Pakistan Journal of Information and Technology 1(2): 80-88, 2002

Enhanced Aggressive Error Recovery for TCP Noor

[1]Emad A. Qaddoura, [2]Bassam El-Asir and [1]Musbah M. Aqel
[1]Faculty of Computer Science and Information Technology, Applied Science University, Amman, Jordan
[2]Department of Electrical Engineering, Jordan University of Science Technology, Irbid, Jordan

Abstract: The congestion control mechanism in TCP introduces idle transmit periods as a result of its small transmission window and exponential retransmission timer. This behavior occurs while recovering from packet losses, due to wireless LAN link errors. Therefore, TCP suffers from low throughput due to poor utilization of the available bandwidth, while recovering from high bit error rates. TCP Noor's Aggressive Error Recovery (AER) avoids the idle transmit periods while recovering from high Bit Error Rates (BER). Enhanced AER (EAER) applies further enhancements to AER which provides higher averaged throughput during variable bit error rates on the channel. EAER applies a dynamic mechanism, which allows TCP's transmission behavior to adjust based on the channel conditions. Simulation results indicate that TCP Noor's EAER gains up to two 250% increase in throughput over TCP Reno while recovering from variable bit error rates.

Keywords: TCP, IP, Wireless, LAN, Congestion

Introduction

The subject of improving TCP's performance over wireless links has been investigated for many years. However, the progress in making tangible performance improvements is still limited. Hence, the industry has yet to adopt a new and enhanced version of TCP. A new working group has been formed at the Internet Engineering Task Force (IETF), Performance Independent Link Characteristics (PILC), to further improve TCP's performance over different link layers including wireless links.

TCP was optimized for reliable wire line links to enhance the network performance while experiencing congestion. As a result, TCP's throughput is very poor over wireless links experiencing channel fading, due to TCP's assumption of congestion. TCP's exponential back-off process, for the retransmission timer, introduces a very conservative retransmission policy for wireless LAN environment. This results in poor throughput while experiencing packet loss. Timeouts in wireless LAN may also result from contention on the channel. Therefore, a faster retransmission policy may be used, since the contention may persist for a shorter time period than congestion in the network. TCP should distinguish between a timeout that is due to bit errors on the link, contention and network congestion.

Current TCP's control congestion mechanism responds to congestion by applying a very conservative approach in transmitting data while recovering from congestion. It is well know that packet loss over wireless links does not necessarily mean congestion in the network, but may be due to experiencing fades over the Radio Frequency (RF) channel. Usually, the RF channel is available for use once the mobile has moved out of the fading situation. However, if a packet is lost, TCP does not utilize the available bandwidth and transmits data at a very low rate. As a result, the overall throughput of TCP is reduced. For example, after a timeout, the transmission window is decreased to one packet. However, since the link's bit errors may disappear sooner than network congestion, narrowing the transmission window does impact the overall throughput for TCP. The combination of the overall conservative behavior over wireless LAN links by TCP Reno contributes to making the current congestion control mechanism a major factor in producing lower throughput than TCP Noor, and prolonged response time.

TCP's congestion control mechanism treats a timeout in the same manner regardless of the cause, (i.e. collision due to local network contention, network congestion, or bit errors). This results in a consistent conservative transmission rate and poor throughput. Local wireless LAN problems, collisions and bit errors, should be handled by a larger transmission rate. Therefore, the generic conservative behavior is not suitable for wireless LAN networks.

The initial work described in (Qaddoura *et al.*, 1999) demonstrates that TCP Noor using AER allows the recovery scheme from link losses to utilize the available bandwidth in a more aggressive manner. TCP Noor provides a larger congestion window. In addition, TCP Noor allows an exponential growth of the congestion window for every acknowledged packet. These modifications result in significant throughput improvements when the bit error rate is high.

In this paper, we extend TCP Noor's AER, now called EAER, to obtain a higher averaged throughput than TCP Reno for variable conditions during high and low bit error rates while applying exponential traffic distribution.

Our simulations indicate that slow start and congestion avoidance must be modified under such high and low bit error rates to be less conservative in transmission of packets. Despite a higher data loss in TCP Noor due to the larger congestion window it performs consistently better than TCP Reno. In some cases TCP

Noor had up to 250% better throughput than TCP Reno.

Our network is constructed of nodes on the same wireless LAN to demonstrate TCP's congestion control algorithm deficiencies under high and low bit error with variable traffic rates. FTP is the selected application, with exponential traffic generation.

TCP Noor's EAER provided a very scalable and robust algorithm. TCP Noor performed better than TCP Reno in the following cases: (i) while transitioning between the good and the bad states ii) while in a bad state (iii) while in a good state, (IV) for low traffic rate (V) for medium traffic rate, and (VI) for high traffic rate. The observed insensitivity to high BER indicates that TCP Noor is also a robust protocol.

Firstly, existing related work in this area is discussed followed by a description of current TCP's congestion control mechanism. Secondly, TCP Noor's modifications are discussed. Thirdly, the simulation model and assumptions are outlined. Finally, the simulation results, the conclusion and the suggested future work are discussed.

Related Work: Aggressive Error Recovery in TCP Noor, (Qaddoura, *et al.*, 1999), demonstrated that an aggressive transmission policy, while recovering from high bit error rates, allows for better bandwidth utilization on the wireless LAN link resulting in higher throughput under such condition. TCP Noor provides a larger congestion window. In addition, TCP Noor allows an exponential growth of the congestion window for every acknowledged packet. Therefore, significant throughput improvements where obtained when the bit error rate is high.

