Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Systems Architecture 51 (2005) 542–569
JOURNAL OF SYSTEMS ARCHITECTURE
www.elsevier.com/locate/sysarc

Equation-based TCP-friendly congestion control under lossy environment

S.W. Ng, Edward Chan *

Department of Computer Science, City University of Hong Kong, 83 Tat Chee Ave., Kowloon, Hong Kong

Received 8 January 2002; received in revised form 4 July 2002; accepted 15 October 2002
Available online 7 April 2005

Abstract

A major problem in supporting multimedia streaming in the Internet is that the streaming protocol used tends to take bandwidth away from competing TCP traffic streams. Thus it is important to devise rate based protocols that are TCP-friendly. In the paper we examine the effectiveness of two important models in devising TCP-friendly rate based protocols in lossy environment such as wireless networks. Using simulation for a variety of network scenarios, it is shown that the rate based protocols based on one model underestimates while the other overestimates TCP bandwidth in the presence of errors. A modification is made to the models which allows us to construct a rate based protocol that can track TCP bandwidth better in the presence of random error and hence exhibit behavior that is more TCP-friendly.

Keywords: Real-time streaming protocols; Congestion control; Wireless networks; TCP friendliness

1. Introduction

Explosive growth in wireless communication technology has made it possible for mobile wireless devices to access the Internet at higher bandwidth. For example, the General Packet Radio Service (GPRS) provides theoretically 171.2 kilobits per second (kbps) in bandwidth. In addition, data rate through a wireless LAN connection are in the range of several Mbps and poised to increase in the near future. Moreover, there has been a similar advance in the development of compression technology. For instance, the new MPEG-4 compression algorithm can transmit a QCIF format video stream in the range of 5 kbps–64 kbps [1]. These developments have not only increased the feasibility of the transmission of continuous media (CM) data through the Internet but also made it more likely that mobile multimedia applications can be deployed in the near future.

* Corresponding author. Tel.: +852 2788 8626; fax: +852 2788 8624.
E-mail address: csedchan@cityu.edu.hk (E. Chan).

1383-7621/$ - see front matter
doi:10.1016/j.sysarc.2002.10.001

However, some major problems must be solved before CM data transmission can be achieved in a shared network environment such as the Internet. It is well known that Transmission Control Protocol (TCP) is the major transport layer protocol in the Internet. TCP has a sophisticated congestion control algorithm which is essential for the reliable transmission of data in the Internet. However, it is not suitable for the transmission of CM which requires the support of a different type of Quality of Service (QoS) [2,5,7]. One the other hand, using a protocol such as UDP which does not have a congestion control mechanism will result in CM traffic dominating the bandwidth, resulting in congestion and delay for other traffic transported by TCP. The basic problem is that when congestion is detected by TCP, it reduces the congestion window by half. This rapid response to congestion however makes TCP very vulnerable to protocols designed for CM which can easily grab all the capacity of the network so willingly given up by TCP.

The solution to this problem is to make these competing protocols TCP-friendly. Since TCP congestion control is mainly based on the well known *Additive Increase Multiplicative Decrease* (AIMD) mechanism for transmission rate adjustment, a protocol that simulates this behavior can share network resources with TCP traffic fairly. There are two main approaches in the design of congestion control, one is AIMD-based and the other is equation-based. The former uses AIMD directly for transmission rate control [3–5] and reduces transmission rate in response to a single congestion control indication. Equation-based congestion control, on the other hand, predicts the transmission rate of TCP traffic by a model. Two well-known equations have been presented by Floyd [6] and Padhye [7]; TFRC presented in [8] is a variant of Padhye's approach which produces smoother changes in the transmission rate.

Many research studies [5,7–9] have shown that both AIMD-based and equation-based congestion control can be used to achieve TCP friendliness in a wired network environment and hence used to construct streaming protocols. However, very few of them deal with the performance of these approaches for a wireless link. Since the performance of TCP is quite different under wireless and wired environments, it is important to evaluate whether the TCP friendliness achieved by the above protocols can be maintained in a wireless environment where high loss rate is the norm. In [10], the behavior of an AIMD-based protocol called RAP is examined in a wireless environment and a number of modifications suggested. TFRC, one of the better known equations based TCP-friendly protocol, is shown to achieve good result in TCP-friendliness for a wired environment [11]. Based on these results we believe a comprehensive evaluation of equation-based congestion control in a wireless environment is important and promising.

There is a vast literature on performance problems for TCP over lossy wireless links, many of which attempt to hide link level losses from TCP by re-transmitting lost data before TCP activates its error recovery mechanism [17–20]. However, our focus in this paper is not on TCP per se but on a rate based protocol that is TCP-friendly. While there is also a large body of work in congestion control over lossy wireless environments some of which dealing specifically with streaming multimedia information, to our knowledge none of the published work in this domain address TCP-friendliness. Finally, as we have noted above, although a number of streaming protocols have been proposed in the literature, none deal specifically with lossy environments. Herein lies the significance of our work: to develop an equation based streaming protocol that is TCP-friendly in a lossy environment. We know of no directly comparable work in the literature.

However, to design an equation-based streaming protocol that is TCP-friendly, a key issue is to understand the behavior of the classical TCP modeling equations under lossy environments and try to emulate its behavior. In this paper, the behavior of two different TCP models under wireless channel will be evaluated. The main problem in using TCP over a wireless connection is that it is unable to distinguish packet drop due to channel error from network congestion, which results in low channel utilization. An effective protocol, on the other hand, must be able to distinguish between the two cases and protect the TCP traffic

stream in a lossy environment to ensure fair allocation of the bandwidth. We first evaluate whether these two models of TCP provide the basis for the design of such a protocol and then proceed to propose enhancements to the models to improve their accuracy.

This paper is organized as follows. Two well-known models used as the basis of equation-based TCP-friendly congestion control protocols will be briefly described in the next section. In Section 3, simulation experiments for both models in lossy wireless environment will be discussed. In Section 4, the performance of the equation models in a different network scenario, a wireless last-hop channel for an Internet environment, will be studied. In Section 5, a modification to the models will be proposed. In Section 6, the performance of the enhanced model will be studied using simulation and shown to outperform the original model in lossy wireless environments. The paper concludes with Section 7.

2. TCP-friendly congestion control

In this section, we will briefly discuss the assumptions and mechanisms of rate based TCP-friendly congestion control. Existing TCP-friendly rate control equations can only coexist with TCP traffic stream that does not enter slow-start after a segment lost is detected [6,7], such as TCP-Reno and TCP-SACK. We will use TCP-Reno in our study since it has a large share of TCP traffic in the Internet [12,13] and its mechanism is relatively simple. It uses fast recovery rather than slow start after a segment lost has been detected, and transmission rate variance is linear after the connection reaches equilibrium. As a result, the transmission rate of each RTT can be estimated more easily.

There are two well-known equations that describe the steady state transmission rate of a TCP connection, proposed by Floyd and Mahdavi [6], and Padhye et al. [7] respectively. These two formulae represent two ways of modeling the behavior of TCP, and in this paper, we will designate the former as the simple TCP model or the *simple model* and the latter the complex TCP model or the *complex model.*

2.1. The simple TCP modeling equation

The simple model proposed by Mahdavi and Floyd has a number of assumptions:

- Segment lost detection is only due to triple duplicate acknowledgements but never due to packet timeout.
- Error model is deterministic i.e., a segment drop occurs after every p segments have been transmitted.
- TCP has sufficient bandwidth and will not result in excessive queuing in the data source or intermediate nodes.

Based on these assumptions, the RTT of the TCP traffic will be nearly constant and equals the sum of the transmission delay and the propagation delay. Moreover, since retransmission is ignored this is basically a pure AIMD(1, 0.5) model. Consequently, the following relationship between the TCP traffic congestion window size W and the probability of segment lost p can be derived:

$$W = \sqrt{\frac{8}{3p}}$$

Using this equation, we can derive the data transmission rate of each round trip as follows:

$$\text{Transmission rate} = \frac{\text{Data of a cycle}}{\text{Duration of a cycle}}$$

$$\frac{\text{MSS} \times \frac{3}{8} \times W^2}{\text{RTT} \times \frac{W}{2}} = \frac{\text{MSS} \times C}{\text{RTT} \times \sqrt{P}}$$

where MSS is the maximum segment size. Normally, $C = \sqrt{3/2}$ although the value of C value can vary with packet loss rate [9]. This model, however, does not consider the effect of retransmission and fast-recovery on TCP traffic when a segment loss is detected.

2.2. The complex TCP modeling equation

In [7], Padhye extended the above model by considering the effect of timeout as well as delayed acknowledgement (ACK). The reason for this modification is that when TCP-Reno is used in

conjunction with drop-tail routers, packet loss are bursty (i.e., correlated) which leads in turn to a significant number of timeouts. Accounting for this effect leads to the following TCP modeling equation (which we call the complex TCP modeling equation):

$$\text{Transmisson rate} = \frac{s}{\text{RTT}\sqrt{\frac{2bp}{3}} + \text{Rto} \times \min\left(1, 3\sqrt{\frac{3bp}{8}}\right) p(1+32p^2)}$$

where RTT is the round trip time, Rto is the retransmission timeout estimated by the Jacobson/Karn algorithm [14,15], b stands for the number of segments acknowledged for each ACK, s is the segment size and p is the loss probability (which is actually the frequency of loss indications per packet sent). The first part in the denominator $\left(\text{RTT}\sqrt{\frac{2bp}{3}}\right)$ represents the effect of triple duplicate ACKs, while the rest of the denominator estimates the effect on transmission rate by TCP timeout.

Experiments in [7,9] show that the simple model deviates from TCP transmission rate when the loss rate p increases. When the loss rate increases, the frequency of segment loss due to timeout increases. Since the simple model does not consider the timeout factor, the complex model provides a better estimate of the TCP transmission rate for a wider range of loss rates.

3. Evaluation of the TCP models

In this section, we will evaluate the two equations under lossy environment through two sets of experiments. In the first set of experiments, simulated traffic is fed into the two models to see how well they predict TCP transmission rate. In the second set of experiments, a full simulation of competing TCP streams and rate-based protocols based on the two models will be conducted and the results analyzed.

3.1. Baseline experiments

We will first examine the behavior of the TCP models using only one TCP Reno connection running through a simple, single channel network. In this baseline experiment, we use the ns-2 simulator [16] to first generate a TCP traffic stream, which is then fed into the two TCP models to derive estimates for the TCP transmission rate. Random error is then added into the channel using ns-2 simulator facilities and the average data rate recorded. We study how the TCP model performs in different situations by performing simulations with different combinations of delay, packet size, bandwidth, and loss rate. Two packet sizes are used in the experiments: 512 and 1024 bytes. The bandwidth available ranges from 5 KBps to 20 KBps in increments of 5 KBps, and propagation delay various from 10 ms to 40 ms. The percentage of error added ranges from 2.5% to 40%, with 2.5% increments.

The overall average transmission rate and packet loss rate p are recorded at the end of each simulation. Since the Reno version of TCP is applied p represents the number of drop signals per packet sent. Figs. 1 and 2 show the results of the simulations.

From the scatter graph, we can see that the simple TCP model tends to overestimate the available bandwidth under lossy channel transmission. On the other hand, the complex TCP model underestimates the available bandwidth. Information about the relationship between data loss rate and transmission rate cannot be derived directly from the figures. To extract this information, we represent the data in percentage of deviation of estimated TCP throughput from TCP traffic. The way of presentation shows the percentage of TCP modeling formulas from TCP Reno traffic. We will examine the experiment results in terms of injected error rate. Fig. 3 shows the relationship of the number of error signals with percentage of errors added to the channel. As shown in Fig. 3, estimated throughput based on the simple TCP model overshoots actual TCP throughput by more than 90% when the percentage of error added increases beyond 10%. The deviation for higher error rates is so large (several 100%) that we do not capture it in Fig. 3.

