Performance Evaluation of TCP with Adaptive Pacing and LRED in Multihop Wireless Networks

G. Sankara Malliga[1], Dr.Dharmishtan K Varughese[2]

[1] Assistant Professor/ECE, RVS College of Enginnering & Technology, Coimbatore, Tamilnadu, India.
[2] Professor/ECE, karpagam College of Engineering, Coimbatore.

Abstract

Transmission Control Protocol (TCP) was designed to provide reliable end-to-end delivery of data over unreliable networks. In practice, most TCP deployments have been carefully designed in the context of wired networks. Ignoring the properties of wireless and Ad-hoc Networks can lead to TCP implementations with poor performance. In a wireless network, however packet losses occur more often due to unreliable wireless links than due to congestion. When using TCP over wireless links, each packet loss on the wireless link results in congestion control measures being invoked at the source. This causes severe performance degradation. If there is any packet loss in wireless networks, then the reason for that has to be found out. If there is congestion, then only congestion control mechanism has to be applied. This work shows the performance of TCP with Adaptive Pacing (TCP-AP) and Link Random Early Discard (LRED) as queuing model in multihop transmission when the source and destination nodes are in mobile nature. The adaptive pacing technique seeks to improve spatial reuse. The LRED technique seeks to react earlier to link overload. This paper consists of simulated environment results under different network scenarios. This work proves that the combination of TCP-AP and LRED give much better result than as the individual technique. Simulations are done with the use of NS-2.

Key Words— *Wireless Networks, TCP, Congestion Control, Spatial reuse, Adaptive Pacing, Link Random Early Discard*

1 Introduction

Wireless Networking has emerged as a promising technology to meet the challenges in next generation networks. The multihop wireless network has to be utilized efficiently to improve the overall performance of the network. Most of the Internet based applications use Transmission Control Protocol (TCP) as a transport protocol, since it provides end-to-end reliable transmission of data [2]. In multihop wireless networks like WMNs (Wireless Mesh Networks), a number of clients can generate TCP and UDP traffic which goes in the same multihop path from one edge node to another edge node or edge node to gateway node. Hence, all the packets going in that path have to be scheduled with an interval of Four Hop Delay (FHD) to reduce the contention between the packets, thereby achieving better spatial channel reuse. This spacing of packet transmissions is required in multihop wireless networks irrespective of higher layer protocols used when the flows are running for more than four hops.

As a result of the advancement of wireless technology and the proliferation of handheld wireless terminals, recent years have witnessed an ever-increasing popularity of wireless networks, ranging from wireless Local Area Networks (WLANs) and wireless wide-area networks (WWANs) to mobile ad hoc networks (MANETs). In WLANs (e.g., the Wi-Fi technology) or in WWANs (e.g., 2.5G/3G/4G cellular networks), mobile hosts communicate

DOI: 10.5121/ijwmn.2010.2410

with an access point or a base station that is connected to the wired networks [3]. Obviously, only one hop wireless link is needed for communications between a mobile host and a stationary host in wired networks. In contrast, there is no fixed infrastructure such as base stations or access points in a MANET. Each node in a MANET is capable of moving independently and functioning as a router that discovers and maintains routes and forwards packets to other nodes. Thus, MANETs are multi-hop wireless networks by nature. Note that MANETs may be connected at the edges to the wired Internet.

TCP is an adaptive transport protocol that controls its offered load (through adjusting its window size) according to the available network bandwidth. It additively increases its congestion window in the absence of congestion and throttles down its window when a sign of congestion is detected. In the wired Internet, congestion is identified by packet loss, which results from buffer overflow events at the bottleneck router. However, it is unclear how well such TCP mechanisms work in a multihop wireless networks [1]. Since in multihop wireless networks, packet loss is not only due to congestion but also due to some wireless network properties such as hidden & exposed terminal, link error, channel errors etc.

