INDIAN INSTITUTE OF MANAGEMENT CALCUTTA

WORKING PAPER SERIES

WPS No. 649/ January 2010

A New Protocol to improve TCP Performance in Network Mobility*

by

Bhaskar Sardar
Department of Information Technology, Jadavpur University, Kolkata, India

Debashis Saha
Professor, IIM Calcutta, Diamond Harbour Road, Joka P.O., Kolkata 700 104 India

&

Mahbub Hassan
School of Computer Science & Engg, University of New South Wales, Sydney, Australia

* A part of this paper was presented in IEEE Vehicular Technology Conference 2006

A New Protocol to improve TCP Performance in Network Mobility *

Bhaskar Sardar
Department of Information Technology
Jadavpur University
Kolkata, India

* A part of this paper was presented in IEEE Vehicular Technology Conference 2006

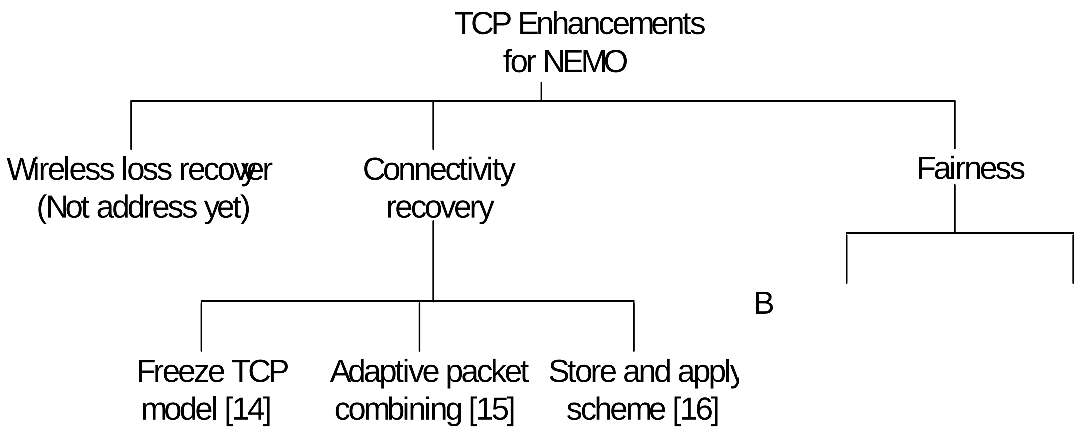
Figure 1: NEMO Connectivity Model

the sender timeout that leads to the retransmission at the FH when the retransmission is being performed on the wireless link. Thus, the protocol requires small RTT in the wireless link to allow multiple local retransmissions.

As a result, several proposals ([11]-[13]) have been made in the literature to improve the performance of snoop protocol. But, those proposals are made only for terminal mobility. So. We do not discuss them here.

B. TCP enhancement schemes for NEMO

As of today, there exist only four proposals to enhance TCP performance in NEMO as shown in Figure 2. We categorize the protocols in three groups: wireless loss recovery, connectivity recovery, and fairness. Most of the proposed protocols try to adapt TCP behavior after a handoff. There are three proposals in connectivity recovery category: Freeze TCP model [14], Adaptive packet combining (APC) [15], Store and apply scheme [16]. There is only one proposal in fairness category: MR based fairness control scheme [17]. In general, the proposals in connectivity recovery category try to adapt TCP behavior after a handoff takes place. Freeze TCP eliminates the negative impact of handoff when the handoff takes place between similar networks. On the other hand, APC, store and apply schemes improves TCP performance when vertical handoff takes place. The MR based fairness control scheme guarantees fair share of available bandwidth to the MHs in NEMO. From Figure 2, we find that no attempt has been made to deal with the negative impact caused by dual wireless link of NEMO. Also, the fairness issue in co-existence of terminal mobility and NEMO is not studied yet.


TCP Enhancements
for NEMO
Wireless loss recovyer
(Not address yet)
Connectivity
recovery
Fairness
B
Freeze TCP
model [14]
Adaptive packet
combining [15]
Store and apply
scheme [16]

connected to the wired infrastructure. With existing agents, any packet loss in this part of the path, whether they occur in the BS-MR link, or in the MR-MH link, will have to be detected and retransmitted by the agent located at the BS. So, although the existing agents are able to detect wireless loss, they are unable to locate the origin of wireless losses. As a result, the existing agents may take long time to detect and recover wireless losses. So, the existing agents may not provide optimum performance in NEMO. Investigating the impact of this additional wireless link on the performance of the widely used TCP protocol, and designing mechanisms to alleviate any negative impacts may solve the problem to some extent. The objective of this paper is to extend the single-point recovery mechanism to multipoint i.e., link-to-link recovery mechanism. In this case, the wireless losses in different wireless links could be recovered independently and simultaneously, thereby decreasing the loss recovery time.

