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

Ph.D. Thesis

Rafał Stankiewicz

Analytical Models of Selected DiffServ Network Elements Supporting Assured Forwarding

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 6345582
fax +48 12 6342372

www.agh.edu.pl
www.eaiie.agh.edu.pl
www.kt.agh.edu.pl

Reviewers:

prof. dr hab. inż. J. Woźniak†
prof. dr hab. inż. Z. Papir‡

† Gdańsk University of Technology
‡ AGH University of Science and Technology

ISBN 978-83-88309-81-6

Cover design by Rafał Stankiewicz

Printed in Poland
by *Drukarnia GS Sp. z o.o., ul. Zabłocie 43, Kraków*

To my wife and children

Acknowledgements

First and foremost, I would like to express my gratitude towards my advisor, Prof. Andrzej Jajszczyk for his broad vision, valuable advice, attention, and patience. I am also indebted to him for the encouragement to present my research results at top telecommunications conferences and journals and for introducing me to famous world researchers. I have been very fortunate to have the opportunity to derive from his great experience. This dissertation would not have been possible without his guidance and patience over the past years.

I would like to thank also Prof. Andrzej Pach for his contribution in the development of my research.

No words are to express my gratitude to Dr. Wiesław Ludwin who, being my MSc supervisor, was the first to teach me how to do research. I am grateful for his readiness to help at any time, valuable advice, and teaching the engineering approach towards problem solving.

I am also indebted to Dr. Edward Chlebus for his undeniable support in learning simulation techniques and experimental data collection as well as analysis methods.

I have been fortunate to work with a very friendly and always ready to help workmates who I wish to thank for creating an enjoyable and pleasant atmosphere to work and relations going beyond the job. I wish to thank Janusz Gozdecki for familiarizing me with the network simulator and Mathematica package. I am indebted to Andrzej Szymański for his valuable comments, discussions and criticizing my ideas. My great gratitude also goes to Dr. Artur Lasoń, the leader of a

team involved in European research projects LION and NOBEL, who I have appreciated to work with and gain the experience from. He has always fresh viepoints on professional issues as well as matters other than academic. I am also indebted to another team member, Jacek Rząsa, for his optimistic attitude to the world, fun and fruitful exchange of ideas. Many thanks are also due to my colleagues: Piotr Żuraniewski for hints regarding statistical credibility of experiments as well as Dr. Krzysztof Juszkiewicz for technical support and Dr. Lucjan Janowski for fruitful discussions.

I would like to thank Dr. Paweł Topa for his support in discovering the world of LaTeX and Dr. Agnieszka Rutkowska for help in mathematics.

Last but not the least, I would like to thank my family, especially my wife Beata, for their support, patience and love. I dedicate the dissertation to my wife and children.

Abstract

Analytical models of selected DiffServ network elements supporting Assured Forwarding (AF) service in a Differentiated Services (DiffServ) network are proposed in the dissertation. Simple analytical models for main network elements: meter/markers and droppers are provided.

Models of the following meter/markers were proposed: Single Rate Three Color Marker (srTCM), Two Rate Three Color Marker (trTCM), and Time Sliding Window Three Color Marker (TSW3CM). The model of the latter involves finding the distribution of the TSW-averaged traffic rate approximated by the rate estimator. A respective probability density function that, if integrated in appropriate bounds, allows finding packet marking probabilities, is proposed. Models for srTCM and trTCM, that use the token bucket mechanism, are based on the classical queuing theory. Namely, models based on the M/M/1/K and D/M/1/K queuing systems are considered.

The implementation of a dropper uses a selected Active Queue Management (AQM) mechanism. Most often, droppers are implemented as Multi-RED queues. Models for the following commonly used types of Multi-RED queues are developed: Weighted RED (WRED) and RED with In/Out and Coupled virtual queues (RIO-C). Both types of queues are scrutinized under two types of configuration of dropping thresholds: staggered and overlapped. The proposed models use the M/M/1/K and M/D/1/K queuing models with balking (impatient customers). Such models are often denoted as M(n)/M/1/K and M(n)/D/1/K, respectively. Additionally, in the case of RIO-C, a Bernoulli scheme is adopted in the model.

Since the models for DiffServ elements are independent of each other, they can be combined to model a more complex system. An example of such a model, supplemented by the model for a TCP throughput by the other author, is presented in the dissertation.

Accuracy of the models is verified by simulations of a DiffServ network. Simulations were performed with the `ns-2` simulator.

The models enable quick finding of meter/marker and dropper characteristics under a particular configuration and traffic volume. It can be also quickly checked how parameter manipulations or changes in the traffic rate affect the characteristics. Additionally, these models used together with the model of the traffic facilitate predicting characteristics of the whole system.An example is the aforementioned modular model with the model for TCP throughput.

Keywords: DiffServ, Assured Forwarding PHB, meter/marker, dropper, srTCM, trTCM, TSW3CM, WRED, RIO-C, queuing theory, TCP, IP, analytical models, modular model of DiffServ network, QoS

Streszczenie

W rozprawie zaprezentowano modele analityczne wybranych elementów architektury usług zróżnicowanych DiffServ (ang. *Differentiated Services*) używanych do realizacji klasy usług zagwarantowanych AF (ang. *Assured Forwarding*). Elementami tymi są: **jednostka pomiarowa i oznaczająca pakiety** (ang. *meter/marker*) oraz **jednostka odrzucająca pakiety** (ang. *dropper*).

Zaproponowano modele analityczne następujących typów jednostek pomiarowych i oznaczających pakiety: srTCM (ang. *Single Rate Three Color Marker*), trTCM (ang. *Two Rate Three Color Marker*) oraz TSW3CM (ang. *Time Sliding Window Three Color Marker*). Zasada działania jednostki TSW3CM oparta jest na estymacji średniej przepływności strumienia ruchu. Zaproponowano funkcję rozkładu prawdopodobieństa estymowanej przepływności strumienia ruchu, która po scałkowaniu w odpowiednich granicach pozwala wyznaczyć prawdopodobieństwa oznaczenia pakietu. W jednostkach typu srTCM i trTCM zastosowany jest mechanizm cieknącego wiadra (ang. *token bucket*). Modele analityczne tych jednostek są zbudowane z użyciem teorii kolejek, a w szczególności na systemach kolejkowych M/M/1/K oraz D/M/1/K.

W praktycznych realizacjach jednostki odrzucającej pakiety stosuje się wybrany mechanizm aktywnego zarządzania kolejką pakietów (ang. *Active Queue Management*). Zwykle stosuje się mechanizm wczesnego wykrywania i zapobiegania przeciążeniom z wieloma kolejkami wirtualnymi, Multi-RED. W pracy zaproponowano modele analityczne najbardziej popularnych praktycznych realizacji kolejki typu Multi-RED: WRED (ang. *Weighted Random Early Detection*) oraz RIO-C (*RED with In/Out and Coupled virtual queues*). Rozważano dwa sposoby konfiguracji wartości progowych dla odrzucania pakietów należących do różnych klas: z pokrywającymi się wartościami progowymi (ang. *overlapped*) oraz rozłącznymi (ang. *staggered*). Modele matematyczne jednostek WRED oraz RIO-C zbudowano opierając się na modelach kolejkowych z niecierpliwymi

klientami i skończonym buforem typu M/M/1/K i M/D/1/K. Takie modele kolejkowe są często oznaczane odpowiendnio jako M(n)/M/1/K oraz M(n)/D/1/K. Ponadto, w modelu kolejki typu RIO-C użyto dodatkowo schemat Bernoulliego.

Zaproponowane modele analityczne poszczególnych elementów sieci DiffServ są niezależne. Pozwala to na stworzenie bardziej złożonego modelu sieci DiffServ. Przykład takiego modułowego modelu, uzupełnionego o model źródła TCP (innego autora) zaprezentowano w pracy.

Poprawność modeli analitycznych zweryfikowano za pomocą symulacji sieci DiffServ. Posłużono się symulatorem `ns-2`.

Zaproponowane modele analityczne pozwlają na szybkie znalezienie charakterystyk jednostek pomiarowych i oznaczających pakiety oraz jednostek odrzucających pakiety dla wybranej konfiguracji oraz dla wybranego natężenia ruchu. Pozwlają też na szybkie sprawdzenie w jaki sposób zmiana parametrów konfiguracyjnych oraz zmiana średniej przepływności strumienia ruchu wpływa na kształt charakterystyk. Ponadto, zaporoponowane modele wraz z modelem ruchu pozwalają przewidzieć zachowanie bardziej złożonego systemu. Jak wspomniano, w pracy zamieszczono przykład modelu modułowego dla wybranego modelu źródła TCP.

Słowa kluczowe: DiffServ, klasa usług zagwarantowanych AF, jednostka pomiarowa i oznaczająca pakiety, jednostka odrzucająca pakiety, srTCM, trTCM, TSW3CM, WRED, RIO-C, teoria kolejek, TCP, IP, modele analityczne, złożony model sieci DiffServ, QoS

Contents

List of Figures

List of Tables

Abbreviations

A-RIO	adaptive RIO algorithm
ACF	autocorrelation function
ACK	TCP acknowledgment
ACT	adaptive CIR threshold mechanism
AF	assured forwarding
AQM	active queue management
ARCH	autoregressive conditional heteroscedasticity
ARIO-D	adaptive RIO algorithm with delay guarantees
ARIO-L	adaptive RIO algorithm with loss guarantees
ARM	active rate management meter/marker
ATB	adaptive token bucket meter/marker
ATM	asynchronous transfer mode
BA	behavior aggregate or behavior aggregate classifier
BB	bandwidth broker
BE	best effort
CBQ	class-based queuing
CBS	committed burst size
CBWFQ	class-based weighted fair queuing
CDF	cumulative distribution function

CIR	committed information rate
CoS	class of service
CR	core router
CS	class selector
DBW	dedicated bandwidth IP transfer capability
DRIO	dynamic RIO
DR	destination router
DRR	deficit round robin
DSCP	differentiated services codepoint
EBS	excess burst size
EF	expedited forwarding
ER	edge router
ETSI	European telecommunications standards institute
ETSW	enhanced time sliding window
EWMA	exponential weighted moving average
FAN	flow aware networking
FAQM	fuzzy active queue management
FM	fuzzy meter/marker
FS	fuzzy scheduler
FSA	flow state-aware routing
FTP	file transfer protocol
GARCH	generalized autoregressive conditional heteroscedasticity
IEEE	institute of electrical and electronics engineers
IETF	internet engineering task force
IP	internet protocol
IPv4	internet protocol, version 4
IPv6	internet protocol, version 6
ItswTCM	improved TSW based three color marker
ITU	international telecommunication union

ITU-T	international telecommunication union — telecommunication standardization sector
MAMT	multiple average multiple threshold realization of Multi-RED
MAST	multiple average single threshold — variant of RED queue
MBM	memory-based marker
MF	multifield classifier
MMFP	Markov modulated fluid processes
MRG32k3a	combined multiple recursive random number generator
MSS	TCP maximum segment size
MTU	maximum transmission unit
Multi-RED	physical queue with multiple RED virtual queues
NP	network performance
OA	ordered aggregate
PBS	peak burst size
PDB	per domain behavior
PDF	probability density function
PHB	per hop behavior
PIR	peak information rate
PME	packet marking engine
PQ	priority queuing
PSC	PHB scheduling class
QoS	quality of service
RED	random early detection
RFC	request for comments
RIO-C	RED with in/out and coupled virtual queues
RIO-DC	RED with in/out and decoupled virtual queues
RNG	random number generator
RTO	retransmission timeout
RTT	round trip time
SAMT	single average multiple threshold realization of Multi-RED

SAST	single average single threshold - variant of RED queue
SBW	statistical bandwidth IP transfer capability
SLA	service level agreement
SLS	service level specification
SR	source router
srTCM	single rate three color marker
SURGE	scalable URL reference generator
TB	token bucket
TCA	traffic conditioning agreement
Tcl	tool command language
TCP	transmission control protocol
TCS	traffic conditioning specification
TD	tail drop
TOS	type of service
trTCM	two rate three color marker
TSW	time sliding window
TSW2CM	time sliding window two color marker
TSW3CM	time sliding window three color marker
TWAM	TCP-window aware marker
UDP	user datagram protocol
URL	uniform resource locator
VPN	virtual private network
VS-ACT	variable structure adaptive CIR threshold
WFQ	weighted fair queuing
WRED	weighted random early detection
WRR	weighted round robin

List of symbols

Lowercase Greek symbols

α	significance level
$\hat{\gamma}(k)$	estimator of the autocovariance function
$\hat{\theta}_i$	estimate of the average size of the physical queue at the bottleneck router for interval i
$\hat{\theta}_i^{green}$	estimate of the average size of the virtual queue of green packets at the bottleneck router for interval i
$\hat{\theta}_i^{red}$	estimate of the average size of the virtual queue of red packets at the bottleneck router for interval i
$\hat{\theta}_i^{yellow}$	estimate of the average size of the virtual queue of yellow packets at the bottleneck router for interval i
λ	parameter of the exponential distribution of random variable T
λ_i	arrival rate given that the queue is in state i (queuing system with balking)
μ	bottleneck capacity, service rate, expressed in packets per second
$\hat{\mu}$	general standard point estimator
ν	size of the virtual queue
ν_g	instantaneous size of the virtual queue of green packets

ν_r	instantaneous size of the virtual queue of red packets
ν_y	instantaneous size of the virtual queue of yellow packets
π_i	probability that there are i packets in a Multi-RED queue
$\pi^C_{i\|T_p\neq 0}$	probability that the token count $T_c = i$, i.e., there are i packets in token bucket C, given that the token bucket P is not in state 0
$\pi_i^{\mathrm{D/M/1/K}}$	probability that D/M/1/K queuing system is in state i
$\pi_{i,j}$	probability that the queuing system representing srTCM is in state $E_{i,j}$
$\pi_i^{\mathrm{M/D/1/\infty}}$	probability that M/D/1/∞ queuing system is in state i
$\pi_i^{\mathrm{M/D/1/K}}$	probability that M/D/1/K queuing system is in state i
π^P_j	probability that the token count $T_p = j$, i.e., there are j packets in token bucket P
$\hat{\rho}\,(k)$	estimator of the autocorrelation function
$\hat{\rho}_{SQ}\,(k)$	estimator of the autocorrelation function of sample's squares
ϱ	notation of the system load in trTCM model
ς	coefficient used for calculation of T^{max}; arbitrarily equal to 0.999
$\hat{\sigma}^2\,[\hat{\mu}]$	unbiased estimator of variance of $\hat{\mu}$
$\tau_{i,j}$	time of the j-th event of packet arrival or departure to the queue, in simulation interval j
φ_i^{all}	total number of packets observed in i-th interval of simulation
$\varphi_i^{all,green}$	total number of green packets observed in i-th interval of simulation
$\varphi_i^{all,red}$	total number of red packets observed in i-th interval of simulation
$\varphi_i^{all,yellow}$	total number of yellow packets observed in i-th interval of simulation
φ_i^{drop}	total number packets dropped at the core router, observed in i-th interval of simulation
$\varphi_i^{drop,green}$	total number of dropped green packets observed in i-th interval of simulation

$\varphi_i^{drop,red}$	total number of dropped red packets observed in i-th interval of simulation
$\varphi_i^{drop,yellow}$	total number of dropped yellow packets observed in i-th interval of simulation

Uppercase Greek symbols

$\Gamma(\cdot)$	gamma function
Δ_x	half of the symmetric confidence interval for standard point estimator $\hat{\mu}$
Λ	rate of the traffic incoming to the dropper (used in general considerations)
Λ_{eff}	effective rate of the traffic eventually entering the dropper's output queue (used in general considerations)
Ψ_i	sum of bytes of all packets

Lowercase Latin symbols

b	number of packets that are acknowledged by a single acknowledgment, parameter of TCP
$cwnd_{max}$	maximum congestion window size, parameter of TCP
$d_{k,m}$	propagation delay along the source-destination-source path traversed by packets of m-th flow of k-th aggregate
d_q	queuing delay at bottleneck router
ln	natural logarithm
$maxTh_g$	minimum threshold for dropping function for green packets
$maxTh_r$	minimum threshold for dropping function for red packets
$maxTh_y$	minimum threshold for dropping function for yellow packets
$minTh_g$	minimum threshold for dropping function for green packets
$minTh_r$	minimum threshold for dropping function for red packets

$minTh_y$	minimum threshold for dropping function for yellow packets
n	number of arriving packets observed during the period $W + T$
n	number of independent observations in simulation process
$n^{\perp}$	minimum possible value of n
$n^{\top}$	maximum possible value of n
$p^{dg}(\cdot)$	function of dropping probability of green packets
$p^{dr}(\cdot)$	function of dropping probability of red packets
$p^{dy}(\cdot)$	function of dropping probability of yellow packets
p^{ed}	packet early drop probability
p-value	probability of obtaining a result at least as extreme as that obtained, used for hypothesis testing
$\hat{p}_i^d$	estimator of i-th sample of the packet drop probability
$\hat{p}_i^{mr}$	estimator of i-th sample of the packet marking probability as red
q	instantaneous size of physical queue
$\bar{q}$	average size of physical queue
$\hat{q}$	estimator of average queue size
$t_{n-1,1-\alpha/2}$	$(1-\alpha/2)$-th quantile of the Student-t distribution with $n-1$ degrees of freedom
w_q	weighting parameter in EWMA algorithm

Uppercase Latin symbols

A	TSW-averaged traffic rate estimated by TSW rate estimator
$A(x)$	probability density function of random variable A
A_k	k-th estimate of the TSW-averaged traffic rate A calculated by TSW rate estimator
B	throughput, average rate of the incoming traffic, expressed in packets per second

B_k	throughput achieved by k-th aggregate
$B_{k,m}$	throughput achieved by individual m-th flow of k-th aggregate
B^{out}	goodput, i.e., traffic served
B_k^{out}	goodput of k-th aggregate
C	notation of the token bucket C
C_{in}	capacity of the input link
CBS	committed burst size
D	throughput, average traffic rate expressed in bits per second
E	notation of the token bucket E
$E[T]$	expected value (mean) of the random variable T (packet inter-arrival time)
$E_{i,j}$	state of the queuing system representing srTCM meaning that the token bucket C is in state i and token bucket E is in state j
EBS	excess burst size
$H(x)$	Heaviside step at $x = 0$
K	maximum size of the queue (in general considerations regarding queuing models
K	number of traffic aggregates in modular model
L^{int}	length of simulation interval
M_k	number of TCP sources constituting aggregate k
P	notation of the token bucket P
P_i^d	overall packet drop probability experienced by a packet if the current number of packets in the queue is i
$P_i^{d,RIO-C}$	overall packet drop probability experienced by a packet in a RIO-C queue if the current number of packets in the queue is i (RIO-C queue)
$P_i^{d,WRED}$	overall packet drop probability experienced by a packet in a WRED queue if the current number of packets in the queue is i

P^{mg}	probability that the packet is marked green
P_k^{mg}	probability that the packet belonging to aggregate k is marked green
P^{mr}	probability that the packet is marked red
P_k^{mr}	probability that the packet belonging to aggregate k is marked red
P^{my}	probability that the packet is marked yellow
P_k^{my}	probability that the packet belonging to aggregate k is marked yellow
P^{qg}	probability that the packet in the queue or in the stream of incoming packets is marked green
P^{qr}	probability that the packet in the queue or in the stream of incoming packets is marked red
P^{qy}	probability that the packet in the queue or in the stream of incoming packets is marked yellow
PBS	peak burst size
$Pmax_g$	maximum drop probability — a parameter for dropping function for green packets
$Pmax_y$	maximum drop probability — a parameter for dropping function for yellow packets
$\bar{P}^d$	overall average packet drop probability imposed by Multi-RED queue mechanism
$\bar{P}_k^d$	packet drop probability experienced by packets belonging to k-th aggregate
$\bar{P}^{dg}$	average drop probability of green packets imposed by Multi-RED queue mechanism
$\bar{P}^{dr}$	average drop probability of red packets imposed by Multi-RED queue mechanism
$\bar{P}^{dy}$	average drop probability of yellow packets imposed by Multi-RED queue mechanism
$\hat{P}^d$	estimator of the packet drop probability

$\hat{P}^{mr}$	estimator of the packet marking probability as red
Q_{LB}	Ljung-Box test statistic
Q_{max}	maximum queue size
Q_{ML}	McLeod-Li test statistic
RTO	retransmission timeout
$RTO_{k,m}$	retransmission timeout of individual m-th flow of k-th aggregate
RTT	round trip time
$RTT_{k,m}$	round trip time experienced by individual m-th flow of k-th aggregate
S	packet size
S_{min}	minimum packet size
T	packet inter-arrival time — random variable
$\boldsymbol{T}(x)$	cumulative distribution function of random variable T
T^{avg}	random variable representing the average inter-arrival time calculated from n observations within the period $W + T$
$T^{\text{avg}}(x)$	probability density function of random variable T^{avg}
T_c	token count for token bucket C
T_e	token count for token bucket E
T_p	token count for token bucket P
W	observation window size in TSW rate estimator

Part I

Introduction and background

1 Introduction

Transmission of documents via telephone wires is possible in principle, but the apparatus required is so expensive that it will never become a practical proposition.

— Dennis Gabor

Telecommunications networks have significantly changed since the year 1962 when Dennis Gabor (British physicist and author of *Inventing the Future*) stated the above sentence. Not only the transfer of documents became possible but invention of the Internet and fibers as well as development of electronics enabled long distance transfer of very large files, images, and video. Engineers and researchers around the world currently face completely different problems in telecommunications area. One of the issues is assurance of a satisfactory level of services over the Internet.

Rapid development of the Internet and modern network applications caused a significant growth of the traffic in the Internet and imposed new demands on the network services. Many existing and emerging applications are very sensitive to the transmission delay and jitter and usually require guaranteed high capacity bandwidth. Real time applications such as Video on Demand, Voice over Internet Protocol, video conferencing, and others are a good example. One of the major Internet applications include also Virtual Private Networks (VPN) that require the traffic to be transported reliably and with a guaranteed speed. Even applications having less strict requirements need a network service of a well-defined level of quality. Internet supporting only best effort services is insufficient to meet such high and diverse requirements. Since network resources are finite even a momentary bandwidth scarcity in some areas of the Internet can cause a significant decrease in quality of some services. Thus the need for the development of appropriate mechanisms for service quality assurance has emerged.

A lot of people and organizations around the world devote their attention to service quality and the effects of their work can be easily seen. Various network protocols and architectures supporting the quality of service assurance are available now and are being still developed.

The most mature network architectures supporting service quality assurance in IP networks are the Integrated Services model (IntServ) [20] and the Differentiated Services model (DiffServ) [14]. The former solution is not used in core networks mainly due to scalability problems. Its applicability is limited to access networks. The DiffServ model was proposed to overcome several drawbacks of IntServ. Although it has some disadvantages, it addresses various quality of service requirements and is being deployed in real networks. DiffServ is supported by network equipment delivered by several leading vendors. Network architectures supporting quality of service at the IP level are not limited to IntServ and DiffServ.

The most recent solutions are Flow Aware Networking (FAN) [129, 130] and Flow State-Aware Routing (FSA) [2]. They are at an early stage of development. They are based on a different concept and are intended to be simple and to overcome drawbacks of IntServ and DiffServ.

Practical implementation of mechanisms supporting service quality assurance is not easy. Selection of particular mechanisms and configuration of network elements guaranteeing high network performance under various traffic conditions is not straightforward. Intuitive setting of configuration parameters is often not sufficient. In the case of DiffServ, many studies have shown that finding a nodes' configuration resulting in fair service differentiation is difficult. The above problems are, among other issues, the driving factors in analytical modeling of network elements. Analytical models help in better understanding of mechanisms and phenomena occurring in networks. They facilitate examination, planning and configuration of networks.

1.1 Scope and thesis

This dissertation proposes analytical models of DiffServ network elements supporting the Assured Forwarding (AF) service [66] in a DiffServ network. Simple analytical models for main network elements: meter/markers and droppers are provided. Accuracy of the models is verified by simulations. The models enable quick finding of meter/marker and dropper characteristics under a particular configuration and traffic volume. They can be also used for quick checking how parameter manipulations affect the characteristics. Additionally, these models together with traffic models facilitate predicting characteristics of the whole system. Example for a selected TCP throughput model is also presented.

The following thesis of this dissertation has been proposed and proved:

> ***It is possible to analytically express characteristics of meter/markers and droppers used for DiffServ network supporting Assured Forwarding PHB.***

The proposed models were intended to be very simple and not requiring high computational expenditure. This aim was achieved for models of individual network elements. In the case of a modular model consisting of models for several network elements and the model for TCP sources, the mathematical closed form solution cannot be found. However, it is shown that the solution can be found numerically.

1.2 Publications

Some of the achievements presented in the dissertation were published in one journal article and three conference papers. The list of relevant publications is as follows:

[59] J. Gozdecki, R. Stankiewicz, and A. Jajszczyk. Quality of service terminology in IP networks. *IEEE Communications Magazine*, 41(3):153–159, Mar. 2003.

[140] R. Stankiewicz and A. Jajszczyk. Modeling of TCP behavior in a DiffServ network supporting assured forwarding PHB. In *Proc. IEEE ICC 2004*, volume 4, pages 2071–2075, Paris, France, Jun. 2004.

[141] R. Stankiewicz and A. Jajszczyk. Analytical models for DiffServ meter/markers. In *Proc. IEEE ICC 2005*, volume 1, pages 158–162, Seoul, Korea, May 2005.

[142] R. Stankiewicz and A. Jajszczyk. New mathematical models for token bucket based meter/markers. In G. Parr, D. Malone, and M. Ó. Fohlú, editors, *Proc. 6th IEEE International Workshop on IP Operations and Management (IPOM)*, volume 4268 of *Lecture Notes In Computer Science*, pages 96–107, Dublin, Ireland, Oct. 2006. Springer-Verlag.

Article [59] is a tutorial overview and clarification of QoS related terminology. Several approaches to QoS definition, including those of main standardization bodies such as ETSI, ITU and IETF are presented. Also, terminology related to IntServ and DiffServ QoS architectures is provided. Some issues discussed in the paper are also presended in the dissertation in Section 2.1 to introduce important background information for the research.

Paper [140] presents a modular model of a DiffServ network composed of meter/markers and droppers. The traffic is generated by TCP sources. The first versions of models for srTCM meter/marker and WRED type of Multi-RED queue serving as a dropper are presented and successfully verified. Results of three experiments with various configurations are provided. The idea of a modular model presented there is used in the dissertation (Section 4.5).

A mathematical model of srTCM as well as models for trTCM and TSW3CM are presented in [141]. All the models are validated by simulations with traffic generated by ftp-like TCP sources. Models of token bucket based meter/markers, namely srTCM and trTCM, are based on the M/M/1/K queuing system. The model of TSW3CM presented there is in its final version that is also presented in the dissertation. However, it is enriched with boundary characteristics in the dissertation.

Improved models of srTCM and trTCM are provided in [142]. In the case of srTCM only minor changes are made to its M/M/1/K based model to remove drawbacks of the previous version. A new, D/M/1/K based, model of trTCM in most cases performing better than the one based on M/M/1/K is presented. Both improved models are validated under two types of traffic: ftp-like and web-like.

1.3 Structure of the dissertation

The dissertation is organized into three parts. The first part (chapters 1 – 3) gives a theoretical background for the research. Chapter 1 serves as a general introduction. In Chapter 2, the terminology related to quality of service in an IP network is provided. It also gives a basic information about the DiffServ model and implementation of Assured Forwarding Per Hop Behavior and practical realization of DiffServ network elements supporting that class of service, namely meter/markers and droppers. Chapter 3 provides an overview of the literature related to research on DiffServ performance evaluation and modeling.

The second part of the dissertation (chapters 4 – 5) presents the results of PhD candidate's research. Analytical models of selected realizations of meter/markers and droppers are introduced and explained in Chapter 4. Section 4.1 presents the assumptions. Models for meter/markers are developed in Section 4.2. Section 4.3 presents analytical models for implementations of droppers. The model for TCP throughput is presented in Section 4.4. Finally, the modular model, based on the models for meter/markers, droppers, and TCP throughput is developed in Section 4.5. In Chapter 5, experimental validation of the proposed analytical models is presented. The technique for performing the experiments, including discussions on tools, the simulation technique and credibility, methods of data analysis and the simulated network architecture are provided in Section 5.1. The following sections present results of the experiments for meter/markers (Section

5.2), droppers (Section 5.3) and modular model (Section 5.4). Comparison of experimental data to the characteristics predicted by the analytical models together with the related discussion is provided for each experiment. The content of this part, except for the model for TCP throughput presented in Section 4.4 and some elements of the model of droppers in Section 4.3, is the original work of the author.

The third part of the dissertation consists of only one chapter. Chapter 6 summarizes the research presented in the dissertation.

The attached appendix provides examples of the technique used to verify the credibility of the experiments.

2 Area of research

QoS is quite often stupid.

— James W. Roberts

The aim of this chapter is to introduce notions, terminology and fundamentals related to the subject of the dissertation. Terminology related to the quality of service is overviewed and clarified. The concept of the Differentiated Services model is described. Issues related to Assured Forwarding service realization is discussed in a greater detail.

2.1 Quality of Service – terminology and definitions

One of the side effects of the common interest in quality of service assurance is a growing havoc in the quality of service related terminology. There are a lot of terms and definitions that are often incoherent and used inappropriately. It is clear that the confusion should be eliminated. This section provides a short overview and clarification of the terminology and definitions. It is based on approaches of the main standardization bodies. Since DiffServ that is dealt with in this dissertation is proposed by Internet Engineering Task Force (IETF) it is natural that the IETF's terminology is adopted and used throughout the dissertation. However, to clarify the terminology this section presents also quality of service related notions defined by International Telecommunication Union (ITU), and European Telecommunications Standards Institute (ETSI).

2.1.1 Quality of Service

Quality of Service (QoS) can be defined and understood in many ways depending on the point of view. The evaluation of QoS depends on various criteria related to the party rating the service. Customers assess it on the basis of a personal impression and in comparison to their expectations, while an engineer expresses the quality in terms of technical parameters. Service providers usually have even a different point of view. This discrepancy may sometimes lead to misunderstandings. Hence, the term QoS is used in many meanings ranging from the user's perception of the service to a set of connection parameters necessary to achieve a particular service quality. This problem is also reflected in the literature.

As presented in [59], various points of view on QoS and the ITU, ETSI and IETF approaches can be compared and reconciled with a reference to the general QoS model proposed by Hardy [65]. Hardy introduces three notions of QoS: *Intrinsic*, *Perceived* and *Assessed* QoS (Fig. 2.1).

General Model

Intrinsic QoS pertains to service features stemming from technical aspects. Thus, the intrinsic quality is determined by a transport network design and provisioning of network access, terminations and connections [65]. The required quality is achieved, among other things, by an appropriate selection of transport protocols, the quality of service assurance mechanisms and related values of parameters. The intrinsic QoS is evaluated by the comparison of measured and expected performance characteristics. User's perception of the service does not influence the intrinsic QoS rating.

Perceived QoS reflects the customer's experience of using a particular service. It is influenced by the customer's expectations compared to observed service performance. In turn, the personal expectations are usually affected by customer's experience with a similar telecommunications service and other customers' opinions. Thus, the quality of service with the same intrinsic features may be perceived by various customers differently. It follows that just ensuring particular service (network) parameters may not be sufficient to satisfy customers who are not concerned with how a service is provided. The quality of the service offered by a provider must reflect the intrinsic QoS as well as some non-technical parameters that are meaningful to the customer and relevant to particular community's expectations.

Assessed QoS starts to be seen when the customer decides whether to continue using the service or not [65]. This decision depends on the perceived quality, service price and responses of the provider to submitted complaints and problems. It follows that even a customer service representative's attitude to a client may

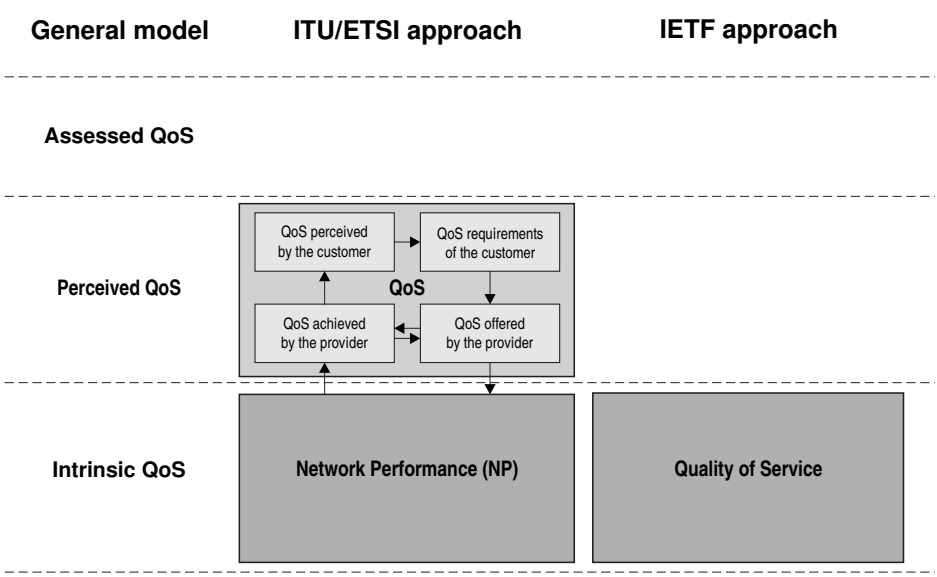


Figure 2.1: General QoS model and ITU/ETSI and IETF approaches

be an important factor in rating the assessed QoS. Neither ITU nor ETSI nor IETF deal with the assessed QoS.

The assurance of a satisfactory level of intrinsic, perceived and assessed QoS may be considered separately. The first is the responsibility of a network provider and depends on network architecture, planning and management. It is mainly a technical problem dealt with by engineers, designers and operators. An appropriate use of the intrinsic QoS capabilities adjusted to a particular service offered, together with market analysis, are necessary to ensure a high level of the perceived QoS. This is the duty of the service provider. Advertising and marketing efforts have an impact on the perceived QoS as well. The assessed QoS mainly depends on the charging policy of a provider as well as reliable customer service representatives and technical support.

IETF approach

IETF focuses on the intrinsic QoS and does not deal with the perceived QoS. It stems from the main objectives of IETF that is concerned with the Internet architecture and its development, dependability and effectiveness. QoS is understood by IETF as *"a set of service requirements to be met by the network while transporting a flow"* [35]. QoS is defined and measured in terms of parameters describing customer's traffic and parameters of particular network components

involved in providing a service that are the key for network efficiency and effectiveness in providing a service.

ITU/ETSI approach

The ITU and ETSI approaches to the quality of service related terminology are almost the same (compare ETSI [45] and ITU [73, 74, 75, 78] documents). They use the same basic definition of QoS expressed for the first time in [73] as *"the collective effect of service performance which determine the degree of satisfaction of a user of the service"*. As this definition suggests, the QoS term in the ITU/ETSI approach adheres mainly to the perceived rather than to the intrinsic QoS. Besides, they introduce the notion of Network Performance (NP) to cover technical facets. They make a clear distinction between QoS, understood as something focused on user-perceivable effects, and NP encompassing all network functions essential to provide a service. QoS parameters are user-oriented and they do not directly translate into network parameters. On the other hand, the network performance parameters determine the quality observed by customers but are not necessarily meaningful to them [75]. But there must exist a consistent mapping between the QoS and NP parameters.

Network performance, as mentioned above, corresponds to the intrinsic QoS. It is defined in [73] as *"the ability of a network or network portion to provide the functions related to communications between users"*. It is closely equivalent to the notion of QoS defined by IETF and it is defined in terms of parameters. A high level of NP is achieved by an appropriate system design, configuration, operation and maintenance [45]. Some network performance parameters are defined by ITU in [73] and [78].

QoS parameters

The intrinsic QoS in packet networks is expressed by at least the following set of parameters that are meaningful for most IP-based services:

- Bit rate of transferring user data available for the service or target throughput which may be achieved.
- Delay experienced by packets while passing the network. It may be considered either in an end-to-end relation or with regard to a particular network element.
- Jitter — variations in the IP packet transfer delay. Again, it can be applied to an end-to-end relation or a single network element.
- Packet loss rate usually defined as the ratio of the number of undelivered packets to sent ones.

These parameters describe the treatment experienced by packets while passing the network. They can be translated into particular parameters of the network architecture components used to assure QoS. They are finally mapped into configuration of network elements. They are also closely connected with protocols used in the network and equipment abilities.

Additionally, the intrinsic QoS may have the following attributes dependent on the network architecture as well as the application demands:

- end-to-end (e.g., in the IntServ model) or limited to a particular domain or domains (e.g., in the DiffServ model),
- applied to all traffic or just to a particular session or sessions,
- unidirectional or bi-directional,
- guaranteed or statistical.

QoS is usually an end-to-end characteristic of communication between end hosts. It should be assured along the whole path between peers but the path may cross several autonomous systems belonging to various network providers. Then performance of all the autonomous systems contributes to the final service quality.

2.1.2 Class of Service

The Class of Service (CoS) is a broad term describing a set of characteristics available with a specific service. Both IETF and ITU-T define the CoS term. It is defined by IETF as *"The definitions of the semantics and parameters of a specific type of QoS"* [35]. The ITU-T definition of the CoS term can be found in [72]. Services belonging to the same class are described by the same set of parameters, which can have qualitative or quantitative values. Usually, the set of parameters within the class is defined without assignment of concrete values, but these values can be bounded.

The idea of service classification is relatively mature. For example, the original IP was intended to provide a simple way of classifying packets but this capability of IP is rarely used. Traffic in ATM networks is divided into classes as well. Currently, concrete service classes are defined within IP-QoS architectures proposed by IETF, such as IntServ and DiffServ. The following three classes are defined within the IntServ architecture: guaranteed, controlled load and best-effort. Also, in DiffServ three classes were initially defined (olympic, premium and best effort) but this classification is currently of historical importance. Examples of current service classes defined by IETF are Assured Forwarding and

Expedited Forwarding within the DiffServ model. They will be discussed in Section 2.2.3. ITU-T introduced an IP transfer capability term which is related to CoS [76]. The following three transfer capabilities are defined:

- Dedicated Bandwidth (DBW) IP transfer capability,
- Statistical Bandwidth (SBW) IP transfer capability,
- Best-effort (BE) IP transfer capability.

Each IP transfer capability is characterized by the service model, traffic descriptor, conformance definition and any QoS commitments. IP transfer capabilities strive for compatibility with CoS defined in IETF QoS architectures.

2.1.3 A contract between the service provider and the customer

IETF defines Service Level Agreement (SLA) as *"a service contract between a customer and a service provider that specifies the forwarding service a customer should receive"* [14]. SLA should be expressed in a way intelligible to a customer. It encompasses basic features of the service and well-defined unambiguous criteria of assessing whether the service delivered is consistent with the contract. On the other hand, limits imposed on the customer must be clear. SLA must consist of responsibility rules for breaking the contract by the service provider as well as by the customer. Regarding the IETF definition, SLA may also include traffic conditioning rules which, at least in part, constitute a Traffic Conditioning Agreement (TCA).

In compliance with the ITU definition, SLA is *"a negotiated agreement between a customer and the service provider on levels of service characteristics and the associated set of metrics. The content of SLA varies depending on the service offering and includes the attributes required for the negotiated agreement"* [77]. SLA may be in form of a document containing names of the parties signing the contract. According to [77] it should be composed of service level objectives, service monitoring components and financial compensation components. Service level objectives encompass QoS parameters or class of the service provided, service availability and reliability, authentication issues, the SLA expiry date, etc. Service monitoring specifies the way of measuring service quality and other parameters used to assess whether the service complies with the SLA. It may also include an agreement on form and frequency of delivering the report on service usage. The financial component may include billing options, penalties for breaking the contract, etc [77].

Summarizing, SLA is a broad term encompassing technical features and parameters of the service as well as legal and charging aspects. SLA parameters and attributes define the IP-based service.

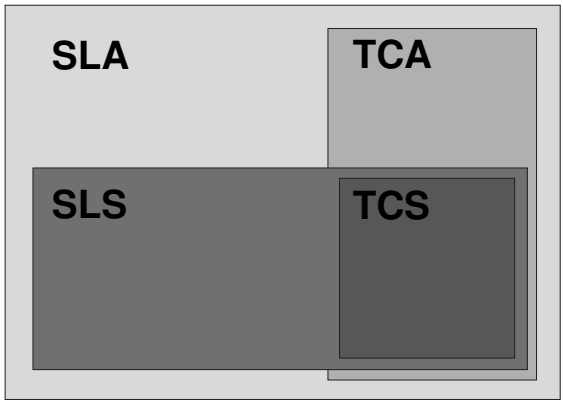


Figure 2.2: Interrelation between SLA, SLS, TCA and TCS

The notion of Service Level Specification (SLS) was introduced to separate a technical part of the contract within SLA. It is defined as *"a set of parameters and their values which together define the service offered to a traffic"* [61]. In other words, it specifies a set of values of network parameters related to a particular service. The IP transport services are technically described by SLSs.

According to the IETF definition, Traffic Conditioning Agreement (TCA) *"encompasses all of the traffic conditioning rules explicitly specified within a SLA along with all of the rules implicit from the relevant service requirements and/or from a DiffServ domain's service provisioning policy"* [14]. In brief, TCA is an agreement specifying packet classification rules and traffic profiles. Traffic profile is a description of the temporal properties of a traffic stream. It can be defined, for example, in terms of token bucket parameters such as a rate and a burst size. The customer is obliged to adjust the generated traffic streams to a contracted profile. In order to enforce customer's traffic conformance to the profile, particular metering, marking, discarding and shaping rules may be defined. If the traffic profile is defined in terms of token bucket parameters, out-of-profile packets are those which arrive when an insufficient number of tokens is available in the bucket. The treatment of out-of-profile packets is also specified by TCA. They may be marked and treated in a different way than in-profile packets or discarded. The concept of in- and out-of-profile can be extended to multiple levels.

Traffic Conditioning Specification (TCS) is a set of parameters with assigned values which unambiguously specify a set of classifier rules and a traffic profile. TCS is a technical part of TCA. TCS is also an integral element of an SLS [61].

Interrelations between SLA, SLS, TCA and TCS are shown in a simplified way in Fig. 2.2 [59].

2.2 Differentiated Services

Differentiated Services model [14] is a solution for providing different levels of service quality in IP networks. It is a framework within which service providers can offer their customers a variety of network services. It offers definitions of service classes, several network building blocks and proposes how to use them. Support for the mechanisms must be implemented in a network equipment. Practical use of DiffServ mechanisms is left to network providers that can built the network and configure its elements in a way assuring realization of the services offered.

2.2.1 Model

The basic assumption of the DiffServ model was to achieve the scalability. The key factors for the DiffServ network scalability are a limited number of service classes served in the network and a simplified functionality of the core nodes accomplished by placing as much of the complexity as possible in the edge routers. This idea is based on the fact that the edge node usually has to deal with lower traffic volumes and lesser number of flows than the core node. Edge routers process individual flows according to a negotiated SLA and assign them into a limited number of service classes. The processing may encompass admission control mechanisms, traffic shaping and dropping. Core routers process packets on the basis of their class membership and, what follows, deal with traffic aggregates rather than individual flows.

A class membership of the packet is unambiguously denoted by the value of Differentiated Services Code Point (DSCP or codepoint) [14]. Six-bit long DSCP is a part of Differentiated Services Field [115] carried in the IP header: the Type of Service (TOS) byte in IPv4 [128] or the Traffic Class byte in IPv6 [38]. Stemming from the length of DSCP, the maximum number of service classes that can be distingushed in the network is 64. Recommended mapping between DSCP values and service classes is defined by respective IETF RFCs.

Assignment of a DSCP to a packet, also called packet marking, is performed at an ingress boundary router. All packets marked with the same DSCP constitute Behavior Aggregate (BA). Packets (flows) of the aggregate are treated in the same manner while passing the network. Forwarding rules applied to a BA at a particular DiffServ compliant node are called Per Hop Behavior (PHB). They have local meaning and can differ between nodes. An additional term, PHB Group, was introduced to distinguish a set of PHBs sharing a common constraint applying to all PHBs within the group. If a constraint within a PHB Group is that ordering of at least those packets belonging to the same microflow must be preserved, the PHB Group is also called PHB Scheduling Class (PSC) [61]. A set of Behavior Aggregates sharing the ordering constraint that ordering of at least those packets belonging to the same microflow must be preserved is called

Ordered Aggregate (OA). Currently, the terms: PHB Group, PSC, and OA apply to the Assured Forwarding PHB Group (see Section 2.2.3).

2.2.2 Network architecture

The area of the network that supports DiffServ mechanisms is called DiffServ domain. It is a contiguous set of DiffServ capable nodes supporting a common service provisioning policy. A consistent set of PHBs is implemented in the nodes in a way assuring QoS guarantees with respect to SLAs. A DiffServ domain has a well-defined boundary consisting of DiffServ boundary nodes. Ingress nodes classify and possibly condition the incoming traffic. In-profile packets are alway assigned a DSCP and classified to the respective PHB. Out-of-profile packets are either marked with a different DSCP or dropped. Incoming traffic may also be shaped. Egress boundary nodes may also be required to perform additional functions to force the outgoing traffic conformace to the TCA with peering domain (either DiffServ capable or not). Boundary nodes usually act as both ingress and egress nodes. Interior nodes, also called core nodes, rely on packet's DSCP and apply PHB rules to the packet accordingly. They may also perform shaping and conditioning functions. Out-of-profile packets may be either forwarded or discarded if the outgoing link is congested. A sample DiffServ domain is illustrated in Fig. 2.3.