I-TCP (Bakre and Badrinath, 1995) recommends the use of two "split connections" for a mobile host to obtain service with a host on the fixed side of the network. One TCP connection is between the mobile host and the base station. The other one is between the base station and the fixed end host. The intention of I-TCP is twofold:(i) hide handoffs from the fixed host and transfer the connection state from the base station which is currently servicing the mobile host to the new base station which is receiving it, and (ii) shield the wire line part from the lossy nature of the wireless side. The simulation results in I-TCP provide in some cases about twice the performance of regular TCP.

A comparative analysis was done in (Balakrishnan, *et al.*, 1995) between several TCP proposals and the Snoop protocol, a TCP aware link layer solution. The authors recommend a link layer with an ability to retransmit packets locally, (between base station and mobile node), and on a much faster time scale than TCP. When packet loss is about 10% or greater, TCP aware link layer can use link level acknowledgments to avoid retransmissions at the transport layers. These retransmissions can result in performance degradation. The simulation indicated a 10-30% throughput gain due to the use of a link layer that is TCP aware, compared to other TCP versions, which use a link layer operating independently of TCP. The authors recommended the use of TCP aware link layer with selective acknowledgments to gain an optimal throughput for TCP.

Both I-TCP and Snoop are designed with the intention of minimizing the impact of wireless losses on TCP connections that are primarily wire line, only one hop in the connection is wireless and the remaining are wire line. Therefore, these solutions may not be appropriate for completely wireless networks.

The solution used in (Chandran, *et al.*, 1998), provides a feedback mechanism to avoid an exponential retransmission timer back-off, and expensive timeouts. It is geared towards mobile ad hoc networks. It avoids the invocation of the congestion control algorithm and relies on dynamic notifications between routing entities. The first notification serves as an indication to the source to refrain from sending packets due to a route failure. The second notification indicates a route restoration and the source can continue packet transmission.

A few internet drafts have been submitted to the Internet Engineering Task Force (IETF) describing the throughput benefit of changing the initial window size to a value that is higher than one. For a wire line simulation environment, (Podri and Nichols, 1998) sets the initial window size to a value of one, three or four segments. The results, in some cases, show increased link utilization and reduced link delay with the simulation of FTP and HTTP sessions.

The Internet draft in (Allman and Floyd., 1998) recommends increasing the initial window up to an upper bound of 4K. Simulation results show improved throughput of up to 30% for satellite systems. For FTP connections, a reduction in transfer time in some cases was about 10% for a small file of 16KB. A small increase in the drop rate was indicated. However, the finish time of a file transfer was still reduced. The draft recommends the use of the increased initial window in short lived TCP connections and connections over links with long RTT.

These Internet drafts are silent about (i) the maximum window size, and (ii) the rate of growth in the window in a form different from TCP Reno, also (iiii) compete wireless networks with WaveLAN. Therefore, an evaluation of these two variables is also required to tailor TCP for the needs of lossy wireless links.

Current TCP's Congestion Control Mechanism: An overview of TCP's slow start, congestion avoidance, fast retransmit, retransmission timer back off and RTT estimation is presented in (Stevens, 1994). In standard TCP, congestion is indicated on timeout or on reception of duplicate acknowledgments.

Two of the main variables that are used to manage the growth of the transmission window in TCP are slow start threshold (SSTHRESH) and congestion window (CWND), which are discussed below.

Slow Start: Slow start is entered at the start of a TCP session and also as a result of congestion. A timeout causes the congestion window (CWND) to be set to one. In addition, slow start initially sets the SSTHRESH to one half of the minimum of the current window and the receiver's advertised window, but not less than two. The congestion window grows exponentially with the arrival of acknowledgments, until SSTHRESH becomes less than CWND. At this stage, the congestion avoidance phase is entered.

Congestion Avoidance: Congestion avoidance indicates a linear growth in the congestion window. The linear growth in the CWND is accomplished by incrementing the CWND by one segment for every full window's worth of acknowledged segments.

Retransmission Timer Policy: The retransmission timer follows an exponential back-off mechanism. Once the retransmission timer expires, it is restarted with a value that is twice its current timer value. A timeout occurs when the retransmission timer counts down to zero.

Fast Retransmit and Fast Recovery in TCP Reno: Fast retransmit allows TCP to potentially retransmit lost packets on receiving three duplicate acknowledgments without having to wait for the retransmission timer to expire. Three duplicate acknowledgments are interpreted to indicate a loss of the segment. In this case, SSTHRESH is set to half of the minimum of the CWND and the receiver's advertised window. The CWND is set to SSTHRESH plus three times the segment size. Each time a duplicate acknowledgment is received, CWND is incremented by the size of one segment. Fast Recovery allows TCP to start congestion avoidance, instead of slow start, after fast retransmission allows the CWND to be larger that SSTHRESH.

When the first acknowledgment for a new data segment is received, CWND is set to the value of SSTHRESH. This results in the activation of the congestion avoidance algorithm.

TCP Noor's Modifications: As mentioned in the introduction section, TCP Noor applies a less conservative transmission policy during high and low bit error. Therefore, our algorithm is designed for better utilization of bandwidth on the wireless LAN. This is accomplished by dynamically adjusting the aggressiveness of the packet source transmission window according to the state of the TCP connection.

Enhanced Aggressive Error Recovery (EAER) Algorithm: Congestion control in standard TCP limits bandwidth utilization in a wireless LAN due to the setting of the congestion window to one on a timeout or upon reception of three duplicate acknowledgments. Under persistent BER, the resultant time period during which the congestion window remains one is prolonged due to the recurring timeouts and packet losses. During this period at most one TCP segment is delivered per round trip time. As a result, the bandwidth utilization for a wireless LAN environment becomes very low and the throughput suffers drastically.

