Examining TCP Parallelization Related Methods for Various Packet Losses

Qiang Fu and Jadwiga Indulska

School of Information Technology and Electrical Engineering, The University of Queensland, Brisbane, QLD 4072, Australia
{qiangfu, jaga}@itee.uq.edu.au

Abstract. The diversity of the networks (wired/wireless) prefers a TCP solution robust across a wide range of networks rather than fine-tuned for a particular one at the cost of another. TCP parallelization uses multiple virtual TCP connections to transfer data for an application process and opens a way to improve TCP performance across a wide range of environments - high bandwidth-delay product (BDP), wireless as well as conventional networks. In particular, it can significantly benefit the emerging high-speed wireless networks. Despite its potential to work well over a wide range of networks, it is not fully understood how TCP parallelization performs when experiencing various packet losses in the heterogeneous environment. This paper examines the current TCP parallelization related methods under various packet losses and shows how to improve the performance of TCP parallelization.

1 Introduction

TCP parallelization uses a set of parallel TCP connections to transfer data for an application process and opens a way to improve TCP performance across a wide range of environments - high bandwidth-delay product (BDP) [1] [2] [11] [12], wireless [5] [6] as well as conventional networks [3] [4].

In [5], a brief test, which involved bit errors due to the purposely degraded RF performance of the earth station in satellite systems, showed that given the same effective window of 192Kbytes better throughput was achieved when a large number of connections were used. Unfortunately there was no further analysis. FEC is widely used for wireless data transmission. It is well known that a small amount of FEC gives the most efficient throughput gain (but not the best throughput) [6]. To fully utilize the available bandwidth, a large amount of FEC has to be added. This wastes a lot of bandwidth. Therefore, opening multiple TCP connections with a small amount of FEC could be an efficient way.

To eliminate the complexity of maintaining multiple connections, some solutions use a single TCP connection to emulate the behavior of a set of standard TCP connections. MulTCP [7] is a typical example, which makes one logical connection behave like a set of standard TCP connections to achieve weighted proportional fairness. Although the purpose of MulTCP is not to achieve high performance in high-BDP networks, its essence has been inherited by Scalable TCP [8], HighSpeed TCP [9] and FAST TCP [10] which are designed for high-BDP networks.

Most of the solutions mentioned above focus on wired networks, especially high-BDP networks and the solutions implemented in the wireless environments do not fully explore the potential of TCP parallelization. In this paper, we examine some of the solutions and explore the potential of TCP parallelization under various packet losses which reflect the heterogeneity of wired/wireless networks. We have a particular interest in MulTCP [7] and the fractional method in [11] [12]. The fractional method can be a representative of the solutions using parallel connections to improve effectiveness in utilizing while maintaining fairness. MulTCP can be a representative of using a single connection to emulate parallel connections.

2 Analytical Models and Simulation Environment

We have developed the bandwidth model shown in (1) based on a set of parallel TCP connections [13] [14]. W_n, n, p, MSS and RTT denote the aggregate window of a set of parallel connections, the number of parallel connections, packet loss rate, Maximum Segment Size and Round-Trip Time, respectively. c is referred to as window reduction factor. In response to a packet loss the window of the involved connection is cut down by $1/c$. m is referred to as window increase factor. The aggregate window (W_n) opens m packets per RTT. The simplicity of the model clearly shows the relationship between the key elements such as window increase factor (m) and window reduction factor (c) and how these elements impact the throughput and aggressiveness of the set of parallel connections.

$$BW_n = \frac{(1/p) * MSS}{(W_n / cmn) * RTT} = \sqrt{\frac{m(2cn-1)}{2p}} \times \frac{MSS}{RTT} \tag{1}$$

Without considering the impact of Fast Recovery and timeouts, the model in (1) cannot show if a single logical connection can achieve the equivalent throughput when emulating a set of parallel connections and if a set of multiple connections can give a better performance when emulating a single standard connection. In (2), a model is presented which takes TCP's Fast Recovery into account. More details about this model are available in [13].

