BUDAPEST UNIVERSITY OF TECHNOLOGY AND ECONOMICS
DEPT. OF TELECOMMUNICATIONS AND MEDIA INFORMATICS

A PERFORMANCE ANALYSIS OF SOME TRAFFIC CONTROL TECHNIQUES IN TCP/IP NETWORKS

Tuan Anh Trinh

Summary of the Ph.D. Dissertation

Supervised by

Dr. Sándor Molnár and Dr. László Gefferth
High Speed Networks Laboratory
Dept. of Telecommunications and Media Informatics
Budapest University of Technology and Economics

Budapest, Hungary
2004

1 Introduction

Starting from a relatively small network used by a relatively small research community in the United States, the Internet is now used by millions of people all over the world and this number is growing increasingly over the years. The success of the Internet, in which the TCP/IP protocol suite plays an important role, is based on its open, flexible but robust architecture.

It hasn't all been a happy story, nevertheless. An incident happened during the early growth phase of the Internet in mid 1980s brought the Internet down. The incident (technically called *congestion collapse*) was later explained by the lack of attention to the dynamics of packet forwarding. The lesson learnt from this incident is that we need to introduce some *transmission control* mechanisms into the design of the Internet. In this respect, the original fix to the congestion collapse of the Internet was provided by Van Jacobson in [Jac88] and the Transmission Control Protocol (TCP) was born. 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. Since then, a number of improvements have been added to the original TCP implementation, this basic idea remained unchanged. As a result, in order to understand the performance of the Internet we need to understand the dynamics and the performance of the TCP protocol which carries over 90% of the total Internet traffic today. Many results have been achieved in this field [MSMO97, PFTK98, VeBo00]. However, the TCP feedback scheme is still not well understood and more comprehensive investigation is still needed.

In addition, traffic in the Internet is composed of flows with different nature and different characteristics, as more and more new IP-based applications are brought into existence. Some of them are congestion-aware and some are not. As a consequence, end-to-end congestion control algorithms such as those in TCP are not enough to prevent congestion in the Internet, and they must be supplemented by control mechanisms inside the network. Since routers are the common points for all flows, it is reasonable to detect and control congestion at these points, at least globally. The Drop Tail buffer management scheme does little in this respect. To face this problem, a number of Active Queue Management schemes [FloJa93, FGS01, FKSS99, ALLY01, HMTG01] were introduced that can efficiently manage the buffers at the routers in order to avoid congestion, and in some cases, to guarantee fairness between competing flows. For example, one of the promising queue management schemes, the Random Early Detection (RED) scheme [FloJa93], was claimed in [FloJa93] to provide: congestion avoidance, appropriate time scales, no global synchronization, maximizing global power and fairness. However, the main problem with these schemes is that they are not yet thoroughly understood. Moreover, they usually have many parameters, and consequently, they are hard to tune [CJOS00].

To summarize the introduction, we argue that traffic control techniques, by keeping the rapidly growing Internet traffic within control, are indispensable and play a crucial role in the success of the Internet. As a result, there is a genuine need to have a better understanding of their 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.

2 Research Objectives

The objective of this dissertation is to provide a comprehensive performance analysis of some key traffic control techniques in TCP/IP networks.

The *first goal* of my research was to investigate the properties of the TCP protocol in different perspectives. The perspectives include the metrics of TCP, modelling of TCP traffic, and game-theoretic analysis of TCP. However, the dynamics of the Internet is not only based on the traffic generated at end-points but also on the queue managements schemes (such as those AQM schemes mentioned above) at the routers inside the networks. In this thesis, the RED mechanism is of particular interest because it's already implemented in a wide range of commercial routers (such as Cisco's routers). However, the performance of RED as well as the tuning of RED's parameters are still very problematic and open issues.

Based on the observations mentioned above, my *second goal* was to provide a comprehensive performance evaluation of the RED mechanism and to provide a way to tune its parameters in order to have better performance.

To sum up, I set the following objectives in my dissertation:

- Characterize the metrics of TCP.
- Provide a unified model for different versions of TCP.
- Investigate the rate control and parameter setting problems of TCP from a game-theoretic point of view.
- Evaluate the performance of the RED mechanism.

3 Methodology

To achieve the goals mentioned above, a combination of mathematical modelling, network simulations and network measurements were applied. The dissertation also proposes a number new characterization methods and control schemes. When investigating the proposals, I combined the techniques mentioned above whenever possible

to validate the results and to gain a better insight into the problem. For the evaluation of TCP/AQM networks, I applied mathematical tools from *queueing theory, mathematical statistics, control theory* and *game theory.*

In Thesis 1, I used simulation results as well as measurement results to validate the proposed new algorithms to measure the metrics of TCP.

The main results of Thesis 2 were obtained by analytical investigation. The results were demonstrated and validated using simulation.

In Thesis 3, I applied the tools from game theory to model the rate control as well as the parameter setting problems of multiple TCP sources. Specifically, in the rate control game, I used the methods and tools in *dynamic games* to analyze the TCP game. In the parameter setting game, I used the methods and tools in *single shot games* to analyze the problem. The claims in this Thesis are supported by rigorous and detail proofs.

The results reported in Thesis 4 were also based on analytical investigation. I applied the results from control theory to develop a new queue management scheme base on RED scheme. The results are validated by using simulation.

4 New Results

4.1 New algorithms to measure the metrics of TCP

In order to understand the dynamics of TCP, we first need to know how it *actually* works. It is sometime not obvious how to measure some important metrics of TCP (such as the congestion window) from simulation as well as in real measurement. In fact, from time to time, we need to *approximate* the theoretically-defined metrics in practice. Moreover, as new theoretical approach requires *new kind of statistics and metrics*, it is essential to know how to measure these new metrics firstly for the sake of validation. Secondly, the new metrics themselves provide us new perspective and insight into the dynamic of TCP.

Thesis 1. *I have introduced, designed and implemented a class of new algorithms to measure the metrics of TCP. The algorithms include the measurement of the number of forward-going packets, the detection of the states of TCP and the measurement of state-based metrics of TCP. I have also shown how to use these metrics to have better insight into the behaviour of TCP.*

4.1.1 Measurement of the forward-going packets of TCP connections

The congestion window of a TCP connection can be approximated by the number of out-going packets in the network (i.e. the number of packets that are sent but not yet acknowledged). This number consists of:

- The number of packets flying towards the receiver (but not yet arrived).
- The number of packets stored at the receiver's buffer (but not yet processed).
- The number of acknowledgements flying back to the sender (but not yet received by the sender).

