Improving Performance of Rate-Based Transport Protocols in Wireless Environments

Luiz Magalhães, Robin Kravets, and Albert F. Harris, III

***Abstract*— Wireless links present higher loss rate than current wired links due to the inherent characteristics of wireless transmission. Lossy environments break the fundamental assumption of current transport protocols that a loss indicates congestion, which results in a reduction in throughput. To avoid changes to the infrastructure, we propose an end-to-end approach to loss discrimination based on network state estimation at the receiver. Discrimination is achieved by correlating the short-term jitter history with anomalous jitter and loss. Through simulation, an experimental rate-based protocol using the discrimination heuristic is compared to a stock TCP and a TCP modified to deal with wireless losses. Our experimental results validate the heuristics and show that better performance can be achieved.**

***Index Terms*—Wireless, Congestion Control, Rate-based Transport Protocols.**

I. Introduction

Current wired network links have very small bit error rates. Therefore TCP [1] was tuned to assume that any losses are caused by congestion. But the growing importance of wireless links which present higher loss rates break this assumption. In response to transmission losses, lost packets should simply be retransmitted, but faced with congestion the sender should reduce its offered load to prevent further losses. In this paper, we describe loss discrimination heuristics that can be combined with congestion control to allow the sender to react appropriately to loss.

The main challenge to loss discrimination lies in the fact that a communication channel may span both wired and wireless links, introducing both congestion and transmission losses into the channel. Solutions have been proposed at all layers of the protocol stack, from reliable link-layers [2] to new transport protocols. One hundred percent reliability at the link layer can interfere with end-to-end estimates of path round trip time (RTT) [3, 4]. If less reliability is provided, the end-to-end mechanisms must still be able to handle transmission losses. Network-layer solutions, such as explicit congestion or loss notification [3, 4], require changes to the infrastructure. Hybrid approaches, such as SNOOP-TCP [5], differentiate the lossy from the more reliable part of the path and try to optimize transmission across each part separately. Deployment of such approaches is limited by the need for intelligent base stations or agents in the network. We support solutions at the transport layer that do not require changes to the infrastructure.

Our contribution is an end-to-end mechanism for loss discrimination. Our rate-based approach uses timing information gathered at the receiver to infer the level of congestion on the path between the sender and the receiver. As losses are detected at the receiver, our heuristics use this timing information to discriminate between congestion and transmission losses.

L. Magalhães is with the Telecommunications Department at Universidade Federal Fluminense, Rua Passo da Pátria 156, Niterói, RJ Brazil (email: schara@telecom.uff.br)

R. Kravets and A Harris III are with the University of Illinois at Urbana-Champaign, 1304 W Springfield Ave., Urbana, IL 61801 USA (e-mail: {rhk, aharris}@cs. uiuc.edu).

II. Dealing with Transmission Losses

Successful loss discrimination is essential for effective communication in environments where losses may be caused by congestion or transmission errors. Without loss discrimination, all losses are attributed to either congestion or transmission. In the first case, transmission losses will be misinterpreted and performance will suffer. In the latter case, congestion losses will be misinterpreted and congestion will increase along the path.

Many solutions have been tried for the problem of adapting transport protocols to networks with heterogeneous loss characteristics. Infrastructure-based solutions try to hide the losses from TCP by adding changes to the intervening path, either at the link-layer or at the network-layer. A second class of solutions proposes changes to TCP. A final class of solutions proposes new transport protocols instead of changing TCP. In this section, we discuss our approach and compare it to existing approaches. A detailed survey of these approaches can be found in [6].

A. Infrastructure-Based Approaches

Infrastructure-based approaches are appealing because they require no changes to TCP and furthermore the link layer has direct knowledge of transmission errors. Link-layer error recovery is based on two mechanisms: Automatic Repeat Request (ARQ) and Forward Error Correction (FEC). While both approaches make a link appear reliable, neither is free. FEC imposes overhead on every packet and is computationally expensive. Because ARQ may increase the latency of the link and generate out-of-order packets, it can lead to worse overall performance for protocols that need reliable RTT estimates and expect packets to arrive in order [5]. FEC will transparently introduce relatively fixed overhead per packet, and so will not adversely affect our timing estimates. ARQ may skew RTT estimates and so render any end-to-end approach less effective. Additionally, any inclusion of reliability at the link layer may impose high overhead for streams that do not require any reliability.