2 TCP’s Challenges in Multihop Wireless Networks

Multihop wireless networks have several characteristics different from wired networks. Firstly, in a typical wireless network that uses IEEE 802.11 MAC packets may be dropped due to either buffer overflow or link-layer contention caused by hidden terminals. Such losses directly affect TCP window adaptation. Secondly, wireless channel is a scarce, shared resource. Improving channel utilization through spatial channel reuse is highly desirable. Multiple nodes that do not interfere with each other should be encouraged to transmit concurrently. How well TCP utilizes the multihop wireless channel through spatial reuse poses another important issue. A fundamental problem is that, TCP has to interact with ad hoc forwarding and IEEE 802.11 MAC in a multihop wireless network, which exhibits features quite different from the wired or wireless cellular networks. Earlier research on TCP performance over ad hoc networks has been focused on the impact of mobility induced factors, such as link breakage and routing failures [4], [5]. However, the interaction between TCP and the underlying multihop forwarding with the IEEE 802.11 MAC is left unaddressed.

TCP is the dominant transport protocol to carry most of Internet data traffic, e.g., Email, FTP, P2P. Each TCP source just iterates its own congestion control algorithm to respond to the congestion signal generated by AQM (Active Queue Management) schemes in the network. So, TCP/AQM is essentially a distributed, close-loop congestion control algorithm. On one hand, the distributed nature of TCP makes it easy to implement, because there is no global information needed. On the other hand, the close-loop nature of the system makes it difficult to analyze the overall performance of TCP/AQM under various configuration, compared to the traditional open-loop teletraffic theory. As an example, the buffer sizing for the Internet router shared by TCP flows has been a delicate issue. Although extensive research efforts have been made so far [6, 7, 8, 9], the problem is still far from being completely understood and solved, mainly because the underlying mechanisms of the close-loop system are quite different from that of usual open-loop queueing theory. Traditionally, the rule-of-thumb for buffer sizing is set to the bandwidth-delay product [6]; for a system with N flows, capacity NC, and the roundtrip-time (RTT) T, the buffer size should be set to NCT, i.e., O(N).

3 Problems and Related Studies

The performance of TCP over multihop wireless network is greatly affected by packet loss due to Link layer contentions rather than loss due to congestion (buffer overflow at the intermediate nodes). To improve the performance of TCP over multihop wireless networks, several variants of TCP such TCP ELFN [8], TCP-Feedback [9], and TCP-AP [1], were proposed. These protocols try to distinguish between congestion and non-congestion losses in the network and appropriately take actions to achieve better performance. The rate based protocols such as ATP [11], estimate the available bandwidth between the source and destination, and transmit the packets at the estimated rate. There are other solutions such as Distributed Link RED (LRED) [1] and Neighborhood RED [12] that incorporate Randomly Early Detection (RED) mechanism in queues to improve the TCP performance. LRED improves the performance of TCP flows by implementing RED at the queues and reacting to continuous packet collisions by increasing the MAC backoff time by one packet transmission time.

Several researchers had identified that the poor performance of TCP in IEEE 802.11 based multihop wireless networks is due to the underlying routing and MAC layer protocols [9]. Due to the broadcast nature of wireless channel, the neighboring nodes in the network can not transmit simultaneously. So the packets of the multihop flow contend with each other for the channel at successive hops, if the data arrived at the source is bursty in nature. It is well known that, TCP generates bursty traffic based on the current congestion window size. This leads to self contention, thus increasing chances for dropping the packets. To solve the above said problem, TCP-AP spreads the transmission of successive packets according to the computed transmission rate. The transmission rate is computed by the round trip time (RTT) at the TCP sender. As the spreading of packets is done at the transport layer, it fails to spread the transmission of packets at the MAC layer. The rate estimate at the TCP sender assumes the bandwidth of all links in the path between source and destination is same. This is always not true as MAC selects the transmission rate based on the quality of the channel. Further, if there are two or more TCP flows between same source and destination, the Link layer contention can not be reduced by TCP-AP.

The spacing of the packet transmissions at the MAC is essential for all higher layer protocols, if the traffic is bursty in nature or the network is congested. Hence, we propose the adaptive pacing at the Link layer in IEEE 802.11 multihop wireless networks. In IEEE 802.11 multihop wireless network, by perfectly scheduling the transmission of packets (i.e., pacing packets with FHD), the self contention of packets belonging to a flow can be reduced which reduces the contention and achieves a maximum transmission rate of Rmax = 1 FHD.