Timeouts in wireless LAN can be attributed to a fading in the RF channel that causes data loss. However, once the fading is over, the channel is available for use. Traditional TCP is not optimized for use on RF channels, and assumes a congestion state resulting in severe degradation of throughput by not utilizing the channel immediately after the fade is over. Therefore, an alternative solution is needed for RF channels that suffer from transient data losses.

EAER in TCP Noor allows for a variable CWND, with a minimum value that is half of the maximum allowed window size, resulting in a continuous large transmit window over the wireless LAN link. Moreover, TCP Noor initiates exponential window growth after receiving one duplicate acknowledgment and at TCP session startup, and a linear window growth otherwise, i.e, after a timeout or receipt of three duplicate acknowledgments.

Operation of EAER

At Initial TCP Start up:

- Set the initial Congestion Window (CWND) to half of the maximum window size
- Set the Slow Start Threshold (SSTHRESH) to the value of the maximum window size. This allows exponential window growth until CWND is equal SSTHRESH. Then, once the CWND >= SSTHRESH the window grows linearly.

Initially, the larger transmit window, CWND, and the exponential transmit window growth, initially allow an aggressive transmission behavior. This results in better utilization of the bandwidth if the channel is available for transmission. However, if a timeout occurs, TCP Noor would transition to a linear window growth, which reduces the amount of traffic on the channel. In several IETF drafts for wire line environment, (Podri and Nichols, 1998) and (Allman and Floyd, 1998), the value of CWND, was recommended to be higher than one to achieve improved throughput.

TCP Reno can not exploit the fast retransmission mechanism at the beginning of slow start, since the CWND is one packet. Therefore, if the first packet is lost, a timeout would occur. This results in making the TCP session experience long idle periods at the start of transmission. Therefore, TCP Noor's larger transmit window at the start of the session makes TCP avoid these long idle periods. This is specially true if the timeouts are repeated.

Upon a Timeout:

- Set the SSTHRESH to the value of the CWND. This allows a linear window growth.

Timeouts indicate a persistent problem on the access channel. This condition, in wireless LAN networks, results in packet loss due to a prolonged contention on the access channel. Otherwise, other data or acknowledgment packets would have been successfully received.

Upon Receiving one Duplicate Acknowledgment:

- Set the CWND to half of the maximum window size.
- Set the SSTHRESH to the value of the maximum window size. This allows an exponential transmission window growth.

A reception of one duplicate acknowledgment, is an indication that the packet was possibly lost due to bit errors on the access channel. The reception of the first duplicate acknowledgment, and the loss of the prior one, are two main factors to this behavior. In addition, it is unlikely that after one duplicate acknowledgment that a timeout would occur. After bit errors, the channel is available for aggressive transmission, which is indicated by allowing the exponential growth in the transmission window.

Upon Receiving Three Duplicate Acknowledgment:

- Set the SSTHRESH to the value of the CWND. This allows a linear window growth.

Three duplicate acknowledgments are treated in the same manner as timeouts. The reception of three duplicate acknowledgments also indicates a persistent problem on the access channel. This condition, in wireless LAN networks, results in a packet loss due to a prolonged contention on the access channel. Otherwise, other data or acknowledgment packets would have been successfully received. Therefore, in TCP Noor's EAER, timeouts and the reception of three duplicate acknowledgments are treated in the same manner.

Set the Maximum Retransmission Timer to Two Seconds. The Retransmission Timer is Allowed to Back-Off Exponentially, to a Maximum of Two Seconds.

EAER's reduced maximum retransmission timer, of two seconds, allows a faster recovery of lost packets over a wireless LAN access channel. A repeated packet loss due to contention allows the exponential timer back-off mechanism, to be extended even further, up to sixty-four seconds in TCP Reno. This results in a prolonged idle period even-though the channel may be usable. Therefore, reducing the maximum value of the retransmission timer to two seconds in TCP Noor results in better utilization of the channel, and higher throughput on the wireless LAN network.

Simulation Model and Assumptions: Network Simulator NS (http://www-mash.cs.berkeley.edu/ns/) was used as the basis for our simulation. In this paper:

- We use the terms segment and packet interchangeably. Also, we use the terms, session and connection interchangeably.
- All nodes in the simulation were on a wireless LAN network, based on WaveLAN.
- WaveLAN channel access protocol based on Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA), was selected for our wireless LAN.
- TCP connections were started in a staggered manner. The start time of a session was based on the node number with which the session was associated. The node number was used to provide an incremental delay for session startup from the base startup time of the simulation.
- Network nodes were instantiated in a sequential order, i.e., first node one, then node two and so on. Each pair of two consecutive nodes had three concurrent FTP sessions with the lower number node being the sender of the FTP files.
- Error injection was installed per connection in both directions, from source to destination and vice versa.
- The LAN bandwidth was set to 10Mbps with one micro second link delay.
- In our simulation, the packet size means, TCP's packet size, which is the summation of TCP's twenty byte header and TCP's payload in bytes. The packet size used for all experiments was ninety-six bytes. This provided a better packet rate for high BER, i.e., 10**-3.
- The capacity of the First-In-First-Out (FIFO) queue was large enough to eliminate buffer overflows.

Channel and Traffic Model: Our simulation model utilizes a 10Mbps CSMA/CA channel access protocol, which is similar to WaveLAN protocol characteristics, with a 10Mbps bandwidth. WaveLAN is an implementation of wireless LAN channel, which is similar to the wireless LAN protocol IEEE 802.11. WaveLAN differs from IEEE 802.11 by not supporting a) Request To Send (RTS) and Clear To Send (CTS) signaling, b) Link level acknowledgment, and c) link level frame fragmentation.

