

Impact of Multihop Wireless Channel on TCP-AP with LRED

G. Sankara Malliga,
Senior Lecturer/ECE,
Karpagam college of engineering,
Coimbatore, Tamilnadu, India.

Dr.Dharmishtan K Varughese
Professor/ECE
Karpagam college of engineering,
Coimbatore, Tamilnadu, India.

ABSTRACT

The 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 Ad-hoc Networks can lead to TCP implementations with poor performance. In a wireless network, however packet losses will 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 and 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 over multihop transmission than the single hop 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 two different network scenarios.

Keywords

Wireless Networks, TCP, Congestion Control, Adaptive Pacing, Link Random Early Discard

1. INTRODUCTION

Wireless networks are becoming very popular and are being installed almost everywhere. Reliable transport protocols such as TCP are tuned to perform well in traditional networks where packet losses occur mostly because of congestion. [1] However, networks with wireless and other lossy links also suffer from significant losses due to bit errors and handoffs. TCP responds to all losses by invoking congestion control and avoidance algorithms, resulting in degraded end-to-end performance in wireless and lossy systems [2]. As a result, many modifications and new solutions have been proposed to improve TCP's performance, such as forward error correction schemes, retransmissions at the link layer, split connections like MTCP(Mobile TCP), Explicit Loss Notification, link layer TCP Aware like Snoop, Performance Enhancing Proxies, Indirect TCP (I-TCP), MAITE(Mobility Awareness Incorporated as TCP Enhancement), etc. Available performance evaluations of TCP over wireless networks are usually incomplete, meaning that the most important TCP versions and the most important solutions are not studied and compared all together [1]. Therefore, we still don't have a good idea about what is the best combination.

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 [9]. However, it is unclear how well such TCP mechanisms work in a multihop wireless network; this is the focus of this work.

Over the past few years, the problem of congestion control has received wide spread attention, both in the Internet context as well as in an ad-hoc network context. Most of this research has focused on modeling, analysis, algorithm development of end-to-end control schemes (such as TCP), and adaptation of such schemes to ad-hoc networks. Given routing path and bandwidth constraints, algorithms have been developed which converge and have a stable operation. Unfortunately, when packets are lost in networks for reasons other than congestion, these measures result in an unnecessary reduction in end-to-end throughput and hence, sub-optimal performance. Communication over wireless links is often characterized by sporadic high bit-error rates, and intermittent connectivity due to handoffs. TCP performance in such networks suffers from significant throughput degradation and very high interactive delays [3].

Recently, several schemes have been proposed to alleviate the effects of non-congestion-related losses on TCP performance over networks that have wireless or similar high loss links [2], [3], [5]. These schemes choose from a variety of mechanisms, such as local retransmissions, split-TCP connections, and forward error correction, to improve end-to-end throughput. However, it is unclear to what extent each of the mechanisms contributes to the improvement in performance.

There are two different approaches for improving TCP performance in such lossy systems. The first approach hides any non-congestion-related losses from the TCP sender and therefore requires no changes to existing sender implementations. The intuition behind this approach is that since the problem is local, it should be solved locally, and that the transport layer need not be aware of the characteristics of the individual links. Protocols that adopt this approach attempt to make the lossy link appear as a higher quality link with a reduced effective bandwidth. As a result, most of the losses seen by the TCP sender are caused by congestion.

At the other end of the solution spectrum, split-connection approaches completely hide the wireless link from the sender by terminating the TCP connection at the base station. Such schemes use a separate reliable connection between the base station and the destination host. The second connection can use techniques such as negative or selective acknowledgments, rather than just standard TCP, to perform well over the wireless link. The third class of protocols, link-layer solutions, lie between the other two classes. These protocols attempt to hide link related losses from the TCP sender by using local retransmissions and perhaps forward error correction over the wireless link. The local retransmissions use techniques that are tuned to the characteristics of the wireless link to provide a significant increase in performance. Since the end-to-end TCP connection passes through the lossy link, the TCP sender may not be fully shielded from wireless losses. This can happen either because of timer interactions between the two layers [16], or more likely because of TCP's

duplicate acknowledgments causing sender fast retransmissions even for segments that are locally retransmitted.

In TCP, reliability is achieved by retransmitting lost packets. Thus, each TCP sender maintains a running average of the estimated round trip delay and the average deviation derived from it. Packets will be retransmitted if the sender receives no acknowledgment (ACK) within a certain timeout interval (e.g., the sum of smoothed round trip delay and four times the average deviation) or receives duplicate acknowledgments. Due to the inherent reliability of wired networks, there is an implicit assumption made by TCP that any packet loss is due to congestion[3]. To reduce congestion, TCP invokes its congestion control mechanisms whenever any packet loss is detected.