A set of one or more contiguous co-operating DiffServ domains constitutes a DiffServ region. The DiffServ domains in a DiffServ region may support different PHB groups internally. A mapping between codepoints and PHBs may also differ. All peering DiffServ domains must establish peering SLAs which define (either explicitly or implicitly) TCAs which specify how transit traffic from one DiffServ domain to another is conditioned at the boundary between the two DiffServ domains [14].

If a DiffServ domain is considered as a network cloud that the traffic is crossing, the Per Domain Behavior (PDB) [116] may be considered. PDB defines the treatment that packets will receive while in transit, edge-to-edge, in a DiffServ domain. It is constructed on the basis of one or more PHBs supported within the domain. The concept of PDB makes it easier to compose cross-domain services as well as making it possible to hide details of network's internals while exposing information sufficient to enable QoS. Measurable parameters of a PDB should be reflected in SLA. An example of PDB can be found in [15].

Provisioning of the resources as well as configuration of PHBs within a DiffServ domain may be either static or dynamic. In the former case, a signaling is unnecessary but as service provisioning is inflexible, there is a limited capability to offer a large variety of SLAs. A support for a dynamic SLA is provided by a Bandwidth Broker (BB). It offers a logically centralized admission control mech-

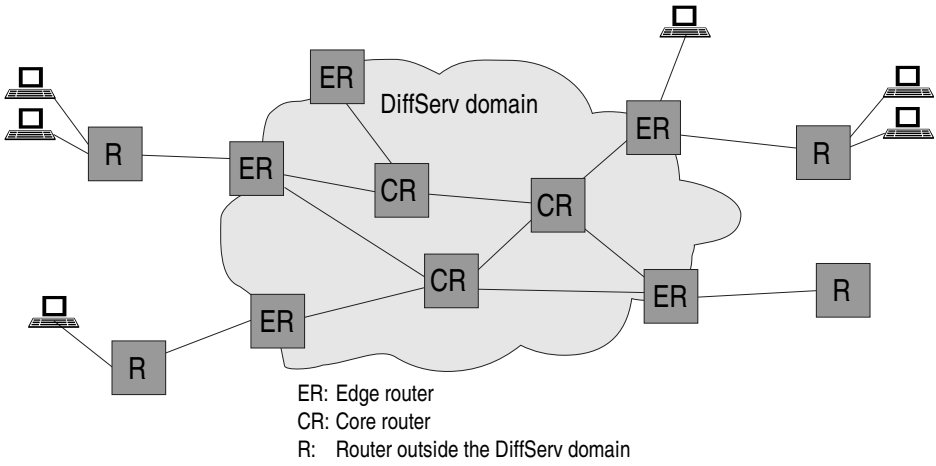


Figure 2.3: DiffServ domain

anism serving the domain. The concept of BB was initially introduced in [117]. Since the resource management is centralized in this concept, the nodes do not need to perform admission control decisions but using signaling if necessary. BB must be aware of a current network state. If BB grants admission for the traffic subject to a particular SLA, it has to configure ingress and egress nodes. It may even have to configure core nodes [134].

2.2.3 Service classes

As mentioned before, DiffServ is not intended to provide guarantees for individual flows. Instead, it provides statistical guarantees to a limited number of predefined service classes. Currently, the DiffServ architecture defines two basic PHBs beyond the best-effort service: Expedited Forwarding PHB (EF) [37] and Assured Forwarding PHB Group (AF) [66]. Forwarding behavior corresponding to the best-effort service is called Default PHB. An additional PHB, called Class Selector (CS) [115], is defined for backward compatibility with a legacy use of IP precedence bits.

Assured Forwarding PHB group

Assured Forwarding PHB is intended to provide the assurance of a minimum throughput even when the network is congested. It means that an aggregated stream should always be able to achieve a guaranteed minimum throughput and

is allowed to consume some part of an excess bandwidth when the network is underutilized. The minimum throughput an aggregate is guaranteed to receive is often called target or contract rate. The bandwidth remaining while the network load is low is intended to be shared by competing traffic aggregates in a fair manner. AF is suitable to most TCP-based applications and, in general, for elastic in nature, bursty traffic requiring low loss assurances.

According to [66], the AF PHB group consists of four AF classes with three drop precedences each. Thus, twelve codepoints, that is twelve PHBs, are used. PHBs within a single AF class share an ordering constraint and constitute PSC. Distinction of three drop precedences within an AF class was introduced to make the implementation of the in/out-of-profile packets concept possible. In an implementation, one can use two or three levels of drop precedence. The notion of packets' colors was introduced. Green packets correspond to in-profile packets, that is, fully conforming to the contract. Yellow and red colors usually reperesent out-of-profile packets but two levels reflecting to what extent the contract is broken are used. Packet marking is then often called packet coloring.

Decision of classifying a packet as in- or out-of-profile is key for AF PHB performance. It is taken at the ingress router on the basis of current traffic characteristics. The two types of entities performing metering and marking packets, based on the token bucket mechanism or based on estimation of the transient traffic rate, will be discussed in Section 2.3.1.

Each AF class is managed independently of other classes and is serviced in a separate physical queue while the drop precedences usually correspond to different thresholds in the Multi-RED (Random Early Detection) [53, 100] queue management scheme. Mapping of all packets belonging to the AF class to a single physical queue forces the preservation of the packet order. Multi-RED queue management allows different treatment of in- and out-of-profile (or green, yellow and red) packets. In general, if the link is lightly loaded all packets are forwarded. In presence of congestion, red packets are dropped with a certain probability first. If the congestion increases the drop probability increases as well and yellow packets may become dropped. It is assumed that in a well configured network green packets are never discarded but, in practice, there is such a possibility.

Expedited Forwarding PHB

The EF PHB is intended to be used to build a low loss, low latency, low jitter, assured bandwidth, end-to-end service through DiffServ domains [37]. Such a service is sometimes referenced to as *premium service* [117]. It is targeted at so called inelastic, real-time applications such as Voice over IP, Video on Demand, video conferencing, etc. Such applications can accept some probability of packet loss but require a minimized delay and delay variation. The main reason for high packet delays and jitter are queuing delays. To avoid large, accumulated queues

it should be assured that the arrival rate of the packets is lower than their departure rate. In an implementation, a separate physical queue is dedicated to the EF traffic. Sufficient resources guaranteeing a certain departure rate can be allocated to the EF queue by various scheduling disciplines such as Priority Queuing (PQ) [150], Weighted Fair Queuing (WFQ) [13, 39], Class-Based Weighted Fair Queuing (CBWFQ) [150], Weighted Round Robin (WRR) [82] or Deficit Round Robin (DRR) [138]. It is also required to perform strict policing and shaping at ingress nodes so as the EF traffic never exceeds its maximum rate at any hop in the network.

Class Selector PHB

There are eight precedence classes within the Class Selector PHB group. The idea is that a higher class gives a higher or equal propability of timely forwarding than the lower class. In contrast to the AF PHB group, packets marked with different CS codepoints may be re-ordered (there is no ordering constraint). CS PHB can be realized by various mechanisms, including priority queuing (PQ), weighted fair queuing (WFQ), Class-Based Queuing (CBQ), or others [115]. As described in [40], CS PHB can be used for implementing relative service guarantees.

2.2.4 DiffServ node components

The DiffServ architecture consists of many elements allowing realization of various PHBs. The DiffServ node is described in [14] as a combination of five elements (Fig. 2.4): classifier, meter, marker, shaper and dropper. The last four elements constitute the traffic conditioner. Another device, called scheduler, that supports resource sharing among several classes completes the list of the DiffServ node building blocks. A DiffServ node may be composed of all elements or only a subset depending on the role the node plays in a DiffServ domain. Functionality of the elements is briefly described below.

Classifier

The task of a traffic classifier is to select a packet in a traffic stream based on some data carried in the packet's header and assign it to one of the service classes supported in the network. This is the first step necessary for processing packets, especially in the ingress router. The knowledge of the packet class membership is necessary to apply appropriate metering, marking, shaping and dropping functions according to the SLA.

There are two types of classifiers: multifield (MF) and behavior aggregate (BA) classifiers. The former may use various information from the packet header including the source and/or destination address, transport protocol, source port

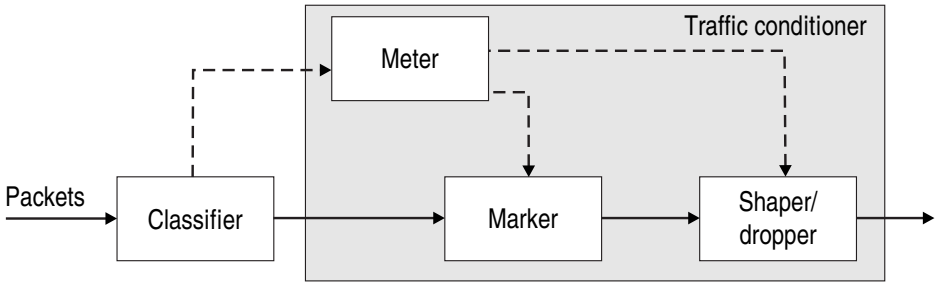


Figure 2.4: Logical model of DiffServ node components

and destination port numbers, codepoint, incoming interface, and others. A combination of one or more fields is used. In the case of the BA classifier only a codepoint is taken into account while deciding which class a packet belongs to.

Meter

A meter is responsible for checking the compliance of the traffic to the traffic profile specified by the contract between an operator and a customer (TCA). It measures temporal properties of the traffic and passes the result to other node elements, mainly to a marker. A single meter entity serves a single traffic aggregate selected by a classifier. In a very spacial case a meter may serve a single flow.

Marker

A marker sets a DSCP of a packet, based on the information from a classifier and meter. A packet is assigned to a particular class and may be marked as in- or out-of-profile, that is, it is assigned to a particular BA. It, in fact, determines the treatment the packet receives while passing the domain.

In practice, the functionality of a meter and marker is implemented in a combined meter/marker device. Examples of meter/marker implementations for AF PHB will be presented in Section 2.3.1.

Shaper

The function of a shaper is to enforce conformance of the traffic stream to a traffic profile. A shaper may delay some or all of the packets in order to shape the stream. This function usually smooths the traffic and reduces burstiness. In implementations, finite buffers are usually used so packets may be discarded if the buffer is full. It may be considered as a way to reduce the average rate of

the traffic. Traffic shaping is especially performed at egress nodes connected to another peering DiffServ domain.

Dropper

A dropper is also responsible for bringing the traffic stream into compliance with the traffic profile but, in contrast to the shaper, it does so by "intelligent" dropping of some packets. Its functionality is realized by a selected active queue management (AQM) technique. The most prominent AQM mechanism is the Random Early Detection (RED) algorithm [53] designed for cooperation with TCP flows. If the congestion starts building up, randomly chosen packets are dropped affecting some TCP flows of the aggregate and forcing them to slow down their sending rates and, finally, to avoid congestion. Extended versions of the RED algorithm, called in general Multi-RED algorithms or Multi-RED queues are commonly used as implementation of a dropper for AF PHB. Two or three virtual queues are usually maintained, each for one drop precedence of a single AF class. Packets enter the queue or are dropped depending on their drop precedence, estimated average queue size and configuration parameters that determine the packet drop probability as a function of the average queue size. Separate sets of parameters maintained for each packet's color assure that packets not conforming to the profile are dropped with a higher probability than packets conforming to the profile. Variants of Multi-RED algorithms will be described in a greater detail in Section 2.3.2.

Scheduler

A core router may serve more than one PHB, e.g., traffic aggregates of several AF, EF and other classes may share the same network resources on the outgoing link. Therefore, separate physical queues are required for each class. In such a case a scheduler is used to define the rule of emptying the queues and, what follows, how the link capacity is shared by traffic aggregates. Examples of scheduler implementations are: PQ, WFQ, DRR, and others. A logical view of a DiffServ node serving multiple service classes sharing link resources is presented in Fig. 2.5.

2.3 Implementation of AF PHB

The subject of this dissertation is analytical modeling of selected DiffServ node elements supporting implementation of AF PHB. Thus, following the general description of the DiffServ model, architecture, capabilities and node components, the practical implementations of a meter/marker and shaper/dropper are presented.

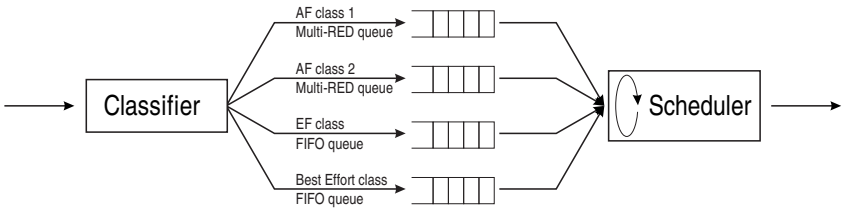


Figure 2.5: Logical view of a DiffServ node serving multiple PHBs

2.3.1 Meter/marker

Most meter/markers realizations can be classified as token bucket based or rate estimator based. The former, uses one or more token buckets that are incremented periodically and decremented on packet arrival. Packets are marked depending on current token buckets' states. Metering and marking functions are usually combined.

Single Rate Three Color Marker (srTCM) [68] and Two Rate Three Color Marker (trTCM) [67] are the most prominent solutions. They are standardized and described in respective IETF RFCs. The simplest meter/marker, i.e., a two color token bucket meter/marker (TB), is a special case of srTCM and trTCM. Traffic Conditioner with Promotion and Fairness Guarantee [30] and Adaptive Token Bucket (ATB) [123] are also good examples of token bucket based meter/markers. Feroz *et al.* proposed TCP-Friendly Marker [49] that supports throughput guarantees for long-lived TCP flows. An interesting solution based on token buckets, called Active Rate Management (ARM) was presented in [29].

Meter/markers from the second group estimate the transient averaged throughput and use the current estimate for marking decision. This idea was initially used in Time Sliding Window Two/Three Color Marker (TSW2CM or TSW3CM) [47]. Since many research results showed several unfairness problems in DiffServ networks, various advanced meter/markers were proposed. Those unfairness problems include unfair sharing of excess bandwidth among aggregates in under-subscribed networks, unfair sharing of bandwith among flows within an aggregate, problems with achieving the target rate by an aggregate in over-subscribed networks, and others. Those problems stem, among other reasons, form heterogeneity of round trip times, packet sizes, target rates, number of flows constituting an aggregate. The simplest solution to address those problems is Improved TSW Three Color Marker (ItswTCM) [145] that differs from TSW3CM in the characteristic of the marking probability as a function of the estimated averaged traffic rate. Additionally, so called intelligent rate estimator based mater/markers using some additional knowledge on the traffic were proposed. Enhanced Time Sliding Window (ETSW) [97] and Intelligent Traffic Conditioner [113] use several

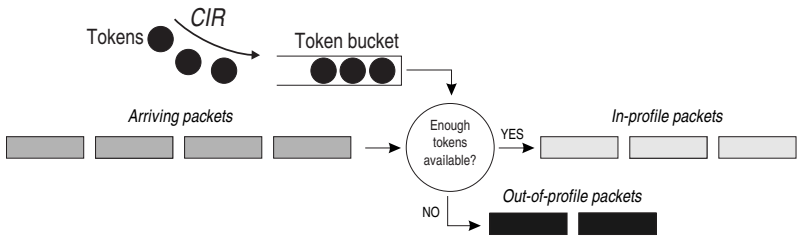


Figure 2.6: Simple token bucket two-color meter/marker

TCP characteristics for computation of the marking probability. Another TCP-friendly meter/marker is RTT and RTO Aware Traffic Conditioner proposed in [64]. The most recent proposal of a meter/marker aware of TCP characteristics is Lightweight Marker [28] that is able to recognize particular flows as well as the phase of a TCP connection and uses several rate estimators. It was designed so as to improve the fairness for short-lived TCP flows. Other rate estimator based meter/markers proposed in the literature are Memory-Based Marker (MBM) [88], Packet Marking Engine (PME) [48], the Adaptive CIR Threshold (ACT) mechanism [31] and Variable Structure Adaptive CIR Threshold (VS-ACT) [33] that, in general, uses not only a current estimate of the averaged throughput for marking decision, but takes into account the past estimates as well. Yeom and Reddy proposed another adaptive marker for aggregated flows [156]. One of the most recent solutions is an adaptive packet marker proposed by Siris and Marinakis [139] whose marking function adapts to changes of the traffic mix.

Another proposal is Equation Based Packet Marking [41, 42] that uses the estimated loss probabilities instead of the estimated averaged traffic rate for marking decision.

The above list of meter/marker realizations encompasses most but not all of the proposals that can be found in the literature. Inventors of a new meter/marker realization usually compare their solution to the previous ones. Comparison of performance of several meter/markers, namely TSW, PME, MBM, ACT, VS-ACT and ARM, can be found in [33].

Token bucket

The simplest two-color meter/marker may be build with a single token bucket (Fig. 2.6). The bucket depth is CBS (Committed Burst Size) and is incremented CIR (Committed Information Rate) times per second. Normally, CBS is expressed in bytes while CIR is expressed in bytes per second. If a packet of size S bytes arrives the bucket is decremented by S, down to the minimum value of 0 and a packet is marked as in-profile or if not enough tokens are available the bucket is not decremented and packet is marked as out-of-profile.

```
If a packet of size S arrives do the following:

if (Tc - S >= 0)
   mark packet as green
   Tc = Tc - S

else if (Te - S >= 0)
   mark packet as yellow
   Te = Te - S

else
   mark packet as red
```

Figure 2.7: Marking algorithm of srTCM in color-blind mode

srTCM

The srTCM consists of two token buckets C and E. Token bucket C is incremented with a rate CIR. Token bucket E is incremented with the same rate but only if token bucket C is full. The rate CIR is expressed in bytes per second. Depths of buckets C and E are CBS and EBS (Excess Burst Size) bytes, respectively. Token buckets C and E are initially (at time 0) full, i.e., the token count $T_c(0) = CBS$ and the token count $T_e(0) = EBS$. The meter/marker may operate in one of two modes: color-blind and color-aware. In the latter case a marker takes into account the current color of the incoming packet while in the former case such information is omitted [68].

As regards the color-blind mode, if a packet of size S bytes arrives and there are enough tokens in bucket C the packet is marked green and the token count T_c is decremented by S. If upon packet's arrival there are not enough tokens in bucket C and there are enough in bucket E the packet is marked yellow and the token count T_e is decremented by S. The packet is marked red if there are not enough tokens in both buckets. The pseudocode of this algorithm for the case of the color-blind mode is also presented in Fig. 2.7. Similarly, the algorithm for the color-aware mode is presented in Fig. 2.8.

trTCM

The idea of trTCM is that a packet is marked red if it arrives with a rate exceeding Peak Information Rate (PIR), yellow if the rate is between CIR and PIR,

```
If a packet of size S arrives do the following:

if (Tc - S >= 0 and the packet is green)
   mark packet as green
   Tc = Tc - S

else if (Te - S >= 0 and the packet is green or yellow)
   mark packet as yellow
   Te = Te - S

else
   mark packet as red
```

Figure 2.8: Marking algorithm of srTCM in color-aware mode

or green if the rate is below or equal to CIR. In practice, trTCM consists of two buckets C and P with depths CBS and PBS (Peak Burst Size) bytes, respectively. Initially, both buckets are full, i.e., $T_c(0) = CBS$ and $T_p(0) = PBS$, respectively. Bucket C is incremented with a rate CIR (bytes per second) while bucket P is incremented with a rate PIR. The fifth parameter of trTCM is its mode: color-blind or color-aware [67].

In the color-blind mode, the packet is marked red if upon its arrival there are not enough tokens available in bucket P. Otherwise, token count T_p is decremented by S and the availability of tokens in bucket C is checked. If there are not enough tokens, the packet is marked yellow. Otherwise the packet is marked green and token count T_c is also decremented by S. This algorithm is also presented in Fig. 2.9. The algorithm for the color-aware mode is similar (Fig. 2.10).

TSW3CM and TSW2CM

In the case of TSW3CM and TSW2CM, components acting as meter and marker may be easily distinguished. The metering function is performed by a rate estimator that estimates the traffic rate averaged over a certain period of time while the marking function is performed by a tagger that marks packets based on the average rate reported by the rate estimator [47].

As stated in [47], any algorithm estimating the averaged traffic rate may be used to implement a rate estimator. The exponential weighted moving average (EWMA) [131] and the time sliding window rate estimation algorithm (also called time-based rate estimation algorithm) [34] are mentioned as examples. However,

```
If a packet of size S arrives do the following:

if (Tp - S < 0)
   mark packet as red
else if (Tc - S < 0)
   mark packet as yellow
   Tp = Tp - S
else
   mark packet as green
   Tp = Tp - S
   Tc = Tc - S
```

Figure 2.9: Marking algorithm of trTCM in color-blind mode

```
If a packet of size S arrives do the following:

if (packet is red or Tp - S < 0)
   mark packet as red
else if (packet is yellow or Tc - S < 0)
   mark packet as yellow
   Tp = Tp - S
else
   mark packet as green
   Tp = Tp - S
   Tc = Tc - S
```

Figure 2.10: Marking algorithm of trTCM in color-aware mode

the latter algorithm is commonly implemented and associated with TSW3CM and TSW2CM.

The TSW rate estimation algorithm presented in Fig. 2.11 estimates the averaged rate of the traffic aggregate the packet belongs to upon arrival of each packet. The estimation is based on the size of the received packet, the time between the arrivals of the current and previous packets and the history over the time window. The algorithm utilizes a low-pass filter decaying function. As shown in [46] this rate estimator decays the past history independently of the traffic stream's packet arrival rate.

```
W      time window size (constant)
t_now  the time of current packet's arrival
t_past the time of previous packet's arrival (initially 0)
A      TSW-averaged traffic rate estimated by the rate
       estimator (initially CIR)
S      the size of current packet

On packet arrival calculate:
    Bytes_in_TSW = A * W
    New_bytes = Bytes_in_TSW + S
    A = New_bytes/(t_now - t_past+W)
    t_past = t_now
```

Figure 2.11: TSW rate estimator algorithm

In the case of TSW3CM, the probabilistic tagger marks arriving packets as green, yellow or red based on the TSW-averaged traffic rate A (averaged traffic rate estimated by the TSW rate estimator) and two constant parameters CIR and PIR. The marking algorithm is presented in Fig. 2.12. The packet is marked green if the TSW-averaged traffic rate, estimated upon its arrival, does not exceed CIR. If the value of A is between CIR and PIR the packet is marked green with probability $\frac{CIR}{A}$ or yellow with probability $1 - \frac{CIR}{A}$. If the current estimate of the TSW-averaged traffic rate exceeds PIR the packet is marked green with probability $\frac{CIR}{A}$, yellow with probability $\frac{(PIR-CIR)}{A}$, or red with probability $1 - \frac{PIR}{A}$.

TSW2CM may be considered as a special case of TSW3CM where $PIR = CIR$. Marking algorithm for TSW2CM is a simplified version of the one for TSW3CM.

The size of the time window W in the TSW rate estimator algorithm is a constant configurable parameter. It should be small enough to reflect the dynamics of the traffic rate. On the other hand, a too small value of the window size causes the estimate of the averaged traffic rate to follow the transient traffic rate closely. Thus, the window size should be large enough to obtain a long-term perspective of the estimated averaged traffic rate. Feng and Clark [34] recommend the window size to have approximately the same value as the round trip time RTT of the TCP flow it the case of a single flow aggregate. Seddigh *et al.* [135] proposed the window size value of 1 second for aggregates consisting of multiple TCP flows of different RTTs. As reported in [47], the 1 second window size is

```
For each packet arrival calculate
estimated averaged traffic rate A and:

if (0 < A <= CIR)
   mark packet as green

else if (CIR < A <= PIR)
   with probability CIR/A mark packet as green
   with probability 1 - CIR/A mark packet as yellow

else if (A > PIR)
   with probability CIR/A mark packet as green
   with probability (PIR - CIR)/A mark packet as yellow
   with probability 1 - PIR/A mark packet as red
```

Figure 2.12: Marking algorithm of TSW3CM probabilistic tagger

also appropriate for aggregates containing UDP flows. It is also recommended to use the window size larger than the largest RTT of TCP flows constituting the aggregate. Research on appropriate configuration of the window size is, however, out of the scope of this dissertation.

2.3.2 Dropper

A dropper based on a selected AQM technique is the second key network component used to implement AF PHB. Most implementations are based on the Multi-RED mechanism with a multiple set of RED parameters applied to packets of different drop precedences. In general, all variants of RED can be classified into four categories (Fig. 2.13) [100]. The two of them, Single Average Multiple Threshold (SAMT) and Multiple Average Multiple Threshold (MAMT) are realizations of Multi-RED. The latter holds multiple virtual queues (adequately to the number of drop precedences supported) and calculates the average queue size for each virtual queue separately. In the SAMT variant virtual queues are not maintained and the average queue size is calculated taking into account all packets regardless of their color. In both, SAMT and MAMT, multiple RED thresholds are used for each drop precedence. These mechanisms enable differentiation of the packet drop probability for each drop precedence.

RED parameters per drop precedence may be configured in the following three ways: overlapped, staggered, and partially overlapped (Fig. 2.14) [100]. The drop

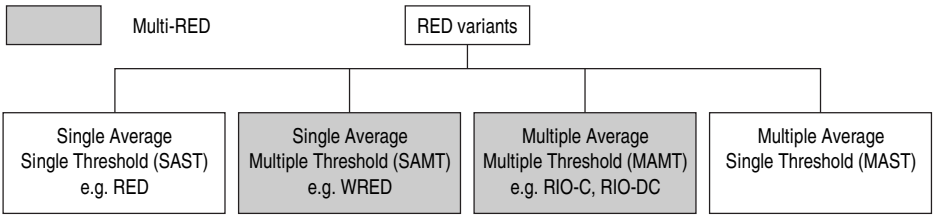


Figure 2.13: Variants of RED

probability curve for a single color is described by three parameters: minimum threshold $minTh$, maximum threshold $maxTh$ and $Pmax$. Letters r, y and g in Fig. 2.14 stand for red, yellow, and green packets, respectively.

The average queue size is estimated by using an exponential weighted moving average (EWMA). It is a low-pass filter. Therefore, the short-term increases of the instantaneous queue size, caused by arrival of bursts of packets or transient congestion, do not cause significant increase in the average queue size estimation. The average queue size is updated on each packet arrival to the queue accordingly to the following formula:

$$avg \leftarrow (1 - w_q)avg + w_q q \tag{2.1}$$

where w_q determines the time constant of the low-pass filter, avg is the estimate of the average queue size, and q is a current size of the queue. The drop probability is calculated for each packet as a function of the latest average queue size approximation and thresholds for the respective color.

Implementation of the SAMT variant is known as Weighted RED (WRED) [150]. There are two popular implementations of the MAMT variant: RIO-C (RED with In/Out and Coupled virtual queues) and RIO-DC (RED with In/Out and De-Coupled virtual queues).

In the case of the WRED queue, a single average queue size is calculated, taking into account all packets in the queue regardless of their color. However, the decision on whether to drop or forward a packet is taken on the basis of different thresholds for packets of different colors.

The RIO queues maintain several virtual queues, as many as the number of drop precedences supported. The RIO-DC queue calculates the average queue size for a particular color counting only packets of that color in the queue. In the case of RIO-C the average queue size for green packets is calculated only from the number of green packets. The average queue size for yellow packets is calculated by counting green and yellow packets, while the average queue size for red packets takes into account all packets in the queue.

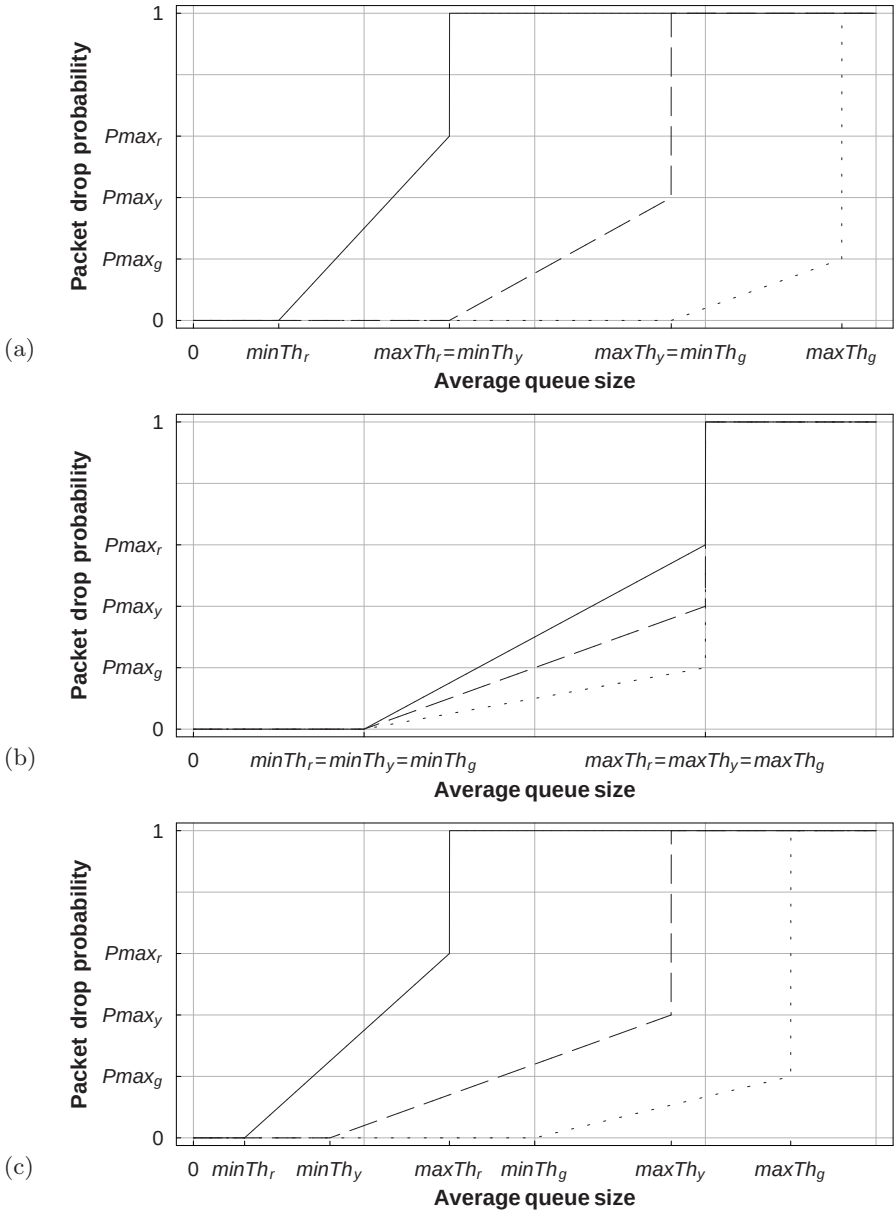


Figure 2.14: Variants of Multi-RED parameter settings: (a) staggered, (b) overlapped, (c) partially overlapped

There are many innovative AQM schemes intended to improve the QoS and fairness guarantees of DiffServ. Most solutions are based on the aforementioned Multi-RED schemes. Orozco and Ros [121] presented the Adaptive RIO (A-RIO) algorithm. It implements the adaptive RED [52] algorithm for all three drop precedences. Several parameters of RED are adaptively adjusted. The thresholds for all precedences are completely overlapped. The authors prove that A-RIO outperforms RIO in terms of stabilizing the queue occupation while maintaining a high QoS throughput and protection of high priority packets. Liu *et al.* [98] proposed two adaptive RED schemes, called ARIO-D and ARIO-L, optimized to improve delay or loss guarantees, respectively. In the case of ARIO-D the parameter $Pmax$ for out-of-profile packets is adaptively changed while keeping $maxTh$ constant. In ARIO-L, $Pmax$ for out-of-profile packets remains constant but $maxTh$ is adaptively adjusted. Chang and Muppala [32] proposed an algorithm called Enhanced RIO that alleviates the impact of aggressive non-adaptive flows. Out-of-profile packets of adaptive and non-adaptive traffic are mapped to different virtual queues. This algorithm also uses adaptive thresholds. Dynamic RIO (DRIO) and (r,RTT)-adaptive [97] are also examples of enhanced AQM mechanisms based on RIO. Those techniques require maintaining per-flow information of state at core routers.

Some researchers proposed a more comprehensive solutions improving the performance of DiffServ. They developed new meter/markers together with new AQM mechanisms interworking with each other. The authors of [5] proposed a Fair Traffic Conditioner and Fair Multi-RED mechanism. In [18] TCP-Window Aware Marker (TWAM) and the Dynamic WRED are proposed. Yaghmaee *et al.* [155] presented a fuzzy traffic conditioner that uses abilities of a fuzzy logic control. The presented comprehensive approach encompasses one more element of a DiffServ architecture – a scheduler. The proposed conditioner consists of the following components: fuzzy meter/marker (FM), fuzzy active queue management (FAQM), and fuzzy scheduler (FS). The FM is based on srTCM. The scheduler is based on WFQ. A comprehensive approach was also proposed in [97] (ETSW and DRIO or the (r,RTT)-adaptive algorithm).

3 Related work

In theory, there is no difference between theory and practice. But, in practice, there is.

— Jan L.A. van de Snepscheut

This chapter gives a survey of the literature related to research on DiffServ performance evaluation and modeling. Section 3.1 provides a general qualitative overview of papers presenting research results on the TCP protocol, RED queue and DiffServ network elemtents. The general overview encompasses research based on either simulation, testing of real networks or analytical modeling. Section 3.2 briefly presents analytical models of TCP dynamics. Sections 3.3 and 3.4 discuss the contribution and scope of some particular papers related to modeling of RED and DiffServ, respectively. In fact, in most cases, the papers present combined models of TCP behavior in a RED or DiffServ controlled network.

3.1 General overview

As regards meter/marker, the most commonly investigated (modeled or simulated) were: the token bucket marker, e.g., [58, 101, 102, 103, 132, 146, 147] and the time sliding window marker [34, 42, 97, 127, 135, 157]. Some papers also dealt with srTCM [145, 155] and trTCM [42, 85, 145]. Researchers usually compared the performance of these meter/markers with some innovative solutions they proposed. It should be emphasized that mathematical models for srTCM and trTCM were not developed. There are also proposals of innovative solutions of meter/markers (they were briefly presented in Section 2.3.1).

Researchers usually deal with two-color marking [9, 12, 17, 28, 29, 33, 58, 88, 97, 101, 102, 103, 105, 132, 136, 139, 146, 147, 157]. It means that packets of

a particular class are marked as in-profile or out-of-profile. Some investigated three-color marking as well [41, 42, 48, 58, 127, 141, 145, 157].

Some researchers deal with modeling networks with RED queue management and without packet marking, e.g., [17, 86, 91, 111, 119]. Nevertheless, these works are important since the presented models are similar to those for Multi-RED queues used in DiffServ networks. Models developed for RED queues were sometimes a basis or inspiration for more complex models for Multi-RED. This is the reason for mentioning research on RED models.

As regards a Multi-RED queue, the following two implementations were most commonly investigated: RIO-C, e.g., [9, 10, 34, 69, 100, 127, 135, 146, 157] and WRED, e.g., [10, 12, 58, 100, 101, 102, 147]. Makkar at al [100] compared performance of the RIO-C and WRED schemes. Barbera *et al.* [10] investigated RIO-C, RIO-DC and WRED types of queue.

Another AQM scheme that should be mentioned is Threshold Dropping. This algorithm does not calculate the drop probability but drops out-of-profile packets or all packets if the average queue size exceeds a given threshold. Such a type of queue was dealt with by Sahu *et al.* [133] and May *et al.* [105].

There are several other AQM algorithms. They were briefly presented in Section 2.3.2.

Researchers often assume, for simplicity, that RTT in the investigated network is homogeneous. Such an assumption is often mentioned in analytical modeling and in network architectures used for simulations. In fact, in the case of simulations a homogeneous propagation delay is assumed so the resulting RTT may slightly fluctuate due to changes of the queuing delay. Nevertheless, the differences in RTT between TCP flows can be neglected. Such an approach is reported, for example, in the following papers: [28, 33, 50, 119, 132, 146]. Several authors, however, took into account the fact that in real networks TCP flows do not experience the same RTT. A traffic aggregate may consist of a large number of flows with different RTTs. This fact was respected in analytical models and simulations presented, for example, in [1, 29, 71, 103, 109, 111, 157].

Most of the analytical models are based on either the renewal theory [12, 27, 101, 102, 103, 104, 122, 132, 146, 157, 158] or fluid models [9, 29, 33, 110, 111, 133]. Models based on the renewal theory are basically used to model various flavors of TCP. They give the throughput in a closed form. The main disadvantage of these models is that they require the packet loss rate and average RTT to be given. Within fluid models, packets are approximated by fluids. Several network characteristics such as the queue size or TCP congestion window size are assumed to change continuously. Models are described by a system of differential equations. These models make it possible to take into account various statistical properties of the system behavior.

In several models, elements of the classical queuing theory are used. The

classical queuing theory is mainly adopted in models of the output queue, e.g., TD, RED, Multi-RED [12, 17, 91, 101, 102, 103, 105, 119]. It is used usually in combined models consisting of a model of an output queue and a TCP source.

Completely distinct approaches, based on loss-competitive and loss-bounded analyses of DiffServ, were considered in [3] and [83], respectively.

From another point of view, combined models encompassing several elements of the network (e.g., a model of a traffic source, meter/markers, droppers, etc.) can be divided into at least two groups: fixed-point methods and control theoretic models [120]. In the former, models for particular elemtents are always distinguishable and can be developed independently. Knowing how the elements interact (e.g., output parameters of one element are input for another) the compound model can be developed. In a control theoretic models a network is described as one large distributed feedback control system and it is usually not possible to separate individual interacting models of particular elements. Both the control theoretic approach and fixed point methods may use various methods (including renewal theory, fluid models, queuing theory) to model particular elements of the system. Examples of the fixed-point models are [23, 50, 109] while the control theoretic models were used in [33, 71, 86, 119].

A comprehensive overview of the modeling approaches with several additional references can be found in [120].

3.2 Models of TCP

The most seminal model of a stationary TCP sending rate was proposed by Padhye *et al.* [122]. This is a model of a long-lived TCP source including most aspects of TCP congestion control: it captures the TCP fast retransmit / fast recovery mechanism and the effect of retransmission timeout (the previous work by Mathis *et al.* [104] took into account only the fast retransmit mechanism). The model is based on the renewal theory. It was developed by a detailed analysis of a TCP congestion window evolution over time. It should be emphasized that this model neglects the slow-start phase of a TCP connection. In the case of long lived TCP flows this phase can be ignored but for short TCP flows it represents a large part of the connection. Cardwell *et al.* [27] extended the Padhye model by supplementing it with a model of a TCP slow start algorithm. The model by Padhye is adopted in this dissertation and presented in Section 4.4.

Some researchers gave their attention to short-lived TCP connections. Such a TCP connection never exits the slow start phase. Such models are important since short-lived TCP connections constitute a non-negligible portion of the Internet traffic. The works by Mellia *et al.* [108] and Garreto *et al.* [56] are examples of such models.

One of the most prominent fluid model of TCP was proposed by Misra *et al.*

[110]. Their model was extended by Altman *et al.* [4]. Another work by Misra *et al.* [111] and the work by Hollot *et al.* [71] are further extentions of fluid models of TCP. They include behavior of TCP in a RED-controlled link.

3.3 Models of RED control

In [17] an analytical model for RED and a simple TD (tail drop) queue are provided. The authors do not develop any advanced model of the traffic. TCP dynamics is not taken into account. The only real Internet traffic characteristic reflected in the model is its burstiness. The authors assume two types of traffic: *bursty* modeled by a batch Poisson process and *smooth* modeled as an ordinary Poisson process. In the former case, the time between arrivals of batches is assumed to have the exponential distribution with a rate λ while the batch size is fixed. The service time is also assumed to have the exponential distribution. The RED queue is modeled by an M/M/1/K queuing system with balking[1] (state dependent arrival rate). The balking probability reflects the drop probability of the RED algorithm. The drop probability is assumed to depend on instantaneous queue size rather than the average queue size that in a real RED algorithm is estimated using EWMA. Such an approximation appears in many papers by several authors. Another approximation is that all packets in an arriving burst experience the same drop probability what is explained by properties of EWMA.

Ohsaki and Murata [119] took a control theoretic approach. They avoided using the classical queuing theory to model the RED queue. Their steady state analysis is based on explicit modeling of the congestion control mechanism of TCP and state transition equations of a queue size. They derived the equilibrium values of the TCP window size and buffer occupancy as well as stability conditions.

Firoiu *et al.* [50] present a fixed-point model for a set of N TCP flows sharing a congested link with a RED mechanism. They develop a feedback control system and do not use the classical queuing theory to model the queue. Their model of a TCP traffic is based on the renewal theory, i.e., the model by Padhye *et al.* [122]. The authors identified the *queue law* governing the queue size which in conjunction with a given control law determines the equilibrium state of the feedback system. A disadvantage of the model is that all flows have the same RTT and are assumed to generate the same average throughput.

In [109], Misra *et al.* used differential equations to model the TCP behavior in a RED-controlled network. They analyzed the impact of the number of TCP flows and RED parameters on the the TCP window size and queue occupancy. Their

[1] Balking is a form of customers' impatience. For more details on queuing systems with impatient customers refer to [60]. See also a footnote on page 57.

model is valid for heterogeneous TCP connections (different RTT and segment size).

The authors of [110, 111] proposed a fluid-based analysis of the TCP behavior in a network of routers deploying a RED queue. Their closed-loop control system based on description of a TCP flow and queue behavior with differential equations is one of the most prominent fluid-based models. It inspired several researchers who developed fluid-based models of RED and DiffServ networks, e.g., [8, 29, 91, 92].

Kuusela *et al.* [91, 92] deal with modeling RED, TCP sources and stability of TCP-RED control. In [92], the authors present an interesting model of a RED queue combining the use of differential equations and the M/D/1/K model of the queue.

The performance of TCP in a RED-controlled network was also studied by Ziegler *et al.* [158, 159]. They also analyzed the impact of several RED parameters on TCP performance and proposed some guidelines on appropriate parameter setting guaranteeing stable system behavior. Their works are based on [122].

3.4 Models of DiffServ

The authors of [105] developed a model for the packet drop probability and waiting time in a DiffServ core router. They concentrate on modeling of three buffer management schemes: tail drop, Multi-RED, and threshold drop. The two color packet marking is assumed. The authors do not model a meter/marker but assume that probabilities that an arriving packet is marked as in- or out-of-profile are fixed and known. It is reported that the RIO-C type of Multi-RED queue is modeled. However, the presented formulas for the drop probability use the same value of queue size what corresponds to WRED or it is a simplification. The queue is modeled as an M/M/1/K queue with balking. Again, balking function corresponds to dropping function of Multi-RED. The authors assume that there is only one traffic aggregate, i.e., there are not several aggregates competing for link capacity. A staggered way of Multi-RED configuration is used. The model was validated through simulations with three types of traffic: (a) Poisson arrivals, (b) superposition of on/off sources with constant bit rate during on periods and exponentially distributed on and off periods, and, (c) superposition of on/off sources with a constant bit rate during on periods and Pareto distribution of on and off periods. It is shown that the simulation results well match the analytical curves. Unfortunately, the paper lacks validation with a realistic traffic, e.g., generated by a population of TCP sources. The assumption on the constant marking probability is unrealistic as well since in a real network if the traffic rate increases the marking probability as out-of-profile increases as well. In the second part of the paper, a model for premium service is presented.

Malouch and Liu also developed a model based on an M/M/1/K queue with balking but, additionally, they dealt with modeling behavior of a token bucket two color meter/marker [101, 102, 103] and took into account dynamics of the TCP traffic. In [101], they proposed a formula for the throughput achieved by an individual TCP source as a function of token bucket parameters, loss probability (here assumed to be known) as well as RTT and RTO parameters of TCP. While deriving the formula for the TCP throughput the authors considered four distinct cases requiring different approaches. The distinction is based on the relation of the TCP congestion window size to the TB parameters. Malouch and Liu do not provide explicit formulas for the marking probability but derive formulas for the average rate of in- and out-of-profile packets. Therefore, the marking probability can be found. The approach is based on the model of TCP behavior by Padhye *et al.* [122]. It should be noted that it is a model of long-lived TCP sources and the formulas for throughput of in- and out-of-profile packets are valid only for this model of a TCP source.

As mentioned, the model by Malouch and Liu presented in [101] makes it possible to find the rate of either in- and out-of-profile packets. This feature is used in [102] and [103] where the model of a Multi-RED queue is developed. The authors say that a RIO type of queue is modeled without explicitly explaining which particular type of queue is assumed. They decided to neglect the EWMA algorithm and assume that the instantaneous queue size is used for dropping decision. Since the same value is used for calculating the drop probabilities of in- and out-of-profile packets it may be concluded that, according to the Multi-RED classification by Makkar *et al.* [100] (see Section 2.3.2), the WRED type of the Multi-RED queue is considered. As mentioned above, it is modeled as an M/M/1/K queue with balking. For completeness of the model, the authors propose formulas approximating values of RTT and RTO. In [102], the model is validated through simulation for staggered configuration of Multi-RED parameters. Experiments for overlapped and partially overlapped configurations can be found in [103]. Moreover, the authors discuss the impact of Multi-RED parameters on QoS characteristics such as throughput, delay and the drop probability.