Fu et al. [1] pointed out the hidden terminal problem in wireless multihop networks and experimentally showed that for a chain topology the optimal windows size for which TCP achieves best throughput, is roughly given by 1/4 of the hop count of the path. Furthermore, they proposed two enhancements on the link layer: adaptive pacing to distribute traffic on the link layer among intermediate nodes in a more balanced way and link layer RED to throttle TCP senders when incipient congestion is detected. Using simulation, they showed that depending on the scenario, these link layer enhancements improve TCP goodput by 5% to 30% due to better spatial reuse. Xu et al. [12] proposed the neighborhood RED (NRED) scheme on routing layer to throttle TCP senders when incipient congestion is detected, by purposely dropping TCP packets on intermediate nodes. Nodes forming a neighborhood manage a virtual distributed queue in order to coordinate the packet drops of individual nodes. Using simulation, the authors showed that NRED could substantially improve fairness in multihop wireless networks.

Sundaresan et al. [11] and Chen et al. [14] introduced two new special-purpose transport protocols for multihop wireless networks. Both protocols employ pure rate-based transmission of packets, where the transmission rate is determined using feedback from intermediate nodes along the path. In [11], the authors propose to dynamically adjust the transmission rate according to the maximum packet queuing delay on intermediate nodes along the network path. Chen et al. [14] also proposed an explicit rate based flow control scheme for multihop wireless network. Using cross-layer information from both the MAC and the routing layer, the sending rate of a flow is conveyed from intermediate nodes along the path in special control headers attached to each data packet.

In contrast to [15], [11], TCP-AP retains the end-to-end semantics of TCP without relying on any cross-layer information from intermediate nodes along the path. As a consequence, TCPAP can be incrementally deployed, since TCP-AP is not only TCP-friendly, but also TCP compatible. Altman and Jiménez [16] proposed a dynamic scheme for delaying

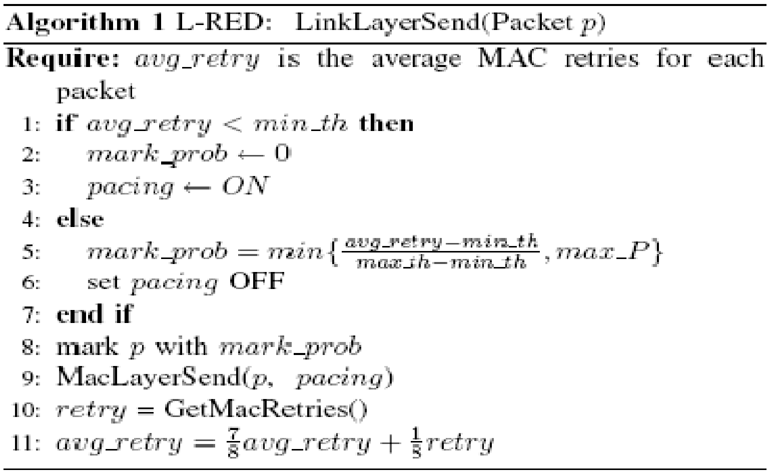
ACKs in order to improve TCP throughput in multihop wireless networks. Using simulation, they showed that for an n hop chain, delaying ACKs yields around 50% more throughput for TCP NewReno. Several authors introduced TCP enhancements for coping with mobility in ad hoc wireless networks over IEEE 802.11. Yu [17] proposed two cross-layer communication mechanisms that further improve TCP performance in case of packet losses due to mobility.

4 Improving TCP Performance

This section describes two techniques to improve TCP performance over multihop wireless networks[1]. The link RED(Random Early Discard) technique seeks to react earlier to link overload. The adaptive pacing technique seeks to improve spatial reuse. The combination of these two techniques is able to improve TCP throughput by as much as 30%.

4.1. Distributed Link RED (LRED)

Our Link RED (LRED) algorithm is based on the observation that TCP can potentially benefit from the built-in dropping mechanism of the 802.11 MAC. The main idea is to further tune up wireless link's drop probability, based on the perceived link drops. While the wired RED provides a linearly increasing drop curve as the queue exceeds a minimum value min_th, LRED does so as the link drop probability exceeds a minimum threshold.