Consider the problem of congestion control over wireless multi-hop networks. Nodes in such networks are radio equipped, and communicate by broadcasting over wireless links. Communication paths between nodes which are not in radio range of each other are established by intermediate nodes acting as relays to forward data toward the destination. The diverse applications of such networks range from community based roof-top networks to large-scale ad-hoc networks.

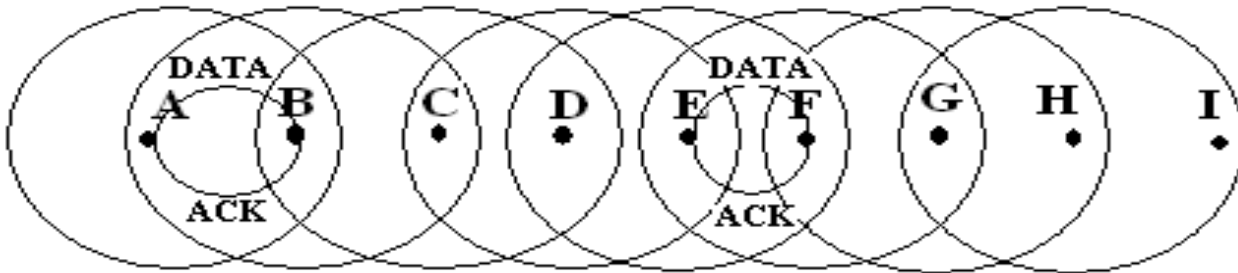


Figure 1. Spatial reuse and contention. An example of 8 hop chain. Optimal spatial reuse is achieved when nodes {A, D, G} and nodes {B, C, E} are scheduled for transmission alternatively. Node D is the hidden terminal for transmission A→B.

Even though the RTS/CTS handshake is employed, hidden terminal problem still persists in an IEEE 802.11 ad-hoc network. A hidden terminal is a potential sending node in the receiver's neighbourhood, which cannot detect the sender and may disrupt the current packet transmission. Consider Figure 1 as an example, in which two adjacent nodes are about 200m apart. The current hardware specifies that for each wireless node, its transmission range is about 250m, its carrier sensing range is 550m, and its interference range is about 550m. The potential sending node D is a hidden terminal of the current transmission pair A – B. When A and B are initiating RTS-CTS handshake, D cannot hear CTS since it is out of the 250m transmission range of node B. Besides, D cannot sense A's DATA transmission since A is out of D's 550m carrier sensing range. Therefore, D may transmit to its intended receiver E at any time. When D is transmitting to E, it will cause collisions at B, since D is within the 550m interference range for B. Therefore, hidden terminal D will cause contention loss at node B. Location-dependent contention, together with multi-hop packet forwarding, also allows for spatial channel reuse.

Specifically, any two transmissions that are not interfering with each other can potentially occur simultaneously; this improves aggregate channel utilization. Figure (1) illustrates an example for spatial reuse, in which pairs of A-B and E-F may transmit simultaneously, but simultaneous transmissions from pairs of A-B and C-D will collide. Improving spatial reuse will result in increased TCP throughput.

2. BACKGROUND

Consider a static, multi hop, wireless ad-hoc network. A single wireless channel is shared for transmissions, and only receives within the transmission range of the sender can receive the packets. The IEEE 802.11 Distributed Coordination Function, the de facto access method used in ad-hoc networks, serves as the wireless MAC protocol. In IEEE 802.11, each packet transmission is preceded by a control handshake of RTS/CTS(Request To Send/Clear To Send) messages. Upon overhearing the handshake, the nodes in the neighbourhood of both the sender and the receiver defer their transmissions until the subsequent DATA-ACK transmissions are completed [4].

Failures in the transmission of control and data packets are usually caused either by the effect of the Hidden Terminal, or by channel errors. [9] Specifically, the sender drops the DATA packet after sending the RTS message seven times and does not receive a CTS from the receiver. DATA packets are dropped after four retransmissions without receiving an ACK.

3. OVERVIEW OF TCP

Before diving into the detailed discussion of questions such as *why TCP performs poorly in wireless networks* and *how TCP performance can be improved*, it is necessary to prepare by presenting an overview of not only the basic functionality of TCP but also the state-of-the-art in TCP. [4] The basic functions of TCP as a transport layer protocol include flow control, error recovery and congestion control, while the state-of-the-art techniques include fast retransmission and recovery, selective acknowledgment, etc., mainly focusing on how to promptly and effectively respond to network congestion.

It is well known that TCP is a connection-oriented transport protocol that is aimed at guaranteeing end-to-end reliable ordered delivery of data packets over wired networks. For this purpose, basic functionalities such as flow control, error control, and congestion control are indispensable. While these functions have a clean-cut definition of their own, in practice they are closely coupled with one another in TCP implementation. It turns out that in wired networks, almost all the packet losses are due to network congestion rather than transmission errors. Thus, in addition to retransmission, TCP responds to packet losses by invoking its congestion control mechanism.

