

**TKN**

Telecommunication  
Networks Group



Technical University Berlin

Telecommunication Networks Group

---

Fuzzy Explicit Window Adaptation:  
Using Router Feedback to Improve  
TCP Performance

Michael Savorić

savoric@ee.tu-berlin.de

Berlin, July 2004

TKN Technical Report TKN-04-009

---

TKN Technical Reports Series  
Editor: Prof. Dr.-Ing. Adam Wolisz

## **Abstract**

In the current Internet, standard TCP congestion control is based on segment loss events possibly assisted by simple explicit router feedback mechanisms like Explicit Congestion Notification (ECN). The performance of TCP congestion control can be improved if a more sophisticated Router Congestion Feedback (RCF) mechanism than ECN is used. One of such approaches is Explicit Window Adaptation (EWA). EWA uses TCP's built-in flow control mechanism to inform TCP senders about the current load in a network path. To reach this, an EWA-capable router is able to decrease the advertised receiver window in TCP acknowledgments if necessary in order to avoid congestion and packet losses. Since the EWA algorithm performed in a router has some shortcomings under certain conditions, a new EWA-related approach is developed. This approach, called fuzzy explicit window adaptation (FEWA), replaces parts of the EWA algorithm in a router with a more accurate fuzzy-based calculation.

In this technical report, FEWA is described in detail. Using two simulation scenarios with different load in the network, the performance of standard TCP is compared with the performances of standard TCP assisted by EWA and standard TCP assisted by FEWA. The simulation results show that standard TCP assisted by FEWA outperform both other approaches in throughput over loaded routers and considerably decreases the number of packet losses in congested routers.

**Keywords:** TCP, congestion control, flow control, distributed network information sharing, fuzzy control

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Explicit Window Adaptation (EWA)</b>	<b>5</b>
2.1	EWA Algorithm . . . . .	5
2.2	Applicability of EWA . . . . .	6
2.3	Shortcomings of EWA . . . . .	6
2.4	Improvements of EWA . . . . .	7
<b>3</b>	<b>Fuzzy Explicit Window Adaptation (FEWA)</b>	<b>8</b>
3.1	FEWA Algorithm . . . . .	8
3.1.1	General Fuzzy Controller . . . . .	9
3.1.2	FEWA Fuzzy Controller . . . . .	10
3.2	Practicability of FEWA . . . . .	12
<b>4</b>	<b>Simulation Model</b>	<b>14</b>
4.1	Network Topology . . . . .	14
4.2	Location of RCF-Capable Routers . . . . .	16
4.3	Traffic Load Models . . . . .	16
4.4	Implementation Details . . . . .	17
<b>5</b>	<b>Performance Evaluation Metrics</b>	<b>19</b>
5.1	Throughput . . . . .	19
5.2	Queue Loss Rate . . . . .	20
<b>6</b>	<b>Performance Evaluation of EWA and FEWA by Simulations</b>	<b>21</b>
6.1	Introduction . . . . .	21
6.2	Scenario 1: TCP NewReno, Lower Mean Network Load . . . . .	23
6.3	Scenario 2: TCP NewReno, Higher Mean Network Load . . . . .	25

6.4	Queue Lengths in Bottleneck Base Stations . . . . .	27
<b>7</b>	<b>Conclusion and Outlook</b>	<b>31</b>
<b>A</b>	<b>Transient Analysis of the Simulations</b>	<b>33</b>
A.1	Transient Analysis Method . . . . .	33
A.2	Transient Analysis Results . . . . .	35
<b>B</b>	<b>Appendix: Statistical Evaluation</b>	<b>38</b>
B.1	Statistical Evaluation Method . . . . .	38
B.2	Statistical Evaluation Results for EWA and FEWA Simulations . . . . .	38
<b>C</b>	<b>Parameters of Fuzzy Explicit Window Adaptation (FEWA)</b>	<b>42</b>
C.1	Linguistic Rules of FEWA . . . . .	42
C.2	Parameters of FEWA . . . . .	45
C.3	Selection of $\alpha_i$ 's for Different Maximum Queue Lengths . . . . .	46
C.3.1	Example: Persistent and WWW Traffic Traversing a Single Bottleneck Router	47

# Chapter 1

## Introduction

In today's Internet, standard TCP connections perform congestion control triggered by segment loss events, e.g., a retransmission timer timeout or the reception of duplicated acknowledgments (dupacks). In addition, explicit feedback mechanisms from routers, e.g., Explicit Congestion Notification (ECN) [15] as a very simple one-bit-information (no congestion/congestion) mechanism, can be used to assist TCP's congestion control by informing TCP senders early enough about (impending) congestion in a router. In this context, it is interesting to investigate other explicit feedback mechanisms from routers that are able to provide TCP senders with more information about the current congestion situation in the network. One of such approaches is the Explicit Window Adaptation (EWA) [10] which is based on TCP's built-in flow control. EWA uses the advertised receiver window in TCP acknowledgments to provide TCP senders with the information which sending window can be currently supported by the network in order to avoid congestion in the network. Although EWA is able to improve the performance of TCP connections [10] it has some shortcomings in cases where a router is lowly loaded for a longer period of time. Therefore, I have adapted the basic EWA algorithm by replacing parts of it with a more appropriate algorithm based on the fuzzy control theory. This new EWA-related algorithm is called Fuzzy Explicit Window Adaptation (FEWA).

The remainder of this technical report is organized as follows. Chapter 2 briefly describes the EWA mechanism and pictures some shortcomings of parts of the EWA algorithm in a router. An improvement of the EWA algorithm, called Fuzzy Explicit Window Adaptation (FEWA), based on a fuzzy congestion controller is explained in Chapter 3. The simulation model used for the performance evaluation of standard TCP, standard TCP assisted by EWA, and standard TCP assisted by FEWA is described in Chapter 4. And the performance metrics used in the performance evaluation are described in Chapter 5. Chapter 6 contains the results of this performance evaluation. Finally, Chapter 7 concludes this technical report and gives an outlook to my future activities in this research area. Appendix A shows the results of the transient analysis of the simulations. Appendix B contains

the statistical evaluation of the simulation results. In Appendix C, the linguistic rules and the parameter values of the FEWA fuzzy controller are depicted. In addition, a rule of thumb for the calculation of the  $\alpha_i$ 's of the FEWA fuzzy controller for different maximum queue lengths is given.

## Chapter 2

# Explicit Window Adaptation (EWA)

The EWA approach [10] has been developed to explicitly inform the senders of TCP connections about the currently available bandwidth over a bottleneck link in a transparent way; the source codes of a TCP sender and a TCP receiver are kept unchanged. In this chapter, I describe the basic algorithm of EWA, show how EWA can be deployed in the current Internet environment, explain some shortcomings of EWA, and depict some possible improvements of EWA.

### 2.1 EWA Algorithm

After every measurement interval  $i$  with a duration of 10 ms, a router with EWA-capabilities measures its current queue length  $Q_i$  and computes the current mean queue length  $\bar{Q}_i$ .  $Q_i$ ,  $\bar{Q}_i$ , and the second-last computed mean queue length  $\bar{Q}_{i-1}$  are then used to calculate a new sending window for each TCP connection traversing this router:

$$\text{sending window} = \max\{\text{MSS}, \alpha \cdot \log_2(B - Q_i) \cdot \text{MSS}\} \quad (2.1)$$

where  $B$  is the maximum queue length in the router (i.e., at the same time at most  $B + 1$  packets can be stored and forwarded in the router), MSS is the maximum segment size, and  $\alpha$  is a dynamically determined factor whose calculation is later explained.  $B$  and  $Q_i$  are expressed in number of packets and MSS is expressed in number of bytes. The logarithmic expression in Equation (2.1) is introduced to reflect that TCP connections which are in slow start are able to send twice as much segments in their next round trip time (RTT) interval than now, i.e., the queue length can exponentially grow in the near future. In addition, Equation (2.1) ensures that every TCP connection is always allowed to send at least one TCP segment with a MSS.

The alterable factor  $\alpha$  in Equation (2.1) is introduced to better utilize the link if only a few TCP

connections are transferring segments over the router.  $\alpha$  is updated every 10 milliseconds as follows:

$$\alpha = f(\alpha, \bar{Q}_i) = \begin{cases} \alpha + w_{\text{up}} & \text{if } \bar{Q}_i < \text{threshold}_{\text{low}} \\ \alpha \cdot w_{\text{down}} & \text{if } \bar{Q}_i > \text{threshold}_{\text{high}} \end{cases} \quad (2.2)$$

with

$$\bar{Q}_i = \frac{127}{128} \cdot \bar{Q}_{i-1} + \frac{1}{128} \cdot Q_i \quad (2.3)$$

The initial value for the utilization factor  $\alpha$  is set to 1, the parameters  $w_{\text{up}}$  to additively increase and  $w_{\text{down}}$  to multiplicatively decrease  $\alpha$  are set to  $1/8$  and  $31/32$ , and the low and high thresholds for the mean queue length are set to 20 % and 60 % of the maximum queue length  $B$ .

The calculated sending window is transferred to each TCP sender by modifying the advertized receiver window in the TCP acknowledgments. It is only *reduced* by the EWA-capable router, but never increased:

$$\text{advertized receiver window} = \min\{\text{sending window}, \text{advertized receiver window}\} \quad (2.4)$$

With this explicit congestion feedback information, the TCP senders are able to react more adequately to the current load in the router than it is possible with other mechanisms, e.g., ECN or RED.

## 2.2 Applicability of EWA

To deploy EWA in a network it is not necessary to equip all routers in the network with EWA-capabilities. It is sufficient to provide only the bottleneck router(s) with EWA.

EWA requires that the segments and acknowledgments of the considered TCP connections pass through the same (bottleneck) routers. If this necessary condition of symmetrical routing can be guaranteed for all (bottleneck) routers in the path from a TCP sender to its TCP receiver, such an approach can be used for an improved end-to-end congestion control based on a flow control between the (bottleneck) routers and the TCP senders.

## 2.3 Shortcomings of EWA

The basic EWA algorithm described in [10] does not consider the realistic case that the queue of a router with EWA-capabilities is below the lower threshold for a longer period of time. In this case, the utilization factor  $\alpha$  perpetually increases without an upper bound. If, after a while, the formerly lowly loaded router is highly loaded due to a large burst of incoming packets, it takes dozens or hundreds of measurement intervals to decrease the utilization factor  $\alpha$  to an appropriate value. In the meantime, the EWA algorithm is not able to prevent the router from congestion and packets get lost. Thus, the history of the load of the router is too heavily weighted in the calculation of the utilization factor  $\alpha$

of the EWA algorithm. This drawback of the EWA algorithm can be softened but not completely prevented if an upper bound for the utilization factor  $\alpha$  is introduced, if the current queue length  $Q_i$  is much higher weighted in Equation (2.3), or if the parameter  $w_{\text{down}}$  is decreased to a value  $\ll 1$ . But it can be only prevented if a modified calculation of the utilization factor  $\alpha$  with a much faster reaction on the current load in the router is used.

## 2.4 Improvements of EWA

The congestion feedback function (2.1) of EWA is one example function to calculate the sending window in a router. Other feedback functions are conceivable. These new feedback functions should adopt the well-chosen logarithmic expression from the EWA feedback function, but can adapt or replace the calculation of the utilization factor  $\alpha$ . The FEWA algorithm described in Section 3 is an example approach for the latter case.

EWA and related approaches can be easily adapted so that symmetrical routing is no longer a necessary condition for its usage. This can be done by introducing a new TCP header option that carries the calculated sending window of the routers in the path from the TCP sender to the TCP receiver. The TCP receiver puts the minimum of this sending window and its currently supported receiver window as the new advertised receiver window in a normal TCP acknowledgment and sends it to the TCP sender. For this variant of EWA the TCP in the receiving end systems have to be slightly changed. Although this new implementation of EWA increases the delay of the EWA control loop between the routers and the TCP senders, it is not certain that this will decrease the overall performance of EWA. A positive effect of this new implementation on the performance of EWA is that the sending window of each EWA-capable router is calculated with a higher probability during the period of time the router has to handle segments from those TCP connections which get the feedback information. This is due to the inherent bursty sending behavior of the window-based congestion control of TCP (and in contrast to rate-based congestion-control mechanisms used in other networks, e.g., in ATM networks [22, 2, 8, 17, 5]). As a result, with the new EWA implementation the correlation between the current sending window and the obtained feedback might be increased from a single TCP sender's point of view. Whether the negative or positive effect of this new EWA implementation dominates the overall performance of EWA cannot be answered a priori. This has to be carefully investigated.