As shown in the figure, the simple TCP model leads to a better estimate when the error rate is small. However, after the percentage of injected error increases beyond 5%, the number of out-of-range estimates increases sharply. The complex

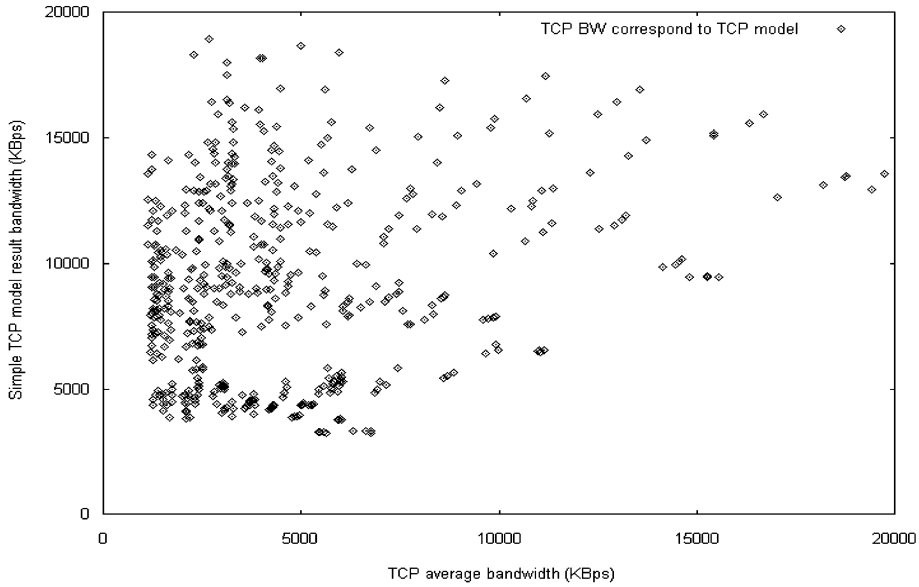

Fig. 1. Estimated throughput of simple TCP model vs measured throughput.

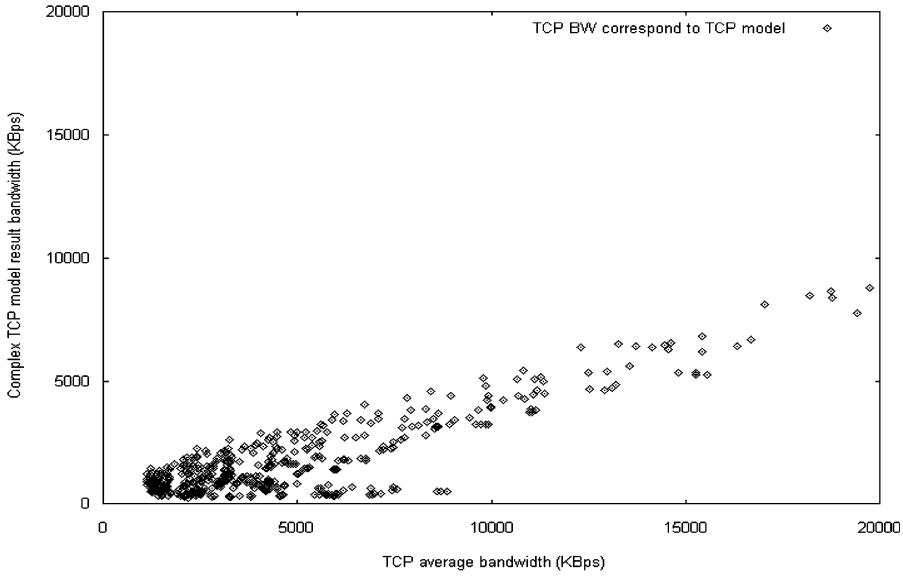

Fig. 2. Estimated throughput of complex TCP model vs measured throughput.

TCP model, on the other hand, results in a large number of out-of-range estimates when the injected error rate is low, then improves when the injected error rate increases and finally worsens when the percentage of injected error increases beyond 25%. From the graph, we can conclude that the simple TCP model overestimates available bandwidth, and has better performance for low error rates than high error rates. The complex TCP model, on the contrary, tends to underesti-

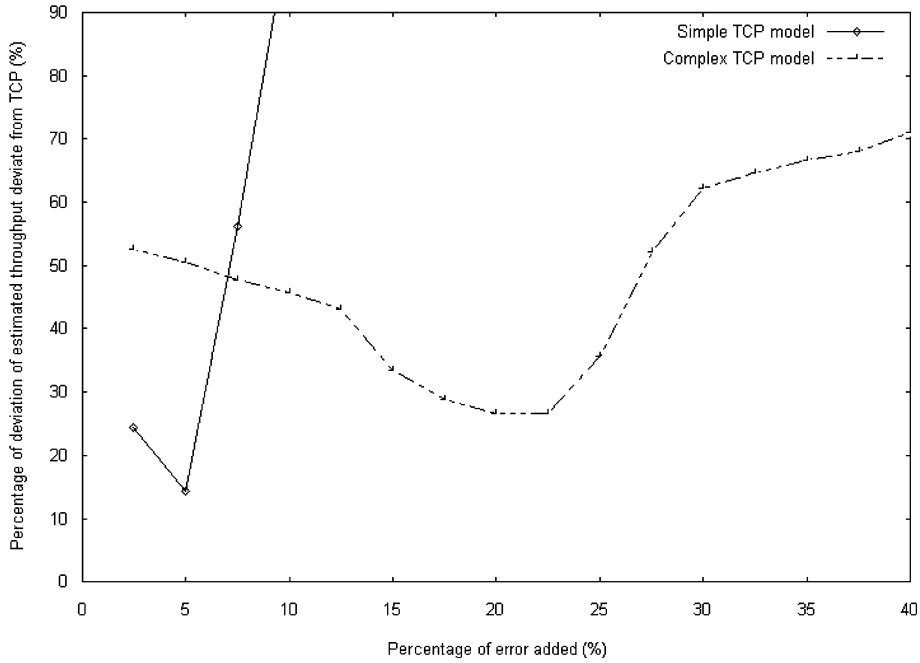

Fig. 3. Deviation of estimated throughput based on simple TCP model from actual TCP throughput vs percentage of error added.

mate the available bandwidth, and has better performance when the percentage of injected error is between 15% and 25%.

3.2. Performance of the TCP models for a lossy channel

One of the objectives of this paper is to study the effectiveness of the two TCP models in supporting TCP-friendly congestion control in a lossy wireless network environment. We construct a simple rate-based congestion control protocol based on each of the two models, and compare their performance in a series of simulation experiments. Transmission rate is controlled by adjusting the inter-packet interval among packets and the interval is adjusted at the end of each RTT round. If packet drop has been detected in a round, the transmission rate will be re-calculated according to the error rate and other relevant parameters in the round. Otherwise, if the current transmission rate is lower than the maximum transmission rate obtained through the TCP modeling equation, the number of packets to be transmitted in the next round will be increased by one by applying the following equation:

$$\text{Packet\_Interval}_{i+1} = \frac{\text{Packet\_Interval}\_i \times 1}{\text{Packet\_Interval}\_i + 1}$$

We construct a simple rate-based congestion control protocol based on each of the two models, and compare their performance in a series of simulation experiments. We call the rate-based protocol based on the simple model the Simple Rate-based Protocol (SRP) and the one that is based on the complex TCP model the Complex Rate-based Protocol (CRP). Note that if the current transmission rate higher than maximum transmission rate, the transmission rate will decrease immediately to the maximum transmission according to the TCP-friendly equation applied. Lost packets are not retransmitted. RTT and Rto are estimated based on the Jacobson/Karn algorithm as in standard TCP.

In most real-life network configurations, a wireless link is used only in the last hop of the connection. For example, mobile clients might connect to their ISP by mobile phone line and the connection

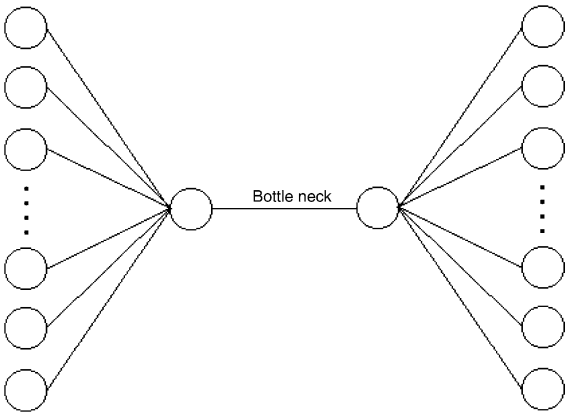

Fig. 4. Dumbbell simulation topology.

Table 1
Simulation parameters of dumbbell topology

Parameter	Value
Bottleneck bandwidth	5 KBps per flow
Bottleneck delay	50 ms
Bottleneck buffer	$5 \times BW \times RTT$ per flow
Bottleneck queue type	RED
Sidelink bandwidth	10 Mbps
Sidelink total delay range	6 ms
TCP Source	TCP/Reno
TCP maximum window	1000
Data packet size	256 bytes
Ack packet size	40 bytes
Use Randomization	True
Simulation length	250 s

is then redirected to connect to the Internet. Since we assume the wired network is reliable, we inject error into the last hop to simulate the presence of a lossy wireless link and examine the effect on the ability of the models to maintain inter-protocol fairness and to predict transmission rate accurately. In this section, we will examine the performance of SRP and CRP in the context of a multi-hop network topology (Fig. 4).

3.3. Simulation topology & parameters

Fig. 4 shows the topology used in the first set of simulation experiments. Circles on the sides represent data sources and data sinks. Sufficient bandwidth is provided for all side link connections so that there is negligible loss and delay on the side links, and bandwidth in the bottleneck link will increase with number of connections. This simulates the effect of a lightly loaded network. Errors will be introduced into the bottleneck channel to simulate the effect of a lossy channel and the effect on TCP friendliness will be examined. All simulation experiments will be run using the ns-2 simulator. TCP Reno is used in all our experiments because it is the ubiquitous protocol on the Internet and also because both the Floyd and Padhye models studied in our paper are based on TCP Reno.

All the experiments described within this section use the simulation parameters in Table 1 unless explicitly stated otherwise.

The TCP-friendliness of the two protocols will be examined first. The number of traffic streams will be increased from 2 to 100 in the first experiment with equal number of traffic streams for TCP traffic and the rate based protocol. Averaged normalized throughout of the each traffic type is used as the performance benchmark for TCP-friendliness. If the normalized throughput is close to 1, then the protocol is sharing bandwidth fairly with TCP.

3.4. Simulation results with no injected errors

Figs. 5 and 6 show the normalized throughput (throughput achieved divided by the bandwidth allocated to the traffic stream) of SRP and CRP in the presence of TCP traffic respectively. It can be seen that SRP grabs more bandwidth share than TCP traffic while CRP traffic obtains less bandwidth than its competing TCP traffic streams. This phenomenon is consistent with our experiments described in Section 3. The simple TCP model on which SRP is based ignores the timeout factor of TCP traffic [7] and overestimates TCP transmission rate; as a result it grabs more than its fair share of bandwidth. From the graphs, it can also be seen that the number of traffic streams has no major effect on TCP-friendliness for both SRP and CRP.

We will now turn to the issue of whether the simple and complex models are adequate in lossy wireless environments.

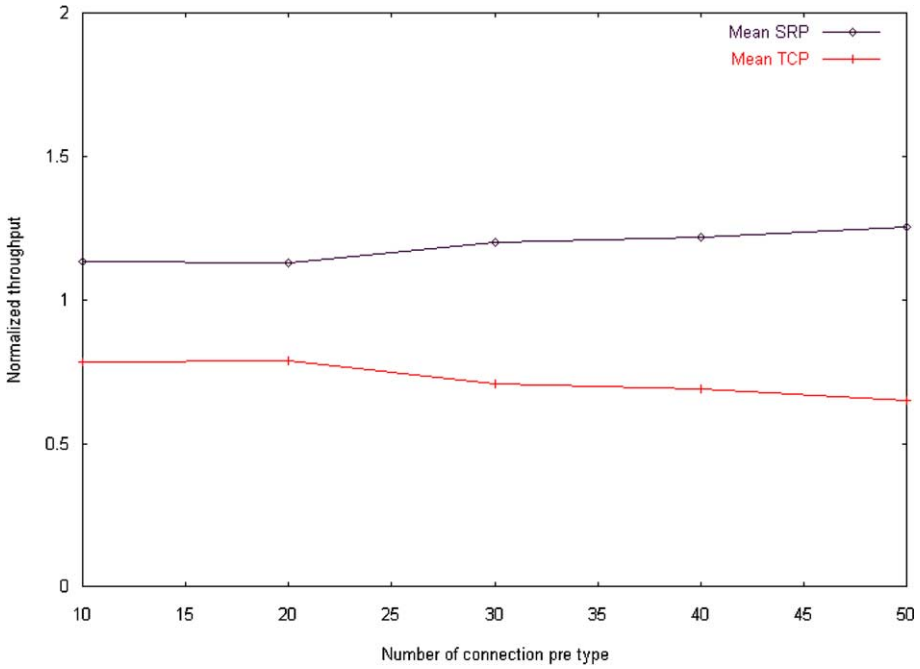

Fig. 5. Normalized throughput of SRP vs number of traffic.