We have assumed a two state Markov chain channel model, (Chaskar, Lakshman and Madhow, 1996) and (Crow, Widjaja, Kim and Sakai). The two states are defined as good state, and bad state. The duration of the good and bad states is exponentially distributed with mean values, m1 and m2, respectively. The bit error rate at the link level for both states was based on a uniform distribution with a mean of e. In addition, the bit error rate for the good state is ber1, and for the bad state is ber2. Fig. 1 depicts the overall system model used.

Our traffic generation used an exponential traffic distribution, (Chaskar *et al.*, 1996), which was based on a repeated cycle of a bursty period, with a mean duration b, followed by a pause period with a mean duration p. This mimics a sequence of file transfer using FTP.

The CSMA/CA channel access protocol used allows a new frame to be transmitted if the channel is not busy. Even-though the channel may not be busy, all transmissions require a contention test. A delay is applied prior to the contention check. If a contention is to occur, an exponential back-off timer is applied. However, if no contention is foreseen, the frame is successfully transmitted as long as no collision occurs. Otherwise, the frame is dropped as a result of collision. The collision may result due to multiple nodes on the channel selecting the same timeslot for transmission as a result of the exponential back-off timer algorithm.

The exponential back-off algorithm is applied when the channel is busy. The algorithm calculates the timeslot for the next contention test based on a random selection. The resultant timeslot is called the contention window. Therefore, the contention window is the number of timeslots a node/station would wait prior to attempting a transmission after detecting the channel is busy. This mechanism minimizes the probability of multiple stations contending for the same timeslot on their next attempt, which should reduce the probability of a collision. Once the retransmission back-off timer reaches its maximum threshold, the frame is dropped due to persistent contention on the link.

The channel access policy used, is very similar to, thought not the same as, Distributed Coordination Function (DCF), which is part of the wireless LAN protocol IEEE 802.11, (Crow, Widjaja, Kim and

Sakai). In the DCF mechanism, there is no guarantee of a minimum time delay to applications supporting time critical applications. Therefore, Quality of Service (QoS) guarantees, are more complex to accomplish when the DCF mechanism is used..

In our discussion, we use several terms/expressions, which are defined below:

- Cumulative Throughput, CT: User data bits for all TCP sessions in the experiment that are successfully received at the receiver TCP side, not including retransmitted and acknowledgment packets.
- Offered Load, OL: Number of cumulative data bits, for all sessions in the simulation, sent by the source TCP including user and acknowledgment packets.
- Link Loss, LZ: Cumulative data bits lost, for all sessions in the simulation, due to corruption on the link as a result of bit errors.
- Contention Loss, CL: Cumulative data bits lost, for all sessions in the simulation, due to a collision, or a repeated contention. This includes retransmitted and acknowledgment packets.

All data bits are based on the full TCP packet size, which includes user data bits, TCP header bits, and IP header bits.

The following formula defines the relationship among the above variables:

$CT \simeq OL - (LZ + CL + O) - A$, where

O = Number of data bits lost due to buffer overflow. O was found to be negligible.

A = Number of bits of acknowledgment packets which were successfully received at both ends of the TCP session

Results and Discussion

Simulation: We ran several sets of experiments to demonstrate the effectiveness of TCP Noor's EAER. All experiments were run with a bit error rate for the bad state, ber2, of 10**-3, 10**-4 and 10**-5, and for the good state, ber1, of 10**-6. The selection of a variable bit error rate, allowed the examination of, TCP Noor's and TCP Reno's, behavior under low and high BER. The selected packet size is ninety-six bytes, and the maximum window size is thirty-two packets. The mean duration for the traffic burst is 1 second, and for the bad state is 0.33 second.

The number of FTP sessions were varied based on the traffic rate required. The low, medium, and high traffic rates were set to 15, 75, and 105 sessions, respectively.

Simulation Experiment One : Impact of Bit Error Rate and Contention: In this experiment, the mean duration for the bad state was 0.33 seconds and for the good state was 1 second. Fig. 2 depicts the throughput obtained for the different ber1 values. Cumulative throughput is the actual TCP packets successfully received at the destination TCP, not including TCP acknowledgments or retransmitted packets. Fig. 3 depicts the ratio of packets lost due to contention to the offered load on the connection, (which includes retransmitted TCP packets and TCP acknowledgments). This ratio provides an insight of how packet loss due to contention is impacted by the variable BER and traffic rates. Fig. 4 depicts the ratio of packet loss due to bit errors on the link, including acknowledgments and retransmitted packets.

Figures 2 depicts that for all traffic rates, TCP Noor maintains a consistent throughput, almost independent of the BER value of the channel for the bad state with a slightly lower throughput at a BER value of 10**-3. The fact that TCP Noor's transmit window, is never less than half of the maximum window size contributes to allowing larger number of in transit packets. TCP Noor's exponential window growth, and the maximum of two seconds for the retransmission timer contribute to consistently maintaining a larger transmit window. TCP Reno narrows the window to a size of one packet after a timeout, as a result of applying slow start. This is further impacted when timeouts are repeated, resulting in continuous low channel bandwidth utilization. In addition, the larger exponential retransmission timer with a maximum of sixty-four seconds slows the recovery process for lost packets. These factors, for TCP Reno, result in a prolonged idle period at the sender, while the channel is available for transmission. However, TCP Noor's maximum retransmission timer of two seconds, reduces the idle periods as indicated by the higher throughput shown in Fig. 2.

TCP Noor's exponential window growth after receiving one duplicate acknowledgment allows an aggressive transmission behavior. However, EAER allows the window to grow linearly after a timeout, or after the reception of three duplicate acknowledgments, and sets the minimum transmit window to half of the maximum window size. This allows for continuous utilization of the channel.