Su and Atiquzzaman [146] presented a performance model of DiffServ with a token bucket marker. The authors found the formula for the bandwidth achieved by an aggregate of multiple flows. The model of the traffic generated by a single long-lived TCP is based on the renewal theory. The bandwidth is expressed as a function of token bucket parameters, the packet drop probability, the number of flows constituting the aggregate, and the maximum segment size. Taking into account the relation between so called committed window size[2] and the maximum, average and half of the maximum congestion window size the authors distinguish

[2]The committed window size is a parameter stemming from token bucket parameter CIR and is defined as $CIR \times RTT/MSS$, where MSS is the maximum segment size

the following four cases in which they analyze the TCP throughput: under-provisioned network, nearly-provisioned network, slightly under-provisioned network and largely over-provisioned network. This distinction is quite similar to the one by Malouch and Liu [101] but was probably developed independently. A separate model of the core router queue is not developed but the packet drop probability is assumed to be given as a paramater for the calculations. This model allows quick checking how changing of token bucket parameters affects the TCP performance. The main limitation of the model is the assumption on homogeneous RTT within the aggregate.

Reference [132] is another example of TCP throughput modeling in a DiffServ network with a TB meter/marker. The model is developed on the basis of the Padhye model of TCP. The drop probability is assumed to be known and is given as an input for the analysis. As regards the drop probabilities of in- and out-of-profile packets, two separate cases were considered: (a) under-subscription case, where the drop probability of in-profile packets equals zero and the drop probability of out-of profile packets is between 0 and 1, and (b) over-subscription case where the drop probability of in-profile packets is above zero and the drop probability of out-of profile packets equals 1. The authors, Sahu *et al.*, adopted this simplifying approximation from [157]. Formulas for TCP throughput in the two cases are derived separately. An impact of RTO on TCP performance is respected. The authors presented a validation of the model with simulations performed in `ns-2`. Separate simulations were performed for each of the cases. Finally, the authors considered the impact of TB parameters on TCP performance and suggested how to select the TB parameters to achieve a given target rate, if possible.

Yeom and Reddy [157] modeled TCP behavior in a DiffServ network as well. Unlike the previously discussed works, they assume that a Time Sliding Window meter/marker is used rather than the token bucket. They consider not only two-color marking but three color marking as well. They, in fact, derived a combined model of throughput of a TCP flow experiencing packet marking with a TSW meter/marker and some packet drop probability. They do not provide formulas for marking probability independent of the TCP model. Any particular realization of the Multi-RED queue is modeled (a RIO queue is reported to be used in simulations). Instead, the packet drop probability was used as a parameter in throughput formulas. The authors developed models for per-flow and per-aggregate marking schemes. The model is developed in a similar way to the TCP model by Padhye *et al.* The analysis is performed independently for the under-subscription case and the over-subscription case (as discussed in the previous paragraph). Convincing validation with simulations performed with `ns-2` simulator were presented.

TSW2CM was also examined by Baumann *et al.* [12]. The authors propose

formulas for marking probabilities under assumption that a single meter/marker entity serves for a single long-lived TCP flow. Thus, it is again a combined model of a TCP flow and a meter/marker. The paper is another example of an approach based on the analysis of periodic evolution of the TCP congestion window, i.e., the renewal theory. However, the impact of RTO on TCP throughput was neglected. Another weakness of this model is the assumption on per-flow instead of per-aggregate marking. The WRED queue used in a core router is modeled with an M/M/1/K queue with balking.

As mentioned before, a distinct group of analytical models of DiffServ networks are fluid analysis based models, e.g., [9, 10, 29, 133]. Barbera *et al.* [9] provided an analytical fluid model of a RIO queue loaded by traffic sources that can be represented as Markov Modulated Fluid Processes (MMFP). Both in- and out-of-profile packets are treated as separate flows generated by separate MMFP sources. The packet marking probability is assumed to be known, i.e., the average rate of in- and out-of-profile packets is given. Barbera *et al.* also derived fluid models for the RIO-C, RIO-DC and WRED types of queue well reflecting the differences between those queue types, and the TB meter/marker [10]. They considered a network loaded simultaneously with greedy and short-lived TCP flows. They assumed two color marking per AF class. They applied the model to a complex network topology.

Chait *et al.* [29] developed a model for AF PHB with two drop precedences as well. They modeled queue behavior and traffic with fluid approximation for long-lived TCP sources only. The authors introduced and analytically described the Active Rate Management (ARM) mechanism at edge markers.

Sahu *et al.* [133] derived analytical models to compare the loss and delay behavior under various combinations of two core router mechanisms: threshold dropping and priority scheduling (not a Multi-RED queue), and two packet marking mechanisms called edge-marking and edge-dropping. They used Markov models for the non-bursty traffic. Bursty sources were modeled by Markov modulated "on-off" sources. They adapted fluid models proposed by Elwalid and Mitra [43, 112].

Fluid-based methods were also used by Muppala *et al.* [33] who proposed a new meter/marker called VS-ACT (see Section 2.3.1). They used a control-theoretic approach and fluid models to perform system stability analysis.

Part II

Analytical models and verification

4 Analytical models for DiffServ elements

A mathematician is a device for turning coffee into theorems.

— Paul Erdös

The author of this dissertation is not a mathematician, he is an engineer. Being aware that mathematicians would argue on adequacy of the methods used he believes that successful experimental validation sufficiently supports some *"engineering"* approach used in the modeling.

The following sections present basic assumptions made to develop the models, models for meter/markers and droppers, a sample model for TCP throughput, and a modular model for behavior of TCP flows and aggregates subject to AF PHB. All the models, except for the model for TCP throughput, have been developed by the author of the dissertation.

4.1 Basic assumptions

The key assumption is that the time between arrivals of consecutive packets constituting the aggregate has the exponential distribution. Such an assumption enables the use of elements of the classical queuing theory in the mathematical model. It is also easier to find approximate distributions for some random variables describing the network. Some researchers show that such an approach is justified in a heavily loaded network [25, 26, 143]. Cao *et al.* [25, 26] showed that connection and packet arrival processes tend locally toward the Poisson distribution and time series of packet sizes and round trip times tend locally toward independence as the rate of new TCP connections increases. Moreover, Hohn *et*

al., experts in the self-similarity area of research, admit that as far as a short time scale is considered the use of Poisson modeling is reasonable [70]. Also a simplified and intuitive explanation can be given: in a congested link there are thousands of flows, therefore, packets belonging to a single flow are interleaved with packets from other flows and, what follows, correlation between consecutive packets in the aggregated stream is reduced. The traffic in a heavily loaded DiffServ network is often smoothed and shows less bursty nature (effect of synchronization of TCP sources is reduced in DiffServ networks). Numerous researchers successfully used the Poisson model of packet arrivals in their works, e.g., [17, 102, 105, 133].

There are also many arguments against Poisson modeling [126, 153], so the model is developed and validated very carefully. Despite the issue of the self-similar nature of the Internet traffic, opponents of Poisson modeling raise the argument that it does not take into consideration real characteristics of TCP congestion control. The last issue is addressed by the use of an appropriate model for the TCP throughput (Section 4.4).

The distribution of a packet size in the Internet is recognized as trimodal. The peaks are around 40, 572 and 1500 bytes resulting from TCP acknowledgments, from TCP implementations without path MTU (Maximum Transmission Unit) discovery and from the maximum Ethernet frame size, respectively [24, 54, 57, 106, 148, 152]. The exact distribution is different in various areas of the Internet but demonstrates the trimodal character in most cases or, less often, bimodal. Moreover, the average packet size in the Internet is reported to be about 400–570 bytes. The average packet size is often considered in research rather than the distribution, for simplicity, e.g., [1, 6, 8, 9, 46, 51, 64, 84, 92, 102, 111, 132, 139, 146, 159]. Many researchers use the average packet size of about 500, 1000 or 1500 bytes in their works. They, in fact, assume that the packet size is a constant rather than a random variable. It is also reflected in simulations performed to study the network behavior or validate the models. A similar assumption has been adopted in this dissertation.

Meter/markers and droppers are modeled independently of other network elements so no special assumptions on network architecture are needed. More assumptions are needed for the modular model but they are provided in the respective section (4.5).

4.2 Meter/markers

Modeling of token bucket based and rate estimator based meter/markers requires different approaches. Generally, on some assumptions, token bucket based meter/markers can be described by the use of the classical queuing theory while modeling of rate estimator based meter/markers needs another approach.

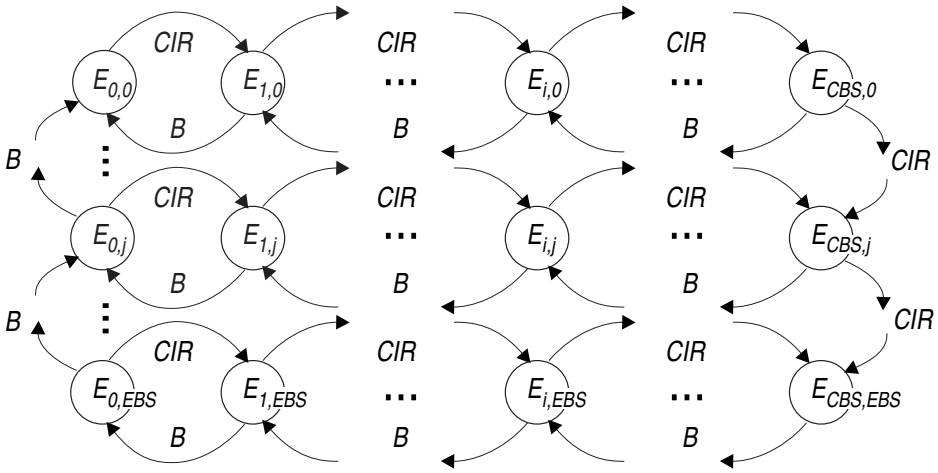


Figure 4.1: State transition graph of the model of srTCM

4.2.1 Token bucket based traffic meter/markers

A token bucket can be represented by a queuing system with a finite buffer. The buffer size represents the depth of the bucket. Mean arrival rate represents the rate of incrementing the token count. The service rate represents the rate of the incoming traffic subject to metering and marking.

Single Rate Three Color Marker

As described in Section 2.3.1, the srTCM consists of two token buckets C and E with depths CBS and EBS, respectively. In order to adapt the classical queuing theory to modeling of srTCM easily, it was assumed that current states of the token buckets are expressed in the number of packets, rather than in bytes. The token count of bucket C is incremented by one CIR times per second. The token count is decremented by one on each packet arrival. Hence, the average rate B of emptying bucket is equal to the inverse of the average packet inter-arrival time and is expressed in packets per second as well.

Intuitively, the D/M/1/K queuing system could be used for modeling of srTCM since the rate of token arrivals to the bucket is deterministic. The time interval between token arrivals is constant and equal to $1/CIR$. Tokens are pulled out form buckets upon each packet arrival. Thus, the service rate is, in general, equivalent to the rate of the traffic subject to metering and marking at srTCM. In turn, the packet inter-arrival time was assumed to be exponentially distributed (see Section 4.1). Therefore, the service time can be assumed exponentially dis-

tributed as well. The model of srTCM based on D/M/1/K was developed, but compared with experimental results it performed poorly. Therefore, it is not presented in the dissertation. It turned out that the model of srTCM based on the M/M/1/K queuing system performs satisfactorily well, so it was decided to present it here. The model was also presented in [142].

To find marking probabilities for srTCM, a state transition graph for such a system was developed (Fig. 4.1). The system consists of two M/M/1/K queues: for bucket C and E. If a queue representing bucket C is in state i and queue representing bucket E is in state j then the system is in state $E_{i,j}$. If a steady state of the system is assumed, it can be described by the following set of algebraic equations:

$$\begin{aligned}
0 &= -CIR\,\pi_{0,0} + B\,\pi_{0,1} + B\,\pi_{1,0} \\
0 &= -(CIR+B)\,\pi_{0,j} + B\,\pi_{0,j+1} + B\,\pi_{1,j} && 1 \le j \le EBS-1 \\
0 &= -(CIR+B)\,\pi_{0,EBS} + B\,\pi_{1,EBS} \\
0 &= -(CIR+B)\,\pi_{i,j} + B\,\pi_{i+1,j} + CIR\,\pi_{i-1,j} && \begin{array}{r}1 \le i \le CBS-1\\ \text{and } 0 \le j \le EBS\end{array} \\
0 &= -(CIR+B)\,\pi_{CBS,0} + CIR\,\pi_{CBS-1,0} \\
0 &= -(CIR+B)\,\pi_{CBS,j} + CIR\,\pi_{CBS-1,j} && 1 \le j \le EBS-1 \\
&\quad + CIR\,\pi_{CBS,j-1} \\
0 &= -B\,\pi_{CBS,EBS} + CIR\,\pi_{CBS,EBS-1} \\
&\quad + CIR\,\pi_{CBS-1,EBS}
\end{aligned} \tag{4.1}$$

where $\pi_{i,j}$ is the probability that the system is in state $E_{i,j}$. The above equations were obtained by writing global balance equations [60] for each state of the system presented in Fig. 4.1, under assumption on a steady state of the system. Obviously, a normalization equation is needed:

$$\sum_{i=0}^{CBS}\sum_{j=0}^{EBS}\pi_{i,j} = 1 \tag{4.2}$$

By solving (4.1) and (4.2) the following formulas for $\pi_{i,j}$ can be obtained:

$$\pi_{i,0} = \frac{\left(\frac{CIR}{B}\right)^i \sum_{k=0}^{CBS-i}\left(\frac{CIR}{B}\right)^k}{\sum_{l=0}^{CBS+EBS}\left(\frac{CIR}{B}\right)^l \sum_{l=0}^{CBS}\left(\frac{CIR}{B}\right)^l} \quad \text{for } i = 0,\ldots,CBS \tag{4.3}$$

$$\pi_{i,j} = \frac{\left(\frac{CIR}{B}\right)^{j+CBS}}{\sum_{l=0}^{CBS+EBS}\left(\frac{CIR}{B}\right)^{l}\sum_{l=0}^{CBS}\left(\frac{CIR}{B}\right)^{l}} \quad \begin{array}{l}\text{for } i = 0,\ldots,CBS \\ \text{and } j = 1,\ldots,EBS-1\end{array} \tag{4.4}$$

$$\pi_{i,EBS} = \frac{\left(\frac{CIR}{B}\right)^{CBS+EBS}\sum_{k=0}^{i}\left(\frac{CIR}{B}\right)^{k}}{\sum_{l=0}^{CBS+EBS}\left(\frac{CIR}{B}\right)^{l}\sum_{l=0}^{CBS}\left(\frac{CIR}{B}\right)^{l}} \quad \text{for } i = 0,\ldots,CBS \tag{4.5}$$

According to [68], a packet is marked red if upon its arrival both buckets are empty, that is, the system is in state $E_{0,0}$. Hence, using the PASTA[3] property, the probability P^{mr} that a packet is marked red is equal to $\pi_{0,0}$. The packet is marked yellow if it arrives at the moment when bucket C is empty and bucket E is not empty, i.e., a system is in state $E_{0,j}$ where $j = 1,\ldots,EBS$. Consequently, the probability P^{my} that a packet is marked yellow equals $\sum_{j=1}^{EBS}\pi_{0,j}$. In the other cases the packet is marked green.

Finally, the probabilities that a packet will be marked red P^{mr}, yellow P^{my} or green P^{mg} are as follows:

$$P^{mr} = \frac{1}{\sum_{l=0}^{CBS+EBS}\left(\frac{CIR}{B}\right)^{l}} \tag{4.6}$$

$$P^{my} = \frac{\sum_{j=1}^{EBS}\left(\frac{CIR}{B}\right)^{j+CBS}}{\sum_{l=0}^{CBS+EBS}\left(\frac{CIR}{B}\right)^{l}\sum_{l=0}^{CBS}\left(\frac{CIR}{B}\right)^{l}} \tag{4.7}$$

$$P^{mg} = 1 - \frac{1}{\sum_{l=0}^{CBS}\left(\frac{CIR}{B}\right)^{l}} \tag{4.8}$$

The above formulas express packet marking probabilities as a function of srTCM parameters, CIR, CBS and EBS, and traffic rate B.

[3]Poisson Arrivals See Time Averages [60]

Two Rate Three Color Marker

As presented in Section 2.3.1, the trTCM consists of two token buckets C and P with depths CBS and PBS, and token counts T_c and T_p, respectively. Again, it was assumed that the current state of buckets is expressed in the number of packets. The token count of bucket C is incremented by one CIR times per second while the token count of bucket P is incremented PIR times per second. The average rate of emptying token bucket P is equal to the average rate of incoming traffic B.

In the case of trTCM, the decision which of the two models, M/M/1/K or D/M/1/K, better describes real meter/marker's characteristics is ambiguous. Depending on the type of traffic used in experiments one or the other model performs better (see Section 5.2.2). Therefore, both models are described. The author of the dissertation has presented the model based on M/M/1/K in [141] and [142]. The model based on D/M/1/K was introduced in [142].

In the case of the M/M/1/K queuing system, the probability of bucket P being in state j can be expressed as follows [87]:

$$\pi_j^P = \frac{\left(\frac{CIR}{B}\right)^j}{\sum_{l=0}^{PBS}\left(\frac{CIR}{B}\right)^l} \tag{4.9}$$

Bucket C is emptied only if bucket P is not empty, thus, the conditional probability that bucket C is in state i is as follows:

$$\pi_{i|T_p\neq 0}^C = \frac{\left(\frac{CIR}{B\left(1-\pi_0^P\right)}\right)^i}{\sum_{l=0}^{CBS}\left(\frac{CIR}{B\left(1-\pi_0^P\right)}\right)^l} \tag{4.10}$$

where π_0^P is the probability that token bucket P is in state 0 (is empty).

A packet is marked red if upon its arrival bucket P is empty. If bucket P is not empty and bucket C is empty then the packet is marked yellow. A packet is marked green in all other cases [67]. Hence, the probabilities that a packet will be marked red, yellow or green are as follows:

$$P^{mr} = \pi_0^P = \frac{1}{\sum_{i=0}^{PBS}\left(\frac{PIR}{B}\right)^i} \tag{4.11}$$

$$P^{my} = \left(1 - \pi_0^P\right) \pi_{0|T_p \neq 0}^C = \frac{1 - P^{mr}}{\sum\limits_{i=0}^{CBS} \left(\frac{CIR}{B\left(1 - P^{mr}\right)}\right)^i} \tag{4.12}$$

$$P^{mg} = 1 - P^{mr} - P^{my} = \left(1 - P^{mr}\right) \left(1 - \frac{1}{\sum\limits_{i=0}^{CBS} \left(\frac{CIR}{B\left(1 - P^{mr}\right)}\right)^i}\right) \tag{4.13}$$

State probabilities for the D/M/1/K system and, what follows, marking probabilities are more difficult to obtain. In fact, closed form formulas for state probabilities are very complex. Therefore, the solution will be provided in steps.

State probabilities for the D/M/1/K system are usually obtained from state probabilities for the M/D/1/K system by using the property of symmetry between those systems [81]. Generally, the probability that the D/M/1/K queue is in state i is equal to the probability that the M/D/1/K queue is in state $K - i$:

$$\pi_i^{\mathrm{D/M/1/K}}(\varrho) = \pi_{K-i}^{\mathrm{M/D/1/K}}(1/\varrho) \tag{4.14}$$

where ϱ is the system load. In turn, state probabilities of the finite M/D/1 queue can be obtained from state probabilities for the infinite M/D/1, i.e., M/D/1/∞ system as follows [79, 92]:

$$\begin{aligned} \pi_j^{\mathrm{M/D/1/K}}(\varrho) &= \frac{\pi_j^{\mathrm{M/D/1/\infty}}(\varrho)}{\pi_0^{\mathrm{M/D/1/\infty}}(\varrho) + \varrho\, G(K)}, \qquad j = 0, \ldots, K-1 \\ \pi_K^{\mathrm{M/D/1/K}}(\varrho) &= 1 - \frac{G(K)}{\pi_0^{\mathrm{M/D/1/\infty}}(\varrho) + \varrho\, G(K)} \end{aligned} \tag{4.15}$$

where $G(K) = \sum_{j=0}^{K-1} \pi_j^{\mathrm{M/D/1/\infty}}(\varrho)$. Finally, state probabilities in the infinite M/D/1 system can be calculated from the following formulas [60, 79]:

$$\begin{aligned} \pi_0^{\mathrm{M/D/1/\infty}}(\varrho) &= 1 - \varrho, \\ \pi_1^{\mathrm{M/D/1/\infty}}(\varrho) &= (1 - \varrho)\left(e^{\varrho} - 1\right), \\ \pi_i^{\mathrm{M/D/1/\infty}}(\varrho) &= (1 - \varrho) \sum_{j=1}^{i} (-1)^{i-j}\, e^{j\,\varrho} \left(\frac{(j\,\varrho)^{i-j}}{(i-j)!} + \frac{(j\,\varrho)^{i-j-1}}{(i-j-1)!}\right), \quad i \geq 2 \end{aligned} \tag{4.16}$$

Formula (4.16) is valid only for $0 < \varrho < 1$ while formula (4.15) is valid for $\varrho > 0$. Formula (4.15) can be also obtained by solving a certain set of equations called Fry's equations of state [79].

Finally, the packet marking probabilities for trTCM modeled using the D/M/1/K queue model can be calculated as follows:

$$P^{mr} = \pi_0^{\mathrm{D/M/1/}PBS}\left(\frac{PIR}{B}\right) \tag{4.17}$$

$$P^{my} = (1 - P^{mr})\,\pi_0^{\mathrm{D/M/1/}CBS}\left(\frac{PIR}{B\,(1 - P^{mr})}\right) \tag{4.18}$$

$$P^{mg} = 1 - P^{mr} - P^{my} \tag{4.19}$$

4.2.2 Rate estimator based traffic meter/markers

The classical queuing theory is not applicable to modeling of rate estimator based meter/markers. Final formulas for packet marking probabilities in this case are derived in the following two steps:

- finding a probability density function (PDF) of the TSW-averaged traffic rate (estimated by the TSW algorithm),
- finding formulas for the marking probability as integrals of the PDF.

The approach to modeling rate estimator based meter/markers is presented by using the example of the TSW3CM meter/marker [47]. This model, except the boundary characteristics discussed at the end of the current section, was presented in [141].

Step 1: Find a distribution of the average rate measured by the TSW algorithm

As presented in Section 2.3.1, TSW3CM has two independent components: a rate estimator and a probabilistic tagger. According to the algorithm of the TSW rate estimator, the TSW-averaged traffic rate A in k-th observation can be expressed as:

$$A_k = \frac{A_{k-1}W + S}{W + T_k} \tag{4.20}$$

where W is the observation window size in seconds, S is the packet size in bits, and T_k is the time between current (k-th) and previous (($k-1$)-th) packet arrivals in seconds. The TSW-averaged traffic rate A is expressed in bits per second.

The window size W is a constant configurable parameter of TSW3CM, S is assumed to be constant while T is a random variable. If the observation window size W tended to infinity, A would be approximately constant, reflecting

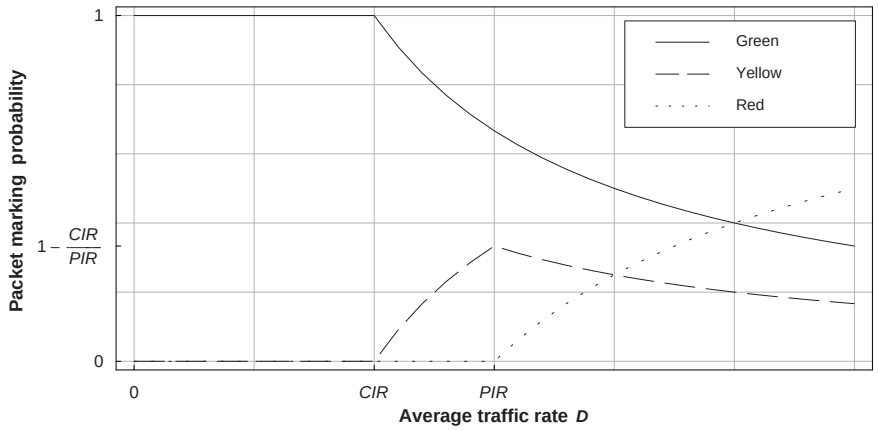


Figure 4.2: Sample characteristics of the packet marking probability (as green, yellow and red) versus the average traffic rate under assumption of the infinite window length in TSW3CM

a long term average throughput of the incoming traffic. Thus, the value of A would be equal to the average traffic rate D expressed as $\frac{S}{E[T]}$, where $E[T]$ is the expected value (mean) of the packets inter-arrival time T. Note that PDF of the distribution of the packet inter-arrival time in such a case is irrelevant. The characteristics of the packet marking probability could be derived directly from the marking algorithm (Fig. 2.12). The characteristics would be as presented in Fig. 4.2.

In reality, the window size W is not infinite. Thus, the TSW-averaged traffic rate A is a random variable. What follows, the characteristics of the packet marking probabilities will depend on the PDF of the TSW-averaged traffic rate A. If the window size is large, the characteristics are closer to those presented in Fig. 4.2. The smaller window size, the more the characteristics deviate from those for the infinite window size. The discrepancy is most significant in areas around CIR and PIR, that is, the most important areas from the algorithm's performance perspective.

Providing an exact formula for the probability density function of A is not easy. In a steady state, PDF of A, denoted as $A(x)$, does not depend on the number of observation k. However, according to (4.20), the random variable A is a function of two random variables: A and T. Therefore, an attempt to derive PDF of A by using the technique for finding PDF of a random variable being a function of other random variables [55] inevitably leads to a complex

integral equation since the unknown PDF, $A(x)$, appears on both sides of the transformation. Analytical solution to such an integral equation cannot be found. Therefore, a simplified but intuitive approach is taken.

Assume that time T between packet arrivals is a random variable and has the exponential distribution with parameter λ:

$$\lambda = \frac{1}{E[T]} = \frac{D}{S} \ [\mathrm{s}^{-1}] \tag{4.21}$$

where D is the average rate of the incoming traffic in bits per second. Note that, in contrast to models of the token bucket based meter/markers, the average traffic rate is expressed here in bits per second rather than packets per second. To avoid confusion, it is denoted by D.

The TSW algorithm estimates the averaged traffic rate within the time of a window of length W plus the time between the last and penultimate packet arrivals. If the window length is large in comparison to the inter-arrival time (i.e., $W \gg T$) it can be assumed that, on average, the following number of packets arrive within the period $W + T$:

$$n = \frac{W}{E[T]} + 1 \tag{4.22}$$

Let T^{avg} be a continuous random variable representing the inter-arrival time averaged over n observations within the period $W + T$. It is calculated as a sum of n different random values of T divided by n:

$$T^{\mathrm{avg}} = \frac{T_1 + T_2 + \ldots + T_n}{n} \tag{4.23}$$

It can be shown that under assumption of $n > 0$ and $\lambda > 0$ the sum of n exponentially distributed random variables with parameter λ each has a gamma distribution with parameters λ, n [7]:

$$g(x) = \frac{\lambda^n x^{n-1} e^{-\lambda x}}{(n-1)!} H(x) \tag{4.24}$$

where $H(x)$ is a Heaviside step at $x = 0$. The requirement on $n > 0$ and $\lambda > 0$ is obviously satisfied. The formula (4.24) can be generalized for non-integer values of n by replacing a factorial in the denominator with a gamma function.

According to formula (4.23), to obtain T^{avg} it is necessary to divide the sum of inter-arrival times by n. Thus, to derive the PDF of T^{avg} the procedure for finding the probability density function of a random variable obtained as function of another random variable of a known PDF [55] must be applied. Then, after

some algebraic manipulation, the following PDF of T^{avg} is obtained from (4.24):

$$T^{\text{avg}}(x) = \frac{(n\lambda)^n x^{n-1} e^{-n\lambda x}}{\Gamma(n)} H(x) \tag{4.25}$$

where $\Gamma(\cdot)$ is a gamma function and $n = \frac{D\,W}{S} + 1$ (from (4.21) and (4.22)). The gamma function is used instead of a factorial since n does not necessarily assume integer values.

Having the averaged inter-arrival time, the TSW-averaged traffic rate may be computed as:

$$A = \frac{S}{T^{\text{avg}}} \tag{4.26}$$

Thus, random variable A is a function of random variable T^{avg}. The PDF of T^{avg} is currently known (formula (4.25)), so the PDF of A can be found by application of the procedure mentioned above [55]. After some algebraic manipulation, from (4.25) and (4.26), the PDF for the TSW-averaged traffic rate estimated by the rate estimator is as follows:

$$A\,(x) = \frac{(nD)^n x^{-n-1} e^{-n\frac{D}{x}}}{\Gamma\,(n)} \tag{4.27}$$

where $n = \frac{D\,W}{S} + 1$.

Step 2: Find formulas for the marking probability as integrals of the PDF

Knowing the PDF for the TSW-averaged traffic rate (4.27) and the marking algorithm, characteristics of the packet marking probabilities can be found. The integrals of PDF for A in appropriate bounds should be solved. In the case of the TSW3CM marking algorithm (Fig. 2.12), the probabilities that a packet will be marked red, yellow or green are as follows:

$$P^{mr} = \int_{PIR}^{\infty} \left(1 - \frac{PIR}{x}\right) A\,(x)\; dx \tag{4.28}$$

$$P^{my} = \int_{CIR}^{PIR} \left(1 - \frac{CIR}{x}\right) A\,(x)\; dx + \int_{PIR}^{\infty} \left(\frac{PIR - CIR}{x}\right) A\,(x)\; dx \tag{4.29}$$

$$P^{mg} = \int_{0}^{CIR} A\,(x)\; dx + \int_{CIR}^{\infty} \frac{CIR}{x} A\,(x)\; dx \tag{4.30}$$

Formulas for the marking probability of Time Sliding Window Two Color Marker (TSW2CM) are very similar. They differ in the bounds of integrals. Characteristics of ItswTCM [145] may be found in a similar way.

Boundary characteristics of marking probabilities

A real meter/marker characteristics may differ from the above model which is only an approximation. Additionally, the inter-arrival time may be not necessarily exponentially distributed. However, in the case of TSW rate estimator it is possible to find boundary distributions and, what follows, boundary values of marking probabilities. It is expected that values of marking probabilities obtained from simulation of TSW3CM operating in a real network, if spread, will lie between the boundary characteristics.

In theory, the minimum inter-arrival time is 0 and, what follows, the maximum possible value of n is infinity, regardless of the window size. In such a case, the TSW-averaged traffic rate would be constant and marking characteristics would be as presented in Fig. 4.2. In practice, the minimum inter-arrival time is finite and can be easily found. It stems from minimum packet size and the maximum possible rate of the traffic incoming to the meter/marker. The maximum traffic rate simply equals to the capacity of the input link. Therefore, the maximum possible value of n, denoted as $n^{\top}$, can be calculated as follows:

$$n^{\top} = \frac{W\, C_{in}}{S_{min}} + 1 \tag{4.31}$$

where C_{in} is the capacity of the input link while S_{min} is the minimum packet size. Substituting n with $n^{\top}$ in (4.27), the upper bound PDF of A and the upper bound characteristics of the marking probabilities can be found.

Calculation of the lower bound characteristics is less obvious. Theoretically, the maximum inter-arrival time is infinite but such an assumption is not realistic. Instead, it was assumed that the maximum inter-arrival time is calculated from cumulative distribution function (CDF) of the inter-arrival time for a certain probability ς. If the time between packet arrivals T has the exponential distribution it has the following CDF:

$$\boldsymbol{T}(x) = 1 - e^{-\lambda x} \tag{4.32}$$

where λ is expressed by (4.21). The maximum inter-arrival time is then calculated by solving the following equation:

$$\varsigma = 1 - e^{-\lambda T^{max}} \tag{4.33}$$

and is finally expressed as:

$$T^{max} = \frac{-\ln(1-\varsigma)\, S}{D} \tag{4.34}$$

The selection of the value of ς impacts the shape of the lower bound characteristics. The proposed value of ς is 0.999, as a rule of thumb.

Finally, the minimum number of packets observed within $W + T$ period, denoted as $n^{\perp}$, can be calculated as:

$$n^{\perp} = \frac{W}{T^{max}} + 1 \tag{4.35}$$

and using (4.34) and (4.35) expressed as:

$$n^{\perp} = \frac{W\ D}{-\ln(1-\varsigma)\ S} + 1 \tag{4.36}$$

The lower bound characteristics can now be calculated by using $n^{\perp}$ and the same technique as for the upper bound characteristics.

4.3 Droppers

In the Multi-RED queue an arriving packet may be subject to early dropping before entering the queue. Thus, not all of the arriving packets finally enter the queue. Assuming the steady state of the system the packet early drop probability p^{ed} could be found. Then, if the incoming traffic rate is denoted by Λ, the effective rate of the traffic entering the queue is $\Lambda_{eff} = \Lambda(1-p^{ed})$. Packets that eventually entered the queue may be dropped if the queue overflows. For example, if the M/M/1/K model of the queue is assumed the packet drop probability due to overflow equals the probability that the queue is in state K, π_K. The overall packet drop probability can be calculated as $1-(1-p^{ed})(1-\pi_K)$.

To find the packet early drop probability the knowledge on the distribution of the exponentially averaged queue size is needed. It is a key problem in that approach. However, as shown in [89, 90], finding the distribution of the exponentially averaged queue size is not easy because the two random variables: the instantaneous queue size and the exponentially averaged queue size are not independent of each other. The authors of those publications developed a model for the joint dynamics of the instantaneous and the exponentially averaged queue length in an M/M/1/K queue. However, analytical solutions for the distribution functions can be found for only a few special cases such as M/M/1/1 or M/M/1/2 systems and under additional assumptions on the arrival rate (λ), service rate (μ), and weighting parameter w_q, e.g., $\lambda/w_q \in \mathcal{N}$, $\mu/w_q \in \mathcal{N}$, or $\lambda = \mu$. For queues with higher values of the maximum queue length K solutions are obtainable numerically, but only for some special cases as well. It should be also emphasized that none of the well known distributions or a combination of them can be fitted to a real distribution of the exponentially averaged queue size since the shape of a real distribution function highly varies depending on queue parameters. A sample graph showing the time evolution of the instantaneous and exponentially averaged queue sizes, obtained from simulation, is presented in Fig. 4.3. The

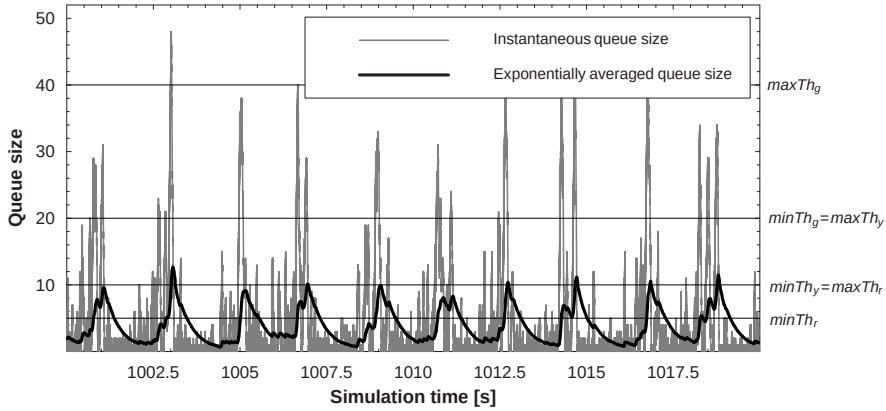


Figure 4.3: Time evolution of the instantaneous and exponentially averaged queue size

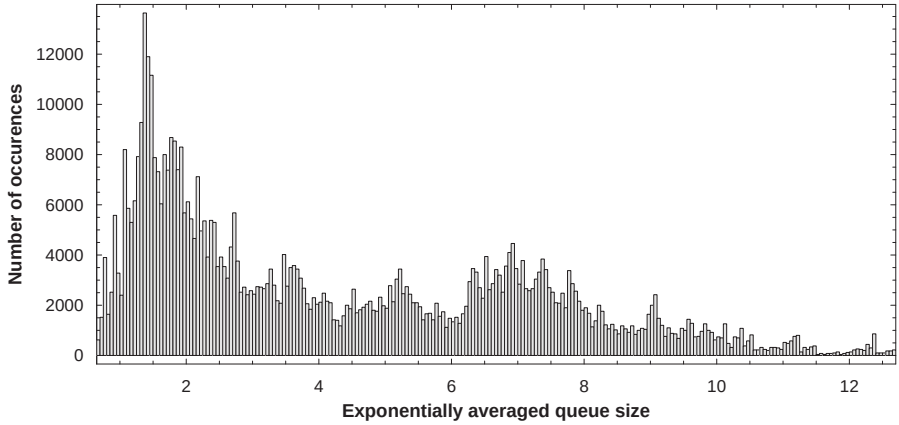


Figure 4.4: Histogram of the probability density of the exponentially averaged queue size

corresponding histogram of the real distribution of the exponentially averaged queue size is shown in Fig. 4.4. Therefore, a different approach is needed.

The difficulty with finding PDF for the exponentially averaged queue size is often avoided by the assumption that the early drop probabilities depend on the instantaneous queue size rather than the exponentially averaged queue size. A similar approach was taken, for example, in [12], [17] and [102].

In the most computationally simple approach presented by the author of the

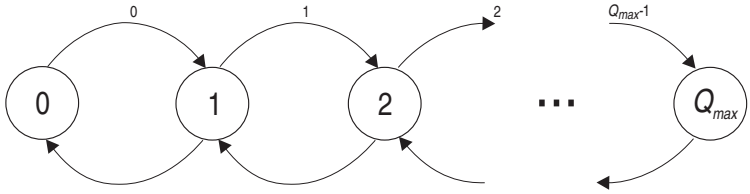


Figure 4.5: State transition graph of a queuing system with balking

dissertation in [140], the M/M/1/K queue is fed with a rate Λ_{eff} and the state probabilities are treated as a probability distribution function of the average queue size and used to calculate the early drop probabilities. This approach gives satisfactory results only in a narrow range of throughput.

An approach that uses a queuing system with balking, that is, a case of a queuing system with impatient customers[4], is more accurate. In such models, the arrival rate is state dependent. The propability that the customer will join the queue depends on its current length. Following the above assumption, the packet arrival rate depends on the instantaneous rather than the exponentially averaged queue size. The state transition graph of a queuing system with balking is presented in Fig. 4.5. The M/M/1/K queuing system with balking is denoted as M(n)/M/1/K, while the M/D/1/K queuing system with balking is denoted as M(n)/D/1/K. A similar notation was used in [63].

In order to develop the model based on queuing systems it was assumed that the rate of packet arrivals and the service rate are expressed in packets per second, not in bits per second. The maximum queue size is denoted as Q_{max} and is expressed in packets. The service rate is denoted as μ and, in this particular considerations, it is the bottleneck capacity. Since the arrival rate depends on the queue state, λ_i denotes the arrival rate when the queue is in state i. It is assumed that $\lambda_0 \geq \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{Q_{max}-1}$.

The next issue is finding the arrival rates for each state i of the system. If the system is in state 0, all arriving packets enter the queue. Thus, if the real rate of packets incoming to the Multi-RED queue is denoted as B then:

$$\lambda_0 = B \tag{4.37}$$

[4]There are three forms of customers' impatience [60]: a) balking, the reluctance to join the queue upon arrival; b) reneging, the reluctance to remain in the queue after joining and waiting; c) jockeying between parallel queues.

For $1 \leq i \leq Q_{max} - 1$ the arrival rates are calculated as follows:

$$\lambda_i = B\left(1 - P_i^d\right) \tag{4.38}$$

where P_i^d is the overall packet drop probability at state i. The probability P_i^d depends on the traffic structure, i.e., the percentage of red, yellow and green packets in the stream and the packet drop probability per color. The probability that the packet in the queue is red is denoted as P^{qr}, yellow: P^{qy} and green: P^{qg}. Obviously, $P^{qr} + P^{qy} + P^{qg} = 1$. The packet drop probability per color is calculated differently for WRED and RIO-C types of queue and derivation of them will be presented separately.

In general, the packet drop probability of red, yellow and green packets in a Multi-RED queue are the functions of the virtual queue size ν:

$$p^{dr}(\nu_r) = \begin{cases} 0 & \text{if } \nu_r < minth_r \\ \dfrac{(\nu_r - minTh_r)\, Pmax_r}{maxTh_r - minTh_r} & \text{if } minTh_r \leq \nu_r \leq maxTh_r \\ 1 & \text{if } \nu_r > maxTh_r \end{cases} \tag{4.39}$$

$$p^{dy}(\nu_y) = \begin{cases} 0 & \text{if } \nu_y < minth_y \\ \dfrac{(\nu_y - minTh_y)\, Pmax_y}{maxTh_y - minTh_y} & \text{if } minTh_y \leq \nu_y \leq maxTh_y \\ 1 & \text{if } \nu_y > maxTh_y \end{cases} \tag{4.40}$$

$$p^{dg}(\nu_g) = \begin{cases} 0 & \text{if } \nu_g < minth_g \\ \dfrac{(\nu_g - minTh_g)\, Pmax_g}{maxTh_g - minTh_g} & \text{if } minTh_g \leq \nu_g \leq maxTh_g \\ 1 & \text{if } \nu_g > maxTh_g \end{cases} \tag{4.41}$$

where ν_r, ν_y, ν_g are sizes of virtual queues for red, yellow and green packets, respectively. The size of a physical queue is denoted as q.

In the case of the WRED queue, the packet drop probabilities for all colors are calculated using the overall number of packets in the queue. Therefore, the virtual queue sizes of all colors are equal to the physical queue size, i.e., $\nu_r = \nu_y = \nu_g = q$. If the queue is in state i the physical queue size q equals i. Thus, the overall packet drop probability at state i, denoted for the WRED queue as $P_i^{d,WRED}$, can be found as follows:

$$P_i^{d,WRED} = P^{qr} p^{dr}(i) + P^{qy} p^{dy}(i) + P^{qg} p^{dg}(i) \tag{4.42}$$

The case of the RIO-C queue is more complex. The size of the virtual queue size of red packets is calculated by summing all packets in the queue, i.e., it equals the physical queue size ($\nu_r = q = i$). The size of the virtual queue of yellow packets is calculated by summing only the number of green and yellow

packets. Therefore, if the physical queue size is i, the actual size of the virtual queue of yellow packets may vary from 0 to i. The former case means that there are only red packets in the queue, while in the latter case there are no red packets. Thus, for a given i the size of the virtual queue of yellow packets is a random variable with the Bernoulli (binomial) distribution. The probability that the virtual queue size ν_y is j given that the physical queue size $q = i$ is as follows:

$$P\left[\nu_y = j \,|\, q = i\right] = \binom{i}{j} \left(P^{qg} + P^{qy}\right)^j \left(1 - P^{qg} - P^{qy}\right)^{i-j} \tag{4.43}$$

where $\binom{i}{j}$ is a binomial coefficient $\frac{i!}{j!\,(i-j)!}$. Only green packets are taken into account while calculating the size of the virtual queue of green packets. The probability that the virtual queue size ν_g is j given that the physical queue size $q = i$ is as follows:

$$P\left[\nu_g = j \,|\, q = i\right] = \binom{i}{j} \left(P^{qg}\right)^j \left(1 - P^{qg}\right)^{i-j} \tag{4.44}$$

The overall packet drop probability at state i, denoted for the RIO-C queue as $P_i^{d,RIO-C}$, can be found as follows:

$$\begin{aligned} P_i^{d,RIO-C} =& P^{qr} p^{dr}(i) + P^{qy} \sum_{j=0}^{i} p^{dy}(j)\, P\left[\nu_y = j \,|\, q = i\right] \\ &+ P^{qg} \sum_{j=0}^{i} p^{dg}(j)\, P\left[\nu_g = j \,|\, q = i\right] \end{aligned} \tag{4.45}$$

Application of formula (4.42) or (4.45) to (4.38) leads to finding λ_i for $1 \leq\ i \leq\ Q_{max} - 1$. The next step is computation of the state probabilities.

Two queuing systems were considered as candidates for a model of Multi-RED queue: M(n)/M/1/Q_{max} and M(n)/D/1/Q_{max}. Intuitively, the latter should be chosen, since, in reality, the service rate is not exponentially distributed but deterministic, stemming from the packet size and bottleneck capacity. However, models of both queuing systems are provided.

The state probabilities in an M(n)/M/1/Q_{max} system are as follows [60]:

$$\pi_i = \frac{\prod_{k=0}^{i-1} \frac{\lambda_k}{\mu}}{\sum_{j=0}^{Q_{max}} \prod_{k=0}^{j-1} \frac{\lambda_k}{\mu}} \tag{4.46}$$

State probabilities for M(n)/D/1/Q_{max} cannot be expressed in a closed form. They can be calculated using the algorithm proposed by Gupta *et al.* [62, 63].

