BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS
BUDAPEST, HUNGARY

# FAIRNESS OF TRANSPORT AND RESOURCE MANAGEMENT PROTOCOLS

Mohamed Muftah Tekala

Summary of Ph.D. Dissertation

Supervised by:
Róbert Szabó, Ph.D.

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
AT
High Speed Networks Laboratory
Dept. of Telecommunications and Media Informatics
Budapest University of Technology and Economics
2007

# 1 Introduction

With the extensive advances in communication technologies and network computation, the Internet has become essential for information exchange around the world. Internet is now used by hundreds of millions of people all over the world and this number of users is continuously increasing. The success of the Internet is based on the behavior of transport control protocol (TCP) and user datagram protocol (UDP).

The original specification for TCP was proposed in 1981 [RFC81], the basic idea behind TCP congestion control is to control network load by having sources adjust their rates according to the level of congestion in the network. While significant enhancements have been proposed and made over the years, its basic principles remain unchanged (i.e., flow control is performed at the end hosts).

The original TCP congestion control was called TCP Reno [Jac90], it is a reliable transport protocol that is tuned to perform well in traditional networks. However, several experiments and analysis have shown that this protocol is not suitable for bulk data transfer in high-bandwidth, large round trip time (RTT) networks because of its slow start and conservative congestion control mechanism. Hence, several alternative congestion control algorithms have been proposed, for instance, High-Speed TCP (HSTCP) [Flo03], TCP Westwood [MCG+01], Fast TCP [CJL04], and Scalable TCP (STCP) [Kel03].

STCP is a simple sender-side alteration to the TCP congestion window update algorithm. It offers a robust mechanism to improve performance in high-speed wide-area networks using traditional TCP receivers. STCP uses fixed increase and decrease parameters to update its congestion window. Hence, the performance of STCP has problems in achieving full utilization of the bandwidth at the start time when it has a long-delay connection, this behavior also gives rise to the unfairness problem of STCP.

In addition to TCP congestion control, the Internet QoS has been one of the most challenging topics of network research. This is because of the diversity of current Internet applications, ranging from simple tasks such as e-mail to demanding real-time applications such as IP telephony, which impose an increasing demand for better performance on the Internet. The IP technology proposed IP-based solutions for wireless networks, like IP-based Radio Access Networks (RAN). These networks impose QoS requirements such as fast dynamic resource reservation, simplicity, low cost, severe congestion handling and easy implementation, along with good scalability properties.

The Internet Engineering Task Force (IETF) standardization body has set up a working group (WG), called Next Steps In Signaling (NSIS) to specify and develop new types of QoS signaling solutions to meet real-time application requirements and the QoS demands imposed by the IP-based wireless networks. Several resource reservation mechanisms defined in the context of IP networks might be used as input to this WG. The most promising are RSVP (resource reservation protocol)[HJ97], RSVP aggregation[FD01], Boomerang [FNM+99], YESSIR (YEt another Sender Session Internet Reservation) [PS98], Feedback control extension to differentiated services [CLG99], Dynamic packet states [SZS+99], and Dynamic Reservation Protocol (DRP) [WC98].

The resource management on differentiated service (RMD) framework was introduced [WJK+02, Jac01, RW01, DT01] to provide the dynamic resource management and admission control in the differentiated service (DiffServ) domain. It is well known that longer path connections suffer from higher connection blocking if there is more than one bottleneck link in their paths. Together, with cross-traffic, this can result in highly uneven blocking ratios among connections of different path lengths.

Based on the above discussion, the dissertation is organized in two parts. In the first part, I have introduced a mathematical analysis of the STCP performance as well as their impact on the performance of the Internet as a whole in order to design more efficient traffic control mechanisms for the Internet. Unfortunately, my analysis of STCP shows that the throughput of a connection is inversely proportional to its round-trip time (RTT). This behavior is the source of the STCP unfairness problem. Hence, I have proposed a formula for the increase/decrease parameters that will fairly share bandwidth among heterogeneous flows.

The second part of the dissertation presents fairness extensions to the QoS framework, called RMD [SW01]. Fairness is investigated with respect to blocking of connections of different path lengths. With RMD, resource reservation request (Request) packets must be transmitted for each arriving connection, which is subject to connection admission control. Refresh packets (Refresh) must be sent periodically during the life-time of the connection if admitted into the system. Connection rejection is performed by marking the request packets, which regardless of their marking, travel all way to the destination and back. The blocking probability of each flow competing on bottleneck link must be the same, resulting in fair distribution of the flows in the bottleneck link.

# 2 Research Objectives

The objectives of this dissertation are twofold: $i)$ to improve the performance of STCP on different networking environments, and $ii)$ to enhance the fairness of resource management on differentiated service networks (RMD) protocol.

STCP wise, my primarily goal was to improve its intra protocol fairness by optimising its protocol parameters.

In the case of the RMD protocol my goal was to research for methods to balance blocking over different path-length resource reservations.

# 3 Methodology

To achieve the goals mentioned above, a combination of mathematical modeling and network simulations were applied. The dissertation also proposes a modification to the STCP, in order to overcome the problems identified during the analysis of the protocol. I have also used network simulation [nsn] to verify the analysis and the proposed modifications.

For the improvement of STCP I have created a comprehensive mathematical analysis to evaluate the performance of the protocol in different scenarios. I have analytically derived formulas for optimal parameter settings of the protocol. Results were also verified by numerical simulations.

For the improvement of RMD, I have modified the protocol operation and defined a new call admission control formula, which incorporates additional architectural components available in the TCP/IP protocol family. The modifications were analyzed and verified by simulations.

# 4 New Results

The new results are centered upon two fields of research: Thesis 1 and 2 are concerned with STCP performance and improvement, while Thesis 3 addresses the fairness problem of Resource Management in Differentiated Services (RMD) framework.