$$\left[2\frac{(cn-1)W_n}{cn} + \frac{m(n_e-1)}{n_e} + \frac{W_n}{cn}\right] \times \left(\frac{W_n}{cmn} - \frac{n_e-1}{n_e}\right) \times \frac{1}{2} + \left[2\frac{(cn-1)W_n}{cn} + \frac{m(n_e-1)}{n_e}\right] \times 1 \times \frac{1}{2} = \frac{1}{p} \tag{2}$$

$$BW_n = \frac{(1/p) * MSS}{\left\{1 + \left(\frac{W_n}{cmn} - \frac{n_e-1}{n_e}\right)\right\} * RTT}$$

In (2), n_e represents the number of connections used to emulate a set of TCP connections. The model in (2) is more accurate than the one in (1) to predict the performance of a set of standard TCP connections [13]. Because the model does not consider the effects of timeouts, we do not expect that it can predict the performance well for a single connection emulating a set of standard TCP connections. Similar to fast recovery, in this case timeouts could significantly affect the throughput performance (compared to the direct use of a set of standard TCP connections). However, the model can show how the performance is affected in these two cases.

Table 1. Simulation parameters.

TCP Scheme	*Reno*
Link Capacity	28.6/8Mbps
Link Delay	100ms (5+90+5)
Packet Size (MSS)	1,000bytes
Buffer Size	Unlimited/Limited
Error (p)	Uniform/Two-state Markov (0.01/0.001)
Simulation Time	100/1,000s

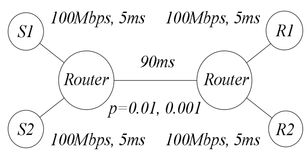


Fig. 1. Simulation topology

The simulations are run on the topology shown in Fig. 1. NS-2 is used to perform all the simulations. Table 1 is a concise list of the system parameters selected for the simulations. The studied connections are established between sending host *S1* and receiving host *R1*. The cross or competing traffic is created between sending host *S2* and receiving host *R2*. A buffer is used on the bottleneck link between the two routers. The buffer size is set unlimited for studying uniform/bursty losses and set limited when investigating buffer overflow losses.

3 Uniform Errors

3.1 Fractional Method

In [11] and [12], a *fractional approach* is introduced. In this scheme, each one of n flows increases its congestion window by one packet per n packets acknowledged and only one of n parallel streams will decrease its window by half in response to a packet loss. Referring to (1), we define Fractional Multiplier (FM) as: $FM=n/m$. If we take the n flows as a whole the approach requires the aggregate window to increase 1 packet per RTT, that is, m=1 and thus $FM=n$. This method does reduce the aggressiveness of the parallel connection by increasing the aggregate window 1 packet per RTT as a single standard TCP connection. However, it only reduces the aggregate window by 1/2n rather than 1/2 in response to a packet loss.

A *combined approach* is also introduced in [11] and [12]. The method combines a single standard TCP connection with a set of parallel connections that opens their window very conservatively. The parallel flows are modified based on the previously mentioned *fractional approach* with a little difference. Each of the n conservative flows increases its congestion window by one packet for every FM_C*n packets it receives. FM_C is referred to as combined fractional multiplier. The FM_C with a value >1 will ensure that the set of fractional streams is less aggressive than if they are modified by the *fractional approach*.

We carried out simulations based on these two approaches. In the simulations, the *fractional* and *combined approaches* compete with 5 standard TCP connections, respectively. We use 5 standard TCP connections as the cross traffic because this approach is also used in [11] and [12]. Fig. 2 and Fig. 3 compare the modeled and the simulation results. The model in (1) is used. For Fig. 2, the packet loss rate and the bottleneck bandwidth are set to 0.001 and 8Mb/s, respectively, to guarantee that the network is congested from time to time. For Fig. 3, the packet loss rate is set to 0.01 and the bottleneck bandwidth is 28.6Mb/s. The loss rate and bandwidth are high enough so that the bottleneck bandwidth cannot be fully utilized.