Having the state probabilities found, the average queue size, packet drop probabilities per color, overall packet drop probability and goodput (the fraction of throughput that finally enters the bottleneck) per aggregate and for the whole system can be calculated. The average queue size is found form the well known formula:

$$\bar{q} = \sum_{i=0}^{Q_{max}} i\,\pi_i \tag{4.47}$$

This formula is valid for both: M(n)/M/1/Q_{max} and M(n)/D/1/Q_{max} queuing systems. The average packet drop probabilities of red, yellow and green packets in the case of the WRED queue are:

$$\bar{P}^{dr} = \sum_{i=0}^{Q_{max}} \pi_i p^{dr}(i) \tag{4.48}$$

$$\bar{P}^{dy} = \sum_{i=0}^{Q_{max}} \pi_i p^{dy}(i) \tag{4.49}$$

$$\bar{P}^{dg} = \sum_{i=0}^{Q_{max}} \pi_i p^{dg}(i) \tag{4.50}$$

respectively. Similarly, the following formulas express the average packet drop probabilities for the RIO-C queue:

$$\bar{P}^{dr} = \sum_{i=0}^{Q_{max}} \pi_i p^{dr}(i) \tag{4.51}$$

$$\bar{P}^{dy} = \sum_{i=0}^{Q_{max}-1} \pi_i \sum_{j=0}^{i} p^{dy}(j)\, P\left[\nu_y = j | q = i\right] + \pi_{Q_{max}} \tag{4.52}$$

$$\bar{P}^{dg} = \sum_{i=0}^{Q_{max}-1} \pi_i \sum_{j=0}^{i} p^{dg}(j)\, P\left[\nu_g = j | q = i\right] + \pi_{Q_{max}} \tag{4.53}$$

The overall packet drop probability is given by the formula:

$$\bar{P}^{d} = P^{qr}\bar{P}^{dr} + P^{qy}\bar{P}^{dy} + P^{qg}\bar{P}^{dg} \tag{4.54}$$

for both WRED and RIO-C queues. The only difference is that $\bar{P}^{dr}$, $\bar{P}^{dy}$ and $\bar{P}^{dg}$ are calculated from formulas (4.48) – (4.50) in the case of WRED or from

(4.51) – (4.53) for RIO-C. The last characteristic to be derived is the system's goodput denoted as B^{out}. It is expressed as:

$$B^{out} = B\left(1 - \bar{P}^{d}\right) \tag{4.55}$$

4.4 TCP throughput

This section presents a model for TCP throughput proposed by Padhye *et al.* [122]. This model is used in this dissertation to present an example of a more complex model of the DiffServ network including models for meter/markers and droppers (see Section 4.5). The model characterizes the steady state throughput for a bulk transfer TCP flow, i.e., a flow generated by a TCP source having an unlimited amount of data to send. Such a TCP source is often called infinite, greedy or long-lived source. The flavor of TCP modeled by Padhye *et al.* is Reno. Such a traffic model was often used by other researchers [1, 27, 34, 41, 50, 51, 92, 93, 101, 102, 119, 122, 135, 146]. The model captures the effect of TCP's fast retransmit and retransmission timeout mechanisms on the throughput. It reflects TCP dynamics very well and is commonly used. It is suitable for modeling traffic generated by applications transferring large files such as peer-to-peer applications or FTP. Such a type of traffic constitutes a significant portion of the whole traffic volume in the Internet [80, 118]. The work by Padhye *et al.* was cited over 500 times.

Throughput generated by a single long-lived TCP source is expressed by the Padhye formula [122] as a function of packet drop probability p, round trip time RTT and retransmission timeout RTO:

$$B = \min\left(\frac{cwnd_{max}}{RTT}, \frac{1}{RTT\sqrt{\frac{2bp}{3}} + RTO\min\left(1, 3\sqrt{\frac{3bp}{8}}\right)p\left(1+32p^2\right)}\right) \tag{4.56}$$

where b is the number of packets that are acknowledged by a single acknowledgment (ACK). The parameter $cwnd_{max}$ denotes the maximum congestion window size of the TCP source. Throughput is expressed in packets per second while $cwnd_{max}$ in packets.

4.5 Modular model

This section presents a modular mathematical model of the system consisting of a DiffServ network implementing AF PHB and several TCP source-destination

pairs generating the traffic. The overall model shows a practical use of the models for meter/markers and droppers proposed in Sections 4.2 and 4.3. The idea of a modular model was introduced in [140].

The system is described by a set of equations. Solving them numerically allows finding the following characteristics of the system:

- packet marking probability per traffic aggregate,
- drop probability per traffic aggregate at a bottleneck router,
- average throughput and goodput,
- average queue length at the bottleneck router,
- queuing delay at the bottleneck router,
- drop probability per AF drop precedence.

The model helps predicting the system behavior under various conditions. One can choose a number of TCP flows and traffic aggregates sharing a single AF class. Several parameters of TCP sources can be chosen. Each traffic aggregate could be assigned a different target rate. Parameters of DiffServ node elements can be configured.

A strong point of the model is that it is modular. It makes it easy to develop a model of a system consisting of various types of DiffServ node elements (meters/markers and droppers) just by changing some of the equations. Various network topologies can be described.

4.5.1 Network architecture and assumptions

A logical diagram of a network architecture assumed for presentation of a modular model is shown in Fig. 4.6. There is a DiffServ domain consisting of two edge routers and one core router. One link, namely from the core to the edge egress router, is a bottleneck, that is, it is congested. For that reason, the core router will be also called the bottleneck router. It is assumed that congestion occurs only at the core router, that is capacities of other links are sufficient to serve the traffic. It means that packet drops occur only in the core router. Packets are assumed to be dropped mainly due to the early drop mechanism rather than tail drop. It means that significant impact of the tail drop on TCP performance is reduced. A scheduler is not used here since the router serves a single AF class and, what follows, there is only one physical queue. The bottleneck capacity is not shared but entirely dedicated to the AF class.

Traffic sources and destinations are located outside the domain. It is assumed that all sources use the Reno flavor of TCP. All sources are long-lived, that is,

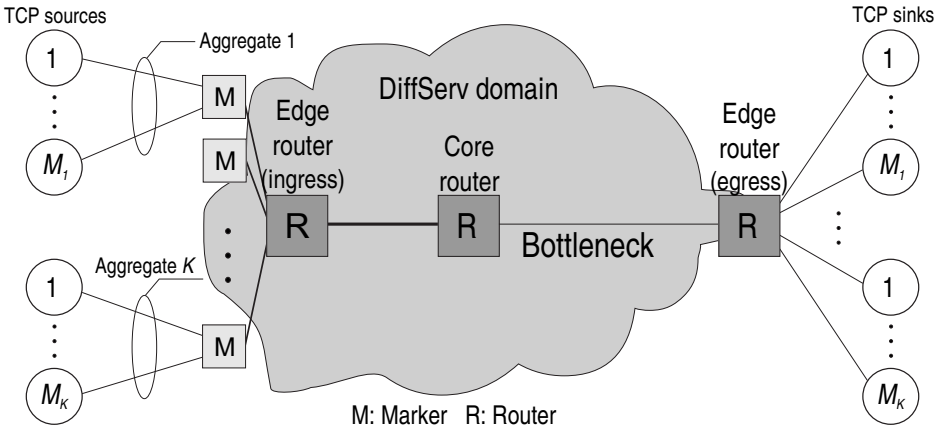


Figure 4.6: Logical diagram of network architecture for modular model

they have unlimited data to send. The number of TCP sources is assumed to be constant for a long period of time. TCP flows differ from each other in value of RTT stemming from link propagation delays and queuing delay at the core router. All TCP flows are grouped into K traffic aggregates (in the simplest case there could be only one aggregate). Each traffic aggregate is assigned a target rate it should achieve while passing a DiffServ domain. Each aggregate k consists of M_k TCP sources.

All packets are assigned to a single AF class but each traffic aggregate is measured and marked by a separate meter/market entity at the edge ingress router. A part of available bottleneck capacity is dedicated to each aggregate by appropriate setting of meter/marker parameters. Packets receive a codepoint indicating one of three drop precedences according to the traffic aggregate's conformance to the contract rate. Individual flows are not distinguished by the meter/marker used to police the aggregate they constitute. The core router, in turn, does not distinguish packets belonging to various aggregates. It relies on the packet color only. Assuming that meter/markers are well configured, the bottleneck capacity should be shared between aggregates fairly and accordingly to aggregates' target rates.

4.5.2 Set of equations

In order to develop a set of equations describing behavior of the system presented in Fig. 4.6 the formulas provided in previous sections must be slightly modified. The throughput generated by single TCP source m belonging to k-th aggregate

is expressed as:

$$B_{k,m} = \min\left(\frac{cwnd_{max}}{RTT_{k,m}}, \frac{1}{RTT_{k,m}\sqrt{\frac{2b\bar{P}_k^d}{3}} + RTO_{k,m}\min\left(1, 3\sqrt{\frac{3b\bar{P}_k^d}{8}}\right)\bar{P}_k^d\left(1+32\ \bar{P}_k^{d\,2}\right)}\right) \tag{4.57}$$

where $k = 1, \ldots, K$ and $m = 1, \ldots, M_k$. This formula is obtained from (4.56), by substituting RTT, RTO, and p with $RTT_{k,m}$, $RTO_{k,m}$, and $\bar{P}_k^d$, respectively. Aggregate k experiences the packet drop probability denoted here as $\bar{P}_k^d$. As assumed before, packets are subject to dropping at the bottleneck router only, so $\bar{P}_k^d$ denotes, in fact, the packet drop probability at that router.

Values of $RTT_{k,m}$ and $RTO_{k,m}$ are computed as follows [158]:

$$RTT_{k,m} = d_{k,m} + d_q \tag{4.58}$$

$$RTO_{k,m} = RTT_{k,m} + 4d_q \tag{4.59}$$

where d_q is a queuing delay at the bottleneck router and $d_{k,m}$ is a propagation delay along the source-destination-source path traversed by packets of m-th flow of k-th aggregate. The propagation delay is assumed to be constant and does not depend on traffic conditions or DiffServ parameters. As implemented by TCP, RTO is computed as the RTT plus four times the variance of the RTT. The variance of RTT is approximated by the average queuing delay at the bottleneck [158]. In turn, queuing delay d_q may be expressed as a ratio of average queue size $\bar{q}$ (formula 4.47) and bottleneck capacity μ:

$$d_q = \frac{\bar{q}}{\mu} \tag{4.60}$$

The total throughput generated by k-th aggregate can be expressed as:

$$B_k = \sum_{m=1}^{M_k} B_{k,m} \tag{4.61}$$

where M_k is the number of TCP flows constituting the aggregate. The sum of all B_k constitutes the overall throughput (a total traffic offered) generated to the network:

$$B = \sum_{k=1}^{K} B_k \tag{4.62}$$

A fraction of traffic is lost at the bottleneck router. Drop probabilities experienced by aggregates may differ. If aggregate k experiences packet drop probability $\bar{P}_k^d$ then, the fraction of traffic of k-th aggregate that finally enters the bottleneck link is as follows:

$$B_k^{out} = B_k \left(1 - \bar{P}_k^d\right) \tag{4.63}$$

The total traffic that is served at the bottleneck queue, i.e., the goodput is as follows:

$$B^{out} = \sum_{k=1}^{K} B_k^{out} \tag{4.64}$$

The bottleneck router does not distinguish packet's aggregate membership. It takes into account only a packet's color while making the decision whether to drop or forward the packet. Thus, packets of a particular color experience the same drop probability. However, the drop probability per aggregate depends on the probability that the packet within the aggregate is red, yellow or green. The packet drop probability for aggregate k can be expressed as follows:

$$P_k^d = P_k^{mr} \bar{P}^{dr} + P_k^{my} \bar{P}^{dy} + P_k^{mg} \bar{P}^{dg} \tag{4.65}$$

In the above formula, P_k^{mr}, P_k^{my} and P_k^{mg} denotes marking probabilities for k-th aggregate obtained form (4.6)–(4.8), (4.11)–(4.13) or (4.28)–(4.30) if srTCM, trTCM or TSW3CM meter/marker, respectively, is used for the aggregate. The index k was introduced here to distinguish meter/markers serving different aggregates. It allows the use of various types of meter/markers and their configuration parameters for various aggregates in the modular model. For example, if the meter/marker serving for aggregate k is srTCM, the probability that the packet is marked red is obtained from (4.6):

$$P_k^{mr} = \frac{1}{\sum\limits_{l=0}^{CBS_k+EBS_k} \left(\frac{CIR_k}{B_k}\right)^l} \tag{4.66}$$

To complete the set of equations describing the whole system, the probabilities that the packet entering the Multi-RED queue is red, yellow or green must be calculated. Since several traffic aggregates, each with a different color structure, contribute to the traffic incoming to the bottleneck router, these probabilities should be calculated as follows:

$$P^{qr} = \frac{\sum_{k=1}^{K} P_k^{mr} B_k}{B} \tag{4.67}$$

$$P^{qy} = \frac{\sum_{k=1}^{K} P_k^{my} B_k}{B} \tag{4.68}$$

$$P^{qg} = \frac{\sum_{k=1}^{K} P_k^{mg} B_k}{B} \tag{4.69}$$

The color structure of aggregate k is expressed by the marking probabilities for that aggregate: P_k^{mr}, P_k^{my}, P_k^{mg}.

In most equations the throughput is expressed in packets per second and denoted by B, in general. However, in the case of the TSW3CM meter/marker, the throughput is expressed in bits per second and denoted by D. Reminding, the relationship between B and D is given by:

$$D = B \cdot S \tag{4.70}$$

where S is the packet size in bits.

The final set of equations describing the network architecture presented in Fig. 4.6 and TCP behavior in such a network can be obtained from formulas (4.57) – (4.69) supplemented with appropriate formulas for packet marking probabilities (Section 4.2) and packet drop probabilities (Section 4.3). This large number of equations can be reduced to a set of $K + 1$ equations with unknowns: $\bar{q}$ and P_k^d for $k = 1, \ldots, K$. Constants are: the meter/marker configuration parameters, Multi-RED parameters and bottleneck queue size, link propagation delays, bottleneck capacity and parameters of TCP sources. These equations are complex and, therefore, the closed form solution cannot be found. The solutions, however, can be found numerically. Apart from the above unknowns, the following characteristics can be found: B, B^{out}, B_k^{out}, P_k^{mr}, P_k^{my}, P_k^{mg}, $\bar{P}_k^d$, $\bar{P}^d$, $\bar{P}^{dr}$, $\bar{P}^{dy}$, $\bar{P}^{dg}$, and d_q.

5 Verification of the models

It doesn't matter how beautiful your theory is, it doesn't matter how smart you are if it doesn't agree with experiment, it's wrong.

— Richard P. Feynman

Numerous simulations were performed to experimentally validate the mathematical models presented in Chapter 4. Selected, representative experiments are presented in this section. Discussion of the results of a particular experiment is provided immediately for each experiment. The presentation is preceded by the description of the simulation model and tools and discussion on the statistical credibility.

5.1 Background

Stochastic simulation is a commonly used tool for performance evaluation of telecommunications systems and validation of various models describing the phenomena occurring in the systems. To assure that the results are credible, several issues must be taken into account. The credibility of simulations depends on the selected tools, type of simulation and technique of collecting data, statistical analysis process and simulation model. The following sections present how those issues were addressed. The network model used for simulations as well as some additional assumptions are also presented.

5.1.1 Tools

The simulations were performed with `ns-2` simulator [114], version 2.27. Although it supports the DiffServ model, several modifications to this version of `ns-2` were necessary. The module responsible for generating simulation traces was adapted to make it possible to collect desired data related to TCP flows and queue sizes. The web traffic model was supplemented with a module generating a more representative web-like traffic consistent with the SURGE (Scalable URL Reference Generator) model [11].

Since each single simulation run produces up to dozens gigabytes of trace, a dedicated software was developed to pre-process the output data from simulation to obtain desirable statistics. Namely, the software gives per interval statistics such as the number of packets (forwarded, dropped, marked, etc.) observed during the interval, the number of bytes, *RTT* statistics and other. Depending on the experiments, the statistics can be collected for the whole system and on per-aggregate and per-flow basis. This software was written in `C++` language. Such a preprocessing not only saves the disk space but quicken the operation as well, since the disk in/out operations are very time-consuming.

The final processing of the simulation data as well as solving the model equations were performed with the *MATHEMATICA*® package [154]. This tool was used for statistical processing, estimation of expected values and confidence intervals for examined variables, and presentation of the results.

5.1.2 Simulation type and technique for data collection

The steady-state type simulation was used. It means that the steady state of the system is assumed to be maintained during the simulation period.

The technique for collecting simulation data used in the research is *the method of batch means* also called *the subinterval method* [149]. Only one run of the simulation is needed in this method. Therefore, the total overhead from the warm-up period of the simulation is less than in, e.g., *the independent replication method* that involves n independent runs of the simulation, each with its own warm-up period. Many of the simulations performed are very time-consuming and have a relatively long warm-up period so the choice of *the method of batch means* resulted in saving time. Especially, simulations with web-like traffic sources are characterized by a very long warm-up period.

5.1.3 Parameter estimation

In general, the simulation process produces a sequence of n independent observations of a random variable: $x_1, x_2, \cdots, x_n$. Let x_i be the observation from i-th

® *MATHEMATICA* is a registered trademark of Wolfram Research

interval. The standard point estimator[5] of mean μ_x is [124, 149]:

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{5.1}$$

In fact, $\hat{\mu}$ is a random variable. The accuracy of such an estimate can be assessed by the probability that unknown average μ_x holds in certain bounds called confidence interval. This probability is defined as follows:

$$P\left(\hat{\mu} - \Delta_x \leq \mu_x \leq \hat{\mu} + \Delta_x\right) = 1 - \alpha \tag{5.2}$$

where Δ_x is the half of the symmetric confidence interval for estimator $\hat{\mu}$, while $1-\alpha$ is a confidence level (α is called a significance level). In other words, if the confidence interval $2\Delta_x$ is found for assumed confidence level $(1-\alpha)$, $100(1-\alpha)\%$ of observations of the unknown average falls within the interval $(\hat{\mu} - \Delta_x, \hat{\mu} + \Delta_x)$.

If observations $x_1, x_2, \cdots, x_n$ can be regarded as realizations of independent and normally distributed random variables the half of the confidence interval is expressed as [149]:

$$\Delta_x = t_{n-1,1-\alpha/2}\frac{\hat{\sigma}\left[\hat{\mu}\right]}{\sqrt{n}} \tag{5.3}$$

where $t_{n-1,1-\alpha/2}$ is the $(1-\alpha/2)$-th quantile of the Student-t distribution with $n-1$ degrees of freedom, and $\hat{\sigma}\left[\hat{\mu}\right]$ is a square root of the unbiased estimator of variance of $\hat{\mu}$:

$$\hat{\sigma}^2\left[\hat{\mu}\right] = \frac{1}{(n-1)}\sum_{i=1}^{n}\left(x_i - \hat{\mu}\right)^2 \tag{5.4}$$

If the number of observations n is greater than 30 then the quantile $t_{n-1,1-\alpha/2}$ can be replaced with the $(1-\alpha/2)$-th quantile of the standard normal distribution. Commonly, the assumed value of parameter α is 0.05. Hence, the confidence level is 95%, i.e., the estimated mean of the observed random variable falls within the confidence interval with the probability 0.95.

Confidence intervals can be found from expression (5.3) even if the observations $x_1, x_2, \cdots, x_n$ do not represent normally distributed random variables. However, the observations must be realizations of independent and identically distributed (i.i.d) random variables. If this condition is met, then according to the central limit theorem, the distribution of $\hat{\mu}$ tends to the normal distribution for large n. Generally, if $n > 100$, expression (5.3) is regarded as a good approximation of the confidence interval [124].

If the observations $x_1, x_2, \cdots, x_n$ are not realizations of i.i.d. random variables then different estimators of the variance of $\hat{\mu}$ are used. This, however, is not the case in experiments in this dissertation.

[5] According to the standard notation, $\hat{a}$ denotes an estimator of parameter a

The discussed problems regarding the estimation of confidence intervals of the estimated sample mean raise some issues important from the perspective of credibility of experimental results. These issues will be discussed in Section 5.1.4.

Examples of data collection and analysis

After ignoring the warm-up period (refer to Section 5.1.4 for the method for determining its length) the rest of the simulation period is divided into equal-length nonoverlapping intervals. During each interval i the following statistics are collected:

- the total number of packets observed φ_i^{all},
- the total number packets dropped at the core router φ_i^{drop},
- the number of observed green $\varphi_i^{all,green}$, yellow $\varphi_i^{all,yellow}$, or red $\varphi_i^{all,red}$ packets,
- the number of green $\varphi_i^{drop,green}$, yellow $\varphi_i^{drop,yellow}$, and red $\varphi_i^{drop,red}$ packets that were dropped,
- the sum of bytes of all packets Ψ_i,
- estimates of the average size of the physical queue at the core router $\hat{\theta}_i$, and virtual queues for green $\hat{\theta}_i^{green}$, yellow $\hat{\theta}_i^{yellow}$ and red $\hat{\theta}_i^{red}$ packets.

Having the above statistics, the average values of system parameters can be estimated. For example, a single sample of the packet drop probability for i-th interval is:

$$\hat{p}_i^d = \frac{\varphi_i^{drop}}{\varphi_i^{all}} \tag{5.5}$$

while the estimator of the packet drop probability is:

$$\hat{P}^d = \frac{1}{n}\sum_{i=1}^{n} \hat{p}_i^d \tag{5.6}$$

In a similar way packet marking probabilities can be found. For example, the estimate of the probability that packet is marked red is obtained from:

$$\hat{p}_i^{mr} = \frac{\varphi_i^{all,red}}{\varphi_i^{all}} \tag{5.7}$$

$$\hat{P}^{mr} = \frac{1}{n}\sum_{i=1}^{n} \hat{p}_i^{mr} \tag{5.8}$$

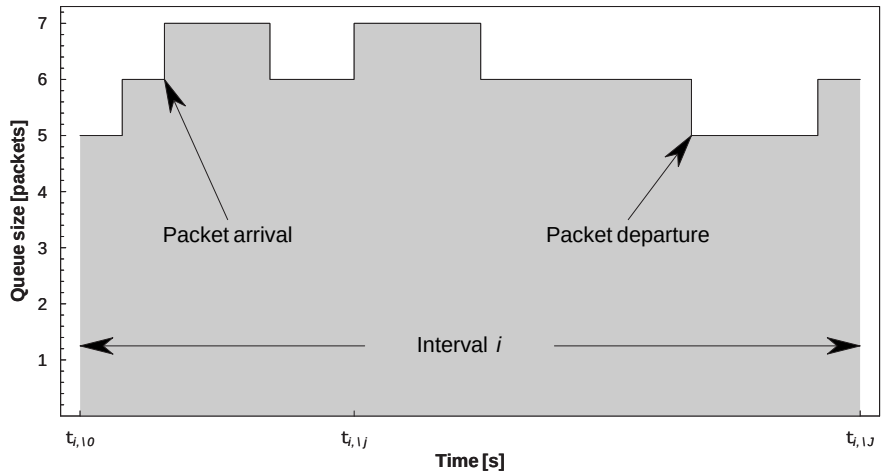


Figure 5.1: Estimation of a single sample of average queue size

Similarly, other packet marking and packet drop probabilities can be estimated. The average throughput is estimated as a ratio of the total number of bytes sent during interval, Ψ_i, to interval length L^{int}:

$$\hat{B} = \frac{1}{n} \sum_{i=1}^{n} \frac{\Psi_i}{L^{int}} \tag{5.9}$$

Depending on particular experiment needs, all the above statistics can be gathered for single flows, aggregates and the sum of all aggregates. Hence, estimates of throughput, probabilities of packet marking and dropping can be obtained for flows, aggregates and for the whole traffic.

A special comment is needed regarding the estimation of queue sizes. To obtain a single sample of the queue size correctly, the changes of its size must be observed on the event by event basis. For example, instantaneous queue size changes on each packet arrival or departure. In the RED algorithm, the average queue size is updated on each packet arrival. The moments of packet arrivals or departures are random so consecutive queue states remain for various durations. Therefore, a single sample $\hat{\theta}_i$ of the average queue size is obtained by calculating the integral of the graph of the instantaneous queue size over time from $\tau_{i,0}$ to $\tau_{i,J}$, where $\tau_{i,0}$ is the beginning of interval i and $\tau_{i,J}$ is the end of interval i. A sample $\hat{\theta}_i$ is calculated by dividing the obtained integral by the interval length:

$$\hat{\theta}_i = \frac{1}{\tau_{i,J} - \tau_{i,0}} \sum_{j=1}^{J} \vartheta_{i,j-1} \left(\tau_{i,j} - \tau_{i,j-1}\right) \tag{5.10}$$

where $\tau_{i,j}$ is the time of event j within interval i. The final estimate of the average queue size is derived as follows:

$$\hat{q} = \frac{1}{n} \sum_{i=1}^{n} \hat{\theta}_i \tag{5.11}$$

A graphical presentation of this method applied to estimate a single sample of the average queue size is presented in Fig. 5.1. Average sizes of virtual queues for red, yellow and green packets are estimated similarly.

Finally, the way of collecting statistics for *RTT* and *RTO* per a TCP flow should be discussed. Note that this applies only to simulations with infinite TCP sources and the modular model. Values of *RTT* and *RTO* collected during the simulation are real estimates calculated by TCP mechanisms. For each TCP source sums of *RTT* and *RTO* as well as squares of *RTT* and *RTO* are collected and later used for calculating means and variance of those parameters.

5.1.4 Simulation credibility

Several issues have to be considered to achieve valuable results of the experiments. First of all, the credibility of the random number generator (RNG) used in the simulator should be confirmed. The second issue is to assess the length of the simulation warm-up period properly, that is to determine the beginning of the steady state of the simulation.

The selection of *the method of batch means* implies another issue critical to the credibility. In this method the output of a long simulation run is partitioned into adjacent and nonoverlapping intervals. Hence, the sequence of samples (batch-means) is produced. All samples are finally used to estimate expected values of random variables and confidence intervals. Therefore, to avoid correlation between random samples from various intervals the interval length must be sufficiently large. It should also be remembered that the interval should be sufficiently long that enough observations can be gathered to estimate means of random variables with a satisfactory accuracy.

Another issue, important for proper estimation of the variance of the sample mean and confidence intervals, is checking the distribution of the observed samples. As described in Section 5.1.3, it also determines the number of samples that should be collected and, what follows, the total length of the simulation.

Random number generator in ns-2

The `ns-2` simulator uses the implementation of the combined multiple recursive generator MRG32k3a proposed by L'Ecuyer [94]. This is one of the generators

mostly recommended due to its reliability and long cycle [125]. The period of the generator is about 3.1×10^{57}. The code of the generator used in `ns-2` was adopted from [96]. The quality of the generator was tested and it performs well with respect to spectral tests [95, 96]. Since the convincing tests for the RNG's credibility are available, the author did not perform additional tests.

Warm-up period

The appropriate determination of the length of the warm-up period is still critical for the credibility of simulation results. Many methods for finding length of the warm-up period can be found in literature [124]. One of the rough methods is observing the current estimate of system parameters. It can be assumed that the simulation achieved a steady-state phase when the current estimate stops moving in one direction and starts oscillating. Another technique is to assume that the first observation of the steady-state phase is such, that is neither minimum nor maximum of the rest of observations. Those techniques, however, can be missleading and can be used only for quick and rough estimation of the warm-up period's length.

A less crude approach is adopted from the central limit theorem [124, 149]. If a given distribution is stationary, then its sample mean is normally distributed with the mean μ and standard deviation:

$$s = \sigma/\sqrt{n} \tag{5.12}$$

where μ and σ are true mean and standard deviation of the sampled distribution, respectively. From (5.12) the following linear equation can be obtained:

$$\ln s = -0.5 \ln n + \ln \sigma \tag{5.13}$$

Therefore, if the simulation achieves a steady-state period, the distribution of sample means becomes stationary and the graph of the logarithm of the standard deviation against the logarithm of n is expected to show a decrease with a slope of ≈ -0.5.

The application of this method requires several independent simulation runs. If the simulation is repeated M times and N observations are made during each run, a single observation is denoted as $X_{n,m}$, where $n = 1, \ldots, N$ and $m = 1, \ldots, M$. The mean of the first observations from all runs is computed. Similarly, means of the second, third, etc., observations are computed. In general, the following sequence of sample means is obtained:

$$\hat{\mu}_n = \frac{1}{M} \sum_{m=1}^{M} X_{n,m} \tag{5.14}$$

This sequence consists of N independent normally distributed random variables. The standard deviation s_N can be calculated as follows:

$$s_N = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N}\left(\hat{\mu}_n - \hat{\mu}_N\right)^2} \tag{5.15}$$

where $\hat{\mu}_N$ is the overall mean computed as $\hat{\mu}_N = N^{-1}\sum_{n=1}^{N}\hat{\mu}_n$. Now, the graph of $\ln s_N$ against $\ln N$ can be drawn. The point, at which the curve begins to decrease with a slope of about -0.5 can be assumed as the end of the warm-up period.

If the fluctuations of samples $\hat{\mu}_n$ are high it may be difficult to catch the curve's trend. Then, it is advisable to apply *moving average* over a number of samples to smooth out the high frequency fluctuations [149]. The moving average of length $2k+1$ is as follows:

$$\overline{\hat{\mu}_n}\,(k) = \begin{cases} (2k-1)^{-1}\sum_{i=-k}^{k}\hat{\mu}_{n+i} & \text{if } n \geq k+1 \\ (2n-1)^{-1}\sum_{i=-n+1}^{n-1}\hat{\mu}_{n+i} & \text{if } n < k+1 \end{cases} \tag{5.16}$$

The value of k should not be too high not to distort the long term trend but large enough to remove short term fluctuations. After performing the moving average algorithm, the smoothed samples $\overline{\hat{\mu}_n}\,(k)$ can be used in (5.15) instead of $\hat{\mu}_n$ from (5.14).

In practice, the method based on the central limit theorem was applied to two representative simulation runs within each group of similar runs. The finally assumed warm-up period was about 50% longer than indicated by this method. The rest of simulations within a group of simulations were verified by the use of rough methods described in the first paragraph of this section. If a wrong warm-up period assesment was suspected the method based on the central limit theorem was applied. On average, 2% of the simulations per group were recognized to have wrongly assesed the length of the warm-up period and were disregarded. Example of determination of the warm-up period is presented in Appendix A.1.

The independence of samples

Since the sequence of samples in *the method of batch means* originates from a single simulation run, there is a risk of the samples being correlated. Therefore, special techniques should be used to check the independence of the samples. In this dissertation, the following two techniques were used in parallel to check whether the correlation between batches is negligible: assessment of the autocorrelation function (ACF) and the Ljung-Box test.

In the first method the autocorrelation function has to be estimated. The recommended estimator of the autocorrelation function is [22, 151]:

$$\hat{\rho}(k) = \hat{\gamma}(k) / \hat{\gamma}(0) \tag{5.17}$$

where $\hat{\gamma}(k)$ is the estimator of the autocovariance of lag k. In fact, $\hat{\rho}(k)$ is the normalized measure of covariance. In turn, the recommended estimator of the autocovariance function is as follows [151]:

$$\hat{\gamma}(k) = \frac{1}{n} \sum_{i=1}^{n-k} (X_{i+k} - \hat{\mu})(X_i - \hat{\mu}) \tag{5.18}$$

where n is the number of samples, X_i is a single sample and $\hat{\mu}$ is the sample mean.

Equation (5.17) is used to plot the graph of the autocorrelation function. For large n and $k = 1, 2, \cdots, h$, $\hat{\rho}(k)$ has a normal distribution with the mean zero and variance $1/n$. Therefore, the confidence intervals for $\hat{\rho}(k)$ can be constructed. The most commonly assumed confidence level is 95%. Hence, the confidence interval is $(-1.96/\sqrt{n}, 1.96/\sqrt{n})$, where 1.96 is 0.975 quantile of the standard normal distribution [22]. It is assumed that there is no point to reject the hypotesis that the samples are uncorrelated if 95% of the sample autocorrelations for lags $k = 1, 2, \cdots, h$ lie within bounds $\pm 1.96/\sqrt{n}$.

Although it is possible to calculate autocorrelations for all lags up to $n - 1$, it is inadvisable to do so because the estimates of $\hat{\rho}(k)$ of a higher order are less reliable. It stems from the fact that the higher k the fever pairs of samples available to calculate the autocovariance. In the worst case, for $k = n-1$ only one sample contributes to estimation of $\hat{\gamma}(k)$. Hence, several recommendations on how to select the maximum lag h were proposed. The most common suggestions are: $h = n/4$ [19], $h = \sqrt{n} + 10$ [36] and $20 \leq h \leq 40$ [21].

The second technique for determining whether the samples are uncorrelated is the Ljung-Box test [22, 99]. This test is based on the following statistic:

$$Q_{LB} = n(n+2) \sum_{i=1}^{h} \frac{\hat{\rho}^2(i)}{n-i} \tag{5.19}$$

where n is the number of samples and h is the number of meaningful lags of the autocorrelation function (maximum lag). The statistic Q_{LB} is a random variable having approximately the chi-square distribution with h degrees of freedom. Since the chi-square distribution with h degrees of freedom and Q_{LB} statistic are known, the corresponding p-value can be calculated. There is no evidence to reject the hypothesis that the samples are uncorrelated on the significance level

α if p-value is grater than α. If p-value $\leq \alpha$ the hypothesis that the samples are uncorrelated is ruled out. The most commonly assumed significance level is 0.05.

There exists a class of processes that the above techniques are not sufficient to verify that their realizations produce i.i.d random variables. The *ARCH* (autoregressive conditional heteroscedasticity) process initially described in [44] and its generalization *GARCH* [16] are good examples. The series produced by the *ARCH* process successfully passes the Ljung-Box test and there is no point to reject the hypothesis of lack of correlation on the basis of ACF as well. Nevertheless, the correlation of samples' squares is significant. On that basis two other tests were developed. The first test relies on the plot of ACF of samples squares and the same rules for accepting or rejecting the hypothesis on lack of correlation between samples (see above). The second technique is the test by McLeod and Li [107]. The McLeod-Li statistic is similar to the one of the Ljung-Box test but the autocorrelation of samples is replaced by autocorrelation of squares of samples $\hat{\rho}_{SQ}$:

$$Q_{ML} = n\,(n+2) \sum_{i=1}^{h} \frac{\hat{\rho}_{SQ}^2\,(i)}{n-i} \tag{5.20}$$

The above techniques are initially applied for an arbitrarily selected length of intervals. Usually, the minimum length necessary to collect enough observations for a reasonable precision of estimation of the sample batch mean is selected. If it occurs that the samples are correlated, the interval is lengthened and the whole procedure is repeated.

Examples of the above tests performed for the experiments related to the dissertation are presented in Appendix A.2.

The distribution of samples

As indicated in Section 5.1.3, if the samples within the series are i.i.d. random variables but are not normally distributed, formula (5.3) can be used to determine confidence intervals if the number of samples exceeds 100. Then, in *the method of batch means*, the single run must be divided into at least 100 intervals what results in a long lasting simulation. If the condition of the normal distribution of samples is satisfied the number of intervals can be reduced to 30. To make sure that the samples are normally distributed, the Shapiro-Wilk normality test was performed [137]. Refer to Appendix A.2 for the example.

5.1.5 Types of traffic

Two types of traffic were used in the experiments. One set of simulations was performed with long-lived TCP sources, that is, always having unlimited data

Table 5.1: Web traffic generator elements' parameters used in simulations

Element	Distribution	PDF	Parameters
Object sizes — Body	Lognormal	$h(x) = \frac{e^{\frac{-(\ln(x)-\mu)^2}{2\sigma^2}}}{x\sigma\sqrt{2\pi}}$	$\mu = 9.357$ $\sigma = 1.318$
Object sizes — Tail	Pareto	$h(x) = \alpha k^{\alpha} x^{-\alpha-1}$	$k = 133000$ $\alpha = 1.1$
Inter-object time	Weibull	$h(x) = \frac{bx^{b-1}}{a^b} e^{-\left(\frac{x}{a}\right)^b}$	$a = 1.46$ $b = 0.382$
Inter-page time	Pareto	$h(x) = \alpha k^{\alpha} x^{-\alpha-1}$	$k = 3$ $\alpha = 1.5$
Page size	Pareto	$h(x) = \alpha k^{\alpha} x^{-\alpha-1}$	$k = 1.699$ $\alpha = 2.43$

to send. A separate group of simulations were done with sources generating a web-like traffic. In both cases, the traffic was generated by TCP Reno sources.

The rationale for simulating such a type of traffic was given in Section 4.5.1. The number of long-lived TCP sources is assumed to be constant during the simulation. The traffic volume of aggregates is varied between experiments by the selection of the number of TCP flows constituting the aggregate.

The web traffic model adapted for the experiments was proposed by Barford and Corvella [11]. On the basis of the observation and measurement of the real web traffic traces they developed the SURGE model (Scalable URL Reference Generator). Their model consists of a few components, each having the probability density function defined. The idea of the model is based on the web page structure and the user behavior. The user downloads consecutive webpages at some interval called *"thinking time"* or inter-page time. The inter-page time was recognized to have the Pareto distribution. The second random variable is a page size. It is expressed as a number of page elements, that is html files, images, flash files, etc. It has the Pareto distribution as well. The next component of the SURGE model is the object size that describes the size of individual elements of the web page. Finally, the time between downloading elements of the web page is defined by the inter-object time. It has the Weibull distribution.

The SURGE model elements, together with respective PDFs and distribution parameters proposed by Barford and Corvella, are shown in Tab. 5.1. This set of parameters was used in the experiments. The traffic volume of aggregates is varied by changing the number of web servers and clients acting in the network.

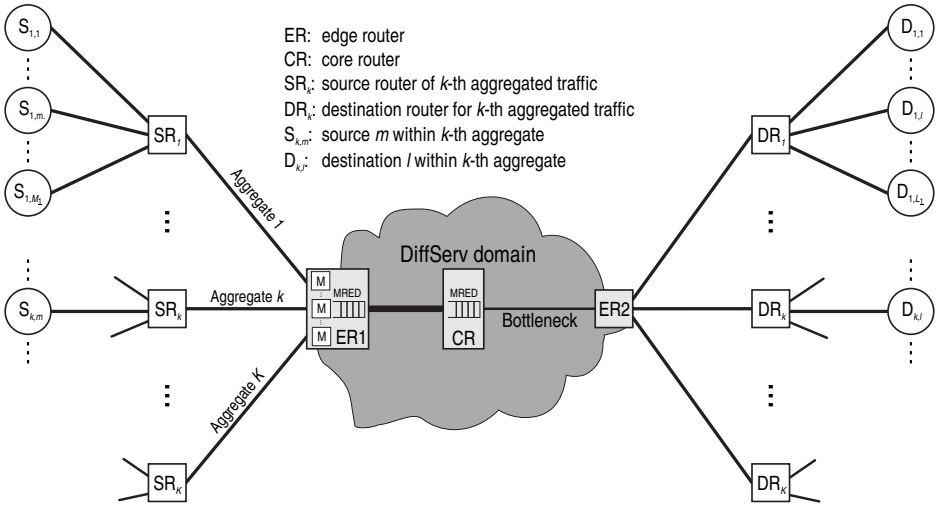


Figure 5.2: Network architecture used in simulations

5.1.6 Network architecture

The general network architecture used for performing simulation experiments is presented in Fig. 5.2. The DiffServ domain consists of two edge routers: ER1 and ER2 as well as a core router CR. All traffic sources and destinations are located outside the domain. The whole traffic generated by sources enters the domain through router ER1 and leaves through router ER2. The traffic volume in the opposite direction is very small and consists only of acknowledgments and, additionally, requests for web pages in the case of web-like traffic. The configuration of the network assures that only packets flowing from sources to destinations can be lost. Drop may occur only at the core router. The packets flowing in the destination-source direction are never lost. These assumptions are especially important in the case of experiments for the modular model because they must be consistent with the assumptions for the model. In the case of experiments with meter/markers or droppers such assumptions do not influence the experimental results and performance of the models. Only the traffic characteristics may be affected.

Traffic sources $S_{k,m}$ are TCP-FTP agents (infinite sources) or web-servers. In the former case TCP sinks serve as traffic destinations $D_{k,l}$. In the case of the web-like traffic, destinations $D_{k,l}$ represent web-clients requesting web pages. $S_{k,m}$ denotes m-th source in k-th aggregate, where $m = 1 \cdots M_k$. $D_{k,l}$ denotes l-th destination in k-th aggregate where $l = 1 \cdots L_k$. The number of sources and

destinations may be selected differently for various experiments. In the case of infinite TCP sources the number of TCP-FTP agents and TCP sinks are equal ($M_k = L_k$ for all k). That is, they always constitute source-destination pairs. The number of such pairs is selected individually for experiments and is constant during the simulation. In the case of simulations with web-like traffic the number of servers and clients is different, namely there are more clients than servers. The number of servers and clients is not changed during the experiment.

It is assumed that all TCP sources use the same value of $cwnd_{max}$. It was equal to 40 packets for most experiments. The receiver does not delay acknowledgments (ACKs), that is, every packet received is acknowledged [50, 132, 144]. Thus, parameter b in formulas (4.56) and (4.57) equals 1. A further assumption is that ACKs are never lost nor delayed while passing the network from the destination to the source host.

The link from CR to ER2 is a bottleneck. Its capacity μ is selected individually for a given experiment. The capacity of link ER1—CR is assumed to be at least 30% greater than μ. As regards links SR_k—ER1 and $S_{k,m}$—SR_k the two approaches were taken. In the first case, the traffic sources are located in a relatively slow Local Area Network. Link capacities increase towards the core network, i.e. the DiffServ domain. They are always sufficient to carry the traffic without losses but are not much greater than the throughput of the aggregate (in case of SR_k—ER1 links) or fraction of the aggregate (in case of $S_{k,m}$—SR_k links). Such a configuration causes the traffic entering the DiffServ domain to be slightly smoothed. In the second approach, it is assumed that capacities of links outside the domain have high capacities and are considerably underutilized. In such a case traffic is not smoothed and may manifest a more bursty nature.

Link propagation delays, excluding $S_{k,m}$—SR_k links, are chosen arbitrarily for each simulation. Propagation delays of links $S_{k,m}$—SR_k are randomly chosen from the uniform distribution with bounds selected individually per traffic aggregate. Therefore, the overall propagation delay is a sum of a randomly chosen propagation delay of link $S_{k,m}$—SR_k and arbitrarily chosen propagation delays of other links. It makes it possible to differentiate flows' RTTs. The connection between propagation delays used in the analytical model and simulation will be explained in a greater detail in Section 5.4.

The number of meter/marker entities at the ingress edge router corresponds to the number of traffic aggregates. The marking algorithm and meter/marker parameters as well as the type Multi-RED queue in DiffServ routers were chosen individually for particular experiments.

The packet size was assumed to be constant during the simulation and equal for all traffic sources. Arguments for such an assumption were discussed in Section 4.1. The packet size used was 500 B or 1 kB including headers, that is, TCP maximum segment size MSS was equal to 460 B or 960 B, respectively.

5.2 Experiments: meter/markers

Each experiment discussed in this section presents characteristics for one particular type and configuration of a meter/marker. In order to obtain the marking probability as a function of the throughput several simulations were run. The throughput of aggregates was varied by changing the number of traffic sources, bottleneck capacity and the number of aggregates sharing the bottleneck capacity. Such a method showed that the meter/marker characteristics and the accuracy of the proposed model do not depend on configuration of other network elements.

Several configurations of meter/markers were tested. Separate experiments were performed with ftp-like traffic and web-like traffic. Results are presented on graphs in both linear and the logarithmic scale. Graphs with the linear scale show the differences between the model and simulation for relatively large marking probabilities, i.e., $0.01 - 1$. The the logarithmic scale is used to facilitate the comparison of the model and simulation in areas where probabilities are small, i.e., $(10^{-7} - 10^{-2})$. The use of the logarithmic scale is also justified by the fact that in those areas discrepancies are sometimes relatively high. It should be noted that very small values of the marking probabilities were sometimes unmeasurable and are reported as equal to 0 and cannot be presented in the logarithmic scale.

5.2.1 Single rate three color marker

Experiment 1 [142] shows a comparison between results obtained from simulation and the analytical model for a sample configuration of srTCM.

Experiment 1

The srTCM parameters in the experiment were configured as follows: $CIR = 2\,\text{Mb/s}$, $CBS = 15\,\text{kB}$, $EBS = 50\,\text{kB}$. Packet size was set to 1 kB. Characteristics of the probability of packet marking as green, yellow and red as a function of the traffic rate for the case of the ftp-like traffic are presented in Fig. 5.3 (the linear scale) and Fig. 5.4 (the logarithmic scale). Characteristics for the case of the web-like traffic are presented in Figures 5.5 and 5.6 in the linear scale and the logarithmic scale, respectively. Solid lines represent characteristics predicted by the model based on the M/M/1/K queuing system (formulas 4.6 — 4.8) while the points mark results obtained from simulation.

