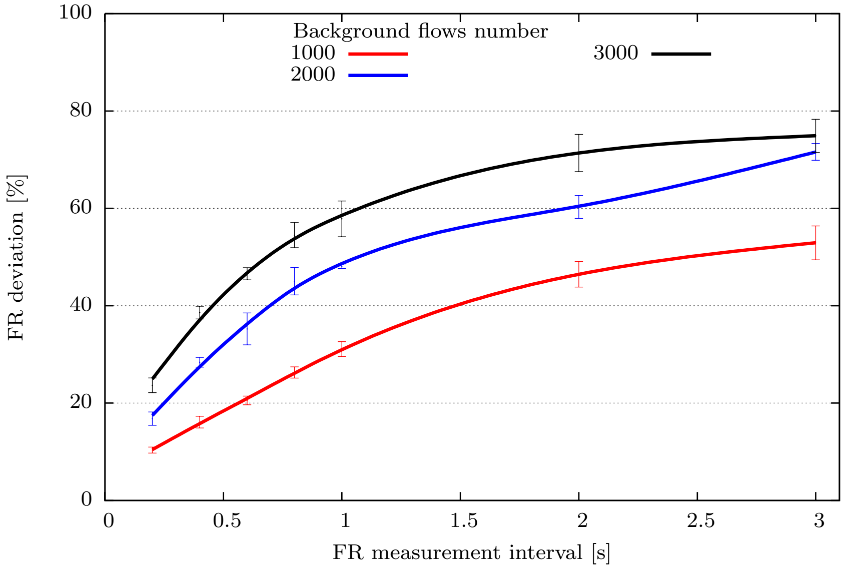
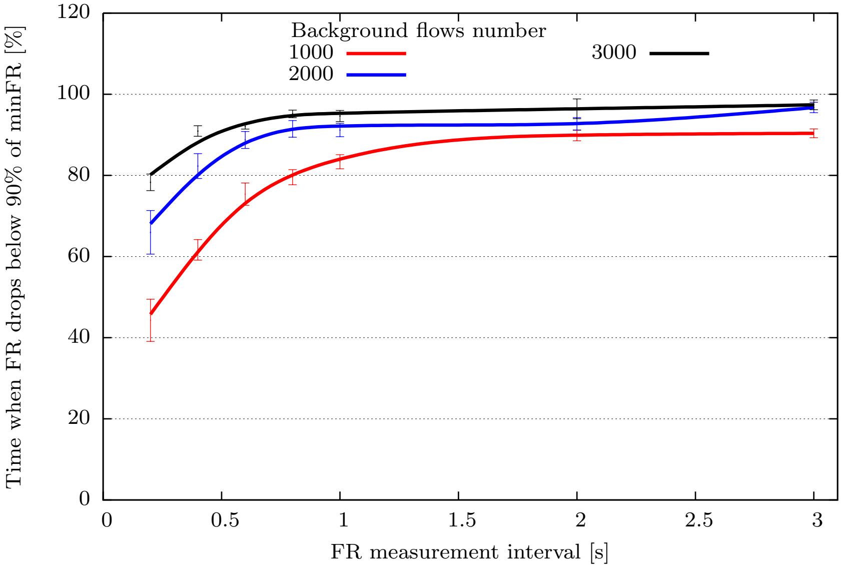


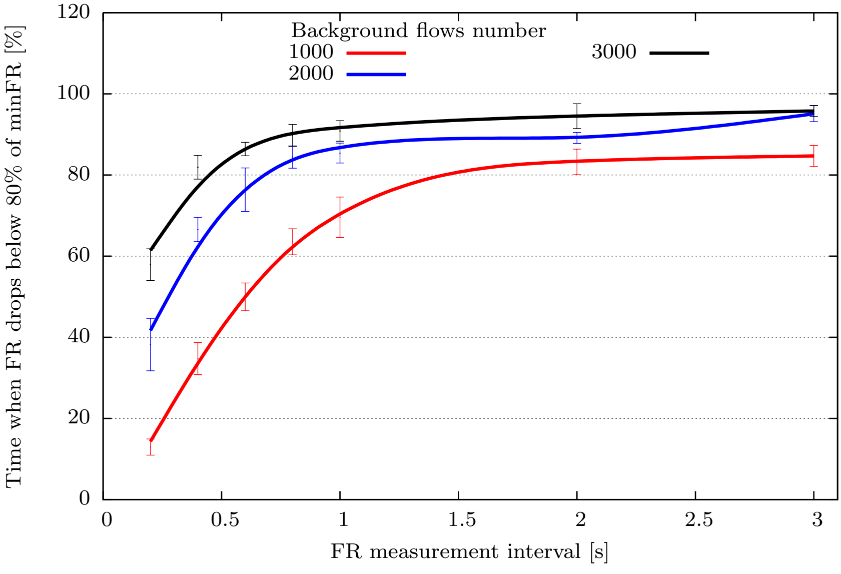
Figure 6.2: FR deviation from minimum FR with respect to FR measurement interval

It can be observed that the amount of time in which FR drops below a certain value is strictly correlated with the mean deviation from the minFR described previously. The longer the interval between the measurements, the more substantial is the time during which FR drops significantly. The values in Figure 6.3 are to be read as follows: if time when FR drops below 80% of the minFR is equal to 50%, it means that half of the time the actual FR is below 80% of the minFR. As an example, consider a 10 Mbit/s FAN link with the minimum FR value set to 5% of the link capacity, i.e., 500 kbit/s. FAN should be able to provide this bitrate to all the flows. However, in the mentioned case, 50% of the time, the actual FR is going to be lower than $80\% \cdot 500$ kbit/s $= 400$ kbit/s. From the absolute values presented in Figure 6.3 it can be seen that minimum FR guarantees have little meaning in plain FAN networks, as under heavy congestion in some cases, more than 90% of the time the actual FR is far below the guaranteed threshold.

There are two approaches to mitigate the problem. One is to reduce the interval between two consecutive measurements of the fair rate. If the FR is estimated more frequently, statistically fewer flows are admitted between the measurements and the system reacts quicker. The downside of this method is that frequent estimations require more computational power from the router's CPU. This issue becomes even more significant in core networks, as those devices deal with numerous flows and must react almost instantly.



(a)



(b)

Figure 6.3: FR drops below (a) 90% and (b) 80% of minFR with respect to FR measurement interval

The FR measurement interval values chosen for this section's experiments were meant to illustrate the problem. In real devices these intervals are bound to be much shorter. It is easy to imagine that routers should be able to provide the measurements once every 0.1 second or even more frequently. However, as seen in Figures 6.2 and 6.3, FR degradations grow with the increased number of active flows. Therefore, for the core devices which deal with numerous flows the problem becomes more significant, up to the point in which further reduction of the measurement interval is no longer an option. At that point the only solution, straightforward, yet viable, is to limit the number of flows that can be admitted between two measurements. We discuss this solution in the following sections.

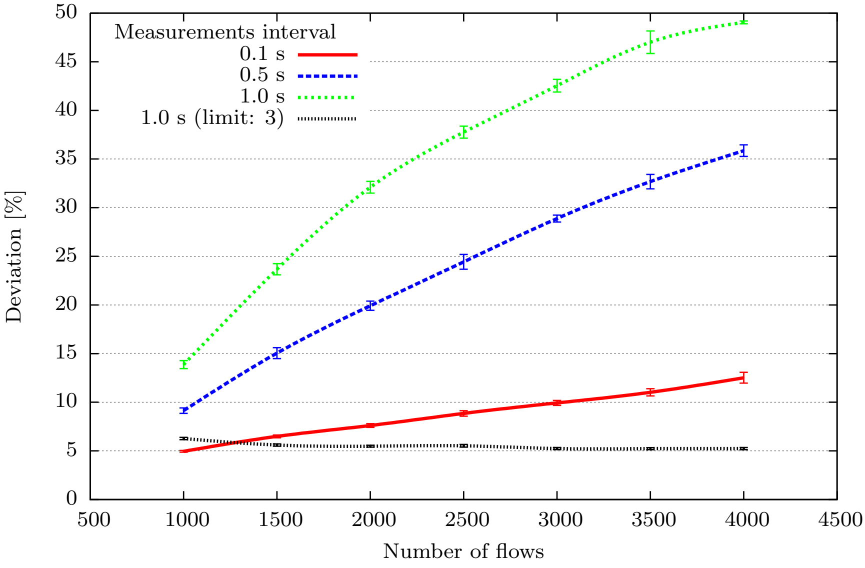
6.2 The limitation mechanism

In the literature, numerous admission control mechanisms have been proposed over the years, mainly for Integrated Services, Differentiated Services, or call admission control procedures in ATM. The PhD dissertation of A. W. Moore [79] contains the detailed comparison of them. However, most of the proposals rely on the fact that at least a limited information about the incoming flow is available through signaling. As FAN does not use any kind of signaling, those methods are not applicable.

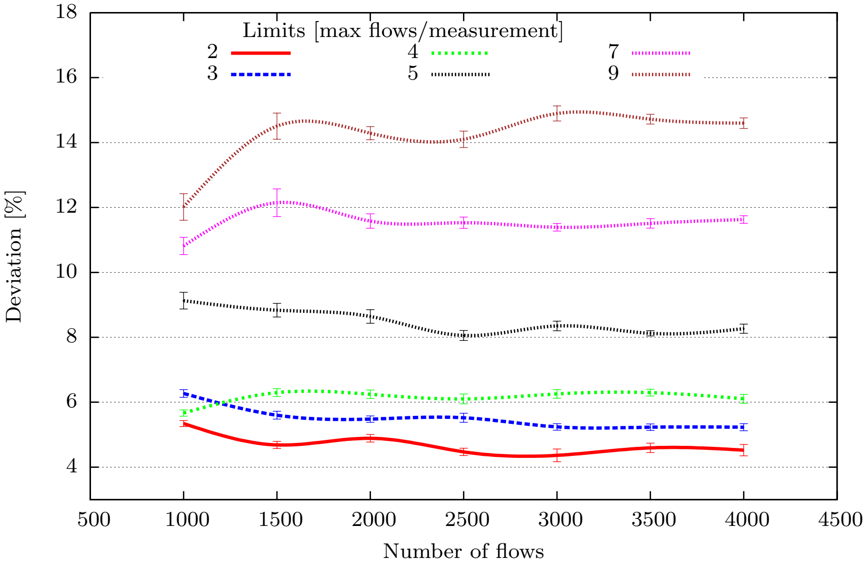
Some of the admission control mechanisms also notice the problem of over-admitting. In [5], it is stated that the system needs to wait for a period of time after any change of the number of connections in progress happens, before the link congestion status can be re-estimated. The author proposes a timescale solution, i.e., to regard a time interval as a function of the number of active flows. As the number grows considerably, the interval is decreased to reduce the probability that a situation in a link changes significantly within that interval. Unfortunately, as already mentioned, increasing the frequency of measurements imposes more strict demands on the router CPUs. Therefore, it is argued and proved in this section, that by providing even the simplest limitation mechanism, we can resolve the over-admitting problem while not increasing the routers' computational power requirements. The results shown in this section have been published in [112].

The limitation mechanism in FAN enhances the functionality of the admission control block. The idea is that between any two consecutive measurements, only a limited, fixed number of new flows may be admitted. This approach protects the admission control block from over-admitting, i.e., from allowing too many new flows to acquire access to the link, which, consequently, degrades the FR.

To provide limitations, we need to introduce only a simple counter, incremented on arrival of each new flow, and reset on each measurement. When it reaches a certain number, all new flows are rejected. This way, the extra CPU power required is hardly noticeable, while the benefits are significant.



(a)



(b)

Figure 6.4: Mean deviation of the measured FR from the minFR threshold with respect to: (a) the measurement interval length, (b) the maximum number of flows accepted in one interval.