We are interested in the first metric (i.e. the number of packets flying to the receiver). These packets are either on propagating or are waiting at some queues of some intermediate routers. If we imagine all these queues as a single "virtual queue" then our algorithm measures the fluctuation of this *virtual queue* over time. While the congestion window represents the traffic control at *end-point*, the virtual queue shows the impact of the congestion window on the *network*. The idea is illustrated in Figure 1.

Thesis 1.1. *[J4, C7, C8]*[1] *I have introduced a new metric of TCP, namely the virtual queue-length, that characterizes the fluctuation of the number of packets in forward*

[1] *References for conference papers start with letter "C", journals with "J".*

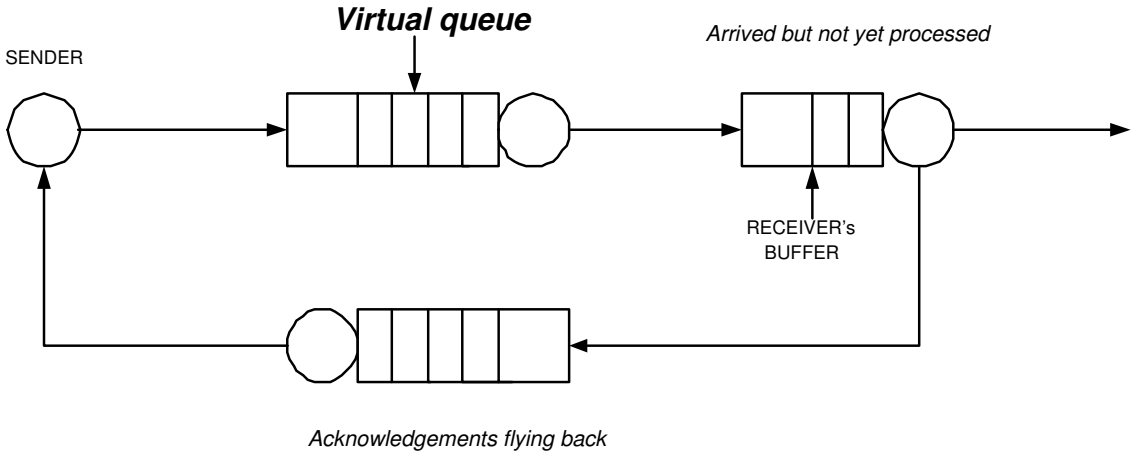

Figure 1: Illustration of the virtual queue

direction of a TCP connection over time. I have designed and implemented a new algorithm to measure this metric. The algorithm consists of the following steps:

1. *At the beginning of the connection, the virtual queue is set to 0.*

2. *Synchronization of the sender and receiver time to a common time (or "absolute" time).*

3. *After synchronization, record the time stamp of the packets sent at the sender and received at the receiver.*

4. *Merge the time stamps recorded in increasing order.*

5. *Each time a packet is sent at the sender, the virtual queue is increased by 1 (a new packet enters the queue). Each time a packet is received at the receiver, the virtual queue is decreased by 1 (a packet leaves the queue).*

One of the benefits of the proposed algorithm is that it works both for simulation traces as well as for measurement traces. As for simulation traces (obtained from NS2), we did not need Step 1 because the simulator has already done the synchronization. However, for traces from real measurements (by using the TCPDUMP tool), synchronization was needed because the time at the receiver and the sender are not necessarily the same. In this case, we used the SYN packets in the TCPDUMP traces to synchronize the two clocks.

The virtual queue algorithm was verified using simulation and real measurements as well:

- The simulation study was carried out by using the NS2 tool. The algorithm was validated against different well-known versions of TCP (Tahoe, Reno, NewReno, SACK).

- Real measurements were carried out between the Budapest University of Technology and Economics (BUTE) and Ericsson Research at Budapest. The packets were recorded by using TCPDUMP tool. The algorithm was validated against the TCPDUMP traces obtained from the measurements.

An illustration of the dynamics of the virtual queue in Congestion Avoidance phase of TCP is shown in Figure 2. In Figure 2, we also illustrate the "induced delay"

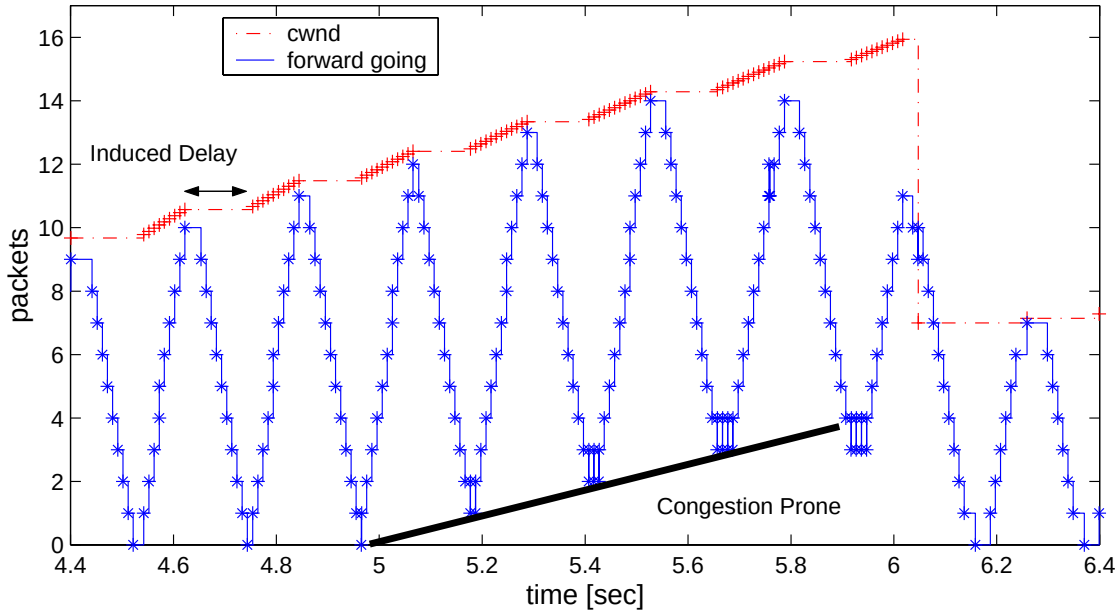

Figure 2: Dynamics of the virtual queue

concept, which is the time elapsed between two consecutive bulk of data sent and the "congestion prone" line, the slope of which indicates congestion in the network.

4.1.2 Detection of the states of TCP

In order to be able to measure some important state-based metrics of TCP, we first need to detect the beginning and the end of the states of TCP during a connection.