Despite the increased packet loss due to contention, as shown in Fig. 3, TCP Noor continued to gain considerable throughput improvement. Using a linear window growth for TCP Noor after a timeout, or the reception of three duplicate acknowledgments, reduced the packet loss due to contention. This indicates that the congestion control mechanism in TCP Reno, which drops the size of the transmission window to one packet as a result of packet loss, degrades throughput considerably. This is especially the case for a high BER value of 10**-3, in which TCP Noor obtained about 250% throughput improvement over TCP Reno.

In Fig. 2, TCP Noor's throughput increased considerably as the offered load increased. This indicates that the proposed solution is scalable. In addition, the drop in throughput for TCP Noor due to a BER value of 10**-3 was not significant in comparison to TCP Reno, where the drop in throughput was about 70%. For TCP Noor, the increased throughput can be contributed to the minimization of the idle periods, as a result of the larger transmission window, and the smaller retransmission timer maximum of two seconds. The observed insensitivity to high BER indicates that TCP Noor is also a robust protocol.

In TCP Noor's EAER, bit errors on the link are interpreted by the reception of one duplicate acknowledgment. The maximum value of the retransmission timer for TCP Noor of two seconds compared to TCP Reno's value of sixty-four seconds contributes to a faster recovery in re-transmitting lost packets due to bit errors on the link. In addition, as already mentioned, it allows for a shorter idle period. Therefore, the response time at the sender TCP is minimized when using TCP Noor.

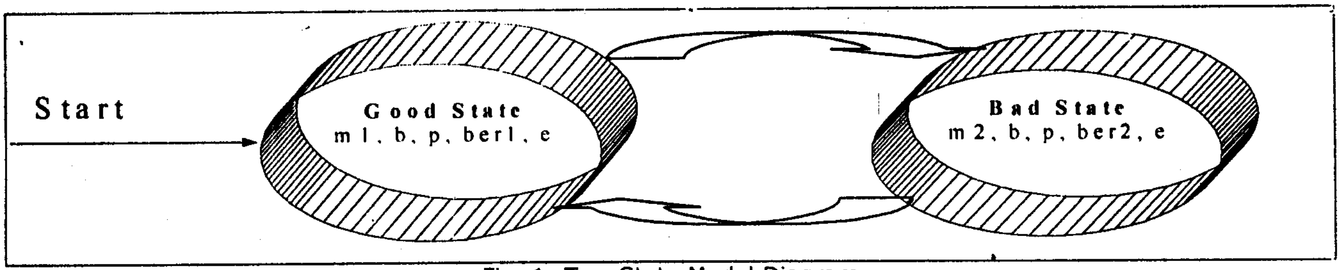

Fig. 1: Two State Model Diagram

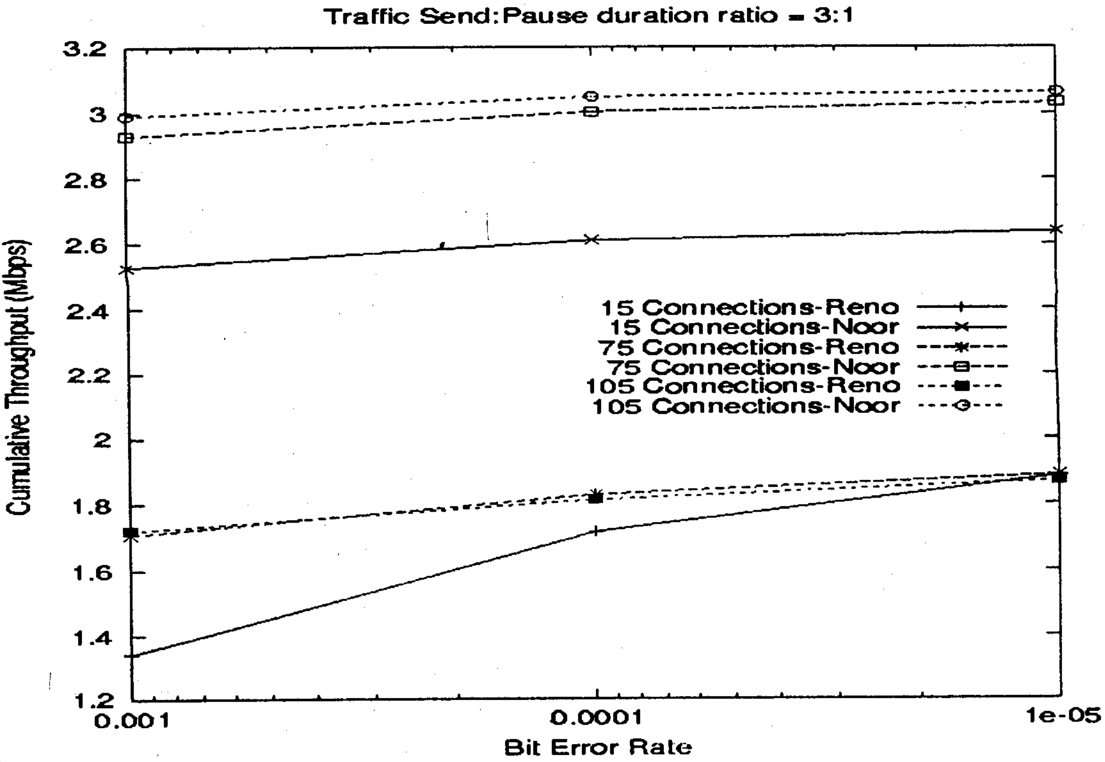

Fig. 2: Comparison of Throughput for Low, Medium and High Traffic Distribution

Simulation Experiment Set Number Two : Impact of Good and Bad State Duration: This experiment demonstrates the impact of a variable mean duration for the bad states which was one-third and two-thirds second. The mean duration for the good state remained one second.