Figure 6.4 shows the mean deviation of the measured FR with respect to the number of active flows and different measurement intervals. All the simulation scenario parameters were presented in the previous section. As shown in Figure 6.4(a), when limiting is not applied (three rising curves), the deviation rises along with the number of active flows, and is greater when larger measurement intervals are set. Both dependencies are natural and were explained in details in Section 6.1. The number of active flows, associated with the statistical intensity of their arrival, impacts the number of flows which request the resources every second, while the measurement interval impacts the duration of that arrival. Both factors contribute to the fact that more or fewer flows may be over-admitted. However, when the limitation is used (the flat line, 3 flows per measurement in this case) we observe almost constant deviation, and much smaller than that when limits are not applied. The fact that the deviation does not increase with the number of active flows helps to administer the network, as the operators can keep the links in a proper condition regardless of the current network overload.

The limit of 3 flows per measurement was chosen experimentally. Figure 6.4(b) shows how various limits impact the FR deviation under the same conditions. Choosing a limit too strict (low) results in under-feeding the link, as the link serves flows faster than they can be admitted. Such a case happened when the limit of 1 flow per measurement was applied (not shown in the Figure, as the deviation was way over the presented scale). On the other hand, choosing a high limit does not solve the problem, as the deviations start to rise. In Figure 6.4(b) we can observe that limiting the admission of new flows to 2–4 per measurement is sufficient under these network conditions. The deviation is around 5-6% and is totally acceptable.

Right now, it is only a matter of experiments to pick the right limit. In the presented simulation scenario, picking the limit of 2-4 flows per measurement is adequate, however, in other situations, especially on links with a different capacity, it will be different.

Table 6.1 shows the mean percentage of time in which FR drops below 90% (a) and 80% (b) of the minFR. As can be seen in Table 6.1, by introducing limitations (marked rows), we can drastically reduce the FR degradation. When limitations are present, the FR drops below 90% of its minimum threshold 5 to 10 times less than in the comparable situation (measurements once every second). The outcome is even more convincing in the second case, as the FR value hardly ever drops below the 80% of the minFR threshold, which is a firm result. Similarly, as in case of the deviation, this characteristic is almost independent of the number of flows when limitations are applied.

For the purpose of comparison, Table 6.1 also shows the times when we increase the frequency of measurements. They show that reducing the inter-measurement time even 10 times does not provide better performance than in-

Table 6.1: The percentage of time in which FR drops below 90% (a) and 80% (b) of the minFR threshold

Measurement interval	Number of flows						
	1000	1500	2000	2500	3000	3500	4000
90% (a)							
0.1	10.96 ± 0.92	25.18 ± 1.39	34.01 ± 1.26	41.60 ± 1.02	47.32 ± 1.33	52.59 ± 1.55	58.06 ± 1.85
0.5	34.82 ± 1.26	60.28 ± 1.81	73.49 ± 0.85	80.12 ± 1.08	83.22 ± 0.44	86.73 ± 0.55	87.26 ± 0.32
1.0	53.61 ± 1.74	76.69 ± 0.65	85.00 ± 0.48	87.49 ± 0.33	89.01 ± 0.51	91.69 ± 0.96	92.29 ± 0.19
1.0 (limit: 3)	**13.90 ± 1.11**	**10.49 ± 0.85**	**8.89 ± 0.97**	**9.16 ± 0.84**	**7.97 ± 0.75**	**8.21 ± 0.76**	**7.58 ± 0.94**
80% (b)							
0.1	0.03 ± 0.06	1.18 ± 0.28	3.50 ± 0.71	7.21 ± 0.71	11.56 ± 0.99	15.64 ± 1.98	21.62 ± 2.59
0.5	9.07 ± 1.52	30.32 ± 2.12	47.53 ± 1.78	59.04 ± 2.09	68.40 ± 0.82	74.00 ± 1.50	76.11 ± 0.56
1.0	24.62 ± 1.30	56.46 ± 2.11	72.73 ± 1.19	77.79 ± 0.54	82.42 ± 0.57	85.83 ± 0.93	84.84 ± 0.32
1.0 (limit: 3)	**0.10 ± 0.06**	**0.01 ± 0.03**	**0.01 ± 0.03**	**0.00 ± 0**	**0.00 ± 0**	**0.00 ± 0**	**0.00 ± 0**

troducing a simplest limitation mechanism. This, essentially, proves that increasing the frequency of measurements is a much worse option to mitigate the FR degradation problem than the limitations. Finally, even if it would be possible to provide a proper frequency of measurements, still the dependency on the number of active flows remain and cannot be neglected, whereas the limitation mechanism solves the problem.

6.3 Dynamic limitations

There is plenty of possibilities concerning the actual procedure of how to limit the number of flows in the limitation mechanism. The method presented in Section 6.2 is the simplest, yet very efficient. In this section, I propose to enhance the method by applying dynamic limitations. Dynamic limitations differ from static ones in that the limit is calculated dynamically and changes over time. The idea is that we gain more flexibility and it is easier to adjust to dynamic changes of the link's traffic.

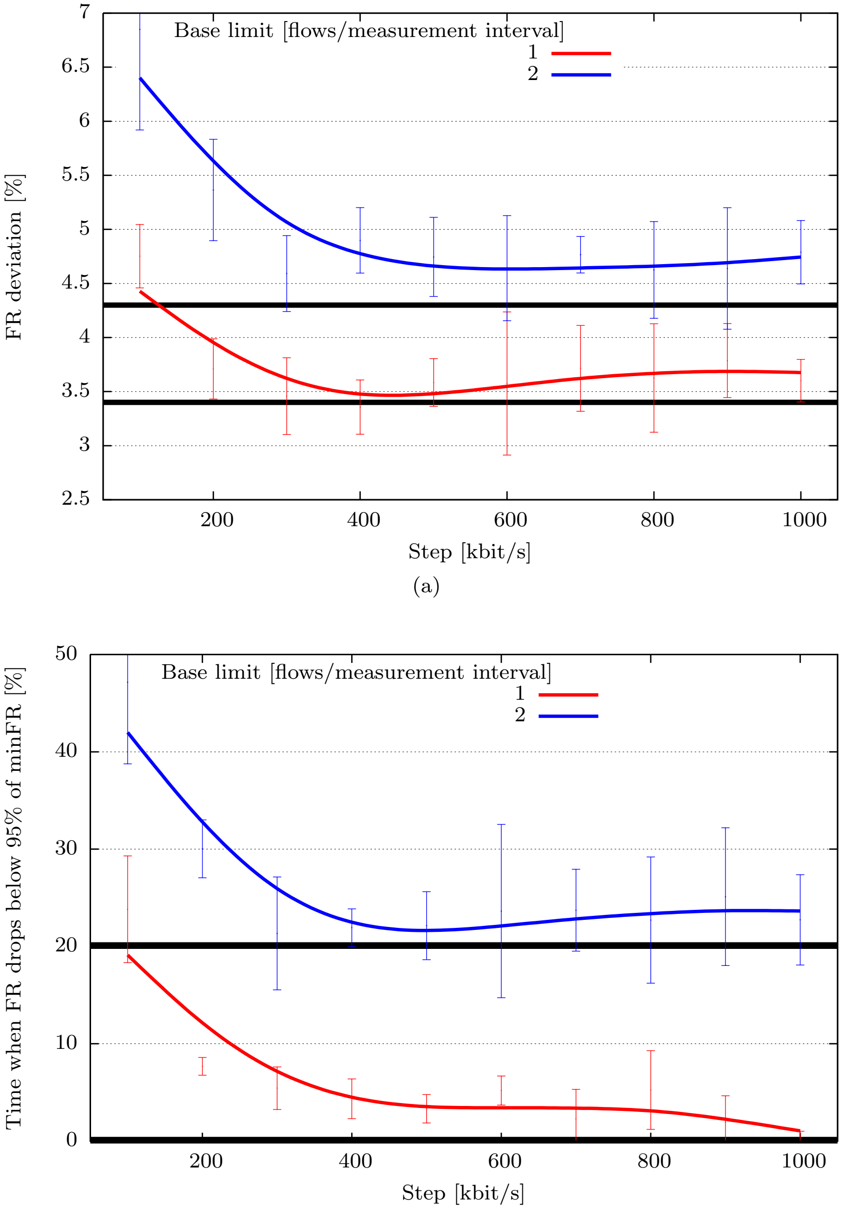
For the realization of the dynamic limit, I propose the graded system. There is a base admission limit, just as in the static limitation mechanism, but this limit is increased by 1 for each *step* the FR is farther from the minFR threshold. The admission limit (AL) in this mechanism is calculated with the following formula:

$$AL = BaseAL + \left\lfloor \frac{FR - minFR}{step} \right\rfloor \tag{6.4}$$

where: AL is the calculated admission limit, $BaseAL$ is the preset base admission limit, FR is the currently measured FR, $minFR$ is minimum FR threshold, *step* is a predefined value which impacts the frequency of changes and $\lfloor x \rfloor$ is the highest integer lower than x. The formula has the following meaning: when $minFR < FR \leq minFR + step$, the admission limit $AL = baseAL$, when $minFR + step < FR \leq minFR + 2 \cdot step$, the admission limit $AL = baseAL + 1$, etc.

This method allows us to use a low base admission limit when the current FR is close to the threshold, thereby better assuring the guaranteed bitrate, and to increase the admission limit when there is more room to do so. The other benefit of this approach is that the system is less prone to the under-admission problem. This issue derives from the fact that in a lightly loaded link, under the static limitation mechanism, we are not able to admit all new incoming flows immediately, even though the system is far from being congested. The dynamic limitation mechanism resolves that problem.

Dynamic limitations in a natural way approach the static limitations as *step* approaches the link capacity. Figure 6.5 illustrates this tendency as it shows the FR deviation and FR drops duration with respect to the *step* parameter.



(a)

(b)

Figure 6.5: FR deviation (a) and FR drops duration (b) with respect to the step parameter

The simulation experiment was identical as presented in Section 6.2. The solid horizontal lines represent the results obtained with the static limitation mechanism under the same network conditions, in which the static limit was equal to *baseAL*. The presented plots can be commented as follows: 1) with both the FR deviation and FR drops duration, a clear tendency approaching the static limits can be observed, 2) the performance of the dynamic limitations is inferior with respect to the static limitation mechanism.

In the presented example, the inferior performance of the dynamic limitation mechanism is caused by the fact that the static limitation of 1 flow per measurement was perfectly sufficient. Therefore, when the possibility of admitting more flows appeared, the performance degraded. True colors of the dynamic limitations, however, can be observed when the static limitation mechanism needs more than 1 flow per measurement to perform adequately. Table 6.2 compares the performance of both the static and the dynamic limitation mechanisms under the same network conditions, only the mean flow size (the amount of data to be sent) is reduced 4 times. The effect of such an action is that much more flows end during a certain time interval, therefore, more new flows may be admitted on the link. Exactly the same effect would have appeared if instead of changing the traffic characteristics, the link capacity was increased four times.

Applying static limitation with the limit of 1 flow per measurement results in severe under-admitting. In this case FR never reaches the minFR threshold, therefore, the assumed warm-up time (until FR reaches the minFR threshold for the first time) does not ever end. Out of the remaining static possibilities, the limit of 3 flows per measurement seems as the best solution: the deviation is relatively low which indicates that there is no problem with under-admitting, however, the FR drops duration is significant. The only better static solution is when the static limit is set to 2, yet the deviation becomes a problem.