Thesis 1.2. *[J1, J3, C2, C4] I have proposed a new algorithm to detect the states of TCP, namely, the Slow Start state, the Congestion Avoidance state, the Fast Recovery*

state and the Time Out (Exponential Back-Off) state. The algorithm consists of the following steps:

1. *At the beginning of the connection, the state variable is set to Slow Start (TCP connections start with Slow Start phase).*

2. *Record the time stamps as well as the values of the congestion window (*cwnd*) and the Slow Start threshold (*ssthresh*) variables from trace files.*

3. *Merge the time stamps of the* cwnd *and* ssthresh *in increasing order.*

4. *Detect changes of states by using the* cwnd *and* ssthresh *variables.*

An example of the operation of the state detection algorithm is illustrated in Figure 3. The TCP connection starts with the Slow Start phase, so at the beginning

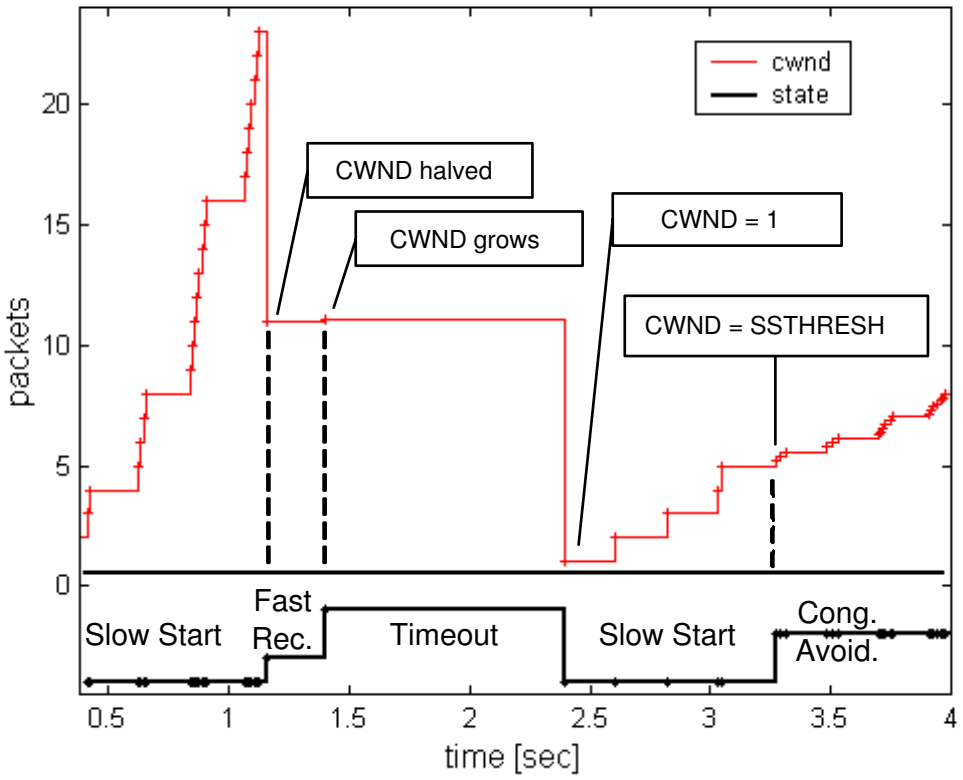

Figure 3: Illustration of the state detection algorithm

the state variable is set to Slow Start. As far as the congestion window (*cwnd*) is

increasing, TCP *stays* in Slow Start. The event that the congestion window is *halved* signals the end of the Slow Start phase and TCP (Reno) enters Fast Recovery. The arrival of the Recovery ACK signals the end of the Fast Recovery phase. If the *next* value of the *cwnd* is 1, then TCP Reno jumps from Fast Recovery to Time Out (otherwise it jumps to Congestion Avoidance). The event that the *cwnd* is set to 1 signals the end of the Time Out. After existing Time Out TCP jumps to Slow Start phase. The event that the *cwnd* equals the the *ssthresh* signal the end of the Slow Start phase and TCP Reno jumps from Slow Start to Congestion Avoidance.

The state detection algorithm was verified against some well-known versions of TCP, namely TCP Tahoe, TCP Reno, TCP NewReno and TCP SACK by simulation using NS2.

4.1.3 Measurement of state-based metrics of TCP

Being able to detect the beginning and the end of the states of TCP, new state-based metrics of TCP can be derived.

Thesis 1.3. *[J1, J3, C2, C4] Based on Thesis 1.2, I have proposed new algorithms to measure the state-related metrics of TCP. The metrics include:*

- *The sojourn time distribution at each state during a TCP connection.*
- *The jumping probabilities from one state to another state during a connection.*
- *The distribution of the number of packets sent in each time slot (RTT).*

As the first step, the algorithms begin with the state detection phase.

1. *The sojourn time distributions were computed by the following steps:*
   - *Collect all the possible states.*
   - *For each state, e.g. state i, measure frequencies of the time (in RTT) TCP spent at state i.*
   - *The set of frequencies constitutes the sojourn time distribution at each state.*
2. *The jumping probabilities from one state to another state were computed by the following steps:*
   - *Collect all the possible state jumps.*
   - *For each state, e.g. state i, count the total number of jumps from state i (N_i^{total}).*

- *Count the the total number of jumps from state i to state j (N_i^j).*
- $p_{ij} = \frac{N_i^j}{N_i^{total}}$.

3. *The distributions of the number of packets sent at each time slot (RTT) at different states were computed by the following steps:*
   - *Collect all the possible states.*
   - *For each state, at each time slot, count the the total packets sent. Collect all the possible values.*
   - *Count the frequencies of each value.*
   - *The frequencies of the number of packets sent at each time slot constitutes the distributions.*

The new state-based metrics provide us new insights into the dynamics of TCP. They are also indispensable in the state-based analysis which are discussed in details in the following sections.

The validation of the algorithms is discussed in the last paragraph of Section 4.2.3.

4.2 A new unified model for TCP

Previous efforts [Kum98, CaMe00, ZCR00] to provide a unified model for different versions of TCP found in the literature differ from each others in one or another sense, but they are all in the *congestion window based modelling* paradigm. They all tried to study the dynamics of the *congestion window* (based on some Markovian assumptions) to estimate (model) the performance of TCP. My argument was that the approach to model the dynamics of the congestion window by a Markov chain with the state space containing all the possible values of the congestion window would result in a huge number of states when the congestion window is getting sufficiently large. Recent developments of TCP, such as FAST TCP, HighSpeed TCP, Scalable TCP ([JWL04, Flo03, Kel03], respectively) that allow the congestion window as large as tens of thousands of packets would require a Markov chain as large as tens of thousands of states. Despite the fact that we do have a large body of literature on numerical methods for solving Markov chains (e.g. matrix-geometric method, see [Neu81] for a comprehensive review on the issue), the Markov chain of this magnitude is computationally infeasible in practice.