Also, since NEMO is likely to co-exist

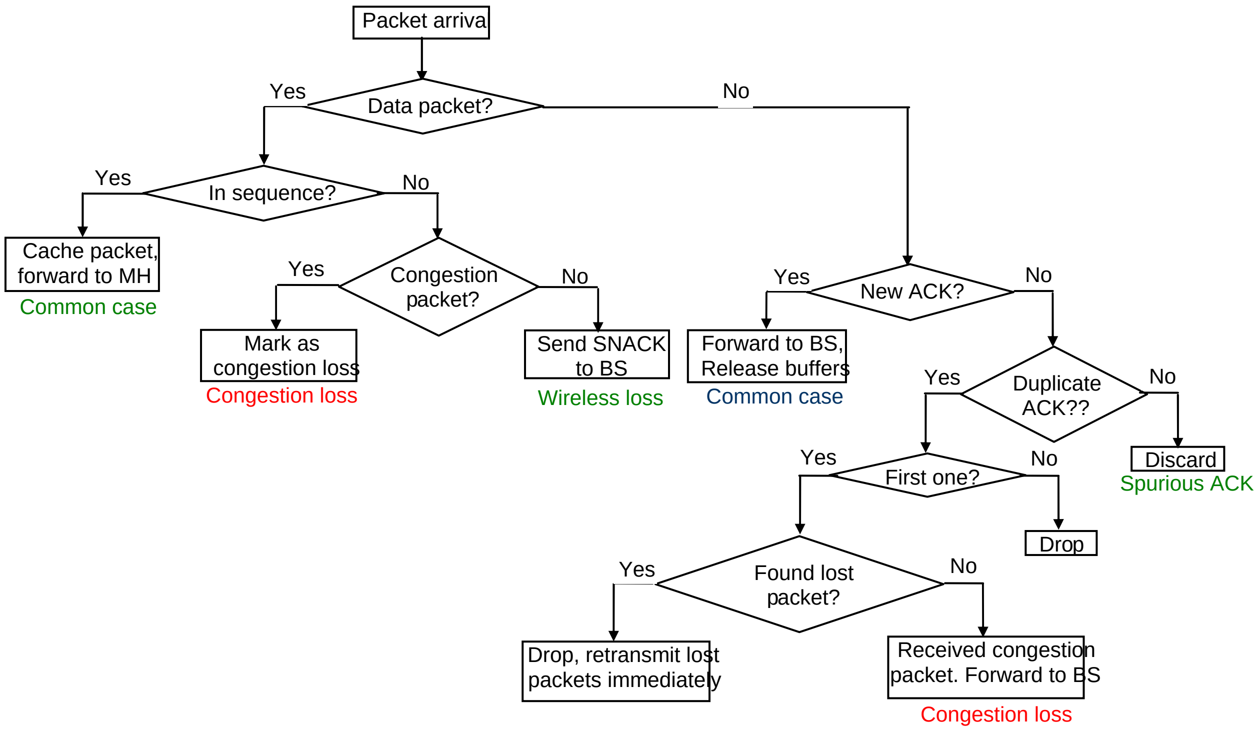


Figure 4: obTCP agent at MR

B. obTCP agent at MR

obTCP agent at MR has four main functions: i) caching TCP packets received from BS, ii) dropping DUPACKs, iii) detecting and reporting packet corruption to the BS, iv) retransmitting packets those are lost in the MR-MH wireless link. If the obTCP agent finds a gap in sequence number of the received packets, it generates a SNACK specifying all the packets those might have been lost in the wireless link and forwards to the BS. If the packets reach the MR in sequence, the obTCP agent stores them in the cache and forwards to the MH. The reason behind caching at MR is that the MHs may be connected to the MR via wireless links. When the packets reach the receiver out of order, MH generates DUPACKs. There can be three reasons for which the MH generates these DUPACKs: the packets might have been lost in the path between MR and MH or in the path between BS and MR, or in the wired network between FH and BS. When DUPACKs reach the MR, the obTCP agent checks its cache. If the packet is found it is retransmitted. Otherwise, it has definitely received an indication (congestion packet) from BS about this packet. If the packet has been lost in wired network, it will get an indication from the BS. In this case, the obTCP agent at MR will not suppress these DUPACKs in order to initiate fast retransmission at the FH.

