In: Design and Analysis of Wireless Networks
Editors: Yi Pan and Yang Xiao, pp. 291-308
ISBN 1-59454-186-8

TRANSPORT LAYER DESIGN IN MOBILE WIRELESS NETWORKS*

HAOWEI BAI†, SHAOJIAN FU‡ AND MOHAMMED ATIQUZZAMAN§

Abstract. In this chapter, we discuss transport layer design issues in mobile wireless networks. The focus is on two transport protocols: Transmission Control Protocol (TCP) and Stream Control Transmission Protocol (SCTP). TCP is the dominant transport protocol which is currently used for most Internet services such as Web browsing, file transfer, remote login etc. SCTP is a recently developed protocol by IETF for signaling message transport over IP networks, and it is being actively studied to explore its applicability to new arenas. Both the protocols were mainly designed with wired network in mind. In mobile wireless environment, the performance of TCP and SCTP are seriously affected due to issues such as low bandwidth, high packet error rate, user mobility, sudden delay spikes, etc. In this section, we explain the impact of these issues and the proposed schemes to improve the performance of TCP. The recent research efforts in SCTP over mobile wireless networks are also surveyed.

1. Introduction. Mobile wireless networks constitute an indispensable part of the global Internet as access networks for providing digital telephony, web browsing, file download, and various location based services. In a error-prone mobile wireless environment, the design of a transport protocol is more complex than in a wired network context. Although the Transmission Control Protocol (TCP) is the dominant transport protocol in the IP protocol suite, it was not initially designed for mobile wireless networks. TCP cannot adapt well to situations arising from wireless mobile environment, such as high link error rate, sudden link delay or bandwidth change, or user mobility. In the past ten years, a number of enhancements to TCP have been proposed to improve its performance over wireless networks. These enhancement fall into two categories: (a) those which are enhancement to the TCP protocol (such as Selective Acknowledgement, window scaling, etc.) and (b) those which are new schemes to give a better end to end performance (such as snooping, splitting a TCP connection, etc).

Recent interest in transmitting voice over IP networks [27] has led IETF to develop a new transport layer protocol, called Stream Control Transmission Protocol (SCTP) [35], for the IP protocol suite. SCTP is a reliable network-friendly transport protocol which can co-exist with TCP in the Internet. The design of SCTP adopted many strengths of TCP (such as window based congestion control, error detection and retransmission, etc.), that made TCP a success during the explosive growth of the Internet. Several of the enhancements (such as Selective Acknowledgement, window scaling, etc.) to the TCP protocol to improve its performance over wireless and mobile networks have already been incorporated in the basic SCTP protocol thereby making it inherently suitable for mobile wireless networks [20]. Moreover, SCTP incorporated several unique features, such as multi-streaming and multi-homing, that are not available in TCP. Although, the initial aim of SCTP was to provide a robust protocol for the transport of signalling messages over an IP network, later developments have made it also useful for a wider range of applications.

In this chapter, we will first provide a review of the characteristics of mobile wireless

*The work of Fu and Atiquzzaman was funded by National Aeronautics and Space Administration (NASA) Grants NAG3-2528 and NAG3-2922.
†Honeywell Labs, MN65-2200, 3660 Technology Drive, Minneapolis, MN 55418, USA. Email: haowei.bai@honeywell.com.
‡School of Computer Science, University of Oklahoma, Norman, OK 73019-6151, USA. Email: sfu@ou.edu.
§School of Computer Science, University of Oklahoma, Norman, OK 73019-6151, USA. Email: atiq@ou.edu.

networks (Sec. 2) and their impact on network performance (Sec. 3). Then we discuss some of the important improvements recently proposed to improve the performance of TCP over wireless mobile environment (Sec. 4). In Sec. 5, we introduce the main features of a new transport protocol (SCTP), such as multi-homing, multi-streaming, congestion control. Recent research activities on running SCTP over mobile wireless networks are reviewed (Sec. 6), which includes effect of link delay spikes on SCTP, SCTP over Mobile IP, mobile handover based on SCTP multi-homing, and SCTP over Ad-hoc networks.

2. Main Characteristics of Mobile Wireless Networks. Mobile wireless networks have a few fundamentally different characteristics from wired networks. They include:

- low bandwidth,
- high link error rate, and
- mobility of end hosts resulting in hand-offs.