In contrast to the congestion window based approaches described above, in this thesis I introduced a new perspective into the modelling paradigm of TCP. I studied the dynamics of the *states* of the TCP itself in order to model it. I investigated the sojourn time distributions at the states as well as the jumping probabilities between

the states and use the results achieved from the investigation to build a model to estimate TCP throughput. The main advantages of my approach are that (1) it provides a *unified* model for all well-known versions of TCP because they share the same logical set of states, and (2) the number of states is *significantly* reduced (in contrast with the number of possible values of the congestion window). Reducing the number of states in a Markov model makes it more computationally feasible and more analytically tractable, and thus, more desirable.

Thesis 2. *I have proposed a new unified model for different versions of TCP. In addition, I have also introduced a new concept to characterize a TCP connection, namely the TCP characterization matrix. Based on the model, I have shown how the average throughput of a long TCP connection can be estimated.*

4.2.1 Construction of the model

The basic idea behind the model is that I tried to mimic inherent operation of TCP in order to model it. During a connection, TCP stays in any of the following states: Slow-Start, Congestion Avoidance, Fast Recovery, Exponential Back-off. TCP can jump from one state to another state in response to external events such as packet loss or Time Out. We consider how much time TCP stays in each state and the distribution of time elapsed at each state. We then consider the jumping probability from one state to another state. This inherent operation of TCP suggests us the use of the D-BMAP process (originally introduced and examined in detail in [BloCa95]). The idea of D-BMAP can be traced back to M. Neuts' work in [Neu81]. Here, we will discuss how to apply the D-BMAP process to model the traffic generated by a TCP connection in a slightly different manner as in [BloCa95]. We propose a discrete-time model for TCP. The states of the background process (modulating process) are the states of TCP itself (i.e. Slow Start, Congestion Avoidance, Loss Recovery and Time Out).
The model consists of the following elements:

- The process is time-slotted: the slot length is the average round-trip time ($\overline{RTT}$).
- The probability of transition from state i to state j is denoted by p_{ij} and the transition probability matrix of the modulating Markov-chain is $\mathbf{P} = \{p_{ij}\}$.
- When the chain is in state l, the TCP source transmits a random number of packets with probability generating function (p.g.f.) $B_l(z) = \sum_i b_i^{(l)} z^i$, where $b_i^{(l)}$ denotes the probability of i arrivals in a slot when the Markov chain is in state l.

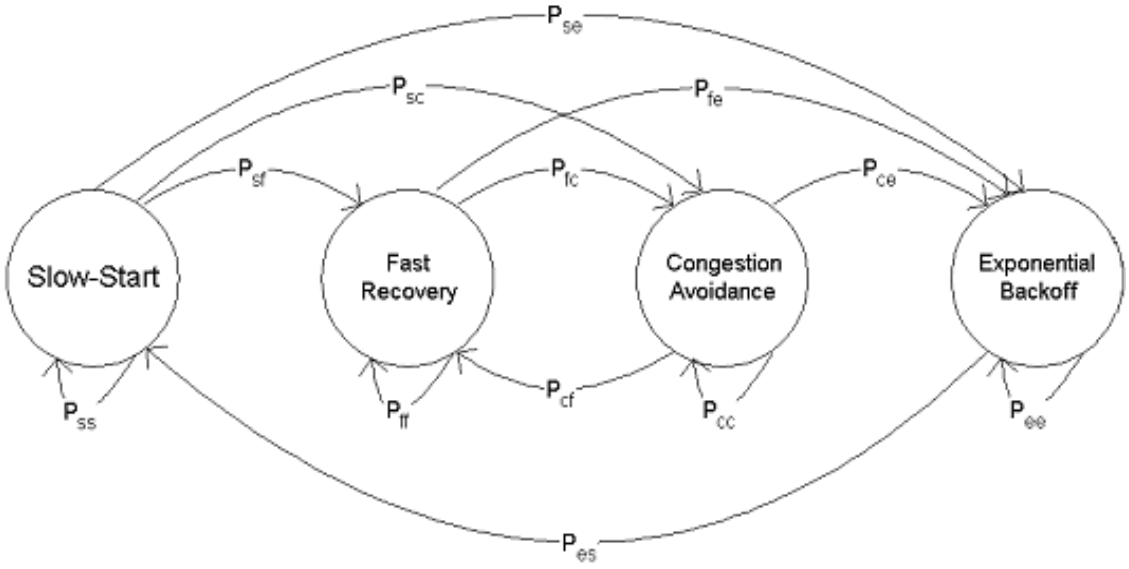

Figure 4: Illustration of the model

Figure 4 illustrates the model.

Thesis 2.1. *[J1, J3, C2, C4] I have proposed a state-based model for TCP. The traffic generated by a TCP source is modelled as a discrete time batch Markovian arrival process (D-BMAP) where the states of the modulating Markov chain are the states of TCP itself.*

The validation of the model is discussed in the last paragraph of Section 4.2.3.

4.2.2 Characterization matrix of TCP

Every individual TCP connection has its own characteristic. One of the useful features of the model presented in Thesis 2.1 is that it provides a simple description for every TCP connection.

Thesis 2.2. *[J1, J3, C2, C4] I have introduced a new concept to characterize a TCP connection, namely the TCP characterization matrix, which is the matrix that characterizes the modulating Markov chain. I have shown how, under certain assumptions, the elements of the TCP characterization matrix can be expressed and computed.*

Denote p_{TD} the probability of triple ACK loss event, p_{TO} the probability of Time Out event and $p_{threshold}$=P[$cwnd = ssthresh$] (the probability that the congestion window ($cwnd$) is equal to the slow start threshold ($ssthresh$)). Let $p_{loss} = p_{TD}+p_{TO}$.

As for the TCP SACK/NewReno case, let $p_{recovery}$ be the probability that TCP stays in Loss Recovery phase and consequently $(1-p_{recovery})$ is the probability TCP jumps from Loss Recovery to Congestion Avoidance.

I have derived the following relationship:

- *The TCP characterization matrix for TCP Reno case can be filled as follows:*

$$\mathbf{P_{Reno}} = \begin{pmatrix} (1-p_{threshold}) & 0 & 0 & p_{threshold} \\ 0 & 0 & \frac{p_{TO}}{p_{loss}} & \frac{p_{TD}}{p_{loss}} \\ 1 & 0 & 0 & 0 \\ 0 & p_{loss} & 0 & 1-p_{loss} \end{pmatrix}$$

  where $p_{loss} = p_{TD} + p_{TO}$.