To avoid such overhead for non-TCP traffic, it has been proposed to make the link-layer TCP-aware. An example of this approach is SNOOP_TCP [5], which changes the wireless base-stations, allowing them to cache unacknowledged packets. If the base station perceives duplicate acknowledges, it suppresses them, and sends the cached data instead. It also retransmits locally cached packets using timeouts, which should be smaller than the sender's timeout. Although this approach is effective, it requires modifications to all base stations involved in the communication. Indirect TCP (I-TCP) [7] splits the TCP connection into two parts, one over the wired and another over the wireless network, allowing each section to be optimized for the appropriate type of losses. The base-station acts as the TCP receiver and is responsible for forwarding data to the wireless host after it has received it. By discriminating between transmission and congestion losses, our heuristics enable successful transmission over channels that experience both types of losses without requiring specialization for either type.

B. Hybrid Approaches

Explicit knowledge of losses can be exposed to the transport-layer from the network and link layer. Explicit Congestion Notification (ECN) [4] allows the routers to inform TCP senders to drop their sending rate. Explicit Loss Notification (ELN) [3] allows the base-station to inform the sender of a transmission loss, so no congestion control measures need be applied. Such approaches require changes to both the transport protocol and the infrastructure. By using end-to-end path information, our heuristics strive to estimate such explicit information.

C. End-to-End Approaches

Since it is expensive to deploy infrastructure-based approaches, adaptations have been proposed for TCP to help it perform better in environments with higher loss rates. There are two challenges to end-to-end approaches. The first is accurate estimation of available bandwidth and the second is appropriate reaction to loss.

Traditional versions of TCP (i.e. Reno and Tahoe [8]), are optimized for minimal losses and so do not react well to multiple losses in the same window [9]. The use of Selective Acknowledgements (SACK) [10] has been suggestion to alleviate some of this problem, but does not address the issue of loss discrimination.

In order to do accurate loss discrimination, an end-to-end protocol needs information about the state of the network that can be estimated through observations of the transmission stream. Ideally, transit time for each packet should be used since this provides information about the network in the direction of the transmission. Solutions such as TCP Vegas [11] strive to achieve good estimates by monitoring RTT. The challenge lies in the fact that asymmetry on the path can cause inaccurate estimates of transit time. TCP Eifel [12] uses timestamps in each TCP packet to provide accurate estimate of RTT, but does not address the issue of different levels of congestion on the reverse path.

TCP Santa Cruz [13] also uses a timestamp returned from the receiver. The goal of TCP Santa Cruz is to estimate the level of queueing in the bottleneck link of a connection and maintain an optimal number of packets in the bottleneck of the connection, without congesting the network. Congestion-control is based on the tracking of network load. The timestamp TCP Santa Cruz uses is very similar to our proposal, although our rate-based approach is simpler and provides more accurate timing information.

Most similar to our heuristic-based approach is TCP-Aware [14], which monitors the transmission stream to determine transmission losses. The limitation of this approach is that TCP-Aware requires that the last hop be both the bottleneck and the lossy link. It is also unclear how accurate the heuristics from TCP-Aware are in the presences of multiple streams.

TCP Pacing [15] is another technique used for better bandwidth usage. The central idea is that the burstiness of TCP can lead to overflowing queues even when there is no congestion. The solution would be to spread out the packets, sending them at the same rate they would be sent, but over a period of time and not back-to-back. We advocate the development of rate-based protocols for the same reason. Although some problems with global synchronization have been discussed, we believe that the congestion avoidance techniques presented in this paper can prevent the synchronization of losses that can lead to link underutilization.

Instead of reengineering TCP to deal with lossy environments, new protocols have been proposed. An example is Wireless Transmission Control Protocol (WTCP) [16], which was designed to provide a reliable transport protocol for CDPD. WTCP uses heuristics based on detecting congestion, and tagging losses as transmission losses if the network is considered uncongested. The presence of congestion is based on observations about long-term jitter.

III. Network Characterization

An end-to-end approach to loss discrimination is limited by the information that can be inferred from the behavior of the transmission stream. Therefore it is necessary to have effective mechanisms for estimating path characteristics, specifically the presence of congestion along the path.

A. Path Characteristics

The basis of our loss discrimination technique is accurate determination of congestion along the path from sender to receiver. In order to monitor congestion, we need to understand the characteristics of the end-to-end channel, specifically the expected amount of bandwidth available to each flow. If multiple flows are present, the available bandwidth should be divided fairly among the flows. In an ideal environment, this can be achieved by the exact knowledge of the available bandwidth. In a dynamic environment, if every flow is trying to achieve maximum bandwidth without causing congestion, dynamic equilibrium can be reached. This can be seen, for example, when multiple TCP flows share the same link.