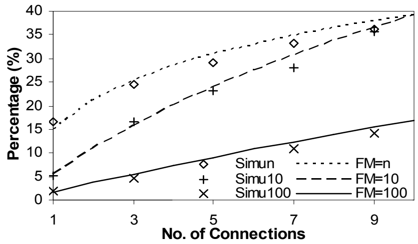


Fig. 2. Throughput share (p=0.001)

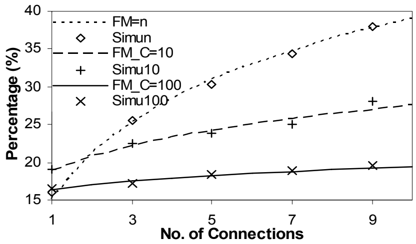


Fig. 3. Throughput share (p=0.01)

If all the schemes used in the figures had the same aggressiveness, then the studied connections would have a throughput share of 1/(1+5), that is, 16.67%. Fig. 2 shows the performance of the *fractional approach* with various *FM* schemes. Fig. 3 shows the throughput share achieved by the *fractional approach* and the *combined approach*. It shows the modeled results match the simulation results well. The good match indicates that the fairness for these two approaches is determined by their effectiveness to increase throughput because the model in (1) is actually a reflection of this effectiveness. These two approaches do not fundamentally address the concern how to maintain fairness in presence of congestion and improve effectiveness in absence of congestion.

3.2 Single Connection Emulating TCP Parallelization

MulTCP can be perceived as a single connection that emulates a set of standard TCP connections and is similar to the single connection based approach that we will examine later. For instance, if a single connection emulates a set of n standard TCP connections, then in (1) and (2) m=n and c=2n. The model in (1) cannot show the performance difference between a single connection emulating a set of parallel connections and the direct use of a set of parallel connections, because it ignores TCP's fast recovery. Therefore the model in (2) is used. In the figures, *n_emu_* denotes n_e in (2), which is the actual number of connections used to emulate the behavior of a set of parallel connections. In Fig. 4, the bottleneck bandwidth is set to 28.6Mb/s while the packet loss rate is set to 0.01. This makes the bottleneck bandwidth always underused. In Fig. 5, the packet loss rate is reduced to 0.001 to examine the fairness performance in the congested network. The figures examine the single connection based approach (*n_emu_*=1) and the 5 connection based approach (*n_emu_*=5) that emulate the behavior of n standard TCP connections. The cross traffic is made up by the same number (that is, n) of standard TCP connections. Therefore, for desired fairness the throughput share achieved by the studied approaches should be 50%. However, Fig. 4 shows that the single connection based approach fails to achieve the desired fairness. By using 5 connections, we find that the fairness is much improved. We notice the difference between the modeled and the simulation results. We attribute the difference to the timeouts ignored in the model. The similar performance is also observed in Fig. 5.

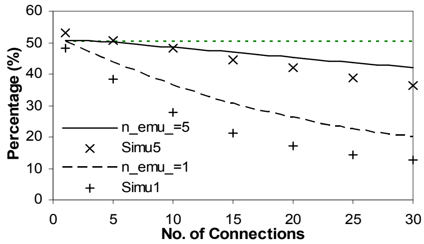


Fig. 4. Throughput share (p=0.01)

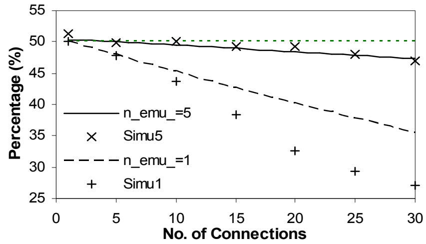


Fig. 5. Throughput share (p=0.001)