- *The TCP characterization matrix for TCP Tahoe case can be filled as follows:*

$$\mathbf{P_{Tahoe}} = \begin{pmatrix} (1-p_{threshold}) & 0 & 0 & p_{threshold} \\ \frac{p_{TD}}{p_{loss}} & 0 & \frac{p_{TO}}{p_{loss}} & 0 \\ 1 & 0 & 0 & 0 \\ 0 & p_{loss} & 0 & 1-p_{loss} \end{pmatrix}$$

  where $p_{loss} = p_{TD} + p_{TO}$.

- *The characterization matrix for TCP NewReno/SACK can be filled as follows:*

$$\mathbf{P_{NewReno/SACK}} = \begin{pmatrix} 1-p_{loss} & p_{loss} \\ p_{recovery} & 1-p_{recovery} \end{pmatrix}$$

  The validation of the results is discussed in the last paragraph of Section 4.2.3.

4.2.3 Estimation of average bandwidth

From the model, we can estimate the average bandwidth of a TCP connection.

Thesis 2.3. *[J1, J3, C2, C4] Based on my model, I have proposed a generic formula to estimate the average bandwidth of a long TCP connection as follows:*

$$\overline{BW} = \Pi(\mathbf{B}'(1))^T \overline{e}[MSS/\overline{RTT}]$$

where $\mathbf{B}(z)$ *matrix is defined as follows:*

$$\mathbf{B}(z) = \begin{pmatrix} p_{00}B_0(z) & p_{10}B_0(z) & ... & p_{N0}B_0(z) \\ p_{01}B_1(z) & p_{11}B_1(z) & ... & p_{N1}B_1(z) \\ . & . & ... & . \\ . & . & ... & . \\ p_{0N}B_N(z) & p_{1N}B_N(z) & ... & p_{NN}B_N(z) \end{pmatrix}$$

and Π *denote the stationary (limit) distribution of the modulating Markov chain,* $\overline{e}$ *is the unit column matrix defined by* $\overline{e} = [1, 1, ..., 1]^T$, $\mathbf{B}'(1) = d\mathbf{B}(z)/dz|_{z=1}$.

Validation The accuracy of the model is examined by extensive simulation analysis using the NS2 tool. The Markov property of the model is validated by the distribution of sojourn time at the states of TCP. Our simulations show that if the loss probability is relatively small then the sojourn time distribution can be approximated by a geometric distribution. This support the Markov assumption in our model. The estimation of the average bandwidth was validated with different range of packet loss probability. The results show that our estimation fits well with the data traces from the simulations. The details are provided in the dissertation.

4.3 A game-theoretic analysis of TCP Vegas

With the emergence of very large bandwidth-delay product networks such as the transatlantic link with a capacity in the range of 1 Gbps - 10 Gbps, new transport protocols have been proposed to better utilize the network in these circumstances. A promising proposal is the FAST TCP, [JWL04]. Since the design of FAST TCP is heavily based on the design of TCP Vegas, there is a need to reconsider the benefits as well as the drawbacks of TCP Vegas in order to have an insight into the performance and possible deployment of FAST TCP in the future Internet.

Thesis 3. *I have analyzed the rate control problem of a general TCP Vegas network from the game-theoretic point of view. I have also shown the impact of the parameter setting of TCP Vegas and FAST TCP on their performance by using a game-theoretic approach.*

4.3.1 Rate control of TCP Vegas

I first considered the rate control problem of a general TCP Vegas network as a non-cooperative game. One of the key questions in a non-cooperative flow control game

in general, and our game in particular, is whether the network converges to (or settles at) an equilibrium point, such that no player can increase its payoff by adjusting its strategy unilaterally. In the game-theory terminology such a point is called a *Nash equilibrium*. The Nash equilibrium in our game also reflects the *balance* of the gain and the cost for each player as well as for the network as a whole. A non-cooperative game may have no Nash equilibria (in its pure strategy space), multiple equilibria, or a unique equilibrium.

Thesis 3.1. *[C1] I have proven that there exists a unique Nash equilibrium (in its pure strategy space) for the TCP Vegas rate control game.*

The details of the proof are provided in the dissertation. To reach this equilibrium, [Ros65] shows that each player can change its own strategy at a rate proportional to the gradient of its payoff function with respect to its strategy and subject to constraints. This method is in fact equivalent to the gradient projection algorithms described in [LoLa99].

The authors of [LoLa99], using optimization framework, also showed that, under certain assumptions on the step size, these algorithms converge to a system wide optimal point (which is also proved to be unique). Furthermore, it is proved in [LPW02], [Low03] that the rate control of TCP Vegas/Drop Tail and TCP Vegas/REM is indeed based on these algorithms. This implies that *the TCP Vegas game described above converges to a unique Nash equilibrium that is system wide optimal.*

4.3.2 Parameter setting of TCP Vegas

As described in [BMP94], TCP Vegas tries to maintain the number of backlogged packets in the network between α and β (the parameters of TCP Vegas). I considered the case when N TCP Vegas flows sharing a single bottleneck link with capacity μ and with a buffer of size B. The TCP Vegas flows (the *players*) are assumed to be selfish (and greedy) - they all try to increase the number of their backlogged packets in the network. We are interested in a situation (i.e. a parameter setting, if at all exists) from where no player would deviate.

Thesis 3.2. *[C1] I have modelled the parameter setting problem of TCP Vegas as a static game. I have demonstrated that:*

- *If the players have the payoff function of the form $f(\alpha_i) = \lambda_i$ (the average throughput), then there exist multiple Nash equilibria for the game. However, in these Nash equilibria, each TCP Vegas flow (player) maintains the number of its own backlogged packets as many as possible. As a result, the buffer is nearly full and the queueing delay is unnecessarily high. A nearly full buffer may cause many difficulties for TCP Vegas (e.g. the estimation of* baseRTT

might be inaccurate if there are already many packets in the queue when the connection starts).

- *If the players have the payoff function of the form* $f(\alpha_i) = \frac{\lambda_i}{\alpha_i}$, *then* $\alpha = (1, 1, ..., 1)$ *is the unique Nash equilibrium of the game. From the system (the network as a whole) point of view, this Nash equilibrium is an* efficient and desirable *equilibrium, since it* minimizes *the total backlogged packets at the buffer, and in so doing, minimizes the queuing delay at the buffer.*

- *If the players have the payoff function of the form*

$$f(\alpha_i) = \begin{cases} \frac{\lambda_i}{\sum_{j=1}^{N} \alpha_j} & \text{if } \sum_{j=1}^{N} \alpha_j < B, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

then there exist multiple Nash equilibria for the game. However, as in the first case, in these Nash equilibria the buffer is close to full making the network vulnerable to congestion.