The challenge lies in the determination of available bandwidth and the translation of this value to the maximum share of bandwidth for a particular flow. If this estimate is too low, the flow will not receive its share of bandwidth and so its throughput will be reduced. On the other hand, overestimates may cause congestion along the path. TCP flows determine this maximum by pushing the limits of the network. Once the limit has been reached, congestion will occur, causing loss in the TCP stream, and TCP will reduce the amount of bandwidth it is using.

In comparison, our heuristics monitor path characteristics in order to determine available bandwidth without causing congestion. Initial estimates of path bandwidth are measured using the packet pair method [17]. Two packets are sent back-to-back and their interarrival time is measured. Since the packets were sent back-to-back, the timing of the arrivals represents the current limit of the network. Packets should thus be sent separated by this time period to achieve maximum bandwidth usage and avoid causing congestion in the network. Since available bandwidth is a moving target, this technique can be used periodically throughout the life of the transmission stream to probe the path for up-to-date estimates of bandwidth. Since our approach strives to avoid causing congestion, the next challenge is how to use implicit information about the characteristics of the path to infer that there is congestion building in the network.

B. Congestion

In end-to-end communication, the receiver is in the best position to collect information about the communication channel. The receiver can distill this information and send the results back to the sender to affect changes in the transmission stream. For rate-based transmission, the receiver expects to receive packets at regular intervals, as determined by the sending rate. Information about the channel can be inferred from the difference between the expected and actual arrival time of packets.

Information at the receiver is based on both observation and protocol parameters. The main piece of observed data is the arrival time of a packet. If routes remain stable, packet size is constant and compression does not affect packet transmission time, the main component of the jitter is the time the packets wait in the routers' queues.

If the service rate is greater than the arrival rate at any router, the queue size should be close to zero, growing only because of burstiness in traffic. Since our definition of expected arrival time is based on the actual arrival time of the previous packet, negative jitter can occur when the first packet experiences longer queueing delays than the second packet. If the service rate at any queue along the path is smaller than the arrival rate, congestion will occur. As the queue grows, the jitter will be positive. At some time the queue will overflow, and packets will be dropped. Even in a stable network queues will grow and shrink. Therefore, jitter values will be positive and negative over time. In a stable environment, the sum of the jitters should stay near zero.

Minimum transmission time occurs when the network is idle (no packets enqueued along the path), and maximum transmission time occurs when the queues at every router are full. The effect of dropped packets is noticeable by the reduced waiting time of succeeding packets that were successfully queued.

If the dropping strategy affects packets within the queue, the net effect is the same. For example, if packet 8 has a higher priority than packet 4, packet 4 may be dropped out of the queue. If all of the odd packets belong to the same stream, the arrival of packet 5 will be earlier than expected and the receiver doesn't have to wait for packet 9 to determine that there was a congestion loss. Essentially, the effects of the loss will be perceived at the arrival of the next packet in the queue after the dropped packet. In this way, a flow will notice the effect of dropped packets from other flows.

IV. CONGESTION AVOIDANCE AND LOSS DISCRIMINATION

Without explicit loss or congestion notification, successful loss discrimination is dependent on implicit mechanisms. These mechanisms can be geared toward identifying congestion losses or identifying transmission losses. An end-to-end approach is limited to information at the transport layer, and so transmission loss determination is not possible. On the other hand, identifying congestion losses is tightly coupled to determination of congestion in the network. Accurate determination of network state is complex and often not feasible. To this end, our loss discrimination mechanisms are based on heuristics that integrate congestion avoidance techniques. By monitoring end-to-end channel characteristics, the state of the path between the sender and receiver can be estimated. As a loss is observed, this information is used to determine if the cause of the loss was from congestion along the path. If no congestion is indicated, the loss is determined to be transmission-based. In effect, the result of the loss discrimination is fed into the congestion control mechanisms to determine how to react.

A. *Ideal Loss Discrimination*

In an idealized network, congestion losses only occur when the bottleneck link has reached saturation. Therefore, any loss occurring when the saturation level has not been reached is a transmission error. As congestion causes queues along the path to fill, travel time for successive packets will increase. Intuition tells us that it should be useful to consider this increase in order to determine the cause of a packet loss. Unfortunately, the complexity of finding this saturation point hampers the effectiveness of this information.