## 4.1 Scalable TCP Friendliness to NewReno TCP

The aim of this research is to study the effect of STCP on the NewReno TCP flows when they share a bottleneck link. In order to understand the relationship between throughput of STCP and NewReno TCP in case of packet drop due to buffer overflow, I conducted a mathematical analysis that predict the performance of the two TCP algorithms sharing bottleneck link. To accomplish this, a system model needs to be introduced first:

- Data sources are infinite, and there are always data to be sent in packets of the size $MSS$.
- There is a single bottleneck with a capacity of $\mu$ packets per second and buffer size $B$ (in packets).
- Two streams share the bottleneck link. The first one is STCP with a congestion window denoted by $W_S$. The second is NewReno TCP, with a congestion window denoted by $W_R$.
- RTT is defined as the sum of two parts. First is the fixed (propagation) delay $T$, which includes all fixed delays for processing, transmission, and propagation. A variable

delay denoted by $D(t)$ is also introduced to represent the random queuing delay (service time); the total RTT is thus $T + D(t)$, or $T_i + D(t)$ for the $i$th connection when the distinction is necessary.

- Congestion avoidance phase was divided into cycles, where the cycle of the period $k$ is the duration between two consecutive packet losses ($\tau_l[k]$).

- The increase and decrease parameters of STCP are denoted by $\alpha$ and $\beta$ respectively.

- Throughput $\lambda$ is the total amount of data sent through the network per unit time, while $\lambda_S$ and $\lambda_R$ are the throughput of the STCP and NewReno TCP streams, respectively.

Figure 1 illustrates the dynamics of the congestion window of both STCP and NewReno TCP as a function of time competing on bottleneck link under DropTail queuing system during slow start and congestion avoidance phases.

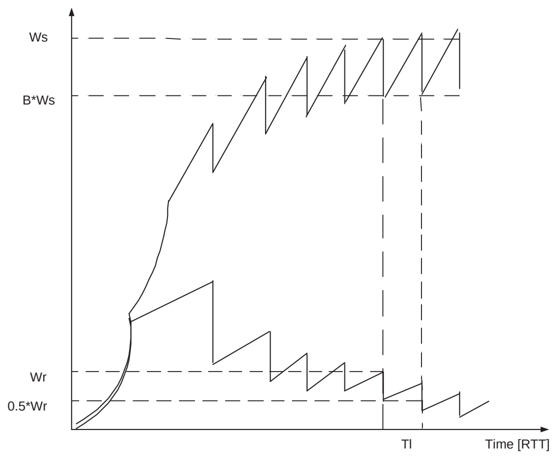



Figure 1: $W$ growth as a function of time

I present a simple approximation derivation of the model under the assumption that the congestion signal is subject to the buffer overflow (self-induced loss). The analysis presents an expressions for the throughput of the two TCP versions that sharing the bottleneck link. The throughput equations can be used to estimate the bandwidth share of each flow in the bottleneck link.

**THESIS 1** (STCP Friendliness to NewReno TCP)**:** *I have modeled the performance of Scalable TCP and NewReno TCP while sharing a bottleneck link. I have analyzed their achieved throughputs for both starting transients and steady-state in congestion avoidance regime. With the analysis I can predict the friendliness of STCP to NewReno TCP in buffer overflow (self-induced loss).*

### 4.1.1 Cycle Duration

In the DropTail queuing system the bottleneck link buffer is fully occupied and there are $\mu RTT$ packets in flight. The typical evolution is cyclical during the congestion avoidance phase. In this phase window size grows slowly to a maximum value $W_l$ at which point there is a packet loss. This is called self-induced loss, where connections on the network sharing bottleneck link suffers from this type of loss.

**THESIS 1.1** (Cycle Duration [**J1**])**:** *I have modeled the duration between two consecutive packet losses when both STCP and NewReno TCP connections sharing a bottleneck link (under the assumption of self-induced loss) by:*

$$\tau_l[k] = \frac{2(1-\beta)W_S[k] + W_R[k]}{\alpha(1+\beta)W_S[k] + 2} \cdot RTT. \tag{4.1}$$

The congestion window of NewReno TCP is controlled by the buffer overflow loss, so that, the evaluation of the performance of NewReno TCP is affected by STCP flow where the increase-decrease parameters make sense. The detailed description and validation of the model are presented in the dissertation.

### 4.1.2 Friendliness with Self-Induced Loss

NewReno TCP suffers from the self-induced loss when it shares a link with STCP. This is because in NewReno TCP the congestion window decreased to a half of its value in response to congestion event and its congestion window grows in increments of one packet for each RTT cycle. This leads to slow recovery from a congestion event when the congestion window is very large. In contrary, STCP functions with less decrease and faster increase in congestion window, which means less time for recovery process and as a consequence increase in average utilization.

**THESIS 1.2** (Friendliness with Self-Induced Loss [**J1**])**:** *I have modeled and analyzed STCP's friendliness to NewReno TCP under the effect of self-induced loss systems. I have shown that, in high-speed wide-area networks (i.e., probability of packet drop is very low) the more aggressive STCP completely squeezes out NewReno TCP from such systems. This is the result of the following obtained formulas for two connections during period $k$:*

$$\lambda_S[k] = \frac{(1+\beta)W_S[k]\ \tau_l[k]}{2\ RTT} \tag{4.2}$$

*where $W_S[k]$ depends on the increase/decrease parameters and the probability of packet drop $q$.*

$$\lambda_R[k] = \frac{7W_R[k]\ \tau_l[k]}{8\ RTT} \tag{4.3}$$

*also, $W_R[k]$ depends on the probability of packet drops $q$.*

The total throughputs of the two flows are given by the sum of the two TCP throughput Eqs. (4.2) and (4.3)

$$\lambda[k] = \lambda_S[k] + \lambda_R[k] \tag{4.4}$$

In most of the cycles $W_R[k] > W_R[k+1]$ and $W_S[k] < W_S[k+1]$, this was repeated until the the connections reach the steady state phase defined by:

$$k \Longrightarrow \infty \qquad \begin{array}{l} U_R[k] \approx 0 \\ U_S[k] \approx 1 \end{array}, \tag{4.5}$$

where $U_R[k]$ and $U_S[k]$ are the fraction of the bandwidth utilized by NewReno TCP and STCP respectively.