## Chapter 3

# Fuzzy Explicit Window Adaptation (FEWA)

In this chapter, I describe a new EWA-related approach, called FEWA, whose queue-operating point is changed and whose calculation of the utilization factor  $\alpha$  is based on a fuzzy controller.

### 3.1 FEWA Algorithm

The FEWA algorithm is similar to the EWA algorithm with two exceptions:

- (1) The sending window (cf. Equation (2.1)) is calculated as follows:

$$\text{sending window} = \max\{\text{MSS}, \text{round}(\alpha \cdot \log_2(B - Q_i)) \cdot \text{MSS}\} \quad (3.1)$$

- (2) And the utilization factor  $\alpha$  (cf. Equation (2.2)) is calculated by a fuzzy controller only dependent on the current and the last but one measured queue length in a router:

$$\alpha = f(Q_i, Q_{i-1}) \quad (3.2)$$

Similar to EWA, the time of a control interval  $i$  of FEWA is set to 10 ms. But this value has to be adapted dependent on the bandwidths in the links the FEWA-capable router is connected with.

This fuzzy controller is related to the fuzzy controllers used in the Fuzzy Explicit Rate Marking Adaptation (FERMA) [19, 21] and in the more conservative FERMA-Modification (FERMAM) [20]. These approaches are located in Asynchronous Transfer Mode (ATM) switches and calculate congestion feedback which is used for the congestion control of Available Bit Rate (ABR) connections in ATM networks [8, 17, 5]. All linguistic rules and the membership functions of FERMAM are reused for FEWA and only a few parameters of the FERMAM fuzzy controller are adapted for FEWA.

### 3.1.1 General Fuzzy Controller

The most important part of a fuzzy control system is the fuzzy logic controller (FLC) [12, 13, 14]. With the FLC the human train of thoughts can be adapted in a simple way by using linguistic variables with their set of linguistic values and a small number of linguistic rules or relational expressions. One of the linguistic rules, linguistic rule R5, in the nonlinear FEWA fuzzy congestion controller (FCC) is for example (see Appendix C.1):

*If the queue (length) is short and  
the rate of change is increasing slowly,  
then the utilization factor should be high.*

(R5)

“queue (length)”, “rate of change”, and “utilization factor” are called linguistic variables; “short”, “increasing slowly”, and “high” are examples of their valid linguistic values. “queue (length)” and “rate of change” are the measured input variables of the linguistic rule while “utilization factor” is the output or control variable set by this linguistic rule.

The design of a FLC is split into two parts: First, the linguistic rules are set (surface structure), and second, the membership functions of the linguistic variables are determined (deep structure), quite often by a more intuitive and pragmatic choice.

Depending on the input parameter  $x$  the membership function  $m_V$  of a linguistic variable  $V$  denotes the weights  $w_{v_k} \in [0, 1]$  of the valid linguistic values  $v_k$ ,  $1 \leq k \leq n_V$ :

$$m_V(x) = (w_{v_1}, \dots, w_{v_{n_V}}) \in [0, 1]^{n_V} \quad (3.3)$$

Here, each membership function  $m_V$  is represented by a composition of  $n_V$  membership functions of the fuzzy sets  $F_{v_k}$  of its corresponding linguistic values  $v_k$ :

$$\begin{aligned} m_V(x) &= m_{v_1}(x) \otimes \dots \otimes m_{v_{n_V}}(x) \\ &= (w_{v_1}, \dots, w_{v_{n_V}}) \end{aligned} \quad (3.4)$$

with the tensor operator  $\otimes$ . Let  $\oplus$  denote the s-norm operator max. For an FLC with a single output variable  $Y$  with the membership function  $m_Y(y) = m_{y_1}(y) \otimes \dots \otimes m_{y_{n_Y}}(y)$  its weighted membership function

$$\begin{aligned} m_Y^*(y) &= w_{y_1} \cdot m_{y_1}(y) \oplus \dots \oplus \\ & \quad w_{y_{n_Y}} \cdot m_{y_{n_Y}}(y) \end{aligned} \quad (3.5)$$

represents the outcome of the  $L$  related fuzzy rules  $v^{(l,1)} \times \dots \times v^{(l,n)} \rightarrow y^{(l)}$ ,  $1 \leq l \leq L$ , with  $n$  linguistic values  $v^{(l,1)}, \dots, v^{(l,n)}$  of  $n$  linguistic variables  $V^{(1)}, \dots, V^{(n)}$  and their weights  $w^{(l,1)}, \dots, w^{(l,n)}$ , dependent on the input-value vector  $(x_1, \dots, x_n)$  of the FLC, i.e., for  $1 \leq l \leq L$ ,  $1 \leq j \leq n$  and exactly one  $k$ ,  $1 \leq k \leq n_{V^{(j)}}$ , it holds:

$$\begin{aligned} v^{(l,j)} &= v_k^{(j)} \\ w^{(l,j)} &= w_{v_k^{(j)}} = m_{v_k^{(j)}}(x_j) \end{aligned} \quad (3.6)$$

If none of the linguistic values  $v^{(l,j)}$  of a linguistic variable  $V^{(j)}$  are used in a linguistic rule  $l$ , i.e., the linguistic rule  $l$  is independent of  $V^{(j)}$ , then  $w^{(l,j)}$  is set to 1.

Applying the often used norms min and max the weights  $w_{y_k}$ ,  $1 \leq k \leq n_Y$ , of  $Y$  can be expressed as:

$$w_{y_k} = \max_{y^{(l)}=y_k} \{ \min\{w^{(l,1)}, \dots, w^{(l,n)}\} \} \quad (3.7)$$

The weights of all linguistic values of the linguistic variables are used to determine the required fuzzy value by a weighted analysis of the linguistic rules with a fuzzy inference engine. Due to computational simplicity the membership function of a linguistic variable is often triangular or trapezoidal shaped (cf. Figures 3.1, 3.2 and Appendix C.2).

### 3.1.2 FEWA Fuzzy Controller

At the beginning of every control interval  $i$  with a fixed duration, e.g., 10 ms, FEWA measures the current queue length  $Q_i$  and its current growth rate  $G_i = Q_i - Q_{i-1}$  or its current fractional growth rate  $\Delta G_i = G_i/B$ , respectively. The membership function of FEWA regarding the queue length relative to the chosen target queue length  $QT$  is shown in Figure 3.1.

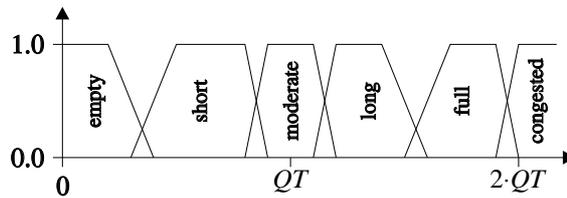


Figure 3.1: Membership function of the queue length  $Q$  of FEWA

The input range of the membership function of the fractional growth rate  $\Delta G$  (see Figure 3.2) has been changed to substantial lower values because of the chosen target queue length  $QT = \lfloor 0.25 \cdot B \rfloor$  at each router with FEWA-capabilities in the network.

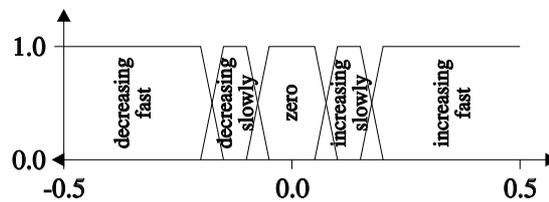


Figure 3.2: Membership function of the fractional queue growth rate  $\Delta G$  of FEWA

Before every control interval  $i$  the weights of all linguistic values are set to 0. Then the weights for  $w_{Q_k}$ ,  $1 \leq k \leq 6$ , for the six linguistic values (“empty”, “short”, “moderate”, “long”, “full”, and

“congested”) of the current queue length  $Q_i$  and the weights  $w_{G_k}$ ,  $1 \leq k \leq 5$ , for the five linguistic values (“decreasing fast”, “decreasing slowly”, “zero”, “increasing slowly”, “increasing fast”) of the current fractional queue growth rate  $\Delta G_i$  are determined. These weights are then used to calculate the weights  $w_{\alpha_k}$ ,  $1 \leq k \leq 6$ , for the linguistic values “very very little”, “very little”, “little”, “medium”, “high”, and “very high” of the utilization factor  $\alpha$  (cf. Appendix C.1). For the linguistic rule (R5), for instance, the last step of this computation can be expressed as (cf. Equation (3.7)):

$$w_{\alpha_5} = \max\{w_{\alpha_5}, \min\{w_{Q_2}, w_{\Delta G_4}\}\} \quad (3.8)$$

There may be other linguistic rules which have an influence on  $w_{\alpha_5}$ . For FERMA, FERMAM, and for the fuzzy controller of FEWA, a different formula has been chosen to calculate the weights of the linguistic values of  $\alpha$ . The linguistic rule (R5), for instance, can then be computationally expressed as:

$$w_{\alpha_5} = w_{\alpha_5} + w_{Q_2} \cdot w_{\Delta G_4} \quad (3.9)$$

There may be other linguistic rules which have an influence on  $w_{\alpha_5}$ , too. By the weights  $w_{\alpha_k}$ ,  $1 \leq k \leq 6$ , the membership function  $m_\alpha$  of  $\alpha$  is converted to  $m_\alpha^*$  (cf. Equation (3.5)). The weighted linguistic values of  $\alpha$  are combined by a special fuzzy calculation, called Center-of-Gravity (COG), to the new utilization factor  $\alpha$  (cf. [14]):

$$\alpha = \frac{\int y \cdot m_\alpha^*(y) dy}{\int m_\alpha^*(y) dy} \quad (3.10)$$

Analog to FERMA and FERMAM, the utilization factor  $\alpha$  is determined by a simplified calculation where only the weights of the linguistic values of  $\alpha$  are considered. Then, Equation (3.10) can be transformed into

$$\alpha = \frac{\sum_{k=1}^6 w_{\alpha_k} \cdot \alpha_k}{\sum_{k=1}^6 w_{\alpha_k}} \quad (3.11)$$

for a simplified computation with  $\alpha_1 = 1.00$ ,  $\alpha_2 = 2.00$ ,  $\alpha_3 = 4.00$ ,  $\alpha_4 = 6.00$ ,  $\alpha_5 = 9.00$ , and  $\alpha_6 = 15.00$  in the current version of the FEWA fuzzy controller. The values for the  $\alpha_k$ ,  $1 \leq k \leq 6$ , are chosen to set the utilization factor  $\alpha$  to a low value of 1 ( $= \alpha_1$ ) if the queue is congested. If the queue is empty, the utilization factor is set to a high value of 15 ( $= \alpha_6$ ). If the queue is neither congested nor empty, the utilization factor  $\alpha$  is set to a value between 1 and 15 according to the linguistic rules of the FEWA FLC (cf. Appendix C.1).

Compared to the FLC of FERMAM, only these  $\alpha_i$ 's are adapted for the FLC of FEWA. Similar to FERMAM, these values are selected to achieve a moderate queue length  $Q$  in the neighborhood of the target queue length  $QT$ . But due to the more complex congestion control in IP-based networks compared to ATM networks and the less available information in IP routers compared to ATM

switches, e.g., the number of TCP flows and their current congestion-control mode (slow start, congestion avoidance) are unknown, the target queue length in IP routers cannot be met as well as it can be done in ATM switches (cf. [21, 20]). Here, the precise values for the  $\alpha$ 's have been chosen both by general considerations and by evaluating various test simulations.

Figure 3.3 shows the current FEWA fuzzy controller's control surface for the utilization factor  $\alpha$  and for the sending window with  $B = 99$  packets and  $QT = 24$  packets, respectively. The fuzzy controller of FEWA is developed in such a manner that the current queue length  $Q_i$  has the main influence on the calculation of the utilization factor  $\alpha$  or of the sending window, respectively; the current (fractional) growth rate  $(\Delta)G_i$  is used to refine this calculation (in Figure 3.3,  $Q_i$  has the dominant influence). For the FEWA algorithm the described membership functions of the linguistic variables (see Appendix C.2) and the values of the utilization factor are the result of a more intuitive and pragmatic choice and not of an analytic approach (that this can work is one of the main advantages of fuzzy-logic controllers compared to other controller types). Nevertheless, this selection provides promising results.

It should be noted that due to the differences in the congestion-feedback functions of FERMAM and FEWA, in the FEWA fuzzy controller the values for the parameters  $\alpha_i$  cannot be independently chosen from the maximum queue length  $B$  in a router. Therefore, these fuzzy controller parameters have to be carefully determined for each maximum queue length  $B$  in FEWA-capable routers in a network. For other maximum queue lengths  $B$  than 99, the  $\alpha_i$ 's should be selected that the new control surface for the sending window patterns the control surface for the sending window shown in Figure 3.3. A rule of thumb for this  $\alpha_i$ -selection can be found in Appendix C.3.