B. *Congestion Avoidance Heuristics*

Information about congestion along the transmission path allows congestion avoidance even when no losses are detected in the transmission stream. If a trend signaling high network utilization is detected, the protocol should slow down. By tracking consecutive interarrival times of packets, the receiver is in fact tracking the recent history of network load. By monitoring jitter for consecutive packets, the receiver can make the following observations:

An increasing trend in interarrival times of packets signals increased network load

A decreasing trend in interarrival time of packets may signal reduced network load or congestion

To address the first observation, it is useful to monitor the consecutive number of arrivals that experience positive jitter. Positive jitter occurs when the packet arrives after its expected arrival time. Consecutive packets experiencing positive jitter indicate increasing queue sizes and potential congestion. In response, the sender should reduce its offered load to avoid adding to the building congestion in the network.

The second observation shows that a trend of negative jitters may indicate contradictory situations. Reduced interarrival time may be caused by reduced network load. The determination of reduced load is left to bandwidth estimation mechanisms. Therefore, the congestion control algorithm need not respond to negative jitter in this first case. Unfortunately, negative jitter may also indicate that the network was very overloaded. Each instance of negative jitter could indicate a congestion loss and sustained negative jitter could indicate that many packets were dropped, allowing successive packets to arrive early (i.e. experience negative jitter). The second case for negative jitter is caused by congestion, but it is difficult to differentiate it from the first case. Therefore, we leave congestion determination in the face of negative jitter to our loss discrimination heuristics.

C. *Loss Discrimination Heuristics*

The use of a loss as an indication of congestion is a very powerful tool. Our goal is to make sure that tool is used accurately. The goal of our loss discrimination heuristics is to determine how to react appropriately to losses. We consider two scenarios when losses occur: loss in the presence of positive jitter and loss in the presence of negative jitter.

Increasing jitter indicates an increased load on the network. Congestion losses will not occur until a queue along the way is full. Therefore, a loss during a period of increasing jitter can be considered a transmission loss. It is important to note that even if the loss were actually a misinterpreted congestion loss, the consecutive observations of positive jitter still indicate congestion to the congestion avoidance heuristics. In this case, the congestion avoidance heuristics will react to the indications of congestion, making it unnecessary for the loss discrimination heuristics to react to the loss.

Negative jitter can indicate congestion losses in the queues along the path or the unloading of the network. To aggressively react to the first situation, a loss followed by negative jitter will always be characterized as a congestion loss. This characterization is conservative since the sender will react to a transmission loss during a reduction in network load as if it were congestion, reducing its sending rate. If the network load is indeed reducing, the probing mechanism will be able to recover from such misinterpreted losses.

It is possible that a congestion loss, which would normally be followed by the observation of negative jitter at the receiver, will actually be followed by the observation of positive jitter. This can occur when the succeeding packet experiences delay in a router between the router at which the congestion loss occurred and the receiver. We expect that these scenarios will be self-regulating – after all, there is a maximum queue size on every router, and even if one router had growing queues, it will eventually reach a maximum size, when the masking is no longer possible and losses will cause negative jitter.

V. EVALUATION

The effectiveness of our loss discrimination techniques is based on the accuracy of the discrimination heuristics. The misinterpretation of a loss can adversely affect the throughput of the stream or increase the amount of congestion in the network. Therefore, we must evaluate the probability of each type of misinterpretation. The goal of our initial simulations is to determine an upper bound on the number of such misinterpretations.

A rate-based protocol transporting bulk data can be modeled as a sequence of CBR streams, with data rate varying according to the packet size and the sending period.

In the first series of simulations, we used ns-2 to simulate a CBR stream following a path that had competing TCP and CBR flows. The TCP flows cause some links to become congested and lead to dropped packets. In this environment, we applied our heuristics to the trace to see with what confidence we could detect the cause of a packet loss. It should be clear that by using CBR flows in this simulation we are not backing off when congestion occurs, which must be part of any congestion control algorithm. We show in the second simulation that there is a different behavior when congestion avoidance is used (with a smaller network loading due to congestion control on the flow).

From the simulation, we obtained the packet number and the arrival time at the destination for each packet. Losses are indicated by gaps in the sequence numbers. Along with the arrival time, the current rate or sending period is needed for jitter calculations. In this simulation, the period is fixed throughout the experiment. All losses observed in the simulation are congestion losses. In a sample run, from the ~120,000 packets sent, 8253 gaps due to congestion were detected in the transmission stream.