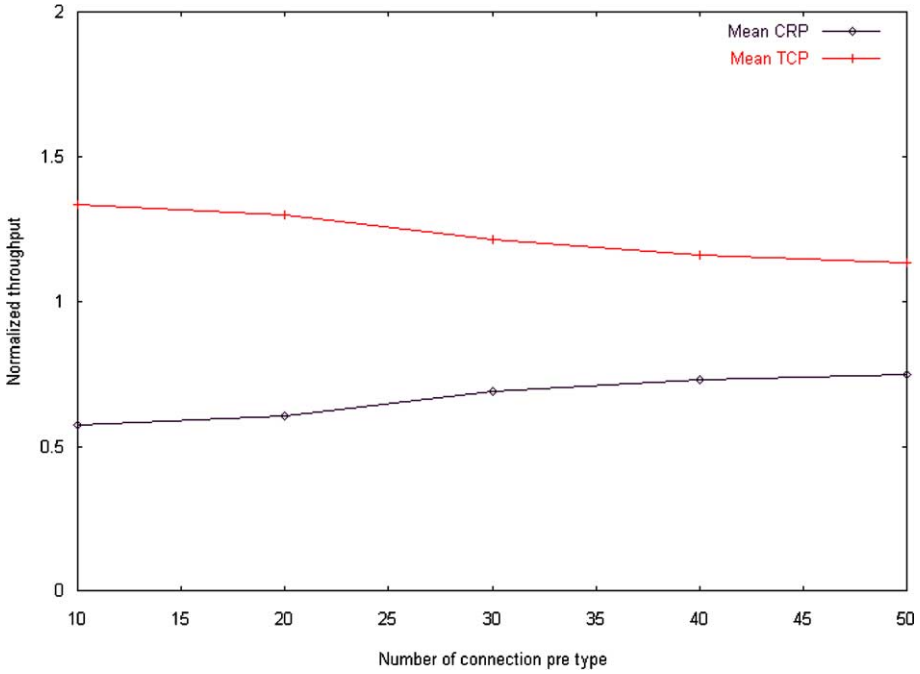

Fig. 6. Normalized throughput of CRP vs number of traffic.

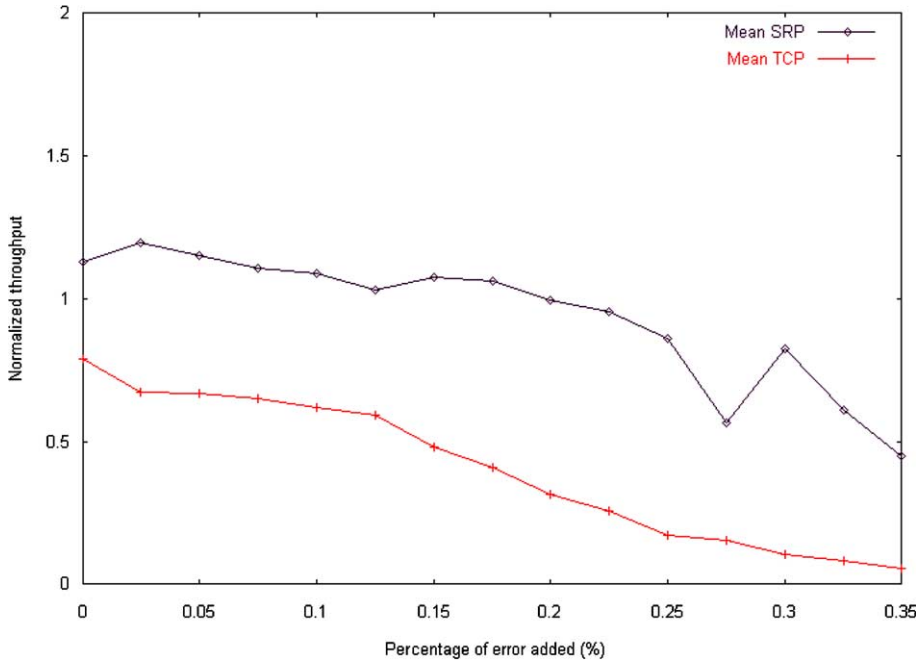

Fig. 7. Normalized throughput of the simple model vs percentage of error added (packet size = 512, RED queue).

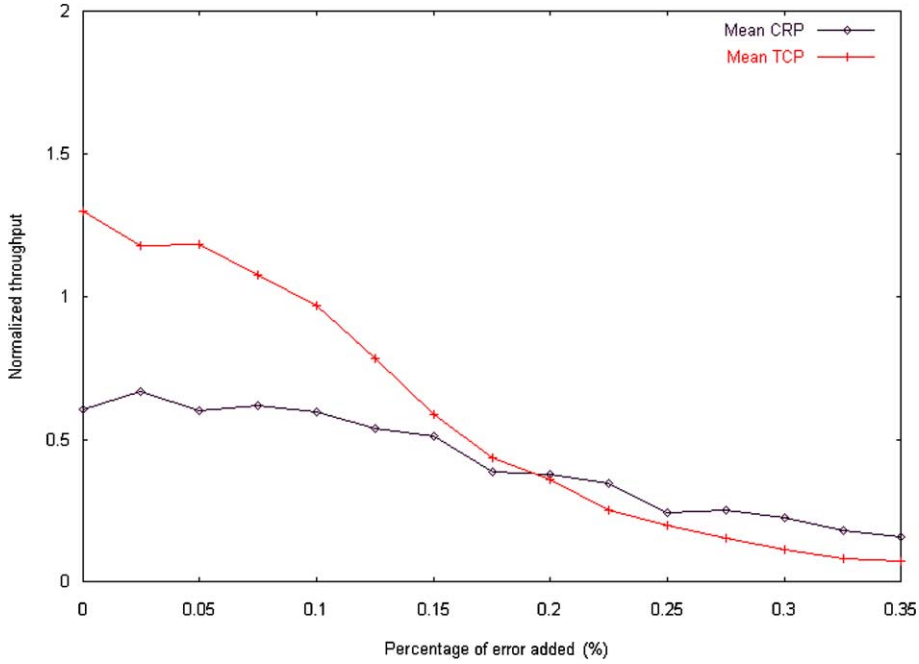

Fig. 8. Normalized throughput of the complex model vs percentage of error added (packet size = 512, RED queue).

3.5. Simulation results with random error

In the experiments described in this section, random error will be added to the bottleneck link of the simulation topology shown in Fig. 4. The percentage of error added will vary from 2.5% to 35% of total number of packets transmitted, with the error generated randomly by the ns-2 simulator. There are 40 traffic streams in the simulation, with 20 streams each for traffic using TCP and the TCP-friendly protocol. Normalized throughput is again the main performance metric for TCP-friendliness.

Figs. 9 and 10 show the effect of random error on the throughput of SRP and CRP in the presence of TCP traffic. Similarly, throughput of SRP also have higher bandwidth share than that of the TCP traffic. It can also be seen that TCP throughput decreases when the injected error rate increases. This is expected since TCP interprets the lost packets as an indication that the network is highly congested and activates its congestion control mechanism consequently. The equation-based protocols should also perform in the same way since they simulate the behavior of TCP. However, it can be seen in Figs. 9 and 10 that the performance of SRP is much better than that of CRP when the error rate is low, and the situation reverses when error rate increases. The reason is that the complex model underestimates the transmission rate of the TCP traffic stream when the error rate is low. As mentioned in a previous section, the most significant difference between the complex model and the simple model is that timeouts are considered in the complex model. In a typical wired network, increase in the error rate implies the network is highly congested and the timeout value will be increased as well. In a wireless link, however, increase in the error rate does not result in an increase in RTT. On the contrary, TCP will interpret the higher error rate as evidence that the network is congested and activates congestion control accordingly. As a result the amount of traffic actually decreases and the RTT of the network decreases.

The complex model, however, makes an incorrect assumption on the relation between error rate and retransmission timeout in this situation, and

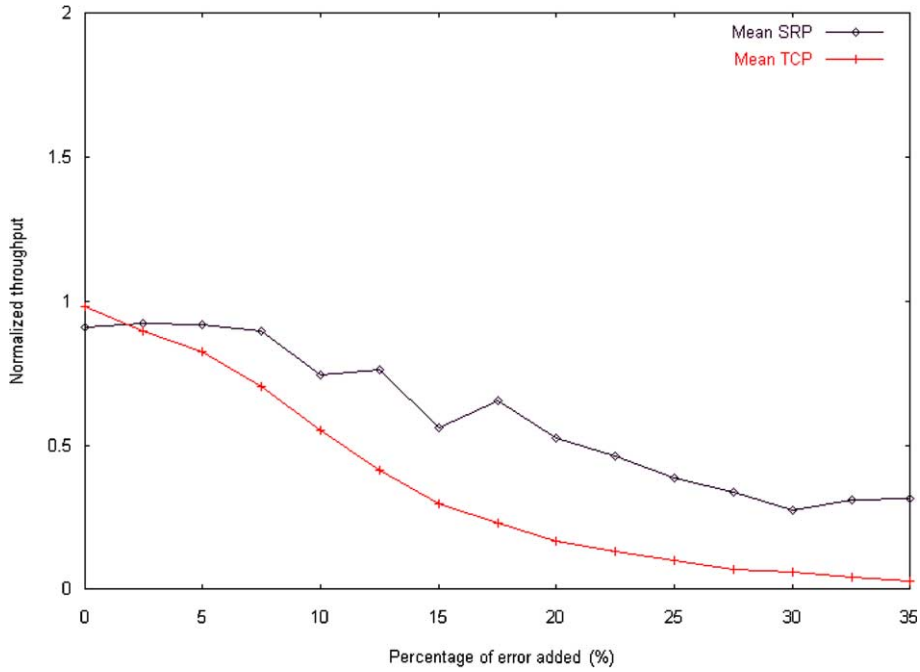

Fig. 9. Normalized throughput of the simple model vs percentage of error added (packet size = 256, RED queue).

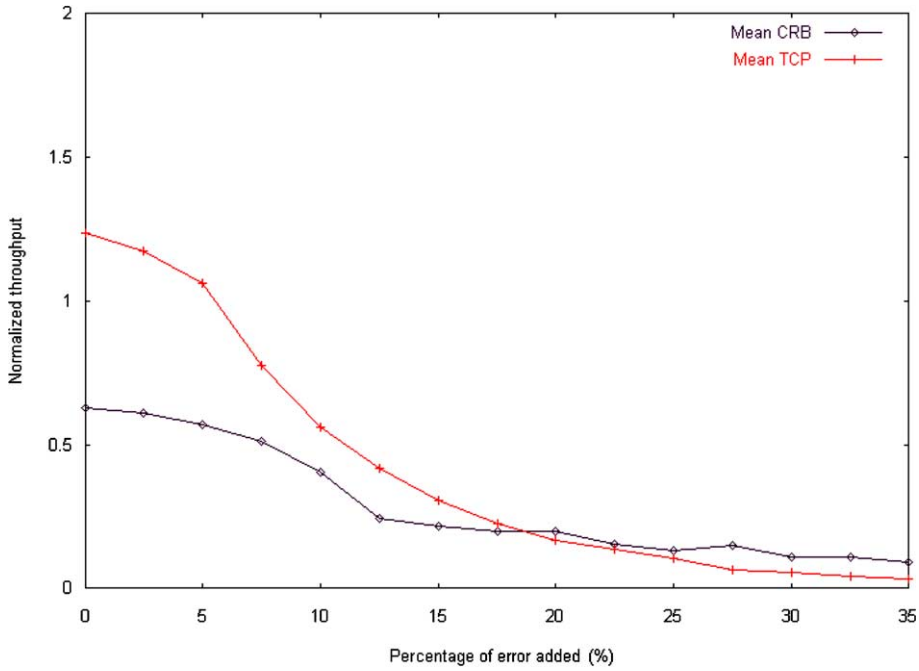

Fig. 10. Normalized throughput of the complex model vs percentage of error added (packet size = 256, RED queue).

over-estimates the impact on TCP timeout and under-estimates the TCP bandwidth share. This is illustrated by Figs. 13 and 14 which show the behavior of the timeout portion and triple duplicate ACKs portion of the denominator in the complex TCP modeling equation for both lossy and non-lossy environments. Note in particular the sharp rises in the bandwidth estimates caused by the timeout portion of the model in the presence of error (Fig. 14).

When the percentage of error increases beyond a certain level, for example, higher than 20%, a large number of packets are dropped. The high packet drop rate decreases the sampling rate of the packet drop signal and reduces the packet drop rate indirectly. As a result, further increases in the percentage of random error added would not increase the recorded error rate proportionally and neither SRP nor CRP estimate the TCP bandwidth share well in this environment.