The probability that a packet will be marked red or green is well predicted by the model for both types of traffic. Some larger discrepancies can be only noticed in the logarithmic scale for the probability of packet marking as red (P^{mr}) when the value of the marking probability is less than ≈ 0.01 (Fig. 5.4 and Fig. 5.6). In that area, values of the marking probability obtained form simulation are spread

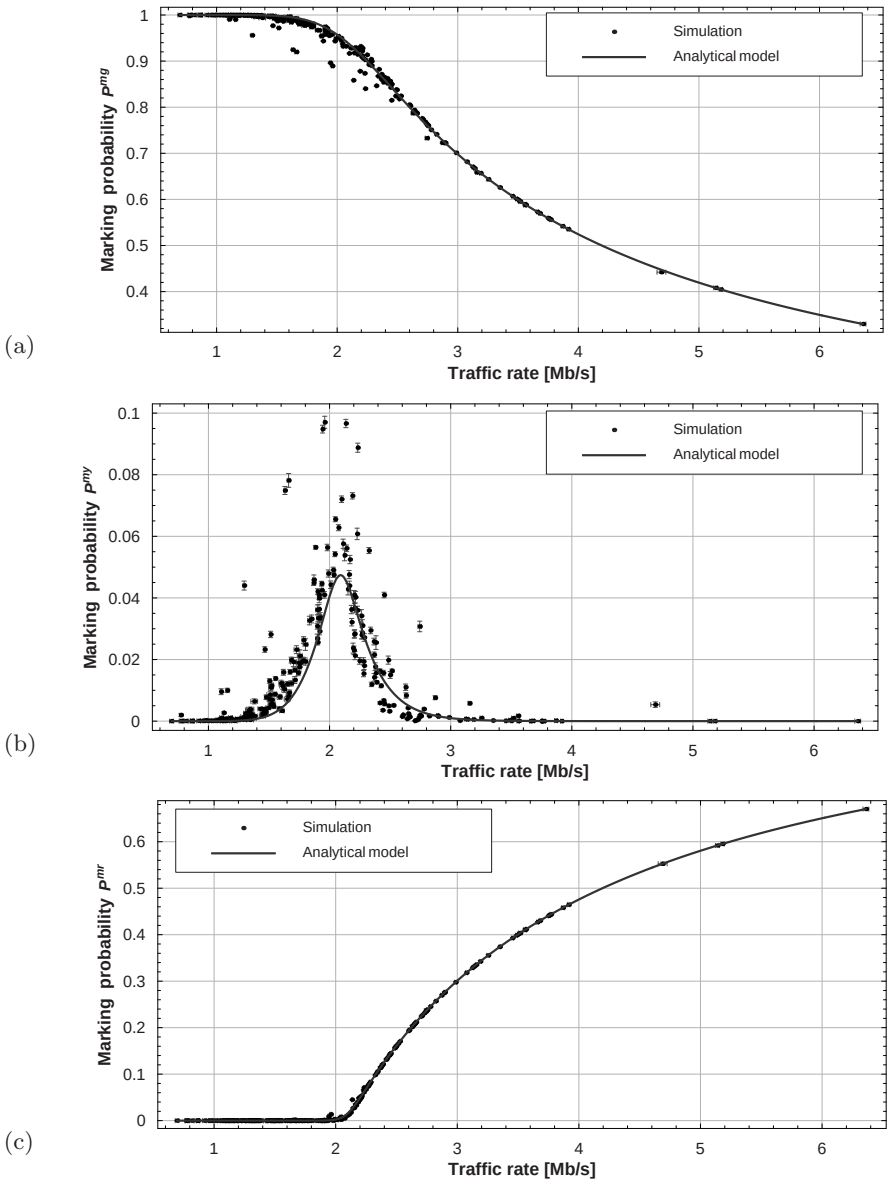


Figure 5.3: *Experiment 1*: srTCM, probability of marking packet as green (a), yellow (b) and red (c) as a function of throughput, ftp-like traffic, the linear scale

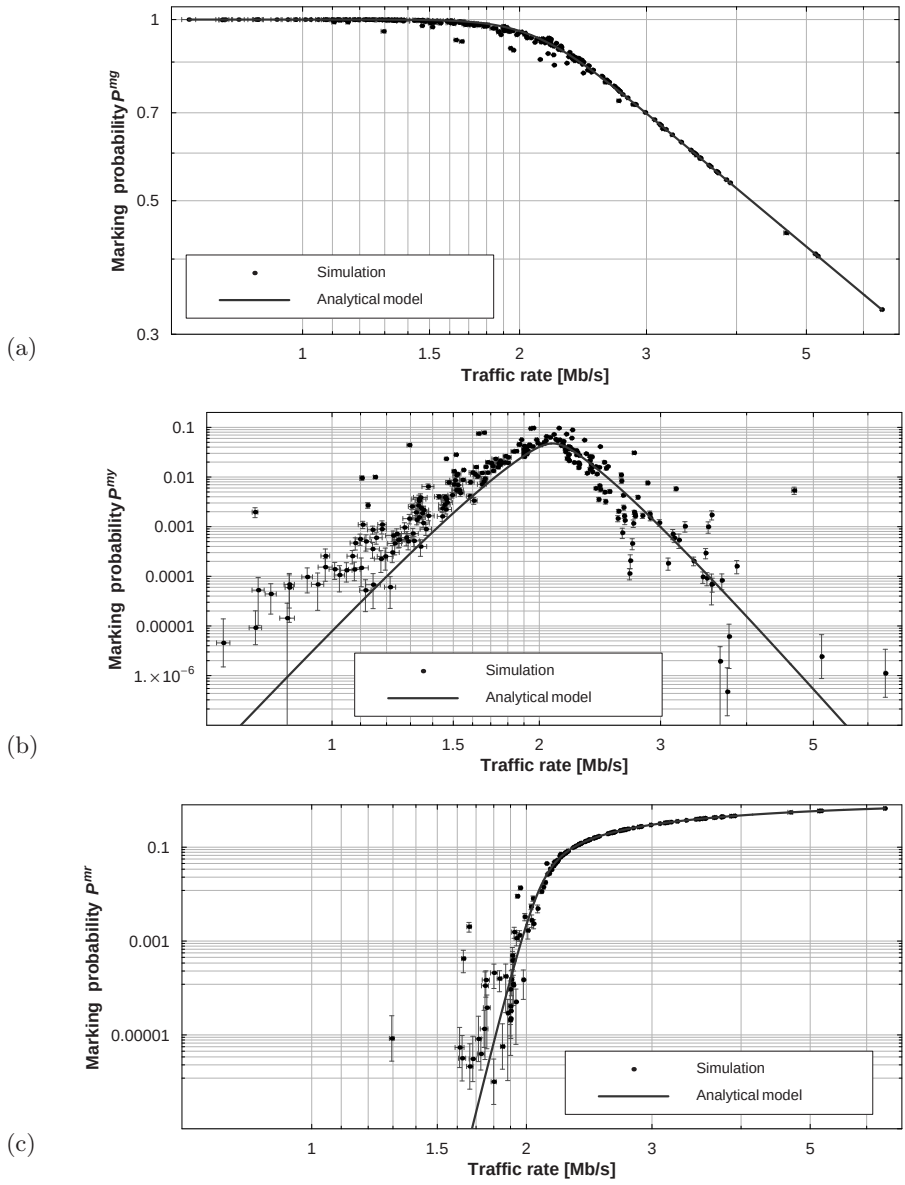


Figure 5.4: *Experiment 1*: srTCM, probability of marking packet as green (a), yellow (b) and red (c) as a function of throughput, ftp-like traffic, the logarithmic scale

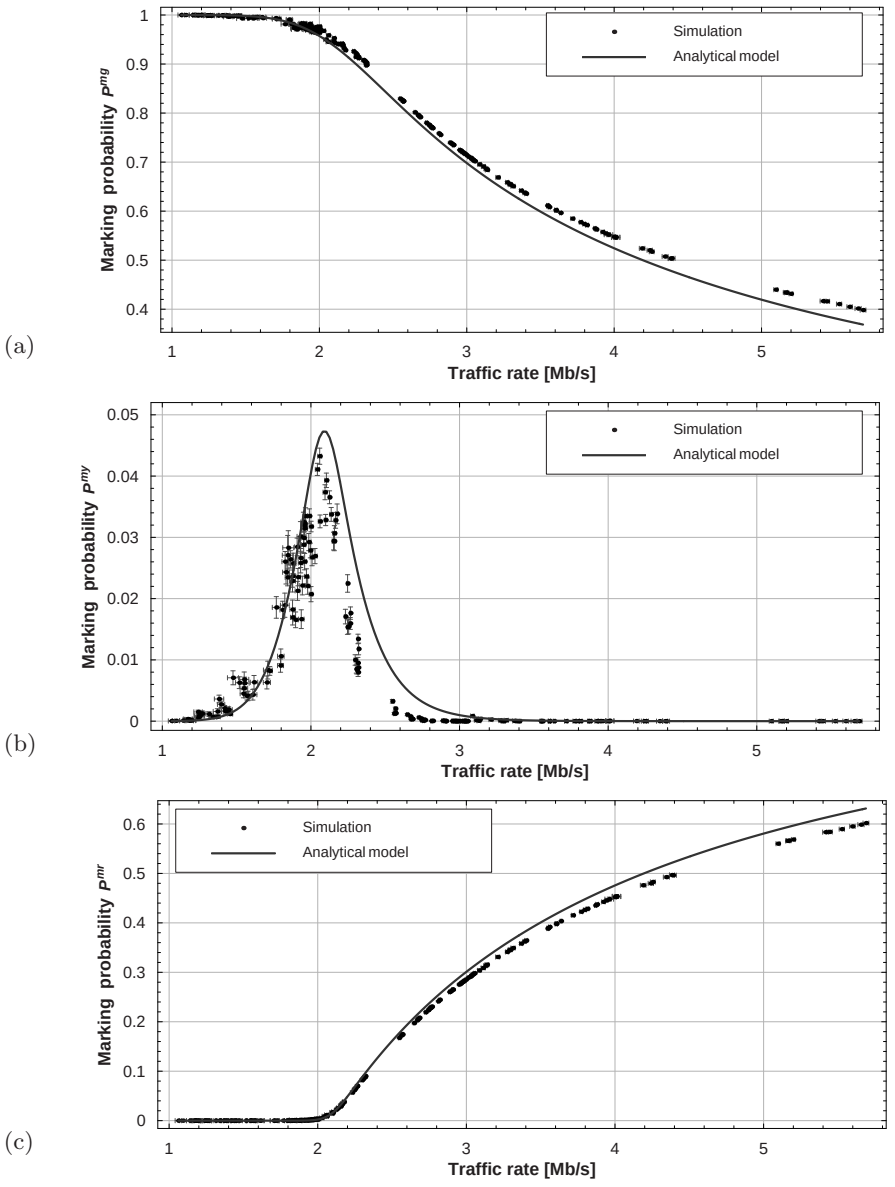


Figure 5.5: *Experiment 1*: srTCM, probability of marking packet as green (a), yellow (b) and red (c) as a function of throughput, web-like traffic, the linear scale

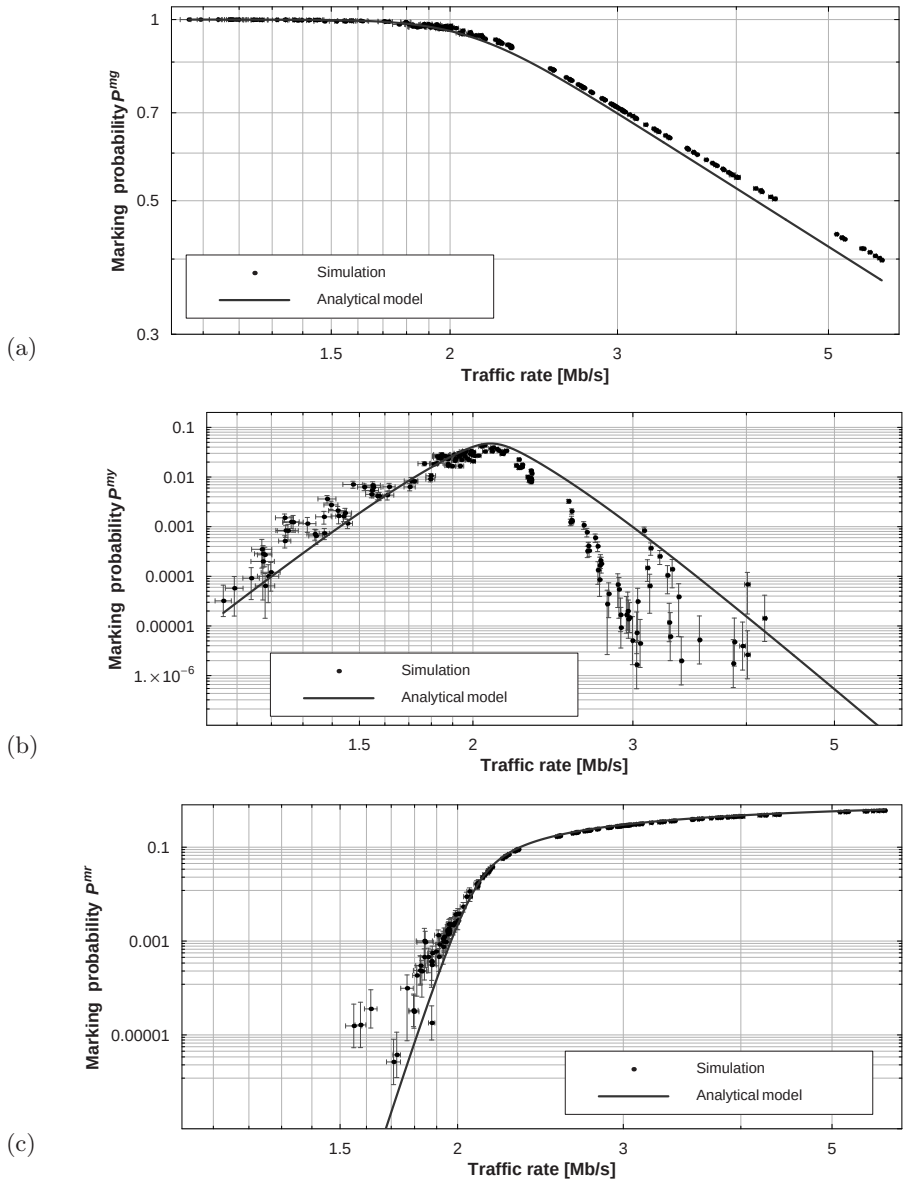


Figure 5.6: *Experiment 1*: srTCM, probability of marking packet as green (a), yellow (b) and red (c) as a function of throughput, web-like traffic, the logarithmic scale

around the analytical curve. It can be, however, assumed that for the traffic rate less than $0.8 \times CIR$ the value of P^{mr} is close to 0. For traffic rates greater than CIR the values of P^{mr} and P^{mg} obtained from simulation form a regular line for both types of traffic. These lines fit the analytical curves very well for the ftp-like traffic. For the web-like traffic P^{mg} becomes increasingly underestimated while P^{mr} becomes increasingly overestimated if the traffic rate increases above CIR. However, this discrepancy is not high. It does not exceed 5% for the traffic rate of about $3 \times CIR$.

The most visible discrepancies are for the probability of packet marking as yellow, P^{my}. However, they are relatively small for the traffic rate around CIR and become greater while the distance from CIR increases. It must be stressed that values of P^{my} obtained from simulation in those areas are very small and do not form a regular line but are considerably spread. Thus, providing an exact formula describing P^{my} is not possible. Moreover, inexactness of the model for such small values of the marking probability is not very critical. Again, it can be assumed that for the traffic rate $< 0.8\,CIR$ the value of P^{my} is ≈ 0. Values of marking probabilities for the traffic rate of about CIR and greater are more important. In that area, P^{my} is predicted quite well around CIR but for a greater traffic volume it again becomes spread but close to zero.

5.2.2 Two rate three color marker

Experiment 2 [142] shows a comparison between results obtained from simulation and the analytical model for a sample configuration of trTCM.

Experiment 2

The parameters of trTCM in this experiment were set as follows: $CIR = 1.5\,\text{Mb/s}$, $PIR = 2\,\text{Mb/s}$, $CBS = 5\,\text{kB}$, $PBS = 20\,\text{kB}$. The packet size was set to 1 kB. Packet marking probabilities as a function of the traffic rate for the ftp-like traffic are shown in Fig. 5.7 (the linear scale) and Fig. 5.8 (the logarithmic scale). Characteristics for trTCM in the case of the web-like traffic in the linear and logarithmic scales are shown in Fig. 5.9 and Fig. 5.10, respectively. Solid lines represent characteristics predicted by the model based on D/M/1/K (formulas 4.17 — 4.19). Dashed lines are used for the M/M/1/K model (formulas 4.11 — 4.13). Points show results obtained from simulation.

The meter/marker behaves slightly differently for the two types of traffic. In the case of the web-like traffic the D/M/1/K model fits the simulation results clearly better than the M/M/1/K model in the whole range of throughput (Fig. 5.9 and Fig. 5.10). Characteristics presented in the linear scale (Fig. 5.9) show some discrepancies for the traffic rate greater than PIR: P^{my} is slightly

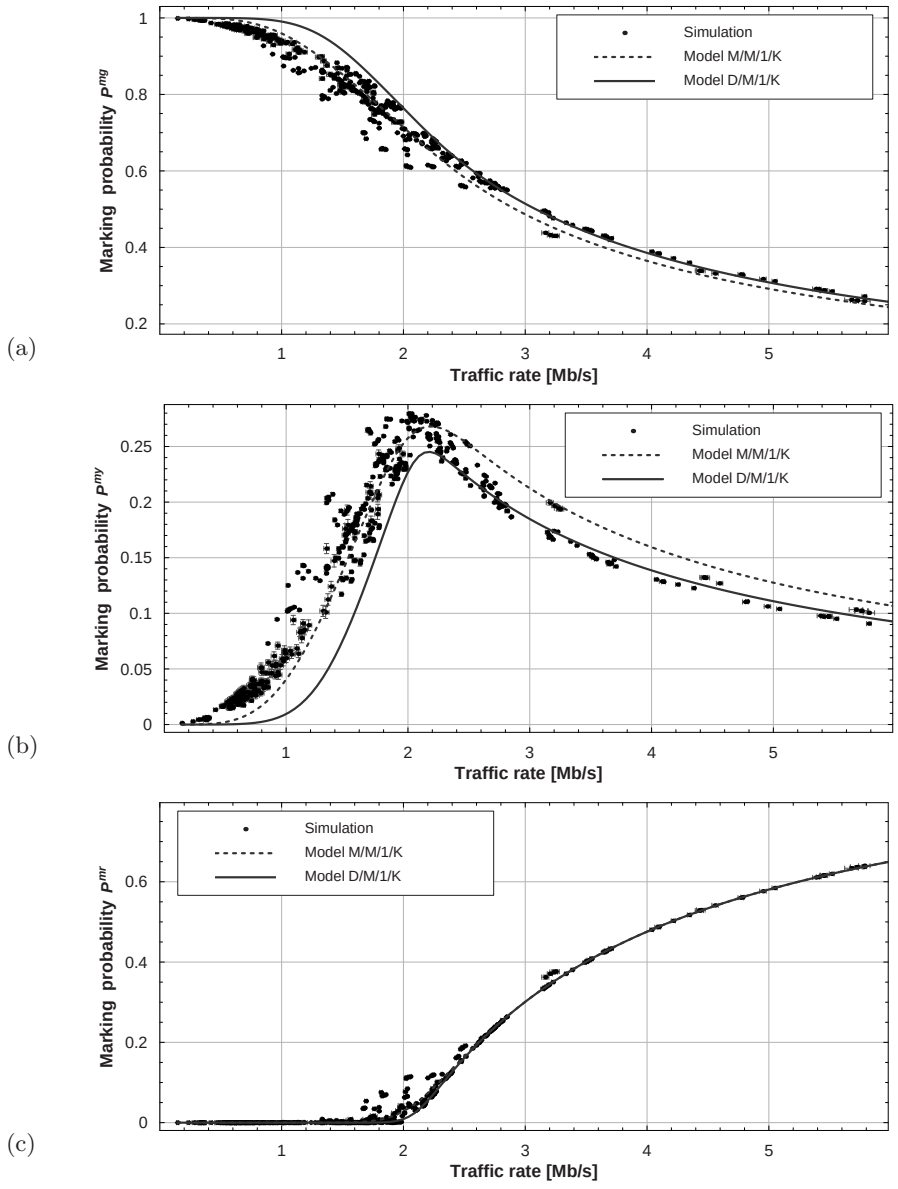


Figure 5.7: *Experiment 2*: trTCM, probability of marking packet as green (a), yellow (b) and red (c) as a function of throughput, ftp-like traffic, the linear scale

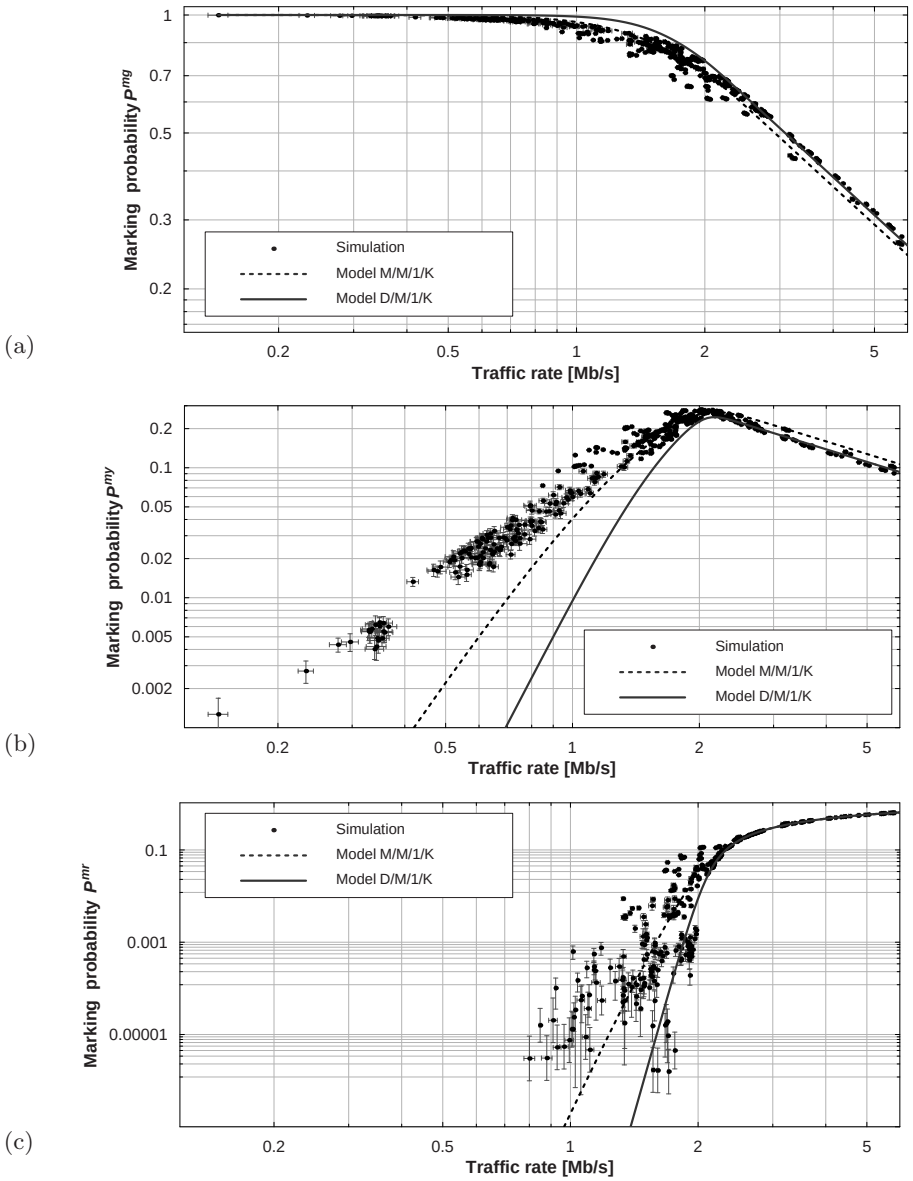


Figure 5.8: *Experiment 2*: trTCM, probability of marking packet as green (a), yellow (b) and red (c) as a function of throughput, ftp-like traffic, the logarithmic scale

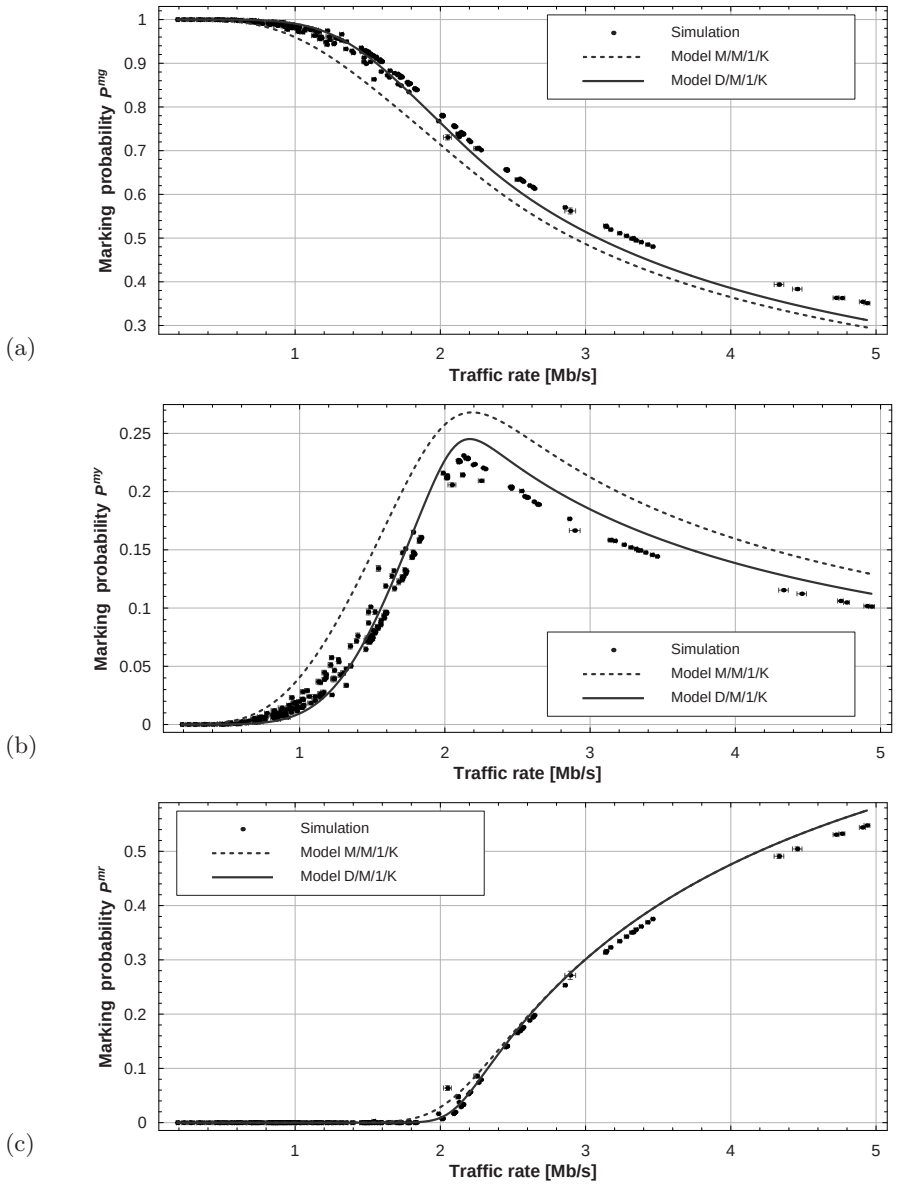


Figure 5.9: *Experiment 2*: trTCM, probability of marking packet as green (a), yellow (b) and red (c) as a function of throughput, web-like traffic, the linear scale

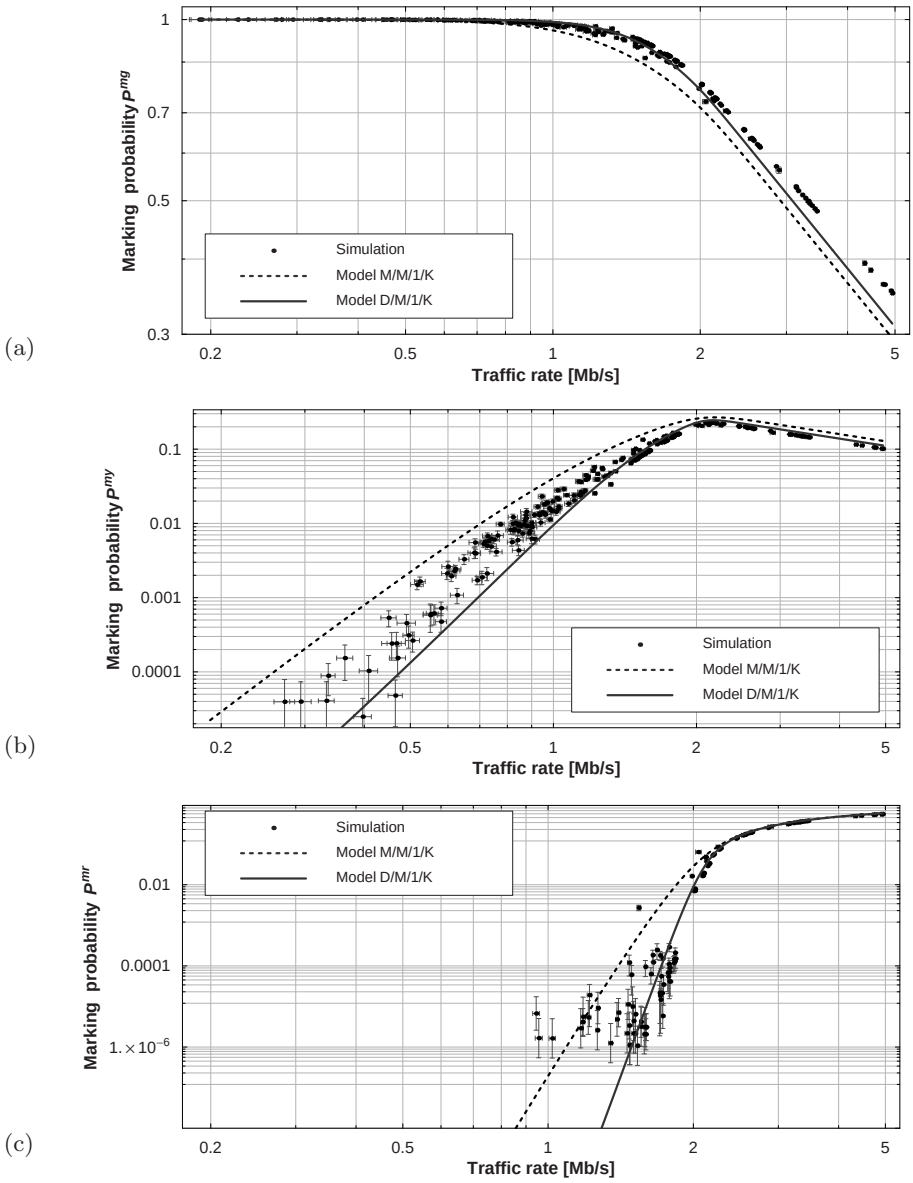


Figure 5.10: *Experiment 2*: trTCM, probability of marking packet as green (a), yellow (b) and red (c) as a function of throughput, web-like traffic, the logarithmic scale

overestimated while P^{mg} is a little bit underestimated. In turn, P^{mr} becomes increasingly overestimated by the model for the throughput exceeding $\approx 1.5 \times PIR$ but the difference is not big. The logarithmic scale (Fig. 5.10) reveals that values of the packet marking probability obtained from simulations are spread for traffic rate $< CIR$ in the case of P^{my} and for traffic rate $< PIR$ for P^{mr}. The points lie between M/M/1/K and D/M/1/K analytical curves. It can be observed, however, that the latter curve is a better approximation of the experimental data. Similarly to the srTCM meter/marker, although the experimental data are spread and do not perfectly fit any analytical curve, it does not disqualify the model. Firstly, values of the marking probability in that area are very small, namely from 10^{-6} to ≈ 0.04. Secondly, such a spread and unpredictable marking probability is a feature of the meter/marker.

In the case of the ftp-like traffic the choice of the analytical model is not straightforward. Generally, if the throughput is below PIR the M/M/1/K based analytical curve fits experimental data better while the D/M/1/K model appears to be better otherwise. Such a conclusion can be derived by scrutinizing characteristics of the packet marking probability as green and yellow, especially those in the linear scale (Fig. 5.7 (a) and (b)). Characteristics in both scales reveal spreading of experimental data for the traffic rate about and below PIR. Points are spread around the M/M/1/K analytical curve for the traffic rate between CIR and PIR. For smaller traffic rates values of the packet marking probability as yellow obtained from simulation are also spread but it can be said that they form a 'thick line' (Fig. 5.8 (b)). P^{my} becomes increasingly underestimated by the M/M/1/K model in that area. The probability that the packet is marked red is very well predicted by both analytical models for the throughput above $1.2 \times PIR$ (Fig. 5.7 (c)). The M/M/1/K and D/M/1/K curves converge in that area and fit the experimental data. For the traffic rate around and below PIR values of P^{mr} become considerably spread (Fig. 5.8 (c)) but the marking probability quickly becomes very small when the traffic rate decreases. It is < 0.01 for the traffic rate below CIR. The choice on which of the two analytical models gives a better approximation is ambiguous.

Summarizing conclusions for experiments with trTCM and ftp-like traffic it can be said that, in general, the M/M/1/K model better approximates the experimental marking probability below PIR while the D/M/1/K model is better for the traffic rate above PIR.

5.2.3 Time sliding window three color marker

Experiment 3 shows a comparison between results obtained from simulation and the analytical model for a sample configuration of TSW3CM.

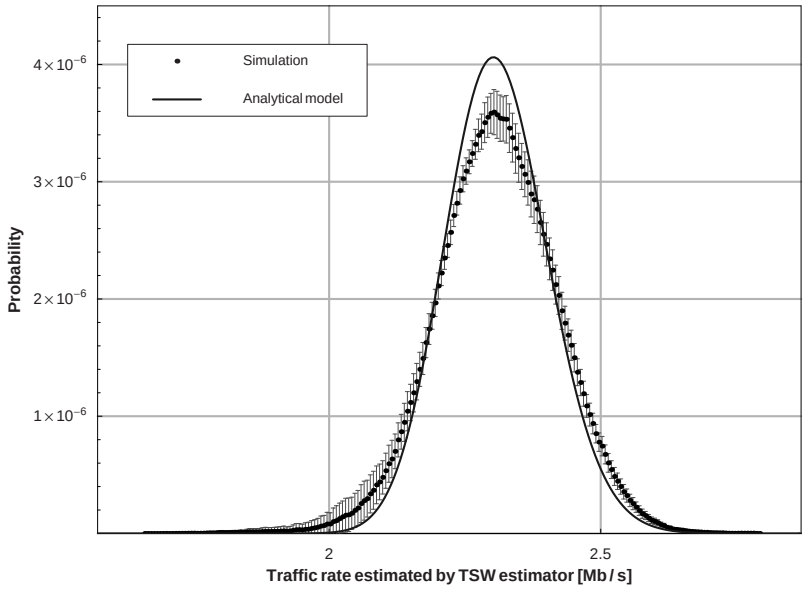


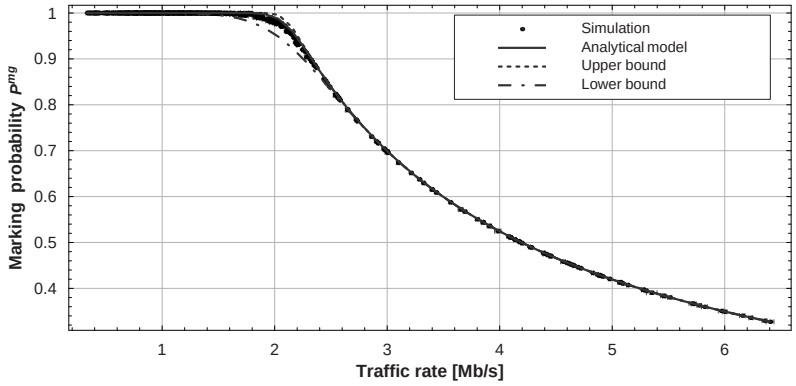
Figure 5.11: *Experiment 3*: PDF of the TSW-averaged traffic rate

Experiment 3

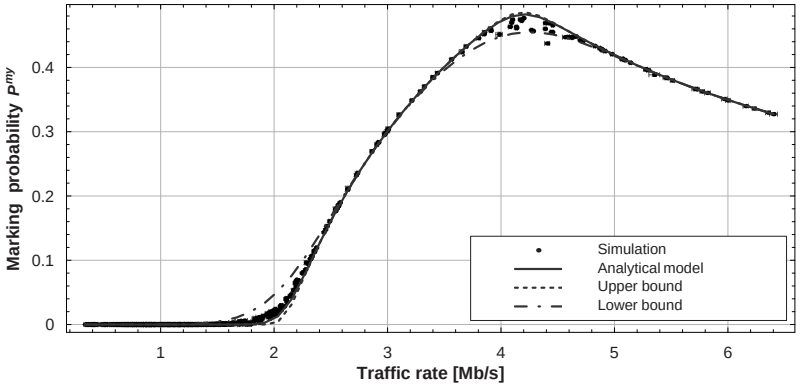
The TSW3CM parameters used in the experiment were as follows: $CIR = 2\,\text{Mb/s}$, $PIR = 4\,\text{Mb/s}$, $W = 1\,\text{s}$. Packet size used in simulations with the ftp-like traffic was 1 kB and 500 B in simulations with the web-like traffic. Fig. 5.11 shows an example of the probability density function of the TSW-averaged traffic rate A (estimated by the TSW rate estimator). Packet marking probabilities as a function of a traffic rate for the ftp-like traffic are shown in Fig. 5.12 (the linear scale) and Fig. 5.13 (the logarithmic scale). Characteristics for TSW3CM in the case of web-like traffic in the linear and logarithmic scales are shown in Fig. 5.14 and Fig. 5.15, respectively. Characteristics predicted by the model (formulas 4.28 — 4.30) are plotted with solid lines. Boundary characteristics are also presented: dashed line and dash-dot line represent upper and lower bound characteristics, respectively. Points show results obtained from simulation.

As shown in Fig. 5.11 the PDF of the TSW-averaged traffic rate (formula 4.27) is well predicted by the model. This characteristic was obtained from an experiment with the web-like traffic. The average traffic rate is 2.3 Mb/s. Looking at the shape of PDF it may seem that other, simpler probability distributions, such as normal or gamma, could be used instead of the proposed one. However, the

(a)



(b)



(c)

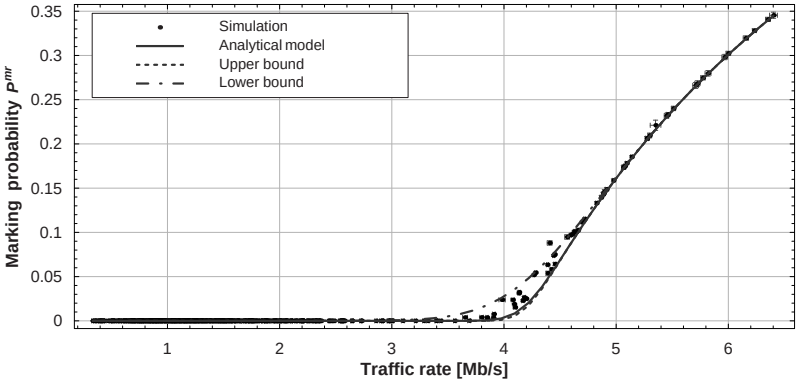


Figure 5.12: *Experiment 3*: TSW3CM, probability of marking packet as green (a), yellow (b) and red (c) as a function of throughput, ftp-like traffic, the linear scale

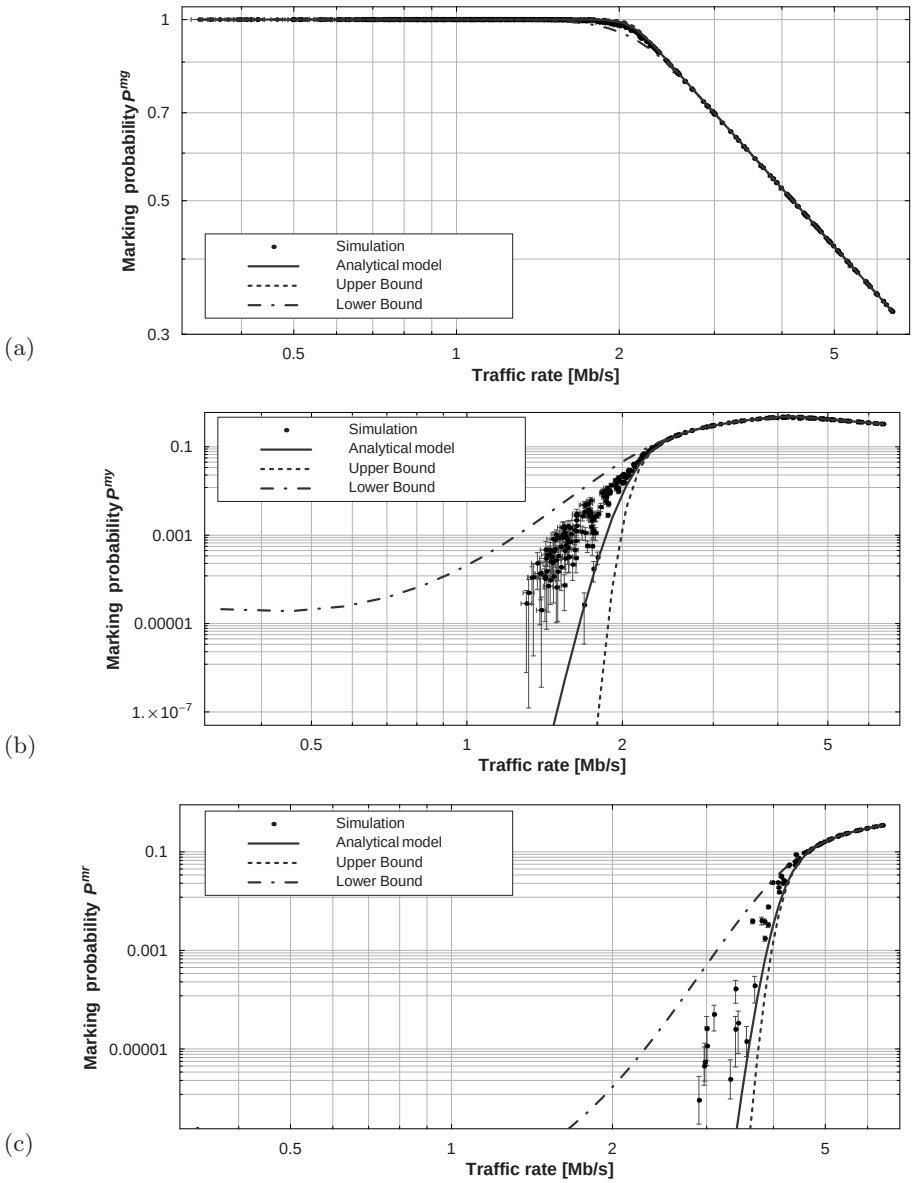
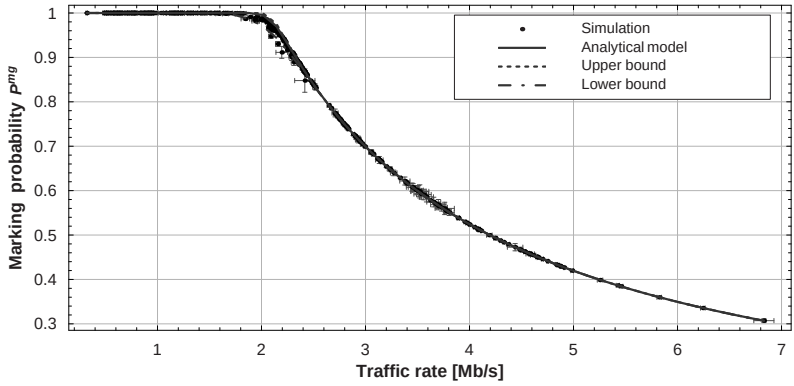
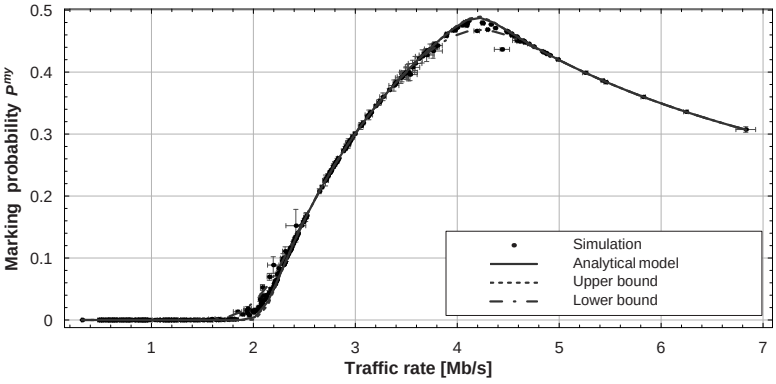


Figure 5.13: *Experiment 3*: TSW3CM, probability of marking packet as green (a), yellow (b) and red (c) as a function of throughput, ftp-like traffic, the logarithmic scale

(a)



(b)



(c)

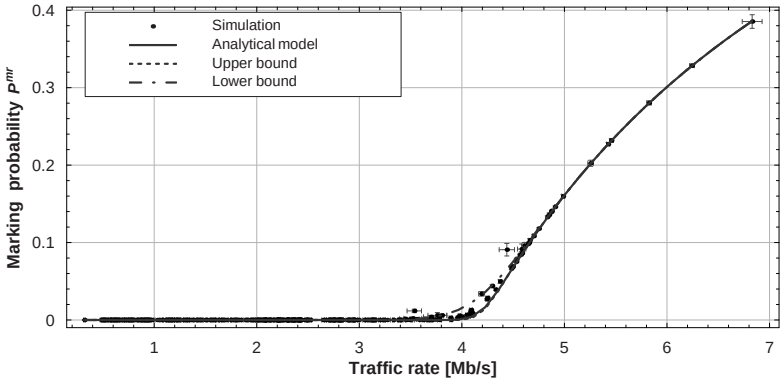


Figure 5.14: *Experiment 3*: TSW3CM, probability of marking packet as green (a), yellow (b) and red (c) as a function of throughput, web-like traffic, the linear scale

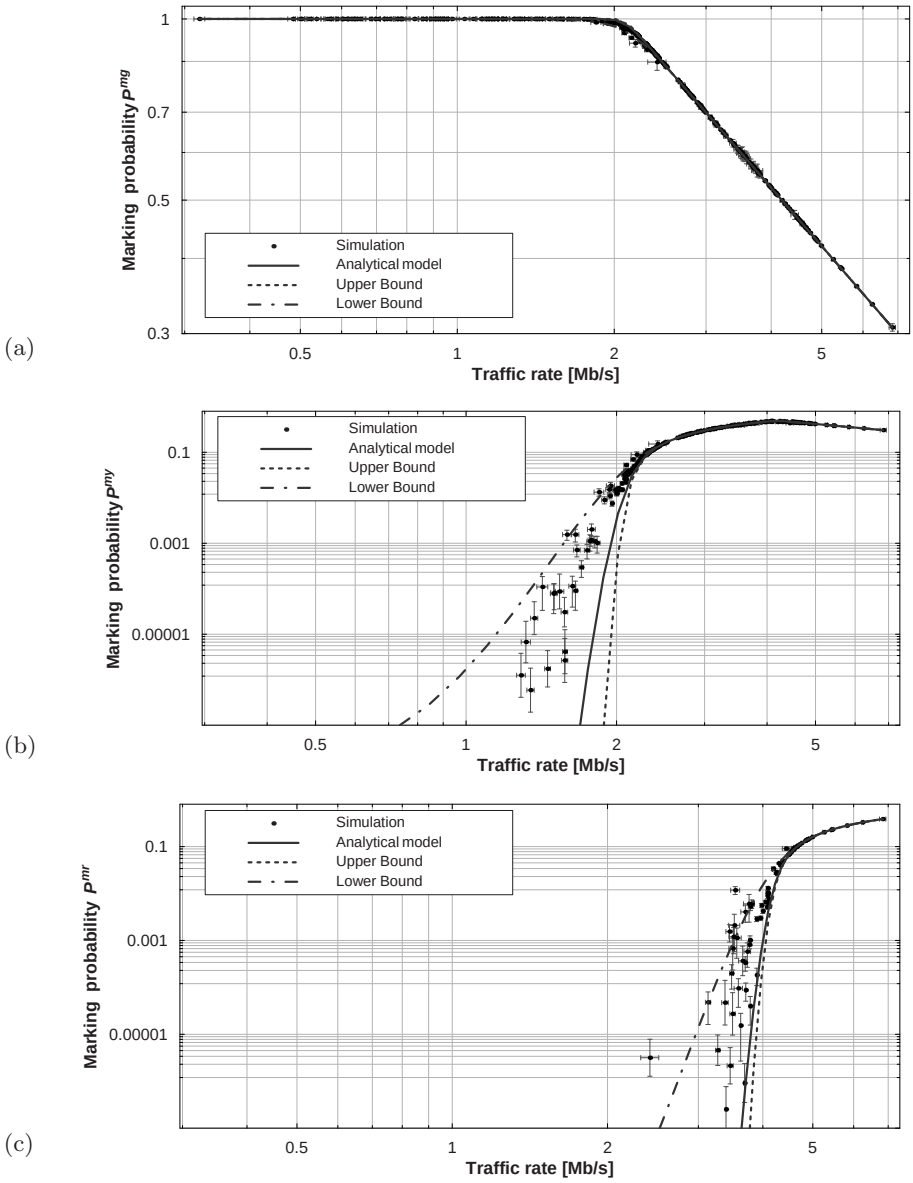


Figure 5.15: *Experiment 3*: TSW3CM, probability of marking packet as green (a), yellow (b) and red (c) as a function of throughput, web-like traffic, the logarithmic scale

relation between distribution parameters and rate estimator parameters would be very difficult to find. Moreover, normal distribution, if any, would work only for the ratio of window size to inter-arrival time greater than ≈ 50.