The dynamic limitation mechanism has more flexibility. In Table 6.2 (b) the case with the base admission limit of 1 flow per measurement is presented. By adjusting the step factor we can observe much better performance. For example, cases with step set to 200 and 300 kbit/s seem to be the best solution. The deviation is kept within reasonable boundaries, whereas the FR drops duration is almost irrelevant. This example shows that using the dynamic limitation mechanism might be beneficial with respect to the static procedure. The key quality provided by this scheme is that we can use the smallest possible limit of 1 flow per measurement under the network conditions in which such a limit leads to severe under-admitting. Although this limit is not used all the time, its benefits are clearly visible.

The presented method to provide dynamic limitations is one that provides more flexibility to the static limitation mechanism, however, there are plenty of other possibilities. One of them would be to define a certain formula to calculate

Table 6.2: Performance of static and dynamic limitation mechanisms: comparison

Static limitation mechanism (a)						
	Limit [flows/measurement]					
Performance factor	1	2	**3**	4	5	6
Deviation	—	16.69 ± 11.07	**7.49 ± 1.52**	7.49 ± 1.05	7.77 ± 0.42	8.97 ± 0.32
95% drop duration	—	3.00 ± 2.07	**14.67 ± 4.14**	28.25 ± 5.50	42.18 ± 3.79	49.17 ± 3.27
Dynamic limitation mechanism (b)						
	Step [kbit/s]					
Performance factor	100	**200**	**300**	400	500	600
Deviation	6.88 ± 0.77	**7.76 ± 1.38**	**9.39 ± 1.80**	11.13 ± 2.08	13.12 ± 1.70	15.19 ± 1.96
95% drop duration	12.14 ± 0.17	**4.22 ± 3.21**	**1.16 ± 1.18**	0.33 ± 0.61	0	0

the current limit based on the network condition. The problem with this approach is that, in FAN, there is not much information available to be based upon. For the sake of simplicity, the amount of provided information was reduced to minimum. As there is no signaling, flows' transmission characteristics or requirements are unknown, and the router does not keep stateful information about single flows. Furthermore, there is no indicator when a flow ends its transmission. A FAN router erases this flow from the protected flow list only when a certain time from the last forwarded packet elapses. If the flow termination information were available instantly, the limitation mechanism might be altered to intelligently compensate for the no longer active flows.

6.4 Predictive approach

As mentioned in Section 6.1, the root of the FR degradations in FAN lies in the very design of the admission control block. The key issue is the fact that admission criteria rely on the information delivered by the scheduling block which implies passive control. Only after the congestion is noticed, can admission control start to block new flows. Therefore, the minimum level of FR in FAN, is not a guaranteed value, as proper actions happen after this boundary is crossed. The active approach would be to undertake measures even before the congestion occurs.

In this section, I propose FR prediction, an active approach to the realization of the admission control routine in FAN. In this mechanism, the admission control block tries to estimate the value of the next FR measurement and take proper actions based on the predicted FR, rather than on the current real measurements. In such a way, two actions can happen:

1. $FR \geq minFR$ and $expectedFR < minFR \Longrightarrow$ the MBAC block will block new flows despite FR being over the threshold,

2. $FR < minFR$ and $expectedFR \geq minFR \Longrightarrow$ the MBAC block will allow new flows despite FR being below the threshold.

From the viewpoint of service assurance, the first action is more important, as it tries to preserve the minimum guaranteed FR. Therefore, two predictive mechanisms are defined: *half prediction* which utilizes the first action and *double prediction* which uses both. The following formula presents the method of estimating the nearest value of FR:

$$expectedFR = FR_t + p \cdot (FR_t - FR_{t-1}) \tag{6.5}$$

where: $expectedFR$ represents the predicted next value of FR, FR_t is the measured FR in time t and p is the predictor. As FAN is a simple architecture, new mechanisms should not overcomplicate it. To implement the proposed scheme, the XP router needs to additionally remember the previously measured value of FR and the admission control routine needs to be altered, yet with no new functionalities.

Predictor p is a number which tries to emulate the dynamics of the changes in the FAN link. When $p = 1$, the difference between the current FR and the previous FR is calculated and this difference is added to the current FR. This way, the system assumes that the current FR tendency is constant. When the changes are more dynamic, especially on high-capacity links, the use of higher predictors might be more adequate.

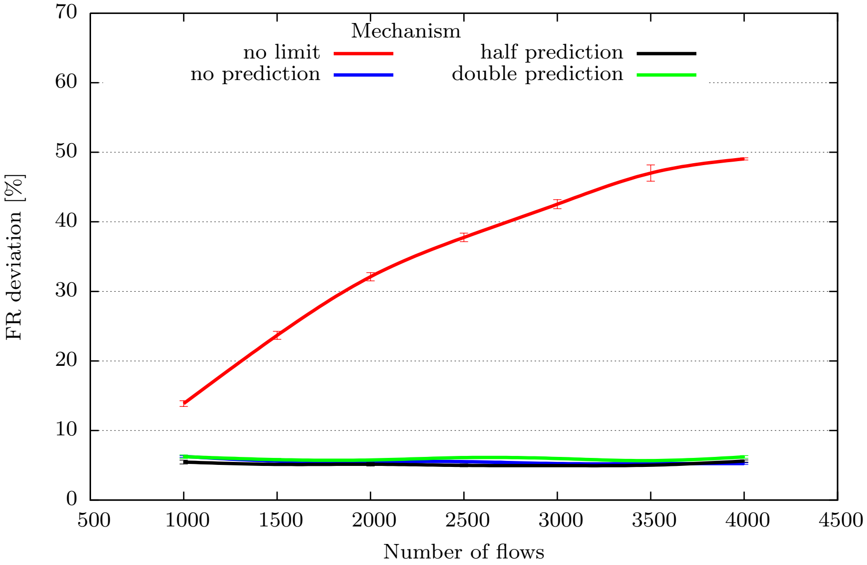


Figure 6.6: FR deviation from minimum FR with respect to the number of active flows

To show the efficiency of the proposed mechanism a number of simulations were performed. The overall scenario setup was presented in Section 6.1. The flow admission limit was set to 3 flows per measurement, and the predictor p was set to 1. Figure 6.6 shows the FR deviation from the minimum FR with respect to the number of active flows when different mechanism are used. As can be observed, the prediction mechanism does not provide significantly lower deviations than standard static limiting mechanism (case: no prediction). It needs to be noted, however, that the deviations observed after the static limitation mechanism is applied are reduced to a completely acceptable level, making it hard to improve any further. The deviations are greatly reduced when compared to the case in which no limiting mechanism is used. Additionally, the deviations are independent of the volume of the carried traffic, represented by the number of active flows.

The deviations remained on the same level as when only static limitations were proposed, however, the amount of time in which FR drops below a certain level can be improved substantially. Table 6.4 shows how often does the measured FR drop below 95%, 90% and 80% of the minimum FR threshold. To compare the efficiency of the proposed mechanisms, the case when no limitations, and the case when static limitation is performed are presented as well. From the numbers in Table 6.4 we can see that the half prediction mechanism outperforms all the other approaches. The time in which FR drops below a certain threshold is shortened by 30-80% compared to the best case with static limitations. Given that static limitations offer drastic reduction of this time compared to the standard FAN routine, the result obtained by the half prediction mechanism must be considered as outstanding.

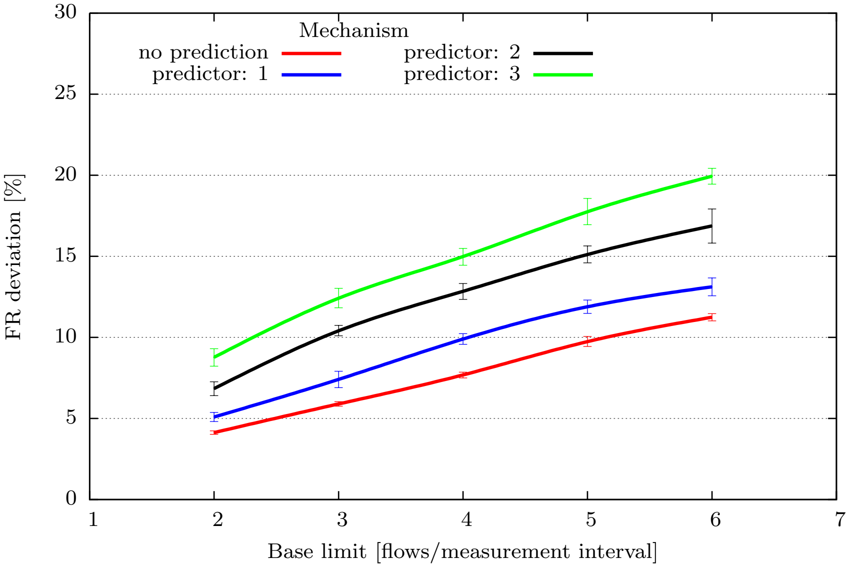
A little bit surprising is the fact that the double prediction mechanism does not provide improvement over static limitations. However, the reason behind such a behavior is twofold. Firstly, as the FR deviation is on a level of a few percent, there is hardly any room for predicting the next values as the FR trend, as well as the over and under the threshold situation changes rapidly. Secondly, the fact that the admission control may admit new flows even when the current FR is below the threshold does not contribute to the reduction of the duration of FR drops.

Similar results are obtained when prediction mechanisms are compared to the static limitation mechanism under three different predictor values. Figures 6.7 and 6.8 show the mean deviation and FR drops duration, as defined in Section 6.1, respectively. The top plot shows the double prediction mechanism, whereas the bottom one presents the half prediction mechanism. As can be observed, under the traffic pattern provided in the simulated scenario, the double prediction mechanism performs better when predictor p is equal to 1. Still, the performance is worse than that obtained with the static limitation mechanism. This tendency is not visible in case of half prediction mechanism. Here, both the FR deviation and the FR drops duration are better than when no predictions are made, however, the relation between various predictors is unnoticeable.

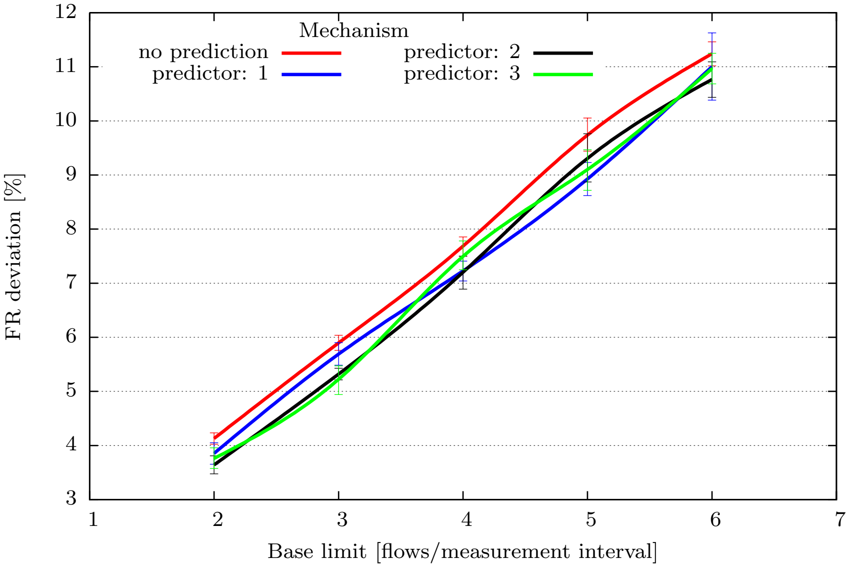
This section shows that the double prediction mechanism does not provide the expected benefits compared to the static limitation mechanism. The half prediction scheme shows superior performance compared to the mechanism which already improves the admission control behavior in FAN. Compared to the original FAN routine, the profits from introducing the half prediction mechanism are substantial. The predictor is a factor which does not seem to have a significant impact on the performance of the half prediction mechanism, however, under dif-