To obtain the fraction of the bandwidth applied by each TCP connection competing on a bottleneck link, I must substitute the increase and decrease parameters ($\alpha = 0.01$, $\beta = 0.875$ and the probability of packet drops $q$) into Eqs. (4.2), (4.3) and (4.4).

The details of the proof are provided in the dissertation.

**Validation** Simulation measurements were carried out in order to verify the model accuracy. Long lived TCP flows traveling along high speed-wide area networks were used. The results of the applied model and the performed simulation measurements were in good agreement.

## 4.2 Optimal Parameters for Scalable TCP

Using a complicated router can dramatically improve fairness in high-speed networks where synchronized losses are likely to occur. Such routers may implement AQM schemes such as adaptive RED, and perhaps a scheme to randomly drop packets as well. Simple routers using the DropTail queuing system, on the other hand, cannot ensure fairness amongst different TCP connections.

There has been much research on improving the performance of TCP in situations where the product of network bandwidth and delay is high (see related works in [Flo04, Jac88, KHR02, Kel03]).

STCP [Kel03] is one result of this work. The original STCP protocol sets the increase and decrease parameters to constant values. Constant values, however, cannot be optimal in every network environment. In long-delay connections, STCP is slow in achieving full utilization of the bandwidth. Moreover, under this protocol heterogeneous flows (i.e., flows with different round trip times) sharing the same bottleneck link will not receive equal portions of the available bandwidth [**C2**]. This fact conducted me to make a mathematical analysis of the protocol's performance.

**THESIS 2** (Optimal Parameters for STCP)**:** *By modeling and analyzing Scalable TCP I have optimized its parameter settings for fair bandwidth sharing and maximum throughput with different round trip time flows in DropTail and lossy link environments.*

My goal in analyzing STCP is to derive the functions for the increase $\alpha(\cdot)$ and decrease $\beta(\cdot)$ parameters that optimize throughput and fairness, in contrast to the approach of heuristically setting $\alpha$ and $\beta$ to constant values. In order to accomplish this, I first introduce a mathematical analysis for single and multiple STCP connection(s) with and without random losses.

### 4.2.1 STCP Model

Scalable TCP, just like most other TCPs, evolves in two distinct phases: i) a "slow start" phase, which begins at $W = 1$ and continues either until $W$ exceeds an initially chosen value of the threshold $ssthresh$ or until packet loss occurs; and ii) a congestion avoidance phase where the window increases more slowly until packet loss occurs. If packets are lost, then a recovery method is invoked where the congestion window is reduced to $\beta W^{\max}$ and the threshold is reset to $ssthresh = W^{\max}/2$. In this section I analyze the throughput and fairness of such STCP connections under buffer overflow packet loss.

An illustration of the dynamics of the congestion window of STCP as a function of time under DropTail queuing system during slow start and congestion avoidance phases is shown in Fig. 2.

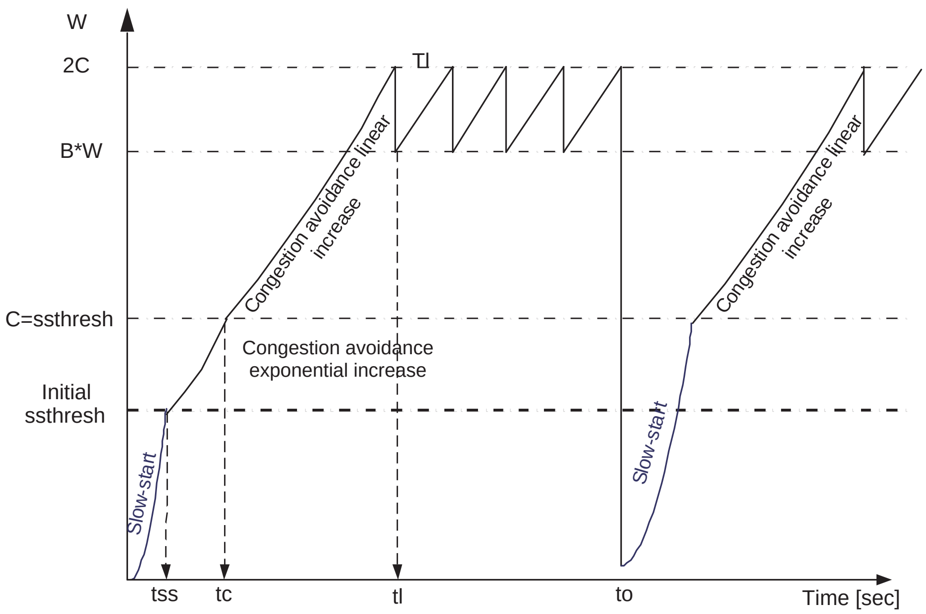



Figure 2: $W$ growth as a function of time

Let $dW/dt$ denote the rate of window growth with respect to time, $dW/da$ the window growth per arriving acknowledgment, and $da/dt$ the rate at which acknowledgments are arriving. A multiplicative increase phase, where $W = W + \alpha$ for each acknowledgment, can be described by the rates

$$dW/da = \alpha \tag{4.6}$$

and

$$\frac{da}{dt} = \begin{cases} W/T & \text{when } W \leq C \\ \mu & \text{when } W > C \end{cases}, \tag{4.7}$$

where $C = \mu T$ is the pipe size (capacity) of the connection, and $T$ is the round trip time. For the purposes of my analysis, this phase is further divided into a non-congested phase ($W \leq C$) and an accumulating backlog phase ($W > C$).