A mobile wireless link usually provides much less bandwidth than a traditional wired link. This is due to the limits on physical layer design and transmission medium. The main causes of high error rate in a mobile wireless link are described below. Authors in [7] provide a good tutorial on how to model the error behavior arising due from these facts.

- Attenuation: This is due to decrease in the intensity of electromagnetic energy at the receiver (e.g., due to long distance), which leads to low signal-to-noise ratio (SNR).
- Intersymbol interference (ISI): This is caused by delay spread (the arrival of a transmitted symbol is delayed), resulting in partial cancellation of the current symbol.
- 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: This is caused by multipath propagation of radio frequency (RF) signals between a transmitter and a receiver. Multipath propagation can lead to fluctuations in the amplitude, phase and angle of the signal received at a receiver.

The mobility of end hosts could result in *blackouts* and *handoffs*. Blackouts are time periods during which a mobile host is temporarily disconnected from the base station. This could be caused by multipath fading. Handoff is the processes by which a mobile host's transmission is transferred from one cell to another. When a mobile host moves from its home cell to another cell, it has to sign off with its home base station, and log on to a new base station. Fig 2.1 illustrates a handoff scenario in a typical cellular network. During the handoff proceess, all packets sent to the home base station may be dropped.

3. Impact of Mobile Wireless Links on TCP Performance. Currently the vast majority of IP traffic is transmitted using TCP. TCP is supported by almost all existing network application programs. The convergence of IP services with mobile wireless networks leads to various access methods to IP services, and the diversity of end-host computing devices. TCP will still be the dominant end-to-end reliable transmission control protocol at least in the near future. However, TCP was initially designed to perform well in networks with reliable wired links and stationary hosts, where packet losses are mainly due to network congestion. TCP assumes that all packet losses are due to network congestion. In the current TCP Reno, two mechanisms have been deployed for loss recovery: the timeout mechanism and the Fast Retransmit and Recovery algorithm [28, 34].

A TCP sender uses packet loss as an implicit signal for network congestion with the assumption that packet losses are mainly caused by congestion. Packet losses are indicated by a timeout or by the receipt of three duplicate ACKs. A TCP sender continuously increases

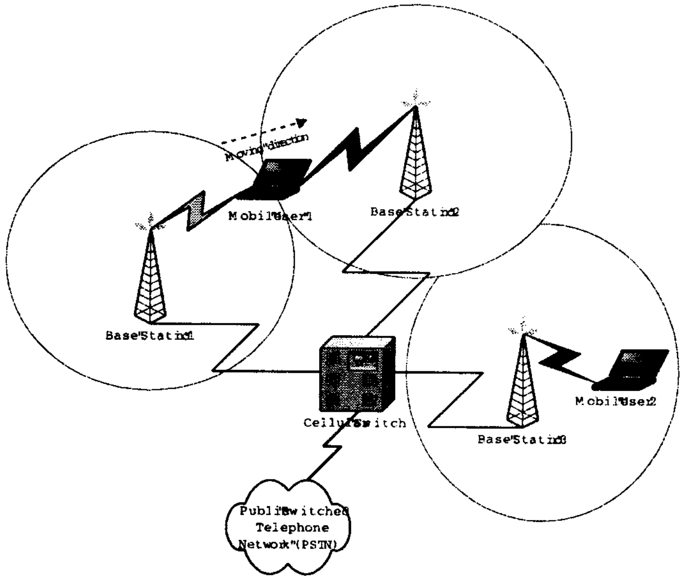

FIG. 2.1. *Handoff in a typical cellular network*

its traffic into the network until either one of above two indications is received. As shown in Fig 3.1, a retransmit timer timeout always forces TCP to the Slow-Start phase, during which the traffic is doubled every Round Trip Time (*RTT*). When *ssthresh* is reached, TCP switches to a Congestion Avoidance phase, during which the traffic increases linearly at one Maximum Segment Size (*MSS*) per *RTT*. The TCP sender will retransmit a packet if it receives three duplicate ACKs for the packet sent immediately before the lost packet. This procedure is called Fast Retransmit (See Fig 3.1). Fast Recovery[1] is used if a TCP sender transmits a lost packet using Fast Retransmit. Fast Recovery halves the congestion window congestion window size as denoted by *W* in Fig 3.1.