Table 6.3: The percentage of time in which FR drops below 95% (a), 90% (b) and 80% (c) of the minFR threshold

Mechanism	Number of flows						
	1000	1500	2000	2500	3000	3500	4000
95% (a)							
no limitation	68.73 ± 1.61	84.79 ± 0.72	89.30 ± 0.32	93.09 ± 1.54	92.17 ± 0.36	94.35 ± 0.33	94.79 ± 0.12
no prediction	35.92 ± 1.18	35.64 ± 1.29	31.43 ± 1.25	33.46 ± 1.78	33.18 ± 0.97	31.83 ± 1.12	30.91 ± 1.38
half prediction	**23.94 ± 3.08**	**23.00 ± 2.64**	**23.43 ± 4.52**	**25.55 ± 1.34**	**24.52 ± 2.01**	**22.99 ± 2.34**	**24.89 ± 2.03**
double prediction	37.37 ± 3.17	38.00 ± 1.51	36.60 ± 1.79	43.92 ± 1.72	41.05 ± 1.56	40.05 ± 2.38	43.68 ± 2.70
90% (b)							
no limitation	53.61 ± 1.74	76.69 ± 0.65	85.00 ± 0.48	87.49 ± 0.33	89.01 ± 0.51	91.69 ± 0.96	92.29 ± 0.19
no prediction	13.90 ± 1.11	10.49 ± 0.85	8.89 ± 0.97	9.16 ± 0.84	7.97 ± 0.75	8.21 ± 0.76	7.58 ± 0.94
half prediction	**3.97 ± 1.60**	**4.80 ± 0.60**	**5.04 ± 1.67**	**3.36 ± 0.63**	**3.12 ± 1.05**	**4.78 ± 1.25**	**6.11 ± 0.65**
double prediction	13.14 ± 3.31	9.87 ± 1.23	11.88 ± 0.72	13.81 ± 1.35	11.52 ± 1.87	13.58 ± 1.47	17.53 ± 2.29
80% (c)							
no limitation	24.62 ± 1.30	56.46 ± 2.11	72.73 ± 1.19	77.79 ± 0.54	82.42 ± 0.57	85.83 ± 0.93	84.84 ± 0.32
no prediction	0.10 ± 0.06	0.01 ± 0.03	0.01 ± 0.03	0.00 ± 0	0.00 ± 0	0.00 ± 0	0.00 ± 0
half prediction	**0**	**0**	**0**	**0**	**0**	**0**	**0**
double prediction	0	0	0	0	0	0	0

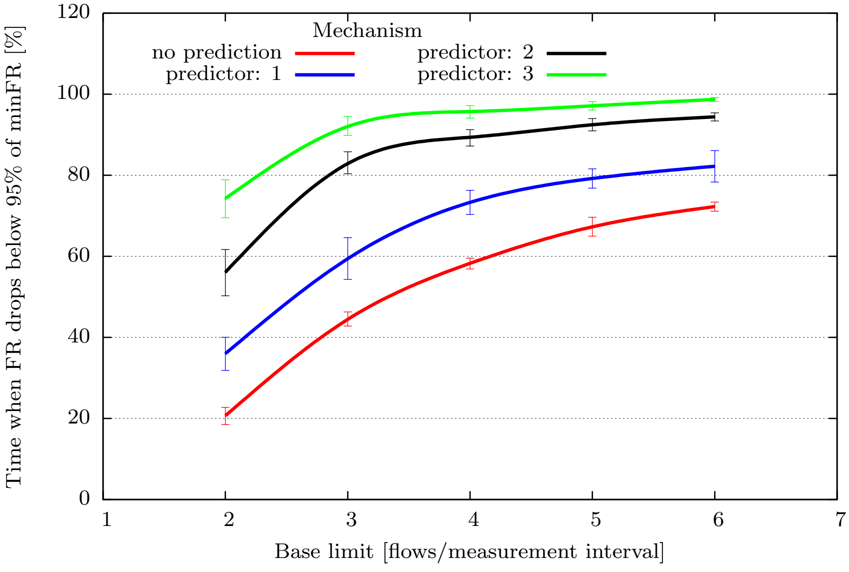


(a)

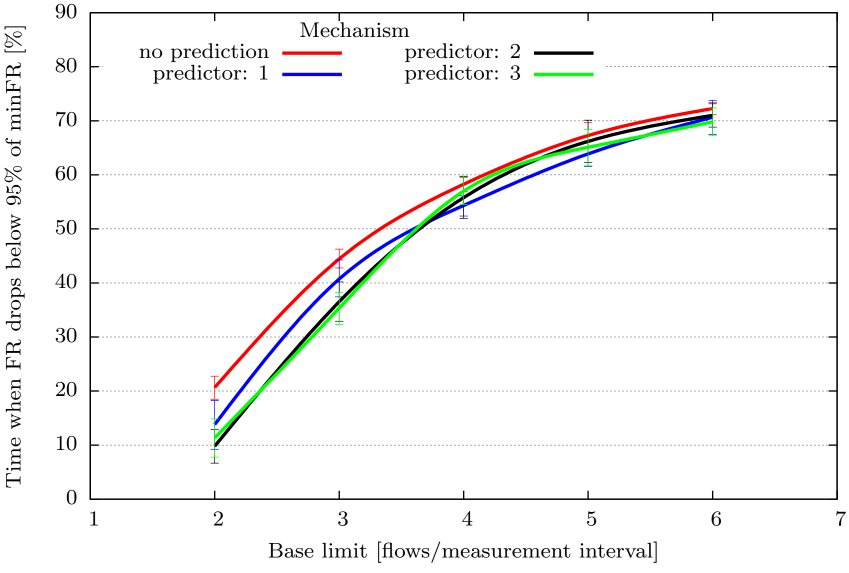


(b)

Figure 6.7: FR deviation from minimum FR with respect to the admission limit and (a) double prediction, (b) half prediction mechanisms



(a)



(b)

Figure 6.8: FR deviation from minimum FR with respect to the admission limit and (a) double prediction, (b) half prediction mechanisms

ferent traffic characteristics, especially related to high-capacity links, the proper choice od a predictor might play an important role.

6.5 Automatic intelligent limitations

The performance benefits obtained by using any of the proposed limitation mechanisms are substantial. However, only provided that the limit, static or dynamic, is chosen correctly. The examples have shown that when those mechanisms are not configured adequately to the traffic characteristic carried in the link, the resulted performance is not better, and in many cases, worse than that obtained with the regular FAN routine. Due to the fact that it is often difficult to predict the traffic characteristic, and the traffic features may change dynamically, there is a need for an automatic approach. In this section, I propose such a scheme which finds the proper limit through the trial and error routine.

```
if ((prevFR > minFR) and (FR < minFR)) {
    max_drop = 0;  deviation = 0;  counter = 0;
}

if (counter >= 0) deviation += FR − minFR;
counter++;

if ((counter > 0) and (deviation / counter > 0.3 ∗ minFR)) {
    AdmissionLimitFR++; counter = −5; deviation = 0;
}

drop = (minFR − FR) / (minFR);
if (drop > max_drop) max_drop = drop;

if ((prevFR < minFR) and (FR > minFR)) {
    #FR drop period has ended
    if (max_drop > 0.15) AdmissionLimitFR−−;
}
```

Figure 6.9: The automatic intelligent limitation mechanism

The pseudocode of the implemented automatic intelligent mechanism is presented in Figure 6.9. For the mechanism to operate, only 4 new variables need to be maintained, i.e., $prevFR$ which remembers the previous value of the FR, *deviation* and *counter* which are used to calculate the mean FR deviation and *max_drop* which represents the maximum FR drop in the current period of time.

The automatic intelligent mechanism monitors the FR measurements on a link. Those measurements are divided into periods of time in which FR is above and below the threshold. In each period, the situation is analyzed and proper actions are undertaken if necessary. If the system entered the below threshold phase (line 1) the variables must be reset (line 2). When FR is below the threshold the system counts the minimum value of FR which is achieved during the period (lines 12 and 13). If this value is lower than the predefined limit (or the FR drop is greater than the predefined limit), it means that due to over-admitting, too much flows were active during that period of time. Therefore, the admission limit is reduced (line 17). Similarly, in the periods in which the FR is above the threshold, the system calculates the deviation as defined in Equation 6.3 (line 5). When this deviation is greater than a predefined limit, it means that too few flows are active at the moment, and the admission limit must be increased (lines 8 and 9). After the admission limit has been increased, the deviation is reset and the counter is set to -5 which gives the system the time of 5 full measurement periods to adjust to the new limit before another actions are undertaken.

Experimentally, the thresholds were determined as follows: the admission limit is reduced when the maximum drop exceeds 15% of the minFR, the admission limit is increased when the deviation exceeds 30% of the minFR. For those values, the system provides sufficient performance while not changing the limit too often. To show the performance of the mechanism several simulations were performed. The scenario parameters were similar to those presented in previous sections of this chapter. The number of active flows was set to 2000 and the mean flow size varied from 2.5 MB to 15 MB. The effect of such a flow size differentiation is that when flows are shorter, they end more frequently and, therefore, more flows need to be admitted on a link in the same period of time. Exactly the same effect is caused by changing the links capacity while not altering traffic characteristics. This set of experiments show that the automatic intelligent mechanism performs well under various traffic characteristics and on links with different capacity.

Figure 6.10 shows the limit which was applied by the automatic mechanism in two exemplary scenarios, with the mean flow size of 2.5 (upper line) and 5 MB (lower line). Initially, the admission limit was set to 2 flows per measurement. As for this traffic characteristic, the limit was much too low, we can see the limit rising from the very beginning of the simulation. When the mean flow size is set to 2.5 MB, the automatic limit varies from 6 to 8, whereas for 5 MB flows, the limit sets itself on the level of 3 to 5 flows per measurement. Such a relationship is natural, as when flows are shorter, more of them need to be admitted in the same period of time, because more of them end in the same period. The simulations have also shown that the performance obtained with the automatic mechanism is not worse than that of the properly configured static limitations.

