BaseStation Assisted TCP: A Simple Way to Improve Wireless TCP

Shaoen Wu, Saâd Biaz, Yiming Ji, Bing Qi

Dept. of Computer Science and Software Engineering
Auburn University
107 Dunstan Hall, Auburn University, AL, 36849-5347, USA
{wushaoe,sbiaz,jiyimin,qibing1}@auburn.edu

Abstract. In recent years, extensive research effort has been devoted to TCP congestion control in hybrid wired-wireless networks. A general agreement is that the TCP sender should respond differently to wireless losses and disconnection, i.e., not slow down as drastically as for congestion losses. Thus, research focus for wireless TCP congestion control is the discrimination between the wireless inherent packet losses and the network congestion packet losses in wired network. In addition, researchers attempt to detect temporary or lengthy wireless disconnection. This paper proposes a simple but novel strategy, dubbed *BSA-TCP* (Base Station Assisted TCP), (1) to accurately discriminate wireless losses from wired network congestion losses and (2) to detect and notify a TCP sender about wireless disconnections. The key distinctive feature of the proposed scheme is its general use for most issues at stake for TCP over wireless: loss discrimination, wireless disconnection and handoffs. It also circumvents the asymmetric problem that acknowledgements might follow different paths from those of data packets. Such asymmetry problem is common to mechanisms that buffer and retransmit wireless lost data packets locally at the base station. The proposed method also addresses energy efficiency.

Key Words: wireless TCP, energy efficiency, loss discrimination

1 Introduction

With wireless network evolving popularly to access Internet, enormous research effort has been dedicated to TCP congestion control over hybrid networks. Extensive efforts include [1–10]. In conventional wired networks, TCP congestion control assumes that the packet loss is due to network congestion. If a packet loss occurs, a TCP sender will throttle its sending rate by reducing its congestion window size [11]. If network congestion is severe, the TCP sender will time out and start from slow-start phase for packet transmission. Therefore the network load is reduced at the cost of throughput. When a wireless link is involved, there are mainly two cases defeating the TCP congestion control. The first case is packet drop due to data bits error or packet loss through existing poorly connected wireless link, which is severely prone to interference. The wireless packet loss misleads TCP sender as congestion loss. The second case results from temporary or lengthy wireless disconnection due to some physical factor, such as handoff, location obstacles or weather [12]. The disconnection can improperly trigger timeouts at the TCP sender. Thus, for packet loss and packet drop under poor wireless channel conditions, a key is loss discrimination, so that the TCP sender reacts appropriately to each type of loss: throttle as usual for congestion losses, but keep up for other losses. However, as for the wireless disconnection, some explicit or implicit feedback is needed to notify TCP sender not to invoke slow-start [11]. As a result, the throughput can be largely improved for those TCP applications, like Web browsing or FTP, through the wireless link. Several innovative solutions were proposed to address these problems and difficulties in wireless TCP congestion control. Some of them can discriminate the wireless loss with some probability, such as [1]. And, others, like [2], [7], [8], [10], proposed mechanisms with local buffering and retransmission of lost wireless packet with the help of some intermediate network components, like the base station or the access point.

Proposed BSA-TCP tries to cover all these problems with simple but efficient techniques. We assume that TCP data packets flow from host in wired network to wireless terminal. This assumption is reasonable because most TCP application servers are wired connection. In the proposed strategy, the base station or wireless access point identifies the congestion loss from packet sequence number gap between two successive arriving packets on wired path between the fixed host (TCP sender) and the base station. Then the base station notifies the mobile host (TCP receiver) of the congestion loss by setting a flag in the header of out of order packets. This flag means for the mobile host that the packet loss is due to congestion. If there is no flag set in packet header, it assumes the loss occurred on the wireless link. The mobile receiver can then notify the loss diagnosis with a flag on the acknowledgment (duplicate acknowledgment in fact) to fixed host (TCP sender). Thus, the sender fixed host can appropriately responds to different packet losses. In addition, the basestation detects wireless disconnection and feeds it back to the fixed host by keeping a *copy* of the last acknowledgment sent from mobile host to fixed host. If the wireless connection is cut off or handoff happens, the base station detects this with cooperation from lower layers and send the acknowledgment copy with a *zero receive window*. This trick exploits the TCP ZWA (Zero Window Advertisement) to force the sender into persist mode [13]. Thus the sender does not suffer from the wireless disconnection.