Algorithm 1 L-RED: LinkLayerSend(Packet p)

```
Require: $avg\_retry$ is the average MAC retries for each packet
if $avg\_retry < min\_th$ then
    $mark\_prob \leftarrow 0$
    $pacing \leftarrow ON$
else
    $mark\_prob = min\{\frac{avg\_retry - min\_th}{max\_th - min\_th}, max\_P\}$
    set $pacing$ OFF
end if
mark $p$ with $mark\_prob$
MacLayerSend($p$, $pacing$)
$retry$ = GetMacRetries()
$avg\_retry = \frac{7}{8}avg\_retry + \frac{1}{8}retry$
```

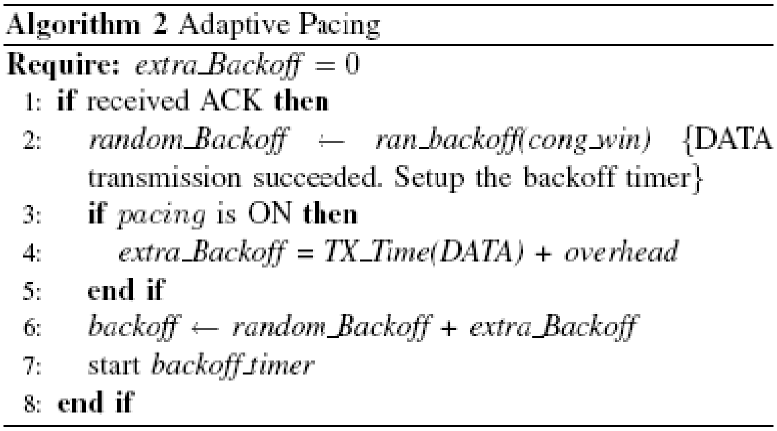
In LRED, the link layer maintains the average number of the retries for recent packet transmissions. The head-of-line packet is dropped/marked from the buffer with a probability based on this average number. At each node, if the average number of retries is small, say less than min th, which means that the node is rarely hidden, packets in the buffer are not dropped/marked. When it gets larger, the dropping/marking probability is computed, and the minimum value of the computed drop probability and a maximum bound max P is used. A feature of this algorithm is that it can integrate with ECN enabled TCP flows. Instead of blindly dropping packets, we can simply mark them at the link layer, and thus allow ECN enhanced TCP flows to adapt their offered load without losing any packets. TCP performance is further improved, by paying the moderate cost of a slightly more complex link-layer design.

To summarize, LRED is a simple mechanism that, by monitoring a single parameter – the average number of retries in the packet transmissions at the link-layer, accomplishes three goals: a) It helps to improve TCP throughput, b) It provides TCP
an early sign of network overload, and c) It helps to improve interflow fairness.

4.2. Adaptive Pacing

Our second technique seeks to take an adaptive pacing approach at the link-layer. The goal is to improve spatial channel reuse, by distributing traffic among intermediate nodes in a more balanced way, while enhancing the coordination of forwarding nodes along the data path. This design works in concert with the 802.11 MAC.

In the current 802.11 protocol, a node is constrained from contending for the channel by a random backoff period, plus a single packet transmission time that is announced by its immediate downstream node. However, the exposed receiver problem [6] persists due to lack of coordination between nodes that are two hops away from each other. Adaptive pacing solves this problem, without requiring nontrivial modifications to the 802.11, or a second wireless channel [8]. The basic idea is to let a node further back-off an additional packet transmission time when necessary, in addition to its current deferral period (i.e. the random backoff, plus one packet transmission time). This extra backoff interval helps in reducing contention drops caused by exposed receivers, and extends the range of the link-layer coordination from one hop to two hops, along the packet forwarding path.



```
Algorithm 2 Adaptive Pacing
Require: $extra\_Backoff = 0$
if received ACK then
    $random\_Backoff \leftarrow ran\_backoff(cong\_win)$ {DATA transmission succeeded. Setup the backoff timer}
    if pacing is ON then
        $extra\_Backoff = TX\_Time(DATA) + overhead$
    end if
    $backoff \leftarrow random\_Backoff + extra\_Backoff$
    start backoff_timer
end if
```

The algorithm works together with LRED as follows: Adaptive pacing is enabled by LRED. When a node finds its average number of retries to be less than min th, it calculates its backoff time as usual. When the average number of retries goes beyond min th, adaptive pacing is enabled and the backoff period is increased by an interval equal to the transmission time of the previous data packet. This way, a better coordination among nodes is achieved under different network load.