Table 6.4 presents the results of the whole experiment, comparing the automatic mechanism with the static limitations. The last row in both parts of the table shows the average admission limit which was applied by the automatic

Table 6.4: Mean deviation and FR drops duration under various limiting configurations and mean flow sizes

Limit [flows/measurement]	Mean flow size [MB]					
	2.5	5	7.5	10	12.5	15
FR deviation from the minFR [%]						
1	—	—	—	—	—	—
2	—	—	28.26 ± 30.10	7.57 ± 1.13	**5.38 ± 0.81**	**4.45 ± 0.64**
3	—	30.82 ± 35.67	**7.61 ± 1.36**	**5.74 ± 0.70**	5.31 ± 0.15	5.21 ± 0.23
4	—	**12.55 ± 4.57**	7.43 ± 0.20	6.32 ± 0.41	6.72 ± 0.36	6.57 ± 0.37
5	—	10.06 ± 0.45	8.02 ± 0.67	7.90 ± 0.41	7.73 ± 0.63	7.96 ± 0.35
6	67.88 ± 57.86	11.52 ± 1.64	9.05 ± 1.00	9.12 ± 0.98	9.13 ± 0.64	9.36 ± 0.26
7	**28.76 ± 7.70**	12.02 ± 0.79	10.65 ± 0.28	11.10 ± 1.04	11.03 ± 0.80	11.64 ± 0.25
8	29.42 ± 6.86	14.34 ± 1.23	11.89 ± 0.74	12.02 ± 1.25	12.50 ± 0.56	13.32 ± 0.78
intelligent	26.77 ± 2.20	11.39 ± 1.15	7.22 ± 0.48	6.03 ± 0.47	5.79 ± 0.27	5.20 ± 0.22
average limit	7.26 ± 0.38	4.10 ± 0.30	3.33 ± 0.21	2.97 ± 0.20	3.17 ± 0.31	2.78 ± 0.36
FR drops duration (below 90% of minFR) [%]						
1	—	—	—	—	—	—
2	—	—	0.00 ± 0.00	0.00 ± 0.00	**0.44 ± 0.59**	**0.07 ± 0.20**
3	—	0.82 ± 0.77	**3.34 ± 1.59**	**4.48 ± 2.28**	6.59 ± 2.59	6.37 ± 2.53
4	—	**5.13 ± 2.39**	10.91 ± 2.73	14.47 ± 3.41	19.19 ± 5.40	18.22 ± 3.49
5	—	13.20 ± 4.63	21.40 ± 3.40	27.03 ± 4.58	27.22 ± 4.41	31.49 ± 2.77
6	2.63 ± 2.30	24.09 ± 4.96	28.90 ± 4.48	34.38 ± 4.85	40.63 ± 6.07	43.08 ± 2.71
7	**8.11 ± 2.11**	32.63 ± 4.06	40.14 ± 1.68	47.82 ± 6.05	51.37 ± 4.19	52.82 ± 3.05
8	17.89 ± 5.70	40.99 ± 0.40	47.24 ± 1.31	51.17 ± 5.45	56.11 ± 2.96	60.11 ± 3.42
intelligent	12.02 ± 1.33	6.38 ± 1.11	6.00 ± 1.50	6.28 ± 1.26	9.53 ± 2.85	6.25 ± 1.48
average limit	7.26 ± 0.38	4.10 ± 0.30	3.33 ± 0.21	2.97 ± 0.20	3.17 ± 0.31	2.78 ± 0.36

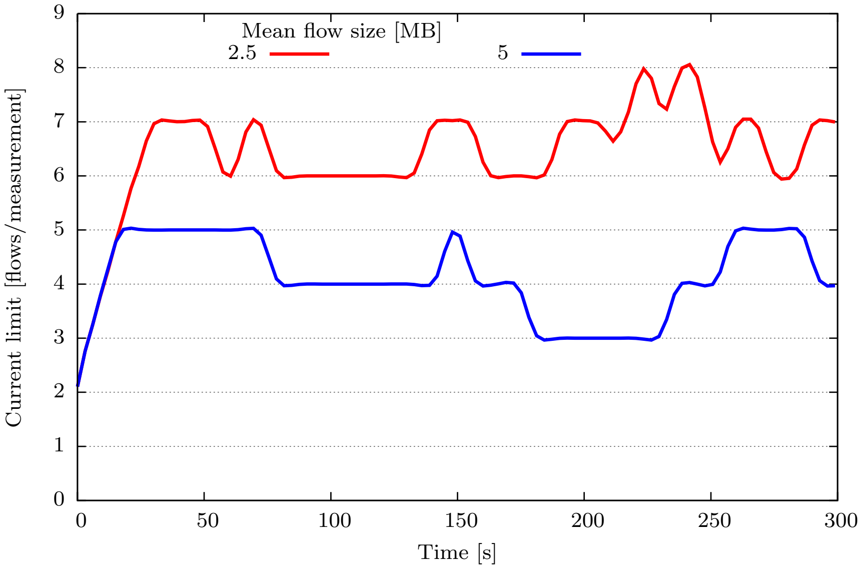


Figure 6.10: Limit applied by the automatic intelligent mechanism over time

mechanism. This is to show that automatization produces great results by accurately finding the best possible static limit. The marked values show the static limit which provides best results in terms of both the deviation and FR drops duration. Unfilled cells represent the case in which the limit was inadequately low which resulted in severe under-admitting (the link did not reach its steady state, as defined in Section A.4). We can see that the average limit applied by the automatic mechanism is very close to the static limit yielding the best results, which proves the efficiency of the automatic mechanism.

For the cases with the mean flow size equal to 12.5 and 15 MB, the best results are obtained with the static limit of 2 flows per measurement, however, the automatic mechanism sets the limit to 3 flows per measurement for the most of the simulation time. This is caused by the fact, that the system does not change the limit when the currently measured performance is sufficient. Even though choosing 3 flows per measurement is suboptimal, the achieved performance is still good enough. It is possible to configure the mechanism to more actively seek for the optimal solution by changing the performance indicators' thresholds, however, such a modification inflicts more frequent admission limit changes.

The most important benefit of the automatic intelligent mechanism is not the achieved performance, but rather the fact that the performance is close to optimal regardless of the current network condition and the traffic characteristics.

By implementing this mechanism the network operator does not need to analyze the link and set the proper limit which is a clear advantage of this solution.

6.6 Limitation mechanisms and network neutrality

It is easy to notice that mechanisms proposed in this chapter do not violate the network neutrality principles. Rather than providing service differentiation, the prediction and the limitation mechanisms aim at assuring the quality of transmission. Although the mechanisms allow certain flows to begin their transmission and blocks the others, the decision process is not based on the individual properties of a flow, especially its source and destination addresses or its application.

The limitation mechanism could be, however, exploited by network operators. For example, when the static limit of 2 flows per measurement is applied, the administrator might want to admit not the first two flows that come to the router, but rather those two flows which are of special interest to the operator. From the end user point of view, the procedure is not going to be noticeable, yet it will provide better performance (in terms of admission probability) for certain flows which will diminish fairness and, therefore, violate the neutrality principle. Nevertheless, despite the possibility of malicious usage, the mechanisms when working as intended, do not infract the net neutrality principle.

6.7 Conclusion

Flow-Aware Networking is a simple and efficient architecture which provides QoS differentiation in the IP networks. This proposition is relatively new and still needs some improvements or additional mechanisms. In this chapter I have shown that frequent degradations of the FR may occur on FAN links when there are too many flows attempting to acquire access to the link's bandwidth. To prevent those degradations, either FR needs to be measured more often, or we need to introduce some sort of limitations. The first option, as explained, consumes much more router's CPU power which is undesirable. Limitations, on the other hand, are viable, easy to implement and the benefits from introducing them are remarkable.

This chapter proposes two variants of the limiting mechanism, i.e., the static hard-coded limit, pre-set by the administrator and the dynamic limit which changes according to the link's current traffic characteristics. Despite the simplicity of the proposed mechanisms, the performance improvement is significant. The simulations have shown that, it is much better to introduce those mechanisms than to increase the frequency of measurement even 10 times.

Additionally, the prediction mechanism which enhances the admission control routine in the FAN routers is presented. Once again the results have shown that we can observe improvement over one obtained by the plain limitation mechanism. Comparing to the standard FAN performance, the performance improvement is even more impressive. Finally, a mechanism which automatically selects the most suitable limit is proposed. This way the system becomes more robust and invulnerable to faulty set-ups. The simulations show that the average admission limit applied by the automatic mechanism is very close to the static limit which provides the best performance, which proves the great efficiency of the automatic mechanism.

Part IV

Finale

7 Conclusions

The goal of the research presented in this dissertation was to prove that it is possible to provide rich service differentiation and quality assurance in Flow-Aware Networks in such a way so that the architecture remains net neutral. Therefore, after the initial chapters which introduce the notion of network neutrality, general concepts of FAN and the comparison of other significant flow-based QoS architectures, new mechanisms are proposed.

The evaluation of all the proposed mechanisms was performed by using the ns-2 network simulator. The presentation of the simulation results is followed by the analysis of the assessed mechanism's performance. All the mechanisms are also analyzed with relation to the network neutrality principle.

FAN provides a QoS assurance for active flows even in terms of overload. To do that, certain flows must be blocked until the network remains congested. This behavior may force some flows to wait for a very long time, which is a real problem for certain applications for which the admission time is crucial. To overcome the described negative behavior, a differentiated blocking scheme is proposed. All flows related to realizing certain services are assigned to a premium class by the admission control blocks. To achieve this goal, the Static Router Configuration, as a way to inform all the nodes which flows should be prioritized, is also proposed. Considering significant benefits, along with a reasonably low cost associated with the proposition, I believe that introducing differentiated blocking along with the SRC approach can greatly improve the end-user perception of the FAN architecture. Lastly, it has been shown that for the purpose of the Internet telephony, the proposed solutions do not significantly interfere with the overall performance of the architecture.

Bitrate differentiation enables FAN networks to provide guarantees on a different level than the minimum fair rate threshold. Moreover, to implement differentiated queueing, only cosmetic alterations to the FAN's queuing disciplines are required.

The proposed mechanisms interfere with the admission control and scheduling blocks of the XP router, the result of which may be temporal performance degradation of the carried traffic. This issue was thoroughly documented and proved to be insignificant to the overall performance of the FAN architecture, provided that the amount of prioritized traffic remains within reasonable boundaries.

Finally, the Class of Service on Demand approach was presented. This scheme utilizes the possibilities that are provided by both differentiated blocking and differentiated queuing. This way the service differentiation possibilities offered by the FAN architecture are greatly enhanced. Moreover, this approach proves that it is possible to provide service differentiation in a net neutral way.

It is shown that frequent degradations of the FR may occur on FAN links when there are too many flows attempting to acquire access to the link's bandwidth. To prevent those degradations, either FR needs to be measured more often, or we need to introduce some sort of limitations. The first option, as explained, consumes much more router's CPU power which is undesirable. Limitations, on the other hand, are viable, easy to implement and the benefits from introducing them are remarkable.

Two variants of the limiting mechanism are proposed, i.e., the static hard-coded limit, pre-set by the administrator and the dynamic limit which changes according to the link's current traffic characteristics. Despite the simplicity of the proposed mechanisms, the performance improvement is considerable. The simulations have shown that it is much better to introduce those mechanisms than to increase the frequency of measurement even 10 times.

Additionally, the prediction mechanism which enhances the admission control routine in the FAN routers is presented. Once again the results have shown that we can observe improvement over one obtained by the plain limitation mechanism. Comparing to the standard FAN performance, the difference is even more impressive. Finally, an intelligent mechanism which automatically selects the most suitable limit is proposed. This way the system becomes more robust and invulnerable to faulty set-ups. The simulations show that the average admission limit applied by the automatic mechanism is very close to the static limit which provides the best performance, which proves the great efficiency of the automatic mechanism.

The whole dissertation has shown and resolved two general problems of FAN networks. Firstly, that the QoS differentiation capabilities of FAN are not as limited as provided by the original idea. Secondly, the level of quality assurance in the original FAN is not impressive, especially when the offered traffic is heavy, and that this can be substantially improved by introducing new mechanisms.

As FAN is a relatively new approach, it has some drawbacks. The mechanisms presented in this dissertation render FAN more robust and help the architecture approach its maturity.

7.1 Achievements and contributions

The achievements and contributions of the dissertation can be summarized as follows:

1. A comprehensive survey of the QoS architectures designed for the IP networks and operating on flows is presented. Nine architectures are compared and contrasted in the most important aspects.

2. A thorough analysis of the Flow-Aware Networking concept is provided. Both the advantages and drawbacks of the solution are described.

3. A differentiated blocking approach is proposed which enables the reduction of the connection waiting times for certain flows.

4. A Static Router Configuration approach is proposed as a viable technique to provide differentiated blocking for local scope services.

5. A differentiated queuing mechanism is evaluated in its two variants: the bitrate differentiation and the fair rate ignoring schemes.

6. The Class of Service on Demand method to provide rich service differentiation without the need of signaling is envisaged. Here, a user decides to which class of service his/her flows should belong. The model proposes the exemplary set of classes and it is shown how such sets can be constructed.

7. The roots of fair rate degradations in FAN networks are identified. They originate from the properties of the admission control block, as well as may be caused by the differentiated blocking approach.