IV. Comparison of Loss Recovery in snoop and obTCP

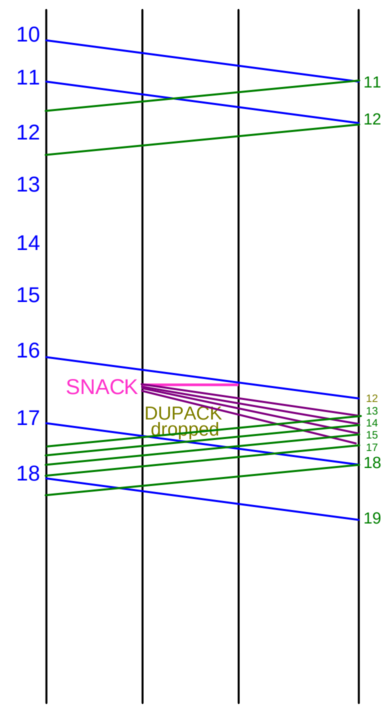


Figure 5: snoop

Figure 6: obTCP

Figure 5 and 6 show an example of link layer mechanisms and point out that these mechanisms must be used very carefully. Assume that packets up to sequence number 11 have been transmitted successfully and that packets 12, 13, 14 and 15 have been dropped in the BS-MR wireless link. Figure 5 depicts how wireless losses are recovered using snoop agents. In this case, the snoop agent treats two wireless links, BS-MR and MR-MH, as a single link BS-MH. When the MH receives packet 16, the MH sends a DUPACK for packet 12. This DUPACK, when received by the snoop agent at BS, makes it retransmit packet 12 from its cache and drop the DUPACK for 12. When the MH receives packet 12 it generates ACK 13. Packet 17 generates DUPACK for packet 13 and snoop agent at BS also drops this. This process continues until all the lost packets are successfully recovered. Therefore, snoop can recover from packet losses in any wireless link but only one packet per RTT over BS-MH wireless link. So, even if snoop is quit effective in dealing with wireless losses, it takes longer time to detect and recover the lost packets as can be seen from the timing diagram. When the packets are lost in MR-MH link, operation of snoop remains same.

In order to rectify the problem of unnecessarily waiting longer for the DUPACK at BS, obTCP includes MR in its design by placing an obTCP agent in MR. In this case, the path from BS to MH consists of two segments: one wireless link between BS and MR, another wireless link between MR and MH. Packet losses in each wireless link are handled separately. So, the wireless losses can be detected at an earlier time than snoop. This is explained in Figure 56 where it can be observed that after receiving the first out-of-order packet at MR, the obTCP agent at MR sends an SNACK packet to BS causing retransmission of all missing packets locally at once, in a lot shorter RTT than if the of packet 16 at MR. On reception of this SNACK packet; the obTCP agent at BS retransmits the requested packets. By using the MR, obTCP helps in reducing the loss recovery time and also enables retransmission of multiple packets in one local (and considerably shorter) RTT thus maintaining a good flow of packets. Note that one could use snoop to recover from multiple losses by introducing the SNACK mechanism at both the BS and the MH. However, that would require changes in the installed base making the deployment of snoop more difficult*. On the other hand, obTCP does not require any modification to the existing TCP implementations, yet is capable of exploiting the SNACK mechanism for recovering from multiple losses.

V. Analysis of Loss Recovery Time

Conventional TCP adjusts its congestion window size (w) according to two algorithms, namely slow start and congestion avoidance, where w is inversely proportional to RTT and square root of loss probability (p) [19]. Given that TCP throughput is directly proportional to the window size, we have:

$$w \propto \frac{1}{RTT} \text{ and } w \propto \frac{1}{\sqrt{p}} \qquad (1)$$

$$\text{Throughput} \propto w \qquad (2)$$

As a result, when either delay or loss probability increases, TCP throughput deteriorates significantly. To overcome this, obTCP attempts to reduce the effect of high loss probability in wireless links by quickly recovering from the wireless losses thereby

---

*

keeping RTT of the connection as low as possible. Hence, the benefit of obTCP depends on loss recovery time

Let us assume that the loss probability in BS-MR and MR-MH links are p_1 and p_2, respectively, and delay in BS-MR and MR-MH links are d_1 and d_2, respectively. In the following subsections, we model the loss recovery time and analyze the effectiveness of link-link loss recovery mechanism of obTCP. For easy reference, Table 1 lists the variables used in this paper.

A. Modeling the Loss Recovery Time

We know that the recovery process for snoop takes place only at the BS. Therefore, wherever the packet is lost (either BS-MR or MR-MH wireless link), the retransmissions happen from BS only. Hence, the effective loss probability for snoop (p_s) is

$$p_s \quad 1 \quad 1 \quad p_1 \quad 1 \quad p_2 \qquad (3)$$

Table 1: List of Notations

Notations	Meaning
d_1, d_2	One way delay in BS-MR and MR-MH wireless link respectively
p_1, p_2	Loss probability in BS-MR and MR-MH wireless link respectively

It is easy to see that, for snoop, the total number of transmissions (N) required to successfully receive an ACK at BS is given by:

$$N = \frac{1}{1 - p_s} \qquad (4)$$

Hence, the recovery time for snoop is given by:

$$R_S = 2*(d_1 + d_2)*\frac{1}{1 - p_s} \qquad (5)$$

As the packet losses in different wireless links are handled separately in obTCP, we consider them as independent events. TDinoop c.1641 0 TD .0003 Tc28289 Tw ence,(ck)6.tal probability in BS-MR and MR-MH link and (ii) linearly with delay in BS-MR and MR-MH link.