4. TCP'S CHALLENGES IN AD-HOC NETWORKS

The performance of TCP degrades in Ad-hoc networks. This is because TCP has to face new challenges due to several reasons specific to these networks: lossy channels, hidden and exposed stations, path asymmetry, network partitions, route failures, and power constraints [4].

4.1. Channel Errors

In wireless channels, relatively high bit error rate because of multipath fading and shadowing may corrupt packets in transmission, leading to the losses of TCP data segments or ACKs. If it cannot receive the ACK within the retransmission timeout, the TCP sender immediately reduces its congestion window to one segment, exponentially backs off its Retransmission Time-Out (RTO) and retransmits the lost packets. Intermittent channel errors may thus cause the congestion window size at the sender to remain small, thereby resulting in low TCP throughput.

4.2. Mobility

Cellular networks are characterized by handoffs due to user mobility. Normally, handoffs may cause temporary disconnections, resulting in packet losses and delay. TCP will suffer a lot if it treats such losses as congestion and invokes unnecessary congestion control mechanisms. The handoffs are expected to be more frequent in next generation cellular networks as the micro-cellular structure is adopted to accommodate an increasing number of users. Thing could be worse if TCP cannot handle handoffs gracefully. Similar problems may occur in wireless LAN, as mobile users will also encounter communication interruptions if they move to the edge of the transmission range of the access point.

4.3. Asymmetry

In wireless networks, the wireless link between a base station and a mobile terminal in nature is asymmetric. Compared with the base station, the mobile terminal has limited power, processing capability, and buffer space. Another asymmetry stems from the vastly different characteristics of wired links and wireless links. The former is reliable and has large bandwidth while the latter is error-prone and has limited and highly variable bandwidth. For example, the bandwidth of a typical Ethernet is 10Mbps (100Mbps or even higher for fast Ethernet) while the highest bandwidth for 3G networks is only about 2Mbps. Therefore, the wireless link is very likely to become the bottleneck of TCP connections.

4.4. Lossy channels

The main causes of errors in wireless channel are the following:

Signal attenuation: This is due to a decrease in the intensity of the electromagnetic energy at the receiver (e.g. due to long distance), which leads to low signal-to-noise ratio (SNR).

Doppler shift: This is due to the relative velocities of the transmitter and the receiver. Doppler shift causes frequency shifts in the arriving signal, thereby complicating the successful reception of the signal.

Multipath fading: Electromagnetic waves reflecting off objects or diffracting around objects can result in the signal traveling over multiple paths from the transmitter to the receiver. Multipath propagation can lead to fluctuations in the amplitude, phase, and geographical angle of the signal received at a receiver.

In order to increase the success of transmissions, link layer protocols implement the following techniques: Automatic Repeat

reQuest (ARQ), or Forward Error Correction (FEC), or both. For example, IEEE 802.11 implements ARQ, so when a transmitter detects an error, it will retransmit the frame, error detection is timer based. Bluetooth implements both ARQ and FEC on some synchronous and asynchronous connections.

Note that packets transmitted over a fading channel may cause routing protocol to incorrectly conclude that there is a new one-hop neighbor. This one-hop neighbor could provide a shorter route to even more distant nodes. Unfortunately, this new shorter route is usually unreliable.

4.5. Hidden and Exposed stations

In Ad-hoc networks, stations may rely on physical carrier-sensing mechanism to determine idle channel, such as in the IEEE 802.11 DCF function. This sensing mechanism does not solve completely the hidden station and the exposed station problems. Before explaining these problems, we need to clarify the "transmission range" term.

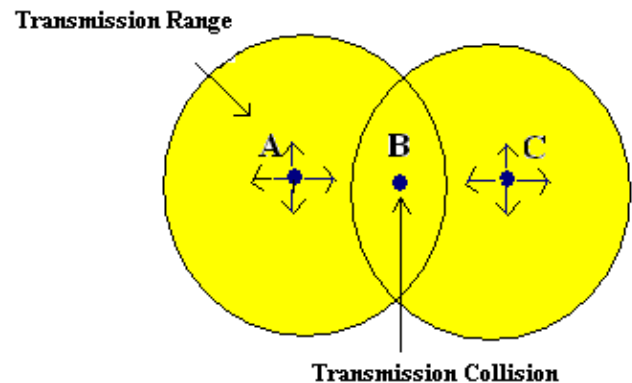


Figure 2. Hidden terminal problem: Packets sent to B by A and C will collide at B.