## 3.2 Practicability of FEWA

In a real IP router there is no need for a complex and computational intensive fuzzy inference engine in the FLC. After the linguistic rules have been found and the linguistic values are tuned by a simulator, the control surface is known and can be stored as a lookup table for selected sampling points requiring only a few kilobytes of read-only memory (ROM) in a FEWA-capable router. In combination with a simple interpolation algorithm FEWA can be implemented in such a way with a very fast response time.

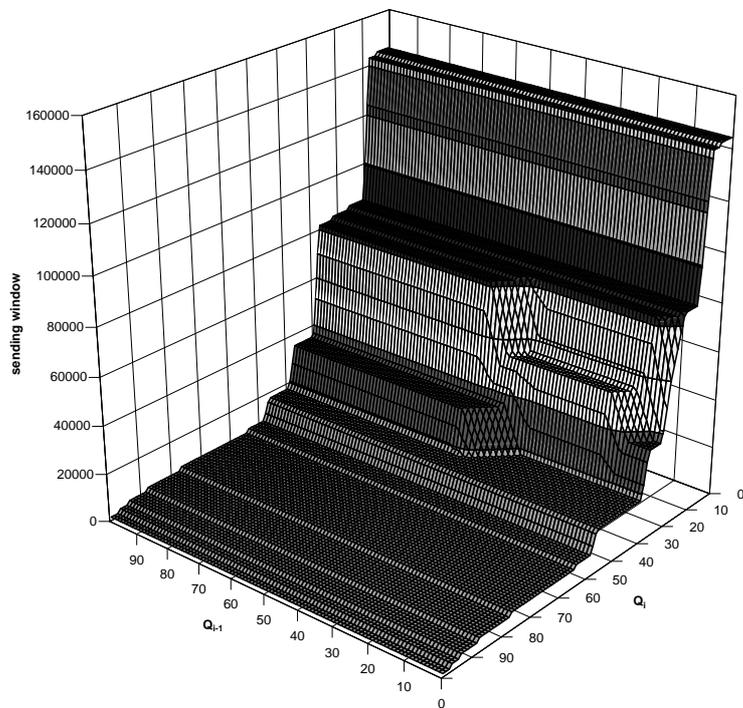
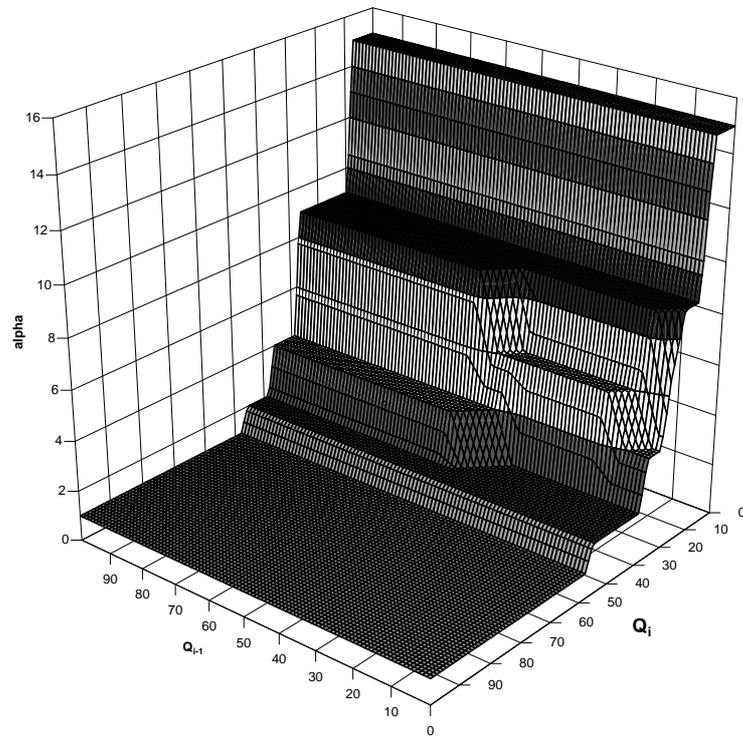


Figure 3.3: Control surfaces of the FEWA fuzzy controller for the utilization factor  $\alpha$  and of the FEWA algorithm for the sending window

## Chapter 4

# Simulation Model

In this chapter, the basic simulation model including the considered network topology with the location of EWA/FEWA-capable routers and the used traffic model are described.

### 4.1 Network Topology

The hierarchically structured network topology of the simulation model is shown in Figure 4.1. It consists of a public Internet part, one core network (CN), two radio access networks (RANs), and six radio cells (RCs) with a (wireless (W)) LAN access technology.

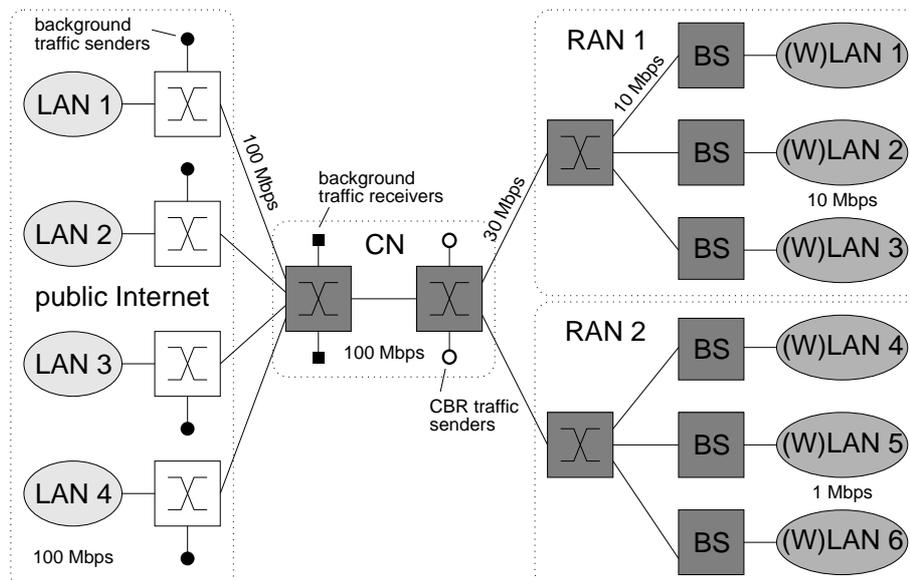


Figure 4.1: Simulated network topology

The public Internet part of the simulation model consists of four LANs and four routers. Each LAN has a bandwidth of 100 Mbps and a delay of  $5 \cdot 10^{-7}$  seconds. All links in the public Internet have a bandwidth of 100 Mbps. The delay between the routers in the public Internet and the ingress router of the core network is set to 50 ms. The link between the ingress router and the egress router of the core network has a bandwidth of 100 Mbps and a delay of 5 ms. The ingress routers of the RANs are connected to the egress router of the core network with links with a bandwidth of 30 Mbps and a delay of 5 ms.

The radio access networks RAN 1 and RAN 2 are intended to represent wireless access networks, e.g., WLAN. I am interested in such a setup as this represents one of the most important scenarios where a large-bandwidth core network is connected with a small-bandwidth radio access network, resulting in bottlenecks at the transition between these networks. Therefore, the RAN routers connected to the (W)LANs 1–6 represent “base stations” ((W)LAN routers) of these networks. The base stations are connected to the ingress RAN router by links with a bandwidth of 10 Mbps and a delay of 5 ms. Each (W)LAN in RAN 1 provides a bandwidth of 10 Mbps while each (W)LAN in RAN 2 provides a bandwidth of 1 Mbps. These values are chosen to reflect typical maximum bandwidths of currently widely used WLAN access technologies. The delays in the (W)LANs are set to a constant, small value. Both the packet error rate  $p$  in the (W)LANs and the handover-rate of the mobile TCP receivers in the (W)LANs is adjustable. With this simulation model, the performance of standard different RCF approaches, e.g., TCP, standard TCP assisted by EWA (TCP+EWA), or standard TCP assisted by FEWA (TCP+FEWA), can be investigated in different scenarios of future IP-based networks. But in this chapter, the performance of standard TCP and standard TCP assisted by EWA or assisted by FEWA is investigated in a network scenario with reliable links in the (W)LANs, immobile TCP receivers, and only the mean network load is varied.

All router queues in the simulation scenario are droptail queues. These router queues have a maximum queue length limit of 100, i.e., every router queue can store at most  $B = 99$  IP packets. The target queue length of FEWA is set to  $QT = 24$ .

In each of the LANs in the public Internet,  $n$  TCP senders are located. The dedicated TCP receivers are distributed over the (W)LANs in the two radio access networks. 16 background TCP senders are connected to the routers in the public Internet, 4 to each of them. The dedicated background TCP receivers are connected to the ingress router of the core network. 12 Background UDP senders are connected to the egress router of the core network, while the dedicated UDP receivers are uniformly distributed over the (W)LANs of the RANs.

## 4.2 Location of RCF-Capable Routers

Both routers in the core network and the routers in the radio access networks are equipped with different RCF mechanisms. These additional router functionalities can be optionally switched on and off; the combinations of standard TCP, TCP+EWA, and TCP+FEWA are considered here.

## 4.3 Traffic Load Models

Properly characterizing traffic loads for interactive Internet users is a difficult undertaking. In order to represent some mixture of different applications, I decided to use one TCP-based application class to generate WWW traffic and one UDP-based application class to generate voice traffic.

The first application class includes TCP connections where the number of segments to send is determined by a WWW traffic model [16]. This traffic model is derived from real HTTP traces in corporate and educational environments and uses three abstraction levels: The session level, the page level, and the packet level. Here, a simplified version of this model is used which consists only of the first two levels. In every WWW session a log-normally distributed number of WWW pages is sent with page sizes that follow a curtailed Pareto distribution, i.e., the page size is limited to 1 MByte. The time between the pages, i.e., the inter-connection or reading time, is gamma distributed. The load in the network can be easily adjusted by using the exponentially distributed session interarrival time with a different parameter. The distributions and parameters chosen for the stochastic variables of the simplified WWW traffic model are shown in Table 4.1.

Table 4.1: Distributions and parameters for the stochastic variables of the simplified WWW model

Stochastic variable	Distribution	Distribution parameter(s)
Inter-session time	Exponential	$\mu = 5.0$ s
Pages per session (pps)	Lognormal	$\mu = 25.807$ pps $\sigma = 78.752$ pps
Inter-page time (reading time)	Gamma	$\mu = 35.286$ s $\sigma = 147.390$ s
Page size	Pareto (limited)	$\alpha = 1.7584$ $\beta = 30458$ Bytes

This traffic model is used for all TCP connections whose receivers are located in the (W)LANs of the RANs. The background traffic TCP connections in the public Internet part of the simulation model use the same distribution for their pages per session and their page size as shown in Table 4.1. But the inter-session time and the inter-page time are deterministically set to zero.

The second application class includes UDP data streams which send every 20 ms an UDP packet

with a payload length of 36 bytes. This class is used to model traffic with a constant bit rate (CBR) like voice-over-IP.

## 4.4 Implementation Details

The whole simulation model is implemented in ns-2 (version 2.1b9) [3]. Since the TCP header in ns-2 does not include an advertized receiver window, I have included it. For all TCP connections in the simulation model, a modified ns-2 implementation of a TCP NewReno [4] sender or receiver is used. The modifications in the TCP senders include the analysis of the advertized receiver window of received TCP acknowledgments and additional statistical performance evaluation methods (cf. Chapter 5). In addition, the limitation of the CWND on the ARWND defined in [1] is also implemented in the TCP senders used in the simulation framework. The TCP receivers in ns-2 are modified to set the advertized receiver window in TCP acknowledgments according to their current capability to receive new TCP segments (then the advertized receiver window is set to a constant value of 373760 bytes (= 256 TCP segments with a maximum segment size of 1460 bytes)) or to perform the Freeze-TCP [6] mechanism before a handover is executed (then the advertized receiver window is temporarily set to zero to stop transmissions from the TCP sender during the handover procedure in order to avoid packet losses due to the handover). In the latter case, also the whole mobility model is implemented in the TCP receivers, since Freeze-TCP needs the exact point in time where the next handover will be executed to inform the TCP sender early enough about the impending handover.