Comparably, since no data packet buffering is required at the base station, the proposed technique eliminates partial work load at the base station and totally maintains the end-to-end semantic. The proposed scheme discriminates packet loss with absolute precision and feeds back wireless disconnections with the cooperation from lower layers. In addition, it doesn't suffer from the asymmetric path problem. For the implementation considered, this mechanism requires limited processing and can coexist with regular TCP.

The remaining of this paper is organized as follows: In Section 2, the related progress and achievement similar to the proposed work strategy is briefly surveyed. Section 3 describes the proposed schema and its advantages and disadvantages. Section 4 discusses the simulation. Finally, Section 5 concludes this paper.

2 Related Research

To discriminate the wireless loss and notify the sender about wireless disconnection, researchers have proposed innovative mechanisms to address these issues in different location of a hybrid wired/wireless path. Generally, these proposals can be characterized into two first-level categories: end-to-end and non-end-to-end.

End-to-end: In this category, the TCP connection retains the end-to-end semantic. Most of the proposals fall into this category because the end-to-end semantic is very important and basic to end-to-end transport protocols and it should be preserved in any technique enhancement. In the end-to-end category, some proposals still require the cooperation of intermediary nodes, like routers, on the wired network. Some involve the base station. And exceptionally, Freeze-TCP [4] is a genuine end-to-end mechanism that does not require any intermediate node's cooperation.

***Router involved*:**In this sub-category, the queue management is expected to help discriminate wireless losses. Since these routers are located before the wireless hop, this type of mechanism still works even if acknowledgments follow different paths from data packets path. The proposal “de-randomize packet loss” [1] is such an innovative mechanism. When congestion happens, the router deliberately, **not randomly**, drops some specific pre-marked packets. Then the mobile receiver can discriminate packet losses with high probability. The receiver notifies the fixed host sender about the nature of the packet loss. This mechanism is very promising in packet loss discrimination because it doesn't split the TCP connection. However, it's difficult to detect wireless disconnections, which cannot be neglected in the wireless environment.

***Base station as a gateway*:** The base station acts as a proxy or gateway to bridge wired path and wireless connection. With cross-layer cooperation or connection splitting, the base station help discriminate packet losses or notify the sender about the wireless disconnections. When the base station buffers data packets and retransmits lost packets (on wireless link), the sender in general is “shielded” from wireless losses: it does not detect them. Since losses are locally recovered, this speeds up the recovery of wireless lost packets.

Snoop [2] is an innovative model typically in this subcategory with a link layer that is TCP-aware. It buffers data packets at the basestation, and examines each acknowledgement to detect wireless packet losses. It locally retransmits wireless lost packets and blocks duplicate acknowledgments related to wireless losses. Snoop improves the throughput by hiding wireless losses from the fixed host (sender) and locally retransmitting wireless losses. However, Snoop requires buffering data packets for each TCP connection. Also, Snoop has no mechanism to cope with wireless disconnections. Another creative idea is **M-TCP** [7]. M-TCP splits the TCP connection into two connections at the *Supervisor Node* (it could be the base station): One TCP connection from sender to BaseStation, and a second TCP connection from BaseStation to mobile host. The BaseStation acknowledges data for the first TCP connection only when the mobile host acknowledges data on the second TCP connection. Note that the BaseStation holds the last acknowledgment from mobile host to fixed host (sender). If no acknowledgment is received by the BaseStation over some period, it infers that wireless disconnection happens. In order to inform the sender, the BaseStation modifies the last acknowledgement with zero window and sends it back to force the sender into persist mode. New acknowledgments from the mobile host will exit the sender out of the persist mode. M-TCP mainly handles the disconnection and low-bandwidth situations adn it is not concerned with packet losses due to packet error. Obviously, M-TCP mechanism requires a precise timing scheme to ensure that the last acknowledgment be sent before the sender times out.