4 Bursty Errors

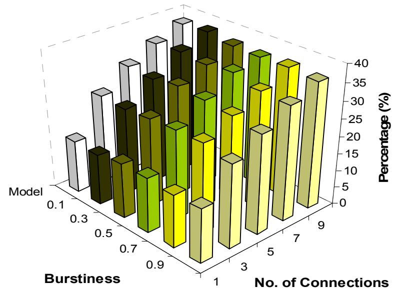
Depending on the bit interleaving depth and channel characteristics, the burstiness of packet losses in wireless links tends to vary. Serious bursty losses could cause synchronization of packet losses between the parallel connections. In this section we study the impacts of burstiness of packet losses on TCP parallelization. We use a discrete-time Markov chain with two states (Bad and Good) to model the error characteristics of the wireless link. Such a model is often used in the literature to analyze losses on wireless links [6] [15] [16] [17]. A packet is lost if the packet is transmitted over the link while the link is in the Bad state, otherwise it is supposed to be correctly received. Suppose that the link is currently in the Bad state for a given time unit. The burstiness is represented by the probability that the link stays in the Bad state for the next time unit. Note that the time unit can be replaced by a packet.

In this section, we set the bottleneck bandwidth to 8Mbps. The average probability of time unit being in the bad state, if not mentioned, is set to 0.01. The length of the time unit can destroy up to 4 back-to-back packets in the bad state. The burstiness of packet losses varies from 0.1 to 0.9. Please note that even if the burstiness is set to 0.1, the loss pattern is significantly different from the uniform losses. This is because one time unit in the bad state can destroy up to 4 back-to-back packets and the uniform distribution of the time units in the bad state requires a burstiness of 0.01.

4.1 Fractional Method

Because of the similarity of the *combined approach* and *fractional approach*, we only examine the performance of the *fractional approach*. Except for the packet loss pattern and bottleneck bandwidth, the simulation environment is same to the corresponding section for the uniform packet losses. In Fig. 6 the packet loss rate is set to 0.001 to study the performance when the network is congested while in Fig. 7 it is set to 0.01 to study the performance when the bottleneck bandwidth is underused.

The figures show the similar performance across various burstinesses and between the two different packet loss rates. Furthermore, the simulation results match well the modeled results. Because the simulation performance under the uniform losses matches well the modeled performance too, there is no clear difference between the throughput share performance under uniform and bursty losses.



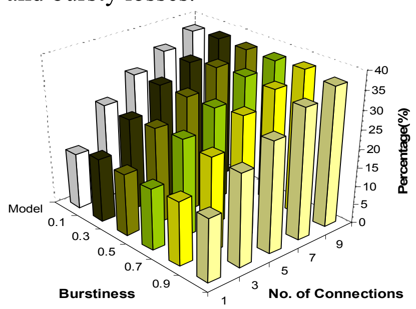
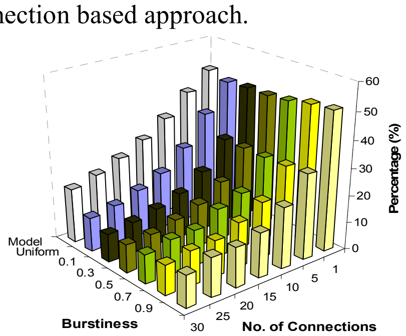


Fig. 6. Throughput share vs. Burstiness (0.001) **Fig. 7.** Throughput share vs. Burstiness (0.01)

4.2 Single Connection Emulating TCP Parallelization

Same to the corresponding section for the uniform losses, the single connection based approach and the 5 connection based approach are examined. The desired throughput share achieved by the studied approaches should be 50%. The setting of bottleneck bandwidth (8Mbps) and packet loss rate (0.01) enables the bottleneck to change from underused to congested as the emulated number of connections increases. Fig. 8 shows the throughput shares achieved by the single connection based approach are similar for the burstiness ranging from 0.1 to 0.9. The similar situation is observed for 5 connection based approach in Fig. 9. Similar to the observations under uniform losses, the 5 connection based approach can achieve a throughput share much closer to 50% than the single connection based approach. We also notice that the throughput share achieved by the studied approach under bursty losses is significantly and consistently lower than under uniform losses. This indicates that under bursty losses it is more difficult for the single connection based approach to achieve a performance equivalent to the direct use of a set of parallel connections. So is it for the 5 connection based approach.