In contrast to a real mobile node there is no possibility to detect an impending handover within the network simulation, e.g., by monitoring the signal strength. Because of this restriction the point in time of the next handover event, the handover time  $T_{ho}$ , is determined by the TCP receiver itself. Within the time interval  $T_{ho} - T_{now} \leq SRTT$  the receiver has the possibility to freeze the TCP sender where  $T_{now}$  is the current simulation time and SRTT is the smoothed round trip time measured by the TCP receiver. In the current implementation of Freeze-TCP, it is necessary that the TCP receiver receives TCP segments that it can acknowledge. Otherwise, the TCP sender cannot be "frozen". I do not introduce a mechanism that a TCP receiver is able to retransmit the last sent TCP acknowledgment with an adapted advertized receiver window of zero to immediately freeze the TCP sender.

In the basic Freeze-TCP mechanism described in [6], three copies of the last acknowledgment prior to the disconnection are sent to inform the TCP sender about the re-established connection. Unfortunately, the authors of Freeze-TCP give no information what to do in case of losing the wakeup acknowledgments. I have identified this as a possible problem and therefore have extended the wakeup-mechanism of Freeze-TCP by including a timeout timer to retransmit these three acknowledgments if the TCP receiver does not receive any TCP segments until this timer timeouts. The number of retransmissions and the timeout interval are adjustable by the user. The default timeout in-

terval is set to twice the SRTT which seems to be an appropriate compromise between the avoidance of unnecessary retransmissions and a reaction in short time due to packet losses. For my simulations, I try to wakeup the TCP sender after the completion of the handover for at most 64 times.

In order to count packet losses in queues and links, the link queue class and the error model class of ns-2 are adapted. The queues in ns-2 belong to the links and they are all output queues from the router's point of view. Therefore, a new basic router class is created that virtually combines all queues which belong to a specific router in the network topology. Since a single queue of a link in ns-2 can belong to different objects of my new router class, I need a specific handling of packets in each of the queues. To reach this, I mark each incoming IP packet in a queue that the router class is able to distinguish between IP packets coming from and going to different neighbored network nodes. For this reason, the link-trace class in ns-2 is adapted.

The basic router class provides three key functionalities: (a) a mechanism to monitor the packet queue and to calculate statistical values, e.g., the smoothed queue length, (b) an algorithm to calculate a router-internal advertized receiver window for each link, and (c) a mechanism that adapts the advertized receiver window in the TCP acknowledgments. As a derivation of the basic router class, the EWA router class is implemented in ns-2. This EWA router class performs the EWA algorithm described in [10] for all of the combined queues in the router class. With one exception: I have detected by evaluating results from earlier simulations that it is advantageous to introduce an upper bound for the parameter  $\alpha$  or for the sending window of the EWA-algorithm. Otherwise, routers which are low-loaded for a longer period of time will increase their sending window without a limit. In my simulations, I use an upper bound for the sending window that is equal to the maximum advertized sending window (373760 bytes) of the TCP receivers. In reality, the parameter  $\alpha$  should be limited (cf. Section 2.4).

New RCF approaches related to EWA (cf. Section 2.4) can be easily implemented in ns-2 by deriving the basic router class. Examples are FEWA and FEWA-based approaches specifically designed for radio network controllers (RNCs) in WCDMA-based networks like UMTS [18]. But also other RCF approaches, e.g., the controllers of the explicit control protocol (XCP) [11] in the routers can be supported by the router class if minor changes in the basic mechanisms of this router class are performed. Therefore, the existing simulation framework can be easily extended to evaluate also the performance of other and in future developed RCF approaches in IP-based networks.

For the traffic model of the simulation, a new traffic generator class is implemented. This traffic generator class provides WWW traffic based on TCP, ON-OFF traffic based on TCP, CBR traffic based on UDP, and UDP-based traffic according to trace files, e.g., video or audio traffic traces.

## Chapter 5

# Performance Evaluation Metrics

In this chapter, I describe the different metrics used for the performance evaluation of different RCF approaches.

### 5.1 Throughput

To compare the simulation results of the pure end-to-end congestion control approach with the simulation results of the different explicit router congestion feedback (RCF) approaches, two different overall mean throughput computations for TCP connections are used. If  $n$  TCP connections are considered and each of these TCP connections has sent  $s_i$  segments in duration  $d_i$ ,  $1 \leq i \leq n$ , then the two overall mean throughput calculations work as follows:

- Computation of the overall mean throughput ( $\bar{T}_1$ ): The sum of sent segments of all  $n$  TCP connections is divided by the overall duration of these TCP connections:

$$\bar{T}_1 = \frac{\sum_{i=1}^n s_i}{\sum_{i=1}^n d_i}$$

- Computation of the connection-oriented mean throughput ( $\bar{T}_2$ ): For each of the  $n$  TCP connections a mean throughput  $t_i$ ,  $1 \leq i \leq n$ , is calculated. All these mean throughput values are then used to compute the overall mean throughput of the TCP connections by a normal non-weighted arithmetic mean calculation independent of the number of segments sent by each of the TCP connections:

$$\bar{T}_2 = \frac{1}{n} \cdot \sum_{i=1}^n \frac{s_i}{d_i} = \frac{1}{n} \cdot \sum_{i=1}^n t_i$$

The first throughput calculation is the more important one, since with this throughput metric the overall throughput of the considered TCP connections can be evaluated. The second calculation gives a connection-oriented mean throughput which can be understood as the mean throughput a single TCP connection can expect.

These two throughput metrics are calculated separately for each (W)LAN by considering all TCP connections with receivers in each of the (W)LANs.

## 5.2 Queue Loss Rate

In addition, I count the number of packet losses in the base stations of the (W)LANs. These values can then be used to calculate queue loss rates for the base stations of the (W)LANs as another performance evaluation metric.

## Chapter 6

# Performance Evaluation of EWA and FEWA by Simulations

### 6.1 Introduction

In the following, two simulation scenarios are considered which differ in their mean load in the network: the first simulation scenario uses  $n = 24$  TCP senders in each LAN of the public Internet to generate WWW traffic in the network, the second one doubles this value. In each scenario the performance metrics of standard TCP is compared with the performance metrics of standard TCP assisted by EWA or standard TCP assisted by FEWA. In addition, for each simulation scenario the queue length process, the queue length histogram, and the complementary cumulative distribution function (CCDF) of the queue length of each downstream queue in the base stations in RAN 2 are shown.

Currently, twenty simulation runs per scenario case are performed. Every run for each simulation scenario with either a lower or a higher mean network load considers a simulated time of 18000 seconds. The transient analysis of the simulations (cf. Appendix A) shows that the throughputs of the TCP connections have a transient phase of about 1800 seconds. Therefore, only the throughput observations after the transient phase can be used to calculate the mean values of the different throughput metrics. In addition, also the mean numbers of packet losses in the downstream queues of the base stations in RAN 2 are calculated by ignoring packet losses during the transient phase.

The last column  $\Delta$  in the following Tables 6.1 and 6.2 denotes whether the simulation results of one approach compared to the simulation results of another approach are statistically significantly different or not. In this column the approaches are compared in the following order: standard TCP  $\leftrightarrow$  TCP+EWA, standard TCP  $\leftrightarrow$  TCP+FEWA, and TCP+EWA  $\leftrightarrow$  TCP+FEWA. A '-' denotes that the first approach of the comparison reaches significantly better results while a '+' denotes that the

second approach reaches significantly better results. And a '=' means that with the simulation results no significant difference between the two compared approaches can be concluded. The details of the statistical evaluation of these simulation results can be found in Appendix B.

## 6.2 Scenario 1: TCP NewReno, Lower Mean Network Load

Table 6.1: Simulation results of scenario 1 —  $\bar{T}_1$  is the overall mean throughput,  $\bar{T}_2$  is the connection-oriented mean throughput,  $\Delta$  denotes the statistical significance of the simulation results

		standard TCP	TCP+EWA	TCP+FEWA	$\Delta$
$\bar{T}_1$ of TCP connections [segments/second]	(W)LAN 1	52.77	52.87	52.83	====
	(W)LAN 2	52.78	52.90	52.69	====
	(W)LAN 3	53.04	52.84	52.89	====
	(W)LAN 4	37.78	37.86	38.68	===+
	(W)LAN 5	37.72	37.79	38.78	===+
	(W)LAN 6	37.82	38.04	38.65	===+
$\bar{T}_2$ of TCP connections [segments/second]	(W)LAN 1	47.59	47.66	47.67	====
	(W)LAN 2	47.65	47.67	47.63	====
	(W)LAN 3	47.71	47.67	47.66	====
	(W)LAN 4	36.36	36.29	36.46	==+
	(W)LAN 5	36.36	36.31	36.46	==+
	(W)LAN 6	36.36	36.42	36.39	====
mean number of packet losses in base station ((W)LAN router)	(W)LAN 1	0.00	0.00	0.00	====
	(W)LAN 2	0.00	0.00	0.00	====
	(W)LAN 3	0.00	0.00	0.00	====
	(W)LAN 4	2581.10	954.30	0.00	+++
	(W)LAN 5	2598.30	912.70	0.00	+++
	(W)LAN 6	2561.50	891.55	0.00	+++

In this scenario with a lower mean network load the base stations in RAN 1 are not congested at all, i.e., no packets are lost in the downstream queues of these base stations, and the base stations in RAN 2 are rarely congested if standard TCP connections are considered. The throughputs of TCP connections traversing the base stations in RAN 1 are comparable for the approaches standard TCP, TCP+EWA, and TCP+FEWA; EWA reaches nearly the same throughput than standard TCP while the throughput loss of FEWA is approximately 0.12 % compared to standard TCP and EWA. For connections traversing the base stations in RAN 2, TCP+FEWA outperforms standard TCP as well as TCP+EWA in most of the throughput metrics. For these TCP connections, the overall mean throughput gain of TCP+FEWA is approximately 2 % compared to both standard TCP and TCP+EWA.

Compared to standard TCP, the EWA algorithm considerably decreases the number of packet losses in congested routers by more than 64 %. But packet losses can be completely prevented in these routers if the EWA algorithm is replaced by the FEWA algorithm.

### 6.3 Scenario 2: TCP NewReno, Higher Mean Network Load

Table 6.2: Simulation results of scenario 2 —  $\bar{T}_1$  is the overall mean throughput,  $\bar{T}_2$  is the connection-oriented mean throughput,  $\Delta$  denotes the statistical significance of the simulation results

		standard TCP	TCP+EWA	TCP+FEWA	$\Delta$
$\bar{T}_1$ of TCP connections [segments/second]	(W)LAN 1	47.72	47.79	47.94	===
	(W)LAN 2	47.72	47.67	47.91	===
	(W)LAN 3	47.97	47.67	47.94	-==
	(W)LAN 4	28.31	28.04	29.30	===
	(W)LAN 5	28.10	27.96	29.48	===
	(W)LAN 6	28.17	28.11	29.42	===
$\bar{T}_2$ of TCP connections [segments/second]	(W)LAN 1	44.98	44.97	45.09	===
	(W)LAN 2	44.94	44.96	45.03	===
	(W)LAN 3	45.06	44.92	45.04	===
	(W)LAN 4	30.46	30.22	30.34	===
	(W)LAN 5	30.42	30.14	30.45	-==
	(W)LAN 6	30.35	30.22	30.40	===
mean number of packet losses in base station (W)LAN router)	(W)LAN 1	0.00	0.00	0.00	===
	(W)LAN 2	0.00	0.00	0.00	===
	(W)LAN 3	0.00	0.00	0.00	===
	(W)LAN 4	14840.70	4636.05	0.00	+++
	(W)LAN 5	15196.55	4706.90	0.00	+++
	(W)LAN 6	14980.55	4626.10	0.00	+++

In this scenario with a higher mean network load the base stations in RAN 1 are still not congested at all, i.e., no packets are lost in their downstream queues, and the base stations in RAN 2 are much more often congested than in the previous scenario if standard TCP connections are considered. For standard TCP connections traversing the uncongested base stations in RAN 1 the throughputs are comparable to the throughputs of TCP connections either assisted by EWA or assisted by FEWA (TCP+FEWA reaches a slight throughput gain compared to the other two approaches). For TCP connections traversing the base stations in RAN 2 TCP+FEWA outperforms both other approaches. For these TCP connections the overall mean throughput gain of TCP+FEWA is approximately 4 % compared to standard TCP and approximately 5 % compared to TCP+EWA. The

connection-oriented mean throughput is slightly increased by both approaches TCP and TCP+FEWA compared to TCP+EWA.

Compared to standard TCP, the EWA algorithm considerably decreases the number of packet losses in congested routers by 69 %. But also in this scenario packet losses can be prevented if the EWA algorithm is replaced by the FEWA algorithm in these routers.