Despite many advantages, generally, the above strategies in this subcategory have the following limitations:

- Local retransmissions at the BaseStation may result in high cost buffering of TCP flows. Because the number of transiting TCP flows is hard to estimate, buffer dimensioning becomes a key issue. Besides, large buffers will require considerable processing when inserting or retrieving packets. This problem is more severe with high-speed wireless links because it needs to buffer more outstanding packets.
- In addition, schemes in this category often face a common problem named *asymmetric path problem* when acknowledgments passes through a reverse path different from that of the data packets. This case is possible, for example in handoff.

No intermediate node involved:Besides above proposals, there is a mechanism named **Freeze-TCP** [4], which doesn't involve any intermediate node. The mobile host detects the impending disconnection to BaseStation with some kind of trigger (for example, from layer-two). Before the disconnection, such as the handoff, occurs, the mobile host freezes the sender with a Zero Window Advertisement (ZWA) in advance. When the connection is restored, the mobile host will re-open the congestion window at the sender with normal acknowledgments. Freeze-TCP is especially efficient for wireless disconnections during handover. However, it does not discriminate wireless losses from congestion losses. Also, it does not handle well unplanned wireless disconnections: some wireless disconnection cannot be predicted due to, for example, an obstacle while roaming. Other similar work in this category includes **TCP-Real** [14] and **TCP-Westwood** [15].

Non end-to-end: Proposals in this category split the connection between the fixed host and mobile host into two TCP connections: these two TCP connections work independently without preserving end-to-end semantic. **I-TCP** [10] is such a strategy. It splits the TCP connection at the BaseStation. The first one is the usual TCP connection from fixed host to base station. The second one from base station to mobile host spans the wireless hop. The base station buffers all data packets and acknowledges receipt to the fixed host. So, the first TCP connection in general deals only with congestion losses and is unaware of losses over the wireless hop: the base station locally retransmits packets lost over the wireless link. Therefore, one major problem is that, even if the wireless connection is fairly poor and the receiver doesn't receive many packets, the original sender (fixed host) will assume that all transmissions are successful . This is the key weakness of I-TCP.

In summary, here are the desirable characteristics that a scheme should fulfill to improve TCP over hybrid wired-wireless networks:

1. The mechanism should be able to discriminate wireless losses and identify a wireless disconnection. These two issues are related to some degree. For example, although some innovative mechanism can discriminate packet losses very well, the

receiver may not be able to send any discrimination feedback if the wireless disconnection lasts too long, leading TCP sender to timeout. This greatly degrades the performance. At the same time, only the base station can accurately diagnose the nature of a packet loss.
2. The end-to-end semantic of TCP should be preserved in the new proposal because it's the core of a transport layer protocol.
3. When a base station is involved, the proposed scheme should take into account the problem of different forward and reverse path problem.
4. When a wireless disconnection occurs, the re-transmission in the wireless connection should be minimized to maximize energy efficiency for power sensitive mobile terminals [6].
5. The last desired characteristic is that the implementation of the new scheme should be simple and compatible with the existing regular TCP.

Some of the above proposals solve the problem of packet loss discrimination while others address the problem of wireless disconnection. A few address the simplicity issue. The proposed BSA-TCP can efficiently address these issues.

3 Basestation Assisted TCP

In this section, we firstly explain how Basestation Assisted TCP **BSA-TCP** works in detail. Then a brief analysis is sketched followed by a discussion regarding implementation concerns.

3.1 BSA-TCP mechanism

Basically, this proposal is a kind of end-to-end mechanism involving the BaseStation as a gateway. But there is no split of the TCP connection and no buffering of data packets. The proposed scheme addresses all issues discussed above. At the same time, it should be easy to implement in practice. Table 1 summarizes the characteristics of different proposals involving a BaseStation. The table includes only those schemes similar or related to the proposed scheme.