The transmission range is the range, with respect to the transmitting station, within which a transmitted packet can be successfully received. A typical hidden terminal situation is depicted in Figure (2). Stations A and C have a frame to transmit to station B. Station A cannot detect C's transmission because it is outside the transmission range of C. Station C (resp. A) is therefore "hidden" to station A (resp. C). Since A and C transmission areas are not disjoint, there will be packet collisions at B. These collisions make the transmission from A and C toward B problematic. To alleviate the hidden station problem, virtual carrier sensing has been introduced. It is based on a two-way handshaking that precedes data transmission. Specifically, the source station transmits a short control frame, called Request-To-Send (RTS), to the destination station. Upon receiving the RTS frame, the destination station replies by a Clear-To-Send (CTS) frame, indicating that it is ready to receive the data frame. Both RTS and CTS frames contain the total duration of the data transmission. All stations receiving either RTS or CTS will keep silent during the data transmission period (e.g. station C in Figure (2)).

However, the hidden station problem may persist in IEEE 802.11 Ad-hoc networks even with the use of the RTS/CTS handshake. This is due to the fact that the power needed for interrupting a packet reception is much lower than that of delivering a packet successfully. In other words, node's transmission range is smaller than the sensing node range. The exposed station problem results from a situation where a transmission has to be delayed because of the transmission

between two other stations within the sender's transmission range. In Figure (3), we show a typical scenario where the exposed terminal problem occurs. Let us assume that A and C are within B's transmission range, and A is outside C's transmission range. Let us also assume that B is transmitting to A, and C has a frame to be transmitted to D. According to the carrier sense mechanism, C senses a busy channel because of B's transmission.

Therefore, station C will refrain from transmitting to D, although this transmission would not cause interference at A. The exposed station problem may thus result in a reduction of channel utilization. It is worth noting that hidden terminal and exposed terminal problems are correlated with the transmission range. By increasing the transmission range, the hidden terminal problem occurs less frequently. On the other hand, the exposed terminal problem becomes more important as the transmission range identifies the area affected by a single transmission.

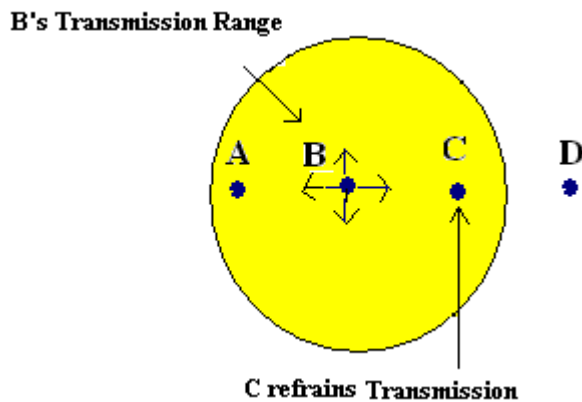


Figure 3. Exposed terminal problem: Because of B's transmission C refrains transmission to D.

4.6. Path asymmetry

Path asymmetry in Ad-hoc networks may appear in several forms like bandwidth asymmetry, loss rate asymmetry and route asymmetry.

Bandwidth asymmetry: Satellite networks suffer from high bandwidth asymmetry, resulting from various engineering tradeoffs (such as power, mass, and volume), as well as the fact that for space scientific missions, most of the data originates at the satellite and flows to the earth. The return link is not used, in general, for data transferring. For example, in broadcast satellite networks the ratio of the bandwidth of the satellite-earth link over the bandwidth of the earth-satellite link is about 1000. On the other hand in Ad-hoc networks, the degree of bandwidth asymmetry is not very high. For example, the bandwidth ratio lies between 2 and 54 in Ad-hoc networks that implement the IEEE 802.11 version g protocol. The asymmetry results from the use of different transmission rates. Because of this different transmission rates, even symmetric source destination paths may suffer from bandwidth asymmetry.

Loss rate asymmetry: This type of asymmetry takes place when the backward path is significantly more lossy than the forward path. In Ad-hoc networks, this asymmetry is due to the fact that packet losses depend on local constraints that can vary from place to place. Note that loss rate asymmetry may produce bandwidth asymmetry. For example, in multi-rate IEEE 802.11 protocol versions, senders may use the Auto-Rate-Fallback (ARF) algorithm for transmission rate selection. With ARF, senders attempt to use higher transmission rates after consecutive transmission successes, and revert to lower rates after failures. So,

as the loss rate increases the sender will keep using low transmission rates.

Route asymmetry: Unlike the previous two forms of asymmetry, where the forward path and the backward path can be the same, route asymmetry implies that distinct paths are used for TCP data and TCP ACKs. This asymmetry may be artifact of the routing protocol used. Route asymmetry increases routing overheads and packet losses in case of high degree of mobility. Because when nodes move, using a distinct forward and reverse routes increases the probability of route failures experienced by TCP connections. However, this is not the case of static networks or networks that have low degree of mobility, like the case of a network with routes of high lifetime compared to the session transfer time. So, it is up to the routing protocols to select symmetric paths when such routes are available in the case of Ad-hoc networks of high mobility.