**THESIS 2.1** (STCP Model [**J2**, **J3**])**:** *I have introduced a mathematical model for STCP's congestion window in the congestion avoidance phase to describe its behavior over asymmetric networks, where the congestion avoidance phase is divided into two regimes:*

*$i$) Non-congested regime ($W \leq C$):*

$$W(t) = W(t_{ss})e^{\alpha\frac{t-t_{ss}}{T}}, \;\; t_{ss} \leq t < t_C, \tag{4.8}$$

*where $t_{ss}$ and $t_c$ are the times at which the congestion window attains the slow-start threshold and pipe capacity, respectively.*

*$ii$) Backlog accumulation regime ($W > C$):*

$$W(t) = C + \alpha\mu(t - t_c), \;\; t_c \leq t < t_l. \tag{4.9}$$

*I define $t_l$ as the value of the congestion window when the first loss happens; i.e., $W^{\max} = W(t_l)$.*

The length of the non-congested period is therefore

$$\tau_c = t_c - t_{ss} = \frac{T}{\alpha}\ln\left(\frac{C}{ssthresh}\right). \tag{4.10}$$

where $ssthresh$ is the Slow-Start threshold.

The backlog accumulation time, from the start of the regime to the first packet drop (recall that the bottleneck uses DropTail FIFO queuing), can be expressed as

$$\tau_l = \frac{W^{\max} - C}{\alpha\mu}. \tag{4.11}$$

The detailed description and validation of the model are presented in the dissertation.

### 4.2.2 Multi-flow STCP Throughput with Self-Induced Loss

The results of the single-flow analysis is adopted to the multi-flow scenario. My assumption is that self-induced losses will result in $\tau_i = \tau$ for all flows independently, which can be measured at each of the sources.

**THESIS 2.2** (STCP Throughput with Self-Induced Loss [**J2**, **J3**]): *Based on the defined regimes in Thesis 2.1, I have derived throughput formulas for the STCP connection under the effect of self-induced loss environment.*

*$i$) Exponential growth:*

$$\lambda_i(t) = \frac{W_{ss} e^{\alpha_i \cdot \frac{(t-t_{ss})}{T_i}}}{T_i}, \quad t_{ss} \leq t < t_C. \tag{4.12}$$

*It is clear that, the throughput is inversely proprtional to the round trip time.*

*$ii$) Linear growth:*

$$\lambda_i = \frac{\tau_l \tan(\alpha)(1+\beta)}{2(T_i + D)(1-\beta)}, \tag{4.13}$$

*where the duration between two consecutive losses ($\tau_l$) is the same for all TCP flows*[1]. $D$ *is the average queuing delay calculated as*

$$D = \frac{\beta \sum W_i^{max} - C + B}{2\mu}. \tag{4.14}$$

**Validation** The accuracy of my model is examined by extensive simulation measurements using the NS-2 tool. The results show a good match with the model. The details are provided in the dissertation.

### 4.2.3 Optimal Increase Parameter $\alpha(.)$ for STCP

As described in the previous section, the original STCP protocol sets the increase and decrease parameters to constant values. Constant values, however, cannot be optimal in every network environment. In long-delay connections, STCP is slow in achieving full utilization of the bandwidth. Moreover, under this protocol heterogeneous flows sharing the same bottleneck link will not receive equal portions of the available bandwidth [**C2**]. This fact led me to investigate how different choices of the increase parameter $\alpha$ affect network performance.

**THESIS 2.3** (Optimal Increase Parameter [**J3**]): *Based on the self-induced loss throughput analysis (Thesis 2.2) I have proposed a new functions for the increase parameter $\alpha(.)$ which fairly shares the bandwidth among competing flows with different round trip times (RTT). Where the congestion window is divided into two regimes:*

*$i$) Exponential growth:*

$$\alpha_i(t) = \frac{T_i}{t - t_{ss}} \ln(T_i) + T_i, \quad t_{ss} \leq t < t_c. \tag{4.15}$$

---

[1] *This value can be measured distributedly.*

*To minimize losses during the exponential phase of the congestion window, an additional scale factor $K$ is introduced: $\alpha_i(T_i) = \alpha_i(T_i)/K$.*

$ii$*) Linear growth:*

$$\alpha_i(T_i) = \frac{\tan^{-1}(T_i + D)}{K}. \tag{4.16}$$

By substituting the derived increase parameter $\alpha(t)$ in Eq. (4.15) the throughput is independent of the the round trip time.

When the backlog accumulation regime starts, the increase parameter switches to Eq. (4.16). Under this condition the throughput formula yields

$$\lambda_i = \frac{\tau_l(1 + \beta_i)}{2K(1 - \beta_i)}. \tag{4.17}$$

Thus setting $\beta_i = \beta$ for all flows should fairly share the bandwidth.

The validation of the parameter is presented in Thesis 2.5.

### 4.2.4 Optimal Decrease Parameter $\beta(.)$ for STCP

By considering random losses in the network, one can investigate the $\beta$ parameter more deeply. Random losses occur due to non-congestion losses when the transmission rate is less than or equal to fair share (e.g., due to link failure, mis-routing etc.). TCP treats the loss of packets as a signal of network congestion and reduces its window when this occurs. The effect of random loss on STCP has been considered previously in [Kel03]. It shows that the congestion window is proportional to the increase parameter $\alpha$ and monotonic in the decrease parameter $\beta$, it is also inversely proportional to the loss probability $q$:

$$W_l = \frac{\alpha}{q(1 - \beta)}. \tag{4.18}$$

**THESIS 2.4** (Bounds on Multiplicative Decrease Parameter[**J3**])**:** *Taking in consideration random losses in the network, I have proposed an optimized setting for parameter $\beta$ within the range specified by Eqs. (4.19) and (4.20), which maximizes the achieved throughput of STCP:*

$$\beta \geq \frac{(T_i + D) \cdot K}{(T_i + D) \cdot K + \tau_l \tan^{-1}(T_i + D)} \tag{4.19}$$

*and*

$$\beta \leq \frac{2(T_i + D) \cdot K - \tau_l \tan^{-1}(T_i + D)}{2(T_i + D) \cdot K}. \tag{4.20}$$

If it is assumed that random loss occurs only when the link capacity is fully utilized (i.e., $W_l \leq \mu T + B$), it is desirable to ensure that $\beta W_l \geq \mu T$ to achieve high throughput.