Table 1. Wireless TCP Mechanisms Comparision

Mechanism	I-TCP	M-TCP	Freeze-TCP	BSA-TCP
End-to-end	no	yes	yes	yes
Loss discrimination	yes	no	no	yes
Wireless disconnection	no	yes	yes	yes
Light buffering at BaseStation	no	no	yes	yes
Asymmetic path problem	yes	yes	yes	no
IPSec problem	yes	yes	no	yes

BSA-TCP retains the end-to-end semantic and efficiently addresses the problems wireless TCP faces up, such as loss discrimination and detection of disconnection. With the cooperation from the BaseStation, the wireless loss can be discriminated and the disconnection can be detected and fed back to the sender. In this proposal, we assume that local retransmissions at wireless link layer work well. This assumption is reasonable because almost all popular wireless link layer protocols, such as link layer of IEEE802.11 and 3G network [16–18], include the local link layer retransmission, like ARQ, with tighter timing constraints than TCP at transportation layer. We also assume that the base station can detect the wireless disconnection from link layer signalling. For instance, when link layer infers that the wireless link is disconnected, it signals to upper layers. In the following, we discuss how BSA-TCP works and solves each problem confronted by wireless TCP.

BSA-TCP requires the cooperation of the sender, the BaseStation and the receiver. Packet discrimination can be signaled using an unused bit on the TCP header or using a new option field. The BaseStation is the best place where (1) to accurately discriminate packet losses and (2) to feed back a wireless disconnection to sender because an acknowledgment message from the mobile host cannot be sent through during a wireless disconnection. Thus, the BaseStation is a key element to the solution. The following explains in detail how different components operate collaboratively and procedurally in this strategy.

Fixed Host (TCP sender): The functionality at the sender is pretty simple. The fixed host must appropriately react to packet losses. It checks the discrimination flag in the acknowledgement: If set, the fixed host infers that the packet loss is a wireless loss and does not trigger any congestion control mechanism to throttle the sending rate: The TCP sender just retransmits the lost packet. All other actions and parameters are the same as in traditional TCP without packet loss. Otherwise, if the discrimination flag is not set, the sender infers that this is a congestion loss and regular TCP congestion control algorithms are invoked. In case of wireless disconnection, the TCP sender gets an implicit feedback with the acknowledgment of zero flow control receive window from BaseStation. Therefore, the TCP sender enters into persistent mode [13], in which it freezes all the parameters and waits for a fresh acknowledgment.

Mobile Host (TCP receiver): In this strategy, the mobile host requires only a slight modification to check whether to set the loss discrimination flag in the acknowledgment (based on the loss notification flag in data packet). When the mobile host receives a data packet, it checks whether a loss discrimination flag is set or not. If the flag is set, it infers that the packet sequence gap is due to the congestion losses. Then the mobile host does not set the wireless loss flag in the acknowledgement so that congestion control is invoked at the sender. If loss discrimination flag is not set in data packet, the mobile host knows that the sequence gap is due to wireless packet drops or loss in the wireless connection. Then it sets the wireless loss flag in all following duplicate acknowledgments so that the TCP sender fixed host identifies the loss as not related to congestion. Then the TCP sender doesn't reduce the congestion window and just retransmits the lost packet. Note that the flag is set in all following duplicate acknowledgments (DACK) in case the first DACK with the flag set is lost on the wireless link.

BaseStation: In this strategy, the BaseStation performs significant tasks: It accurately discriminates packet losses and detects the wireless disconnection.

Loss discrimination: the BaseStation monitors the sequence number of every data packet passing through it towards the mobile host. If there is sequence gap between two successive data packets, it accurately infers that the packet loss or out of order is due to network congestion ahead of wireless link. If that case, the BaseStation sets a flag in the out of order packet header to notify the mobile host that the sequence gap is NOT due to a wireless loss. Or, the BaseStation will keep data packets unchanged. This notification cooperates with above work of Fixed Host and Mobile Host to achieve the wireless discrimination.

Disconnection detection: The wireless disconnection from BaseStation to mobile host can be detected by the local link layer retransmission with timer mechanism. As we mentioned above, most of the wireless link layer protocols today are integrated with the retransmission capability. And normally the timer value for this link layer retransmission is contingent than that of timer for the TCP packet loss.

We assume our agent at the BaseStation will get a notification or trigger from the link layer when link layer retransmission times out. This is a reasonable assumption because the IEEE802.21 organization is working on a standard concerning such notifying triggers [19]. The BaseStation can then infer the wireless link is unavailable to mobile host.