As the errors in the wireless links are independent, losses may occur in both links or in any one link. If the losses take place in BS-MR link only, then Equation (7) can be rewritten as:

$$R_{gain} \quad 2 * d_2 * \frac{p_1}{1 \quad p_1} \tag{8}$$

If losses occur in MR-MH link only, then Equation (7) can be rewritten as:

$$R_{gain} \quad 2 * d_1 * \frac{p_2}{1 \quad p_2} \tag{9}$$

The observations from Equations (8) and (9) can be summarized as:

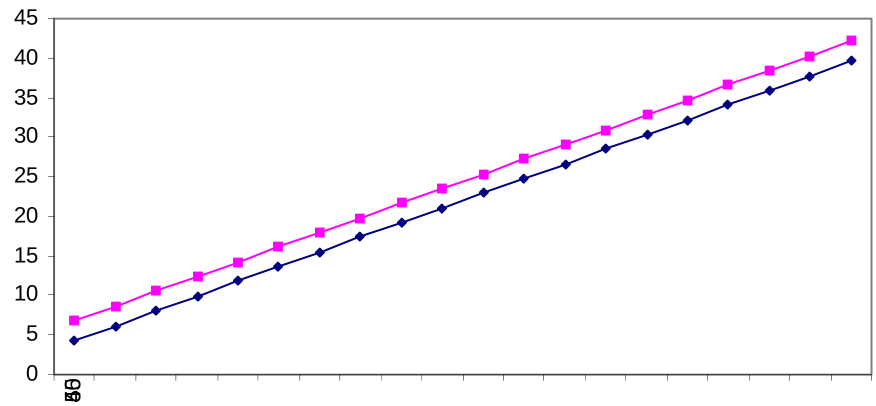
1. Gain in Recovery time increases exponentially with loss probability of the erroneous link.
2. Gain in Recovery time increases linearly with delay on the error free link.
3. Gain in Recovery time is constant with delay in the erroneous link.

Now, we establish the relationship between window size, throughput and loss recovery time. It is obvious that RTT of a TCP connection is directly proportional to loss recovery

From Equation (11), it is clear that window size is larger for obTCP, which results in higher throughput for obTCP. This is due to the fact that recovery time in snoop is higher than obTCP. Note that throughput improvement is linearly related with delay and exponentially with loss probabilities as in the case of gain in loss recovery time.

B. Numerical Analysis

We first examine the case when losses are present on both links. Figure 7 and Figure 8 show the performance gain of obTCP over snoop in terms of gain in recovery time. For Figure 7, we use p_1=0.15, p_2=0.05, and, for Figure 8, we use d_1=20ms, and d_2=20ms. In Figure 7, we plot R_{gain} as a function of delay in MR-MH link d_2. From Figure 7, it can be seen that gain increases linearly with delay in both links. In Figure 8, we plot gain in recovery time R_{gain} as a function of loss probability in BS-MR link p_1. From Figure 8, it is evident that gain increases exponentially with loss probability in BS-MR and MR-MH link. Hence, with small increase in loss probability in BS-MR and MR-MH link, obTCP can achieve significantly higher performance than snoop.



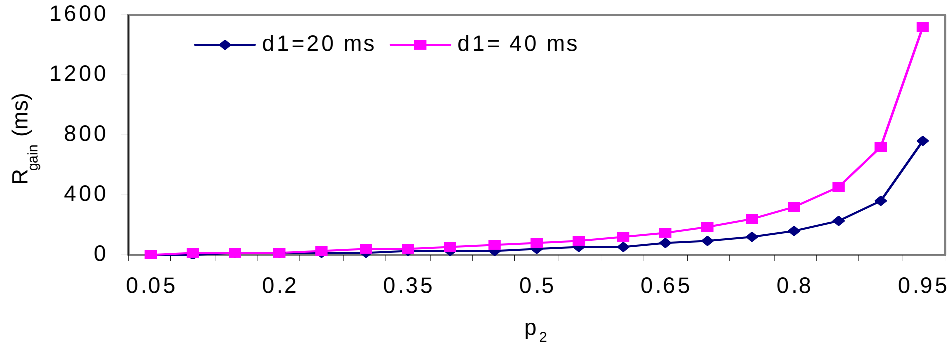
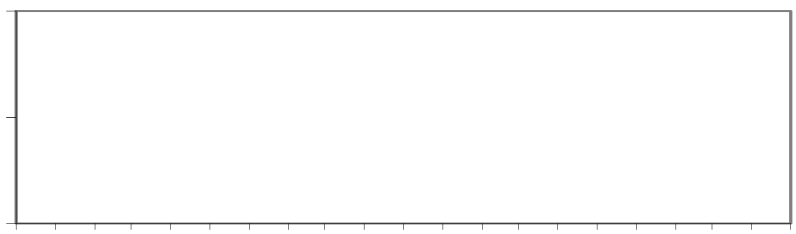