The decrease parameter $\beta$ of a flow should thus be inversely proportional to the time between consecutive losses ($\tau_l$). So that, an efficient window decreasing ratio can maintain high utilization of the bottleneck link even though the probability of loss has increased.

### 4.2.5 Numerical Evaluation of Optimal Parameters for STCP

The proposed modified version of STCP will be referred to as Modified STCP, or MSTCP for short.

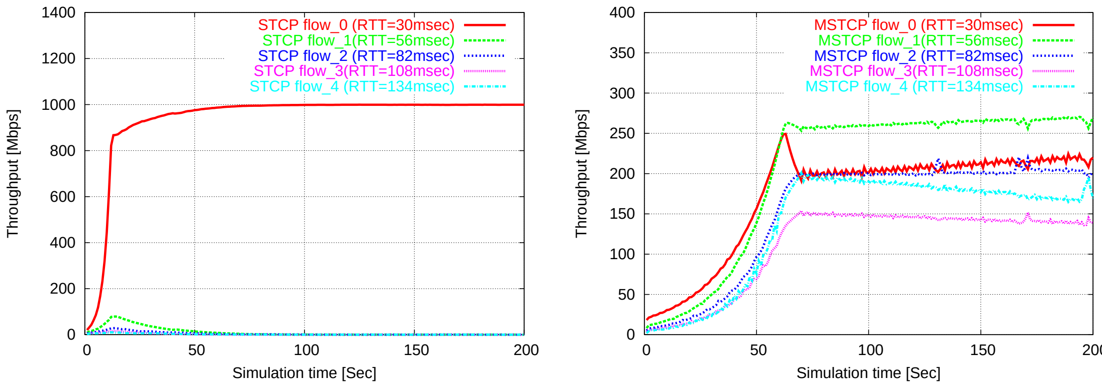
For my numerical analysis I used a dumbbell network with one bottleneck link. In all the scenarios described in the dissertation, the capacity of the shared link is either 1 Gbps or 100 Mbps. Link delays vary from scenario to scenario. The two routers use FIFO queuing and DropTail buffer management. Their buffers were set to the bandwidth-delay product of the network. The bandwidth between hosts and their routers was 10 Gbps, and the link delay between hosts and their routers was 1 ms. The packet size was set to 1500 bytes, and the maximum window size was large enough (83000 packets) to saturate the bottleneck. I used Selective Acknowledgment TCP (SACK) [MMFR96] for both clients and servers. In addition to the TCP agents I used an FTP application to transmit large datasets. All FTPs began transmitting simultaneously (at time zero), and the simulation ran for a total of 200 s unless indicated otherwise. The individual traffic mixing scenarios are discussed in detail in the dissertation.

**THESIS 2.5** (Numerical Validation [**J3**])**:** *I have numerically evaluated the proposed optimal parameter settings for STCP and showed that regardless of the model simplifications and approximations* $\alpha(.)$ *achieves fair bandwidth allocation and* $\beta(.)$ *maximized the achieved throughput.*

I simulated the activity of multiple flows with varying RTT values on the network. The RTTs were chosen according to the following formula[2]: $T_i = (i+1)T_{base} - 4i$ ms ($i = 0...n-1$, where $n$ is the number of flows). The scenario was carried out using the standard STCP algorithm as well as the proposed modified STCP algorithm under the DropTail queue management system.

Fig. 3(a) reveals that in standard STCP the shortest RTT connection far out-paces the others. With the proposed fairness modifications, however, the longer RTT connections obtain a significantly larger share of the bandwidth (Fig. 3(b)).

[2]without any specific reasons




(a) STCP (b) MSTCP

Figure 3: Throughput of five flows under DropTail queuing ($\mu$ =1 Gbps, $T_{base}$ =30 ms)

## 4.3 Fairness Improvement on RMD Framework

In addition to the basic best-effort service offered by the traditional Internet, Differentiated Service introduces two additional packet-forwarding mechanisms (Per-Hop Behaviors PHBs) called Expedited forwarding (EF) and Assured Forwarding (AF) [ea99]. Assured forwarding implementations succeed in providing minimum rate guarantees, but they do not achieve fair distribution of the network capacity among all users. Clark and Fang [CF98] proposed the Time Sliding Window Two Colour Marker (TSW2CM), which calculates the average rate of packet arrival. If the average rate is below the target rate, then the packet is marked as IN, otherwise it is marked as OUT. This can guarantee the target rates in under-subscribed networks, but suffers from unfairness problems. The Memory-Based Marker (MBM) [AJ01] proposes a simple marking algorithm for TCP aggregates that adjusts the marking probabilities based on rate changes, and comparison of the average rate with the target rate. MBM improves fairness in some cases, but in other cases its performance remains far from optimum. Nandy *et al.* [Nea00] proposed different intelligent traffic conditioners to deal with different causes of unfairness, which is thus not a general solution.

By the nature of resource reservation protocols, if a connection set-up must travel along many resource-scarce links, the probability of blocking is higher than for paths with fewer bottleneck links. This, however, would certainly result in lower blocking ratios for short connections and higher ratio for other connections. This can be viewed as an undesirable result of these protocols that may be overcome by introducing special treatment of longer-path flows. Hence, I proposed methods to ensure *fair* or balanced blocking over communication paths of different lengths in the RMD environment.

The advantage of this method is that it regulates the bandwidth share fairly among flows and improves link utilization.

**THESIS 3** (Fairness Issues for RMD): *I have proposed a methods for the resource management in DiffServ (RMD) to ensure fair (or balanced) blocking over different communication path length. This method improves fair network bandwidth allocation among different RMD connections.*

### 4.3.1 Flow Differentiation

For RMD, resource reservation request (Request) packets must be transmitted for each arriving connection, which is subject to connection admission control, and refresh packets (Refresh) must be sent periodically during the life-time of the connection if admitted into the system.

One way of differentiating the connection length is to use the TTL field encoded in the IP header. TTL is always set to an initial value (usually 32) at the time of packet creation and is automatically decreased by one for each hop the packet passes through. Packets are dropped when TTL reaches zero. The underlying reason behind this mechanism is to avoid communication loops in the connection-less Internet.