In good wireless channel condition, when the BaseStation relays the acknowledgement back to the TCP sender, it retains one *copy* of the last acknowledgement. If it detects wireless disconnection, it resets the receive windows in the copy of the acknowledgement to 0 and resends this acknowledgement back to the TCP sender fixed host. Because the receive window is different from the previous acknowledgement, this acknowledgement will not be thought as a Duplicate Acknowledgement at the TCP sender. The receive window 0 in the acknowledgement forces the TCP sender into persist mode, in which the TCP sender freezes all parameters in congestion control and waits for the update acknowledgement to reactivate. This is different from M-TCP [7]. In M-TCP, the BaseStation holds the last acknowledgement **itself**, not the *copy* of it. Thus, M-TCP requires a delicate timing mechanism to make sure the absence of that last acknowledgement will not cause TCP sender to time out. Since our proposal doesn't hold the acknowledgement itself, just a copy of it, no timing mechanism is required with the proposed solution. When the wireless connection is restored, the BaseStation gets notification from the link layer and re-sends the copy of the last acknowledgement with the receiving window unchanged (original non null receive window) back to TCP sender. Or it relays back a new acknowledgement from mobile host. Then this non-zero receive window acknowledgement activates the TCP sender to normal transmission status with the congestion window restored. It's obvious that this reopened congestion window is far more aggressive and thus the throughput should improve significantly.

Asymmetric Path: In this strategy, there is no TCP connection split at the BaseStation. The BaseStation doesn't buffer data packets. Therefore it avoids the problem of acknowledgements transmitted on different path from that of data packets: For example, after a handoff, the acknowledgement might go through a new BaseStation. When the TCP sender receives the acknowledgement, it need not know where the acknowledgement passed through and just responds as end-to-end.

From the above discussion, we can conclude that the BSA-TCP has the following advantages:

- It does not split a TCP connection. Thus, it retains the end-to-end semantic.
- At the same time, because no packet is buffered at BaseStation, this strategy does not require the monitoring of acknowledgements from the mobile host and therefore avoids the asymmetric acknowledgement path issue.
- Since the BaseStation is the bridge between wired and wireless links, it accurately discriminates the wireless loss.
- The ability to detect wireless disconnections can efficiently reduce spurious retransmissions. Thus this proposal is energy efficient to some degree.
- Furthermore, precise wireless packet loss discrimination and feedback of wireless disconnection efficiently eliminates TCP timeouts resulting from wireless link, and thus avoids the low-efficient slow-start phase.
- Additionally, it's compatible with the regular TCP. If the TCP sender does not "understand" the loss discrimination setting, it will ignore this setting and will respond conservatively as the regular TCP sender. And if the receiver mobile host has no such implementation, it will not discriminate the loss flag from the BaseStation and just act as the regular TCP receiver. Thus there is almost no influence on the regular TCP as the bottom line operation.

Note also that, although there is a little implementation needed for wireless discrimination at the sender, as for detecting wireless disconnection, this functionality runs at the BaseStation, and there is no extra implementation needed at the sender. The reason is that, even with the regular TCP, the 0 window acknowledgement from BaseStation will force TCP sender to work for wireless disconnection.

However, the BSA-TCP confronts one common problem for all mechanisms that involve monitoring TCP header packets in intermediate nodes such as a BaseStation. If IPSec is used, intermediate nodes, including the BaseStation, cannot read the TCP header [3] and extract information such as sequence numbers. Another disadvantage is to work cross layer, which is supposed to be a trend as to complex wireless network in the future.

3.2 BSA-TCP performance and implementation brief analysis

In [1], the authors derived the enhanced formula to estimate the TCP connection throughput with the wireless link. Consider a discriminator with accuracies A_c and A_w is used for congestion loss and wireless loss with TCP (A_c represents the ratio of congestion losses correctly diagnosed. A_w is similarly defined for losses not related to congestion). p_c and p_w respectively represent the congestion packet loss rate and wireless packet loss rate. The throughput can be expressed as:

$$Thrg_w = \frac{1}{RTT\sqrt{\frac{2bp_{cN}}{3}} + T_0 min(1, 3\sqrt{\frac{3bp_{cw}}{8}})p_{cw}(1+32p_{cw}^2)} \qquad (1)$$