In the context of satellite networks, there has been a lot of research on how to improve TCP performance. But since satellite networks are out of the scope of the report, we will limit ourselves to list three techniques introduced by these proposals, which we believe might be useful in Ad-hoc networks.

The first one is "TCP header compression" that reduces the size of the TCP ACKs on the backward path. The second one is "ACK filtering" that reduces the number of TCP ACKs transmitted, by taking advantage of the fact that TCP ACKs are cumulative. The third one is "ACK congestion control" that let the receiver also control the congestion on the backward path. This is done by dynamically maintaining a delayed-ACK factor d by the receiver, and by sending one ACK for every d data packet received. The difference between ACK filtering and ACK congestion control is that the first one is a link layer technique that can be implemented at intermediate nodes, however the second one is a TCP layer technique that is implemented at the TCP sink. Unfortunately, these techniques alone cause problems such as increasing sender's burst traffic and also slowing down the sender's congestion window growth. So, it is necessary to adapt the sender congestion control algorithm to avoid these problems.

The adaptive delayed-ACK proposed in [1] aims to reduce the contention on the channel, by reducing the number of TCP ACKs transmitted. This proposal also alleviates the asymmetry problem in SANETs. We have not found any other proposal dealing with the asymmetry problem in Ad-hoc networks.

4.7. Network partition

An Ad-hoc network can be represented by a simple graph G . Mobile stations are the "vertices". A successful transmission between two stations is an undirected "edge". Network partition happens when G is disconnected. The main reason of this disconnection in MANETs is node mobility. Another factor that can lead to network partition is energy constrained operation of nodes. An example of network partition illustrated in Figure 4. In this figure, dashed lines are the links between nodes. When node D moves away from node C this results in a partition of the network into two separate components. Clearly, the TCP agent of node A can not receive the TCP ACK transmitted by node F. If the disconnectivity persists for a duration greater than the retransmission timeout (RTO) of node A, the TCP agent will trigger the exponential backoff algorithm [3], which consists of doubling the RTO whenever the timeout expires. Originally, TCP does not have indication about the exact time of network reconnection. This lack of indication may lead to long idle periods during which the network is connected again, but TCP is still in the backoff state.

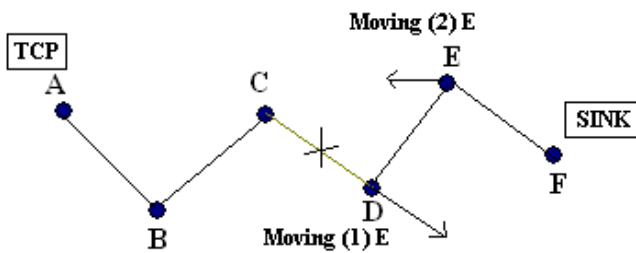


Figure 4. Network partition scenario: When D is moving away from C. The network is reconnected when E is moving toward C.

4.8. Routing failures

In wired networks route failures occur very rarely. In MANETs they are frequent events. The main cause of route failures is node mobility. Another factor that can lead to route failures is the link failures due to the contention on the wireless channel, which is the main cause of TCP performance degradation. The route reestablishment duration after route failure in Ad-hoc networks depends on the underlying routing protocol, mobility pattern of mobile nodes, and traffic characteristics. As already discussed in Section II-D, if TCP sender's does not have indications on the route re-establishment event, the throughput and session delay will degrade because of the large idle time. Also, if the new route established is longer or shorter, in term of hops, than the old route TCP will face a brutal fluctuation in Round Trip Time (RTT).

In addition, in Ad-hoc networks, routing protocols that rely on broadcast Hello messages to detect neighbors' reachability, may suffer from the "communication gray zones" problem. In these zones data messages cannot be exchanged although broadcast Hello messages and control frames indicate that neighbors are reachable. So on sending a data messages, routing protocols will experience routing failures. In Lundgren et al. have conducted experiments and have subsequently concluded that the origin of this problem is heterogeneous transmission rates, absence of acknowledgment for broadcast packets, small packet size of Hello messages, and fluctuations of wireless links.

4.9. Power constraints

Because batteries carried by each mobile node have limited power supply, processing power is limited. This is a major issue in Ad-hoc networks, as each node is acting as an end system and as a router at the same time, with the implication that additional energy is required to forward and relay packets. TCP must use this scarce power resource in an "efficient" manner. Here, efficiency means minimizing the number of unnecessary retransmissions at the transport layer as well as at the link layer². In general, in Ad-hoc networks there are two correlated power problems: the first one is "power saving" that aims at reducing the power consumption; the second one is "power control" that aims at adjusting the transmission power of mobile nodes. Power saving strategies have been investigated at several levels of a mobile device including the physical layer transmissions, the operation systems, and the applications. Power control can be jointly used with routing or transport agents to improve the performance of Ad-hoc networks power constraints communications reveal also the problem of cooperation between nodes, as nodes may not participate in routing and forwarding procedures in order to save battery power.