Figure 10: Effect of losses in MR-MHlink when BS-MR link is error free



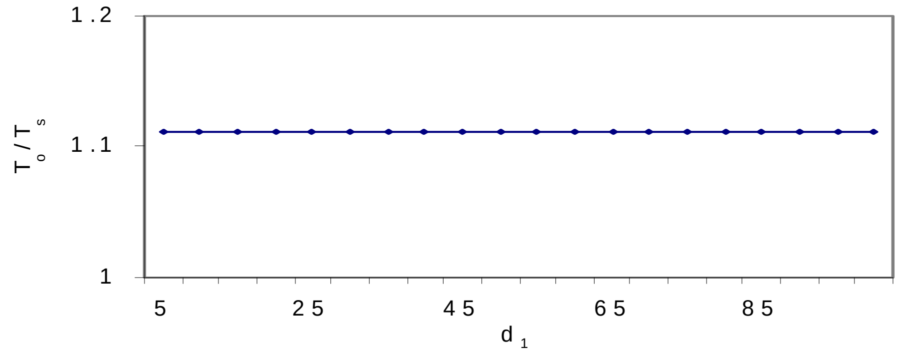
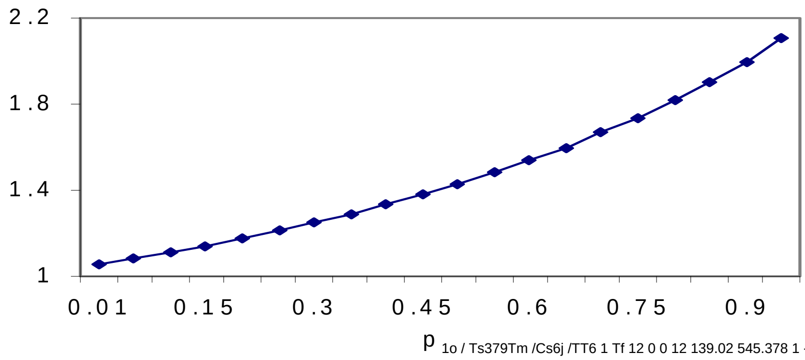


Figure 12: Variation of throughput ratio for delay in BS-MR link



1o / Ts379Tm /Cs6j /TT6 1 Tf 12 0 0 12 139.02 545.378 1 -.0001 Tc -.0011 Tw (Figure 12: V3riation of throughput)Tj 15

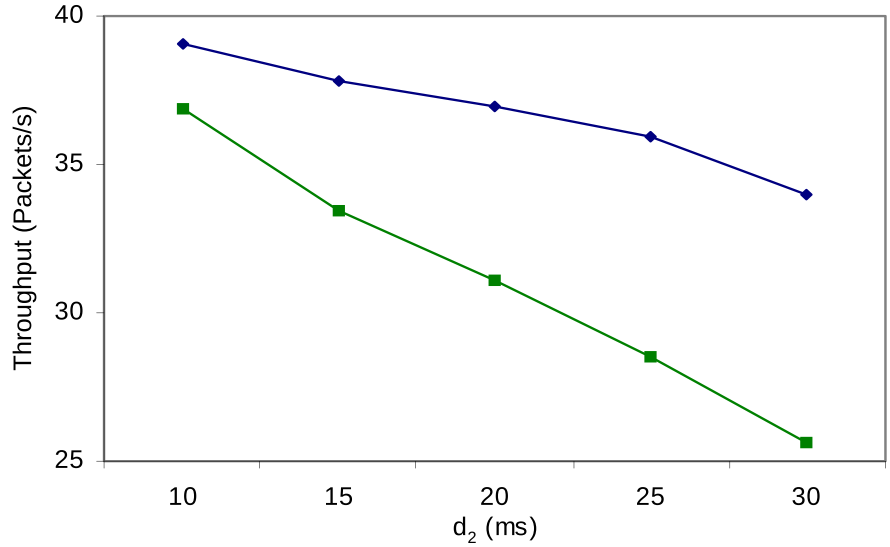
channel conditions are good (low delay variation, negligible loss probability), both snoop and obTCP performs similarly with negligible gain in throughput of obTCP. For the sake of clarity of presentation, we present results for three cases only: losses occur in BS-MR link only, wireless links are identical, i.e. loss probability in both links are same, and effect of delay in erroneous link. The results presented in this paper are taken from 2000 sec run of the simulation.

Table 2: TCP Parameters

Parameter	Value
TCP version	Reno
Packet size	1000 Bytes
Initial congestion window	2 packets
Maximum congestion window	16 packets
Initial slow start threshold	10 packets

A. Effect of losses in BS-MR wireless link

Figure 14 and Figure 15 show the throughput performance of obTCP and snoop for d_1=10ms and p_2=0.0001. For Figure 14, we use p_1=0.1, and, for Figure 15, we use p_1=0.2. It is interesting to note that performance of snoop degrades more sharply than obTCP with increasing delay in MR-MH link, which indicates that throughput gain increases with increase in delay of MR-MH link.