where $p_{cN} = A_c p_c + (1 - A_w)p_w$ and $p_{cw} = p_c + p_w$. Without discrimination, the traditional TCP throughput should take $p_{cN} = p_c + p_w$ and $p_{cw} = p_c + p_w$ with $A_c = 1$

and $A_w = 0$. Since BSA-TCP accurately discriminates all types of losses, then $A_c = 1$ and $A_w = 1$. Moreover, timeouts at TCP sender are eliminated for the wireless losses. Therefore, the value p_{cw} in above equation reduces to p_c. As a result, the throughput for BSA-TCP can be expressed as:

$$Thrg_{BSA} = \frac{1}{RTT\sqrt{\frac{2bp_c}{3}} + T_0 min(1, 3\sqrt{\frac{3bp_c}{8}})p_c(1+32p_c^2)} \tag{2}$$

In other words, the throughput is the same as on strictly wired network. Thus the theoretical throughput can be achieved through the BSA-TCP, without considering wireless disconnection, should be

$$\frac{Thrg_{BSA} - Thrg_w}{Thrg_w} \tag{3}$$

In practical implementation, as for the loss flag presented in currently TCP packet header, two possible ways are available. One is to use an unused or reserved bit in one field of TCP header. The other method is to use options of the TCP header.

4 Simulation and Analysis

We evaluate the performance improvement with network simulator *ns-2* [20]. Mainly, the performance improvement is due to two contributions: one from the packet loss discrimination and the other from the wireless disconnection detection. Since our strategy in detecting wireless link disconnection is essentially similar to that of M-TCP [7], it is reasonable to expect the same improvement. Thus, this paper only focuses on evaluating throughput improvement resulting from the sole contribution of packet loss discrimination.

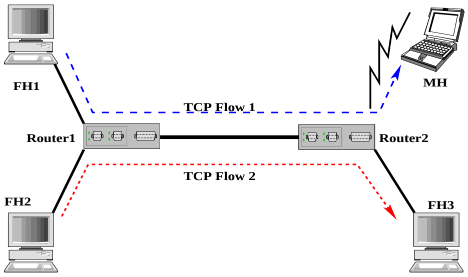
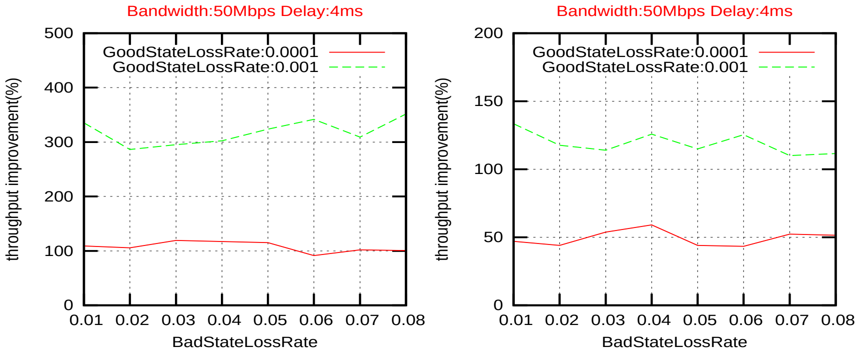


Fig. 1. Simulation Network Architecture

Based on the popular IEEE802.11, wireless link bandwidth is chosen from 10 Mbps through 100 Mbps in our experiments, with the concern that current WLAN normally covers data rate from 11 Mbps to 108 Mbps (in recently IEEE802.11g). The propagation delay varies from 2ms to 8ms for wireless link. Two-state Markov wireless loss is taken as the error model in target wireless connection, in which the link good state and bad state probability are respectively 0.7 and 0.3. When link is in bad state, the wireless packet error rate $L_{BadStateLossRate}$ ranges from 0.001 to 0.1, and 0.0001 to 0.01 for $L_{GoodStateLossRate}$ in good state. Thus the overall loss rate in the wireless link in experiment should be:

$$L = 0.7L_{GoodStateLossRate} + 0.3L_{BadStateLossRate} \tag{4}$$

We evaluate the performance improvement for TCP-Newreno and TCP-Sack. The basic configuration is in Figure 1. Because of the space limitations, we just include part of the results here.