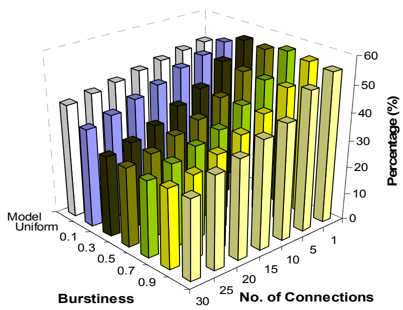


Fig. 8. Throughput share vs. Burstiness (1) **Fig. 9.** Throughput share vs. Burstiness (5)

5 Buffer Overflow Errors

In this section, we use a limited buffer on the bottleneck link between the two routers to study the effect of buffer size on the throughput performance.

5.1 Fractional Method

Only the *combined approach* is examined, because of its similarity to the *fractional approach*. The simulations are carried out for 100 and 1000 seconds. In the figures, the legend items with a suffix of "s" indicate the simulation time is 100s, otherwise it is 1000s. The buffer size is indicated by the number in the legend items and the unit is packet. For instance, B1000s means that the buffer size is 1000 packets and the simulation time is 100s. Fig. 10 shows dramatically different performance with the various buffer sizes and simulation times. In terms of simulation time, the throughput share achieved with the 100 second simulation is much lower than if the simulation time is 1000s. In terms of buffer size, a small buffer size gives the *combined approach* a better performance than a large buffer size. Recall that the set of parallel connections increases its aggregate window less than 1 packet per RTT. This loss of aggressiveness is compensated by the more aggressive window reduction factor: in response to a packet loss its aggregate window will only be closed by $1/2n$ (n denotes the number of parallel connections in the set). Let us assume that the buffer size is very large so that for a given time there are no packet losses. In this case, the set of parallel connections cannot take the advantage of its more aggressive window reduction factor and only the less aggressive window increase factor stands. As a result, the *combined approach* will give a throughput share less than the one predicted by the model.

As the buffer size decreases, packet losses occur increasingly often. This brings back the advantage of the aggressive window reduction factor that the set of parallel connections is supposed to enjoy. Therefore, if for a given simulation time the buffer size is small enough or for a given buffer size the simulation time is long enough so that there are regular packet losses, then the throughput share achieved by the *combined approach* is expected to match well the modeled results. However, it does not happen in the figure. The reason is that the analysis so far is limited to the assumption that all the packets have the equal probability of a packet loss and this assumption does not hold. By using the combined fractional multiplier (FM_C), the set of parallel connections is much less aggressive than the standard TCP connection and only opens its aggregate window by 1/10 packets per RTT. Therefore, the standard TCP connections create burstier traffic and are more susceptible to packet losses. When the simulation time is short the set of parallel connections has not yet fully enjoyed the advantage of the lower loss rate before the simulation run is completed, because of its too conservative window increase factor. When the simulation run is finished, its aggregate window has not yet been able to reach its full (maximum) size. As the simulation time gets longer, its aggregate window becomes larger and yields higher throughput before a packet loss occurs. This is why the throughput share with the simulation time of 1000s is much higher than the one achieved when the simulation time is 100s. As the set of parallel connections has a lower packet loss rate, it is understandable that the throughput share achieved by the *combined approach* is higher than the modeled throughput share if the buffer size is small enough or the simulation time is long enough to guarantee regular packet losses.