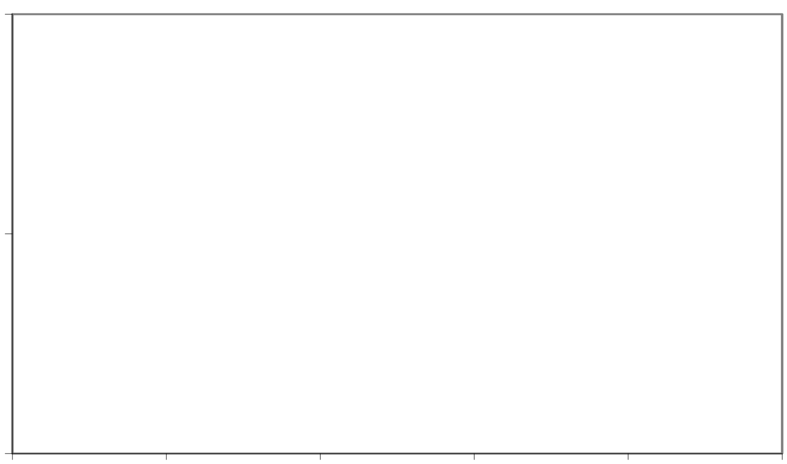


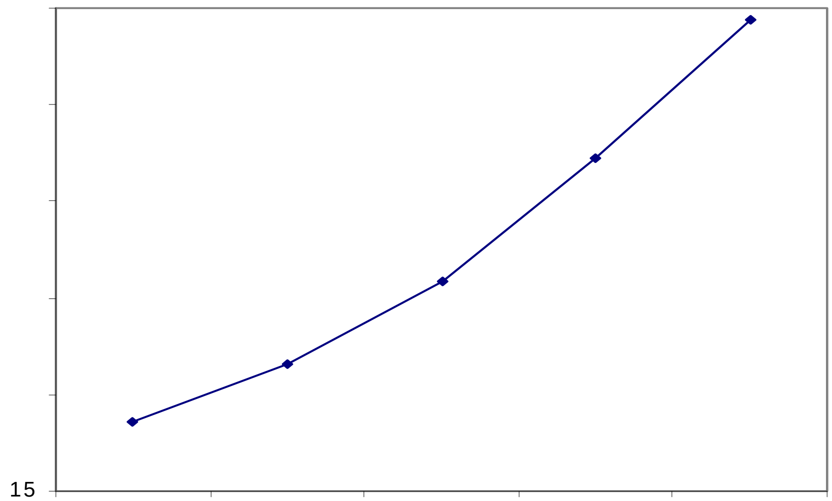
40
35
30
25
Throughput (Packets/s)
10
15
20
25
30
d2 (ms)

snoop for p_1=0.1 and p_1=0.2 respectively. It can be seen from Figure 16 that with losses in BS-MR link, throughput gain increases linearly with delay in MR-MH link.

We see the exponential increase of throughput improvement from Figure 17 for d_1=20ms, d_2=20ms and negligible loss probability p_2=0.0001. obTCP achieved an improvement of over 39% over snoop.

The higher performance of obTCP over snoop can be explained as follows: When packets are lost in BS-MR wireless link, the obTCP agent at MR detects the loss immediately and requests BS obTCP agent to retransmit the lost packets. So, the recovery mechanism has immediate reaction. But, in snoop, it has to wait for RTT over MR-MH wireless link to even detect the loss. Also, as SNACK mechanism is used over BS-MR wireless link, multiple packet losses are recovered in one RTT over BS-MR wireless link. But, in snoop, only one lost packet is recovered in one RTT over BS-MR and MR-MH wireless link.




15

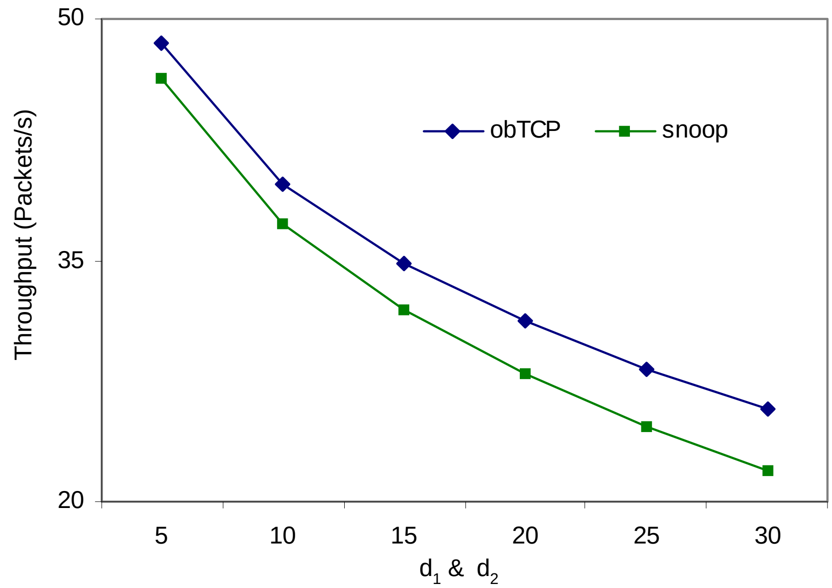


Figure 18: Throughput performance for identical links

C. Effect of Delay in erroneous link

In Section V, through numerical analysis, we have shown that gain in recovery time is independent of the delay in erroneous links. We conduct several simulations to verify our claims. Results are shown in Figure 20-Figure 21. For Figure 20 we use p_1=0.1, p_2=0.0, d_2=20ms, and for Figure 21 we use p_1=0.0, p_2=0.1, d_1=20ms. Figures show that our claim matches for simulation experiments too. In this case, when the delay is increased in erroneous links, both protocols are affected by this increased delay. So, the performance gain depends only on the delay on the error free link, which is kept constant. Hence, the performance gain becomes constant and the absolute value of gain depends on the delay of the error free link.