## 6.4 Queue Lengths in Bottleneck Base Stations

In this section, the queues of the base stations in RAN 2 are considered in more detail. Figure 6.1 shows typical queue length processes in the downstream queue of a base station in RAN 2 for both simulation scenarios and for a measurement interval of 120 s. In the scenario with a lower mean network load and compared to standard TCP, TCP+EWA is able to slightly decrease the maximum queue length in these base stations. But in the scenario with a higher mean network load, standard TCP and TCP+EWA are both not able to prevent packet losses due to congestion. For the scenario with a lower mean network load, TCP+FEWA reaches considerably lower maximum queue lengths. In the scenario with a higher mean network load TCP+FEWA considerably reduces the frequency of very full queues. This observation is formalized by the histograms and the complementary cumulative distribution functions (CCDFs) of the queue length observations after the transient phase, shown in Figure 6.2 and Figure 6.3, respectively. The histograms depict the queue length of the downstream queues of the base stations 4–6, the base stations in RAN 2, for both simulation scenarios. To show that the obtained results are similar for the base stations 4–6 for each of the considered approaches these histograms are put on top of each other. For the same queues the CCDFs depict the probability to observe queue lengths that are above a given queue length. The y-axis representing this probability is logarithmically scaled to focus on the tail probabilities for larger queue lengths. In contrast to the other two approaches, it can be seen that with TCP+FEWA the occurrence of larger queue lengths in the considered queues is very unlikely or is even prevented.

For standard TCP, the queues in the base stations of RAN 2 have a mean queue length of 7.16 in the scenario with a lower mean network load and 19.53 in the scenario with a higher mean network load. If standard TCP is assisted by EWA, the first value is slightly increased to 7.21 while the second value is considerably decreased to 16.65. Standard TCP assisted by FEWA reaches the lowest mean queue lengths of 4.64 and 11.41, respectively. More importantly, TCP+FEWA prevents that packets are lost due to a congested queue. Thus, routers that are equipped with FEWA are able to handle much more TCP connections without being congested.

In the simulated network topology and with the given traffic model the base stations in the RAN 2 are lowly to moderately loaded most of the simulated time. Therefore, the utilization factor  $\alpha$  in the EWA algorithm is increased to such high values in periods of time with a lower load in the base stations that it can not be decreased as fast as necessary in periods of time with highly loaded or even congested base stations. Thus, the EWA algorithm is not able to adequately control the sending window of the TCP senders to avoid congestion and packet losses in the base stations if a realistic traffic model for IP-based networks is used. Only the FEWA algorithm is able to early react on impending congestion in the base stations and adequately limit the sending window of TCP senders such that no packets at all are lost due do congestion in the base stations.

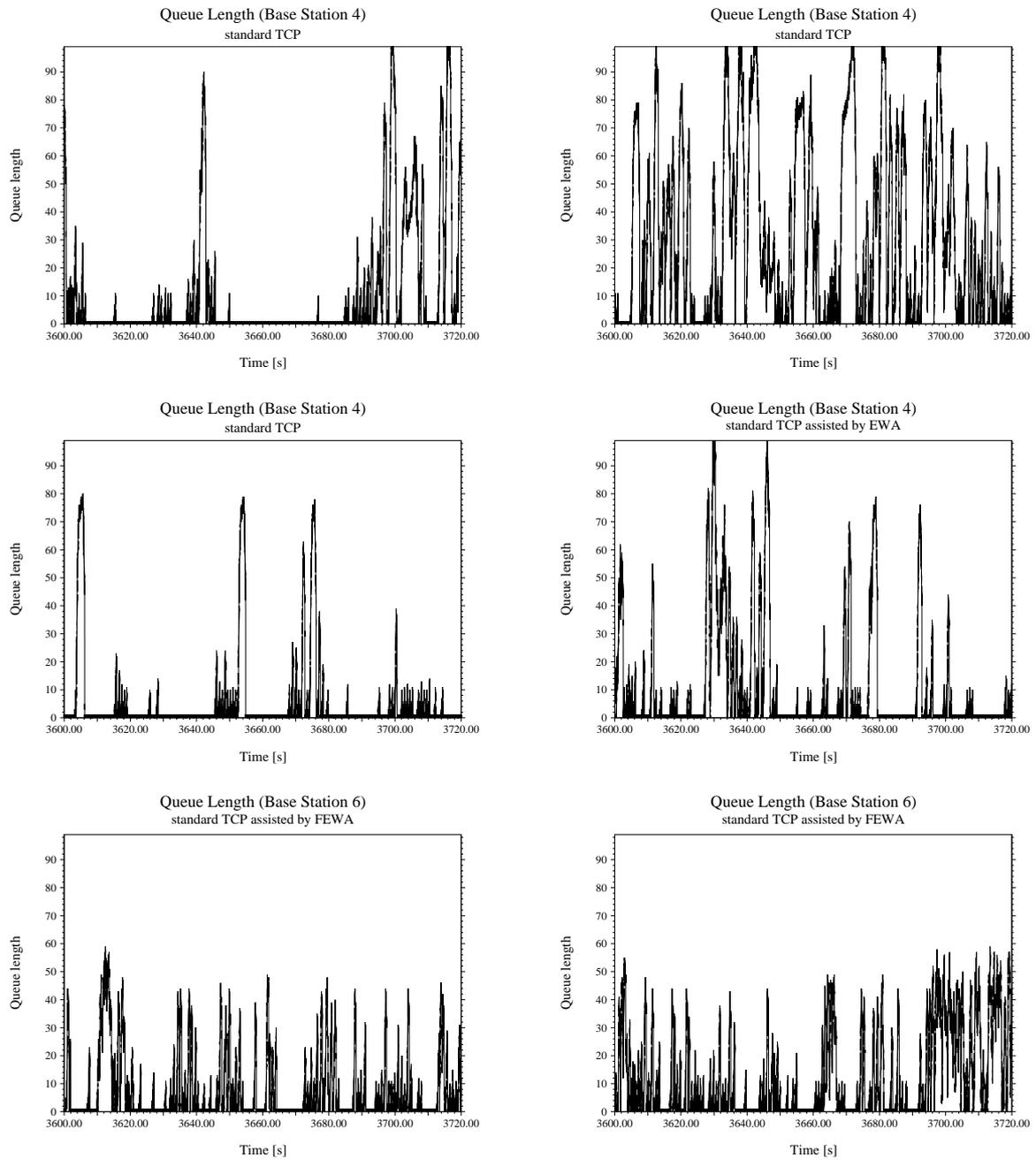


Figure 6.1: Queue length process in the downstream queue of a base station in RAN 2 for standard TCP, TCP+EWA, and TCP+FEWA (lower mean network load on the left side, higher mean network load on the right side)

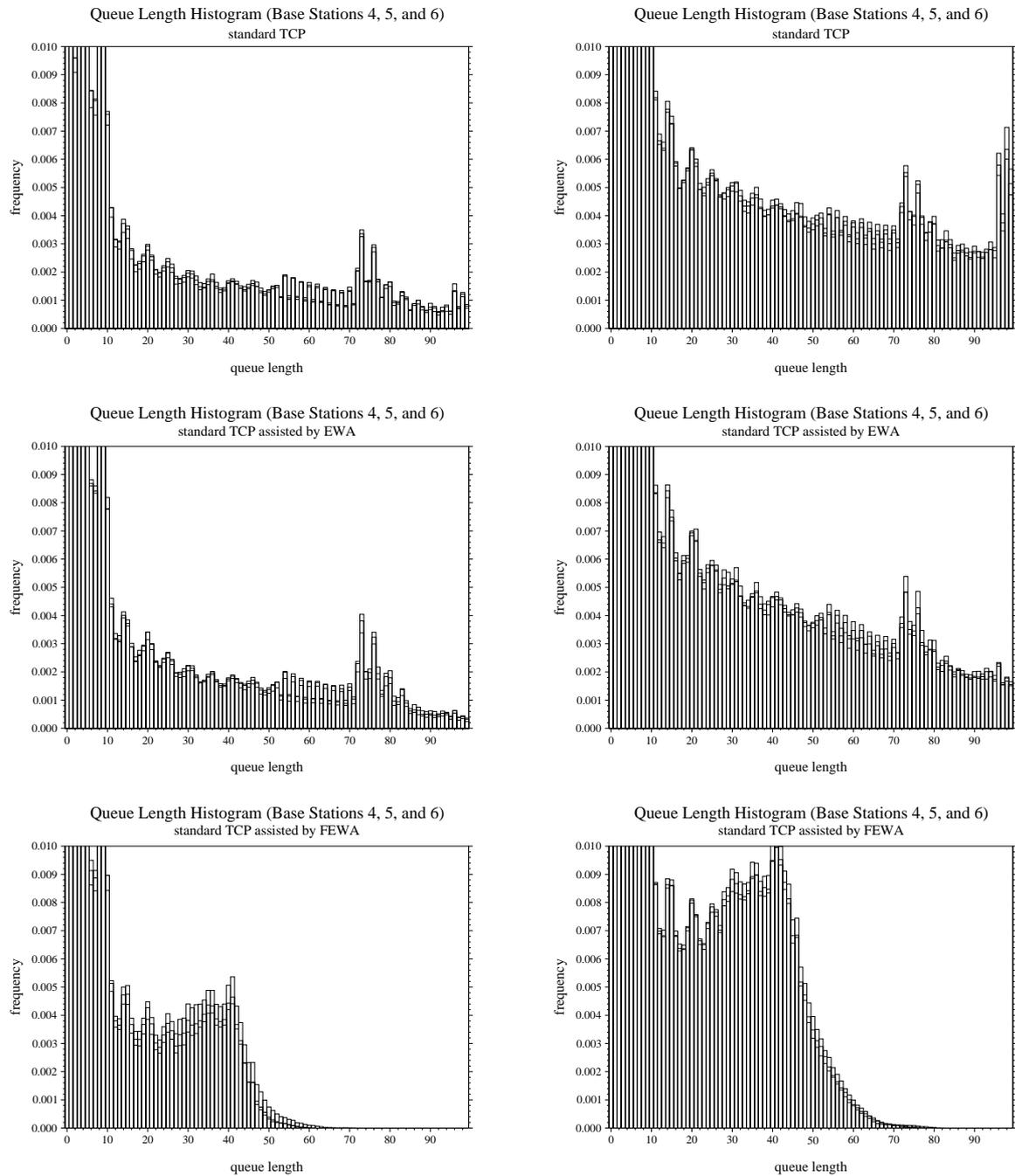


Figure 6.2: Queue length histograms in the downstream queues of the base stations in RAN 2 for standard TCP, TCP+EWA, and TCP+FEWA (lower mean network load on the left side, higher mean network load on the right side, for each of the considered approaches the histograms of the base stations 4–6 are put on top of each other)