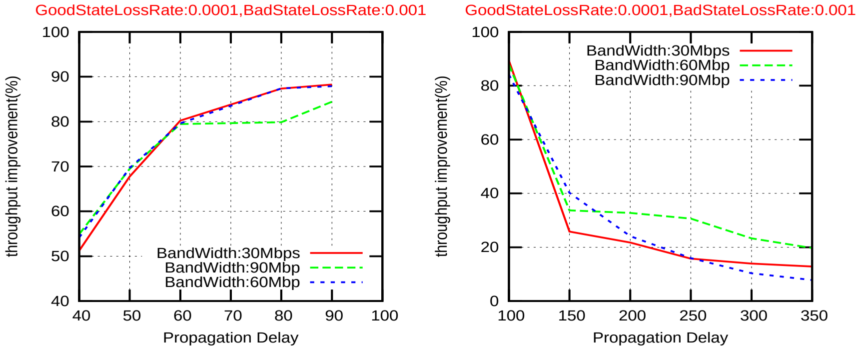
(a) TCP-Sack (b) TCP-Newreno

Fig. 2. Throughput Improvement in different TCP flavors

In Figure 2(a), and Figure 2(b), the x-axis represents the wireless packet loss rate in bad state, which ranges from 0.01 to 0.08. The y-axis represents the BSA-TCP throughput improvement in percentage over the regular TCP (no wireless loss discrimination). The short one hop delay used in these two experiments stands for the delay at the last hop (wireless access) in WLAN network. These figures show that TCP-Newreno and TCP-Sack exhibit a significant performance improvement even when the good state packet loss rate is very low. The improvement is more dramatic when the good state packet loss is higher. In general, the throughput improvement is over 50% for TCP-NewReno, 100% for TCP-Sack.

Overall, the throughput improvement is more significant for TCP-Sack than for TCP-Newreno. TCP-Sack exploits better the discrimination of losses and is more efficient in avoiding spurious retransmissions.

Figure 3(a) illustrates the impact of bandwidth (at the bottleneck) and large propagation delay on throughput improvement. Figure 3(a) plots three curves for bandwidths 30 Mpbs, 60 Mpbs, and 90 Mbps. The y-axis is the improvement compared to regular TCP and x-axis is the propagation delay value of the wireless lossy link. This figure illustrates that wireless bandwidth doesn't have much effect on improvement. Higher round trip time results in better throughput improvement when loss discrimination is used. Without loss discrimination, regular TCP takes more time to recover from halving the congestion window. As the round trip time increases, regular TCP gets more penalized by halving the congestion window when losses occur. Loss discrimination



(a) Short Delay (ms) (b) Large Delay (ms)

Fig. 3. Throughput Improvement in different Bandwidth

eliminates this penalty for wireless losses. But, as shown in Figure 3(b), when the propagation delay is too large, the improvement decreases drastically. This is because long propagation delay results in less packets lost in wireless link and thus less "room" for improvement.

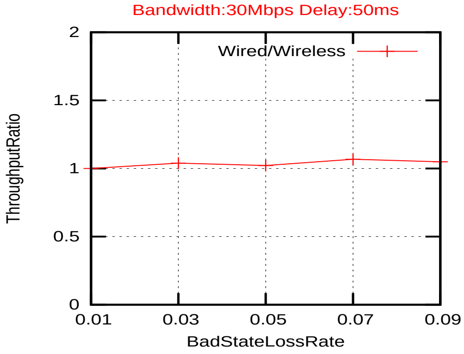


Fig. 4. BSA-TCP Fairness on Throughput

Fairness is also evaluated when BSA-TCP is introduced. As shown in Figure 1, two different TCP flows pass through the same backbone. One is from a fixed host to another fixed host connected to network through wired links. The other is from a fixed host to mobile host with BSA-TCP. The result demonstrated in Figure 4 reports the fairness. Here, the x-axis is the wireless packet loss rate in bad state. The y-axis, $ThroughputRatio$, represents the throughput of TCP flow 1 through wireless link with BSA-TCP over that of the TCP flow 2 in the wired regular TCP connection. It's observed that the ratio varies around 1, which justifies the fairness.