We also studied the effect of packet size and queueing mechanisms on the above results. Drop tail queue is used instead of RED and the result is shown in Figs. 11 and 12. The results are very similar to those obtained using RED queues. This confirms the findings of other researchers who have made the same observations in similar experiments [4]. In Figs. 7 and 8 we show the results when a larger packet size of 512 bytes is used in the experiments. The same trend is observed, and the deviation of the simple and complex models from TCP bandwidth is even more pronounced than the case with packet size of 256 bytes.

4. Performance of the TCP models for a lossy last-hop wireless link

Proxy cache servers are widely deployed in the Internet between servers and their clients to improve user response time and reduce network traffic. This is particularly common for the transmission of CM due to its high data transmission rate and the need for fast response. The importance of a TCP-friendly rate based protocol in the design of multimedia proxy cache servers has been highlighted recently [5].

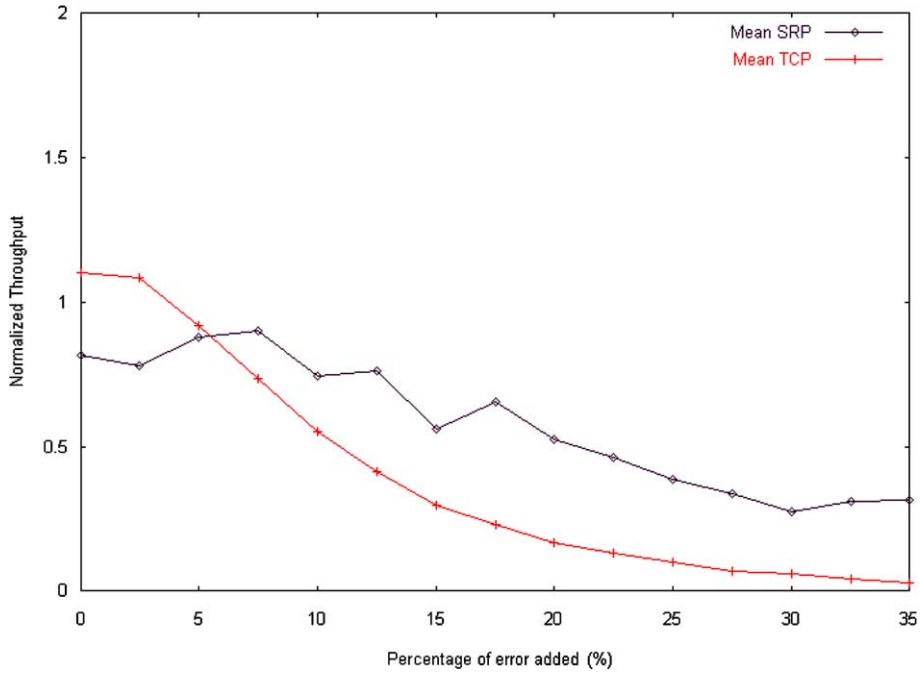

Fig. 11. Normalized throughput of the simple model vs percentage of error added (packet size = 256, drop tail queue).

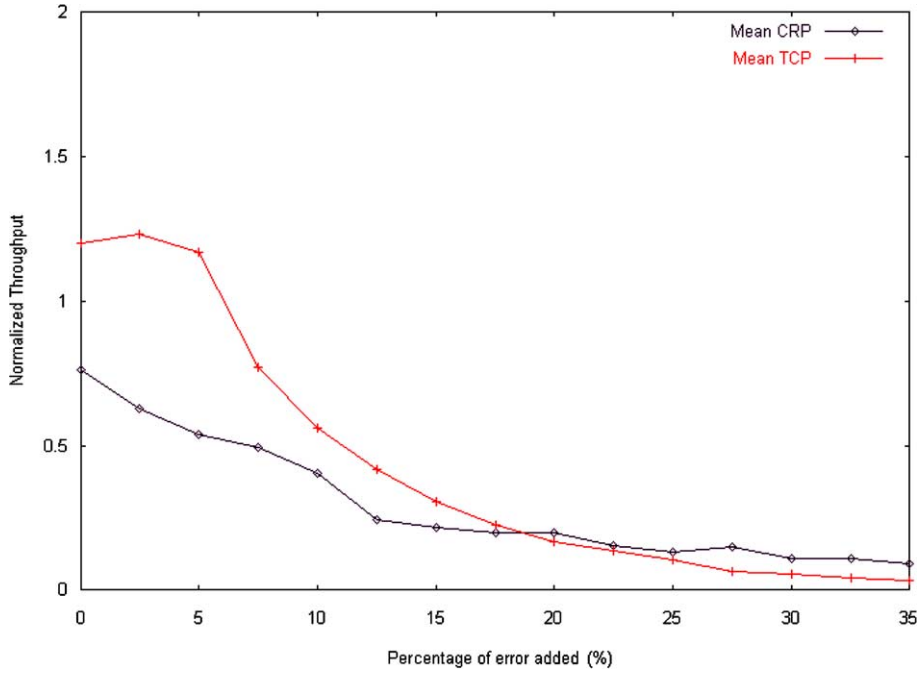

Fig. 12. Normalized throughput of the complex model vs percentage of error added (packet size = 256, drop tail queue).

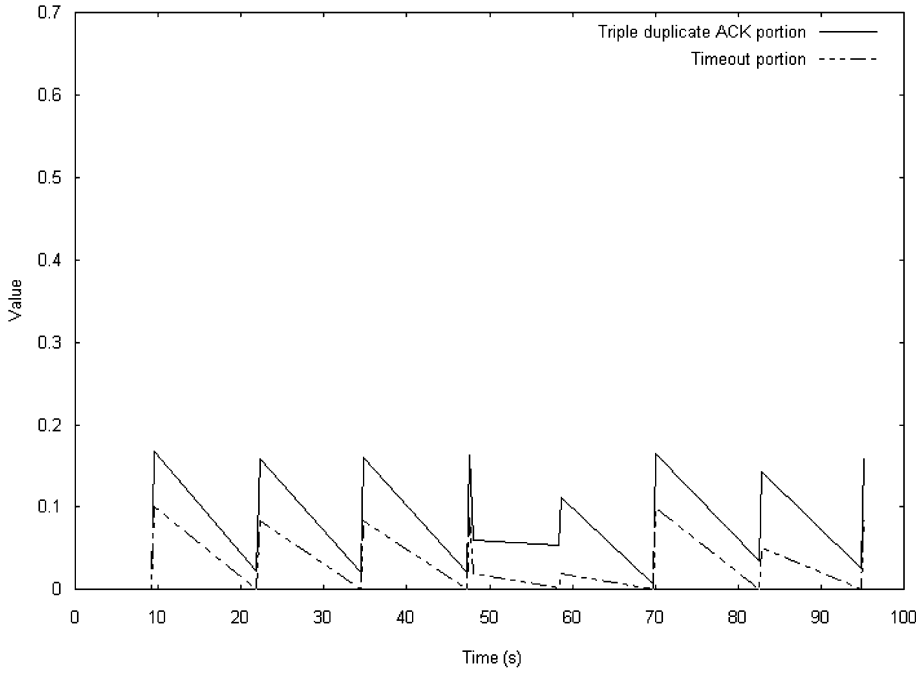

Fig. 13. Estimates for timeout and triple duplicate ACK in the complex model (no error added).

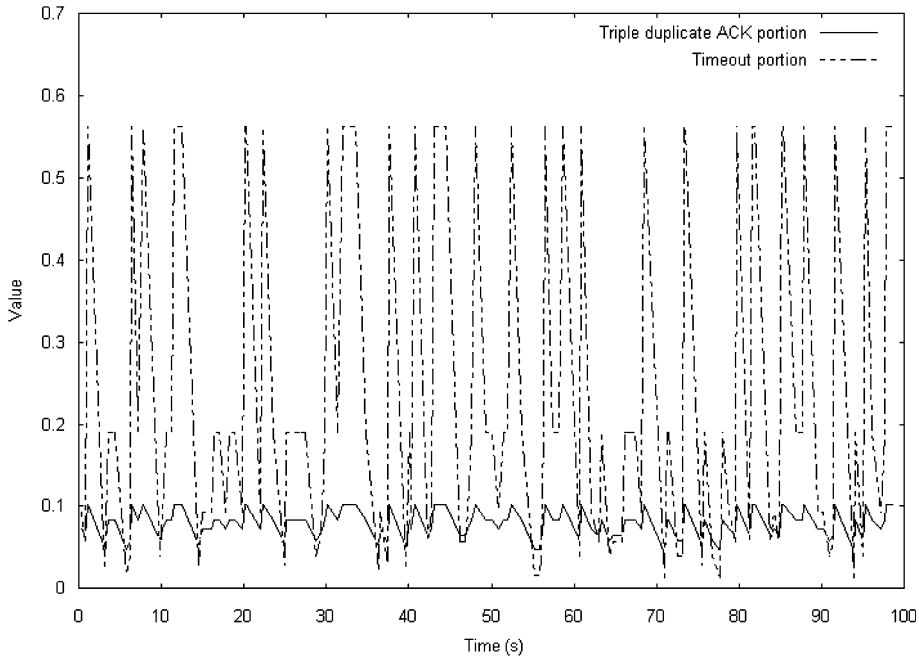

Fig. 14. Estimates for timeout and triple duplicate ACK in the complex model (30% error added).

In this section, we will examine the feasibility of using equation-based rate based protocols such as SRP and CRP in the CM proxy cache server. Besides the impact of wireless lossy channel on TCP friendliness, we are particularly interested in the impact on the RTT variance of the last hop transmission and the interaction between the equation-based TCP-friendly protocols and TCP traffic streams which have just traversed the entire Internet (i.e., end to end TCP traffic).

4.1. Details of the simulation topology and parameters

Fig. 15 shows the topology used in the simulation experiments in this section. As before, circles on both sides of the figure represent data source and sinks respectively. However, this configuration is different from that shown in Fig. 4 in that the data sources only contain TCP traffic streams, while the data generated by the proxy cache server is transported exclusively by the equation-based TCP-friendly protocols (i.e., SRP and CRP). Also, the bandwidth of the wireless bottleneck is kept constant for all the experiments. There is only one traffic stream using SRP or CRP. Since a single mobile client is simulated in this section, it is reasonable to assume that only a single rate based protocol traffic stream is used to carry all CM data from the proxy server to the client. The TCP traffic stream is currently modeled as long term TCP traffic. It is probably more realistic to represent typical TCP traffic in the Internet using on-off TCP traffic (such as those generated by the use of HTTP in Web browsing, although it can be argued that persistent TCP connections are supported in HTTP v. 2). However, we are more concerned with TCP-friendliness between the protocols under study and TCP over a reasonably long period of time, and hence we use steady TCP traffic streams in our experiments.

All experiments within this section use the simulation parameters in Table 2 unless explicitly stated otherwise.

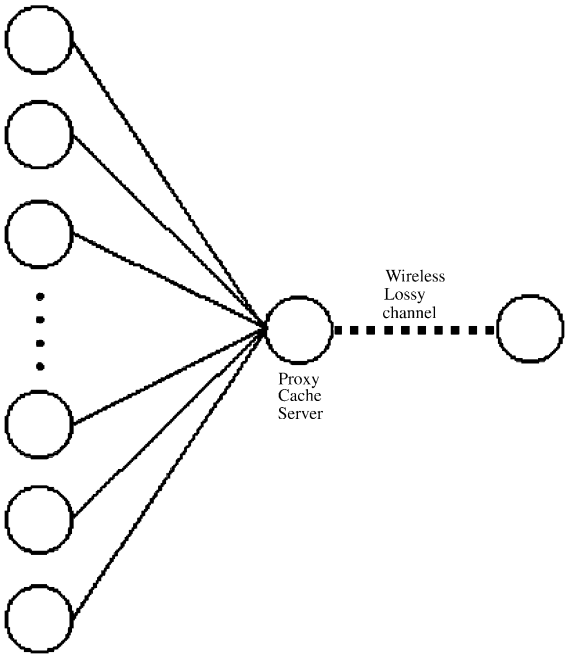

Fig. 15. Simulation topology with proxy cache server.

4.2. Evaluation of the protocols in the absence of end to end TCP traffic

In our first experiment, we evaluate the performance of the protocols in the absence of end to end TCP traffic, i.e., the TCP traffic originates from the proxy cache server. Figs. 16 and 17 show the normalized throughput of SRP and CRP respectively. From Fig. 16, it is clear that in SRP the throughput of the protocol continues to grow with increasing number of TCP traffic stream even after the link is congested. On the other hand, by taking timeout into consideration, the complex

Table 2
Simulation parameters of last-hop topology

Parameter	Value
Last hop bandwidth	150 kbps
Last hop delay	200 ms
Last hop buffer	$4 \times BW \times RTT$
Last hop queue type	RED
Sidelink bandwidth	10 Mbps
Sidelink total delay range	6 ms
TCP Source	TCP/Reno
TCP maximum window	1000
Use Randomization	True
Simulation length	250 s

model achieves fairness between TCP and the TCP-friendly equation-based protocol. The reason is as follows. RTT can have a large variance within a short period in the presence of a network buffer queue. When the queue is empty, RTT detected by SRP is small and the protocol instantly increases the data transmission rate. However, once the channel is fully utilized, data packets have to wait in the network queue and the RTT is now the sum of the normal transmission delay and the queuing delay. This increase in RTT results in packet timeout for both rate-based protocol and TCP traffic. Since the simple TCP model does not consider the effect of timeout, SRP will continue to grab a greater share of the bandwidth even though the last hop is heavily loaded.