5. CONGESTION CONTROL

Congestion control is a distributed algorithm to share network resources among competing users. It is important in situations where the availability of resources and the set of competing users vary over time unpredictably, yet efficient sharing is desired. These constraints, unpredictable supply and demand and efficient operation, necessarily lead to feedback control as the preferred approach, where traffic sources dynamically adapt their rates to congestion in their paths. On the Internet, this is performed by the Transmission Control Protocol (TCP) in source and destination computers involved in data transfers.

First, queuing delay can be more accurately estimated than loss probability both because packet losses in networks with large bandwidth-delay product are rare events (probability on the order 10^{-8} or smaller), and because loss samples provide coarser information than queuing delay samples. Indeed, measurements of delay are noisy, just as those of loss probability. Each measurement of packet loss (whether a packet is lost) provides one bit of information for the filtering of noise, whereas each measurement of queuing delay provides multi-bit information. This makes it easier for an equation-based implementation to stabilize a network into a steady state with a target fairness and high utilization. Second, the dynamics of queuing delay seems to have the right scaling with respect to network capacity. This helps to maintain stability when network scales up in capacity.

6. PROBLEMS AND RELATED STUDIES

Fu et al. [9] 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. [16] 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.

Sundaresam et al. [17] and Chen et al. [18] 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 [17], 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. [18] 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 [5], [17], 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, TCP-AP can be incrementally deployed, since TCP-AP is not only TCP-friendly, but also TCP compatible.

Altman and Jiménez [19] proposed a dynamic scheme for delaying ACKs in order to improve TCP throughput in multihop wireless networks. Using simulation, they showed that for an h-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. Holland and Vaidya [21] introduced explicit link failure notification (ELFN) as a feedback mechanism from the network in order to help TCP, distinguish congestion losses and losses induced by link failures. Yu [20] proposed two cross-layer communication mechanisms that further improve TCP performance in case of packet losses due to mobility. This work focuses on TCP performance under two different wireless network scenarios. Results may well be utilized together with their findings in order to adapt TCP-AP for mobile wireless multihop networks.

7. IMPROVING TCP PERFORMANCE

This section describes two techniques to improve TCP performance over multihop wireless networks[4]. 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%.

7.1. 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

```

1: if avg_retry < min_th then
2:  mark_prob ← 0
3:  pacing ← ON
4: else
5: mark_prob = min{  $\frac{avg\_retry - min\_th}{max\_th - min\_th}$ , max_p }
6: set pacing OFF
7: end if
8: mark p with mark_prob
9: MacLayerSend(p,pacing)
10: retry = GetMacRetries()
11:  $avg\_retry = \frac{7}{8} avg\_retry + \frac{1}{8} retry$ 

```

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 inter-flow fairness.