The results of Thesis 3.2 imply that the parameter setting problem of TCP Vegas is very sensitive and could result in very inefficient Nash equilibria (if at all exist).

4.3.3 Parameter setting of FAST TCP

FAST TCP was recently proposed as a transport protocol for very high bandwidth-delay product networks. However, little is known about how to set its parameters as well as what is the impact of the parameter setting to the performance of the network.

Thesis 3.3. *[C1] I have extended the analysis in Thesis 3.2 to the case of FAST TCP.*

- *I have showed how FAST TCP can be considered as a "faster" TCP Vegas as follows:*

$$w(t+1) = \begin{cases} w(t) + \gamma(\alpha - \mathit{diff}) & \text{if } \mathit{diff} < \alpha, \\ w(t) - \gamma(\mathit{diff} - \alpha) & \text{if } \mathit{diff} > \alpha, \\ w(t) & \text{otherwise.} \end{cases} \tag{2}$$

where $\mathit{diff} = \frac{RTT - baseRTT}{RTT} w$.

- *Based on the observation (Equation 2), I have modelled the parameter setting problem of FAST TCP as a static game and demonstrated that if the players have the payoff function of the form* $f(\alpha_i) = \lambda_i$ *(the average throughput),*

then there exist multiple Nash equilibria for the game. However, in these Nash equilibria, each FAST TCP flow (player) maintains the number of its own backlogged packets as many as possible. As a result, the buffer is close to full and the queueing delay is unnecessarily high. A nearly full buffer may cause many difficulties for FAST TCP (e.g. the estimation of baseRTT *might be inaccurate if there are already many packets in the queue when the connection starts).*

From Equation 2 we can see that FAST TCP increases (or decreases) its window size by $\gamma(\alpha - \mathsf{diff})$, instead of 1 as in TCP Vegas. If α, γ are chosen so that $\gamma(\alpha - \mathsf{diff}) = 1$, then FAST TCP would behave *exactly* like TCP Vegas. In fact, this feature (the *amount* of window increment/decrement) is the major difference between FAST TCP and TCP Vegas. The results in Thesis 3.3 were derived and validated by analytical analysis.

4.4 Performance analysis of RED

One of the most promising active queue management schemes being proposed for deployment in the Internet is the Random Early Detection (RED) scheme. However, research results on RED performance are highly mixed, especially in the field of tuning its parameters. In fact, one of the inherent weaknesses of RED is parameter sensitivity. Extensive research has been devoted to this issue and many publications have highlighted various aspects of this issue. However, the question of how to configure the parameters of RED for optimal performance is still open. In addition, the impact of RED mechanism on different issues of Internet performance (such as fairness) is still not clear and requires more clarification and analysis.

Thesis 4. *I have pointed out the advantages as well as the weaknesses of the Random Early Detection mechanism by reconsidering its proportional loss property and the Exponential Weighted Moving Average (EWMA) algorithm. In addition, I have also proposed a novel adaptive Active Queue Management (AQM) scheme based on the RED mechanism and shown that how the new mechanism can improve RED in a number of router-based performance metrics.*

4.4.1 Proportional loss revisited

Loosely speaking, the proportional loss property means that the fraction of marked packets for each connection is proportional to that connection's share of the bandwidth. RED is claimed to possess this property [FloJa93]. In addition, proportional loss is widely adopted in the fairness analysis of RED, [LiMo97], [HBT99]. Since TCP flows account for a large portion of Internet traffic, TCP arrivals are mainly of interest.

Thesis 4.1. *[J2, C5] I have proven that packet losses between flows in TCP/RED networks is non-proportional by applying the ASTA properties.*

The details of the proof are provided in the dissertation. It should be noted that M. May *et al* in [MBB00] have already suggested that the proportion loss property is true *only if* the arrival flows are Poisson arrivals. However, their analysis is based on the PASTA (Poisson Arrivals See Time Averages) property of Poisson processes. We take one step further by observing that the PASTA property can be generalized to ASTA (Arrivals See Time Averages), [MeYa95], and Burke in [Bur76] has shown that the composite stream of exogenous Poisson arrivals and feedback customers is *not* Poisson even though this stream sees a time average. In conclusion, our analysis strengthen the observation in [MBB00] with specific application to TCP/RED network.

4.4.2 A new AQM scheme

Some solutions have been proposed to overcome the difficulties of tuning RED parameters. We can mention here the Adaptive RED of Feng [FKSS99] and the Adaptive RED of Floyd [FGS01]. These proposals try to change the max_p parameter to adapt to the changing condition of incoming traffic. My argument was that the inflexibility of RED not only lies in its max_p parameter but also on the EWMA mechanism. As a result, RED parameters can be tuned in an on-line manner by making the EWMA adaptive to the changing condition of the incoming traffic.

Thesis 4.2. *[J2, C5] I have proposed a new AQM scheme (fuzzy RED) to alleviate the inflexibility of RED tuning. The scheme consists of the following elements:*

- *Only modify the EWMA part of the RED mechanism, all other parts of the RED mechanism are kept intact.*
- *Changing the weighting parameter (w_q) according to network conditions by using the Fuzzy EWMA.*
- *Defining the Control Rules and Specification of Fuzzy Labels (HIGH, MEDIUM, LOW).*

The flow diagram of the mechanism is illustrated in Figure 5. Consider a discrete time system with q_k, the queue length at the buffer at time k, as the state variable. The system can span a spectrum varying from 'steady' (stationary) to 'noisy' (non-stationary). Let $\hat{q}_k$ be the estimate of q_k, then observation noise (error) is $q_k - \hat{q}_k$. To see the relation between error and the predictor, we define scaled error as $|q_k - \hat{q}_k|/\hat{q}_k$.

The first question we need to deal with is how to define the control rules. We assume that when the queue stays in its stationary (stable) state, the *estimation error*