Pearson's chi-square tests were used to verify the proposed PDF of the TSW-averaged traffic rate. In some cases there was no evidence to reject the hypothesis that the proposed PDF agrees with the experimental data. In some cases, however, such a hypothesis was rejected. Nevertheless, the PDF itself is not the object of interest from the TSW3CM modeling perspective. It should be accurate enough to predict marking probabilities which are obtained by integrating the PDF. That is why some inaccuracies, even causing the Pearson's chi-square test to fail, are acceptable.

Analytical marking probability curves were obtained from formulas (4.28) — (4.28). Characteristics plotted in a the linear scale (Fig. 5.12 and Fig. 5.14) show a very good fit of the model and experimental data, for both types of traffic. Some small spreading of simulation data can be noticed only in close surroundings of CIR and PIR but most of the measured marking probabilities hold between the lower and the upper bound obtained for $n^{\perp}$ (4.36) and $n^{\top}$ (4.31), respectively. As revealed by the logarithmic scale (Fig. 5.13 and Fig. 5.15), if the marking probability becomes less than 0.01 the experimental data becomes spread. Most of results still lie between analytically determined bounds.

It is natural that for a higher throughput the accuracy of the model increases. The higher throughput is, the smaller inter-arrival time and, what follows, the value of n is greater. Then the averaging property of the TSW rate estimator increases and is less vulnerable to momentary changes of the traffic rate.

A general conclusion, after analyzing also results of several experiments beyond the presented one, is as follows. If the window size W is large in comparison to the mean inter-arrival time and, what follows, the value of n is in order of dozens or at least greater than 50, the points obtained form the simulation are not spread around the analytical curve but they match the curve very well. A small variation of the marking probability stems from the small variation of the TSW-averaged traffic rate estimated by the TSW algorithm which, in turn, is explained by a long window. On the other hand, if the window size is small, the TSW algorithm is more sensitive to temporary changes in the real traffic rate and the variation of the traffic rate estimates is greater. Thus, values of the marking probability obtained form simulations are more spread but still most of them hold in the proposed bounds, that in turn are wider for small windows. The increased spread of experimental data is most visible for traffic rates below PIR and decreases when the traffic rate increases. Also, a proposed PDF for the TSW-averaged traffic rate estimated by the TSW algorithm is farther from reality if the window size is small in comparison to the inter-arrival time. An example of the experiment with window size 0.1 s is presented in [141].

5.3 Experiments: droppers

Presentation of experimental validation of models of Multi-RED is more complex than in the case of meter/markers. The problem is that there are four variables that impact the characteristics, namely the throughput and the percentage of red, yellow and green packets in the traffic stream, that is the traffic structure. Changing of only one of these variables changes all characteristics. Thus many experiments were performed to check models' accuracy under various input data. Clear presentation of Multi-RED characteristics as a function of all inputs in one plot is not possible. It was decided to present the results in the most natural way. The tests were performed for the network consisting of a Multi-RED queue as well as one or more meter/markers. The overall throughput was changed from values below the bottleneck capacity, usually 0.4 — $0.7 \times \mu$, to approximately 1.1 — $1.2 \times \mu$. The traffic color structure was changing accordingly to the meter/markers' configuration and traffic volume. Therefore, the traffic structure may differ between points in a single plot.

A number of experiments were performed to validate the analytical models for Multi-RED queues. This section presents results of four sample experiments in total. Two types of queues: WRED and RIO-C were tested. Each type of queue was examined in both the staggered and overlapped configuration of thresholds (Section 2.3.2). Each configuration was tested under two types of traffic. Experimental results are compared with the M(n)/M/1/K and M(n)/D/1/K models. Confidence intervals were calculated but are usually too small to be visible at the scale of plots. To make the comparison of results for different bottleneck capacities easier, the ratio ρ of the throughput to the bottleneck capacity is used on the x-axis of plots instead of the throughput.

The most important characteristics are the overall packet drop probability, $\bar{P}^d$, and, closely related, goodput B^{out}. Appropriate prediction of them is most desired and useful for network designers. The average queue size, $\bar{q}$ is also interesting since another important QoS parameter — a queuing delay — can be calculated from it. The usability of per color drop probabilities, i.e., $\bar{P}^{dr}$, $\bar{P}^{dy}$ and $\bar{P}^{dg}$, is slightly lower but these characteristics may be also taken into account while examining or planning the network configuration and behavior.

Summary conclusions and recommendations are provided at the end of this section.

5.3.1 WRED

The following two experiments present results for the WRED type of Multi-RED queue with staggered and overlapped thresholds configuration.

Table 5.2: *Experiment 4*: WRED queue configuration

Q_{max}	60	packets
$minTh_r$	5	packets
$maxTh_r$	10	packets
$minTh_y$	10	packets
$maxTh_y$	20	packets
$minTh_g$	20	packets
$maxTh_g$	40	packets
$Pmax_r$	0.2	
$Pmax_y$	0.1	
$Pmax_g$	0.02	

Experiment 4

The configuration parameters of the WRED queue with staggered way of thresholds configuration used in the experiment are juxtaposed in Tab. 5.2. The bottleneck capacity μ was 10 Mb/s. The packet size used in simulations with the ftp-like traffic was 1 kB. For experiments with the web-like traffic it was equal to 500 B. Six characteristics of the queue, including green, yellow and red packet drop probabilities, the overall packet drop probability, average queue size and goodput are presented in Fig. 5.16 and Fig. 5.17 for the ftp-like and web-like traffic, respectively.

In the case of the ftp-like traffic the accuracy of the M(n)/M/1/K and M(n)/D/1/K models is comparable. They both fit the experimental data very well, but the M(n)/D/1/K model is slightly better. It is most visible for values of ρ between 0.9 and 1.05. In this area curves predicted by M(n)/M/1/K model differs from the analytical clearly more than those approximated by M(n)/D/1/K model. The latter model is clearly better in the case of web-like traffic. It predicts the overall drop probability very well while the M(n)/M/1/K model overestimates it in the whole range of throughput. Also, the average queue size characteristic predicted by the M(n)/D/1/K model is better, even though it is overestimated for $\rho > \approx 1.05$. Both models considerably overestimates drop probability of green and yellow packets for $\rho > 1$ but the M(n)/D/1/K model still gives closer results. The discrepancy in the case of green packet drop probability is not critical since the drop probability takes small values, ≈ 0.

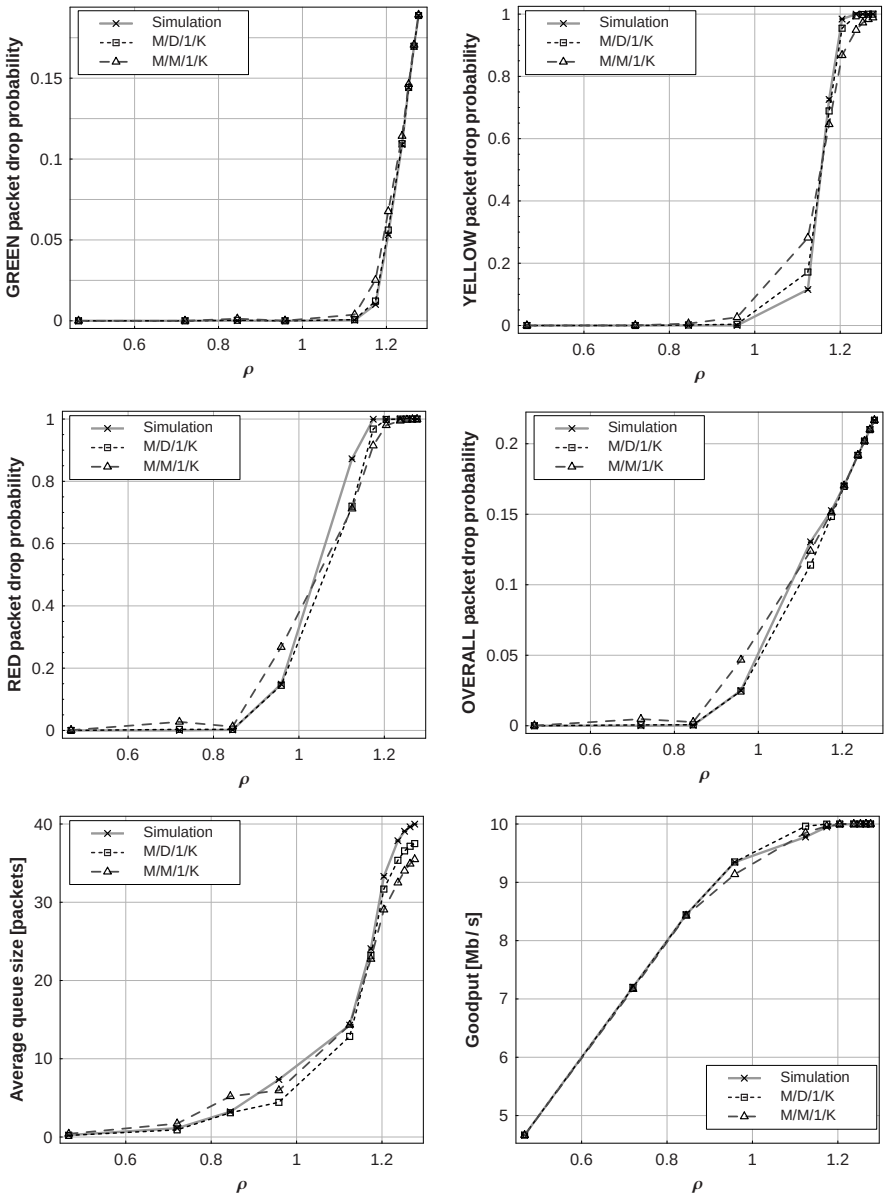


Figure 5.16: *Experiment 4*: Characteristics of the WRED queue with staggered configuration of thresholds, ftp-like traffic

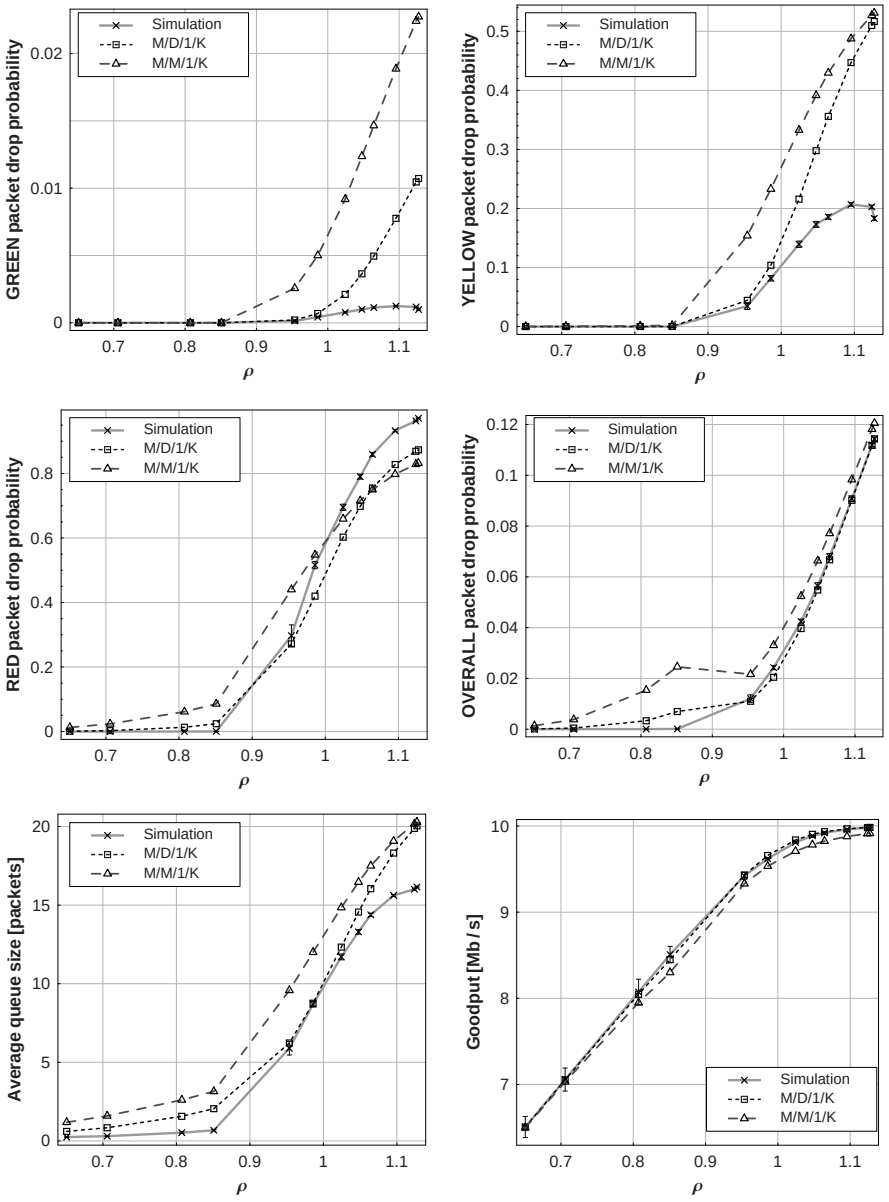


Figure 5.17: *Experiment 4*: Characteristics of the WRED queue with staggered configuration of thresholds, web-like traffic

Table 5.3: *Experiment 5*: WRED queue configuration

Q_{max}	60	packets
$minTh_r$	5	packets
$maxTh_r$	40	packets
$minTh_y$	5	packets
$maxTh_y$	40	packets
$minTh_g$	5	packets
$maxTh_g$	40	packets
$Pmax_r$	0.2	
$Pmax_y$	0.1	
$Pmax_g$	0.02	

Experiment 5

This experiment presents the verification of the model for the WRED queue with overlapped configuration of thresholds. The configuration parameters used in the experiment are juxtaposed in Tab. 5.3. The bottleneck capacity μ was 10 Mb/s. The packet size was set to 1 kB for both types of traffic. The queue characteristics for the ftp-like traffic are presented in Fig. 5.18 while those for the web-like traffic are shown in Fig. 5.19.

Simulations of the network with the WRED queue with the overlapped threshold configuration show that both analytical models give comparable results. The overall packet drop probability is very well predicted by the M(n)/D/1/K model. However, the deviation of the curve representing the overall packet drop probability approximated by the M(n)/M/1/K model is very small. The M(n)/D/1/K model performs better for per-color drop probability characteristics as well. In this case inaccuracies of the M(n)/M/1/K model are more visible. Conclusion regarding the characteristic of the average queue size is similar to the one regarding overall packet drop probability.

The situation is different for the web-like traffic. In that case the M(n)/M/1/K model gives a better approximation of experimental data, but the differences between the two models are not very high. Both models overestimated $\bar{P}^{dg}$ for $\rho > 1.05$ and underestimated $\bar{P}^{dr}$.

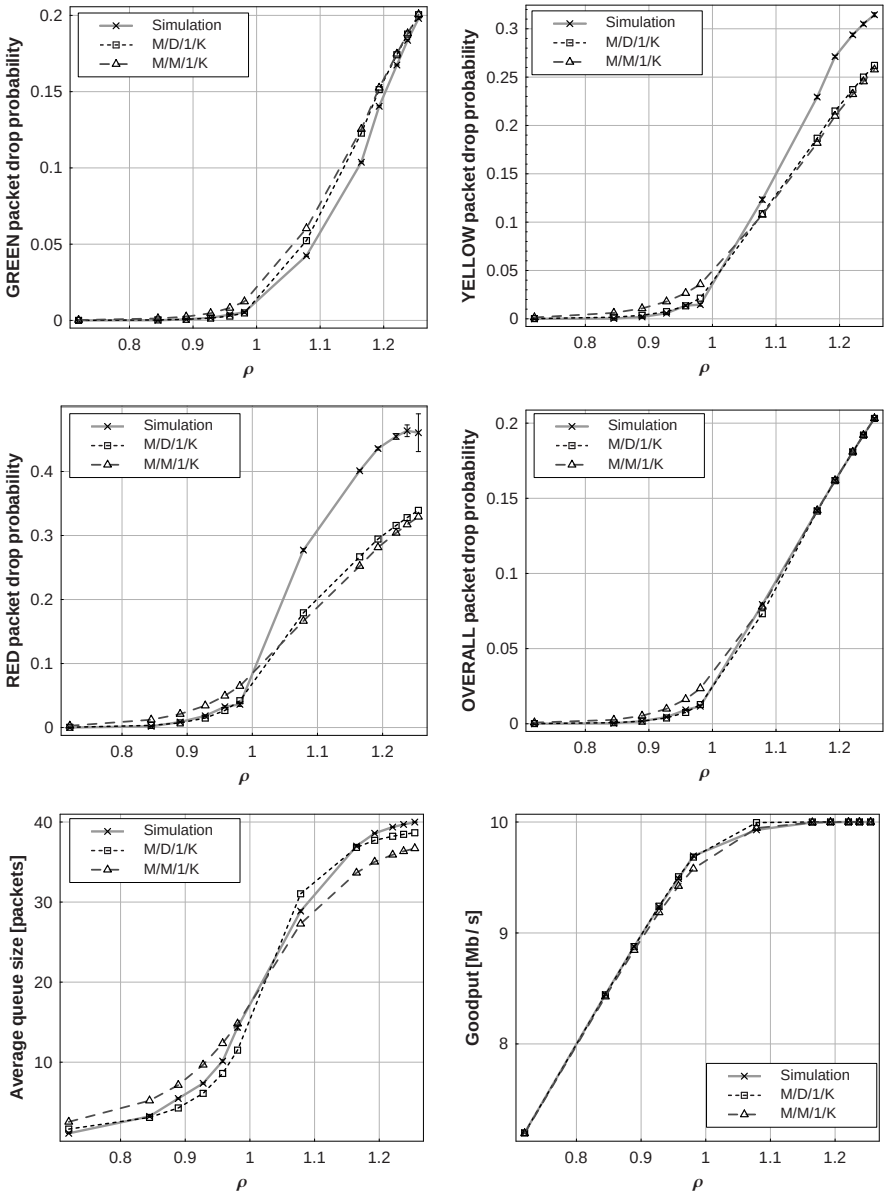


Figure 5.18: *Experiment 5*: Characteristics of the WRED queue with overlapped configuration of thresholds, ftp-like traffic

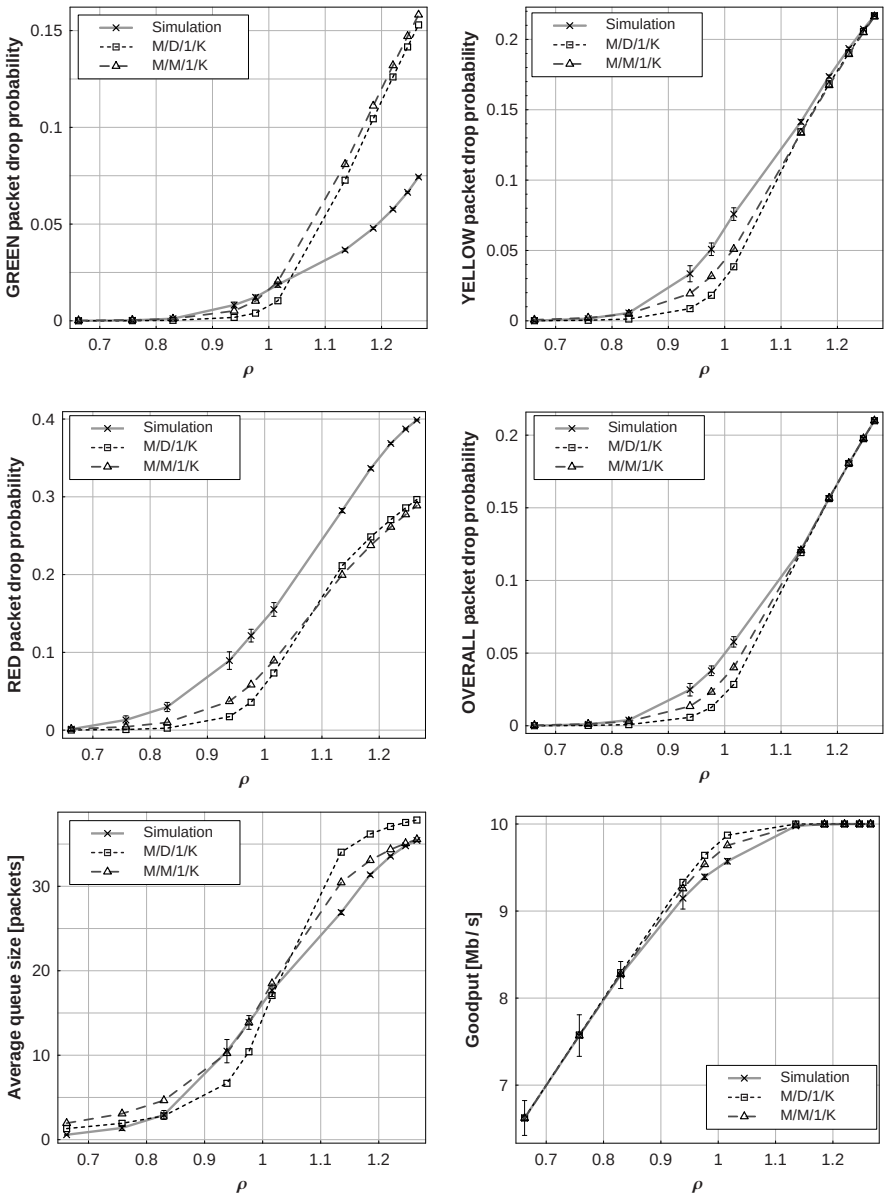


Figure 5.19: *Experiment 5*: Characteristics of the WRED queue with overlapped configuration of thresholds, web-like traffic

Table 5.4: *Experiment 6*: RIO-C queue configuration

Q_{max}	60	packets
$minTh_r$	5	packets
$maxTh_r$	10	packets
$minTh_y$	10	packets
$maxTh_y$	20	packets
$minTh_g$	20	packets
$maxTh_g$	40	packets
$Pmax_r$	0.2	
$Pmax_y$	0.1	
$Pmax_g$	0.02	

5.3.2 RIO-C

The following two experiments present results for the RIO-C type of Multi-RED queue with staggered and overlapped thresholds configuration.

Experiment 6

The RIO-C queue with staggered configuration of thresholds is verified by this experiment. The configuration parameters of the queue are presented in Tab. 5.4. The bottleneck capacity μ and packet size were set to 10 Mb/s and 1 kB, respectively, for the group of simulations with the ftp-like traffic. In simulations with the web-like traffic the bottleneck capacity was 7 Mb/s while the packet size was 500 B. The queue characteristics for the ftp-like traffic are presented in Fig. 5.20 while those for the web-like traffic are shown in Fig. 5.21.

Conclusions for that experiment are similar to the conclusions regarding the WRED queue with the staggered configuration of thresholds (experiment 4). The accuracy of characteristics approximation in the case of ftp-like traffic by the M(n)/M/1/K and M(n)/D/1/K models is similar but the latter outperforms the former in simulations with the web-like traffic. It must be emphasized that in the case of overall packet drop probability the M(n)/D/1/K analytical curve fits experimental data very well.

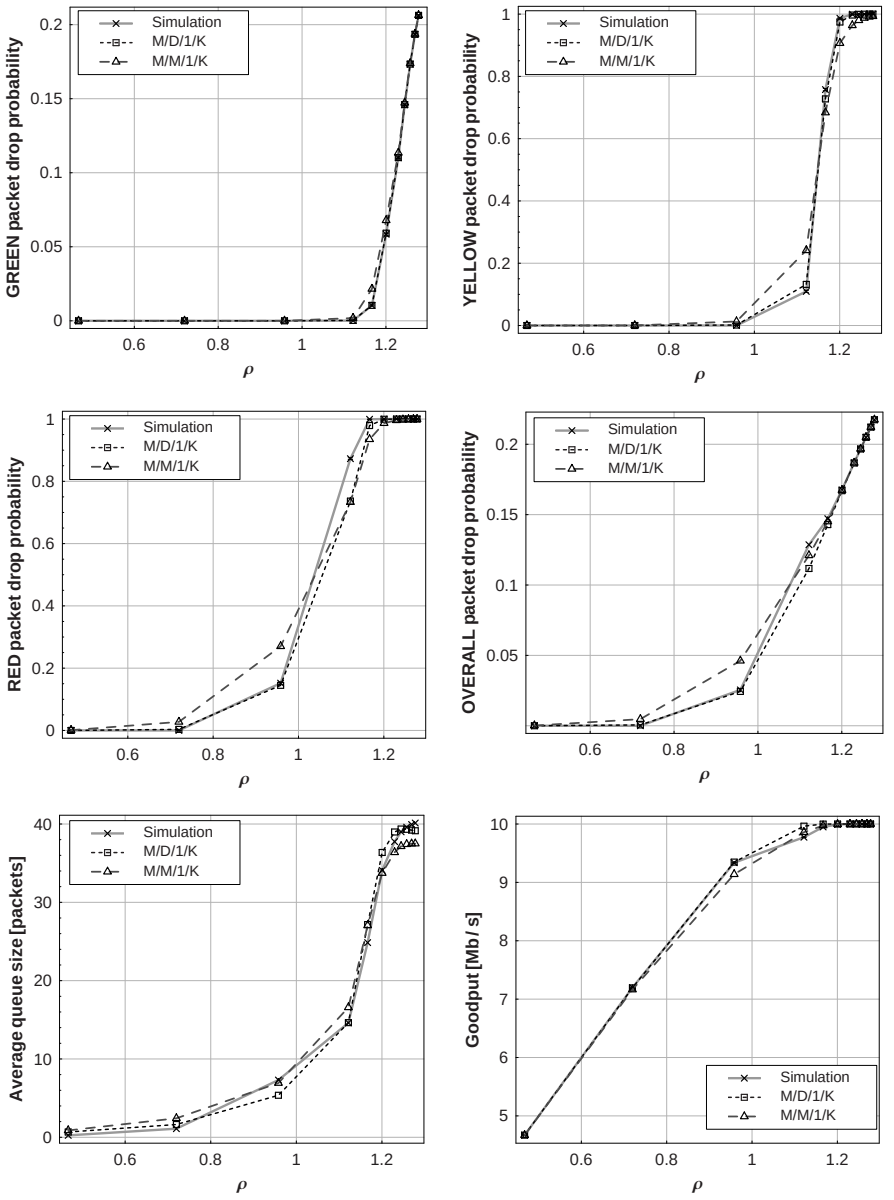


Figure 5.20: *Experiment 6*: Characteristics of the RIO-C queue with staggered configuration of thresholds, ftp-like traffic

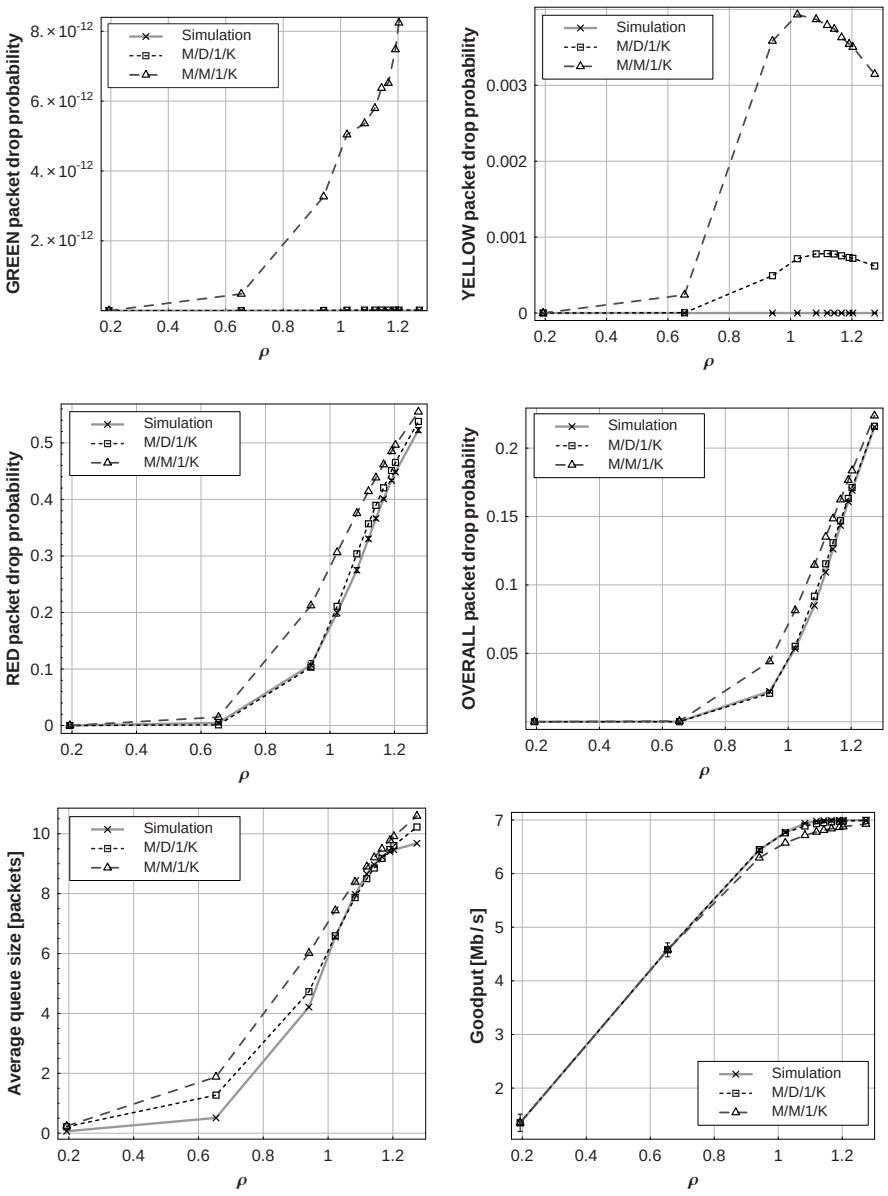


Figure 5.21: *Experiment 6*: Characteristics of the RIO-C queue with staggered configuration of thresholds, web-like traffic

Table 5.5: *Experiment 7*: RIO-C queue configuration

Q_{max}	60	packets
$minTh_r$	5	packets
$maxTh_r$	40	packets
$minTh_y$	5	packets
$maxTh_y$	40	packets
$minTh_g$	5	packets
$maxTh_g$	40	packets
$Pmax_r$	0.2	
$Pmax_y$	0.1	
$Pmax_g$	0.02	

Experiment 7

Table 5.5 presents the configuration parameters of the RIO-C queue with overlapped thresholds. Packet size was 1 kB in the case of the ftp-like traffic and 500 B in the case of web-like traffic. The bottleneck link capacity was 7 Mb/s. Experimental results and analytical curves are presented in Fig. 5.22 and Fig. 5.23 for the ftp-like and web-like traffic, respectively.

The accuracy of characteristics estimation by both analytical models is acceptable. The appearing discrepancies between experimental data and analytical curves are not disqualifying. Clearly, however, the M(n)/D/1/K model performs better in the case of ftp-like traffic. It offers a very precise approximation. The M(n)/M/1/K model is better in the case of web-like traffic.

5.3.3 Summary conclusions

In general, both M(n)/M/1/K and M(n)/D/1/K give comparable results and can be considered as well predicting Multi-RED queue characteristics. As presented in experiments 4 — 7 the better one can be indicated depending on the configuration and type of traffic. A comparison of the accuracy of the models is shown in Tab. 5.6. Letter "M" stands for the M(n)/M/1/K model while "D" means the M(n)/D/1/K model. Bold letters indicate which of the models performs better. Letters in brackets show the other model that gives only slightly worse results.

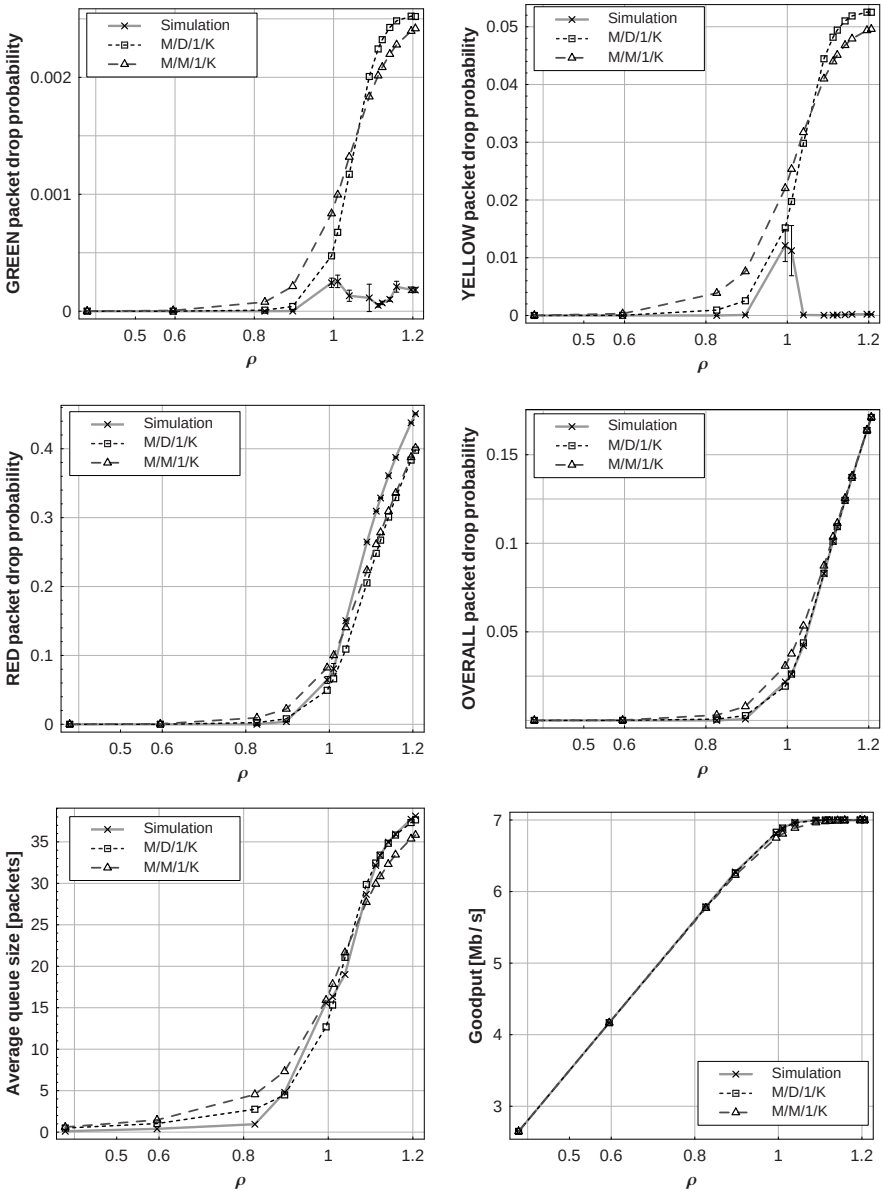


Figure 5.22: *Experiment 7*: Characteristics of the RIO-C queue with overlapped configuration of thresholds, ftp-like traffic

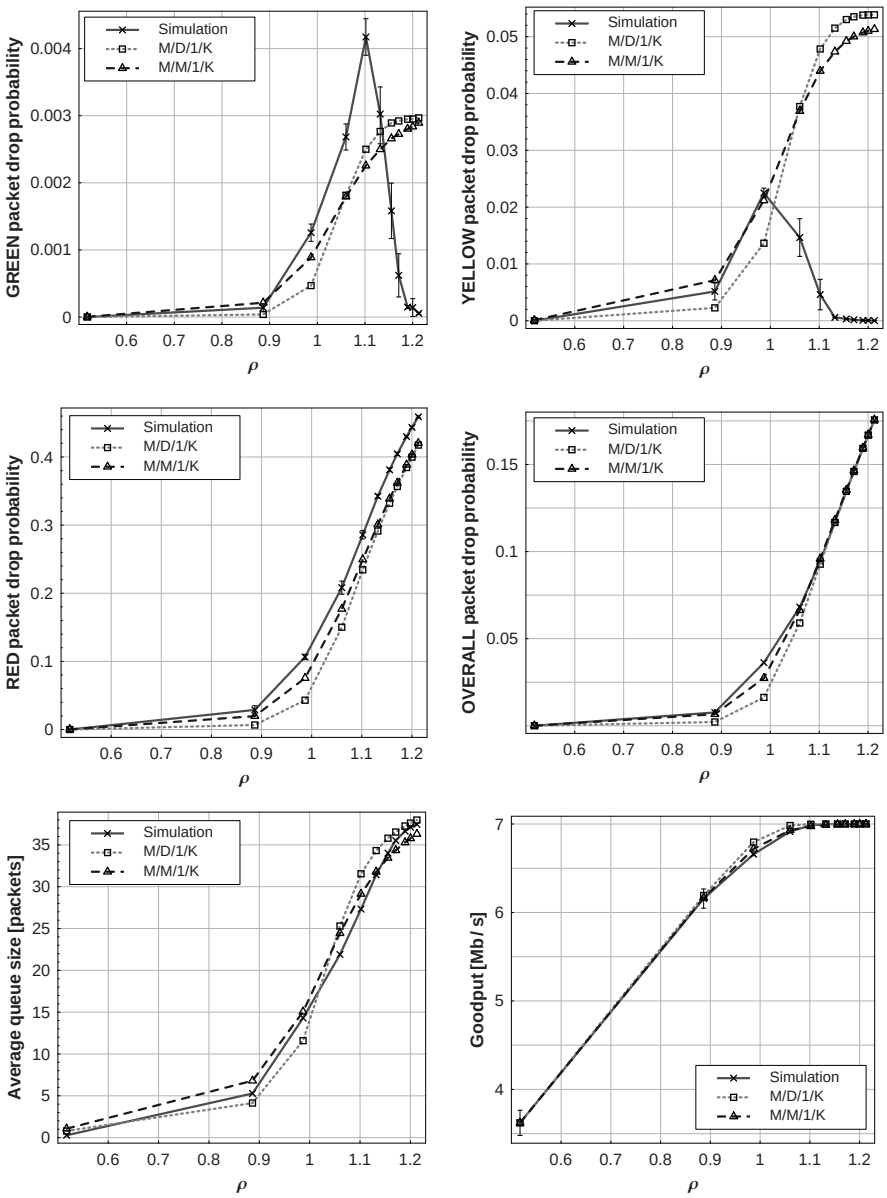


Figure 5.23: *Experiment 7*: Characteristics of the RIO-C queue with overlapped configuration of thresholds, web-like traffic

Table 5.6: Comparison of the performance of Multi-RED models under various configurations

Queue type	Thresholds configuration	Type of traffic	
		ftp-like	web-like
WRED	staggered	**D** (M)	**D**
WRED	overlapped	**D** (M)	**M** (D)
RIO-C	staggered	**D** (M)	**D**
RIO-C	overlapped	**D** (M)	**M**

The general recommendation is as follows. If the type of traffic is unknown, the M(n)/M/1/K model should be used for overlapped configurations of both WRED and RIO-C queues. The M(n)/D/1/K model is recommended for staggered configurations.

5.4 Experiments: modular model

Numerous network configurations were tested to verify the modular model. Presentation of all potential combinations of meter/marker, dropper (Multi-RED queue), thresholds configuration, range of propagation delays, and other parameters that can be configured, is not possible. Just taking into account three types of meter/marker, two types of Multi-RED queue and two ways of configuring thresholds in a Multi-RED queue raises twelve combinations. Moreover, the number of aggregates, the number of flows per aggregate, the bottleneck capacity, and the propagation delays can be selected. Meter/marker types and their configurations per-aggregates may also vary. Thus, results of only five representative experiments are presented here. The selected experiments encompass various configurations and reflect most common features of the modular model. The overview of the most distinguishing parameters used in experiments 8 — 12 are juxtaposed in Tab. 5.7.

Parameters of the simulated network are discussed in a greater detail for each experiment separately. The simulation results are compared with analytical characteristics evaluated by solving the system of equations presented in Section 4.5. In particular, marking probabilities of srTCM, trTCM and TSW3CM are calculated by using models described in Sections 4.2.1, 4.2.1, and 4.2.2, respectively. In the case of WRED and RIO-C packet drop probabilities were derived with the

Table 5.7: Overview of configuration used in experiments 8 — 12

Experiment number	Meter/ marker	Dropper (queue)	Thresholds configuration	Number of aggregates	Bottleneck capacity
8	srTCM	WRED	staggered	4	2 Mb/s
9	trTCM	WRED	staggered	3	5 Mb/s
10	TSW3CM	WRED	overlapped	2	5 Mb/s
11	trTCM	RIO-C	staggered	2	2 Mb/s
12	TSW3CM	RIO-C	overlapped	3	2 Mb/s

use of models described in Section 4.3. The variant based on the M(n)/D/1/K queuing system was used.

Selection of propagation delays resulting in differentiation of RTT needs a special attention. To make the following considerations more clear, propagation delay of link $S_{k,m}$—SR_k is denoted as $\delta_{k,m}$ while bounds for the uniform distribution from which $\delta_{k,m}$ is randomly chosen are denoted as δ_k^{min}—δ_k^{max}. If the propagation delay of other links is denoted as $\delta_{k,m}^{sum}$, the propagation delay of m-th flow of k-th aggregate introduced in formula (4.58) is expressed as $d_{k,m} = \delta_{k,m} + \delta_{k,m}^{sum}$.

As mentioned in Section 5.1.6, the propagation delays $\delta_{k,m}$ are randomly chosen from the uniform distribution of arbitrarily selected bounds. The number of TCP-FTP sources is M_k and is also selected arbitrarily for aggregate k. It means that M_k various random values of the propagation delay are assigned to links $S_{k,m}$—SR_k ($m = 1 \cdots M_k$).

In the analytical model, it is assumed that only the distribution of propagation delays and the number of TCP sources are known. Moreover, an approximation is made regarding the number of different values of the propagation delay per aggregate. It is assumed that there are z different values of propagation delays. These values are selected from the same range as in the corresponding simulation as follows. The smallest propagation delay is equal to $\delta_k^{min} + \left(\delta_k^{max} - \delta_k^{min}\right) / (2z)$. The subsequent $z-1$ values are obtained by adding step $\left(\delta_k^{max} - \delta_k^{min}\right) / z$. The number of flows experiencing the same propagation delay is then M_k/z. This method was applied to simplify equations and quicken numerical computation. In all experiments presented below, $z = 5$ was chosen. Higher values of z did not significantly improve the results.

In the set of equations describing the modular model, the throughput and goodput are calculated in packets per second and denoted as B and B^{out} (see

Section 4.5). The only exception is the model of TSW3CM were the throughput is expressed in bits per second and denoted as D. In presentation of experiments' results, the throughput and goodput are expressed in Mb/s because it is more natural. Nevertheless, the notation with B was kept to keep consistence with the notation used in the set of equations in Section 4.5. Similarly, in the analytical model meter/marker parameters such as CIR and PIR are expressed in packets per second and CBS, PBS, EBS are expressed in packets. In the presentation of the results more natural measures are used, namely Mb/s and bytes, respectively. Thus, if, for example, CBS is reported to be 10 kB and packet size used in the experiment is 500 B, the value of CBS used for calculations in the model is 20 packets.