As stated above, since the M-TCP [7] has exhaustedly explored wireless disconnections, this work does not replicate such simulation efforts.

5 Conclusion

This paper introduces a novel strategy, dubbed BSA-TCP, for wireless TCP congestion control. With the BaseStation helping in packet loss discrimination and feeding back the disconnection notification, BSA-TCP efficiently addresses the wireless disconnection detection, loss discrimination and asymmetric path issues common in wireless TCP. We also find that the BaseStation is necessary in the wireless TCP when the disconnection is considered and if we want to discriminate the loss precisely and deterministically. With consideration of the easy implementation in practice and the wireless TCP problems it can solve, this strategy is pretty efficient.

References

1. Biaz, S., Vaidya, N.: “De-Randomizing” Congestion Losses to Improve TCP Performance Over Wired-Wireless Networks. IEEE/ACM Transactions on Networking **13** (2005) 596–608
2. H. Balakrishnan, S. Seshan, E.A., Katz, R.: Improving TCP/IP Performance over Wireless Networks. In: the 1st ACM Intl’ Conf. On Mobile Computing and Networking (Mobicom). (1995)
3. V.Tsaoussidis, I.Matta: Open Issues on TCP for Mobile Computing. In: Wireless Communications and Mobile Computing. Volume 2. (2002)
4. Goff, T., Moronski, J., Phatak, D.: Freeze-TCP: A True End-to-End Enhancement Mechanism or Mobile Environ Environments. In: INFOCOM. (2000)
5. Bakshi, B., Krishna, P., Vaidya, N., Pradhan, D.: Improving Performance of TCP over Wireless Networks. In: the IEEE 17th ICDCS’97. (1997) 365–373
6. Balakrishnan, H., Padmanabhan, V., Seshan, S., Katz, R.: A Comparison of Mechanisms for Improving TCP Performance over Wireless Links. ACM/IEEE Transactions on Networking **5** (1997) 756–769
7. Brown, K., Singh, S.: M-TCP: TCP for Mobile Cellular Networks. In: the ACM SIGCOMM Computer Communication Review. (1997) 19–43
8. Haas, Z., Agrawal, P.: Mobile-TCP: An Asymmetric Transport Protocol Design for Mobile Systems. In: the IEEE International Conference on Communications (ICC’97). (1997)
9. Bhagwat, P., Bhattacharya, P., Krishna, A., Tripathi, S.K.: Enchancing Throughput over Wireless LANs using Channel State Dependent Packet Scheduling. In: INFOCOM. (1996) 1133–40
10. Bakre, A., Badrinath, B.: I-TCP: Indirect TCP for Mobile Hosts. In: Proceedings of the IEEE ICDCS’95. (1995)
11. Jacobson, V.: Congestion avoidance and control. In: Symposium proceedings on Communications architectures and protocols. (1988) 314–329
12. Tanenbaum, A.S.: Computer Networks. Prentice Hall (2002)
13. Richard, S.: TCP/IP Illustrated, Volume 2: The Implementation. Addison-Wesley (1994)
14. Tsaoussidis, V., Zhang, C.: Tcp-real: Receiver-oriented congestion control. Computer Networks **40** (2002) 477–497
15. C. Casetti, M. Gerla, S.M.M.Y.S., Wang, R.: Tcp westwood: Bandwidth estimation for enhanced transport over wireless links. In: In Proceedings of ACM Mobicom. (2001) pp287–297
16. IEEE802.11: http://standards.ieee.org/getieee802/download/802.11-1999.pdf. (1999)
17. 3GPP: TSG, RAN Physical layer procedures (FDD), version 5.3.0 (2002)
18. Bender, P., Black, P., Grob, M., Padovani, R., Sindhushayana, N., Viterbi, A.: CDMA/HDR: A Bandwidth-Efficient High-Speed Wireless Data Service for Nomadic Users. In: IEEE Communications Magazine. (2000) 70–77
19. IEEE802.21: http://www.ieee802.org/21/archived_docs/Documents/ fmip_and_l2trigger.pdf. (2005)
20. ISI: (http://www.isi.edu/nsnam/ns/)