In order to determine the worst-case effect of our heuristics, we do not simulate transmission losses, but instead calculate the effect had a transmission loss occurred prior to the arrival of each packet. We assume that the transmission loss would leave all other timing aspects unchanged. This is a reasonable assumption if the last link is the link subject to transmission losses and this link is not congested. It should be noted, however, that the results will not mirror that of an actual run, but instead give a worst-case result. The reason is simple: if we assume, for example, single losses, both the previous and the following packet have to arrive. In our case, we calculate the timing of single losses for every packet, not every other packet. We can do that because we have the timing information for the packets that have arrived, and we can consider a scenario where, for each packet, their direct predecessor and successor have arrived, but the packet itself has been lost.

There are two interesting two performance measurements: how many real congestion losses will be detected as transmission losses, and how many transmission losses will be considered congestion losses. If too many congestion losses are considered transmission losses, the use of the heuristic can increase congestion. If transmission losses are considered congestion losses, the performance of protocols that use this heuristic will be negatively affected.

Based on our simulation for single transmission losses, the change in timing can mask up to 26% of congestion losses, as seen in Table 1. The results are similar for two and three consecutive losses. As the number of consecutive transmission losses grows, it becomes harder to discriminate congestion from transmission error. The loss of timing information masks the indicators of congestion. This may point to a more advanced heuristics where it becomes more conservative as the number of consecutive errors grows. We evaluate our heuristics over varying queue sizes. We successfully evaluate between 65 and 96% of the congestion losses.

TABLE 1
CBR-BASED TRANSMISSION LOSS CLASSIFICATION.

Queue Size	Conse-cutive Losses	Transmission Losses Detected	Mistaken Transmission Losses
5	1	70848	1583
10	1	70349	3084
15	1	74418	85
25	1	74034	624
40	1	75297	522
5	2	70762	4169
10	2	68277	3570
15	2	69234	5035
25	2	69070	3034
40	2	73045	803
5	3	58725	929
10	3	57643	2337
15	3	60420	157
25	3	59684	1453
40	3	61327	715

Table 2 shows the corresponding results for transmission errors. Since our simulation did not impose transmission error, this is simply an evaluation of the situation had each packet actually been lost. For transmission losses, this is a worst cases analysis. We use these results to help understand how well our heuristics perform.In the second set of simulations, we use a specialized rate-based protocol that reacts to the congestion indications. The goal of this simulation is to show that we can significantly reduce the number of congestion losses during the transmission by adhering to our congestion and loss heuristics. We run our protocol between the same nodes. There is a TCP stream running across the bottleneck at node 2. We do not simulate transmission losses in this simulation either but instead use the same calculations to obtain the worst-case findings.

TABLE 2
CBR-BASED CONGESTION CLASSIFICATION.

Queue Size	Conse-utive Losses	Congest. Detected	Mistaken Congest. Losses	Actual Congest	Loss Detection %
5	1	6670	1583	8253	0.808191
10	1	5754	3084	8838	0.651052
15	1	7503	85	7588	0.988798
25	1	7673	624	8297	0.924792
40	1	7348	522	7870	0.933672
5	2	12285	4169	16454	0.746627
10	2	13903	3570	17473	0.795685
15	2	10137	5035	15172	0.668139
25	2	13347	3034	16381	0.814785
40	2	14846	803	15649	0.948687
5	3	23721	929	24650	0.962312
10	3	23262	2337	25599	0.908707
15	3	22594	157	22751	0.993099
25	3	22673	1453	24126	0.939775
40	3	22565	715	23280	0.969287

As is shown in Table 3, the number of losses due to congestion is dramatically reduced by reacting to the congestion indicators. It can also be seen that the heuristics are still accurately discriminating between congestion and transmission losses. Similar results are shown for transmission loss determination in Table 4.

Table 3
Protocol-Based Congestion Loss Classification.

Queue Size	Consecutive Losses	Transmission Losses Detected	Mistaken Transmission Losses
5	1	25836	147
10	1	30175	153
15	1	31989	163
25	1	31393	163
40	1	36082	202
5	2	26660	318
10	2	31294	345
15	2	32955	356
25	2	32742	359
40	2	37647	410
5	3	27527	501
10	3	32248	549
15	3	34037	562
25	3	34251	559
40	3	39049	627

TABLE 4
PROTOCOL-BASED TRANSMISSION LOSS CLASSIFICATION.