7.2. 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

```

1: if received ACK then
2:  random_Backoff ← ran_backoff(cong_win)
   {DATA transmission succeeded. Setup the
   backoff timer}
3: if pacing is ON then
4:  extra_Backoff = TX_Timed(DATA) + overhead
5: end if
6: backoff ← random_Backoff + extra_Backoff
7: start backoff_timer
8: 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.

8. SIMULATION MODEL AND RESULTS

Here ns-2 network simulator with the CMU extensions for IEEE 802.11 wireless LAN is used. Here, the two network scenarios are considered. In one scenario, that consists of 3 nodes and in another scenario, that consists of 10 nodes. The routing protocol chosen for analysis is AODV. The analysis done in four cases such 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.

8.1. Simulation scenario I

This scenario runs a single TCP connection having a 2-nodes network over an area of a size of 500m and 400m. The initial locations of nodes 0 and 1 are (5,2) and (390,385). At time 10 sec, node 0 starts moving towards (20,18) at a speed of 1m/sec. At time 50 sec, node 1 starts moving towards (25,20) at a speed of 15m/sec. The network chosen for analysis is shown in Figure. [5]. The simulation lasts at 150 sec.

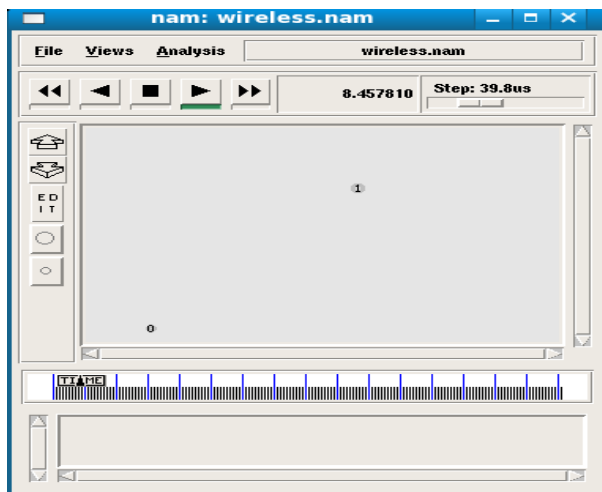


Figure. 5 Simulation Scenario 1 at Simulation Time nearly 10 sec

8.2. Simulation scenario II

This scenario also 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. [6]. 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.

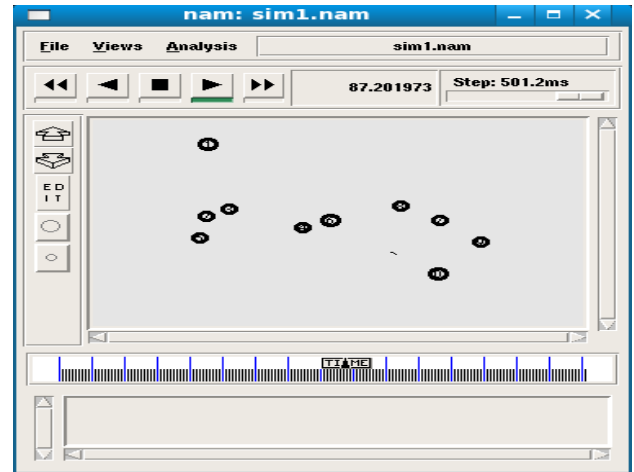


Figure. 6 Simulation Scenario 2 at Simulation Time nearly 87 sec

8.3. Analysis of simulation results

For the Network Scenario 1, at the beginning, the nodes are too far away and a connection cannot be set. The first TCP signaling packet is transmitted at time about 40 sec. After 150 sec also node 1 had moved and far away so that transmission can not take place. At the time during 40 sec to 150 sec, node 0 and 1 are able to initiate TCP connection between node 0 and node 1. During that time only performance is good and window size increases and reaches maximum. All other time the window size will be nearly one. This is due to the mobility nature of the node. The AODV routing protocol is creating an alternative route for transmission. The window evolution for the above mentioned cases is given in Figure. [7].

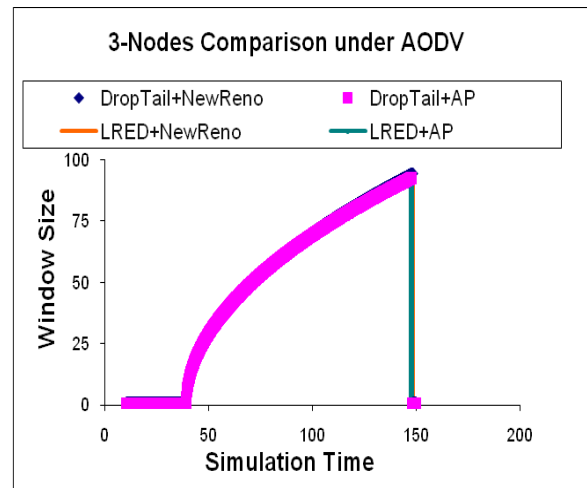


Figure. 7 Window Size for TCP with AP and LRED for Network Scenario 1

For the Network Scenario 2, during the time 10 sec to 75sec, there will be multiple hops 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. [8].

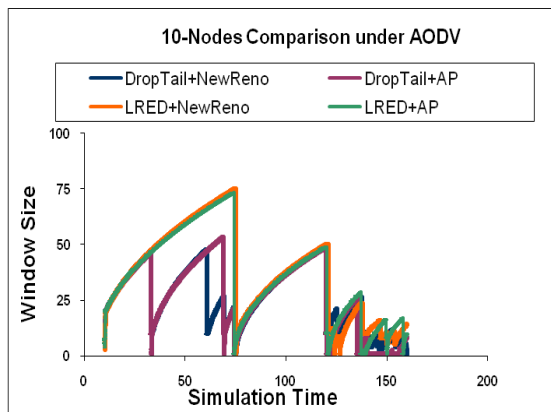


Figure. 8 Window Size for TCP with AP and LRED for Network Scenario 2

9. 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. In this work only the window size is taken as the parameter for analysis. In future, Throughput, queuing delay and packet loss have to be considered. 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. 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.

10. REFERENCES

- [1] Sarma Vangala and Miguel A. Labrador “Performance of TCP over Wireless Networks with the Snoop Protocol” 27th Annual IEEE Conference on Volume , Issue , 6-8 Nov. 2002 Page(s): 600 – 601; DOI: 10.1109/LCN.2002.1181836
- [2] Hari Balakrishnan, Venkata N. Padmanabhan, Srinivasan Seshan and Randy H. Katz “A Comparison of Mechanisms for Improving TCP Performance over Wireless Links” ACM SIGCOMM Computer Communication Review Volume 26 , Issue 4 Pages: 256 - 269 Year of Publication: 1996 ISSN:0146-4833