5 Simulation Model and Results

Here ns-2 network simulator with the CMU extensions for IEEE 802.11 wireless LAN is used. Here, two network scenarios are considered. In one scenario, that consists of 10 nodes and in the another scenario, that consists of 30 nodes. The routing protocol chosen for analysis is AODV since this routing protocol works better than DSDV and DSR. Here analysis had been done in four cases as: (i) Simple TCP NewReno with queuing DropTail (ii) TCP-AP with queuing DropTail (iii) TCP NewReno with queuing LRED (iv) TCP-AP and queuing LRED.

5.1. Simulation scenario I

This scenario runs a single TCP connection between two mobile nodes over an area of 500m by 500m. The initial conditions of nodes 0 and 1 are (50,90) and (450,410) respectively. All other eight nodes are at (250,250). At time 0.1 sec, node 0 moves to (420,100) and node 1 moves to (10,460) with a speed of 5m/s and also all other eight nodes are moving that will be shown in Figure. (1). Once again at time 100s, node 0 moves towards (2,450) with speed 25 m/s and node 1 moves to (490,40) with 15 m/s speed.

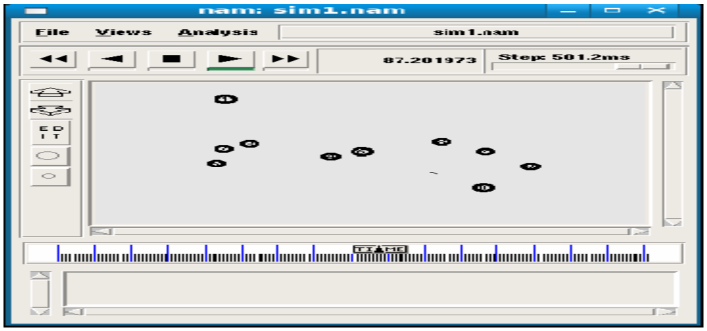


Figure. 1 Simulation Scenario 2 at Simulation Time nearly 87sec

5.3. Simulation scenario II

This scenario consists of 30 nodes moving around the workspace area of 750m x 750m. The nodes are having random motion. The simulation time taken is 500 seconds. Multiple TCP connections and a single TCP connection are provided. That is analyzed. Nodes 8 and 15, 0 and 1, 5 and 12 are initially having single hop and then having multiple hops. The network chosen for analysis is shown in Figure. (2).

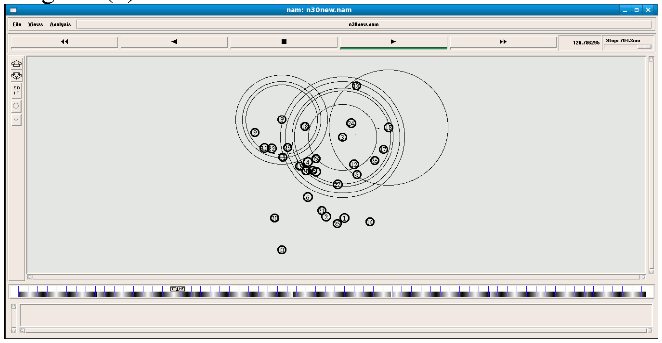


Fig. 2 Simulation Scenario 3 at Simulation Time nearly 127 sec

5.4. Analysis of simulation results

For the Network Scenario 1, during the time 10sec to 75sec, their will be multiple hop between the nodes 0 and 1. So that LRED gives maximum window size during that time. During 80 sec to 120sec, nodes 0 and 1 are closer together, so that window size will be decreased than the previous situation. The window evolution for the above mentioned cases is given in Figure. (3).