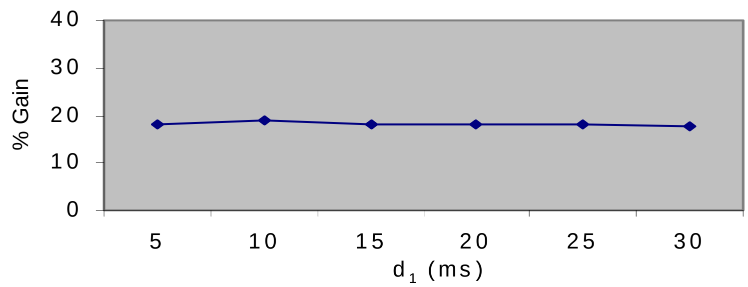


Figure 20: Constant gain for delay in erroneous link BS-MR

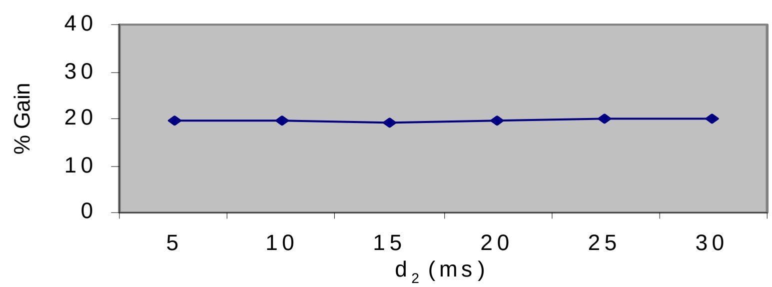


Figure 21: Constant gain fordelay in erroneous link MR-MH

F. RTT seen by TCP Reno

In this section, we describe some notable reasons for which obTCP achieved better performance than snoop in all simulations described above. We study how quicker recovery from wireless losses helps TCP to keep RTT of the connection as low as possible, which, in turn, helps in fast growth of congestion window. We use d_1=10ms, d_2=10ms, p_1=0.1 and p_2=0.0001. Figure 22 shows the Smoothed RTT (SRTT) of the connection measured between t_1=1000s and t_2=1050s of the simulation. We see that the RTT of the connection is low most of the time for obTCP, which helps in faster growth of TCP congestion window. For example, consider that a packet is lost in MR-MH wireless link. In case of snoop, the loss detection and its possible retransmission will take one RTT spanning BS-MR and MR-MH links. However, in case of obTCP, the agent at MR detects the loss and retransmits the lost packet quickly, which takes one RTT spanning MR-MH link only. As a result, the ACKs for the lost and recovered packets reach the sender much faster, which results in release of new packets faster in obTCP than snoop. Hence, obTCP achieved better performance than snoop.

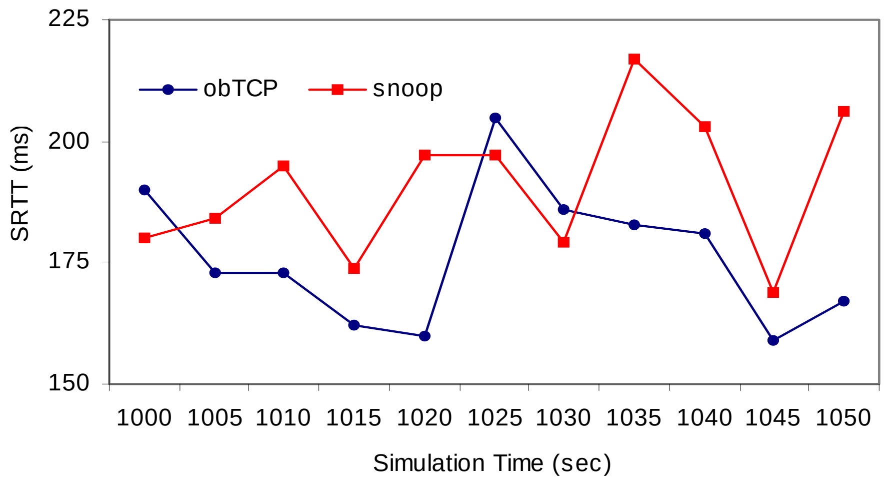


Figure 22: Comparison of smoothed RTT

VII. Throughput Models

We now develop the throughput models for snoop and obTCP in NEMO. We take a similar approach presented in [23], i.e. the window behavior is modeled in terms of rounds. To derive the duration of each round, we use the loss recovery time analysis presented in Section V.

A. Snoop

$$L_t \quad W' * p_s \qquad (13)$$

where p_s is given by Equation (3).