In Fig. 5, for both TCP Noor and TCP Reno, the considerable increase in throughput was for the change in BER from 10**-3 to 10**-4. At a BER value of 10**-5, TCP Noor performed about the same for both duration ratios for the bad and good states, 2:3, and 1:3. This is due to having a consistently larger transmission window despite the increase in the mean duration of the bad state. The increase in TCP Noor's throughput is due to, a smaller maximum retransmission timer, minimum transmit window of half the maximum, and the exponential transmit window growth after receiving one duplicate acknowledgment. In addition, as a result of the aggressive transmission behavior in TCP Noor, the idle periods are minimized. This results in narrowing the idle period's duration, even-though the BER value has increased.

TCP Reno maintained a noticeable difference in throughput even at BER of 10**-5. This is due to its conservative transmission behavior. Allowing the transmit window size of one packet after a timeout, and the exponential retransmission timer back-off algorithm up to a maximum of sixty four seconds, results in a lower throughput as the mean duration

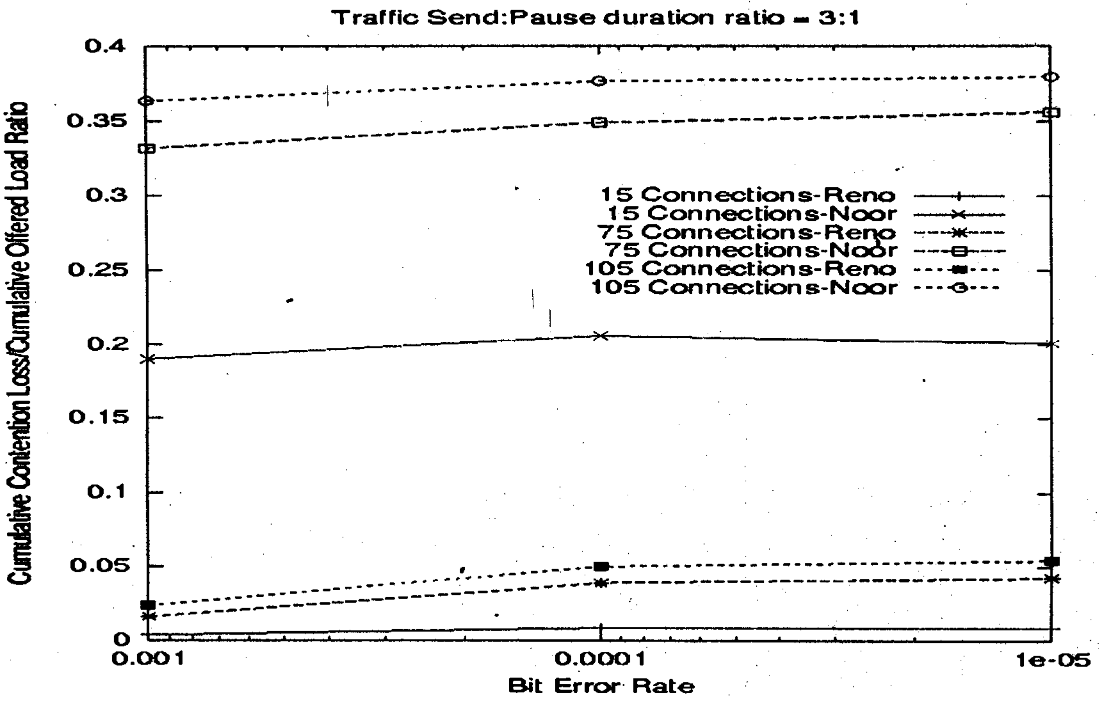

Fig. 3: Comparison of Packet Loss Due to Contention to Offered Load Ratio for Low, Medium and High Traffic Distribution

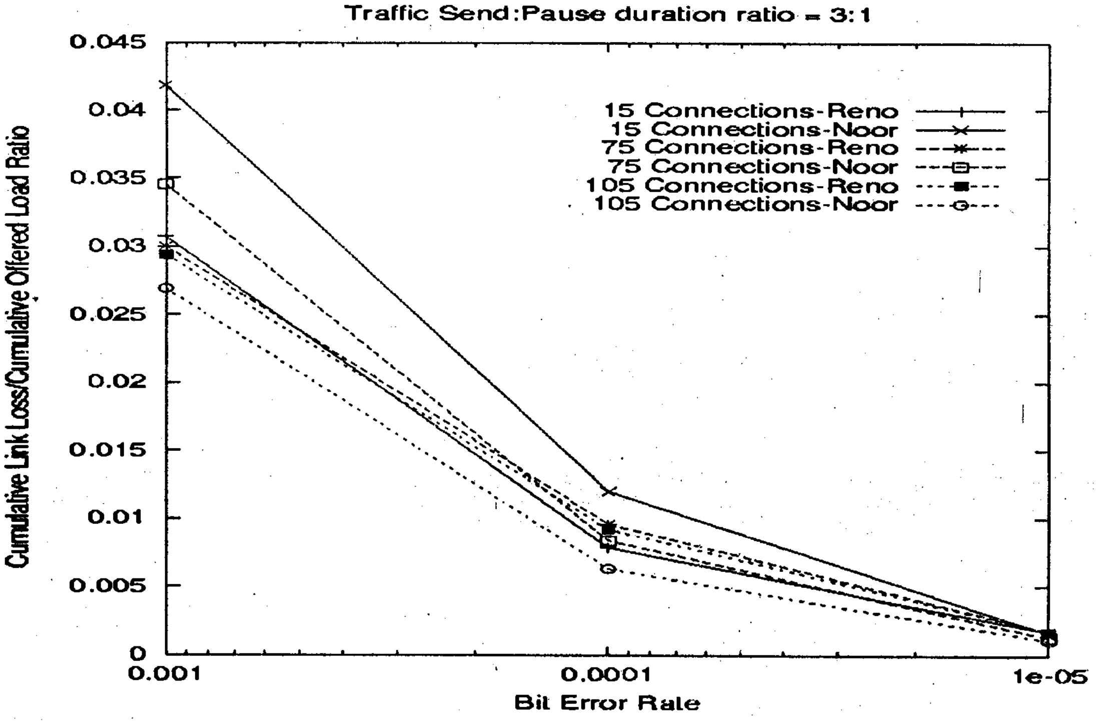

Fig. 4: Comparison of Packet Loss Due to Bit Errors to Offered Load Ratio for Low, Medium and High Traffic Distribution