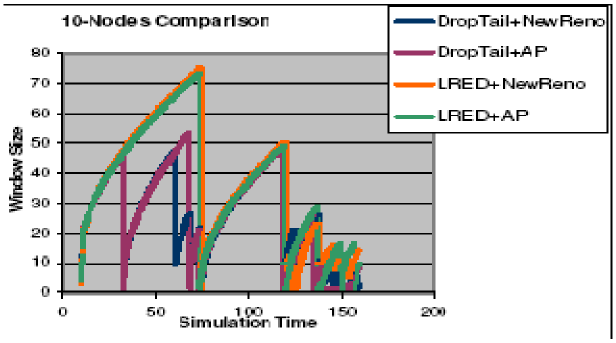


Fig. 3 Window Size for TCP with AP and LRED forNetwork Scenario 2

For the Network Scenario 2, at the beginning, the nodes are placed in random position and then moving with different speeds to different position. When multiple TCP connections are provided, window size is low only. When single TCP connection is provided, according to the number of hops between the source and destination, the performance is varied. Whenever multiple hops are provided, the window size is increased. Between 250s and 325s, nodes 0 and 1 are having multiple nodes in between them. Like that nodes 5 and 12 are having multiple hops during the time 130s to 470s. And also nodes 8 and 15 are having multiple hops during 140s to 345s. So that in those cases, the window size in increasing up to the maximum of 100. The window evolution for the above mentioned cases is given in Figure. (4).

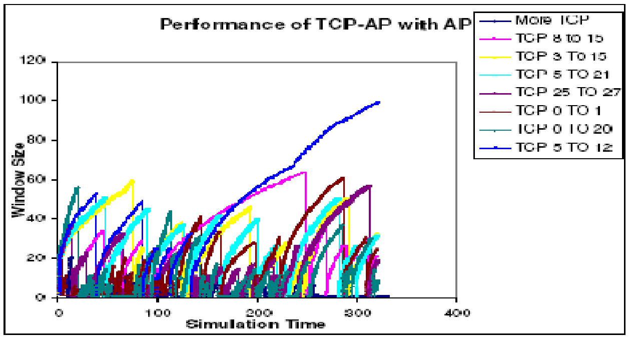


Figure. 4 Window Size for TCP with AP and LRED for Network Scenario 3

For the second scenario, the throughput, delay and jitter had been calculated under various combinations of TCP versions and between various nodes. That comparison is shown in Tables 1 and 2.

	NewReno+DropTail	**NewReno+LRED**	**AP+LRED**
flowID	0	0	0
flowType	tcp	tcp	tcp
srcNode	5	5	5
destNode	12	12	12
startTime	1	1	1
stopTime	499	499	499
receivedPkts	7959	6319	12500
avgTput[kbps]	127.632	101.341	200.517
avgDelay[ms]	43.5882	67.7284	27.6009

Table 1 Comparison on various versions of TCP

Nodes	**5 to 12**	**0 to 1**	**0 to 20**	**3 to 15**	**5 to 21**	**8 to 15**	**25 to 27**
flowID	0	0	0	0	0	0	0
flowType	tcp	tcp	tcp	tcp	tcp	tcp	tcp
srcNode	5	0	0	3	5	8	25
destNode	12	1	20	15	21	15	27
startTime	1	1	1	1	1	1	1
stopTime	499	499	499	499	499	499	499
receivedPkts	12501	25820	27824	34790	19531	18928	27010
avgTput[kbps]	212.51	438.77	472.83	591.21	331.9	321.7	459.03
avgDelay[ms]	27.603	23.684	20.966	13.353	35.17	44.847	24.419

Table 2 Performance over various source and destination pairs

6 Conclusion

TCP is a natural choice for reliable data delivery in these scenarios. This work systematically studies the impact of node mobility on TCP performance in wireless networks. Here when the destination node moved due to mobility, that impacts the performance of TCP. To gain more insight, proposed the two link layer techniques, LRED and Adaptive Pacing, which improve the window size of TCP flows by much better. The throughput is increased and delay is decreased when there is multiple hops. The performance of TCP-AP and LRED are better when the source and destination nodes are having multiple hops. Whereas when the nodes are close enough, the performance will be low compared to the previous case. The combination of TCP-AP and LRED is producing much better result than as a single one. TCP-AP and LRED are the techniques for solving Hidden and Exposed terminals in wireless networks. In addition to that channel errors and Energy Bandwidth tradeoff of the nodes have to be considered that is going to be implemented in Enhanced LRED.