Results of each experiment are summarized in three tables and one figure containing two graphs. Rows in the tables denoted by 'M' give parameter values predicted by the model while rows denoted by 'S' contain experimental data. Experimental results are provided with confidence intervals. In the case of figures, the legend explains which data belongs to experimental or analytical results.

One feature of the modular model, noticeable in many experiments, is that it has a tendency to overestimate the throughput of the aggregates. It stems from the fact that the Padhye formula overestimates the throughput of a flow experiencing smaller values of RTT. The discrepancy depends on the particular configuration but, in general, it is not very significant if RTT is greater than 50 — 100 ms. For RTTs less than 10 ms the discrepancy may be as high as 50%. However, the discrepancy in the Padhye model itself is compensated by models of other elements. The explenation of inexactness of Padhye formula is that it was developped for networks without DiffServ mechanism, that is, for a network in which packet drops occucres due to queue overflow. In a DiffServ network, packets are dropped randomly depending on their color. It may happen that a flow experiences a packet drop relatively earlier (in relation to evalution of the TCP congestion window size) than in the case of tail dropping. Therefore, for effectively the same packet drop probability, the throughput generated by a TCP source may by lower.

Experiment 8

In this experiment there were $k = 4$ traffic aggregates, each constituted by $M_k = 20$ flows generated by separate long-lived TCP source. Traffic aggregates were subject to measurement and packet marking at the network edge where srTCM meter/markers were used. Their configuration parameters were selected differently for each traffic aggregate.

Each flow experienced different propagation delay and, what follows, different RTT. As mentioned before, different propagation delays are achieved by varying the propagation delay $\delta_{k,m}$ of link $S_{k,m}$—SR_k the m-th flow of k-th aggregate

Table 5.8: *Experiment 8*: Configuration of srTCM and ranges of propagation delays

k	δ_k^{min}—δ_k^{max}	CIR_k	CBS_k	EBS_k
1	150—420 ms	1 Mb/s	10 kB	20 kB
2	100—400 ms	0.25 Mb/s	2 kB	10 kB
3	120—300 ms	0.5 Mb/s	5 kB	10 kB
4	200—600 ms	0.25 Mb/s	10 kB	20 kB

Table 5.9: *Experiment 8*: WRED queue configuration

Q_{max}	60	packets
$minTh_r$	5	packets
$maxTh_r$	10	packets
$minTh_y$	10	packets
$maxTh_y$	20	packets
$minTh_g$	20	packets
$maxTh_g$	40	packets
$Pmax_r$	0.2	
$Pmax_y$	0.1	
$Pmax_g$	0.01	

passes by. Values of $\delta_{k,m}$ were randomly chosen for each flow from the range arbitrarily selected for each aggregate. The configuration of meter/markers as well as ranges of the propagation delay for each aggregate are juxtaposed in Table 5.8. Packet size was set to 500 B.

The bottleneck capacity was 2 Mb/s. The WRED variant of the Multi-RED queue with staggered configuration of thresholds was used at the bottleneck router. Configuration parameters of the queue are shown in Tab. 5.9.

Results for the experiment are presented in Tables 5.10 — 5.12 and Figure 5.24. Table 5.10 presents general system characteristics such as the total throughput B, total goodput B^{out}, total average packet drop probability at the bottleneck router $\bar{P}^d$ and average queue size $\bar{q}$. It can be noticed that in this case

Table 5.10: *Experiment 8*: General characteristics of a system consisting of srTCM meter/markers and WRED queue

	B [Mb/s]	B^{out} [Mb/s]	$\bar{q}$ [packets]	$\bar{P}^d$
M	2.245	1.995	17.696	0.1114
S	2.141 ±0.00157	1.916 ±0.0012	17.571 ±0.0561	0.105 ±0.0003

Table 5.11: *Experiment 8*: General, per-color characteristics of a system consisting of srTCM meter/markers and WRED queue

	P^{qr}	P^{qy}	P^{qg}	$\bar{P}^{dr}$	$\bar{P}^{dy}$	$\bar{P}^{dg}$
M	0.1093	0.05123	0.8395	0.8257	0.3661	0.00287
S	0.08806 ±0.000519	0.07186 ±0.000277	0.8401 ±0.000488	0.9136 ±0.0028	0.3299 ±0.0032	0.001027 ±0.000031

Table 5.12: *Experiment 8*: Per-aggregate characteristics of a system consisting of srTCM meter/markers and WRED queue

k		B_k [Mb/s]	B_k^{out} [Mb/s]	$\bar{P}_k^d$	P_k^{mr}	P_k^{my}	P_k^{mg}
1	M	1.059	0.9981	0.05715	0.05537	0.02399	0.9206
	S	0.9947 ±0.00104	0.9425 ±0.00107	0.05243 ±0.000179	0.03099 ±0.000518	0.04974 ±0.000405	0.9193 ±0.000566
2	M	0.2902	0.2304	0.2060	0.1385	0.2456	0.6159
	S	0.283 ±0.000915	0.2271 ±0.000657	0.1976 ±0.000811	0.143 ±0.00212	0.2856 ±0.0016	0.5714 ±0.0012
3	M	0.5888	0.5076	0.1379	0.1508	0.02992	0.8192
	S	0.5612 ±0.000616	0.4923 ±0.000459	0.1227 ±0.000381	0.1159 ±0.000901	0.0406 ±0.000784	0.8435 ±0.000735
4	M	0.3078	0.2591	0.1582	0.1878	0.002413	0.8098
	S	0.3022 ±0.000402	0.2543 ±0.000223	0.1585 ±0.000913	0.1727 ±0.00133	0.002582 ±0.000353	0.8247 ±0.00118

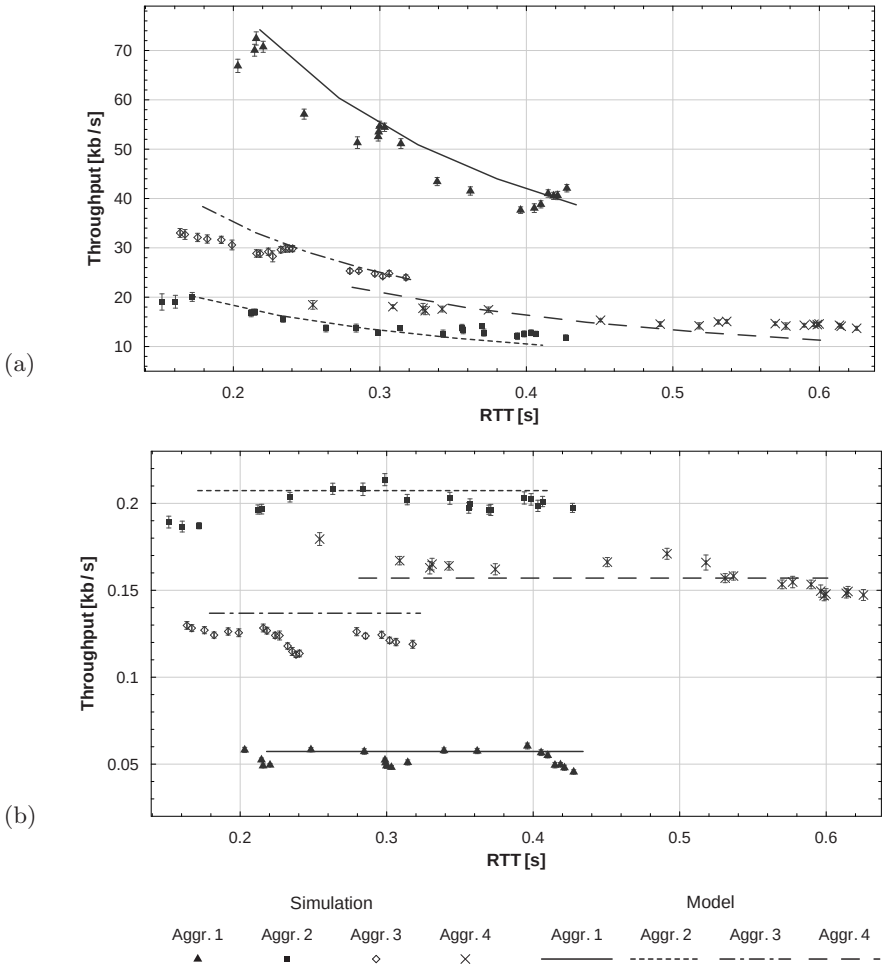


Figure 5.24: *Experiment 8*: Per-flow characteristics: throughput (a) and packet drop probability (b); srTCM + WRED

all the characteristics are well predicted by the model. Values of throughput and goodput are slightly overestimated by the model.

In Table 5.11, per-color characteristics are shown. Namely, the structure of the overall traffic, expressed by the values of P^{qr}, P^{qy}, and P^{qg} as well as packet drop probabilities per-color are given. Packet drop probabilities are predicted quite well. The largest discrepancy is for P^{dg} but its value is very small. As regards the

traffic structure, P^{qg} is best predicted. In comparison to the experimental data the model suggests that more packets in the overall traffic would by marked as red and less packets would be marked as yellow. It can be explained by scrutinizing the per-aggregate characteristics (Tab. 5.12). The overestimation of aggregates' throughput imply the discrepancies in the marking probabilities. However, the marking probabilities are predicted acceptably well. The largest discrepancies appear when the marking probability has small values, that is, for P_k^{my} for $k = 1$ and $k = 3$ as well as for P_k^{mr} for $k = 1$. It is remarkable that per-aggregate packet drop probabilities as well as the throughput and goodput are predicted very well.

Figure 5.24 (a) presents the throughput achieved by individual flows in a function of their RTT. Points with confidence levels show results from simulation. Lines present approximate characteristics derived from the model. There are z connected points calculated for z propagation delays as described in the introduction to the current section. Analytical curves fit experimental data well but the aforementioned phenomena of overestimation of throughput for smaller values of RTT can be noticed. The slope of the analytical curve clearly increases faster when RTT decreases.

The graph presented in Fig. 5.24 (b) supplements the comparison of the average per-aggregate packet drop probability presented in Tab. 5.12 with the view on the drop probability experienced by individual flows. The analytical curve's slope equals 0 since it was assumed that packets of all flows within the aggregate experience the same drop probability. The graph proves that above assumption is correct. Packets belonging to different individual flows are dropped with a similar probability.

Another observation that can be made from the experiment is that all aggregates almost achieve their target rates. It can be, however, noted that the sum of CIR's of all aggregates equals the bottleneck capacity and there is no excess bandwidth to be shared. It such a case, if all aggregates are generating the traffic volume about or above their CIR's, they all should achieve their target rates. This goal of DiffServ is achieved. If at least one of the aggregates would send packets with a rate below its CIR, the other aggregates would send packets with a rate exceeding their CIR's sharing the extra bandwith available.

Experiment 9

The network configuration with trTCM meter/markers and a WRED queue with staggered thresholds' configuration is scrutinized in this experiment. The number of traffic aggregates was 3. The number of flows constituting each aggregate was 40. Unlike in the previous experiment, all meter/markers were configured in the same way. The aggregates were intended to share the bottleneck capacity equally. Parameters of trTCM were as follows: $CIR = 1\,\mathrm{Mb/s}$,

Table 5.13: *Experiment 9*: WRED queue configuration

Q_{max}	60	packets
$minTh_r$	5	packets
$maxTh_r$	15	packets
$minTh_y$	15	packets
$maxTh_y$	30	packets
$minTh_g$	30	packets
$maxTh_g$	45	packets
$Pmax_r$	0.2	
$Pmax_y$	0.1	
$Pmax_g$	0.02	

$PIR = 1.5\,\text{Mb/s}$, $CBS = 10\,\text{kB}$, $PBS = 20\,\text{kB}$. The packet size was set to 1 kB. Propagation delays for aggregates $k = 1, 2, 3$ were selected form the following ranges: 700 ms — 1.6 s, 150 — 300 ms, and 100 — 300 ms, respectively. The bottleneck capacity was 5 Mb/s. Configuration parameters of the WRED queue are shown in Tab. 5.13.

Results for the experiment are presented in Tables 5.14 — 5.16 and Figure 5.25. The general system characteristics (Tab. 5.14) are predicted satisfactorily well by the model. For example, the goodput, B^{out} is overestimated by less than 5%, what is a very good approximation. The packet drop probability is also overestimated by the model but it is implied by the overestimation of throughput. A higher value of $\bar{P}^d$ compensates for the overestimated B. The difference between simulated and analytical average queue sizes is not dismissive.

As presented in Tab. 5.15, per-color characteristics are very well predicted. Only the drop probability of green packets is highly underestimated but, in fact, $\bar{P}^{dg}$ takes very small values, i.e., in order of 10^{-4}. As discussed in Section 5.3, such a high discrepancy may occur for very small drop probabilities but can be neglected. The value of P^{qg} is about 7% underestimated by the model in favor of P^{qr}. As in the previous experiment, it can be expounded by the overestimation of the throughput caused by the Padhye formula.

The per-aggregate characteristics (Tab. 5.16) were also well approximated by the model. An interesting conclusion concerning fairness in sharing excess bandwidth can be drawn from these characteristics. The sum of CIR's is 3 Mb/s. Since

Table 5.14: *Experiment 9*: General characteristics of a system consisting of trTCM meter/markers and WRED queue

	B [Mb/s]	B^{out} [Mb/s]	$\bar{q}$ [packets]	$\bar{P}^d$
M	5.725	4.993	15.282	0.1278
S	5.311 ±0.0038	4.756 ±0.00363	14.499 ±0.0227	0.104 ±0.00026

Table 5.15: *Experiment 9*: General, per-color characteristics of a system consisting of trTCM meter/markers and WRED queue

	P^{qr}	P^{qy}	P^{qg}	$\bar{P}^{dr}$	$\bar{P}^{dy}$	$\bar{P}^{dg}$
M	0.2236	0.2558	0.5206	0.5536	0.01591	0.00000589
S	0.1893 ±0.000503	0.254 ±0.000339	0.5567 ±0.000327	0.5357 ±0.0012	0.0121 ±0.00028	0.00002655 ±0.0000111

Table 5.16: *Experiment 9*: Per-aggregate characteristics of a system consisting of trTCM meter/markers and WRED queue

k		B_k [Mb/s]	B_k^{out} [Mb/s]	$\bar{P}_k^d$	P_k^{mr}	P_k^{my}	P_k^{mg}
1	M	1.541	1.481	0.03869	0.06143	0.2946	0.644
	S	1.391 ±0.0035	1.335 ±0.00366	0.04048 ±0.000313	0.05739 ±0.000497	0.2547 ±0.00122	0.6879 ±0.00119
2	M	2.065	1.744	0.1555	0.2739	0.2447	0.4815
	S	1.924 ±0.00261	1.685 ±0.00224	0.1242 ±0.000387	0.2219 ±0.00114	0.2585 ±0.000332	0.5196 ±0.000843
3	M	2.119	1.768	0.1657	0.2924	0.2385	0.4691
	S	1.996 ±0.00322	1.736 ±0.00239	0.1301 ±0.00048	0.2497 ±0.00139	0.2492 ±0.000453	0.501 ±0.000968

the bottleneck capacity is 5 Mb/s an extra bandwith exists. It is intended for the aggregates to share it fairly. It appears that this goal is not achieved. Aggregate 1, experiencing the highest RTTs, achieves a significantly lower throughput than the other ones despite experiencing the lowest packet drop probability. The presented results prove that the distribution of the flows' RTTs within the aggregate may have a visible impact on fairness offered by DiffServ.

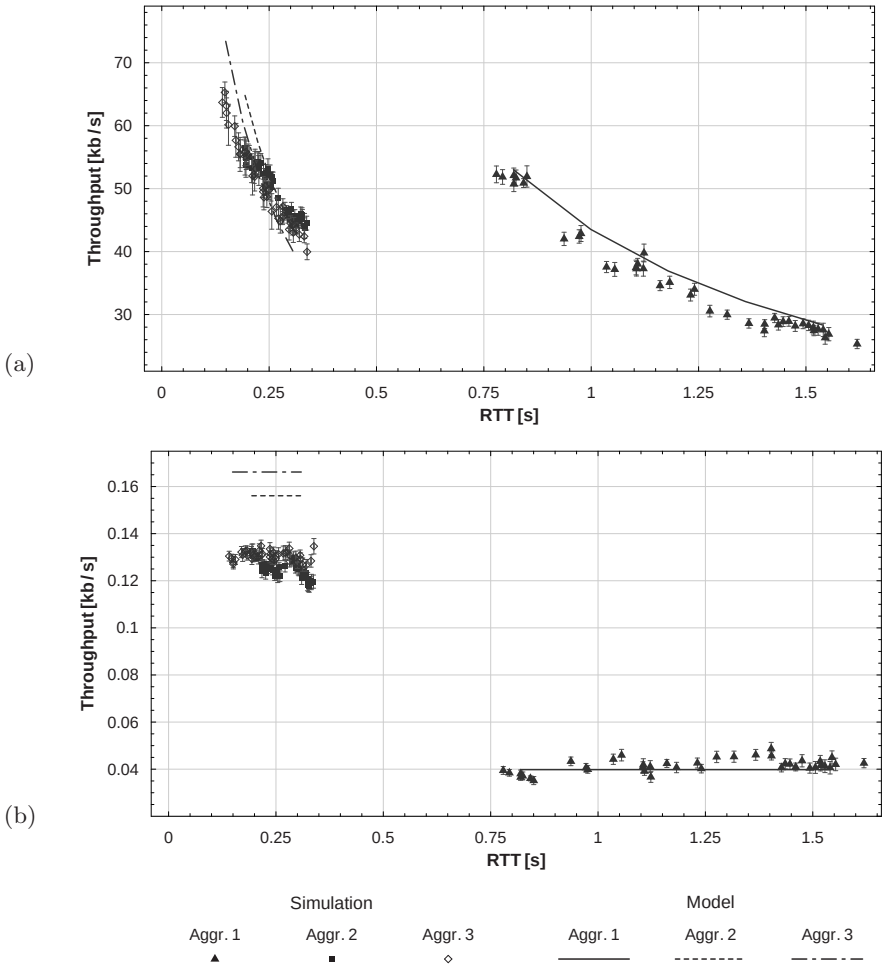


Figure 5.25: *Experiment 9*: Per-flow characteristics: throughput (a) and packet drop probability (b); trTCM + WRED

Fig. 5.25 shows per-flow throughput (a) and packet drop probability (b) characteristics. Fig. 5.25 (a) reveals unfairness between individual flows within an aggregate. For example, a flow of the 3rd aggregate experiencing $RTT \approx 150\,\text{ms}$ achieves throughput $\approx 66\,\text{kb/s}$ while another flow with $RTT = 350\,\text{ms}$ sends packets with a rate of $40\,\text{kb/s}$ only. Per-flow drop probabilities obtained from simulation are approximately constant within the aggregate (Fig. 5.25 (b)).

Table 5.17: *Experiment 10*: WRED queue configuration

Q_{max}	60	packets
$minTh_r$	10	packets
$maxTh_r$	45	packets
$minTh_y$	10	packets
$maxTh_y$	45	packets
$minTh_g$	10	packets
$maxTh_g$	45	packets
$Pmax_r$	0.4	
$Pmax_y$	0.1	
$Pmax_g$	0.005	

Experiment 10

The network configuration studied in this experiment consisted of TSW3CM meter/markers and a WRED queue with overlapped thresholds configuration. Each of the two aggregates was measured and marked by separate instance of TSW3CM but the configuration of both meter/markers was identical. Namely, CIR was set to 2 Mb/s, PIR was 2.5 Mb/s while observation window size W was set to 1 second. The propagation delay for the first aggregate was selected from the range of 300 — 700 ms while for the second one from the range of 500 — 800 ms. The number of flows constituting each aggregate was 60. The packet size used in the experiment was 1 kB. The bottleneck capacity was 5 Mb/s. Configuration parameters of the queue are shown in Tab. 5.17.

Results for the experiment are presented in Tables 5.18 — 5.20 and Figure 5.26. Values of the total throughput B and goodput B^{out} as well as the overall packet drop probability $\bar{P}^d$ are predicted very well in this experiment (Tab. 5.18). A significant difference in the simulated and predicted average queue size $\bar{q}$ is apparent. As discussed in Section 5.3, such high discrepancies are possible for the model of WRED but, nontheless, they do not necessarily imply the failure of the model. It the case of current experiment all other parameters are predicted very well. According to simulation the average queue size equals 26.625 packets and results in the average queuing delay at the bottleneck router of about 40 ms. The value of $\bar{q}$ predicted by the model is 35.672 packets what implies the queuing delay of 54 ms. Therefore, the difference in the queuing delay is in order

Table 5.18: *Experiment 10*: General characteristics of a system consisting of TSW3CM meter/markers and WRED queue

	B [Mb/s]	B^{out} [Mb/s]	$\bar{q}$ [packets]	$\bar{P}^d$
M	5.392	4.999	35.672	0.073
S	5.204 ±0.0026	4.836 ±0.00299	26.625 ±0.115	0.0707 ±0.00017

Table 5.19: *Experiment 10*: General, per-color characteristics of a system consisting of TSW3CM meter/markers and WRED queue

	P^{qr}	P^{qy}	P^{qg}	$\bar{P}^{dr}$	$\bar{P}^{dy}$	$\bar{P}^{dg}$
M	0.07652	0.1817	0.7418	0.3153	0.1061	0.0399
S	0.07561 ±0.00138	0.1588 ±0.00112	0.7656 ±0.000559	0.4012 ±0.00312	0.1068 ±0.00123	0.03056 ±0.000506

Table 5.20: *Experiment 10*: Per-aggregate characteristics of a system consisting of TSW3CM meter/markers and WRED queue

k		B_k [Mb/s]	B_k^{out} [Mb/s]	$\bar{P}_k^d$	P_k^{mr}	P_k^{my}	P_k^{mg}
1	M	2.796	2.57	0.08097	0.1062	0.1786	0.7152
	S	2.725 ±0.00416	2.512 ±0.00447	0.07788 ±0.000294	0.09793 ±0.00178	0.1694 ±0.000788	0.7327 ±0.00132
2	M	2.596	2.429	0.06442	0.04454	0.1851	0.7704
	S	2.479 ±0.00398	2.324 ±0.00438	0.06277 ±0.000336	0.05106 ±0.00167	0.1471 ±0.00216	0.8018 ±0.00122

of 14 ms which is no more than 4% contribution to the smallest RTT reported in the simulation. Thus, it not only does not disrupt other results but even improve them by compensating the overestimation of the throughput by the Padhye formula. Note that compensation is stronger for lower RTTs. Such a conclusion is supported also by the per-aggregate results (Tab. 5.20). The overestimation of B_k and B_k^{out} is very small, especially for the first aggregate ($k = 1$). Very accurate prediction of the per-aggregate throughput results in a high conformity between experimental and model driven packet marking probabilities. Also per-aggregate drop probabilities are approximated very well by the model. Thus, not

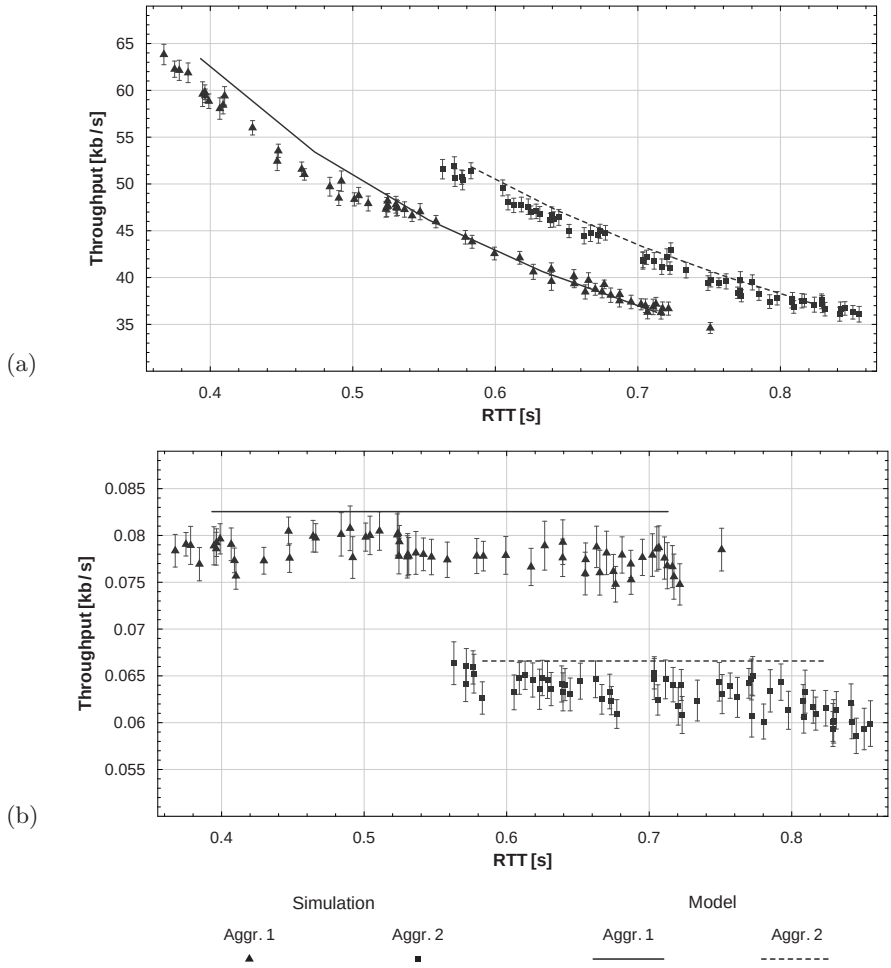


Figure 5.26: *Experiment 10*: Per-flow characteristics: throughput (a) and packet drop probability (b); TSW3CM + WRED

surprisingly, per-color characteristics shown in Tab. 5.19 yield high accuracy of the model. Also per-flow characteristics presented in Fig. 5.26 support the above theses.

The problem of unfairness in sharing the excess bandwith is again visible in that experiment. Both aggregates exceeded their target rates (CIRs) but aggregate 1, experiencing a set of relatively lower RTTs, consumed slightly more bottleneck capacity than aggregate 2.

Table 5.21: *Experiment 11*: RIO-C queue configuration

Q_{max}	50	packets
$minTh_r$	5	packets
$maxTh_r$	10	packets
$minTh_y$	10	packets
$maxTh_y$	20	packets
$minTh_g$	20	packets
$maxTh_g$	40	packets
$Pmax_r$	0.2	
$Pmax_y$	0.1	
$Pmax_g$	0.02	

Experiment 11

The network configuration examined in that experiment was consisting of trTCM meter/markers and a RIO-C queue with staggered thresholds' configuration. There were 2 traffic aggregates sharing the bottleneck capacity assumed in that experiment to be 2 Mb/s. The number of flows constituting each aggregate was 40. Both meter/markers were configured in the same way. Parameters of trTCM were as follows: $CIR = 0.95$ Mb/s, $PIR = 1.4$ Mb/s, $CBS = 10$ kB, $PBS = 20$ kB. The packet size was set to 1 kB. Propagation delays for aggregates 1 and 2 were randomly chosen form the ranges: 70 ms — 1.2 s and 50 — 300 ms, respectively. Configuration parameters of the RIO-C queue are shown in Tab. 5.21.

Results for the experiment are presented in Tables 5.22 — 5.24 and Figure 5.27. Values of the total throughput B and goodput B^{out} as well as the overall packet drop probability $\bar{P}^d$ are predicted unquestionably very well (Tab. 5.22). Differences are less than 0,5%. The average queue size is overestimated by the model but this fact has a neglectable impact on the whole model's accuracy.

The traffic color structure is also well approximated by the model (Tab. 5.23). Values of P^{qg} and P^{qy} obtained from simulation and the model are very close and are ≈ 0.2 and ≈ 0.79, respectively. Only P^{qr} is almost 100% overestimated by the model but this value is approximately 100 times smaller than P^{qy} and does not signifficantly impact the results. Similarily, packet drop probabilities $\bar{P}^{dr} \approx 1$ and $\bar{P}^{dy} \approx 0.7$ are well predicted. Only the value of probability of dropping green packets is very small (experimantal value is ≈ 0.001) and is overestimated by the model.

Table 5.22: *Experiment 11*: General characteristics of a system consisting of trTCM meter/markers and RIO-C queue

	B [Mb/s]	B^{out} [Mb/s]	$\bar{q}$ [packets]	$\bar{P}^d$
M	2.3479	1.9998	23.437	0.1482
S	2.3474 ±0.00145	1.9966 ±0.000219	21.424 ±0.0373	0.149 ±0.0005

Table 5.23: *Experiment 11*: General, per-color characteristics of a system consisting of trTCM meter/markers and RIO-C queue

	P^{qr}	P^{qy}	P^{qg}	$\bar{P}^{dr}$	$\bar{P}^{dy}$	$\bar{P}^{dg}$
M	0.004888	0.2061	0.789	0.9874	0.6878	0.00207
S	0.002505 ±0.000104	0.2041 ±0.000486	0.7934 ±0.000513	1. ±0.	0.7159 ±0.00151	0.001105 ±0.0000627

Table 5.24: *Experiment 11*: Per-aggregate characteristics of a system consisting of trTCM meter/markers and RIO-C queue

k		B_k [Mb/s]	B_k^{out} [Mb/s]	$\bar{P}_k^d$	P_k^{mr}	P_k^{my}	P_k^{mg}
1	M	1.108	0.974	0.1209	0.001366	0.1713	0.8274
	S	1.103 ±0.00117	0.9607 ±0.000947	0.1288 ±0.00038	0.001518 ±0.000129	0.1694 ±0.000691	0.8291 ±0.00072
2	M	1.24	1.026	0.1727	0.008035	0.2373	0.7547
	S	1.245 ±0.00174	1.036 ±0.000915	0.1677 ±0.000767	0.003379 ±0.000209	0.2348 ±0.00113	0.7619 ±0.00124

As regards per-aggregate characteristics (Tab. 5.24), those for aggregate 1 are predicted very accurately. No significant differences between experimental and analytical data can be noticed. In the case of aggregate 2 only the packet marking probability as red, P_2^{mr} is overestimated. It results in aforementioned overestimation of P^{qr}. There is almost no difference between the analytically and experimentally obtained throughput achieved by aggregate 1. However, as Fig. 5.27 (a) reveals, the per-flow throughput is not absolutely well predicted. The throughput of flows experiencing lower RTTs is slightly overestimated while the throughput of flows experiencing higher RTTs is slightly underestimated. This is because the repeatedly discussed feature of the Padhye formula revealing in DiffServ networks is still present in that case. Nonetheless, since the analytical

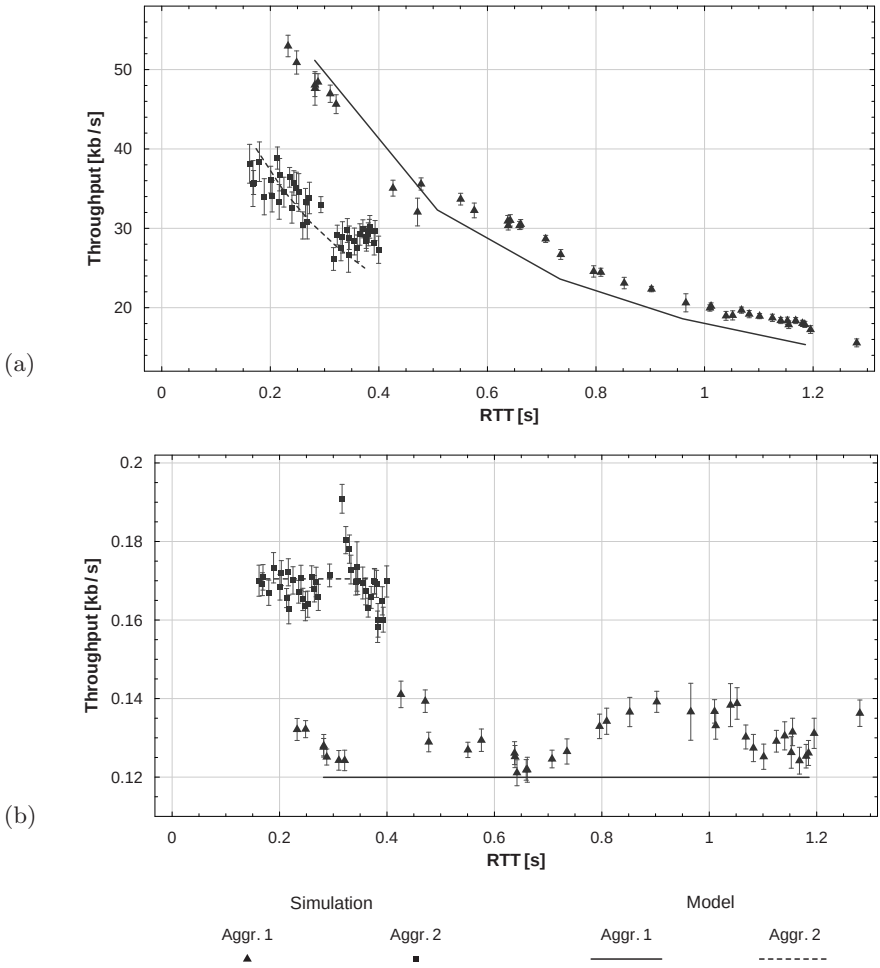


Figure 5.27: *Experiment 11*: Per-flow characteristics: throughput (a) and packet drop probability (b); trTCM + RIO-C

curve crosses the trend of experimental points, the resulting value of B_1 is very well approximated. In the case of aggregate 2 the feature of Padhye formula is elusive since individual flows' throughputs obtained from simulation are spread.

The problem of unfairness between aggregates consisting of flows of different RTTs is again visible in that experiment. Aggregate 1 received slightly smaller fraction of the bottleneck capacity than aggregate 2 while both aggregates should receive exactly half of it (Tab. 5.23).

Table 5.25: *Experiment 12*: Configuration of TSW3CM and ranges of propagation delays

k	δ_k^{min}—δ_k^{max}	CIR_k	PIR_k	W_k
1	150—600 ms	1 Mb/s	2 Mb/s	1 s
2	250—600 ms	0.25 Mb/s	0.5 Mb/s	1 s
3	200—800 ms	0.5 Mb/s	1 Mb/s	1 s

Table 5.26: *Experiment 12*: RIO-C queue configuration

Q_{max}	60	packets
$minTh_r$	10	packets
$maxTh_r$	45	packets
$minTh_y$	10	packets
$maxTh_y$	45	packets
$minTh_g$	10	packets
$maxTh_g$	45	packets
$Pmax_r$	0.4	
$Pmax_y$	0.1	
$Pmax_g$	0.005	

Experiment 12

The network configuration studied in this experiment consisted of TSW3CM meter/markers and a RIO-C queue with the overlapped thresholds configuration. There were 3 traffic aggregates sharing the bottleneck capacity unevenly. Configuration of TSW3CM for each aggregate as well as ranges of propagation delays are presented in Tab. 5.25. Each aggregate were constituted by 20 TCP flows. Packet size was set to 500 B. Configuration parameters of the queue are shown in Tab. 5.26.

Results for the experiment are presented in Tables 5.27 — 5.29 and Figure 5.28. In the case of that experiment the discussion on the results starts from scrutiny of per-aggregate results. As presented in Tab. 5.29, the throughput achieved by aggregate 1 is slightly overestimated while the throughputs of the other, especially aggregate 2, are underestimated. Such a relation is replicated

Table 5.27: *Experiment 12*: General characteristics of a system consisting of TSW3CM meter/markers and RIO-C queue

	B [Mb/s]	B^{out} [Mb/s]	$\bar{q}$ [packets]	$\bar{P}^d$
M	2.111	1.999	37.572	0.05308
S	2.08 ±0.000797	1.978 ±0.000935	34.372 ±0.124	0.049 ±0.00013

Table 5.28: *Experiment 12*: General, per-color characteristics of a system consisting of TSW3CM meter/markers and RIO-C queue

	P^{qr}	P^{qy}	P^{qg}	$\bar{P}^{dr}$	$\bar{P}^{dy}$	$\bar{P}^{dg}$
M	0.0000884	0.1716	0.8283	0.4334	0.2639	0.00938
S	0.003342 ±0.000443	0.1765 ±0.000764	0.8202 ±0.00106	0.3767 ±0.0344	0.1582 ±0.00218	0.02425 ±0.000339

Table 5.29: *Experiment 12*: Per-aggregate characteristics of a system consisting of TSW3CM meter/markers and RIO-C queue

k		B_k [Mb/s]	B_k^{out} [Mb/s]	$\bar{P}_k^d$	P_k^{mr}	P_k^{my}	P_k^{mg}
1	M	1.0997	1.064	0.03275	$0.1{\cdot}10^{-21}$	0.09184	0.9082
	S	1.007 ±0.0034	0.9749 ±0.00353	0.03216 ±0.000253	0. ±0	0.04512 ±0.00119	0.9549 ±0.00119
2	M	0.39797	0.3565	0.1041	0.000469	0.3713	0.6282
	S	0.4419 ±0.00182	0.4054 ±0.00185	0.08268 ±0.000496	0.01564 ±0.00243	0.4238 ±0.00159	0.5606 ±0.00277
3	M	0.6134	0.5788	0.05644	$0.079{\cdot}10^{-9}$	0.1849	0.8151
	S	0.631 ±0.00178	0.5981 ±0.00182	0.0522 ±0.000262	0.0000168 ±0.0000153	0.2128 ±0.00242	0.7871 ±0.00243

in the goodputs of aggregates. As mentioned before, a natural behavior is overestimation of the throughput due to the feature of the Padhye formula. In the case of that configuration it was compensated by overestimation of the average queue size (Tab. 5.27) and drop probabilities for aggregates 2 and 3, $\bar{P}_2^d$ and $\bar{P}_3^d$, respectively. The inaccuracies in throughput estimation imply some discrepancies in estimation of packet marking probabilities but the resulting P^{qy} and P^{qg} are predicted very well.

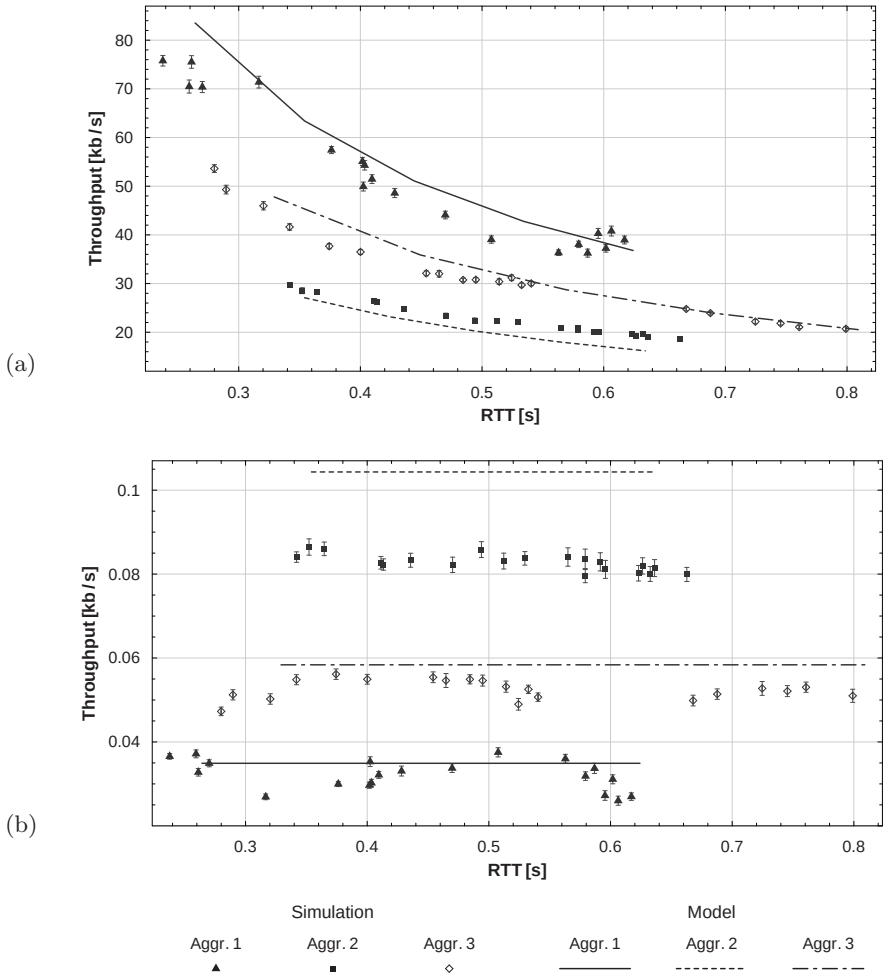


Figure 5.28: *Experiment 12*: Per-flow characteristics: throughput (a) and packet drop probability (b); TSW3CM + RIO-C

Per-flow characteristic of the throughput confirms that fact of noticeable underestimation of B_2 (Fig. 5.28). Apparently, all points representing throughputs of individual flows of aggregate 2 lie above the analytical curve. An opposite behavior is noticeable for aggregate 1.

General drop probability characteristics, including $\bar{P}^{dr}$, $\bar{P}^{dy}$, $\bar{P}^{dg}$, and $\bar{P}^{d}$, obtained from simulation and the model noticeably differ but the discrepancy still

does not disqualify the model. The experimental overall packet drop probability is 0.049 while predicted by the model is 0.053. Finally, the overall throughput and goodput are well predicted.

Generally, in the currently discussed experiment the discrepancies between model and simulation are most visible, especially regarding per-aggregate characteristics. Nevertheless, the model still provides acceptable approximation of the system characteristics. The model gives an approximate view on the system behavior and reflects its real characteristics.

It is not to say that the modular model applied to a network configuration consisting of TSW3CM meter/markers and a RIO-C queue is least tractable. Among all the researched configurations less and more accuracy of the modular model was observed. As mentioned in the introduction to the current section, the presented selection of experiments on one hand encompasses a variety of configurations but on the other hand was chosen in a way assuring that most common features of the modular model are reflected. Thus, experiments demonstrating various levels of the model's accuracy are presented.

Part III

Finale

6 Conclusions

The goal of the research presented in the dissertation was to develop analytical models of DiffServ network elements supporting Assured Forwarding PHB. Several mathematical methods, including the queuing theory and the probability theory, were used. The dissertation presents analytical models of meter/markers and droppers, proves that the models can be used for finding several characteristics of meter/markers and droppers, and proposes how the models can be used in a wider context, i.e., for developing and solving models of networks consiting of several nodes and DiffServ network element instances.

The implementations of meter/markers are categorized into two groups: token bucket based and rate estimator based. In the first group, srTCM and trTCM meter/markers are considered. An example of a rate estimator based meter/marker is TSW3CM. The models for the above meter/markers make it possible to calculate packet marking probabilities as a function of meter/marker configuration parameters and the average rate of the traffic that is subject to the metering and marking process.

The popular implementations of a dropper are Multi-RED queues. Two types of Multi-RED queue are under research: WRED and RIO-C. In both cases, two types of threshold configuration were examined, namely, staggered and overlapped. Proposed models make it possible to calculate the overall packet drop probability, per-color packet drop probability, and the average queue size in a function of Multi-RED queue configuration parameters and the throughput of the traffic incoming to the queue.

The models of individual DiffServ network elements can be also used for building a model of a more complex network architecture. Example of such an application of the models, called modular model, is also presented. The modular model enables finding packet marking probability per aggregate, packet drop probability per aggregate, overall packet drop probability, summary and per-aggregate average throughput and goodput, average queue length at the bottleneck router,

and packet drop probability per color (per drop precedence). The model helps predicting the system behavior under various conditions.

Experimental validation of all the proposed analytical models is provided. Presentation of simulation results is followed by a discussion of the results including consideration on possible explanation of some discrepancies between the model and the simulation. For the cases where two models are considered, guidelines for choosing the one most appropriate under particular circumstances are provided. The methodology for assuring credibility of experiments is widely discussed.

The models enable quick finding of meter/marker and dropper characteristics under a particular configuration and traffic volume. Additionally, these models together with traffic models facilitate predicting characteristics of the whole system. The models can be used in planing of a network architecture. They can be useful for taking a decision on choosing concrete values of DiffServ elements' parameters in a real network. Appropriate selection of configuration parameters of network elements is crucial for providing a certain level of QoS for customers as well as for fulfilling the SLA. Carefull scrutiny of meter/markers and droppers characteristics as well as characteristics of a compound network architecture supports such a decision. The models are usefull for network engineers. They are also valuable for students who can help them to better understand how meter/markers and droppers work. Therefore, the models are also valuable for teachers.

6.1 Achievements and contributions

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

1. A novel idea of developing analytical models of individual DiffServ network elements is proposed and realized.

2. The model of Single Rate Three Color Marker (srTCM) based on the M/M/1/K system is developed in the dissertation and successfully validated. Closed form formulas for packet marking probabilities as red, yellow and green are proposed.

3. Two alternative models of Two Rate Three Color Marker (trTCM) are proposed. The first model is based on the M/M/1/K queuing system while the second on D/M/1/K. In the former case, closed form formulas for marking probabilities are presented. Solutions for the model based on the D/M/1/K queuing system can by found by solving a set of equations describing the state probabilities. A method to find those solutions is presented step by step. The accuracy of both models is validated experimentally, compared

and discussed. Final guidelines for choosing between the models are presented.