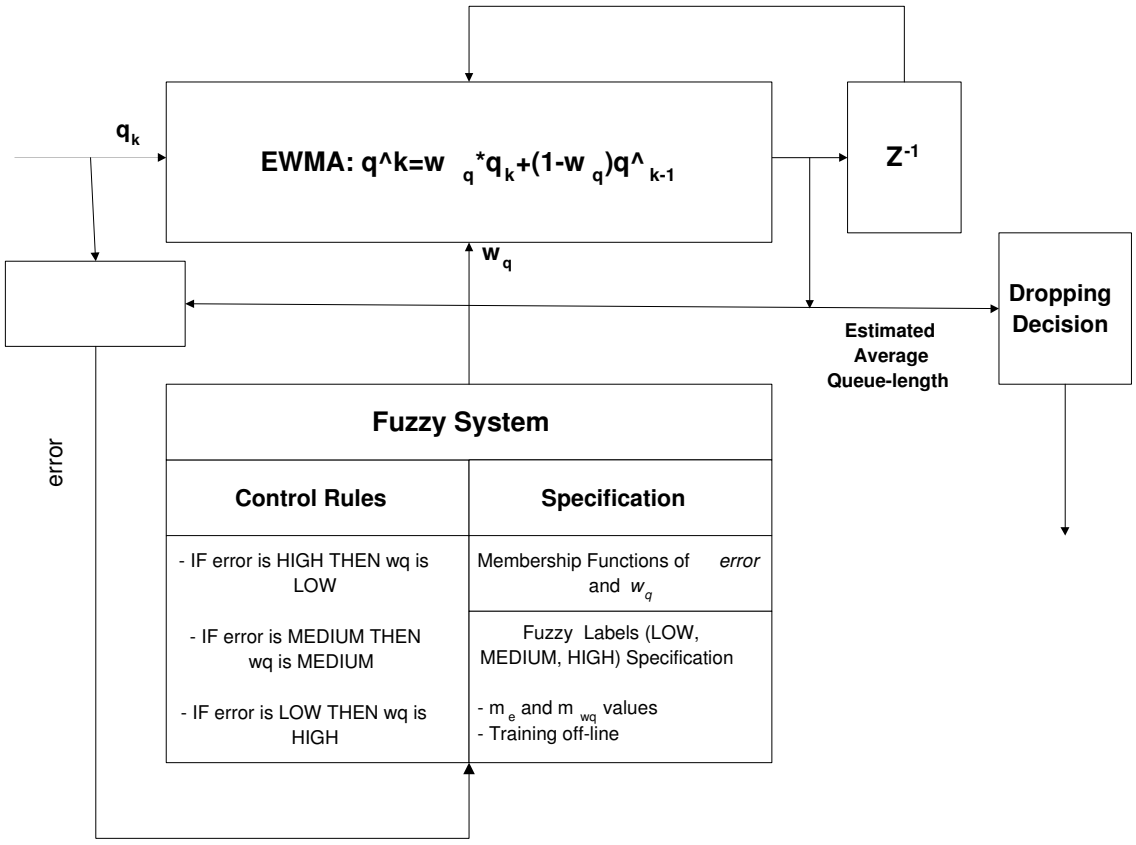

Figure 5: Flow diagram of the fuzzy RED mechanism

is small. That is, if the dynamics of queue-length in the buffer has little perturbation, then the exponential averaging technique will produce a predictor that is usually close to the actual system state (error is small). In this case, w_q should be large. In contrast, when there is a large variation in queue-length, past history cannot predict the future well (the error is high). In this case, we set w_q low, so that the estimator can track changes in the system. Finally, since we do not have a good grasp of the state dynamics, we only define three gradations in the values of w_q and *error*. In addition, keeping the number of gradations minimal reduces the overhead computing time for the algorithm. Thus, we adopt the following control rules:

- IF *error* is HIGH THEN w_q is LOW.
- IF *error* is MEDIUM THEN w_q is MEDIUM.
- IF *error* is LOW THEN w_q is HIGH.

Secondly, we need to answer the question: HIGH, MEDIUM, LOW are related to what? The answer for this question is equivalent to defining the membership functions for *error* and w_q. For the sake of simplicity, we use the trapezoid form (the conventional and simplest form) for these two variables.

Validation I have carried out a comparative performance analysis of different adaptive RED schemes by simulations. I have investigated different scenarios: stationary cases, multiple sources with different RTT, different number of flows, dynamically changing number of flows. The simulations show that, in the case of a high workload and a high level of variation, fuzzy RED, by tracking system variation in an on-line manner, improves RED performance in a number of important router-based metrics like packet loss rate, average queueing delay, link utilization, and global power. The details are provided in the dissertation.

5 Application of the Results

The objective of this thesis was to discuss, analyze, and improve the traffic control techniques in TCP/IP networks.

The algorithms to measure the metrics of TCP presented in Chapter 2 allow us to have a better understanding of the dynamics of TCP traffic from the practical point of view. Network engineers and researchers can use these algorithms to infer about the condition of the networks as well as for validation of the theoretical results.

Chapter 3 provided a unified model for some well-known versions of TCP. It also introduces a new way to characterize TCP connections. The results can be applied in network dimensioning of TCP/IP networks.

The game-theoretic analysis presented in Chapter 4 helps us to understand the rate control and the parameter setting problems of TCP Vegas. The results of rate control games can be used in network pricing mechanisms of TCP Vegas networks. The results of parameter setting games can be applied in configuring network resources.

A comprehensive performance analysis of the Random Early Detection mechanism was presented in Chapter 5. I revisited a number of concepts and found that previous thinking needs to be changed. I suggested that RED generally does not provide the capability to apportion loss between flows. I highlighted the problems associated with the tuning of RED and suggest an alternative approach. A possible future application is to deploy my proposed scheme as part of the queue management module of the routers.

Acknowledgements

First of all, I would like to thank Dr. Sándor Molnár for his kind encouraging guidance and support. My research work has been done at the High Speed Network Labs (HSNLabs). I wish to express my grateful thanks to the head of the laboratory, Dr. Tamás Henk, for his support. I would also like to thank Dr. László Györfi, Tamás Éltető, Dr. Attila Vidács, Dr. Miklós Telek and Dr. András Veres for their helpfulness and fruitful advice contributing to my research work - Trịnh Anh Tuấń.

References

[FGS01] Sally Floyd, Ramakrishna Gummadi, and Scott Shenker. *Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management,* ACIRI Technical Report, 2001.

[BloCa95] C. Blondia, O. Casals, *Statistical multiplexing of VBR sources: A matrix-analytic approach* Performance Evaluation 16, pp. 5-20, 1992.

[FKSS99] W. Feng, D. Kandlur, D. Saha, and K. G. Shin. *BLUE: A New Class of Active Queue Managemnent Algorithms.* UM CSE-TR-387-99, April 99.

[BMP94] L. Brakmo, S. O'Malley, and L. Peterson, *TCP Vegas: new techniques for congestion detection and avoidance,* IEEE/ACM SIGCOMM 94, London, UK, Sept. 1994.

[Bur76] P. Burke, *Proof of a Conjecture on the Interarival-Time Distribution in M/M/1 Queue with Feedback* IEEE Trans. on Communication, vol. 24, 1976.

[CaMe00] C. Casetti and M. Meo, *A new approach to model the stationary behavior of TCP connections.* In Proc. of IEEE INFOCOM, pages 367–375, March 2000.

[CJOS00] M. Christiansen, K Jeffay, D. Ott, F. D. Smith, *Tuning RED for Web traffic* ACM SIGCOMM'00, Stockhom, 2000.