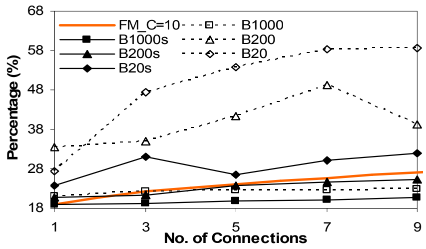


Fig. 10. Throughput share

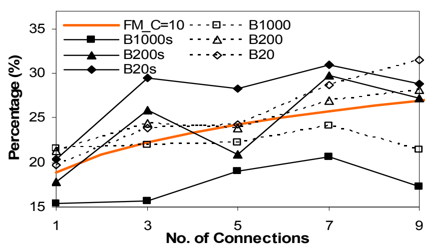


Fig. 11. Throughput share with UDP traffic

The traffic in Fig. 10 is self-similar traffic, because all the traffic is created by competing TCP flows. In Fig. 11, we introduce UDP traffic, which is created by a single UDP ON/OFF source (exponential). The sending rate is 1.6Mbps when the source is ON and thus takes up 20% bandwidth. Both the ON and OFF periods are set to 250ms. It shows that with the simulation time of 100s the performance of the *combined approach* is not stable and is similar to the performance in Fig. 10. However, when the simulation time is 1000s, the performance converges towards the curve predicted by the model. This means that the UDP traffic changes the packet loss pattern of the competing flows. The UDP traffic has a pacing effect on the TCP traffic and thus closes the difference of packet loss rate between the standard TCP flows and the set of parallel connections. As a result, the throughput share achieved by the *combined approach* is close to the modeled throughput share.

5.2 Single Connection Emulating TCP Parallelization

In contrast to the fractional method, the single connection based approach is more aggressive than the standard TCP connections. As a result, the finding is opposite to the one observed for the fractional method. Fig. 12 shows the performance of the single connection based approach with and without UDP traffic. The legend items with a suffix of "b" indicates that the UDP traffic mentioned in the previous section is introduced. It shows that high throughput share is achieved by the single connection based approach with large buffer size because large buffer size means lower packet loss rate. Furthermore, the UDP traffic makes the throughput share higher than if no UDP traffic is introduced. Although the single connection based approach has the same level of aggressiveness as the total competing TCP flows, it creates more back-to-back packets for the same number of returning ACKs. Therefore its traffic is burstier and thus more likely to cause buffer overflow (packet losses). Once again, the UDP traffic closes the difference of packet loss rate between the competing TCP flows, and thus, the throughput share achieved by the single connection based approach with UDP traffic is higher than without UDP traffic. With UDP traffic introduced, Fig. 13 compares the single connection based approach with the 4 connection based approach. The legend items with a suffix of "_1" denote the single connection based approach. Besides the advantage of the use of a small number of connections to emulate a set of standard TCP connections shown in the previous sections under random packet losses, the 4 connection based approach can reduce traffic burstiness.

The 4 connection based approach can not only improve its throughput share, but also the efficiency to utilize the bottleneck bandwidth. Fig. 14 shows the total throughput achieved by the studied approach along with its competing TCP flows. For dark color columns the 4 connection based approach is the studied approach while the light color columns on the immediate right indicate the performance of the corresponding single connection based approach. For example, 4/20 indicates that the 4 connection based approach is performed with a buffer size of 20 packets while 1/20 indicates the performance of its corresponding single connection based approach. It clearly shows that the 4 connection based approach always has a better utilization of the bottleneck bandwidth than the single connection based approach. The smaller the buffer size is, the larger the improvement gain on the utilization is.

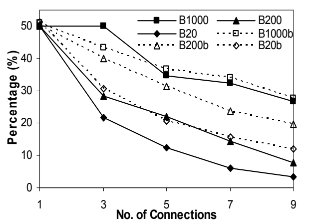


Fig. 12. Throughput share: UDP traffic vs. no UDP traffic

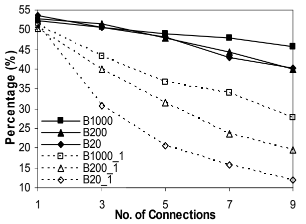


Fig. 13. Throughput share with UDP traffic

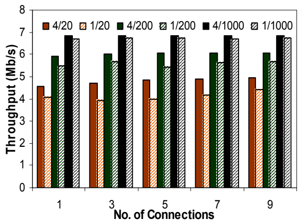


Fig. 14. Utilization of bottleneck bandwidth

6 Conclusions

In this paper, we analyzed TCP parallelization related methods for various packet losses. For the fractional method, the analysis shows that our model is capable to predict the performance of the method under uniform, bursty and buffer overflow losses. It shows that the fractional method does not really address the concern how to maintain fairness in presence of congestion and improve effectiveness in absence of congestion: the fairness and the effectiveness are achieved at the cost of each other. For a given context our model can help determine the balance point, where the desired effectiveness/fairness can be achieved at an acceptable cost of one another. We also examined how buffer size and competing traffic affect the performance of the fractional method. It shows that the performance can be different over various buffer sizes. The achieved fairness and effectiveness can be deviated significantly from the expected level. In [11] [12], the *combined approach* was considered more suitable than *fractional approach*. Our model and analysis do not show that there is a mechanism in favor of the *combined approach*.