**THESIS 3.1** (Admitted Load[**C1**]): *I proposed a method to differentiate resource request based on the time to live (TTL) field encoded in the IP header. This is done by calculating the aggregated offered and admitted load for each connection type (path length) during each refresh period ($\tau$) within the interior nodes. The aggregated offered load of differentiated connections during refresh period $\tau$ can be calculated by*

$$\begin{aligned} A_{\mathrm{TTL}=i}[k] = \; & \textstyle\sum_{t\in[k\tau,(k+1)\tau)} \mathrm{Request}_{\mathrm{TTL}=i}(t) + \\ & \textstyle\sum_{t\in[k\tau,(k+1)\tau)} \mathrm{Refresh}_{\mathrm{TTL}=i}(t), \end{aligned} \tag{4.21}$$

*And the admitted (under service) volume during refresh period $\tau$ is,*

$$\begin{aligned} S_{\mathrm{TTL}=i}[k] = \; & \textstyle\sum_{t\in[k\tau,(k+1)\tau)} \mathrm{Request}^{+}_{\mathrm{TTL}=i}(t) + \\ & \textstyle\sum_{t\in[k\tau,(k+1)\tau)} \mathrm{Refresh}_{\mathrm{TTL}=i}(t), \end{aligned} \tag{4.22}$$

*where* $\mathrm{Request}^{+}$ *are successful reservations.*

Naturally, $\mathrm{Request}(.) = \mathrm{Request}^{+}(.) + \mathrm{Request}^{-}(.)$, Where, $\mathrm{Request}^{-}$ are blocked or rejected reservations. Connection rejection is performed by marking the request packets, which, regardless of their marking, travel all the way to the destination and back. Hence, the aggregated blocking resources of each connection type is easily calculated for each period $k$ according to the formula:

$$B_{\mathrm{TTL}=i}[k] = A_{\mathrm{TTL}=i}[k] - S_{\mathrm{TTL}=i}[k]. \tag{4.23}$$

To smooth the estimation, I used the exponential weighted moving averaging approach so that each update of the above variables was calculated according to the formula $x[k] = (1-\Omega)x`[k] + \Omega x[k-1]$, where the ` values were measured/counted.

### 4.3.2 Call Admision Control (CAC)

To address the unfair bandwidth allocation problem of different path length flows I propose an algorithm, which calculates blocking ratios of the different connection lengths from previous periods and sets target blocking for the next time frame, or more formally:

**THESIS 3.2** (Differential Call Admission Control [**C1**])**:** *Based on the offered load expression (Thesis 3.1), I have given a call admission control method which balance the blocking over different TTL flows by checking for every newly arriving flow's request (*$\mathrm{Request}_{\mathrm{TTL}=i}(t)$*):*

$$A_{TTL=i}[k+1] + \mathrm{Request}_{\mathrm{TTL}=i}(t) \leq S^*_{TTL=i}[k+1] \tag{4.24}$$

*for admittance, where* $S^*_{TTL=i}[k+1]$ *is the acceptance thresholds on each* $\mathrm{TTL}$ *class of the traffic at period* $k+1$*. The thresholds are calculated according to the following equations:*

$$S^*_{\mathrm{TTL}=i}[k+1] = C\frac{A_{\mathrm{TTL}=i}[k]}{\sum_j A_{\mathrm{TTL}=j}[k]}, \tag{4.25}$$

*where* $C$ *is the link capacity.*

Hence, the CAC function of Eq. (4.24) should balance the blocking probability over the competing flows:

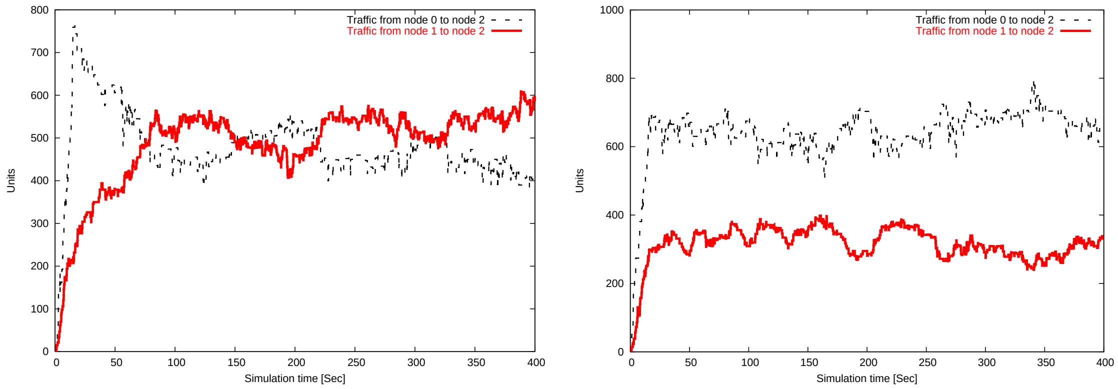
$$\frac{B_{\mathrm{TTL}=i}[k]}{A_{\mathrm{TTL}=i}[k]} = \frac{B_{\mathrm{TTL}=j}[k]}{A_{\mathrm{TTL}=j}[k]}, \quad j \neq i. \tag{4.26}$$

### 4.3.3 Algorithm Validation

For numerical results I used ns-2 simulator [nsn] and its RMD extension. Different scenarios were investigated to present insights into the performance of the proposed method. In all of my experiments I assumed Poisson call arrival model with exponential holding times. In most of the system were overload/congested in at least one of its links. In order to highlight the effect of different blocking ratios for different path length connection I aggregated the congestion on certain link in the network.