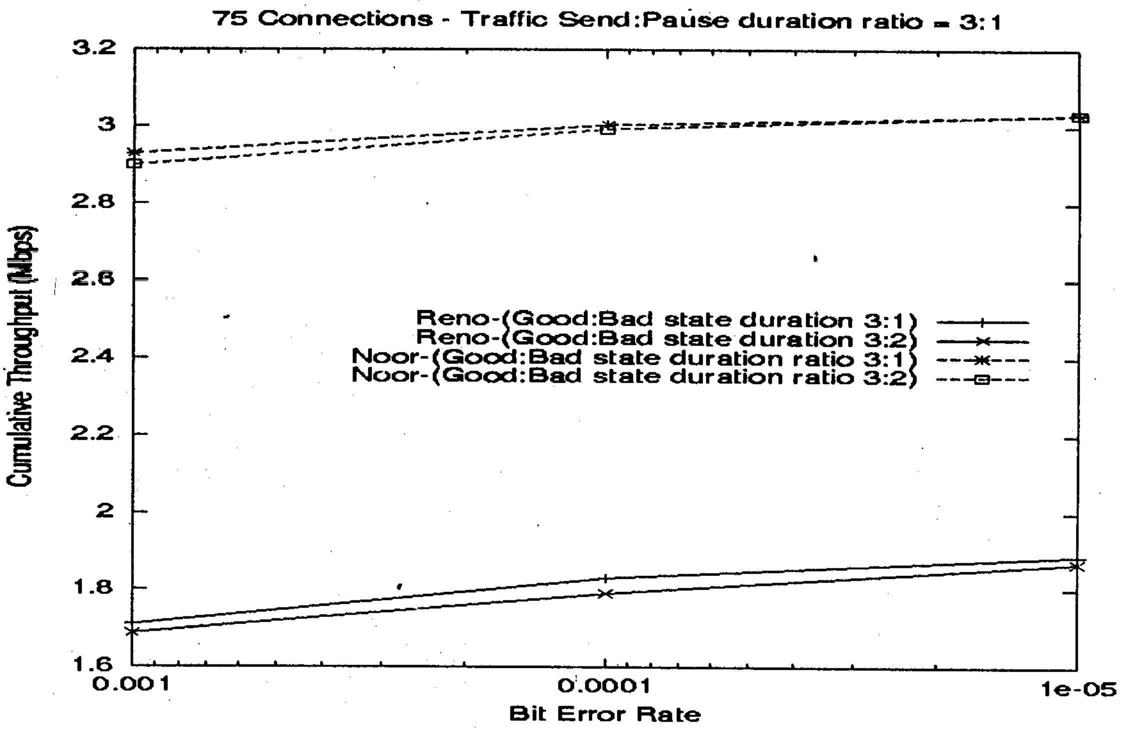

Fig. 5: Comparison of Throughput with Different BER for TCP Noor and TCP Reno

of the bad state increases. This is an indication of idle periods of no transmission for TCP Reno. In addition, the idle periods are prolonged further due to increased frequency of packet losses.

We have also performed other experiments with different values of the mean duration for the good and bad states while maintaining the same ratio as above, (3:1), and obtained about the same throughput results.

Acknowledgment

We would like to thank the Networks Research Group at Lawrence Berkeley Labs for making their network simulator NS publicly available.

Conclusion

Standard TCP is optimized for wire line networks and has poor performance in wireless LAN networks while experiencing high bit error rates. The proposed solution, EAER in TCP Noor, emphasizes the importance of considering changes in TCP's congestion control mechanism for wireless LAN networks. We have demonstrated the effectiveness of Enhanced Aggressive Error Recovery (EAER) in comparison to the congestion control mechanism in TCP Reno. EAER allows a larger transmit window by setting the congestion window to a minimum value of half the maximum transmit window including the initial transmit window for TCP. This allows TCP to reduce the idle periods by the transmission of packets. In addition, EAER allows exponential window growth after receiving one duplicate acknowledgment and at TCP session startup, and a linear window growth after a timeout or three duplicate acknowledgments. This results in better utilization of the bandwidth. Finally, the smaller maximum value of the retransmission timer of two seconds allows for faster recovery after a packet loss due to bit errors. The results indicate a significantly higher averaged throughput for TCP Noor in comparison with TCP Reno of about 250% in some cases. The poor throughput for TCP Reno, while experiencing bit errors and channel contention, is due to its assumption of network congestion. TCP Reno reacts to network congestion by applying a very conservative transmission policy, a small transmission window of one packet and the maximum retransmission timer of sixty-four seconds. If a packet loss occurs during the startup of slow start, the idle period extends. This is further impacted, if the packet loss is repeated. Our simulation results indicate that TCP Noor's EAER, takes advantage of its aggressive transmission nature and provides a significantly higher averaged throughput. As TCP Noor maintains its performance improvements for higher traffic, and longer bad state duration, it is both scalable and robust.

We have accomplished these results without any dependency on input from the link layer interface and transparently of any TCP applications. TCP Noor can inter-work with other versions of TCP. Therefore, TCP

Noor does not have to be populated to all nodes in order to inter-work with other versions of TCP. This is possible due to the fact that TCP Noor's modification only apply to the sender of the TCP traffic.