8. A static limitation mechanism which in a simple, yet very efficient way, reduces the fair rate degradations is proposed. The simulations show that it is much better to introduce the limitation mechanism than to increase the FR measurement frequency even 10 times.

9. Simulations have shown that the static limitation mechanism can be improved by providing the dynamics to the system. In the dynamic limitation mechanism the current admission limit varies and depends on the current link congestion status. The results show that in some conditions, the performance can be better that that of the static mechanism.

10. The proposed limitation mechanisms provide significant performance benefits, however, only if the limit is properly set. An automatic intelligent mechanism was developed to relieve that necessity. The simulation analysis shows that the mechanism's accuracy in finding the optimal admission limit is very high.

11. A predictive approach proved to provide superior performance compared to the limitation mechanism. Given that static limitations alone provide a substantial performance increase, the gain obtained from combining the predictive approach and the limitations is even larger.

12. An in-depth analysis of the network neutrality debate is provided. The opinions of both sides are presented and objectively discussed. Also, the impact of net neutrality on the QoS architectures is shown. The analysis of all the proposed new mechanisms with relation to the net neutrality was provided.

In the light of the presented achievements it can be stated that the thesis: ***It is possible to provide Quality of Service differentiation mechanisms in Flow-Aware Networks which follow the Net Neutrality concept***, has been proved.

Appendices

A Simulation experiment credibility

In this dissertation, conclusions are drawn from the simulations. This powerful tool allows to evaluate even the smallest details and to show their exact impact on the overall performance. However, in order to be valid, and to be able to form a conclusive line of reasoning, the experiments must be carried out properly. This appendix shows how the simulations were performed throughout the dissertation, how the data were gathered and analyzed and how the simulation environment was tested.

A.1 The network simulator

All the simulations the results of which are presented in this dissertation have been performed in the ns-2 network simulator [82] version 2.33. Ns-2 is a discrete event simulator targeted at networking research. Ns-2 provides substantial support for simulation of the TCP/IP protocol stack which represents the transmission in the current Internet. This simulation environment is particularly useful for evaluating new proposals, as it is licensed for use under version 2 of the GNU General Public License, which essentially allows everybody to modify its source-code to provide new functionalities, mechanisms, protocols, etc.

The general functionality of the Flow-Aware Networks is not implemented in the core of ns-2. The implementation of this architecture has been provided within the research project "FAN" founded by France Telecom in which the author of this dissertation participated. On top of the general functionality of FAN, I have implemented the proposed new mechanisms.

The FAN functionality was thoroughly tested before the research. Tyszer in [108] suggests certain steps to be undertaken to validate the simulation environment. These steps include:

- check if the model appears to be reasonable on its face to a field expert,
- test for sensitivity (slight changes in the model attributes should not result in significantly different results),
- test for degeneracy (removal of the portion of the model should result in the model's behavior that reflects this action),
- test for absurd conditions (imposing some unrealistic conditions may reveal some modeling flaws).

The first step was conducted during the "FAN" research project with the main founder of the architecture, i.e., James Roberts. The remaining tests were performed internally and abundantly repeated upon any alterations of the simulator's source-code.

A.2 Random number generation

Pawlikowski in [91] explains that the use of appropriate pseudo-random generators of independent uniformly distributed numbers is one of two, often neglected, necessary conditions of a credible simulation study. Pseudo-random number generation in ns-2 is performed by the RNG class. Starting from version 2.1b9, this class contains an implementation of the combined multiple recursive generator MRG32k3a proposed by L'Ecuyer [70]. The C++ code which resides in the core of the ns-2 simulator, was adapted from [71].

The MRG32k3a generator provides $1.8 \cdot 10^{19}$ independent streams of random numbers, each of which consists of $2.3 \cdot 10^{15}$ substreams. Each substream has a period (i.e., the number of random numbers before overlapping) of $7.6 \cdot 10^{22}$. The period of the entire generator is $3.1 \cdot 10^{57}$. When a new RNG object is created (each random variable is used as a separate RNG object), it is automatically seeded to the beginning of the next independent stream of random numbers. Used in this manner, the implementation allows for a maximum of $1.8 \cdot 10^{19}$ random variables.

All the experiments presented in this dissertation have been repeated many times to allow for statistical analysis of the results. For each replication, a different substream was used to ensure that the random number streams are independent. Each random variable in a single replication can produce up to $7.6 \cdot 10^{22}$ random numbers before overlapping, which is far greater than what was needed.

Following the ns-2 manual, the proper setting of the random number generator is as follows: (the code below is a part of the used simulation script, available at [84])

```
if {$argc > 1} {
    puts ”Usage:␣ns␣rng−test.tcl␣\[replication␣number\]”
    exit
}
set run 1
if {$argc == 1} {
    set run [lindex $argv 0]
}
if {$run < 1} {
    set run 1
}

# seed the default RNG
global defaultRNG
# setting seed to 0 provides non−deterministic behavior
$defaultRNG seed 0

# create the RNGs and set them to the correct substream
set arrivalRNG [new RNG]
set sizeRNG [new RNG]
for {set j 1} {$j < $run} {incr j} {
    $arrivalRNG next−substream
    $sizeRNG next−substream
}
```

Figure A.1: Setting the random number generation in ns-2

A.3 Statistics and confidence intervals

Proper analysis of the gathered data is required to provide reliable results from which conclusions can be drawn. All the results presented in this dissertation were gathered using the independent replication method, as described in [108]. It is the most direct approach to estimate the characteristics of steady-state distributions. The essence of this method is to run the simulation a number of times, starting with the same initial conditions, but using different, not overlapping, random number sequences.

For each independent run, the duration of the transient phase (the warm-up time) must be individually estimated. This, essentially, partitions the total simulation time into transient and steady-state phases. The method of estimating the transient period used in this dissertation is presented in Section A.4. Using the described procedure, we can obtain independent point-estimates. The average of these estimates forms the final point estimate with a confidence interval calculated by applying standard rules of statistics.

In this dissertation, the Student's t-distribution was used to calculate the 95% confidence intervals. It is an adequate estimate for the mean of a normally distributed population in situations where the sample size n is small ($n < 30$). Equations A.1, A.2 and A.3 were used to calculate the mean ($\overline{X}(n)$), standard deviation ($S(n)$) and the confidence intervals, respectively. The symbols have the following meaning: n is the number of replications, X_i is the i-th point-estimate, $t_{n-1,1-\frac{\alpha}{2}} \frac{S(n)}{\sqrt{n}}$ is the critical point of the distribution with $n-1$ degrees of freedom and $1-\alpha$ represents the desired confidence.

$$\overline{X}(n) = \frac{1}{n} \sum_{i=1}^{n} X_i \tag{A.1}$$

$$S(n) = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} \left[X_i - \overline{X}(n)\right]^2} \tag{A.2}$$

$$P\left[\overline{X}(n) - t_{n-1,1-\frac{\alpha}{2}} \frac{S(n)}{\sqrt{n}} < EX < \overline{X}(n) + t_{n-1,1-\frac{\alpha}{2}} \frac{S(n)}{\sqrt{n}}\right] = 1 - \alpha \tag{A.3}$$

The number of replications that were performed throughout the research varied. Typically it was 10, however for certain cases more runs needed to be performed in order to shorten the confidence intervals, whereas, in some cases, even as few as 5 replications were sufficient. Nevertheless, in all cases the number of replications were chosen such that the resulting confidence intervals were relatively small.

A.4 Transient period

As mentioned in Section A.3, the total simulation time can be divided into the transient phase and the steady-state. Just after initialization, any queuing process with nondeterministic, random streams of arrival and/or random service times is in a transient phase, during which its characteristics vary with time. This is caused by the fact that queuing systems or networks initially traverse along nonstationary trajectories, as, e.g., initially, links do not carry traffic, queues are empty, users are inactive, etc. After a period of time, if the system is stable, it approaches its statistical equilibrium on a stationary trajectory, or remains permanently on a nonstationary trajectory if the system is unstable. In this dissertation, for a stable system, a permanently congested FAN link was considered.

The transient period does not characterize the stable system, therefore, all the data collected during this period must be disregarded. In [90], eleven known rule-of-thumb approaches to determining the duration of the initial warm-up period are presented. As can be judged from the number of well known approaches, there is no universal method to determine the duration of the transient period.

Each rule can be applied under proper circumstances, otherwise providing poor estimation.

For the simulations presented in this work, the following general rule was applied: “the initial transient period ends when the measured indicator stops heading in one direction and starts oscillating”. This indicator was the FR parameter which, in FAN, shows how much the system is congested. The transient period is visible e.g., in Figure 5.17 on page 99 which presents the measured FR values over time. Initially, the link is empty, therefore, the value of FR is equal to the link capacity. As the link starts to carry traffic, FR starts to drop. This tendency continues until the FR reaches the minFR threshold, when the admission control block starts to deny new flows. As a consequence, FR starts to oscillate around its threshold. Unless stated otherwise, in the dissertation, the transient period was set (individually for each simulation run) until the FR value reached the minFR threshold for the first time.

Bibliography

[1] AT&T chief, FCC chair clarify on Net Neutrality. online, March 2006. http://www.zdnet.com/news/at-38t-chief-fcc-chair-clarify-on-net-neutrality/147323, downloaded on: 2010-11-10. *(Cited on page 18.)*

[2] V. Alwayn. *Advanced MPLS Design and Implementation.* Cisco Press, Indianapolis, IN, USA, 2002. *(Cited on page 57.)*

[3] Anagran. Eliminating Network Congestion Anywhere with Fast Flow Technology from Anagran. http://www.anagran.com, 2005. White Paper, downloaded on 10th June 2009. *(Cited on page 52.)*

[4] Telecommunications Industry Association. QoS Signaling for IP QoS Support, May 2006. Recommendation TIA-1039. *(Cited on page 67.)*

[5] N. G. Bean. Robust connection acceptance control for ATM networks with incomplete source information. *Annals of Operations Research*, 48(4), August 1994. *(Cited on page 110.)*

[6] N. Benameur, S. Ben Fredj, F. Delcoigne, S. Oueslati-Boulahia, and J.W. Roberts. Integrated Admission Control for Streaming and Elastic Traffic. In *Proc. Second International Workshop on Quality of Future Internet Services, QofIS 2001*, Coimbra, Portugal, September 2001. *(Cited on page 29.)*

[7] N. Benameur, S. Ben Fredj, S. Oueslati-Boulahia, and J.W. Roberts. Quality of Service and flow level admission control in the Internet. *Computer Networks*, 40:57–71, August 2002. *(Cited on page 29.)*

[8] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. IETF RFC 2475, December 1998. *(Cited on pages 23, 47, 49 and 53.)*

[9] B. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13:422–426, July 1970. *(Cited on page 63.)*

[10] T. Bonald, S. Oueslati-Boulahia, and J. Roberts. IP traffic and QoS control. In *Proc. World Telecommunications Congress, WTC 2002*, Paris, France, September 2002. *(Cited on page 24.)*

[11] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture an Overview. IETF RFC 1633, June 1994. *(Cited on pages 23, 47, 49 and 51.)*

[12] R. Braden and L. Zhang. Resource ReSerVation Protocol (RSVP) — Version 1 Message Processing Rules. IETF RFC 2209, September 1997. *(Cited on page 66.)*

[13] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. IETF RFC 2205, September 1997. *(Cited on page 66.)*

[14] C. Cárdenas and M. Gagnaire. Evaluation of Flow-Aware Networking (FAN) architectures under GridFTP traffic. *Future Generation Computer Systems*, 25:895–903, September 2009. *(Cited on page 42.)*

[15] C. Cardenas, M. Gagnaire, V. López, and J. Aracil. Admission control for Grid services in IP networks. In *Proc. Advanced Networks and Telecommunication Systems, ANTS 2007*, Bombay, India, December 2007. *(Cited on page 42.)*