In the next set of experiments random error is added to the last hop to see the effect of channel error on the two protocols. The number of TCP traffic streams is set to 10 and number of rate based protocol stream is still set to 1. Figs. 18 and 19 show the throughout of TCP, SRP and CRP against different error rates in the last hop.

From the experiments, we can see that TCP traffic performs as expected. The share of bandwidth consumed by TCP keeps decreasing when the percentage of error added to the last hop channel increases. Moreover, consistent with the experiments in the previous sections, TCP bandwidth share is between that of SRP and CRP. In addition, SRP keeps diverging from the TCP bandwidth when the error rate is high while CRP performs worse when the error rate low. This means that applying TCP-friendly rate-based protocol at the last hop channel with random error added does not have much effect on the performance of both TCP models.

4.3. TCP-friendliness of rate-based protocol with end to end TCP traffic

The next scenario to be investigated is the TCP friendliness of the protocols in the presence of end to end TCP traffic, i.e., the TCP traffic originates elsewhere and has traversed the Internet before arriving at the proxy cache server as its last hop.

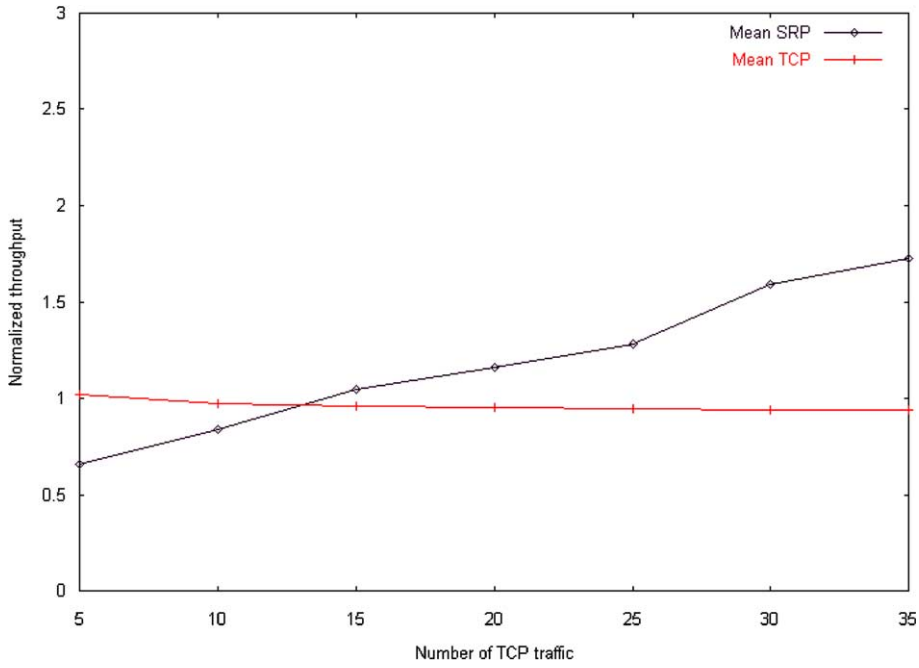

Fig. 16. Normalized throughput of SRP on the last hop vs number of TCP streams.

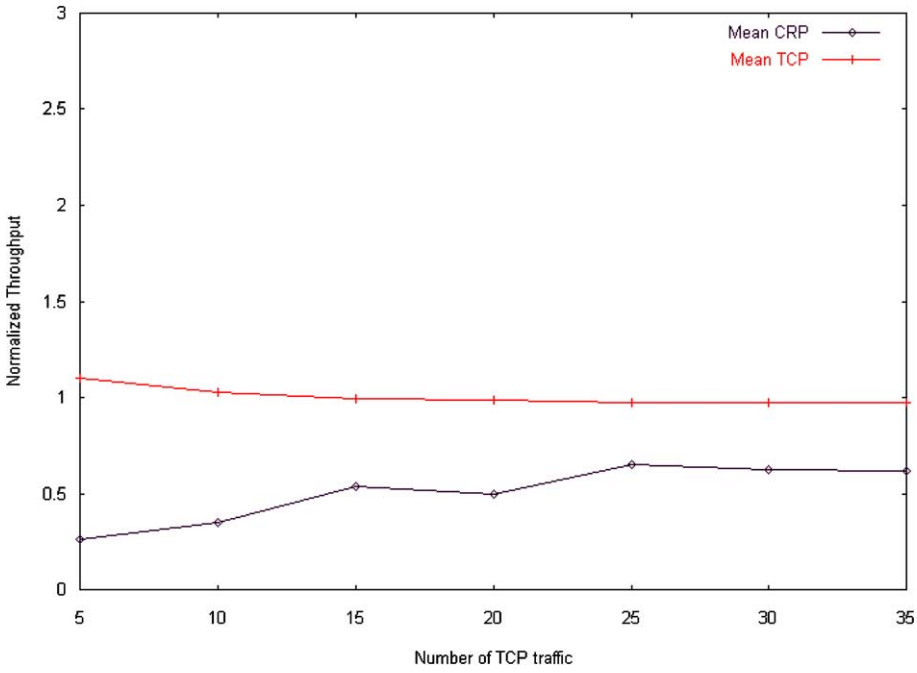

Fig. 17. Normalized throughput of CRP on the last hop vs number of TCP streams.

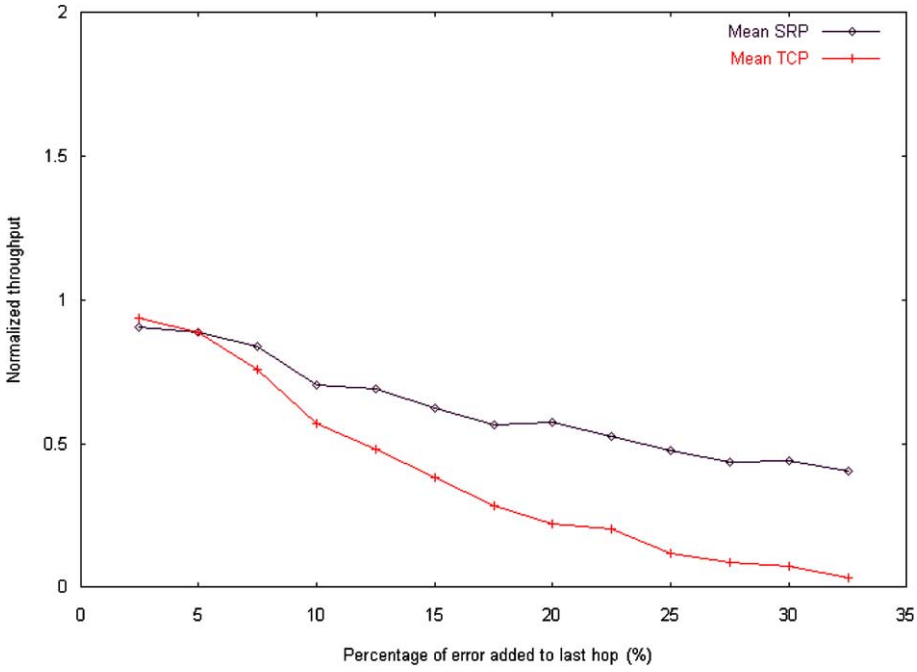

Fig. 18. Normalized throughput of SRP model on the last hop vs % of error added.

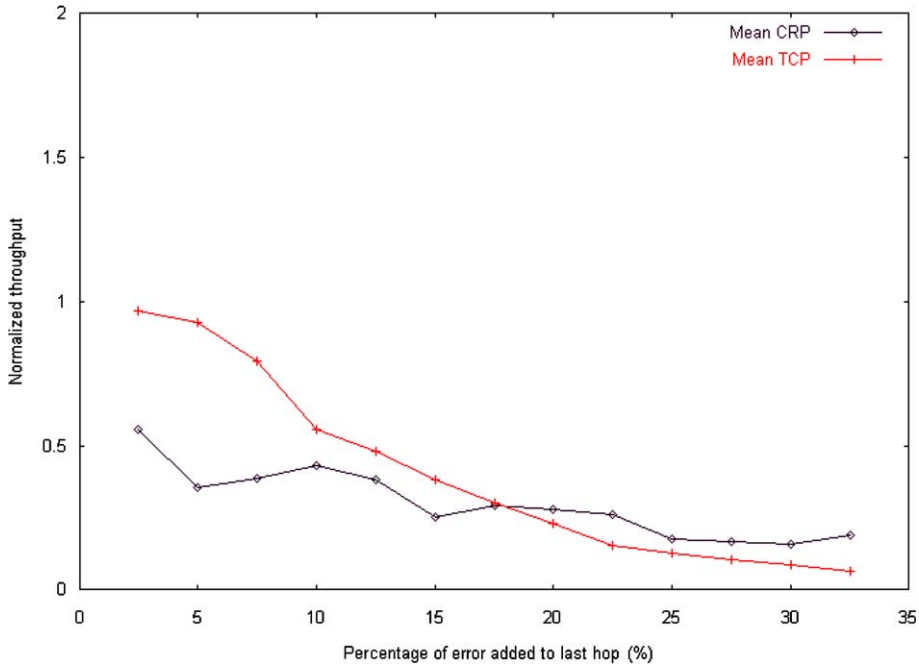

Fig. 19. Normalized throughput of CRP on the last hop vs % of error added.

Other than error rate p, the round trip time RTT is an essential factor for the TCP-friendly rate based protocol in estimating the bandwidth of the TCP traffic and achieving TCP-friendliness. However, the significant difference between RTT of last hop traffic and end-to-end traffic may result in incorrect estimation of TCP friendliness of the equations, and this is where we will now focus our attention.

The topology shown in Fig. 15 will be used for this set of experiments. To simulate the effect of the end to end TCP traffic stream, the propagation delay for the side links are increased and varied for the simulation experiments. Table 3 shows the default simulation parameters for the experiments in this section.

The simulation experiments are set up so that competition for network resources only occur on the last hop so that the effect of changes in RTT of neighboring TCP traffic on CRP and SRP can be studied. This is achieved by providing relatively large bandwidth and queue sizes to the TCP traffic on the side links to avoid any segment drops in that part of the network.

Figs. 22 and 23 show results of the simulation experiments using a packet size of 1024 bytes. CRP is more TCP-friendly than SRP in this experiment. These figures also highlight that TCP traffic throughput is very dependent on the propagation delay of the side links. In Fig. 20, we can see a large variation in protocol throughput for SRP. Moreover the throughput of the TCP traffic ecreases steadily as the propagation delay

Table 3
Simulation parameters of simulation with end-to-end TCP

Parameter	Value
Bottleneck bandwidth	150 kbps
Bottleneck delay	200 ms
Bottleneck buffer	$4 \times BW \times RTT$
Bottleneck queue type	RED
Sidelink bandwidth	1 Mbps
TCP Source	TCP/Reno
TCP maximum window	1000
Number of TCP-friendly rate-based protocol streams	1
Number of TCP traffic streams	10
Use Randomization	True
Simulation length	250 s

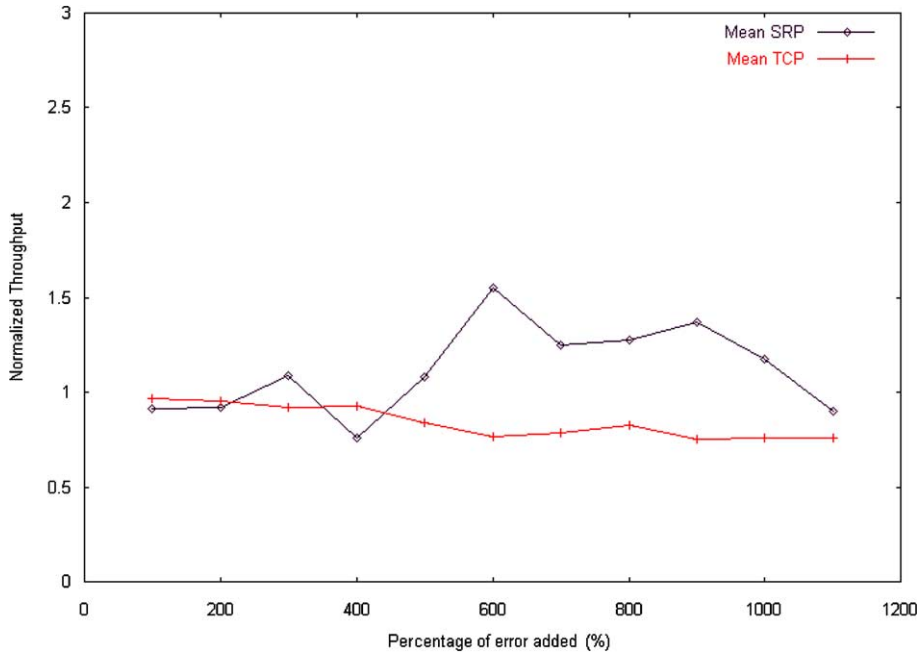

Fig. 20. Normalized throughput of SRP on the last hop vs propagation delay of TCP traffic (packet size = 256).