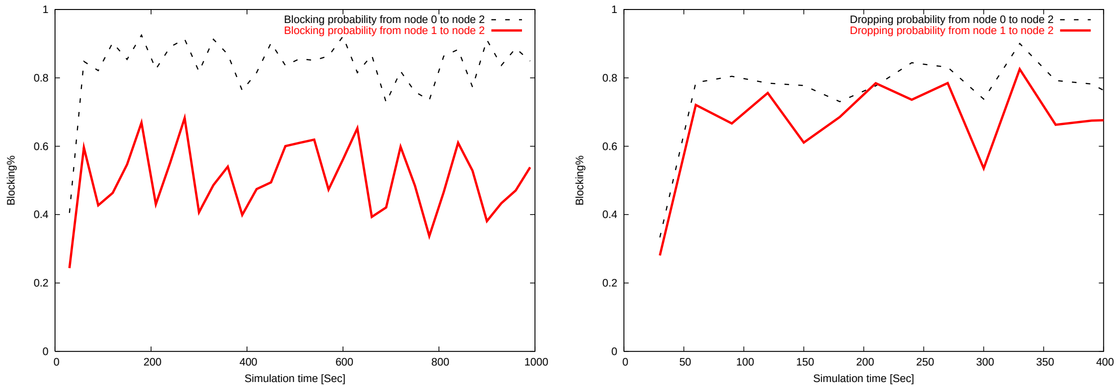
**THESIS 3.3** (Numerical Evaluation [**C1**])**:** *I have numerically validated the proposed RMD fairness method. Different scenarios were investigated to present some insights into the performance of the proposed method. The cascade network toplology showed that, the proposed method works well, where the competing RMD flows traveling through different number of hops. Moreover, my numerical results reveal the limitation of the proposed algorithm when merging paths with equal source TTL values were simulated.*

I simulated 3-nodes cascade network in case of standard RMD algorithm and my proposed modification. Here Node-0 and Node-1 generated traffic towards Node-2. Node-0 generated 3000 units of traffic while Node-1 generated 1000 units of traffic. Both of the internal links were set to 1000 units capacity. Obviously, both of the links acted as bottleneck in this scenario. However, the first link cut the source in Node-0 back to 1000 units, just as the source in Node-1, hence for link Node-1–Node-2 both sources offered virtually the same load. Therefore both shared the available link capacity evenly regardless of their original offered loads (see Fig. 4(a)). Also their respective blocking ratios were unbalanced due to the fact that the source in Node-0 blocked in two consecutive link (see Fig. 5(a)). However, by applying my method different TTL values and path lengths can be accounted for and the blocking ratios can be adjusted as connections passes through the network (see Fig. 4(b) and Fig. 5(b)).




(a) Standard RMD (b) Modified RMD

Figure 4: Admitted traffic per source




(a) Standard RMD (b) Modified RMD

Figure 5: Blocking of sources

# 5 Conclusions

The objectives of this dissertation was to improve the performance of STCP on different network environments, and the fairness of Resource Management on DiffServ networks.

The mathematical model presented in Thesis 1 gives accurate estimation of the steady state fraction of bandwidth saturated by both (Scalable and NewReno) TCP protocols when they compete on bottleneck link under self-induced losses. Both mathematical analysis and simulations show that, the STCP was the dominant protocol on the bandwidth when sharing bottleneck link with NewReno TCP.

Based on the throughput equations obtained by the mathematical analysis of STCP, in Thesis 2 I proposed a formula that dynamically optimize the congestion window increase and decrease parameters according to the measured RTT of the connections. Under this modification, the increase parameter assigns larger windows to connections with longer RTTs. As a result, better balance is achieved in the allocation of bandwidth among connections with different RTTs. Simulations on a dumbbell network show that this modification significantly improves the performance of STCP on connections with long delays.

In Thesis 3, I introduced a method to decrease the problem of unfair share of network bandwidth among different RMD connections. By using time to live (TTL) field to differentiate the flows passing through different hops, and according to the dynamics of RMD, the aggregated offered load of each connection type during refresh periods can be calculated. Accordingly, the probability of packet drop for different TTL connections are the same.

# 6 Applicability of the Results

My research work on transport control protocol proposed a model that predicts the performance of STCP and NewReno TCP competing on bottleneck link under the self-induced loss. The protocol was verfied by simulation results. Network engineers and researchers can use this algorithm to predict the performance of protocol on real networks.

In addition to the STCP friendliness model, I proposed modification that dynamically adapts the increase/decrease functions of STCP protocol. In order to improve the protocol's fairness. The model can be applied to the STCP protocol in real network connection, where the parameters of the increase/decrease functions are available (measured) during the STCP connection.

In the RMD framework long-path connections sever from sharing bottleneck link with short-path connections. I have proposed new call admission control algorithm that solves the problem of imbalance blocking of the flows through the interior nodes of the DiffServ domain. My model might be part of the RMD QoS algorithm, which is being standarised in the NSIS working group of the IETF. Where the NSIS working group is considering protocols for signaling information about a data flow along its path in the network.

# Acknowledgments

First of all, I would like to thank Róbert Szabó for his kindness, encouragement, guidance and support. Without his help this dissertation could have not been completed. The research presented in this thesis was carried out at High Speed Networks Laboratory (HSNLab). I wish to express my grateful thanks to the head of the laboratory, Dr Tamás Henk, for his support. The research work presented here was partially supported by the Inter-University Center for Telecommunications and Informatics (ETIK), Hungary. Moreover, I would like to express my appreciation and gratitude to the boss of the team Dr. Sándor Molnár and our team at the research center for their support.

The second part of the dissertation was done on the RMD extension developed at Ericsson Research Hungary. Furthermore, I would like to express my great thanks to András Császár for his help.

Last but not least, I would like to thank my family for their continual encouragement throughout the years. This work is dedicated to them.

# References

[AJ01] K.R.R. Kumar A.L. Ananda and A. Jacob. A memory-based approach for a TCP-friendly traffic conditioner in DiffServ networks. In *9th International Conference On Network Protocols*, 2001.

[CF98] D. Clark and W. Fang. Explicit allocation of best-effort packet delivery service. *IEEE/ACM Trans. Netw*, 6(4):362–372, 1998.