4. The model for Time Sliding Window Three Color Marker (TSW3CM) is developed. The probability density function of the TSW-averaged traffic rate estimated by the Time Sliding Window rate estimator is proposed. The PDF applied to packet marking functions of TSW3CM makes it possible to find marking probabilities. Also boundary characteristics of the marking probabilities are proposed. The model can be easily applied to other similar meter/markers such as TSW2CM or ItswTCM that differ from TSW3CM only in marking functions and, thus, only the integral formulas for marking probabilities must be changed.

5. Two approaches to modeling a Multi-RED queue are proposed. They both are based on queuing systems with balking, namely M(n)/M/1/K and M(n)/D/1/K.

6. The model of a WRED queue based on the M(n)/M/1/K queuing system, presented in the dissertation, was previously considered and scrutinized by other researchers. However, it is validated in a wider range of network conditions and configurations. It is compared with a novel model based on the M(n)/D/1/K queuing system. The latter model, proposed by the author of the dissertation proves to perform better than the previous one. It predicts WRED characteristics better.

7. Development of both models of RIO-C, based on the M(n)/M/1/K queuing system and the M(n)/D/1/K queuing system, are original achievements of the author of the dissertation. Since the sizes of virtual queues in RIO-C do not equal the size of the physical queue, an additional technique is used to calculate packet drop probabilities appropriately. Namely, the Bernoulli scheme is applied.

8. It is proved, that the models for individual DiffServ network elements is useful for creating the analytical model of a more complex network architecture encompassing also the traffic model. The analytical model of a TCP Reno source, developed by the other author, is used. Results of five experiments encompassing various network configurations are presented. Namely, the following architectures are examined: srTCM meter/markers and WRED queue, trTCM meter/markers and WRED queue, TSW3CM meter/markers and WRED queue, trTCM meter/markers and RIO-C queue, and TSW3CM meter/markers and RIO-C queue.

9. As presented experiments comparing experimental and analytical characteristics of the discussed models show, the models approximate the real network characteristics accurately.

10. A clarification of the QoS-related terminology is an additional achievement of the author.

11. As a side effect of the experimental validation of the models, it was shown that the Padhye formula for TCP throughput applied for predicting TCP behavior in a DiffServ network tends to overestimate the achieved throughput for small values of RTT. Additional conclusions have been drawn regarding the problem of fairness in DiffServ networks.

In the light of the presented achievements it can be stated that the thesis: ***It is possible to analytically express characteristics of meter/markers and droppers used for DiffServ network supporting Assured Forwarding PHB***, has been proved.

Appendices

A Simulation credibility

The appendix presents sample results of the tests performed to verify the credibility of statistical data collected from simulations. The methods for determining the warm-up period of the simulation as well as verfication of samples' autocorrelation described in Section 5.1.4 were used.

A.1 Determining the warm-up period

The data was collected in intervals of length of 20 seconds. The number of independent runs M was 32. An estimate of the value of packet drop probability $\hat{p}_i^d$ was found in each interval i. The number of intervals was 500. A moving average algorithm with parametr $k = 10$ was used (formula 5.16). Then, the graph of logarithm of standard deviation s_N againts logarithm of N was plotted (Fig. A.1). In the example presented by Tyszer [149], it was easy to find the point where the slope of the curve clearly changes sign and immediately becomes ≈ -0.5. The curve's slope in Fig. A.1 approaches the value of -0.5 slowly so determination the of the warm-up period is not obvious. It was assumed that the wam-up period is over when the slope of the curve is about -0.49. To make the assesement of the curve easier, the dashed line with the slope of -0.5 was drawn in Fig. A.1. This point where the warm-up period ends is indicated by an arrow. The decision was also supported by analyzing Figures A.2 and A.3.

Finally, it was decided that the number of the last sample belonging to the warm-up period of the simulation is 50 and the length of the warm-up period in seconds is 1000. The longest warm-up period among all the experiments was 1500 s. However, it did not exceed 150 s in most cases.

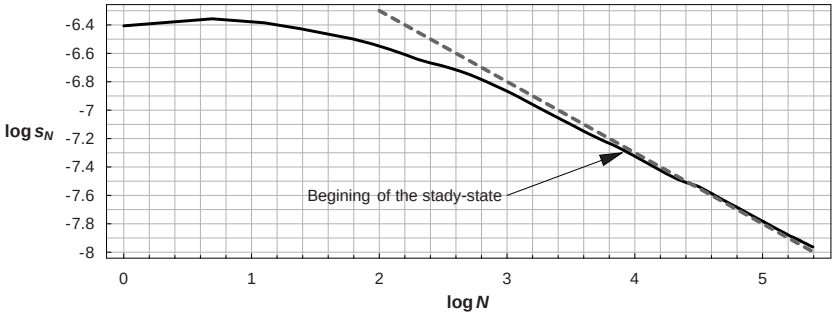


Figure A.1: Graph of logarithm of standard deviation againts logarithm of N

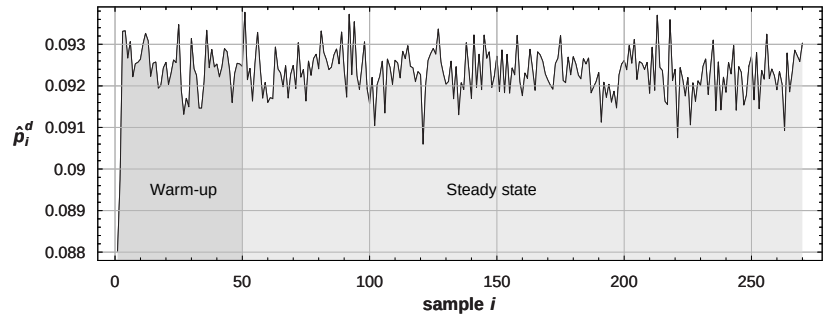


Figure A.2: Packet drop probability $\hat{p}_i^d$ as a fuction of sample number N

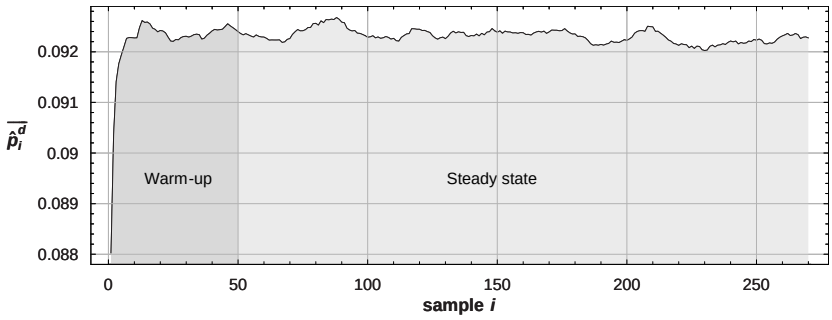


Figure A.3: Packet drop probability as a fuction of sample number N smoothed by moving average algorithm with $k = 10$

A.2 Verification of autocorrelation

As mentioned in Section 5.1.2 *the method of batch means* was used in simulations. Therefore, verification of the indepentence of samples was necessary. In this example the interval length was 100 s (used in many simulations). The number of simulations was $n = 100$. To verify that the samples obtained form the simulation the autocorrelation function was drawn (Fig. A.4). The number of lags taken into account was $n/4$. Values of autocorrelation functions for all lags from 1 to 25 lie within the confidence interval represented in Fig. A.4 by dashed horizontal lines.

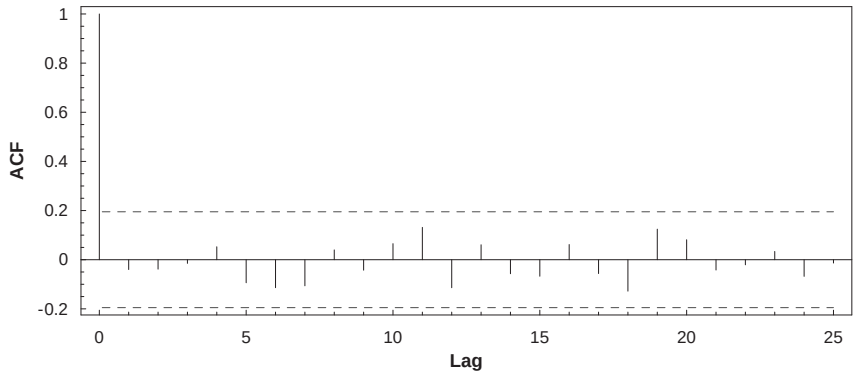


Figure A.4: Autocorrelation function

Ljung-Box and McLeod-Li tests were performed as well. The results are presented in Tab. A.1. The value statistics of Ljung-Box and McLeod-Li tests were obtained from formulas (5.19) and (5.20), respectively. According to both tests, on the significance level of 0.05, there is no evidence to reject the hypothesis that the samples are uncorrelated (p-value > 0.05). It means that the samples are independent and the *GARCH* effect is not present.

Table A.1: Results of Ljung-Box and McLeod-Li tests

Ljung-Box test	*McLeod-Li test*
$Q_{LB} = 16.8433$	$Q_{ML} = 16.8493$
p-value= 0.8873	p-value= 0.8872

Bibliography

[1] A. A. Abouzeid and S. Roy. Modeling random early detection in a differentiated services network. *Computer Networks*, 40(4):537–556, Nov. 2002. *(Cited on pages 34, 44 and 61.)*

[2] J. Adams, A. IJsselmuiden, and L. Roberts. An advanced QoS protocol for real-time content over the internet. In H. de Meer and N. Bhatti, editors, *Proc. 13th International Workshop on Quality of Service (IWQoS)*, volume 3552 of *Lecture Notes In Computer Science*, pages 164–177, Passau, Germany, Jun. 2005. Springer-Verlag. *(Cited on page 4.)*

[3] W. A. Aiello, Y. Mansour, S. Rajagopolan, and A. Rosen. Competitive queue policies for differentiated services. In *Proc. IEEE INFOCOM 2000*, volume 2, pages 431–440, Tel Aviv, Israel, Mar. 2000. *(Cited on page 35.)*

[4] E. Altman, K. Avrachenkov, and C. Barakat. A stochastic model of TCP/IP with stationary random losses. *IEEE/ACM Transactions on Networking*, 13(2):356–369, Apr. 2005. *(Cited on page 36.)*

[5] I. Andrikopoulos, L. Wood, and G. Pavlou. A fair traffic conditioner for the assured service in a differentiated services Internet. In *Proc. IEEE ICC 2000*, volume 2, pages 806–810, New Orleans, USA, Jun. 2000. *(Cited on page 32.)*

[6] P. Anelli and G. Le Grand. Differentiated services over shared media. In L. Wolf, D. Hutchinson, and R. Steinmetz, editors, *Proc. 9th International Workshop on Quality of Service (IWQoS)*, volume 2092 of *Lecture Notes In Computer Science*, pages 288–293, Karlsruhe, Germany, Jun. 2001. Springer-Verlag. *(Cited on page 44.)*

[7] C. Ash. *The Probability Tutoring Book*. IEEE Press, 1993. *(Cited on page 52.)*

[8] M. Barbera, A. Lombardo, and G. Schembra. A fluid-based model of time-limited TCP flows. *Computer Networks*, 44(3):275–288, Feb. 2004. *(Cited on pages 37 and 44.)*

[9] M. Barbera, A. Lombardo, G. Schembra, and C. A. Trecarichi. A fluid-flow model of RIO routers loaded by Markov modulated fluid processes. In *HPSR 2004: Workshop on High Performance Switching and Routing*, pages 67–71, Phoenix, USA, Apr. 2004. *(Cited on pages 33, 34, 40 and 44.)*

[10] M. Barbera, A. Lombardo, G. Schembra, and C. A. Trecarichi. Fluid flow analysis of TCP flows in a DiffServ environment. Accepted for publication in European Transactions on Telecommunications, 2006. *(Cited on pages 34 and 40.)*

[11] P. Barford and M. Crovella. Generating representative web workloads for network and server performance evaluation. In *Proc. ACM SIGMETRICS 1998*, volume 26, pages 151–160, Madison, USA, Jul. 1998. *(Cited on pages 68 and 77.)*

[12] M. Baumann, D. Marandin, and S. Sulaiman. Combined modelling of TCP and multi-RED in DiffServ networks. *European Transactions on Telecommunications*, 16(3):217–224, May 2005. Wiley. *(Cited on pages 33, 34, 35, 39 and 56.)*

[13] J. C. R. Bennet and H. Zhang. WF2Q: Worst-case fair weighted fair queueing. In *Proc. IEEE INFOCOM 1996*, volume 1, pages 120–128, San Fransisco, USA, Mar. 1996. *(Cited on page 20.)*

[14] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services. *IETF RFC 2475*, Dec. 1998. *(Cited on pages 4, 14, 15, 16, 17 and 20.)*

[15] R. Bless, K. Nichols, and K. Wehrle. A lower effort per-domain behavior (PDB) for differentiated services. *IETF RFC 3662*, Dec. 2003. *(Cited on page 17.)*

[16] T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, Apr. 1986. *(Cited on page 76.)*

[17] T. Bonald, M. May, and J. Bolot. Analytic evaluation of RED performance. In *Proc. IEEE INFOCOM 2000*, volume 3, pages 1415–1424, Tel Aviv, Israel, Mar. 2000. *(Cited on pages 33, 34, 35, 36, 44 and 56.)*

[18] C. Bouras and A. Sevasti. Performance enhancement of an AF service using TCP-aware marking and dynamic WRED. In *Proc. 10th IEEE Symposium on Computers and Communication*, pages 642–647, Cartagena, Spain, Jun. 2005. *(Cited on page 32.)*

[19] G. E. P. Box and G. M. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day, Oakland, USA, 2 edition, 1976. *(Cited on page 75.)*

[20] R. Braden, D. Clark, and S. Shenker. Integrated services in the Internet architecture: an overview. *IETF RFC 1633*, Jun. 1994. *(Cited on page 4.)*

[21] P. J. Brockwell and R. A. Davis. *Time Series: Theory and Methods*. Springer-Verlag, New York, USA, 2nd edition, 1991. *(Cited on page 75.)*

[22] P. J. Brockwell and R. A. Davis. *Introduction to Time Series and Forecasting*. Springer-Verlag, New York, USA, 2nd edition, 2002. *(Cited on page 75.)*

[23] T. Bu and D. Towsley. Fixed point approximations for TCP behavior in an AQM network. In *Proc. ACM SIGMETRICS 2001*, volume 29, pages 216–225, Massachusetts, USA, Jun. 2001. *(Cited on page 35.)*

[24] R. Cáceres, P. Danzig, S. Jamin, and D. Mitzel. Characteristics of wide-area TCP/IP conversations. In *Proc. ACM SIGCOMM 1991*, volume 21, pages 101–112, Zürich, Switzerland, Sep. 1991. *(Cited on page 44.)*

[25] J. Cao, W. S. Cleveland, D. Lin, and D. X. Sun. The effect of statistical multiplexing on internet packet traffic: Theory and empirical study. Technical report, Bell Labs, 2001. *(Cited on page 43.)*

[26] J. Cao, W. S. Cleveland, D. Lin, and D. X. Sun. On the nonstationarity of Internet traffic. In *Proc. ACM SIGMETRICS 2001*, volume 29, pages 102–112, Massachusetts, USA, Jun. 2001. *(Cited on page 43.)*

[27] N. Cardwell, S. Savage, and T. Anderson. Modeling TCP latency. In *Proc. IEEE INFOCOM 2000*, volume 3, pages 1742–1751, Tel Aviv, Israel, Mar. 2000. *(Cited on pages 34, 35 and 61.)*

[28] C. Casetti and M. Mellia. A lightweight marker with partial state information for diffserv networks. *Computer Networks*, 49(4):541–560, Nov. 2005. *(Cited on pages 24, 33 and 34.)*

[29] Y. Chait, C. V. Hollot, V. Misra, D. Towsley, H. Zhang, and J. C. S. Lui. Providing throughput differentiation for TCP flows using adaptive two-color marking and two-level AQM. In *Proc. IEEE INFOCOM 2002*,

volume 2, pages 837–844, New York, USA, Jun. 2002. *(Cited on pages 23, 33, 34, 37 and 40.)*

[30] C.-J. Chang, Y.-H. Cheng, and L.-F. Lin. The traffic conditioner with promotion and fairness guarantee schemes for DiffServ networks. In *Proc. IEEE ICC 2003*, volume 1, pages 238–242, Anchorage, USA, May 2003. *(Cited on page 23.)*

[31] X. Chang and J. K. Muppala. Adaptive marking threshold for improving bandwidth assurance in a differentiated services network. In *Proc. IEEE GLOBECOM 2003*, volume 6, pages 3073–3077, San Francisco, USA, Dec. 2003. *(Cited on page 24.)*

[32] X. Chang and J. K. Muppala. Applying adaptive virtual queue to improve the performance of the assured forwarding service. In *Proc. IEEE ICC 2005*, volume 1, pages 100–104, Seoul, Korea, May 2005. *(Cited on page 32.)*

[33] X. Chang and J. K. Muppala. On improving bandwidth assurance in AF-based DiffServ networks using a control theoretic approach. *Computer Networks*, 49(6):816–839, Jun. 2005. *(Cited on pages 24, 33, 34, 35 and 40.)*

[34] D. D. Clark and W. Fang. Explicit allocation of best-effort packet delivery service. *IEEE/ACM Transactions on Networking*, 6(4):362–373, Aug. 1998. *(Cited on pages 26, 28, 33, 34 and 61.)*

[35] E. Crawley, R. Nair, B. Rajagoplan, and H. Sandick. A framework for QoS-based routing in the Internet. *IETF RFC 2386*, Aug. 1998. *(Cited on pages 11 and 13.)*

[36] J. D. Cryer. *Time Series Analysis*. Duxbury Press, Boston, USA, 1986. *(Cited on page 75.)*

[37] B. Davie, A. Charny, J. C. R. Bennett, K. Benson, J. Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis. An expedited forwarding PHB. *IETF RFC 3246*, Mar. 2002. *(Cited on pages 18 and 19.)*

[38] S. Deering and R. Hinden. Internet Protocol, version 6 (IPv6). *IETF RFC 2460*, Dec. 1998. *(Cited on page 16.)*

[39] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. In *Proc. ACM SIGCOMM 1989*, volume 19, pages 1–12, Austin,USA, Sep. 1989. *(Cited on page 20.)*

[40] C. Dovrolis and P. Ramanathan. A case for relative differentiated services and the proportional differentiation model. *IEEE Network*, 13(5):26–34, Sep. 1999. *(Cited on page 20.)*

[41] M. A. El-Gendy and K. G. Shin. Equation-based packet marking for assured forwarding services. In *Proc. IEEE INFOCOM 2002*, volume 2, pages 845–854, New York, USA, Jun. 2002. *(Cited on pages 24, 34 and 61.)*

[42] M. A. El-Gendy and K. G. Shin. Assured forwarding fairness using equation-based packet marking and packet separation. *Computer Networks*, 41(4):435–450, Mar. 2003. *(Cited on pages 24, 33 and 34.)*

[43] A. I. Elwalid and D. Mitra. Fluid models for the analysis and design of statistical multiplexing with loss priorities on multiple classes of bursty traffic. In *Proc. IEEE INFOCOM 1992*, volume 1, pages 415–425, Florence, Italy, May. 1992. *(Cited on page 40.)*

[44] R. F. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of UK inflation. *Econometrica*, 50(4):987–1007, Jul. 1982. *(Cited on page 76.)*

[45] ETSI Technical Report ETR003 second edition. *Network Aspects (NA); General Aspects of Quality of Service (QoS) and Network Performance (NP)*, Oct. 1994. *(Cited on page 12.)*

[46] W. Fang. *Differentiated Services: Architecture, Mechanisms and an Evaluation.* PhD thesis, Princeton University, Nov. 2000. *(Cited on pages 27 and 44.)*

[47] W. Fang, N. Seddigh, and B. Nandy. A time sliding window three colour marker (TSWTCM). *IETF RFC2859*, Jun. 2000. *(Cited on pages 23, 26, 28 and 50.)*

[48] W. Feng, D. D. Kandlur, D. Saha, and K. G. Shin. Adaptive packet marking for maintaining end-to-end throughput in a differentiated services Internet. *IEEE/ACM Transactions on Networking*, 7(5):685–697, 1999. *(Cited on pages 24 and 34.)*

[49] A. Feroz, S. Kalyanaraman, and A. Rao. A TCP-friendly traffic marker for IP differentiated services. In *Proc. 8th International Workshop on Quality of Service (IWQoS)*, pages 138–147, Jun. 2000. *(Cited on page 23.)*

[50] V. Firoiu and M. Borden. A study of active queue management for congestion control. In *Proc. IEEE INFOCOM 2000*, volume 3, pages 1435–1444, Tel Aviv, Israel, Mar. 2000. *(Cited on pages 34, 35, 36, 61 and 79.)*

[51] S. Floyd and K. Fall. Promoting the use of end-to-end congestion control in the Internet. *IEEE/ACM Transactions on Networking*, 7(4):458–472, Aug. 1999. *(Cited on pages 44 and 61.)*

[52] S. Floyd, R. Gummadi, and S. Shenker. Adaptive RED: An algorithm for increasing the robustness of RED's active queue management. Available at http://www.icir.org/foyd, 2001. *(Cited on page 32.)*

[53] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, Aug. 1993. *(Cited on pages 19 and 22.)*

[54] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and S. C. Diot. Packet-level traffic measurements from the Sprint IP backbone. *IEEE Network*, 17(6):6–16, Nov.-Dec. 2003. *(Cited on page 44.)*

[55] W. A. Gardner. *Introduction to Random Processes*. Macmillan Publ. Company, USA, 1986. *(Cited on pages 51, 52 and 53.)*

[56] M. Garetto, R. L. Cigno, M. Meo, and M. A. Marsan. Modeling short-lived TCP connections with open multiclass queuing networks. *Computer Networks*, 44(2):153–176, Feb. 2004. *(Cited on page 35.)*

[57] O. Gonzalez, C. M. Gauger, D. Sass, M. L. Garcia-Osma, A. J. Elizondo-Armengol, B. Puype, I. Lievens, P. Demeester, G. Retvari, P. Fodor, J. Tapolcai, and M. Düser. Traffic modelling and traffic engineering for next generation transport networks - results from the NOBEL project. In *Proc. 10th European Conference on Networks and Optical Communications (NOC 2005)*, pages 19–26, London, UK, Jul. 2005. *(Cited on page 44.)*

[58] M. Goyal, A. Durresi, P. Misra, C. Liu, and R. Jain. Effect of number of drop precedences in assured forwarding. In *Proc. IEEE GLOBECOM 1999*, volume 1, pages 188–193, Rio de Janeiro, Brasil, Nov. 1999. *(Cited on pages 33 and 34.)*

[59] J. Gozdecki, R. Stankiewicz, and A. Jajszczyk. Quality of service terminology in IP networks. *IEEE Communications Magazine*, 41(3):153–159, Mar. 2003. *(Cited on pages 5, 10 and 15.)*

[60] D. Gross and C. M. Harris. *Fundamentals of Queueing Theory*. John Wiley & Sons. Ltd, USA, 3rd edition, 1998. *(Cited on pages 36, 46, 47, 49, 57 and 59.)*

[61] D. Grossman. New terminology and clarifications for DiffServ. *IETF RFC 3260*, Apr. 2002. *(Cited on pages 15 and 16.)*

[62] U. C. Gupta and T. S. S. Srinivasa Rao. Computing steady state probabilities in $\lambda(n)$/G/1/K queue. *Performance Evaluation*, 24(4):265–275, Feb. 1996. *(Cited on page 60.)*

[63] U. C. Gupta and T. S. S. Srinivasa Rao. On the analysis of single server finite queue with state dependent arrival and service processes: M(n)/G(n)/1/K. *OR Spectrum*, 20(2):83–89, Jun. 1998. *(Cited on pages 57 and 60.)*

[64] A. Habib, B. Bhargava, and S. Fahmy. A round trip time and time-out aware traffic conditioner for differentiated services networks. In *Proc. IEEE ICC 2002*, volume 2, pages 981–985, New York, USA, Apr. 2002. *(Cited on pages 24 and 44.)*

[65] W. C. Hardy. *"QoS" Measurement and Evaluation of Telecommunications Quality of Service*. John Willey & Sons. Ltd, England, 2001. *(Cited on page 10.)*

[66] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured forwarding PHB group. *IETF RFC 2597*, Jun. 1999. *(Cited on pages 4, 18 and 19.)*

[67] J. Heinanen, T. Finland, and R. Guerin. A two rate three color marker. *IETF RFC2698*, Sep. 1999. *(Cited on pages 23, 26 and 48.)*

[68] J. Heinanen and R. Guerin. A single rate three color marker. *IETF RFC2697*, Sep. 1999. *(Cited on pages 23, 25 and 47.)*

[69] S. Herrería-Alonso, A. Suárez-González, M. Fernández-Veiga, R. F. Rodríguez-Rubio, and C. López-García. Improving aggregate flow control in differentiated services networks. *Computer Networks*, 44(4):499–512, Mar. 2004. *(Cited on page 34.)*

[70] N. Hohn, D. Veitch, and P. Abry. Cluster processes, a natural language for network traffic. *IEEE Transactions on Signal Processing*, 51(8):2229–2244, Aug. 2003. *(Cited on page 44.)*

[71] C. V. Hollot, V. Misra, D. Towsley, and W-B. Gong. A control theoretic analysis of RED. In *Proc. IEEE INFOCOM 2001*, volume 3, pages 1510–1519, Anchorage, USA, Apr. 2001. *(Cited on pages 34, 35 and 36.)*

[72] ITU-T Recommendation E.417. *Framework for the Network Management of IP-Based Networks*, Feb. 2001. *(Cited on page 13.)*

[73] ITU-T Recommendation E.800. *Terms and Definitions Related to Quality of Service and Network Performance Including Dependability*, Aug. 1993. *(Cited on page 12.)*

[74] ITU-T Recommendation G.1000. *Communications Quality of Service: A Framework and Definitions*, Nov. 2001. *(Cited on page 12.)*

[75] ITU-T Recommendation I.350. *General Aspects of Quality of Service and Network Performance in Digital Networks, Including ISDNs*, Mar. 1993. *(Cited on page 12.)*

[76] ITU-T Recommendation Y.1221. *Traffic Control and Congestion Control in IP Based Networks*, Mar. 2002. *(Cited on page 14.)*

[77] ITU-T Recommendation Y.1241. *Support of IP-based Services Using IP Transfer Capabilities*, Mar. 2001. *(Cited on page 14.)*

[78] ITU-T Recommendation Y.1541. *Internet Protocol Data Communication Service - IP Packet Transfer and Availability Performance Parameters*, Feb. 1999. *(Cited on page 12.)*

[79] V. B. Iversen. *Teletraffic Engineering Handbook*. ITC and ITU-D SG 2/16, Sep. 2002. *(Cited on page 49.)*

[80] T. Karagiannis, A. Broido, N. Brownlee, K. C. Claffy, and M. Faloutsos. Is p2p dying or just hiding. In *Proc. IEEE GLOBECOM 2004*, volume 3, pages 1532–1538, Dallas, Texas, Dec. 2004. *(Cited on page 61.)*

[81] J. Keilson. The ergodic queue length distribution for queueing systems with finite capacity. *Journal of the Royal Statistical Society*, 28(1):190–201, 1966. *(Cited on page 49.)*

[82] S. Keshaw. *An Engineering Approach to Computer Networking*. Addison-Wesley, 1997. *(Cited on page 20.)*

[83] A. Kesselman and Y. Mansour. Loss-bounded analysis for differentiated services. *Journal of Algorithms*, 46(1):79–95, Jan. 2003. *(Cited on page 35.)*

[84] H. Kim and D. J. Farber. The failure of conservative congestion control in large bandwidth-delay product networks. In *Proc. Internet Society, Internet Global Summit (INET)*, Honolulu, Hawai, USA, Jun. 1995. *(Cited on page 44.)*

[85] H. Kim, C. Yoo, and W.Y. Jung. Simulation study on the effect of the trTCM parameters. In *Proc. 10th IEEE International Conference on Telecommunications*, volume 2, pages 1482–1488, Tahiti (French Polynesia), Feb. 2003. *(Cited on page 33.)*

[86] M. Kisimoto, H. Ohsaki, and M. Murata. On transient behavior analysis of random early detection gateway using a control theoretic approach. In *Proc. IEEE Control Systems Society Conference on Control Applications (*CCA/CACSD *2002)*, volume 2, pages 1144–1149, Sep. 2002. *(Cited on pages 34 and 35.)*

[87] L. Kleinrock. *Queueing Systems, Volume 1: Theory.* John Wiley & Sons. Ltd, USA, 1975. *(Cited on page 48.)*

[88] K. R. R. Kumar, A. L. Ananda, and L. Jacob. TCP friendly traffic conditioning in DiffServ networks: A memory-based approach. *Computer Networks*, 38(6):731–743, Apr. 2002. *(Cited on pages 24 and 33.)*

[89] E. Kuumola, J. Resing, and J. Virtamo. Analytical results on the stochastic behaviour of an averaged queue length. In *Proc. 16th Nordic Teletraffic Seminar, NTS 16*, pages 88–99, Aug. 2002. *(Cited on page 55.)*

[90] E. Kuumola, J. Resing, and J. Virtamo. Joint distribution of instantaneous and averaged queue length in an M/M/1/K system. In *Proc. 15th Teletraffic Congress Specialist Seminar "Internet Traffic Engineering and Traffic Management"*, pages 58–67, Würzburg, Germany, Jul. 2002. *(Cited on page 55.)*

[91] P. Kuusela, P. Lassila, and J. Virtamo. Stability of TCP-RED congestion control. In *Proc. 17th International Teletraffic Congress (ITC)*, pages 655–666, Salvador da Bahia, Brasil, Sep. 2001. *(Cited on pages 34, 35 and 37.)*

[92] P. Kuusela, P. Lassila, J. Virtamo, and P. Key. Modeling RED with idealized TCP sources. In *Proc. 9th IFIP Conference on Performance Modelling and Evaluation of ATM & IP Networks*, pages 155–166, Budapest, Hungary, Jun. 2001. *(Cited on pages 37, 44, 49 and 61.)*

[93] T. V. Lakshman and U. Madhow. The performance of TCP/IP for networks with high bandwidth-delay products and random loss. *IEEE/ACM Transactions on Networking*, 5(3):336–350, Jun. 1997. *(Cited on page 61.)*

[94] P. L'Ecuyer. Good parameters and implementations for combined multiple recursive random number generators. *Operations Research*, 47(1):159–164, Jan.-Feb. 1999. *(Cited on page 72.)*

[95] P. L'Ecuyer and R. Couture. An implementation of the lattice and spectral tests for multiple recursive linear random number generators. *INFORMS Journal on Computing*, 9(2):206–217, 1997. *(Cited on page 73.)*

[96] 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 73.)*

[97] W. Lin, R. Zheng, and J. Hou. How to make assured services more assured. In *Proc. IEEE 7th International Conference on Network Protocols (ICNP)*, pages 182–191, Toronto, Canada, Oct. 1999. *(Cited on pages 23, 32 and 33.)*

[98] W. Liu, Z. Yang, J. He, and C. T. Chou. Two adaptive AQM algorithms for quantitative differentiated services. In *Proc. IEEE GLOBECOM 2004*, volume 3, pages 1703– 1707, Dallas, Texas, Dec. 2004. *(Cited on page 32.)*

[99] G. M. Ljung and G. E. P. Box. On a measure of lack of fit in time series models. *Biometrika*, 65(2):297–303, Aug. 1978. *(Cited on page 75.)*

[100] R. Makkar, I. Lambadaris, J. H. Salim, N. Seddigh, B. Nandy, and J. Babiarz. Empirical study of buffer management scheme for DiffServ Assured Forwarding PHB. In *Proc. 9th International Conference on Computer Communications and Networks*, pages 632–637, Las Vegas, USA, Oct. 2000. *(Cited on pages 19, 29, 34 and 38.)*

[101] N. Malouch and Z. Liu. On steady state analysis of TCP in networks with differentiated services. In *Proc. 17th International Teletraffic Congress (ITC)*, Salvador da Bahia, Brasil, Sep. 2001. *(Cited on pages 33, 34, 35, 38, 39 and 61.)*

[102] N. Malouch and Z. Liu. Performance analysis of TCP with RIO routers. In *Proc. IEEE GLOBECOM 2002*, volume 2, pages 1633–1637, Taipei, Taiwan, Nov. 2002. *(Cited on pages 33, 34, 35, 38, 44, 56 and 61.)*

[103] N. Malouch and Z. Liu. Performance evaluation of RIO routers. Technical Report RR-4469, INRIA Sophia-Antipolis Research Unit, May 2002. *(Cited on pages 33, 34, 35 and 38.)*

[104] M. Mathis, J. Semke, J. Mahdavi, and T. Ott. Macroscopic behavior of the TCP congestion avoidance algorithm. *Computer Communication Review*, 27(3):67–82, Jul. 1997. *(Cited on pages 34 and 35.)*

[105] M. May, J. Bolot, A. Jean-Marie, and C. Diot. Simple performance models of differentiated services schemes for the Internet. In *Proc. IEEE INFOCOM 1999*, volume 3, pages 1389–1394, New York, USA, Mar. 1999. *(Cited on pages 33, 34, 35, 37 and 44.)*

[106] S. McCreary and K. Claffy. Trends in wide area IP traffic patterns: A view from Ames Internet Exchange. In *Proc. ITC Specialist Seminar on IP Traffic Measurement, Modeling, and Management*, Monterey, California, USA, Sep. 2000. *(Cited on page 44.)*

[107] A. I. McLeod and W. K. Li. Diagnostic checking ARMA time series models using squared-residual autocorrelations. *Journal of Time Series Analysis*, 4(4):269–273, 1983. *(Cited on page 76.)*

[108] M. Mellia, I. Stoica, and H. Zhang. TCP model for short lived flows. *IEEE Communications Letters*, 6(2):85–87, Feb. 2002. *(Cited on page 35.)*

[109] A. Misra, T. Ott, and J. Baras. The window distribution of multiple TCPs with random loss queues. In *Proc. IEEE GLOBECOM 1999*, volume 3, pages 1714–1726, Rio de Janeiro, Brasil, Nov. 1999. *(Cited on pages 34, 35 and 36.)*

[110] V. Misra, W. Gong, and D. Towsley. Stochastic differential equation modeling and analysis of TCP window size behavior. Technical Report ECE-TR-CCS-99-10-01, Dept. of Electrical and Computer Engineering, Univ. of Massachusetts, Oct. 1999. *(Cited on pages 34, 36 and 37.)*

[111] V. Misra, W-B. Gong, and D. Towsley. Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. In *Proc. ACM SIGCOMM 2000*, volume 30, pages 151–160, Stockholm, Sweden, Aug. 2000. *(Cited on pages 34, 36, 37 and 44.)*

[112] D. Mitra. Stochastic theory of a fluid model of producers and consumers coupled by a buffer. *Advances in Applied Probability*, 20(3):646–676, Sep. 1988. *(Cited on page 40.)*

[113] B. Nandy, N. Seddigh, P. Pieda, and J. Ethridge. Intelligent traffic conditioners for assured forwarding based differentiated services networks. In *Proc. IFIP High Performance Networking*, pages 540–554, Paris, France, May 2000. *(Cited on page 23.)*

[114] Network simulator ns-2. Available at http://nsnam.isi.edu/nsnam. *(Cited on page 68.)*

[115] 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*, Dec. 1998. *(Cited on pages 16, 18 and 20.)*

[116] K. Nichols and B. Carpenter. Definition of differentiated services per domain behaviors and rules for their specification. *IETF RFC 3086*, Apr. 2001. *(Cited on page 17.)*

[117] K. Nichols, V. Jacobson, and L. Zhang. A two-bit differentiated services architecture for the Internet. *IETF RFC 2638*, Jul. 1999. *(Cited on pages 18 and 19.)*

[118] A. M. Odlyzko. Internet traffic growth: Sources and implications. In B. B. Dingel, W. Weiershausen, A. K. Dutta, and K.-I. Sato, editors, *Proc. Optical Transmission Systems and Equipment for WDM Networking II*, volume 5247 of *SPIE Proceedings Series*, pages 1–15, Sep. 2003. *(Cited on page 61.)*

[119] H. Ohsaki, M. Murata, and H. Miyahara. Steady state analysis of the RED gateway: stability, transient behavior, and parameter setting. *IEICE Transactions on Communications*, E85-B(1):107–115, Jan. 2002. *(Cited on pages 34, 35, 36 and 61.)*

[120] J. Olsén. *Stochastic Modeling and Simulation of the TCP protocol.* PhD thesis, Uppsala University, Department of Mathematics, Sweden, Oct. 2003. *(Cited on page 35.)*

[121] J. Orozco and D. Ros. An adaptive RIO (A-RIO) queue management algorithm. In G. Karlsson and M. I. Smirnov, editors, *Proc. 4th International Workshop on Quality of Future Internet Services (QoFIS)*, volume 2811 of *Lecture Notes In Computer Science*, pages 11–20, Stockholm, Sweden, Oct. 2003. Springer-Verlag. *(Cited on page 32.)*

[122] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP throughput: A simple model and its empirical validation. In *Proc. ACM SIGCOMM 1998*, volume 28, pages 303–314, Vancouver, Canada, May 1998. *(Cited on pages 34, 35, 36, 37, 38 and 61.)*

[123] E.-C. Park and C.-H. Choi. Adaptive token bucket algorithm for fair bandwidth allocation in DiffServ networks. In *Proc. IEEE GLOBECOM 2003*, volume 6, pages 3176–3180, San Francisco, USA, Dec. 2003. *(Cited on page 23.)*

[124] K. Pawlikowski. Steady-state simulation of queueing processes: A survey of problems and solutions. *ACM Computing Surveys*, 22(2):123–170, Jun. 1990. *(Cited on pages 69 and 73.)*

[125] 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, Jan. 2002. *(Cited on page 73.)*

[126] V. Paxson and S. Floyd. Wide-area traffic: The failure of Poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, Jun. 1995. *(Cited on page 44.)*

[127] P. Pieda, N. Seddigh, and B. Nandy. The dynamics of TCP and UDP interaction in IP-QoS differentiated services networks. In *Proc. 3rd Canadian Conference on Broadband Research*, Ottawa, Canada, Nov. 1999. *(Cited on pages 33 and 34.)*

[128] J. Postel. Internet Protocol. *IETF RFC 791*, Sep. 1981. *(Cited on page 16.)*

[129] J. Roberts. Flow aware networking for effective quality of service control. In *Proc. IMA Workshop on Scaling Phenomena in Communications Networks*, Minneapolis, USA, Oct. 1999. *(Cited on page 4.)*

[130] J. Roberts and S. Oueslati-Boulahia. Quality of service by flow aware networking. *Philosophical Transactions of Royal Society*, 358(1773):2197–2207, Aug. 2000. *(Cited on page 4.)*

[131] S. W. Roberts. Control chart tests based on geometric moving average. *Technometrics*, 1:239–250, 1959. *(Cited on page 26.)*

[132] S. Sahu, P. Nain, D. Towsley, C. Diot, and V. Firoiu. On achievable service differentiation with token bucket marking for TCP. In *Proc. ACM SIGMETRICS 2000*, volume 28, pages 23–33, Santa Clara, USA, Jun. 2000. *(Cited on pages 33, 34, 39, 44 and 79.)*

[133] S. Sahu, D. Towsley, and J. Kurose. A quantitative study of differentiated services for the Internet. In *Proc. IEEE GLOBECOM 1999*, volume 3, pages 1808–1817, Rio De Janiero, Brazil, Dec. 1999. *(Cited on pages 34, 40 and 44.)*

[134] J. B. Schmitt. *Heterogeneous Network Quality of Service Systems*. Kluwer Academic Publishers, USA, 2001. *(Cited on page 18.)*

[135] N. Seddigh, B. Nandy, and P. Pieda. Bandwidth assurance issues for TCP flows in a differentiated services networks. In *Proc. IEEE GLOBECOM 1999*, volume 3, pages 1792–1798, Rio de Janeiro, Brasil, Nov. 1999. *(Cited on pages 28, 33, 34 and 61.)*

[136] N. Seddigh, B. Nandy, P. Pieda, J. H. Salim, and A. Chapman. An experimental study of assured services in a DiffServ IP QoS network. In R. O. Onvural, S. Civanlar, P. J. Doolan, and J. V. Luciani, editors, *Proc. SPIE, Internet Routing and Quality of Service*, volume 3529, pages 217–230, Dec. 1998. *(Cited on page 33.)*

[137] S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611, Dec. 1965. *(Cited on page 76.)*

[138] M. Shreedhar and G. Varghese. Efficient fair queuing using deficit round-robin. *IEEE/ACM Transactions on Networking*, 4(3):375–385, Jun. 1996. *(Cited on page 20.)*

[139] V. A. Siris and H. M. Marinakis. Adaptive packet marking for achieving fairness in DiffServ networks. *Computer Communications*, 29(1):52–58, Dec. 2005. *(Cited on pages 24, 33 and 44.)*

[140] R. Stankiewicz and A. Jajszczyk. Modeling of TCP behavior in a DiffServ network supporting Assured Forwarding PHB. In *Proc. IEEE ICC 2004*, volume 4, pages 2071–2075, Paris, France, Jun. 2004. *(Cited on pages 5, 6, 57 and 62.)*

[141] R. Stankiewicz and A. Jajszczyk. Analytical models for DiffServ meter/markers. In *Proc. IEEE ICC 2005*, volume 1, pages 158–162, Seoul, Korea, May 2005. *(Cited on pages 5, 6, 34, 48, 50 and 96.)*

[142] R. Stankiewicz and A. Jajszczyk. New mathematical models for token bucket based meter/markers. In G. Parr, D. Malone, and M. Ó. Fohlú, editors, *Proc. 6th IEEE International Workshop on IP Operations and Management (IPOM)*, volume 4268 of *Lecture Notes In Computer Science*, pages 96–107, Dublin, Ireland, Oct. 2006. Springer-Verlag. *(Cited on pages 5, 6, 46, 48, 80 and 85.)*

[143] A. Stepanenko, C. C. Constantinou, T. N. Arvanitis, and K. Baughan. Statistical properties of core network Internet traffic. *IEE Electronics Letters*, 38(7):350–351, Mar. 2002. *(Cited on page 43.)*

[144] W. Stevens. TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms. *IETF RFC 2001*, Jan. 1997. *(Cited on page 79.)*

[145] H. Su and M. Atiquzzaman. ItswTCM: A new aggregate marker to improve fairness in DiffServ. In *Proc. IEEE GLOBECOM 2001*, volume 3, pages 1841–1846, San Antonio, USA, Nov. 2001. *(Cited on pages 23, 33, 34 and 53.)*

[146] H. Su and M. Atiquzzaman. Comprehensive performance model of differentiated service with token bucket marker. *IEE Proceedings-Communications*, 150(5):347–353, Oct. 2003. *(Cited on pages 33, 34, 38, 44 and 61.)*

[147] S. Tartarelli and A. Banchs. Random early marking: Improving TCP performance in DiffServ assured forwarding. In *Proc. IEEE ICC 2002*, volume 2, pages 970–975, New York, USA, Apr. 2002. *(Cited on pages 33 and 34.)*

[148] K. Thompson, G. Miller, and R. Wilder. Wide area Internet traffic patterns and characteristics. *IEEE Network*, 11(6):10–23, Nov.-Dec. 1997. *(Cited on page 44.)*

[149] J. Tyszer. *Object-Oriented Computer Simulation of Discrete-Event Systems*. Kluwer Academic Publishers, USA, 1999. *(Cited on pages 68, 69, 73, 74 and 139.)*

[150] S. Vegesna. IP *Quality of Service*. Cisco Press, USA, 2001. *(Cited on pages 20 and 30.)*

[151] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S-Plus*. Springer-Verlag, 1994. *(Cited on page 75.)*

[152] K. Wajda and K. Zaraska. Low-cost measurement concept for Ethernet traffic. In *Proc. 46th IEEE International Symposium ELMAR 2004*, Zadar, Croatia, Jun. 2004. *(Cited on page 44.)*

[153] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson. Self similarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level. *IEEE/ACM Transactions on Networking*, 5(1):71–86, Feb. 1997. *(Cited on page 44.)*

[154] S. Wolfram. *The Mathematica Book*. Wolfram Media and Cambridge University Press, 4th edition, 1999. *(Cited on page 68.)*

[155] M. H. Yaghmaee, M. B. Menhaj, and H. Amintoosi. Design and performance evaluation of a fuzzy based traffic conditioner for differentiated services. *Computer Networks*, 47(6):847–869, Apr. 2005. *(Cited on pages 32 and 33.)*

[156] I. Yeom and A. L. N. Reddy. Adaptive marking for aggregated flows. In *Proc. IEEE GLOBECOM 2001*, volume 4, pages 2263–2267, San Antonio, USA, Nov. 2001. *(Cited on page 24.)*

[157] I. Yeom and A. L. N. Reddy. Modeling TCP behavior in a differentiated services network. *IEEE/ACM Transactions on Networking*, 9(1):31–46, Feb. 2001. *(Cited on pages 33, 34 and 39.)*

[158] T. Ziegler, C. Brandauer, and S. Fdida. A quantitative model for the parameter setting of RED with TCP traffic. In L. Wolf, D. Hutchinson, and R. Steinmetz, editors, *Proc. 9th International Workshop on Quality of Service (IWQoS)*, volume 2092 of *Lecture Notes In Computer Science*, pages 202–216, Karlsruhe, Germany, Jun. 2001. Springer-Verlag. *(Cited on pages 34, 37 and 64.)*

[159] T. Ziegler, S. Fdida, C. Brandauer, and B. Hechenleitner. Stability of RED with two-way TCP traffic. In *Proc. IEEE 9th International Conference on Computer Communications and Networks*, pages 214–219, Las Vegas, USA, Oct. 2000. *(Cited on pages 37 and 44.)*

Index