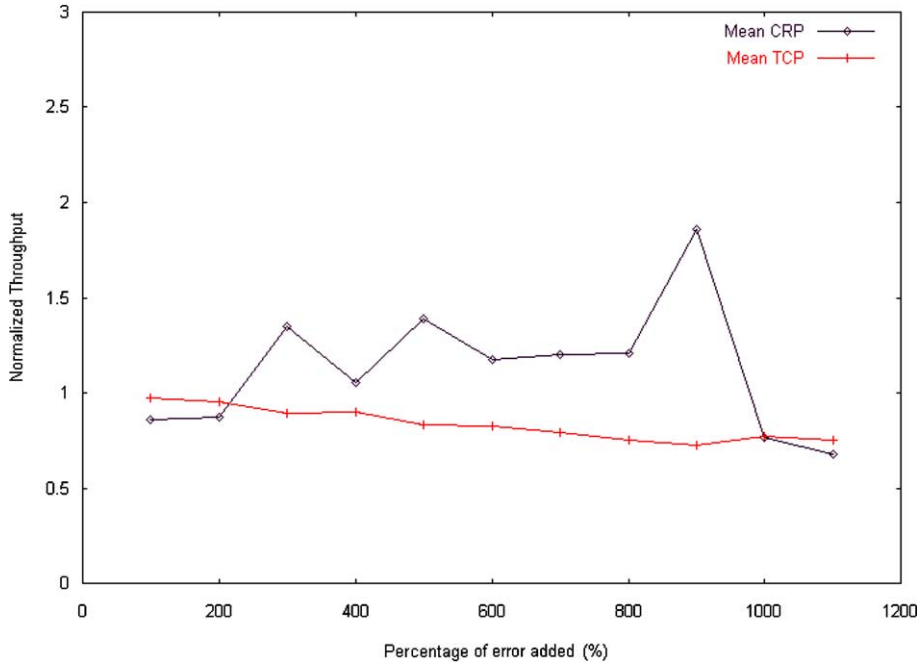

Fig. 21. Normalized throughput of CRP on the last hop vs propagation delay of TCP traffic (packet size = 256).

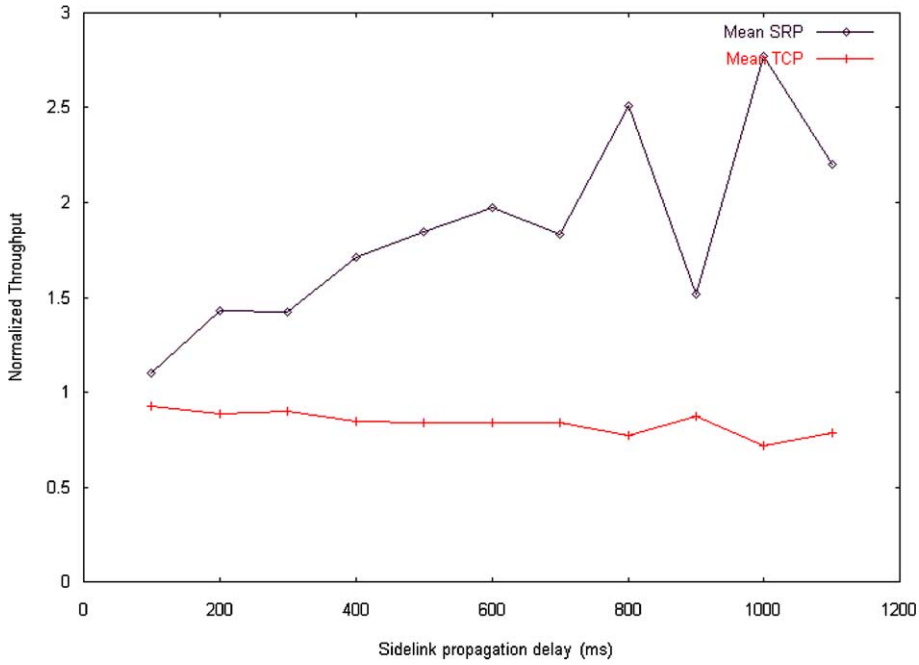

Fig. 22. Normalized throughput of SRP on the last hop vs propagation delay of TCP traffic (packet size = 1024).

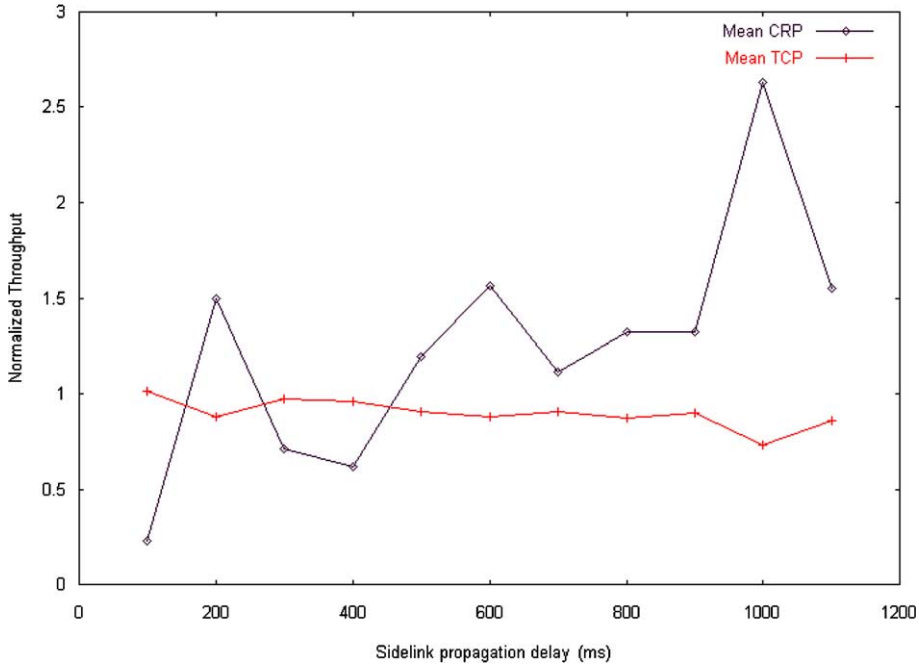

Fig. 23. Normalized throughput of CRP on the last hop vs propagation delay of TCP traffic (packet size = 1024).

increases; this trend is even more obvious in Fig. 20. As mentioned in the beginning of this section, RTT is a decisive parameter in both the simple and the complex models. When the protocol deployed in the last hop has a direct connection from data sink to data source, the recorded RTT on the last hop will be far less than the end to end TCP traffic streams. Since the TCP-friendly protocol estimates the transmission rate of the TCP streams based on its own small RTT value, there is a clear mismatch with the result that bandwidth allocated to TCP will decrease. This effect is not as pronounced when the packet size is reduced to 256 bytes (Figs. 20 and 21), showing some dependency of this effect on packet size.

5. An enhanced TCP model for links with random errors

As we have seen in the previous sections, neither model performs very well in a lossy network environment. In this section, we will present a modified model that will estimate TCP bandwidth better in such an environment. Since the complex model is a more detailed model, we have chosen to use it as the basis of our enhancement.

The complex TCP model makes a fairly large number of simplifying assumptions so that it can track the behavior of TCP using relatively simple formula. A basic assumption of the complex TCP model is that when the k th segments is dropped, all segments after the dropped segment will also be discarded, which is essentially an attempt to emulate the drop tail queue behavior of routers when the network is congested [7]. In this scenario, packet timeout after packet drop will be triggered in any one of the following three cases. First, *cwnd*, the current congestion window size, is smaller than three. Second, the drop is on the second segment or before. These two conditions imply that no more than two duplicated acknowledgments can be received by data source and hence triple duplicated acknowledgement will never be triggered. Third, only two segments were successfully transmitted in the last round of transmission. This situation is same as previous two cases but occurs in the last round (refer to [7] for a definition of a round and other details of the model which we will not duplicate here).

The above assumptions are not valid when random error exists in the communications channel. In a channel with random error, packet drops are not correlated, which violates the basic assumption of the complex model. Consider the transmission history of TCP traffic on a lossy channel in Fig. 24. Initial packet drop is followed by three successfully transmitted packets and hence triple duplicated acknowledgment will be triggered. However, in the complex TCP model, this case will be considered as a packet timeout and the averaged timeout interval will be incorporated in TCP throughput calculation. This incorrect assumption leads to a low throughput estimate in TCP modeling formula in lossy data transmission.

In order to remedy this problem, let us first consider the events in one round of transmission. Assume that the current congestion windows size is w and the packet drop rate is p. Let Q be the event that a TCP back off is triggered by packet timeout. The probability that a timeout occurs in this transmission round, given at least one packet has been dropped, is

$$P(Q) = \sum_{k=0}^{2} {}_{w}C_{k} \frac{(1-p)^{k} p^{(w-k-1)}}{1-(1-p)^{w}}$$

where the factor ${}_{w}C_{k}$ stands for the number of combinations that k packets are successfully

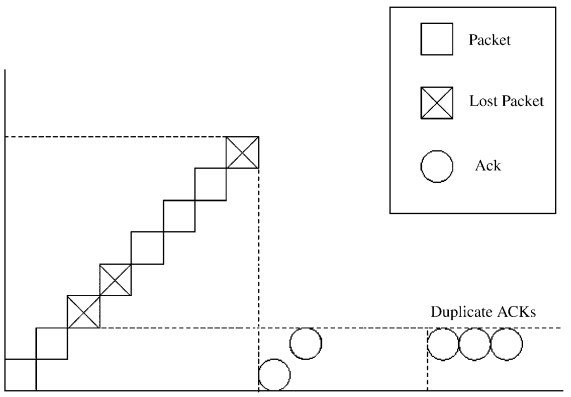

Fig. 24. Example of triple duplicate acknowledgements triggered by packet loss.

transmitted out of the w packets. Note that such a timeout occurs only if less than three acknowledgments has been received, otherwise triple duplicate acknowledgement will be triggered. This equation is the probability of not more than two packets successfully transmitted after a packet is dropped.

After rearrangement of the equation, we have

$$P(Q) \approx \frac{(2-p)p^{(w-3)}}{2(1-(1-p)^w)} \quad \text{where } w = \sqrt{\frac{8}{3p}}$$

After this refinement, the complex TCP modeling formula becomes:

$$T = \frac{s}{RTT\sqrt{\frac{2p}{3}} + R_{to}\left(\frac{(2-p)p^{(w-3)}}{2(1-(1-p)^w)}\right)p(1+32p^2)}$$

6. Performance of the modified TCP-friendly model

In this section, we will examine the performance of the modified TCP modeling formula using experiments similar to those we have done in Section 3.

6.1. Baseline experiments

Just as in Section 3.1, we examine the bandwidth estimation properties of the complex model and our modified TCP model by generating TCP traffic using the ns-2 simulator and substituting into the modeling equation to produce TCP bandwidth estimation. The results are shown in Figs. 25 and 26.

From the figures, we can see clearly that the complex TCP model tends to underestimate the available bandwidth while the modified TCP model is more accurate in its estimate. Next we examine the performance of the two models in the presence of errors.