Each lost packet is recovered separately. So, the total recovery time $L_t * R_s$, where R_s is the loss recovery time for each packet and is given by Equation (5). Hence, the end-to-end RTT seen by TCP Reno (t) is given as:

$$t \quad 2d_0 \quad 2d_1 \quad 2d_2 \quad L_t * R_s \qquad (14)$$

So, the mean duration of X rounds, $E[A]$, is:

$$E[A] \quad X * t \qquad (15)$$

The end-to-end throughput, for snoop, can be derived as in [23]:

$$_s \quad \frac{E[Y]}{E[A]} \quad \frac{W'}{2d_0 \quad 2d_1 \quad 2d_2 \quad L_t * R_s} \quad \frac{W'}{2d_0 \quad 2d_1 \quad 2d_2 \quad W' * p_s * R_s} \qquad (16)$$

B. ObTCP

To derive throughput expression for obTCP we take similar approach as presented for snoop. The window evolution is the same as shown in Figure 23. The mean number of packets transmitted is given by Equation (12). In a particular round, total number of packet losses in BS-MR link (L_{t1}) is:

$$L_{t1} \quad W' * p_1 \qquad (17)$$

Let us assume that n_1 packets are recovered per NACK packet. So, the number of

Similarly, we can obtain the loss recovery time in MR-MH link. The total loss recovery time in MR-MH link is $\frac{L_{t2}}{n_2} * \frac{2d_2}{1 \quad p_2}$, where L_{t2} is the total number of packet losses in MR-MH link, and n_2 is the number of lost packets retransmitted. We note that retransmission of packets from MR follows go-back-n ARQ technique.

Hence, the end-to-end RTT, seen by TCP Reno is given as:

$$t \quad 2d_0 \quad 2d_1 \quad 2d_2 \quad \frac{L_{t1}}{n_1} * \frac{2d_1}{1 \quad p_1} \quad \frac{L_{t2}}{n_2} * \frac{2d_2}{1 \quad p_2} \qquad (18)$$

The mean duration of X rounds, $E[A]$, is:

$$E[A] \quad X * t \qquad (19)$$

where t is given by Equation (18).

Finally, the end-to-end throughput is given by modifying Equation (16) as follows:

$$_o \quad \frac{W'}{2d_0 \quad 2d_1 \quad 2d_2 \quad \frac{L_{t1}}{n1} * \frac{2d_1}{1 \quad p_1} \quad \frac{L_{t2}}{n_2} * \frac{2d_2}{1 \quad p_2}} \qquad (20)$$

C. Model Validation

In this Section, we describe numerical results for the throughput model of snoop and obTCP (Equation (16) and Equation (20)). We also provide simulation result for obTCP to validate the proposed models. The wired network is assumed to be error free. The delay in wired network from FH to BS is 50 ms, in wireless link between BS and MR is 20 ms, in wireless link between MR and MH is 10 ms. We use a fixed loss probability 0.1% in MR-MH link and vary the loss probability in BS-MR link from 0.1% to 10%. Table 3 shows the TCP parameters used in our simulations. To validate the proposed models with simulation results, we assume 5 ms MAC delay at BS and MR for every packet transmission. The results are shown in Figure 24.

Table 3: TCP parameters for throughput model validation

Parameter	Value
TCP Version	TCP Reno
Packet Size	1024 Bytes
Initial Congestion Window	2 Packets
Maximum Congestion Window	16 Packets
Initial Slow Start Threshold	12 Packets

use the fairness index function of [24] to quantify the fairness between obTCP and snoop. The fairness index function is expressed as:

$$\frac{\left(\sum_{i=1}^{n} {}_{i}\right)^{2}}{n \sum_{i=1}^{n} {}_{i}^{2}} \qquad (21)$$

where n is the number of flows (i.e., sources in the network) through the bottleneck link, and ${}_i$ is the fraction of the bottleneck link bandwidth obtained by flow i. The value of fairness obtained through this method ranges from (1/n) (i.e., extremely unfair) to 1 (perfectly fair), with 1 indicating equal allocation to all sources.

Link utilization

The network topology is shown in Figure 25. For each link the tuple (b*,d*) indicates the bandwidth and delay of that link. The bandwidth and delay values are summarized in

erroneous link. Second, throughput improvement increases linearly with delay on the error free link. Finally, throughput improvement is constant with delay in the erroneous

[10] N. Vaidya and M. Mehta, "Delayed Duplicate Acknowledgements: A TCP Unaware Approach to Improve Performance of TCP over Wireless," Texas A&M University, Technical Report 99-033, February 1999.

[11] S. Lee, S. H. Lee, J. S. Lim, "Fast snoop scheme for TCP connections in wired-wireless environments," IEICE trans. comm., Vol. E91-B, No. 9, pp. 2998-2999, September 2008.

[12] J. Kim, K. Chung, "C-snoop: Cross layer approach to improving TCP performance over wired and wireless networks," IJCSNS, Vol. 7, No. 3, pp. 131-137, March 2007.

[13] S. Vandarkar, A. L. Reddy, N. Vaidya, "TCP-DCR: A novel protocol for tolerating wireless channel errors" , IEEE trans. mobile computing, Vol. 4, No. 5, pp. 517-529, October 2005.