[CJL04] D. X. Wei C. Jin and Steven H. Low. FAST TCP: motivation, architecture, algorithms, performance. *IEEE INFOCOM*, 2004.

[CLG99] H. Chow and A. Leon-Garcia. A feedback control extension to differentiated services. Internet Draft, March 1999. <draft-chow-diffserv-fbctrl-01.txt>.

[DT01] A. Marquetant O. Pop R. Szabo G. Dinnyes and Z. Turanyi. Novel enhancements to load control a soft state lightweight admission control protocol. In *QofIS'2000 - 2nd International Workshop on Quality of future Internet Services*, September 2001.

[ea99] J. Heinanen et. al. Assured Forwarding PHB Group. RFC 2597, IETF, 1999.

[FD01] F. Baker C. Iturralde F. L. Faucheur and B. Davie. Aggregation of RSVP for IPv4 and IPv6 Reservation. RFC 3175, IETF, September 2001.

[Flo03] S. Floyd. HighSpeed TCP for large congestion windows. RFC 3649, IETF, Dec 2003.

[Flo04] S. Floyd. Limited slow-start for TCP with large congestion windows. RFC 3742, IETF, Mar 2004.

[FNM+99] G. Feher, K. Nemeth, M. Markosz, I. Cselenyi, J. Bergkvist, D. Ahlard, and T. Engborg. Boomerang - A Simple Protocol for Resource Reservation in IP Networks. In *IEEE Workshop on QoS Support for Real-Time Internet Applications*, June 1999. Vancouver, Canada.

[HJ97] R. Braden Ed. L. Zhang S. Berson S. Herzog and S. Jamin. Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification. RFC 2205, IETF, September 1997.

[Jac88] V. Jacobson. Congestion Avoidance and Control. *SIGCOMM 1988, conference*, August 1988.

[Jac90] V. Jacobson. Modified TCP Congestion Avoidance Algorithm. Technical report, 1990.

[Jac01] M. Jacobsson. Resource Management in Differiated Service - A Prototype Implementation. Master's thesis, Computer Science/TSS, University of Twente, June 2001.

[Kel03] Tom Kelly. Scalable TCP: Improving Performance In Highspeed Wide Area Networks. *ACM SIGCOMM Computer Communication Review, Volume 33, Number 2*, 2003.

[KHR02] Dina Katabi, Mark Handley, and Chalrie Rohrs. Congestion control for high bandwidth-delay product networks. *ACM Sigcomm 2002*, August 2002.

[MCG+01] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang. TCP Westwood bandwidth estimation for enhanced transport over wireless links. *presented at the Mobicom 2001, Rome, Italy*, Jul. 2001.

[MMFR96] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. *TCP Selective Acknowledgment Options*. RFC 2018, IETF, October 1996.

[Nea00] B. Nandy and et. al. Intelligent traffic conditioners for assured forwarding based differentiated services networks. In *IFIP High Performance Networking*, 2000.

[nsn] The network simulator ns 2. http://www.isi.edu/nsnam/ns/.

[PS98] Ping P. Pan and Henning Schulzrinne. YESSIR: A Simple Reservation Mechanism for the Internet. In *Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, pages 141–151, Jul 1998. Cambridge, England.

[RFC81] Transmission Control Protocol. RFC 793, DARPA, September 1981.

[RW01] G. Heijenk G. Karagiannnis V. Rexhepi and L. Westberg. DiffServ Resource Management in IP-based radio Access Networks. In *Wireless Personal Multimedia Communications (WPMC'01)*, Aalborg, Denmark, September 2001.

[SW01] L. Westber M. Jacobson G. Karagiannis S. Oosthoek D. Partain V. Rexhepi R. Szabo and P. Wallentin. Resource Management in DiffServ (RMD) framework. Internet-draft, IETF, April 2001.

[SZS+99] I. Stoica, H. Zhang, S. Shenker, R. Yavatkar, D. Stephens, A. Malis, Y. Bernet, Z. Wang, F. Baker, J. Wroclawski, C. Song, and R. Wilder. Per hop behaviours based on dynamic packet states. Internet Draft, February 1999. <draft-stoica-diffserv-dps-00.txt>.

[WC98] P. P. White and J. Crowcroft. A Dynamic Sender-Initiated Reservation Protocol for the Internet. In *Proceedings, HPN'98*, 1998.

[WJK+02] L. Westberg, M. Jacobsson, G. Karagiannis, M. de Kogel, S. Oosthoek, D. Partain, V. Rexhepi, and P. Wallentin. Resource Management in Diffserv On DemAnd (RODA) PHR. Internet draft, IETF, draft-westberg-rmd-od-phr-01.txt, February 2002.

# Publications


[J] **Journal Papers**

[**J1**] **Mohamed Tekala**, Róbert Szabó "Modeling Scalable TCP Friendliness to NewReno TCP", *IJCSNS International Journal of Computer Science and Network Security*, Vol.7 No.3. March 2007.

[**J2**] **Mohamed Tekala** and Róbert Szabó, "Throughput Analysis of Scalable TCP Congestion Control", *Periodica Polytechnica*, *Electrical Engineering*, To appear 2007

[**J3**] **Mohamed Tekala** and Róbert Szabó, "Dynamic Adjustment of Scalable TCP Congestion Control Parameters", *Computer Communication*, Under review. 2007.

[C] **Conference Papers**

[**C1**] **Mohamed Tekala** and Róbert Szabó, "Fairness Issues of Resource Management in DiffServ (RMD) Protocol", In Proceedings *EUNICE-2003*, Balatonfured, Hungary, 8-10 September 2003.

[**C2**] **Mohamed Tekala** and Róbert Szabó, "Evaluation of Scalable TCP", In Proceedings *The 3rd IEEE conference in Egypt (ACS-2005)*, Cairo, Egypt, January 7-11 2005.

[**C3**] **Mohamed Tekala** and Róbert Szabó, "Dynamic Adapting of Scalable TCP Congestion Control Parameters", In Proceedings *The 4th IEEE conference (AICCSA-2006)* Shargah, UAE, March 2006.