Fig. 27 shows the deviation of estimated TCP bandwidth from actual bandwidth. We can see that the modified model is much more accurate, particular at error rate less than 20%. Performance is not as good beyond 20% added error;

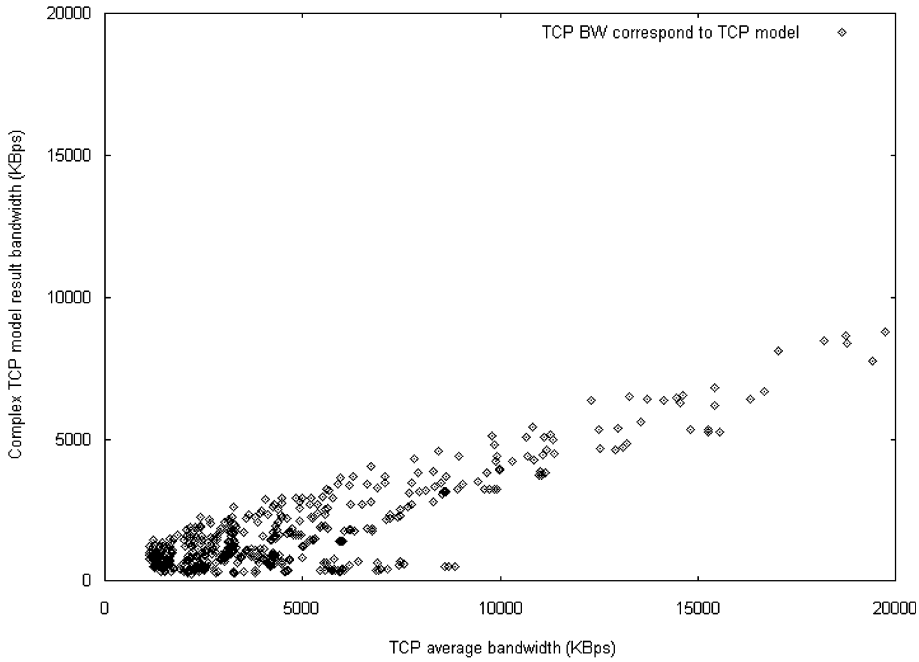

Fig. 25. Estimated bandwidth of the complex TCP model vs the measured bandwidth.

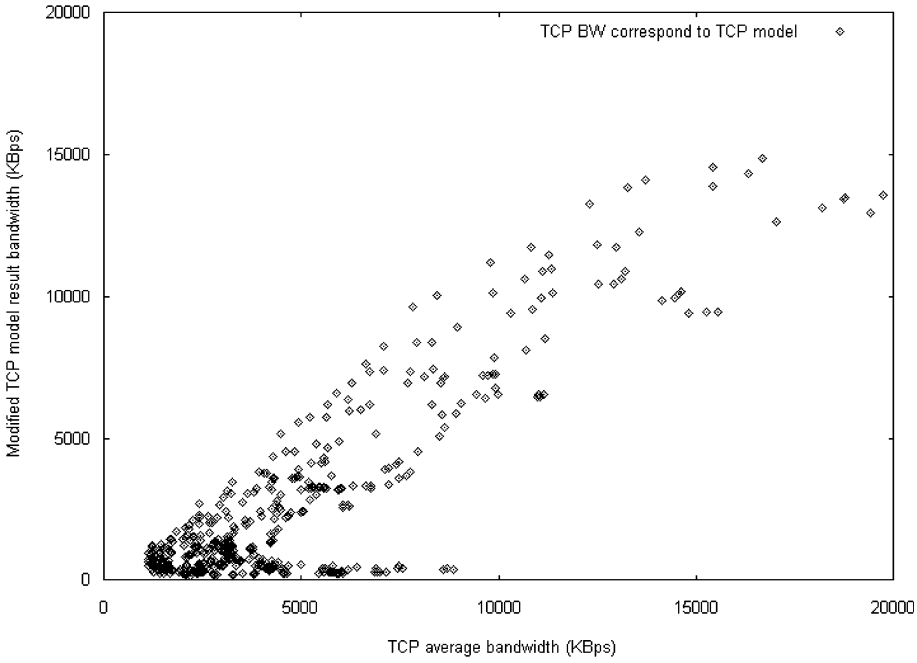

Fig. 26. Estimated bandwidth of modified TCP model vs the measured bandwidth.

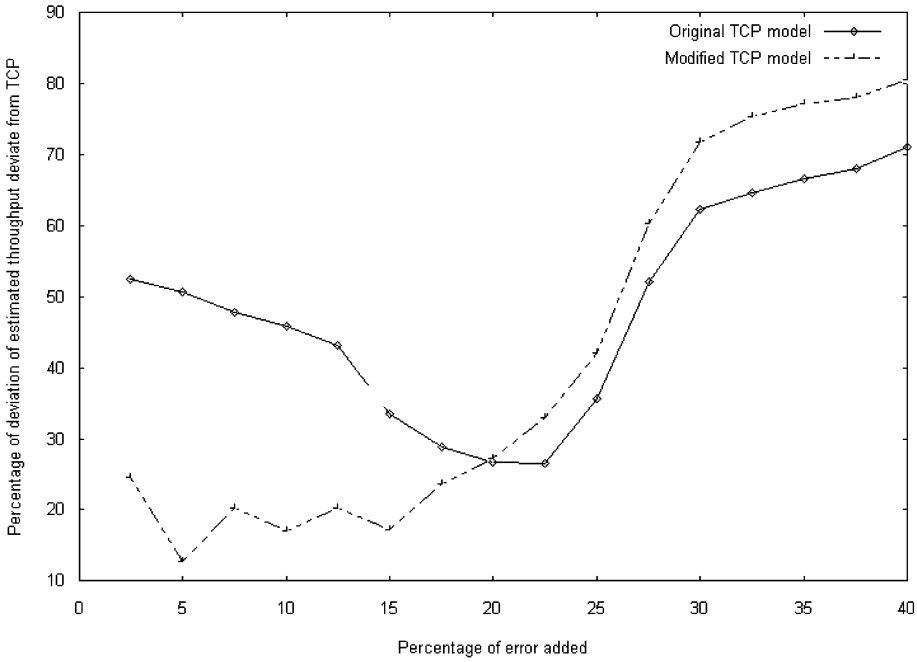

Fig. 27. Percentage of deviation of the throughput of the modified model from actual TCP throughput vs percentage of error added.

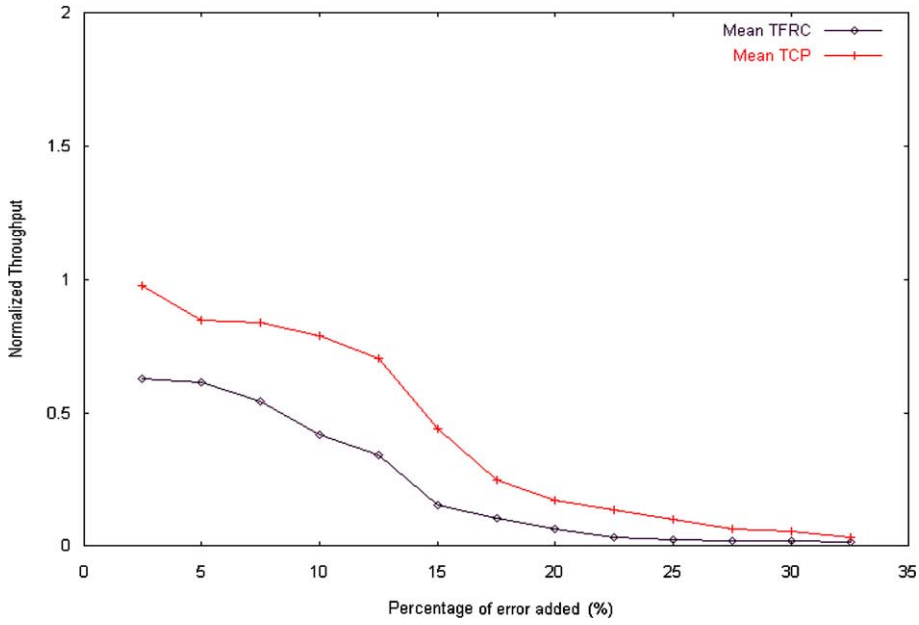

Fig. 28. Normalized throughput of the TFRC vs percentage of error added for the dumbbell topology.

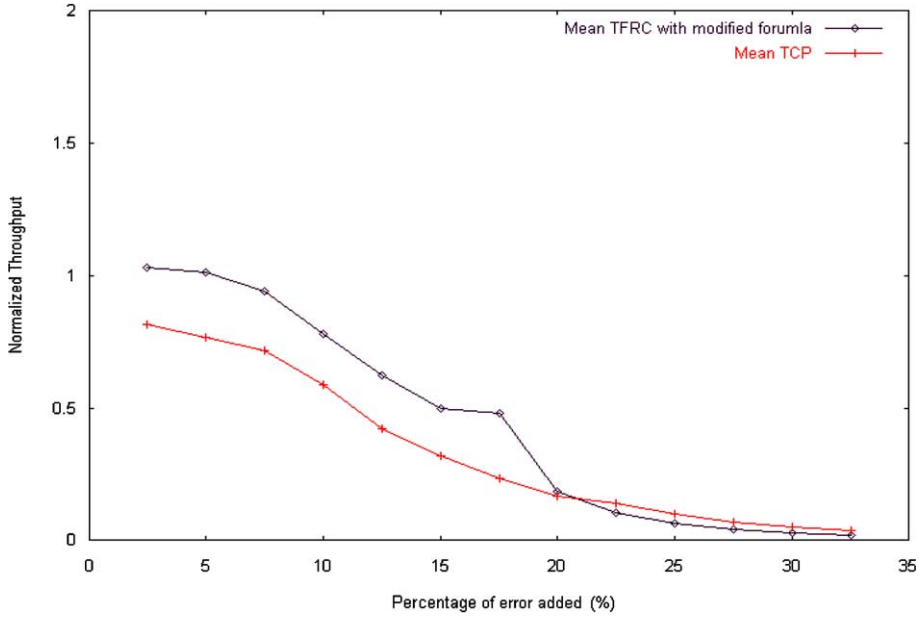

Fig. 29. Normalized throughput of the TFRC with modified TCP model vs percentage of error added for the dumbbell topology.

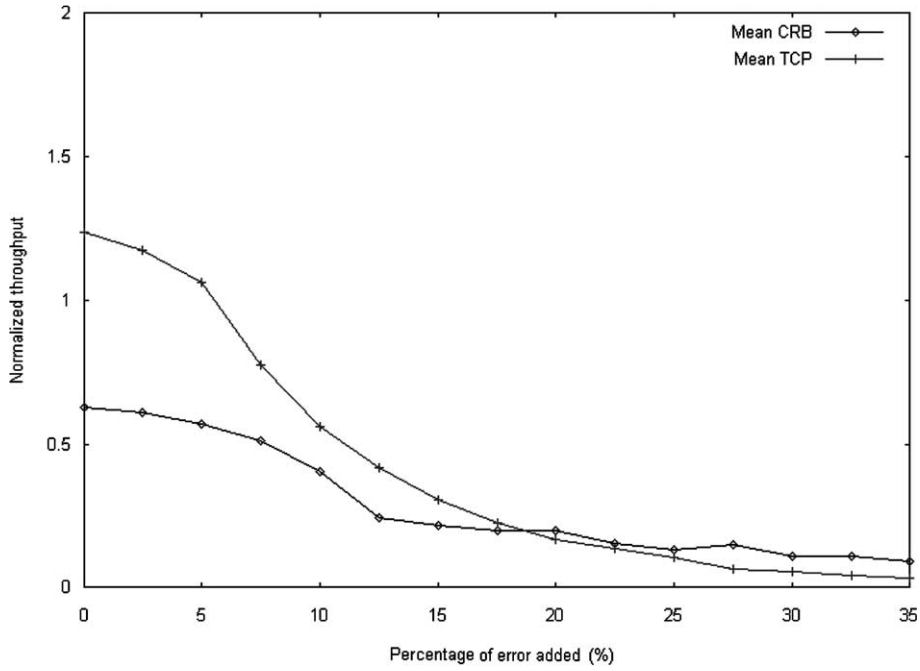

Fig. 30. Normalized throughput of CRP vs % of error added.