[JWL04] Cheng Jin, David X. Wei and Steven H. Low *FAST TCP: motivation, architecture, algorithms, performance,* IEEE INFOCOMM'04, March 2004

[HBT99] P. Hurley, J. Boudec, and P. Thiran, *A Note on the Fairness of Addictive Increase and Multiplicative Decrease* ITC 16, UK, 1999.

[Flo03] Sally Floyd, *HighSpeed TCP for Large Congestion Window,* RFC 3649, December 2003

[Jac88] V. Jacobson, *Congestion avoidance and control,* Proceedings of ACM SIGCOMM'88, August 1988.

[Kes91] S. Keshav, *A Control-theoretic Approach to Flow Control* Proc. ACM SIGCOMM 1991, Sept. 1991

[Kum98] A. Kumar, *Comparative Performance Analysis of Versions of TCP in a Local Network with a Lossy Link,* IEEE/ACM Transactions on Networking, 1998.

[LiMo97] D. Lin and R. Morris, *Dynamics of Random Early Detection* SIGCOMM'97

[LoLa99] S. Low and D. Lapsley, *Optimization flow control, I: basic algorithm and convergence,* IEEE/ACM Transactions on Networking, 7(6):861-874, December 1999.

[Low03] S. Low, *A duality model of TCP and queue management algorithms,* IEEE/ACM Transactions on Networking, October 2003.

[LPW02] S. Low, L. Peterson, and L. Wang, *Understanding Vegas: a duality model,* Journal of ACM, 49(2):207-235, March 2002.

[MBB00] M. May, T. Bonald, and J. Bolot. *Analytic Evaluation of RED Performance* Proc. of INFOCOM'00, 2000.

[MeYa95] B. Melamed and D. Yao, *The ASTA Property* Frontiers in Queuing: Models, Methods, and Problems, CRC Press, 1995.

[MSMO97] M. Matthis, J. Semske, J. Mahdavi, and T. Ott, *The Macroscopic Behavior of the TCP Congestion Avoidance Mechanism.* Computer Communication Review, 27(3), July 1997.

[Neu81] Marcel Neuts, *Matrix-Geometric Solutions in Stochastic Models - An Algorithmic Approach,* The Johns Hopkins University Press, Baltimore, Maryland, 1981.

[PFTK98] J. Padhye et al, *Modeling TCP Reno Throughput: A Simple Model and Its Empirical Validation.* SIGCOMM'98, 1998.

[HMTG01] C. V. Hollot, V. Misra, D. Towsley and W. Gong, *A Control Theoretic Analysis of RED* INFOCOM 2001, Alaska, April 22-26, 2001

[FloJa93] Sally Floyd and Van Jacobson *Random Early Detection Gateways for Congestion Avoidance* IEEE/ACM Transactions on Networking, vol. 1, no. 4, August 1993, pp. 397-413

[ALLY01] S. Athuraliya, V. Li, S. Low, Q. Yin, *REM: active queue management,* IEEE Network, June 2001.

[Ros65] J. B. Rosen, *Existence and uniqueness of equilibrium points for concave n-person games,* Econometrica, vol. 33, pp. 520-534, Jul. 1965.

[Kel03] Tom Kelly, *Scalable TCP: Improving Performance in Highspeed Wide Area Networks,* ACM SIGCOMM Computer Communication Review, Vol. 33, Issue 2, pp. 83-91, April 2003.

[VeBo00] A. Veres, M. Boda, *The Chaotic Nature of TCP Congestion Control.* INFOCOM 2000, Tel Aviv, 2000.

[ZCR00] M. Zorzi, A. Chockalingam, and R. Rao *Throughput analysis of TCP on channels with memory,* IEEE Journal on Selected Areas in Communications, vol. 18, no. 7, pp. 1289–1300, 2000.

Publications

Journal papers

[J0] S. Molnár, **T. A. Trinh**. Congestion Games in TCP Vegas and Their Applications in FAST TCP. Submitted to *Telecommunications Systems*, 2004.

[J1] **T. A. Trinh**, S. Molnár. Modelling and Analysis of TCP Traffic: A State-based Approach. *under submission*, 2004.

[J2] **T. A. Trinh**, S. Molnár. A Comprehensive Performance Analysis of Random Early Detection Mechanism. *Telecommunications Systems*, 25 (1-2): 9-31, January - February, 2004.

[J3] **T. A. Trinh**, S. Molnár. Modelling TCP Traffic: A State-based Approach. *Periodica Polytechnica, Electrical Engineering*, Vol. 48, No. 1, pp. 1-14, 2004.

[J4] **T. A. Trinh**, T. Éltető. On the Stability of TCP. *Journal on Communications,* November-December 2000.

Conference papers

[C0] **T. A. Trinh**, S. Molnár. Understanding TCP Vegas and FAST TCP: A Game-Theoretic Perspective. Submitted to *IEEE/IFIP Networking 2005.*

[C1] **T. A. Trinh**, S. Molnár. A Game-Theoretic Analysis of TCP Vegas. In *Proc. of QofIS'04 - Quality of Service in the Emerging Networking Panorama*, **Springer** *Lecture Notes in Computer Science 3266 (LNCS 3266)*, pp. 338-347, Barcelona, Spain, September 29 - October 1, 2004.

[C2] **T. A. Trinh**, S. Molnár. A Novel Approach to Model TCP Traffic. In *Proc. of IEEE GLOBECOM 2004*, Dallas, Texas, USA, November-December 2004.

[C3] **T. A. Trinh**, B. Sonkoly, S. Molnár. A Study of HighSpeed TCP: Observations and Re-evaluation. In *Proc. of EUNICE 2004,* Tampere, Finland, June 2004.

[C4] **T. A. Trinh**, S. Molnár. A State-based Analysis of TCP. In *Proc. of IFIP Workshop on Next Generation Networks,* Hungary, 8-10 September 2003.

[C5] **T. A. Trinh**, S. Molnár. RED Revisited. In *Proc. of The 10th International Conference on Telecommunication Systems Modeling and Analysis,* CA, USA, October 3-6, 2002.

[C6] **T. A. Trinh.** On the Estimation of Average Queue-length in RED. In *Proc. of PCH Conference on Telecommunications,* Budapest, Hungary, April 2001.

[C7] **T. A. Trinh**, T. Éltető, L. Györfi. On Some Metrics of TCP. In *Proc. of The 25th International Conference on Local Area Networks*, Florida, USA, November 2000.

[C8] **T. A. Trinh.** On the Stability of TCP. In *Proc. of Students Scientific Conference,* Budapest Univ. of Technology and Economics, Budapest, Hungary, November 1999.