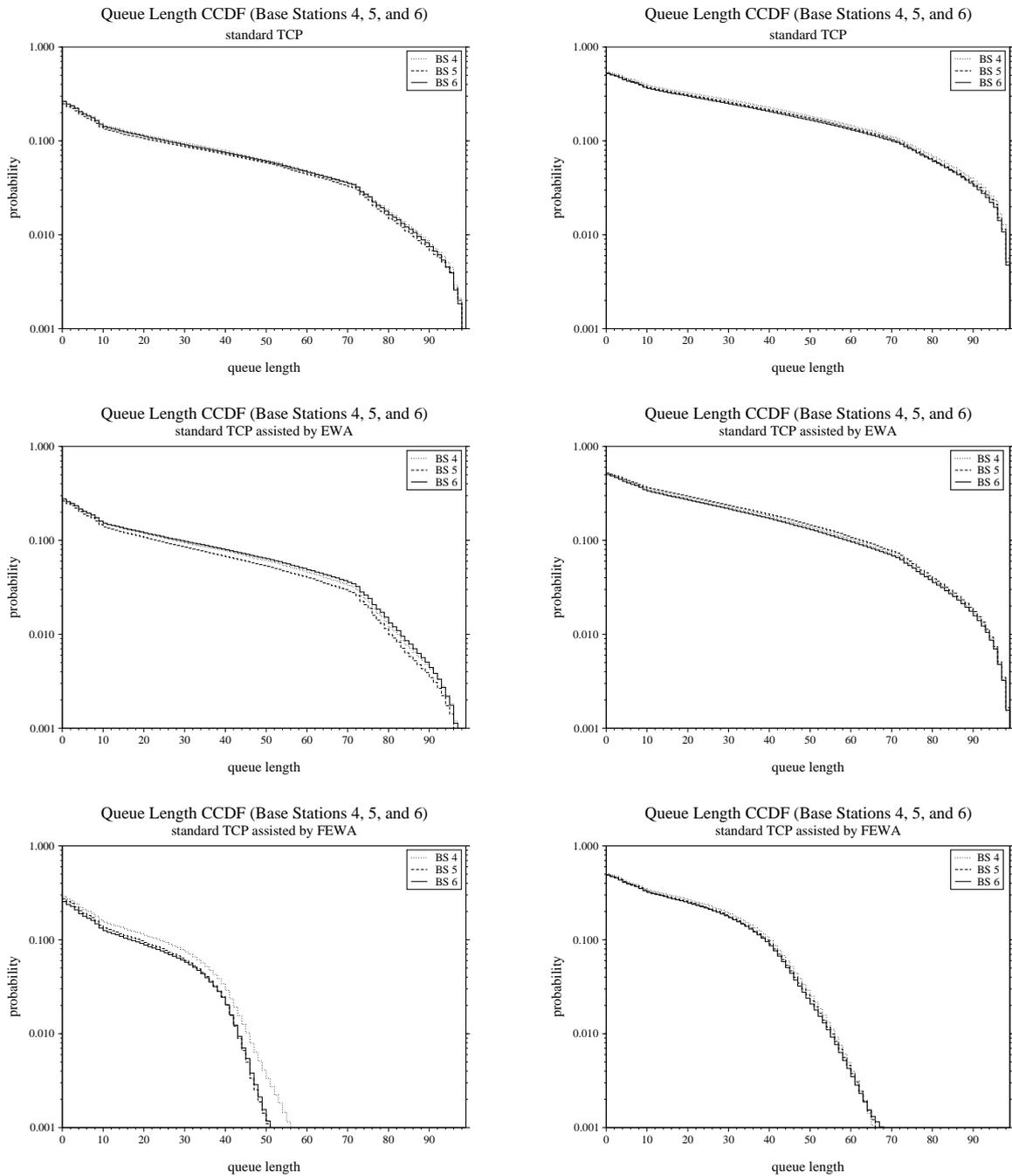


Figure 6.3: Queue length complementary cumulative distribution functions in the downstream queues of the base stations in RAN 2 for standard TCP, TCP+EWA, and TCP+FEWA (lower mean network load on the left side, higher mean network load on the right side)

## Chapter 7

# Conclusion and Outlook

Using mechanisms that provide explicit congestion feedback information from routers appears very promising to improve the performance of TCP connections. One of these approaches, EWA, suffers from a major shortcoming as it cannot adequately support bursty traffic. Therefore, I have developed a new EWA-related algorithm called FEWA which adapts the main congestion-control algorithm in EWA-capable routers by a more appropriate fuzzy-based calculation. In this chapter, I have investigated the performance of standard TCP and standard TCP either assisted by EWA or assisted by FEWA.

Using two simulation scenarios with different mean network loads, I achieve the following results: If routers in the network are lowly to moderately loaded in the mean that congestion is likely to happen, the throughput of TCP connections traversing these routers is increased if TCP's end-to-end congestion control is assisted by well-designed explicit congestion feedback approaches from routers. In these cases, TCP+FEWA is able to outperform standard TCP with a throughput increase of 2–4 % and TCP+EWA with a throughput increase of even 2–5 %. In addition, FEWA completely prevents packet losses due to congestion in the base stations of RAN 2. If routers in the network are not congested at all, the throughput of TCP connections traversing these routers are not impacted or only very slightly decreased compared to standard TCP if TCP's end-to-end congestion control is assisted by EWA or assisted by FEWA. Thus, EWA and FEWA only reduce the sending window of TCP senders if it is necessary in order to reduce or completely avoid losses due to (impending) congestion in routers.

Considering these results, the implementation of FEWA in routers to assist TCP's end-to-end congestion control is highly recommended. FEWA reaches considerable performance gains with a low additional computational complexity in routers (cf. Section 3.2). Therefore, FEWA is another example that fuzzy-based congestion control algorithms are able to more adequately react on changing load in routers than approaches based on other control-theoretic assumptions, e.g., time series.

In addition, it might be interesting to investigate how the fuzzy controller of the FEWA algorithm can be combined with other existing approaches, e.g., RED or Explicit Control Protocol (XCP) [11]. In combination with RED a modified FEWA fuzzy controller can then be used to determine the packet loss probability dependent on the current queue length. And the efficiency controller of XCP can be replaced by a modified FEWA fuzzy controller to reach a higher performance of aggregated traffic over a router with XCP-capabilities. The latter approach will be investigated in more detail in an ongoing project with Ericsson Research.

## Appendix A

# Transient Analysis of the Simulations

### A.1 Transient Analysis Method

The method used for the transient analysis of the simulation results is called initial data deletion [7]. Suppose there are  $m$  replications with  $n_i$  observations in replication  $i$ ,  $1 \leq i \leq m$ . The  $j$ th observation in the  $i$ th replication is denoted  $x_{ij}$ . Let  $n$  be the minimum of all  $n_i$ . Then the method works as follows:

- (1) By averaging across the replications a mean value  $\bar{x}_j$  for the  $j$ th observation is calculated:

$$\bar{x}_j = \frac{1}{m} \sum_{i=1}^m x_{ij}, \quad j = 1, 2, \dots, n \quad (\text{A.1})$$

- (2) Using the values from (1) an overall mean value  $\bar{\bar{x}}$  is calculated:

$$\bar{\bar{x}} = \frac{1}{n} \sum_{j=1}^n \bar{x}_j \quad (\text{A.2})$$

- (3) Vary  $l$  between 1 and  $n - 1$ :

- (a) Delete the first  $l$  observations from the simulation results and calculate an overall mean value  $\bar{\bar{x}}_l$  for the remaining  $n - l$  observations:

$$\bar{\bar{x}}_l = \frac{1}{n-l} \sum_{j=l+1}^n \bar{x}_j \quad (\text{A.3})$$

- (b) Compute the relative change  $\Delta_l$  in the overall mean value:

$$\Delta_l = \frac{\bar{\bar{x}}_l - \bar{\bar{x}}}{\bar{\bar{x}}} \quad (\text{A.4})$$

A plot of  $\Delta_l$  as a function of  $l$  shows the transient phase of the simulation results. If the plot stabilizes at a point  $l^*$ , then the first  $l^*$  observations of the simulations reflect the transient phase of the simulation results and should not be considered for the mean calculation.

## A.2 Transient Analysis Results

As an example, the following Figure A.1 shows the transient analysis for the throughput of TCP connections traversing the base station of radio cell 6 (cf. Figure 4.1) in a simulation scenario with lower mean network load. Figure A.2 shows the same  $l-\Delta_l$  plot in more detail.

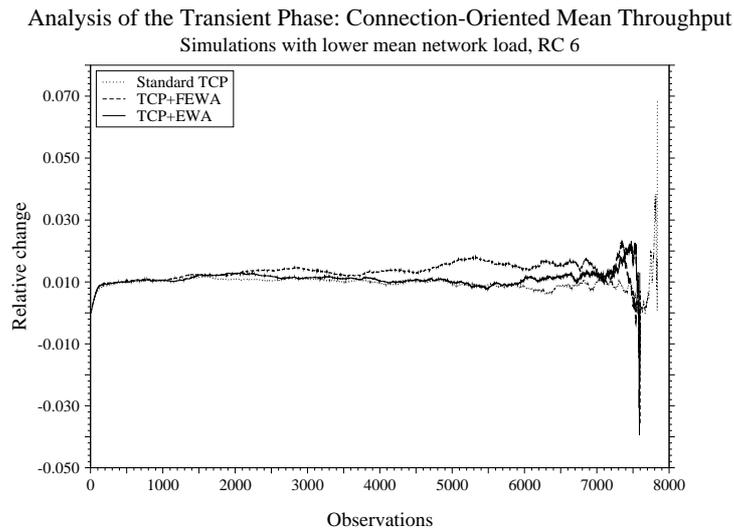


Figure A.1: Transient analysis for an example simulation of part II with lower mean network load

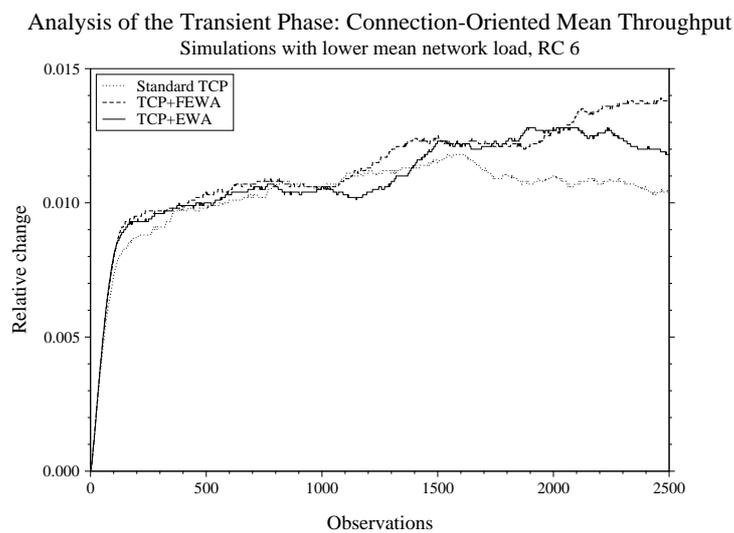


Figure A.2: Zoomed transient analysis for an example simulation of part II with lower mean network load

Another example pictured in Figure A.3 shows the transient analysis for the throughput of TCP connections traversing the base station of radio cell 6 (cf. Figure 4.1) in a simulation scenario with higher mean network load. Figure A.4 shows the same  $l-\Delta_l$  plot in more detail.

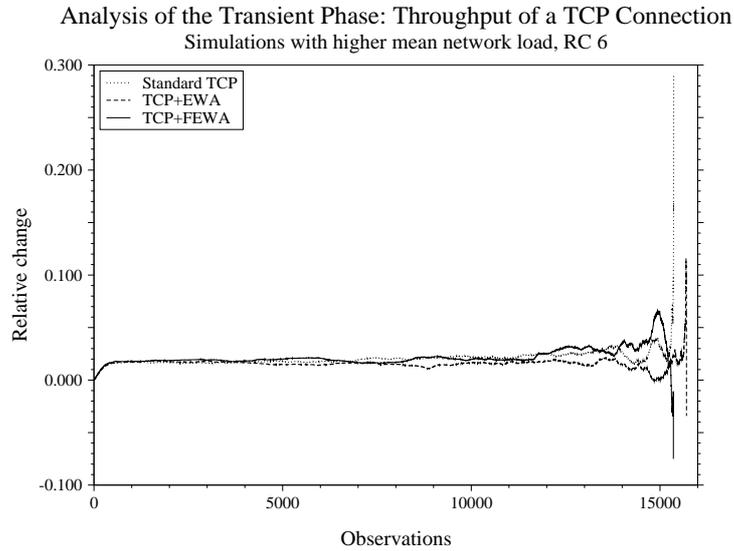


Figure A.3: Transient analysis for an example simulation of part II with higher mean network load

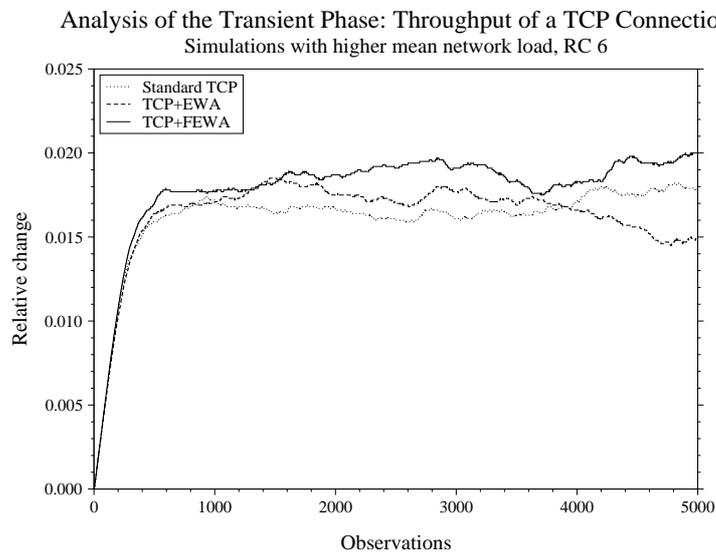


Figure A.4: Zoomed transient analysis for an example simulation of part II with higher mean network load

In both examples and in other simulation scenarios of part II, a transient phase of approximately

10 % of all TCP throughput observations can be stated. This can be converted to a transient phase of less than 10 % of the whole simulated time of 18000 s. Therefore, I have determined a transient phase of 1800 s. I calculate the mean values of the performance metrics by using all TCP throughput observations and packet loss events after the transient phase of 1800 s and leave out all TCP throughput observations and packet loss events during this transient phase.