Further study is recommended to expand the simulation and handle a wider range of experiments. It would also be useful if similar experiments are applied over cellular systems. As the wireless multi-media applications require more bandwidth, the bandwidth of the wireless link becomes a very important resource that should be fully utilized especially while recovering from high bit error rates.

For further research, we recommend:

- Interaction of TCP Noor's Enhanced Aggressive Error Recovery, (EAER), for wireless LANs with wireline networks. Specially, the interaction with approaches such as I-TCP.
- Interaction of TCP Noor's EAER with wireless LANs with ad hoc networks.
- Reduction of contention in TCP Noor's EAER while maintaining significant throughput improvement.
- Understanding impact of a variable congestion window based on the maximum allowed window size, for improving the Quality of Service. For example, a user with higher bandwidth requirements may have a larger transmit window.

References

Allman, M., and F. Floyd, 1998. draft-floyd-incr-init-win-03.txt, "Increasing TCP's Initial Window", IETF Draft.

Bakre, A., and B. Badrinath, 1995. "I-TCP": "Indirect TCP for mobile hosts", in Proc. 15th Int. Conference on Distributed Computing Systems (ICDCS).

Balakrishnan, H., S. Seshan, R. Katz, 1995. "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks". ACM Wireless Networks, 4.

Balakrishnan, H., V. Padmanabhan, S. Seshan, R. Katz, 1997. "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links", IEEE/ACM Transactions on Networking, 6.

Banerjee, S., J. Goteti, G. Krishnamoorthy, " Extending TCP for Wireless Networks", http://www.cs.umd.edu/users/suman/docs/711s97/nodel.html.

Brakmo, L., S. O'Malley, L. Peterson, 1994. "TCP Vegas: New Techniques for Congestion Detection and Avoidance", in Proc. ACM SIGCOMM.

Caceres, R., P. Dauzig, S. Jamin, D. Mitzel, 1991. "Characteristics of Wide Area TCP/IP Conversation", in Proc. ACM SIGCOMM, pp 101-112.

Chandran, K., S. Raghunthan, S. Venkatesan, R. Prakash, 1998. "A Feedback Based Scheme For Improving TCP Performance In Ad-Hoc Wireless Networks", in Proc. 18th Int. Conference on Distributed Computing Systems (ICDCS).

Chaskar, H., T. Lakshman, U. Madhow, "The Design of Interfaces For TCP/IP Over Wireless", in Proc. IEEE Milcom, '96.

Cohen, R., S. Ramanathan, 1998. "TCP for High Performance in Hybrid Fiber Coaxial Broad-Band Access Networks", IEEE/ACM Transactions on Networking, 1.

Comer, D.E,, 1991. "Interworking with TCP/IP", 1, Prentice-Hall Inc.

Crow, B., I. Widjaja, J. Kim, P. Sakai, 1997. "IEEE 802.11 Wireless Local Area Networks", IEEE Communications Magazine.

Hoe, J., 1996. "Improving the Start-up Behavior of a Congestion Control Schem for TCP", in Proc. of SIGCOMM.

Jacobson, V., R. Braden, D. Borman, 1992. "TCP Extensions for High Performance", IETF RFC 1323.

Jacobson, V., 1998. "Congestion Avoidance and Control", in Proc. of SIGCOMM..

Jacobson, V., 1990. "Modified TCP Congestion Avoidance Algorithm," Message to end2end-interest mailing list, ftp://ftp.ee.lbl.gov/ email/ vanj.90apr30.txt.

Jakes, W., 1993. "Microwave Mobile Communications", reissued by IEEE Press.

Lakshman, T., U. Madhow, 1997. " The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss". IEEE/ACM Transactions on Networking.

Mathis, M., J. Mahdavi, 1996. "Forward Acknowledgment: Refining TCP Congestion Control", ACM/SIGCOMM.

Mathis, M., Mahdavi, J., Floyd, S., Romanow, A., 1996. "TCP Selective Acknowledgment Options", Request for Comment (RFC-2018).

Mosberger, D., L. Peterson, P. Bridges, S. O'Malley, 1996. "Anaylysis of Techniques to Improve Protocol Processing Latency". ACM/SIGCOMM.

Nanda, S., R. Ejzak, B. Doshi, 1994. "A Retransmission Scheme for Circuit-Mode Data on Wirless Links". IEEE J. on Selected Areas in Communications.

Paxon, V., 1994. "Empirically Derived Analytic Models of Wide-Area TCP Connections", IEEE/ACM Transactions on Networking 4.

Podri, K., K. Nichols, 1998. draft-ietf-tcpimpl-poduri-00.txt, "Simulation Studies of Increased Initial TCP Window Size", IETF Draft.

Qaddoura, E., R. Prakash, and L. Tamil, 1999. "Aggressive Error Recover For TCP Over Wireless Links", to appear in Integrated Computer-Aided Engineering J. in a special issue on Distributed Computing and Networking.

Ramakrishnan, K., Jain, R., 1998. "A Binary Feedback Scheme for Congestion Avoidance in Computer Networks with Connectionless Network Layer". In Proc. ACM SIGCOMM, 1998.

Ramakrishnan, K., R. Jain, "Dynamics of TCP Traffic over ATM Networks", ACM Transactions on Computer, 2.

RFC 793, "Transmission Control Protocol", DARPA Internet Program, Protocol Specification, 1981.

Romanow, A., Floyd, S., 1994. "Dynamics of TCP Traffic over ATM Networks". ACM/SIGCOMM.

Stevens, W.R., 1994. "TCP/IP Illustrated, The Protocols". Addison Wesley Longman, Inc.