References

[1] Z. Fu, P.Zerfos, H. Luo, S. Lu, L. Zhang, M. Gerla, “The Impact of Multihop Wireless Channel on TCP Throughput and Loss”, IEEE INFOCOM'03, San Francisco , March 2003 Volume: 3, On page(s): 1744- 1753

[2] A. Antony Franklin, B. Venkata Ramana and C. Siva Ram Murthy “A Link Layer Adaptive Pacing Scheme for Improving Performance of Wireless Mesh Networks”

[3] Xiang Chen, Hongqiang Zhai, Jianfeng Wang and Yuguang Fang, “A Survey on Improving TCP Performance over Wireless Networks” ACM Computing Surveys (CSUR) archive Volume 34 , Issue 3 (September 2002) Pages: 357 - 374 Year of Publication: 2002; DOI 10.1007/0-387-23808-5_23

[4] G. Holland and N. Vaidya, “Analysis of TCP Performance over Mobile Ad Hoc Networks,” ACM MOBICOM 1999

[5] J. P. Monks, P. Sinha and V. Bharghavan, “Limitations of TCP-ELFN for Ad Hoc Networks,” MOMUC 2000

[6] C. Villamizar and C. Song, “High Performance TCP in Ansnet,” ACM Computer Communication Review, vol. 24, no. 5, pp. 45–60, 1994.

[7] G. Appenzeller, I. Keslassy, and N. McKeown, “Sizing Router Buffers,” in Proceedings of ACM SIGCOMM, (Portland, OR), 2004.

[8] A. Dhamdhere, H. Jiang, and C. Dovrolis, “Buffer Sizing for Congested Internet Links,” in Proceedings of IEEE INFOCOM, (Miami, FL), March 2005.

[9] D. Y. Eun and X. Wang, “Stationary Behavior of TCP/AQM with Many Flows under Aggressive Packet Marking,” in IEEE International Conference on Communication, (Seoul, Korea), 2005.

[10] S. M. ElRakabawy, A Klemm, and C. Lindemann, “TCP with Adaptive Pacing for Multihop Wireless Networks”, in Proceedings of ACM MOBIHOC, May 2005, pp. 288-299.

[11] K. Sundaresan, V. Anantharaman, H.-Y. Hsieh, and R. Sivakumar, “ATP: A Reliable Transport Protocol for Ad-hoc Networks”, in Proceedings of ACM MOBIHOC, June 2003, pp. 64-75.

[12] K. Xu, M. Gerla, L. Qi, and Y. Shu, “Enhancing TCP Fairness in Ad Hoc Wireless Networks Using Neighborhood RED”, in Proceedings of ACM MOBICOM, September 2003, pp. 16-28.

[13] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, “A Comparison of Mechanisms for Improving TCP Performance Over Wireless Links”, IEEE/ACM Transactions on Networking, vol. 5, no. 6, pp. 756-769, 1997.

[14] K. Chen, K. Nahrstedt, and N. Vaidya, “The Utility of Explicit Rate-Based Flow Control in Mobile Ad Hoc Networks”, Proc. IEEE Wireless Communications and Networking Conference (WCNC 04), Atlanta, GA, 2004. ISBN: 0-7803-8344-3

[15] R. Yavatkar and N. Bhagwat “Improving End-to-End Performance of TCP over Mobile Internetworks” Mobile Computing Systems and Applications, Proceedings., Publication Date: 8-9 Dec 1994 On page(s): 146-152; Digital Object Identifier: 10.1109/MCSA.1994.513474

[16] E. Altman and T. Jimenez, “Novel Delayed ACK Techniques for Improving TCP Performance in Multihop Wireless Networks”, Proc. Personal Wireless Communications (PWC 03), Venice, Italy, 2003. DOI 10.1007/b12004

[17] X. Yu, “Improving TCP Performance over Mobile Ad Hoc Networks by Exploiting Cross-Layer Information Awareness”, Proc. ACM MOBICOM 04, Philadelphia, PA, 2004. DOI: 10.1109/CJECE.2004.1425806