## Appendix B

# Appendix: Statistical Evaluation

### B.1 Statistical Evaluation Method

The statistical evaluation method used for this comparison is called the t-test for unpaired observations of two alternatives and is described in detail in [7]. The main idea of this method is to compute a confidence interval for the difference of the mean values of both alternatives for a given confidence level. Then the decision criterion is:

- If the confidence interval includes zero, then the two alternatives can not be distinguished.
- If the confidence interval is above/below zero, then the first/second alternative is the better one.

Tests with confidence intervals give not only a yes-no answer like other hypothesis tests, they also give an answer to the question how precise the decision is. A narrow confidence interval indicates that the precision of the decision is high whereas a wide confidence interval indicates that the precision of the decision is rather low.

This t-test for unpaired observations of two alternatives is used for the statistical evaluation of the simulation results for both the overall mean throughput ( $\bar{T}_1$ ) and the connection-oriented mean throughput ( $\bar{T}_2$ ) for TCP connections with receivers in each of the (W)LANs and the mean number of packet losses in the base station of each (W)LAN.

The values for this statistical evaluation are produced by ten independent simulation runs of the considered simulation scenario.

### B.2 Statistical Evaluation Results for EWA and FEWA Simulations

In the following Tables B.1 and B.2 the statistical evaluation of the simulation results are shown. For each confidence interval also the confidence level (0.90, 0.95 or 0.99) is depicted. If the simulation

results for standard TCP, standard TCP assisted by EWA, or standard TCP assisted by FEWA are not significantly different even for the confidence level 0.90, then the confidence interval for the confidence level 0.90 is stated.

Table B.1: Statistical evaluation of the simulation results of scenario 1

	standard TCP $\leftrightarrow$ TCP+EWA	standard TCP $\leftrightarrow$ TCP+FEWA	TCP+EWA $\leftrightarrow$ TCP+FEWA	
$\bar{T}_1$ of TCP connections [1 - $\alpha$ : conf()]	(W)LAN 1	0.90 : (- 0.38, + 0.19)	0.90 : (- 0.29, + 0.18)	0.90 : (- 0.23, + 0.31)
	(W)LAN 2	0.90 : (- 0.38, + 0.14)	0.90 : (- 0.18, + 0.37)	0.90 : (- 0.09, + 0.52)
	(W)LAN 3	0.90 : (- 0.06, + 0.46)	0.90 : (- 0.13, + 0.44)	0.90 : (- 0.31, + 0.22)
	(W)LAN 4	0.90 : (- 0.39, + 0.23)	0.99 : (- 1.39, - 0.42)	0.99 : (- 1.25, - 0.40)
	(W)LAN 5	0.90 : (- 0.36, + 0.22)	0.99 : (- 1.53, - 0.60)	0.99 : (- 1.47, - 0.52)
	(W)LAN 6	0.90 : (- 0.57, + 0.12)	0.99 : (- 1.31, - 0.35)	0.99 : (- 1.19, - 0.03)
$\bar{T}_2$ of TCP connections [1 - $\alpha$ : conf()]	(W)LAN 1	0.90 : (- 0.18, + 0.06)	0.90 : (- 0.17, + 0.03)	0.90 : (- 0.13, + 0.11)
	(W)LAN 2	0.90 : (- 0.15, + 0.12)	0.90 : (- 0.11, + 0.15)	0.90 : (- 0.11, + 0.18)
	(W)LAN 3	0.90 : (- 0.08, + 0.16)	0.90 : (- 0.07, + 0.18)	0.90 : (- 0.11, + 0.14)
	(W)LAN 4	0.90 : (- 0.08, + 0.22)	0.90 : (- 0.26, + 0.07)	0.90 : (- 0.32, - 0.02)
	(W)LAN 5	0.90 : (- 0.10, + 0.20)	0.90 : (- 0.27, + 0.06)	0.90 : (- 0.32, + 0.02)
	(W)LAN 6	0.90 : (- 0.26, + 0.14)	0.90 : (- 0.22, + 0.17)	0.90 : (- 0.19, + 0.26)
mean number of packet losses in base station [1 - $\alpha$ : conf()]	(W)LAN 1	0.90 : (- 0.00, + 0.00)	0.90 : (- 0.00, + 0.00)	0.90 : (- 0.00, + 0.00)
	(W)LAN 2	0.90 : (- 0.00, + 0.00)	0.90 : (- 0.00, + 0.00)	0.90 : (- 0.00, + 0.00)
	(W)LAN 3	0.90 : (- 0.00, + 0.00)	0.90 : (- 0.00, + 0.00)	0.90 : (- 0.00, + 0.00)
	(W)LAN 4	0.99 : (+ 1329.62, + 1923.98)	0.99 : (+ 2296.08, + 2866.12)	0.99 : (+ 845.53, + 1063.07)
	(W)LAN 5	0.99 : (+ 1406.58, + 1964.62)	0.99 : (+ 2329.37, + 2867.23)	0.99 : (+ 814.12, + 1011.28)
	(W)LAN 6	0.99 : (+ 1384.33, + 1955.57)	0.99 : (+ 2283.15, + 2839.85)	0.99 : (+ 802.83, + 980.27)

Table B.2: Statistical evaluation of the simulation results of scenario 2

	standard TCP ↔ TCP+EWA	standard TCP ↔ TCP+FEWA	TCP+EWA ↔ TCP+FEWA	
$\bar{T}_1$ of TCP connections [1 - $\alpha$ : conf()]	(W)LAN 1	0.90 : (- 0.37, + 0.22)	0.90 : (- 0.52, + 0.07)	0.90 : (- 0.44, + 0.14)
	(W)LAN 2	0.99 : (- 0.54, + 0.39)	0.90 : (- 0.41, + 0.03)	0.90 : (- 0.51, + 0.04)
	(W)LAN 3	0.90 : (+ 0.04, + 0.56)	0.90 : (- 0.27, + 0.34)	0.90 : (- 0.56, + 0.03)
	(W)LAN 4	0.90 : (- 0.21, + 0.74)	0.99 : (- 1.74, - 0.23)	0.99 : (- 1.99, - 0.51)
	(W)LAN 5	0.90 : (- 0.33, + 0.61)	0.99 : (- 2.19, - 0.58)	0.99 : (- 2.39, - 0.65)
	(W)LAN 6	0.90 : (- 0.42, + 0.53)	0.99 : (- 2.05, - 0.45)	0.99 : (- 2.03, - 0.58)
$\bar{T}_2$ of TCP connections [1 - $\alpha$ : conf()]	(W)LAN 1	0.90 : (- 0.13, + 0.15)	0.90 : (- 0.27, + 0.04)	0.90 : (- 0.28, + 0.04)
	(W)LAN 2	0.90 : (- 0.16, + 0.11)	0.90 : (- 0.21, + 0.02)	0.90 : (- 0.21, + 0.07)
	(W)LAN 3	0.90 : (+ 0.02, + 0.25)	0.90 : (- 0.14, + 0.18)	0.90 : (- 0.28, + 0.04)
	(W)LAN 4	0.90 : (- 0.05, + 0.52)	0.90 : (- 0.16, + 0.39)	0.90 : (- 0.40, + 0.16)
	(W)LAN 5	0.90 : (+ 0.00, + 0.57)	0.90 : (- 0.36, + 0.29)	0.90 : (- 0.64, + 0.01)
	(W)LAN 6	0.90 : (- 0.15, + 0.40)	0.90 : (- 0.37, + 0.27)	0.90 : (- 0.46, + 0.11)
mean number of packet losses in base station [1 - $\alpha$ : conf()]	(W)LAN 1	0.90 : (- 0.00, + 0.00)	0.90 : (- 0.00, + 0.00)	0.90 : (- 0.00, + 0.00)
	(W)LAN 2	0.90 : (- 0.00, + 0.00)	0.90 : (- 0.00, + 0.00)	0.90 : (- 0.00, + 0.00)
	(W)LAN 3	0.90 : (- 0.00, + 0.00)	0.90 : (- 0.00, + 0.00)	0.90 : (- 0.00, + 0.00)
	(W)LAN 4	0.99 : (+ 9022.65, +11386.65)	0.99 : (+13670.93, +16010.47)	0.99 : (+ 4394.15, + 4877.95)
	(W)LAN 5	0.99 : (+ 9493.13, +11486.17)	0.99 : (+14211.41, +16181.69)	0.99 : (+ 4478.07, + 4935.73)
	(W)LAN 6	0.99 : (+ 9035.85, +11673.05)	0.99 : (+13677.97, +16283.13)	0.99 : (+ 4344.95, + 4907.25)

## Appendix C

# Parameters of Fuzzy Explicit Window Adaptation (FEWA)

In this appendix, I present the linguistic rules of the FEWA algorithm. Furthermore, I state the chosen parameters of the membership functions  $m_{\Delta Q}$  and  $m_{\Delta G}$  of the linguistic variables  $\Delta Q$  and  $\Delta G$ .

### C.1 Linguistic Rules of FEWA

*If the queue (length) is empty,  
then the utilization factor should be very high.* (R1)

*If the queue (length) is short and  
the rate of change is decreasing fast,  
then the utilization factor should be high.* (R2)

*If the queue (length) is short and  
the rate of change is decreasing slowly,  
then the utilization factor should be high.* (R3)

*If the queue (length) is short and  
the rate of change is zero,  
then the utilization factor should be high.* (R4)

*If the queue (length) is short and  
the rate of change is increasing slowly,  
then the utilization factor should be high.* (R5)

*If the queue (length) is short and  
the rate of change is increasing fast,  
then the utilization factor should be medium.* (R6)

*If the queue (length) is moderate and  
the rate of change is decreasing fast,  
then the utilization factor should be high.* (R7)

*If the queue (length) is moderate and  
the rate of change is decreasing slowly,  
then the utilization factor should be medium.* (R8)

*If the queue (length) is moderate and  
the rate of change is zero,  
then the utilization factor should be medium.* (R9)

*If the queue (length) is moderate and  
the rate of change is increasing slowly,  
then the utilization factor should be medium.* (R10)

*If the queue (length) is moderate and  
the rate of change is increasing fast,  
then the utilization factor should be little.* (R11)

*If the queue (length) is long and  
the rate of change is decreasing fast,  
then the utilization factor should be little.* (R12)

*If the queue (length) is long and  
the rate of change is decreasing slowly,  
then the utilization factor should be little.* (R13)

*If the queue (length) is long and  
the rate of change is zero,  
then the utilization factor should be very little.* (R14)

*If the queue (length) is long and  
the rate of change is increasing slowly,  
then the utilization factor should be very little.* (R15)

*If the queue (length) is long and  
the rate of change is increasing fast,  
then the utilization factor should be very little.* (R16)

*If the queue (length) is full and  
the rate of change is decreasing fast,  
then the utilization factor should be very little.* (R17)

*If the queue (length) is full and  
the rate of change is decreasing slowly,  
then the utilization factor should be very little.* (R18)

*If the queue (length) is full and  
the rate of change is zero,  
then the utilization factor should be very little.* (R19)

*If the queue (length) is full and  
the rate of change is increasing slowly,  
then the utilization factor should be very little.* (R20)

*If the queue (length) is full and  
the rate of change is increasing fast,  
then the utilization factor should be very little.* (R21)

*If the queue (length) is congested,  
then the utilization factor should be very very little.* (R22)

## C.2 Parameters of FEWA

Each row  $k$  of the following tables C.1, C.2 shows the angle points  $(x_{k,1}; y_{k,1}), \dots, (x_{k,4}; y_{k,4})$  of the membership function  $m_{v_k}$  of the linguistic value  $v_k$ ,  $1 \leq k \leq n_v$ .