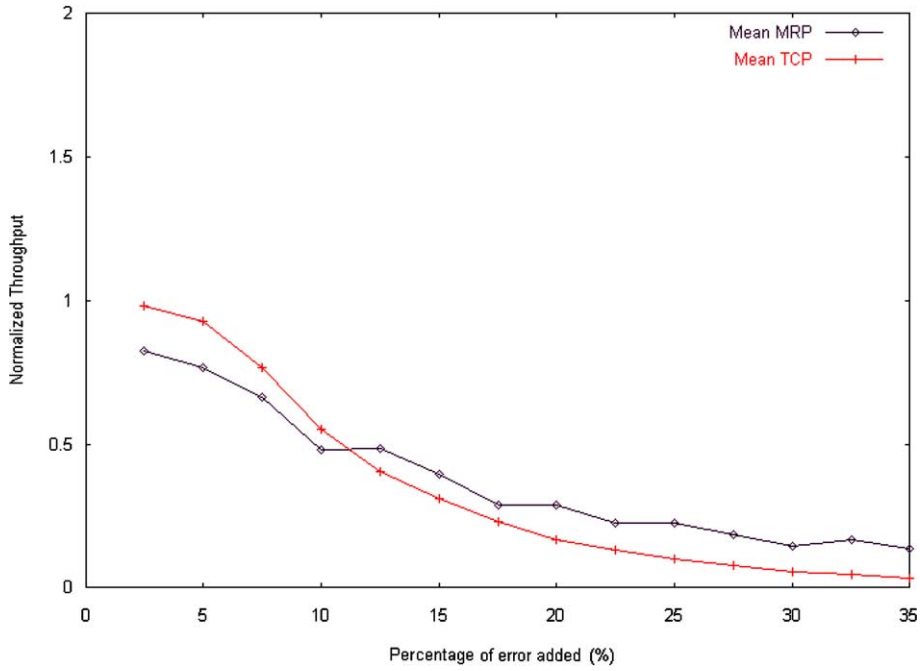

Fig. 31. Normalized throughput of the MRP vs % of error added.

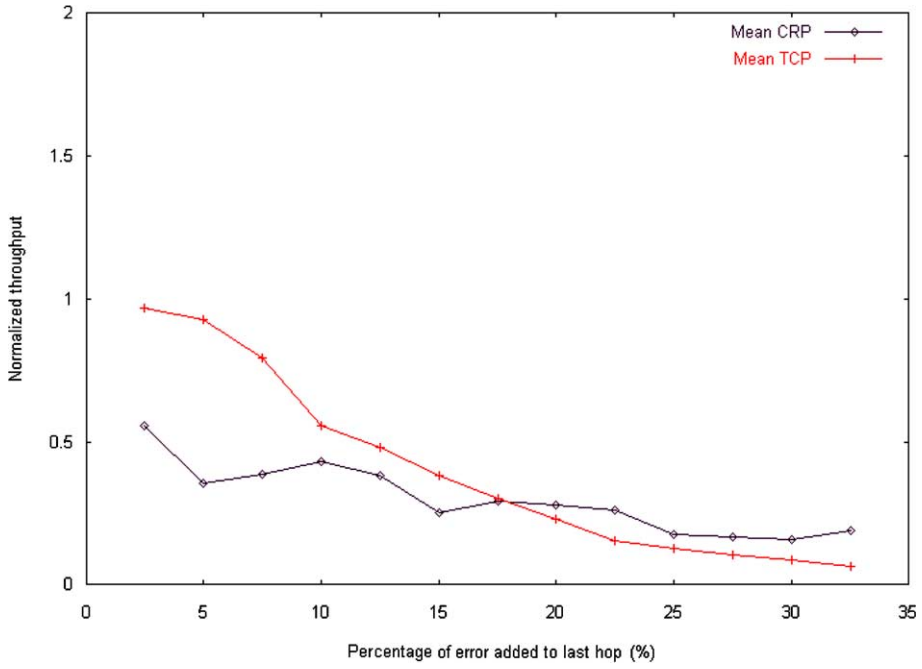

Fig. 32. Normalized throughput of CRP vs percentage of error added under the last hop topology.

however at such a high error rate it is very difficult to estimate TCP bandwidth correctly and a link with such a high error rate is not likely to be very useful for data transmission purposes.

6.2. Simulation experiment for a typical network topology

We now construct a rate based protocol based on the method described in Section 3.2 and call it the Modified Rate based Protocol (MRP). The performance of MRP will be compared with CRP to demonstrate the effectiveness of the modified TCP model we proposed in the previous section. The network topology in Fig. 4 is used. The simulation topology and parameters are the same as that described in Section 3.3 and Table 1.

The result is shown in Figs. 28 and 29. Both figures show that throughput of network connection decreases with error rate added increases. This proves that random error does decrease the throughput of network traffic. CRP throughput, represented by dotted line in Fig. 28, deviates significantly from mean TCP throughput, represented by the solid line, especially when the added error rate is low. Both throughput decreases when the error rate increases, but the TCP throughput decreases much more sharply in the case of CRP. We can conclude that MRP, based on the modified TCP modeling formula, achieves both TCP-friendliness and provides an accurate TCP throughput estimate.

6.3. Simulation experiments for a lossy last-hop wireless link

In this section we use the topology (Fig. 15) and parameters (Table 2) described in Section 4.1 to study the behavior of the modified TCP model for a lossy last-hop wireless link.

From Figs. 30 and 31 once again we see that MRP clearly maintains a throughput similar to that of the TCP connections, demonstrating that

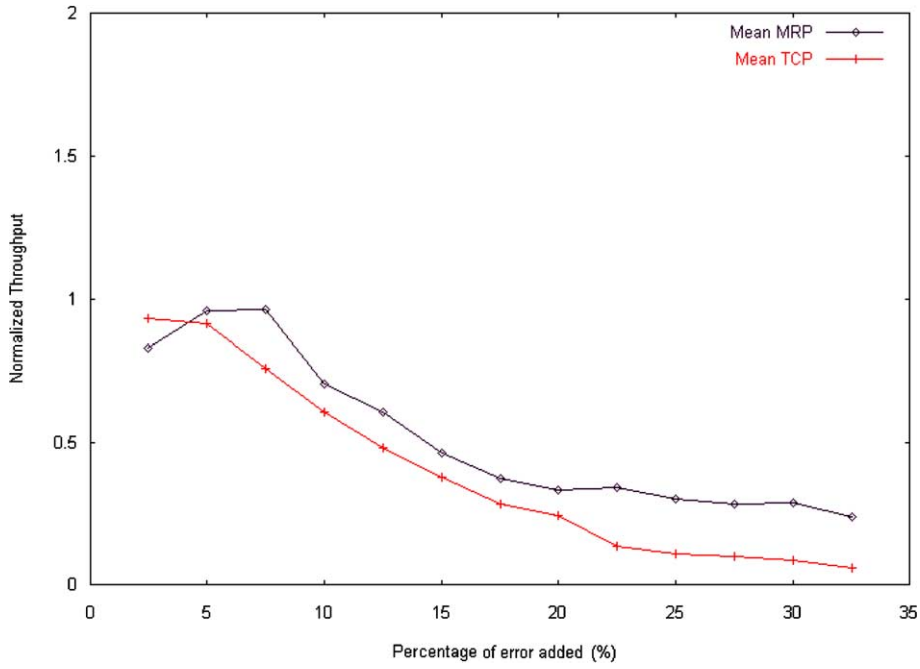

Fig. 33. Normalized throughput of MRP vs percentage of error added under the last hop topology.

it is more accurate than CRP (which consistently underestimates TCP bandwidth when error added is below 20% and represents that of a typical operating environment) in estimating TCP bandwidth.

Thus we can conclude from the results in this section and the previous one that the modified TCP model is superior to the complex TCP model in the presence of random errors under a variety of network scenarios.

6.4. Experiments on existing TCP-friendly protocol

In order to further prove the modification can estimate TCP traffic throughput more accurately, we will apply the modified TCP modeling formula to a well known TCP friendly rate based protocol, TFRC [8]. The simulation experiment in Section 6.2 is repeated, except that we use a modified version of TFRC based on our model and compare it with the original TFRC.

The results are shown in Figs. 32 and 33. Fig. 32 shows that TFRC also underestimates TCP throughput, which indicates that it is also affected by the incorrect assumptions on packet drop pattern in a lossy link. It can be seen in Fig. 33 that the modified TFRC achieved a higher throughput and that both the share of the link between the modified TFRC and TCP traffic streams is reasonably equitable.

7. Conclusion and evaluation

In this paper, the performance of two simple TCP-friendly rate based protocols based on two different models under wireless lossy channel is compared under a variety of network scenarios. Our simulation result shows that the simple TCP model formula tends to overestimate TCP bandwidth share while the complex TCP model tends to underestimate TCP bandwidth share. Moreover, the simple TCP model has the better estimate

when the amount of random error added is low, while the complex TCP model works better at higher error rates. The reason for this phenomenon is that the effect of timeout is considered in the complex TCP model. In a wireless channel, a high error rate has little effect on RTT (unlike the case of high error rate due to congestion). In contrast, high error rate in lossy channel actually results in a smaller RTT for the rate based protocol since the TCP traffic streams will back off and decrease transmission. This misinterpretation of the error rate leads to a wrong bandwidth estimate in the case of the complex model. Note however that if the injected random error rate is very high, neither model performs particularly well.

Another focus of this paper is applying an equation-based TCP-friendly protocol as the last hop transmission protocol for streaming data to mobile client over a wireless link. Based on simulation experiments, it is found that CRP can estimate TCP traffic accurately when error rate added is high, while SRP performs better when the error rate is low. Moreover, TCP-friendliness for both models decreases in the presence of TCP traffic with longer RTT. This shows the substantial difference between the RTT of equation-based TCP-friendly protocol deployed on the last hop and the end-to-end TCP traffic affects TCP-friendliness significantly.

We then propose a modified TCP model that provides a more accurate estimate of TCP bandwidth in the presence of random errors. Using a series of experiments based on different topologies, it is demonstrated that the modified TCP model does perform better when the amount of error added is reasonably low (less than 20%).

Acknowledgement

The work described in this paper was supported by a strategic research grant from the City University of Hong Kong [Project no. 7001601].

References

[1] R. Koenen, Mpeg-4 Overview, http://dorgo.ceslt.stet.it/mpeg/standard/mpeg-4/mpeg-4.htm.

[2] R. StevensTCP/IP Illustrated, vol. 1, Addison-Wesley, 1994.

[3] S. Floyd, M. Handley, J. Padhye, A Comparison of Equation-based and AIMD Congestion Control, May 2000.

[4] Y. Yang, S. Lam, General AIMD Congestion Control. Tech. Rep. TR-2000-09, University of Texas at Austin, May 2000.

[5] R. Rejaie, M. Handley, D. Estrin, RAP: An end-to-end rate-based congestion control mechanism for realtime stream in the Internet, in: Proceedings of IEEE Infocom, 1999.

[6] S. Floyd, J. Mahdavi, TCP-friendly unicast rate-based flow control. Note sent to end2end-interest mailing list, January 1997.

[7] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, Modeling TCP throughput: A simple model and its empirical validation, IEEE/ACM Transactions on Networking 8 (2) (2000), April.

[8] S. Floyd, M. Handley, J. Padhye, J. Widmer, Equation-Based Congestion Control for Unicast Applications. in: Proceedings of ACM SIGCOMM, August 2000.

[9] M. Matthew, J. Semke, J. Padhye, The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm. in: Proceedings of ACM SIGCOOM, July 1997.

[10] E. Chan, S.W. Ng, A Rate-based Streaming Protocol for Wireless Networks, in: Proceedings of IPDPS 2001.

[11] S. Hassan, M. Kara, Simulation-based Performance Comparison of TCP-Friendly Congestion Control Protocols. in: Proceedings of PEW 2000.

[12] V. Paxson, Automated Packet Trace Analysis of TCP Implementations, in: Proceedings of ACM SIGCOMM, September 1997.

[13] V. Paxson, End-to-End Internet Packet Dynamics, IEEE/ACM Transactions on Networking 7 (3) (1999) 277–292, June.

[14] V. Jacobson, Congestion Avoidance and Control. in: Proceedings of ACM SIGCOMM, August 1988.

[15] P. Karn, MACA—A New Channel Access Method for Packet Radio, in: Proceedings of the ARRL/CRRL Amateur Radio Ninth Computer Network Conference, 1990.

[16] ns-2 simulator. http://www.isi.edu/nsnam/ns/.

[17] A. Bakre, B.R. Badrinath, I-TCP: Indirect TCP for Mobile Hosts, in: Proceedings of the 15th IEEE Int'l Conf. on Distributed Computing Systems, May 1995.

[18] C. Parsa, J.J. Garcia-Luna-Acceves, Improving TCP performance over wireless networks at the link layer, ACM Mobile Networks and Applications Journal 5 (1) (2000) 57–71.

[19] H. Balakrishnan, S. Seshan, R. Katz, Improving reliable transport and handoff performance in cellular wireless networks, ACM Wireless Networks 1 (4) (1995).

[20] K. Brown, S. Singh, M-TCP: TCP for Mobile Cellular Networks, ACM Computer Communications Review 2 (5) (1997) 19–43.

Edward Chan received his B.Sc. and M.Sc. in Electrical Engineering from Stanford University and the Ph.D. in Computer Science from Sunderland University. He is currently an Associate Professor in the Department of Computer Science at the City University of Hong Kong. His research interest includes power-aware and mobile computing as well as high speed computer networks.

S.W. Ng received his B.Sc. and M.Phil. in Computer Science from the City University of Hong Kong. His research interest includes network protocols and mobile computing.