[16] C. Cardenas, M. Gagnaire, V. López, and J. Aracil. Performance evaluation of the Flow-Aware Networking (FAN) architecture under Grid environment. In *Proc. IEEE Network Operations and Management Symposium, NOMS 2008*, pages 481–487, Paris, France, April 2008. *(Cited on page 42.)*

[17] A. Chapman and H. T. Kung. Automatic Quality of Service in IP Networks. In *Proc. Canadian Conference on Broadband Research, CCBR 1997*, pages 184–189, Ottawa, Canada, April 1997. *(Cited on pages 49, 51 and 57.)*

[18] C. Cárdenas, M. Gagnaire, V. López, and J. Aracil. Admission control in Flow-Aware Networking (FAN) architectures under GridFTP traffic. *Optical Switching and Networking*, 6:20–28, January 2009. *(Cited on page 42.)*

[19] J. Crowcroft. Net neutrality: the technical side of the debate: a white paper. *ACM SIGCOMM Computer Communication Review*, 37(1):49–56, 2007. *(Cited on pages 12, 20 and 21.)*

[20] K. Deeb, Sean P. O'Brien Sr., and Matthew E. Weiner. A survey on network neutrality; a new form of discrimination based on network profiling. *International Journal of Networking and Virtual Organisations*, 6(4):426–436, 2009. *(Cited on page 17.)*

[21] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. In *Proc. ACM SIGCOMM 1989*, pages 1–12, New York, NY, USA, 1989. ACM. *(Cited on pages 53 and 61.)*

[22] S. Dharmapurikar, P. Krishnamurthy, T.S. Sproull, and J.W. Lockwood. Deep packet inspection using parallel bloom filters. *Micro, IEEE*, 24(1):52 – 61, 2004. *(Cited on page 20.)*

[23] J. Domżał,, K. Wajda, S. Spadaro, J. Sole-Pareta, and D. Careglio. Recovery, Fairness and Congestion Mechanisms in RPR Networks. In *12 th Polish Teletraffic Symposium PSRT*, Poznan, Poland, September 2005. *(Cited on pages 7 and 42.)*

[24] J. Domzal. *Congestion Control in Flow-Aware Networks.* PhD thesis, AGH University of Sciene and Technology, Poland, 2009. ISBN: 978-83-88309-57-1. *(Cited on page 41.)*

[25] J. Domzal and A. Jajszczyk. New Congestion Control Mechanisms for Flow-Aware Networks. In *Proc. IEEE International Conference on Communications ICC 2008*, Beijing, China, May 2008. *(Cited on pages 41 and 70.)*

[26] J. Domzal and A. Jajszczyk. The Flushing Mechanism for MBAC in Flow-Aware Networks. In *Proc. 4th EURO-NGI Conference on Next Generation Internet Networks, NGI 2008*, pages 77–83, Krakow, Poland, April 2008. *(Cited on pages 41 and 70.)*

[27] J. Domzal and A. Jajszczyk. The Impact of Congestion Control Mechanisms for Flow-Aware Networks on Traffic Assignment in Two Router Architectures. In *Proc. International Conference on the Latest Advances in Networks, ICLAN 2008*, Toulouse, France, December 2008. *(Cited on page 41.)*

[28] J. Domzal and A. Jajszczyk. Approximate Flow-Aware Networking. In *Proc. IEEE International Conference on Communications ICC 2009*, Dresden, Germany, June 2009. *(Cited on pages 33, 34 and 42.)*

[29] J. Domzal, R. Wojcik, and A. Jajszczyk. The Impact of Congestion Control Mechanisms on Network Performance after Failure in Flow-Aware Networks. In *Proc. International Workshop on Traffic Management and Traffic*

Engineering for the Future Internet, FITraMEn 2008, Book: Traffic Management and Traffic Engineering for the Future Internet, Lecture Notes on Computer Science 2009, Porto, Portugal, December 2008. *(Cited on pages 7 and 42.)*

[30] J. Domzal, R. Wojcik, and A. Jajszczyk. QoS-Aware Net Neutrality. In *Proc. The First International Conference on Evolving Internet, INTERNET 2009,*, pages 147 –152, Cannes, France, August 2009. *(Cited on pages 6, 20, 43 and 100.)*

[31] J. Domzal, R. Wojcik, and A. Jajszczyk. Reliable Transmission in Flow-Aware Networks. In *Proc. IEEE Global Communications Conference GLOBECOM 2009*, pages 1 –6, Honolulu, USA, December 2009. *(Cited on pages 6, 7 and 41.)*

[32] J. Domzal, R. Wojcik, A. Jajszczyk, V. López, J.A. Hernandez, and J. Aracil. Admission control policies in Flow-Aware Networks. In *Proc. 11th International Conference on Transparent Optical Networks, ICTON 2009*, pages 1 –4, Azores, Portugal, July 2009. *(Cited on pages 7 and 42.)*

[33] J. Domzal, R. Wojcik, K. Wajda, A. Jajszczyk, V. López, J.A. Hernandez, J. Aracil, C. Cardenas, and M. Gagnaire. A multi-layer recovery strategy in FAN over WDM architectures. In *Proc. 7th International Workshop on Design of Reliable Communication Networks, DRCN 2009*, pages 160 –167, Washington, USA, October 2009. *(Cited on pages 6, 7 and 42.)*

[34] L. Drzewiecki and M. Antoniak-Lewandowska. Flow Simulator — a flow-based network simulator. In *Proc. The International Conference on "Computer as a Tool", EUROCON 2007*, Warsaw, Poland, September 2007. *(Cited on page 43.)*

[35] N. Economides and J. Tag. Net Neutrality on the Internet: A Two-sided Market Analysis. Working Papers 07-14, NET Institute, September 2007. *(Cited on page 18.)*

[36] S. Ben Fredj, S. Oueslati-Boulahia, and J.W. Roberts. Measurement-based Admission Control for Elastic Traffic. In *Proc. 17th International Teletraffic Congress, ITC 2001*, Salvador, Brasil, December 2001. *(Cited on page 29.)*

[37] P. Ganley and B. Allgrove. Net neutrality: A user's guide. *Computer Law & Security Report*, 22(6):454 – 463, 2006. *(Cited on pages xvii and 15.)*

[38] Google and Verizon. A joint policy proposal for an open Internet. online, August 2010. `http://googlepublicpolicy.blogspot.com/2010/`

`08/joint-policy-proposal-for-open-internet.html` , downloaded on 2010-11-02. *(Cited on page 21.)*

[39] G. Goth. Net Neutrality's Unpublicized Achilles' Heel. *IEEE Internet Computing*, 10(3):4–6, May/June 2006. *(Cited on page 10.)*

[40] G. Goth. The Global Net Neutrality Debate: Back to Square One? *IEEE Internet Computing*, 14(4):7 –9, July 2010. *(Cited on page 20.)*

[41] P. Goyal, H. M. Vin, and H. Cheng. Start-time Fair Queuing: A Scheduling Algorithm for Integrated Services Packet Switching Networks. *IEEE/ACM Transactions on Networking*, 5:690–704, October 1997. *(Cited on page 34.)*

[42] J. Gozdecki, A. Jajszczyk, and R. Stankiewicz. Quality of Service terminology in IP networks. *IEEE Communications Magazine*, 41(3):153–159, March 2003. *(Cited on page 57.)*

[43] S. Greenstein. Four nightmares for net neutrality. *Micro, IEEE*, 26:12–13, November/December 2006. *(Cited on page 16.)*

[44] R. Guerin, S. Blake, and S. Herzog. Aggregating RSVP-based QoS Requests. IETF Internet draft, November 1997. *(Cited on page 47.)*

[45] R. W. Hahn and S. Wallsten. The Economics of Net Neutrality. *The Economists' Voice*, 3, 2006. *(Cited on page 18.)*

[46] F. Halsall. *Computer Networking and the Internet.* Pearson Education Limited, Harlow, England, 2005. *(Cited on page 57.)*

[47] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured Forwarding PHB Group. *IETF RFC 2597*, June 1999. *(Cited on page 59.)*

[48] G. Held. Net neutrality may be a necessity. *International Journal of Network Management*, 17(1):1–1, 2007. *(Cited on pages 10 and 16.)*

[49] S. Herzog. RSVP Extensions for Policy Control. IETF RFC 2750, January 2000. *(Cited on page 66.)*

[50] ITU-T Recommendation Y.2121. Requirements for the support of flow-state-aware transport technology in an NGN, January 2008. *(Cited on pages 50, 54, 56, 64 and 68.)*

[51] ITU-T Recommendation Y.2122. Flow aggregate information exchange functions in NGN, June 2009. *(Cited on pages 65 and 68.)*

[52] V. Jacobson, K. Nichols, and K. Poduri. An Expedited Forwarding PHB. IETF RFC 2598, June 1999. *(Cited on page 59.)*

[53] A. Jajszczyk and R. Wojcik. Emergency Calls in Flow-Aware Networks. *Communications Letters, IEEE*, 11:753–755, September 2007. *(Cited on pages 5, 6, 41, 70 and 75.)*

[54] Y. Jiang, P.J. Emstad, A. Nevin, V. Nicola, and M. Fidler. Measurement-Based Admission Control for a Flow-Aware Network. In *Proc. 1st Conference on Next Generation Internet Networks - Traffic Engineering, NGI 2005*, pages 318–325, Rome, Italy, April 2005. *(Cited on page 29.)*

[55] A. Joch. Debating net neutrality. *Communications of the ACM*, 52(10):14–15, 2009. *(Cited on page 19.)*

[56] S. Jordan. Implications of Internet architecture on net neutrality. *ACM Transactions on Internet Technology*, 9(2):1–28, 2009. *(Cited on page 20.)*

[57] J. Joung. Feasibility of Supporting Real-Time Traffic in DiffServ Architecture. In *Proc. 5th International Conference on Wireless/Wired Internet Communications, WWIC 2007*, volume 4517, pages 189–200, Coimbra, Portugal, May 2007. *(Cited on page 65.)*

[58] J. Joung, J. Song, and S. S. Lee. Flow-Based QoS Management Architectures for the Next Generation Network. *ETRI Journal*, 30:238–248, April 2008. *(Cited on pages 50, 54, 60 and 68.)*

[59] S. Kaczmarek and M. Landowski. Performance of FAN Conception of Traffic Control in IP QoS Networks. In *Proc. 1st International Conference on Information Technology, IT 2008*, Gdansk, Poland, May 2008. *(Cited on page 43.)*

[60] B. Kim. A comparison of network neutrality debates between US and South Korea. In *Proc. 11th international conference on Advanced Communication Technology, ICACT 2009*, pages 1785–1790, Piscataway, NJ, USA, February 2009. IEEE Press. *(Cited on page 20.)*

[61] K. Kompella and J. Lang. Procedures for Modifying the Resource reSerVation Protocol (RSVP). IETF RFC 3936, October 2004. *(Cited on page 66.)*

[62] A. Kortebi, L. Muscariello, S. Oueslati, and J. Roberts. On the scalability of fair queueing. In *Proc. Third Workshop on Hot Topics in Networks, ACM HotNets-III 2004*, San Diego, USA, November 2004. *(Cited on pages 33 and 37.)*

[63] A. Kortebi, L. Muscariello, S. Oueslati, and J. Roberts. Evaluating the number of Active Flows in a Scheduler Realizing Fair Statistical Bandwidth Sharing. In *Proc. International Conference on Measurement and*

Modeling of Computer Systems, ACM SIGMETRICS 2005, Banff, Canada, June 2005. *(Cited on pages 33 and 37.)*