Table C.1: Parameters of the linguistic variable  $\Delta Q = Q/QT$  (see Figure 3.1)

$k$	$m_{\Delta Q_k}$
1	( 0.00; 1), ( 0.00; 1), ( 0.20; 1), ( 0.40; 0)
2	( 0.30; 0), ( 0.50; 1), ( 0.80; 1), ( 0.90; 0)
3	( 0.80; 0), ( 0.90; 1), ( 1.10; 1), ( 1.20; 0)
4	( 1.10; 0), ( 1.20; 1), ( 1.40; 1), ( 1.60; 0)
5	( 1.50; 0), ( 1.70; 1), ( 1.90; 1), ( 2.00; 0)
6	( 1.90; 0), ( 2.00; 1), ( 2.00; 1), ( 2.00; 1)

Table C.2: Parameters of the linguistic variable  $\Delta G = G/B$  (see Figure 3.2)

$k$	$m_{\Delta G_k}$
1	( -1.00; 1), ( -1.00; 1), ( -0.20; 1), ( -0.15; 0)
2	( -0.20; 0), ( -0.15; 1), ( -0.10; 1), ( -0.05; 0)
3	( -0.10; 0), ( -0.05; 1), ( 0.05; 1), ( 0.10; 0)
4	( 0.05; 0), ( 0.10; 1), ( 0.15; 1), ( 0.20; 0)
5	( 0.15; 0), ( 0.20; 1), ( 1.00; 1), ( 1.00; 1)

Notice:

- If  $(x_{k,1}; y_{k,1})$  is equal to  $(x_{k,2}; y_{k,2})$ , then  $m_{v_k}(x) = y_{k,1}$  for all  $x \leq x_{k,1}$ .
- If  $(x_{k,3}; y_{k,3})$  is equal to  $(x_{k,4}; y_{k,4})$ , then  $m_{v_k}(x) = y_{k,4}$  for all  $x \geq x_{k,4}$ .

### C.3 Selection of $\alpha_i$ 's for Different Maximum Queue Lengths

In this section, a rule of thumb for the calculation of the  $\alpha_i$ 's will be derived that can be used for queues with a maximum queue length different from  $B = 99$ . One assumption for this rule of thumb is that only  $B$  is changed. All other parameters, linguistic variables, and linguistic rules of FEWA are unchanged, at least in relation to the new maximum queue length  $B'$ .

Let  $\alpha = (\alpha_1, \dots, \alpha_6)$  be the parameter set for the FEWA fuzzy controller of a queue with a maximum queue length  $B$ , e.g.,  $B = 99$ . For another queue with a maximum queue length  $B'$ , e.g.,  $B' = 999$ , the parameter set  $\alpha' = (\alpha'_1, \dots, \alpha'_6)$  of this queue is selected that the control surface of the new FEWA fuzzy controller matches the control surface shown in Figure 3.3. Thus,

$$\alpha \cdot \log_2(B - Q) = \alpha' \cdot \log_2(B' - Q') \quad (\text{C.1})$$

for comparable current queue lengths  $Q = f_m \cdot B$  and  $Q' = f_m \cdot B'$ ,  $0 \leq f_m \leq 1$ , expressed using the maximum queue length or  $Q = f_t \cdot QT$  and  $Q' = f_t \cdot QT'$ ,  $0 \leq f_t \leq T = B/QT = B'/QT'$ , expressed using the target queue length. Then

$$\alpha \cdot \log_2((1 - f_m) \cdot B) = \alpha' \cdot \log_2((1 - f_m) \cdot B') \quad (\text{C.2})$$

This is equivalent—at least for queue lengths where only one membership function applies—to

$$\alpha_k \cdot \log_2((1 - f_k) \cdot B) = \alpha'_k \cdot \log_2((1 - f_k) \cdot B') \quad 1 \leq k \leq 6 \quad (\text{C.3})$$

with  $x_{k,1} \cdot T^{-1} \leq f_k \leq x_{k,4} \cdot T^{-1}$  (cf. Table C.1). It follows that

$$\alpha'_k = \frac{\log_2((1 - f_k) \cdot B)}{\log_2((1 - f_k) \cdot B')} \cdot \alpha_k = \frac{\log_2(1 - f_k) + \log_2(B)}{\log_2(1 - f_k) + \log_2(B')} \cdot \alpha_k \quad 1 \leq k \leq 6 \quad (\text{C.4})$$

But this equation has no unique solution for possible values of  $f_k$  if  $B$  and  $B'$  differ. What can be done is to solve this equation by selecting a single value as a representative for every validity interval of a membership function, e.g.,

$$f_k^* = \frac{x_{k,2} + x_{k,3}}{2} \cdot T^{-1} \quad 1 \leq k \leq 6, \quad (\text{C.5})$$

and calculate an approximation of

$$\alpha'_k \approx \frac{\log_2(1 - f_k^*) + \log_2(B)}{\log_2(1 - f_k^*) + \log_2(B')} \cdot \alpha_k = \theta'_k \cdot \alpha_k \quad 1 \leq k \leq 6 \quad (\text{C.6})$$

for this membership function by using the single value. This is the rule of thumb.

### C.3.1 Example: Persistent and WWW Traffic Traversing a Single Bottleneck Router

For a maximum queue length  $B = 99$  the parameter set  $\alpha = (1, 2, 4, 6, 9, 15)$  has been identified as a good choice. Based on this result, the parameter set of a queue with a maximum queue length  $B' = 999$  is  $\alpha' = (0.66, 1.31, 2.60, 3.87, 5.70, 9.42)$ .

The following Figure C.1 shows the histograms of the queue with  $B = 999$  of a highly loaded router for the two RCF approaches EWA and FEWA compared to standard TCP. This example simulation scenario considers a traffic mixture of persistent and WWW-based TCP connections which is related to the simulation scenarios considered in [9, 10]. But even in this good-case scenario for EWA, FEWA is able to outperform EWA with a performance gain of more than 25 % considering the overall mean throughput.

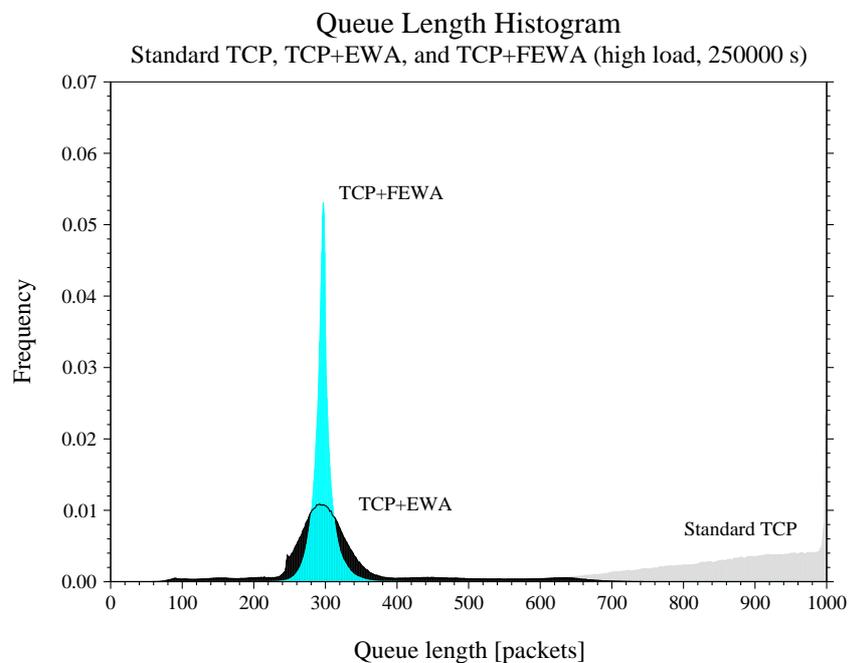


Figure C.1: Queue length process in a highly loaded bottleneck router for standard TCP, TCP+EWA, and TCP+FEWA

# Acknowledgments

The work presented in this technical report has been supported by a research contract with Ericsson Eurolab, Aachen, Germany.

Many thanks to Cornelius Mahlo (mahlo@cs.tu-berlin.de) who has implemented the traffic models for ns-2 (version ns-2.1b9). In addition, together with me he is the developer of a ns-2 framework in which the performance of different congestion feedback mechanisms in routers, e.g., EWA, FEWA, or XCP, can be investigated and evaluated under various network conditions.

# Acronyms

**ABR** Available Bit Rate

**AIMD** Additive Increase Multiplicative Decrease

**ARWND** Advertized Receiver Window

**ATM** Asynchronous Transfer Mode

**CBR** Constant Bit rate

**COG** Center-of-Gravity

**CWND** Congestion Window Size

**EC** Efficiency Controller

**ECN** Explicit Congestion Notification

**EWA** Explicit Window Adaptation

**FCC** Fuzzy Congestion Controller

**FERM** Fuzzy Explicit Rate Marking

**FERMA** FERM Adaptation

**FERMAM** FERMA-Modification

**FEWA** Fuzzy Explicit Window Adaptation

**FLC** Fuzzy Logic Controller

**IP** Internet Protocol

**MSS** Maximum Segment Size

**RCF** Router Congestion Feedback

**RED** Random Early Detection

**RTT** Round Trip Time

**TCP** Transmission Control Protocol

**UDP** User Datagram Protocol

**UMTS** Universal Mobile Telecommunication System

**WCDMA** Wideband Code Division Multiple Access

**WWW** World Wide Web

**XCP** Explicit Control Protocol

## Bibliography

- [1] M. Allman, S. Floyd, and C. Partridge. Increasing TCP's initial window. RFC 3390, October 2002.
- [2] M. de Pruecker. *Asynchronous Transfer Mode - Die Lösung für Breitband-ISDN*, volume 2. Prentice Hall, 1994.
- [3] K. Fall and K. Varadhan. *The ns Manual*. VINT Project, [http://www.isi.edu/nsnam/ns/doc/ns\\_doc.pdf](http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf), 2001.
- [4] S. Floyd and T. Henderson. The NewReno modification to TCP's fast recovery algorithm. RFC 2582, April 1999.
- [5] The ATM Forum. Traffic management specification version 4.1. AF-TM-0121.000, March 1999.
- [6] T. Goff, J. Moronski, D. S. Phatak, and V. Gupta. Freeze-TCP: A true end-to-end TCP enhancement mechanism for mobile environments. In *Proceedings of IEEE INFOCOM'00*, pages 1537–1545, 2000.
- [7] R. Jain. *The Art of Computer Systems Performance Analysis*. Wiley & Sons, 1991.
- [8] R. Jain. Congestion control and traffic management in ATM networks: Recent advances and a survey. *Computer Networks and ISDN Systems*, 28(13):1723–1738, 1996.
- [9] L. Kalampoukas, A. Varma, and K. K. Ramakrishnan. Explicit window adaptation: A method to enhance TCP performance. In *In Proceedings of IEEE INFOCOM'98*, pages 242–251, 1998.
- [10] L. Kalampoukas, A. Varma, and K. K. Ramakrishnan. Explicit window adaptation: A method to enhance TCP performance. *IEEE/ACM Transactions on Networking*, 10(3):338–350, June 2002.
- [11] D. Katabi, M. Handley, and C. Rohrs. Congestion control for high bandwidth-delay product networks. In *Proceedings of ACM SIGCOMM'02*, pages 89–102, August 2002.

- [12] C. C. Lee. Fuzzy logic in control systems: Fuzzy logic controller—part I. *IEEE Transactions on Systems, Man and Cybernetics*, 20(2):404–418, March 1990.
- [13] C. C. Lee. Fuzzy logic in control systems: Fuzzy logic controller—part II. *IEEE Transactions on Systems, Man and Cybernetics*, 20(2):419–435, March 1990.
- [14] A. Pitsillides, Y. A. Şekercioglu, and G. Ramamurthy. Effective control of traffic flow in ATM networks using fuzzy explicit rate marking (FERM). *IEEE Journal on Selected Areas in Communications*, 15(2):209–225, 1997.
- [15] K. Ramakrishnan, S. Floyd, and D. Black. The addition of explicit congestion notification (ECN) to IP. RFC 3168, September 2001.
- [16] A. Reyes-Lecuona, E. González-Parada, E. Casilari, and J. Casasola. A page-oriented WWW traffic model for wireless system simulations. In *Proceedings ITC 16*, pages 1271–1280, 1999.
- [17] M. Savorić. Untersuchung von Überlastabwehrverfahren für ABR-Dienste in drahtlosen ATM-Netzen. Master’s thesis, University of Frankfurt, September 1998.
- [18] M. Savorić. Distributed congestion management approaches in IP-based networks — first performance results for WCDMA links. project report EF-TUB-D8, December 2003.
- [19] M. Savorić and U. R. Krieger. Adaptation of the fuzzy explicit rate marking to ABR flow-control in a wireless ATM network. In *Proc. of MMB99*, pages 25–30, September 1999.
- [20] M. Savorić and U. R. Krieger. The performance of a fuzzy flow-control scheme for WWW data transfer in a wireless ATM network. In *Proc. of ITC Specialist Seminar on Mobile Systems and Mobility*, pages 145–156, March 2000.
- [21] M. Savorić, U. R. Krieger, and A. Wolisz. The impact of handover protocols on the performance of ABR flow-control algorithms in a wireless ATM network. *European Transactions on Telecommunications (ETT)*, 11:419–430, August 2000. Special Issue on Service Quality Control in Multimedia Wireless Networks.
- [22] G. Siegmund. *ATM – Die Technik des Breitband-ISDN*, volume 2. R. v. Decker, 1994.