Queue Size	Consec-utive Losses	Congest. Detected	Mistaken Congest. Losses	Actual Congest.	Loss Detection %
5	1	421	147	568	0.741197
10	1	466	153	619	0.752827
15	1	496	163	659	0.752656
25	1	443	163	606	0.731023
40	1	511	202	713	0.716690
5	2	783	318	1101	0.711172
10	2	846	345	1191	0.710327
15	2	893	356	1249	0.714972
25	2	805	359	1164	0.691581
40	2	948	410	1358	0.698085
5	3	1095	501	1596	0.686090
10	3	1186	549	1735	0.683573
15	3	1238	562	1800	0.687778
25	3	1118	559	1677	0.666667
40	3	1319	627	1946	0.677801

VI. CONCLUSIONS

In this paper we proposed a simple approach to loss discrimination based on accurate determination of congestion in the path from the sender to receiver. Congestion determination is based on simple observations of the arrival time of packets at the receiver. First, by monitoring the recent history of jitter, we can perceive network loading and try to prevent congestion by reducing the sending rate at times of increased load. Second, packet loss due to congestion causes subsequent packets to arrive early, causing what we call "negative jitter," allowing us to tag such losses as congestion losses. Lost packets that do not cause changes in the expected arrival times can be safely tagged as transmission losses. Transport protocols can then react wisely to loss, by reducing the sending rate in the presence of congestion, and maintaining the current rate in case of transmission loss.

Our simulations show that the accuracy of the heuristic is directly tied to the regularity of the flow. Using a CBR stream, the heuristic was more accurate than with the time-varying flow. This observation may indicate that ack-clocked protocols such as TCP may not obtain good results using these heuristics because of their high variance in transmission times and their inherent burstiness. On the other hand, rate-based protocols may greatly benefit from the use of such heuristics to control their flow.

REFERENCES

[1] J. Postel, "Transmission Control Protocol," Internet Engineering Task Force, Request for Comments (Standard) RFC 793, September 1981 1981.

[2] A. DeSimone, M. C. Chuah, and O. C. Yue, "Throughput Performance of Transport-Layer Protocols over Wireless LANs," presented at Globecom '93, 1993.

[3] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," in *Proceedings of the SIGCOMM '96 Symposium*, 1996.

[4] S. Floyd, "TCP and Explicit Congestion Notification," *ACM Computer Communications Review*, vol. 24, 1994.

[5] H. Balakrishnan, S. Seshan, E. Amir, and R. Katz, "Improving TCP/IP Performance over Wireless Networks," in *First ACM International Conference on Mobile Computing and Networking (MOBICOM)*, 1995.

[6] K. Pentikousis, "TCP in Wired-Cum-Wireless Environments," *IEEE Communications Surveys*, 2000.

[7] A. Bakre and B. R. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts," in *IEEE International Conference on Distributed Computing Systems (ICDCS) '95*, 1995.

[8] D. E. Comer, *Internetworking with TCP/IP: Principles, Protocols and Architecture*, vol. 1: Prentice Hall, 1995.

[9] K. Fall and S. Floyd, "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP," *Computer Communications Review*, 1996.

[10] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP Selective Acknowledgement Options," Internet Engineering Task Force, Request for Comments RFC 2018, October 1996 1996.

[11] L. S. Brakmo and L. L. Peterson, "TCP Vegas: End to end congestion avoidance on a global internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, 1995.

[12] R. Ludwig and R. H. Katz, "The Eifel Algorithm: Making TCP Robust Against Spurious Retransmissions," *ACM Computer Communications Review*, vol. 30, 2000.

[13] C. Parsa and J. J. Garcia-Luna-Aceves, "Improving TCP Congestion Control Over Internets with Heterogeneous Transmission Media," presented at IEEE International Conference on Network Protocols (ICNP'99), 1999.

[14] S. Biaz and N. H. Vaidya, "Discriminating Congestion Losses from Wireless Losses using Inter-Arrival Times at the Receiver," presented at IEEE Symposium ASSET'99, 1999.

[15] A. Aggarwal, S. Savage, and T. Anderson, "Understanding the Performance of TCP Pacing," presented at IEEE INFOCOM '00, 2000.

[16] P. Sinha, N. Venkitaraman, R. Sivakumar, and V. Bharghavan, "WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks," presented at ACM Mobicom '99, 1999.

[17] S. Keshav, "A Control-Theoretic Approach to Flow Control," presented at Proceedings of the SIGCOMM '92 Symposium, 1992.

[18] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, 1993.