[64] A. Kortebi, L. Muscariello, S. Oueslati, and J. Roberts. Minimizing the Overhead in Implementing Flow-aware Networking. In *Proceedings of Symposium on Architectures for Networking and Communications Systems, ANCS 2005*, Princeton, USA, October 2005. *(Cited on page 31.)*

[65] A. Kortebi, S. Oueslati, and J. Roberts. MBAC algorithms for streaming flows in Cross-protect. In *Proc. Next Generation Internet Networks EuroNGI Workshop*, Lund, Sweden, June 2004. *(Cited on pages 29 and 31.)*

[66] A. Kortebi, S. Oueslati, and J. Roberts. Implicit Service Differentiation using Deficit Round Robin. In *Proc. 19th International Teletraffic Congress, ITC 2005*, Beijing, China, August/September 2005. *(Cited on pages xv, 25, 37, 38 and 39.)*

[67] A. Kortebi, S. Oueslati, and J. W. Roberts. Cross-protect: implicit service differentiation and admission control. In *Proc. High Performance Switching and Routing, HPSR 2004*, pages 56–60, Phoenix, AZ, USA, 2004. *(Cited on pages xv, 25, 28, 29, 34, 35, 37 and 50.)*

[68] R. Kuroda, M. Katsuki, A. Otaka, and N. Miki. Providing flow-based quality-of-service control in a large-scale network. In *Proc. 9th Asia-Pacific Conference on Communications, APCC 2003*, volume 2, pages 740–744, Penang, Malaysia, September 2003. *(Cited on pages 49, 52 and 62.)*

[69] P. Larouche. Law and technology: The network neutrality debate hits Europe. *Communications of the ACM*, 52(5):22–24, 2009. *(Cited on pages 11 and 20.)*

[70] P. L'Ecuyer. Good Parameters and Implementations for Combined Multiple Recursive Random Number Generators. *Operations Research*, 47(1):159–164, January/February 1999. *(Cited on page 140.)*

[71] P. L'Ecuyer, R. Simard, E. J. Chen, and W. D. Kelton. An object-oriented random number package with any long streams and substreams. *Operations Research*, 50(6), 2002. *(Cited on page 140.)*

[72] L. Lessig. Network Neutrality: Critical push. online, May 2006. `http://lessig.org/blog/2006/05/network_neutrality_critical_pu.html`, downloaded on 2010-11-02. *(Cited on page 16.)*

[73] J.-S. Li and C.-S. Mao. Providing flow-based proportional differentiated services in class-based DiffServ routers. In *IEE Proceedings on Communications*, volume 151, pages 82–88, February 2004. *(Cited on pages 49 and 53.)*

[74] R. E. Litan and H. J. Singer. Unintended Consequences of Net Neutrality Regulation. *Journal on Telecommunications and High Technology Law*, 5(3):533–572, 2007. *(Cited on page 18.)*

[75] V. López. *End-to-end quality of service provisioning in multilayer and multidomain environments*. PhD thesis, Universidad Autonoma de Madrid, 2010. *(Cited on page 41.)*

[76] V. López, C. Cardenas, J. A. Hernandez, J. Aracil, and M. Gagnaire. Extension of the Flow-Aware Networking (FAN) architecture to the IP over WDM environment. In *Proc. 4th International Telecommunication Networking Workshop on QoS in Multiservice IP Networks*, Venice, Italy, February 2008. *(Cited on pages 41 and 42.)*

[77] A. Mankin, F. Baker, B. Braden, S. Bradner, M. O'Dell, A. Romanow, A. Weinrib, and L. Zhang. Resource ReSerVation Protocol (RSVP) — Version 1 Applicability Statement Some Guidelines on Deployment. IETF RFC 2208, September 1997. *(Cited on page 66.)*

[78] L. Massoulie and J. Roberts. Arguments in Favour of Admission Control for TCP Flows. In *Proc. 11th International Teletraffic Congress, ITC 1999*, Edinbourg, June 1999. *(Cited on page 29.)*

[79] A. W. Moore. *Measurement-based management of network resources*. PhD thesis, University of Cambridge, April 2002. *(Cited on page 110.)*

[80] B. Nandy, N. Seddigh, A. Chapman, and J. Hadi Salim. A Connectionless Approach to Providing QoS in IP Networks. In *Proc. 8th Conference on High Performance Networking, IFIP 1998*, Vienna, Austria, September 1998. *(Cited on pages 49, 51 and 61.)*

[81] NETCompetition.org. online, October 2010. `http://netcompetition.org/`, downloaded on: 2010-11-10. *(Cited on page 19.)*

[82] Network Simulator ns-2. Available at http://nsnam.isi.edu/nsnam. *(Cited on pages 31, 43 and 139.)*

[83] K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. IETF RFC 2474, December 1998. *(Cited on pages 23, 56 and 68.)*

[84] Exemplary ns-2 FAN simulation script. Available at http://kt.agh.edu.pl/∼wojcik/phd. *(Cited on page 140.)*

[85] A. Odlyzko. The delusions of net neutrality, August 2008. Available at: http://www.dtc.umn.edu/~odlyzko/, downloaded on 2010-11-02. *(Cited on page 19.)*

[86] P. Ohm. When network neutrality met privacy. *Communications of the ACM*, 53(4):30–32, 2010. *(Cited on page 18.)*

[87] S. Oueslati and J. Roberts. A new direction for quality of service: Flow-aware networking. In *Proc. 1st Conference on Next Generation Internet Networks - Traffic Engineering, NGI 2005*, Rome, Italy, 2005. *(Cited on pages 20, 25, 26 and 29.)*

[88] S. Oueslati and J. Roberts. Comparing Flow-Aware and Flow-Oblivious Adaptive Routing. In *Proc. 41st Annual Conference on Information Sciences and Systems, CISS 2007*, Baltimore, MD, USA, March 2007. *(Cited on page 42.)*

[89] B. M. Owen. The Net Neutrality Debate: Twenty Five Years After United States v. AT&T and 120 Years After the Act to Regulate Commerce. *Stanford Law and Economics Olin Working Paper No. 336. (Cited on page 14.)*

[90] K. Pawlikowski. Steady-state Simulation of Queueing Processes: A Survey of Problems and Solutions. *ACM Computing Surveys*, 22(2):123–170, June 1990. *(Cited on page 142.)*

[91] K. Pawlikowski, H-D. J. Jeong, and J-S. R. Lee. On Credibility of Simulation Studies of Telecommunications Networks. *IEEE Communications Magazine*, 40(1):132–139, January 2002. *(Cited on page 140.)*

[92] J. Polk and S. Dhesikan. A Resource Reservation Protocol (RSVP) Extension for the Reduction of Bandwidth of a Reservation Flow. IETF RFC 4495, May 2006. *(Cited on page 66.)*

[93] K. Psounis, R. Pan, and B. Prabhakar. Approximate Fair Dropping for Variable-Length Packets. *Micro, IEEE*, 21:48–56, January 2001. *(Cited on pages 33 and 42.)*

[94] J. Roberts. Internet Traffic, QoS and Pricing. In *Proc. the IEEE*, volume 92, pages 1389–1399, September 2004. *(Cited on page 25.)*

[95] J. Roberts and S. Oueslati. Quality of Service by Flow Aware Networking. *Philosophical Transactions of The Royal Society of London*, 358:2197–2207, 2000. *(Cited on pages 24 and 50.)*

[96] J. W. Roberts and L. Massoulie. Bandwidth Sharing and Admission Control for Elastic Traffic. In *Proc. ITC Specialist Seminar*, Yokohama, October 1998. *(Cited on page 29.)*

[97] L. Roberts. Internet founder ponders the web's future. *IT Professional*, 2:16–20, September/October 2000. *(Cited on page 52.)*

[98] L. Roberts. Micro-Flow Management, May 2007. Caspian Networks, INC, US patent application no. 2007/0115825 A1. *(Cited on pages 58, 62 and 67.)*

[99] L. Roberts and A. Henderson. System, Method, and Computer Program Product for IP Flow Routing, July 2007. Anagran, INC., US patent application no. 2007/0171825 A1. *(Cited on page 62.)*

[100] F. B. Schneider. Network Neutrality versus Internet Trustworthiness? *IEEE Security and Privacy*, 6(4):3–4, 2008. *(Cited on page 14.)*

[101] M. Shreedhar and G. Varghese. Efficient Fair Queuing Using Deficit Round-Robin. *IEEE/ACM Transactions on Networking*, 4:375–385, June 1996. *(Cited on page 37.)*

[102] J. Song, S.S. Lee, and Y. S. Kim. DiffProbe: One Way Delay Measurement for Asynchronous Network and Control Mechanism in BcN Architecture. In *Proc. 8th international conference on Advanced Communication Technology, ICACT 2008*, volume 1, pages 677–682, Phoenix Park, Republic of Korea, February 2006. *(Cited on page 65.)*

[103] I. Stoica. *Stateless Core: A Scalable Approach for Quality of Service in the Internet.* PhD thesis, Carnegie Mellon University, Pittsburgh, USA, December 2000. *(Cited on pages 49, 51, 55, 58 and 62.)*

[104] I. Stoica, S. Shenker, and H. Zhang. Core-stateless fair queueing: achieving approximately fair bandwidth allocations in high speed networks. *ACM SIGCOMM Computer Communication Review*, 28(4):118–130, 1998. *(Cited on page 49.)*

[105] I. Stoica and H. Zhang. Providing guaranteed services without per flow management. In *Proc. ACM SIGCOMM 1999*, pages 81–94, New York, NY, USA, 1999. *(Cited on pages 49, 51, 61 and 62.)*

[106] M. B. Tariq, M. Motiwala, N. Feamster, and M. Ammar. Detecting network neutrality violations with causal inference. In *Proc. 5th international conference on Emerging networking experiments and technologies, CoNEXT 2009*, pages 289–300, New York, NY, USA, December 2009. *(Cited on page 21.)*

[107] The Savetheinternet.com coalition. online, October 2010. `http://www.savetheinternet.com/`, downloaded on 2010-11-02. *(Cited on pages 11, 16 and 18.)*

[108] J. Tyszer. *Object-Oriented Computer Simulation of Discrete-Event Systems.* Kluwer Academic Publishers, 1999. *(Cited on pages 139 and 141.)*

[109] B. van Schewick and D. Farber. Point/Counterpoint Network neutrality nuances. *Communications of the ACM*, 52(2):31–37, 2009. *(Cited on pages xvii, 15 and 18.)*

[110] D.J. Weitzner. Net Neutrality... Seriously this Time. *Internet Computing, IEEE*, 12(3):86 –89, 2008. *(Cited on page 16.)*

[111] M. Welzl and M. Muhlhauser. Scalability and Quality of Service: A Trade-off? *IEEE Communications Magazine*, 41:32–36, June 2003. *(Cited on page 24.)*

[112] R. Wojcik, J. Domzal, and A. Jajszczyk. Fair Rate Degradation in Flow-Aware Networks. In *Proc. IEEE International Conference on Communications ICC 2010*, pages 1 –5, May 2010. *(Cited on pages 5, 6 and 110.)*

[113] R. Wojcik and A. Jajszczyk. Flow oriented approaches to QoS assurance. *ACM Computing Surveys (to be published)*, 2011. *(Cited on pages 5, 6 and 48.)*

[114] J. Wroclawski. The Use of RSVP with IETF Integrated Services. IETF RFC 2210, September 1997. *(Cited on page 66.)*

[115] XP. Xiao. *Technical, Commercial and Regulatory Challenges of QoS: An Internet Service Model Perspective.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2008. *(Cited on pages 19 and 71.)*

[116] L. Zhang. Virtual clock: a new traffic control algorithm for packet switching networks. In *Proc. ACM SIGCOMM 1990*, pages 19–29, New York, NY, USA, 1990. ACM. *(Cited on page 61.)*

Index