- [3] R. Caceres and L. Iftode “Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments” IEEE Journal on Selected Areas in Communications, Volume 13, Issue 5, Jun 1995 Page(s):850 - 857. Digital Object Identifier: 10.1109/49.391749
- [4] 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
- [5] R. Yavatkar and N. Bhagwat “Improving End-to-End Performance of TCP over Mobile Internetworks” Mobile Computing Systems and Applications, 1994. Proceedings., Publication Date: 8-9 Dec 1994 On page(s): 146-152; Digital Object Identifier: 10.1109/MCSA.1994.513474
- [6] V. Bharghavan, “Performance Analysis of a Medium Access Protocol for Wireless Packet Networks,” Wireless Networks Volume 10 , Issue 5 (September 2004) Pages: 519 - 529 Year of Publication: 2004 ISSN:1022-0038
- [7] Ibtissam El Khayat, Pierre Geurts, and Guy Leduc, “Improving TCP in wireless networks with an adaptive machine-learned classifier of packet loss causes” Proceedings of the International Conference on Networking (Networking 2005) - May 2005 Pages 549-560 <http://www.montefiore.ulg.ac.be/services/stochastic/pubs/2005/EGL05>, DOI: 10.1007/b136094
- [8] M. Gerla, R. Bagrodia, L. Zhang, K. Tang, and L. Wang, “TCP over Wireless Multihop Protocols: Simulation and Experiments,” Proceedings of IEEE ICC'99, Vancouver, Canada, Jun. 1999 Volume: 2, On page(s): 1089-1094. Digital Object Identifier: 10.1109/ICC.1999.765450
- [9] 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
- [10] I. Ali, R. Gupta, S. Bansal, A. Misra, A. Razdan and R. Shorey, “Energy Efficiency and Throughput for TCP Traffic in Multi-Hop Wireless Networks,” IEEE INFOCOM'02, New York, 2002 Volume: 1, On page(s): 210- 219, DOI: 10.1109/INFCOM.2002.101926
- [11] Xiang Chen, Hongqiang Zhai, Jianfeng Wang, and Yuguang Fang, “TCP performance over mobile ad-hoc networks”, Proceedings of the 10th annual international conference on Mobile computing and networking Philadelphia, PA, USA SESSION: Ad-hoc networks Pages: 231 - 244 Year of Publication: 2004 ISBN:1-58113-868-7
- [12] Vicente E. Mujica V., Dorgham Sisalem Radu Popescu-Zeletin and Adam Wolisz, “TCP-Friendly Congestion Control over Wireless Networks”, In Proc. Of European Wireless 2004, Barcelona Spain, February 2004. Digital Object Identifier: 10.1049/iet-com:20070284
- [13] M. Gerla, K. Tang, and R. Bagrodia, “TCP Performance in Wireless Multihop Networks,” Proceedings of IEEE WMCSA'99, New Orleans, LA, Feb. 1999 Page: 41 ISBN:0-7695-0025-0
- [14] Saad Biaz Dai Yawen, “Enhancing Congestion Control for Wireless Links” ACM SIGCOMM' 2003 DOI: ieeecomputersociety.org/10.1109/ISCC.2004.1358551
- [15] A. DeSimone, M. C. Chuah, and O. C. Yue. “Throughput Performance of Transport-Layer Protocols over Wireless LANs” In *Proc. Globecom '93*, December 1993 Volume 1, 29 Nov-2 Dec 1993 Page(s):542 – 549; Digital Object Identifier: 10.1109/GLOCOM.1993.318140

- [16] K. Xu, M. Gerla, L. Qi, and Y. Shu, "Enhancing TCP Fairness in Ad Hoc Wireless Networks using Neighborhood RED", Proc. ACM MOBICOM 03, San Diego CA, 2003. ISBN:1-58113-753-2
- [17] K. Sundaresan, V. Anantharaman, H.-Y. Hsieh, and R. Sivakumar, "ATP: A Reliable Transport Protocol for Ad Hoc Networks", Proc. ACM MobiHoc, Annapolis, MA, 2003. ISBN:1-58113-684-6
- [18] 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
- [19] 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
- [20] 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
- [21] H. Holland and N. Vaidya, "Analysis of TCP Performance over Mobile Ad Hoc Networks", Proc. ACM MOBICOM 99, Seattle, WA, 1999. ISBN:1-58113-142-9