For the single connection based approach represented by MulTCP [7], the analysis shows that the single connection based approach cannot achieve the equivalent performance by the direct use of the set of parallel connections. The model, which takes fast recovery into account, clearly shows how the performance is affected by using the single connection based approach. In high speed and error-prone networks, it is too risky to use this method. It can cause tremendous performance penalty due to unnecessary congestion control and timeouts. With the same level of aggressiveness, the single connection based approach creates burstier traffic than TCP parallelization, and thus is more prone to buffer overflow losses. The model shows that using a small number of connections to emulate a set of parallel TCP connections can achieve a performance close to the performance of the set of parallel connections. This method has the advantages of TCP parallelization while reducing the complexity of managing a large number of connections. The method can reduce the burstiness of its traffic and thus performance degradation by buffer overflow losses. Compared to the single connection based approach, it can not only improve the throughput performance, but also the efficiency to utilize the bottleneck bandwidth.

References

1. B. Allcock, ed al., "Data Management and Transfer in High-Performance Computational Grid Environments", *Parallel Computing*, 28(5), 2002.
2. R. Grossman, ed al., "Experimental Studies Using Photonic Data Services at IGrid 2002", *Future Computer Systems*, 19(6), 2003.
3. H. Balakrishnan, ed al., "An Integrated Congestion Management Architecture for Internet Hosts", *ACM SIGCOMM*, Sept. 1999.
4. L. Eggert, J. Heidemann, and J. Touch, "Effects of Ensemble–TCP", *ACM Computer Communication Review*, 30(1), 2000.
5. M. Allman, H. Kruse and S. Ostermann, "An Application-Level Solution to TCP's Satellite Inefficiencies", *WOSBIS*, Nov. 1996.
6. C. Barakat, E. Altman, "Bandwidth Tradeoff between TCP and Link-Level FEC", *Computer Networks*, 39(2), June 2002.
7. J. Crowcroft and P. Oechslin. "Differentiated end-to-end Internet services using a weighted proportionally fair sharing TCP", *Computer Comm. Review*, 28(3), 1998.
8. Tom Kelly, "Scalable TCP: Improving performance in highspeed wide area networks", *Computer Communication Review* 32(2), April 2003.
9. Sally Floyd, "HighSpeed TCP for Large Congestion Windows", *RFC 3649*, Dec. 2003.
10. C. Jin, D. Wei and S. Low, "FAST TCP: motivation, architecture, algorithms, performance", *INFOCOM*, 2004.
11. T. Hacker, B. Noble and B. Athey, "The Effects of Systemic Packet Loss on Aggregate TCP Flows", *Supercomputing*, 2002
12. T. Hacker, B. Noble, B. Athey, "Improving Throughput and Maintaining Fairness using Parallel TCP", *INFOCOM*, 2004.
13. Q. Fu, J. Indulska, "The Impact of Fast Recovery on Parallel TCP connections", *HET-NETs* 2004, Ilkley, UK.
14. Q. Fu, J. Indulska, "Features of Parallel TCP with Emphasis on Congestion Avoidance in Heterogeneous Networks", *Advanced Wired and Wireless Networks*, pp. 205-228, eds. T. Wysocki, A. Dadej and B. Wysocki, Springer-Verlag 2004.
15. H. Chaskar, T. V. Lakshman, and U. Madhow, "On the Design of Interfaces for TCP/IP over Wireless", *MILCOM*, 1996.
16. A. Chockalingam, M. Zorzi, and R.R. Rao, "Performance of TCP on Wireless Fading Links with Memory", *ICC*, 1998.
17. E.N. Gilbert, "Capacity of a Burst-Noise Channel", Bell Sys. Tech. Journal, Sept. 1960.