When wireless links are involved in the network connection, packet losses are mainly caused by link errors and/or hand-offs. When TCP experiences these errors, it incorrectly infers that the network is congested and unnecessarily invokes the congestion control schemes. TCP's unnecessary reduction of the congestion window size decreases the network throughput, and increases the end-to-end delay. Fig 3.2 shows the variation of congestion window size, caused by congestion losses and link corruption losses. It shows that packet losses due to network congestion happened at time 19 sec, 22 sec, 34 sec, and 50 sec, while a packet was lost due to link corruption at time 57 sec. At time 57 sec, TCP incorrectly decreased its congestion window size. Table 3.1 [39] shows experimental results of TCP throughput over an IEEE 802.11 and an IEEE 802.11b wireless LAN. This shows that a traditional TCP algorithm in a wireless environment significantly degrades network performance.

4. Improving TCP Performance in Mobile Wireless Networks. Many schemes have been proposed to improve the performance of TCP over wireless links. These can be classified into two approaches. In the first approach, the sender is *aware* of the existence of wireless links in the network, and attempts to either distinguish losses due to wireless

[1]The first version of TCP, TCP Tahoe does not support Fast Recovery.

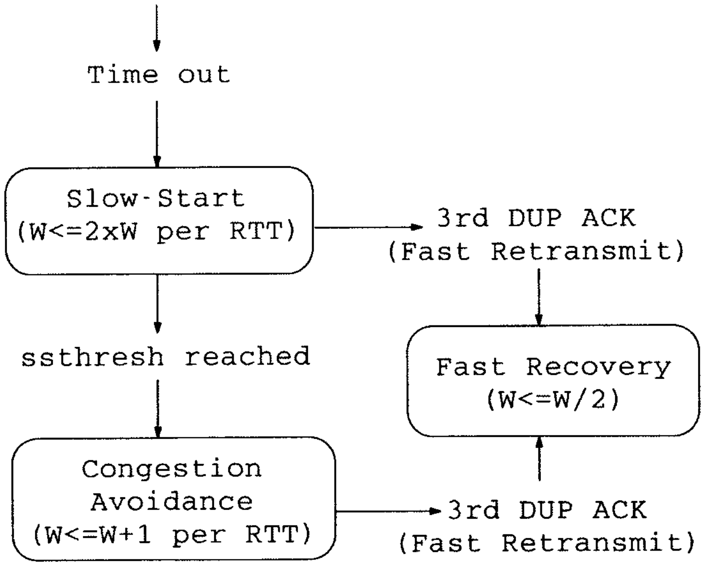

FIG. 3.1. *TCP Reno congestion control algorithm*

TABLE 3.1
TCP throughput over IEEE 802.11 connections

Connection	Data transmission rate	TCP throughput	Effective Bandwidth
IEEE 802.11	2 Mbps	0.98 Mbps	49%
IEEE 802.11b	11 Mbps	4.3 Mbps	39.1%

links from those due to congestion to prevent the sender from invoking congestion control algorithms when the packet loss is caused by wireless errors [10], or quickly recover from packet losses. In the second approach, the TCP sender is *unaware* of the losses due to wireless links. The non-congestion related losses are hidden from the TCP at the fixed host (sender), and hence the TCP at the fixed host remains unmodified. In the rest of this section, we describe some schemes proposed for both wireless aware and unaware approaches.

4.1. Wireless Aware TCP. In this approach, the fixed host (sender) is aware of the existence of wireless links in the network and tries to either distinguish wireless link corruption losses from network congestion losses, or quickly recover from packet loss events. We discuss TCP extensions based on this approach.

4.1.1. Limited Transmit. This mechanism [4] is effective in the cases of a large number of packet losses within a congestion window, or the congestion window size is small [29]. The Limited Transmit scheme extends Fast Retransmit and Fast Recovery algorithms [34] for TCP flows with small congestion windows that are not likely to generate three duplicate acknowledgements to trigger Fast Retransmit. Using Limited Transmit, if there are unsent packets in the sender's queue, the sender sends a new packet in response to the arrival of each of the first two duplicate acknowledgements. Authors in [4] have shown that over half of a busy server's retransmissions were due to the expiration of TCP retransmission timer. Furthermore, roughly 25% of these retransmissions could have been avoided using

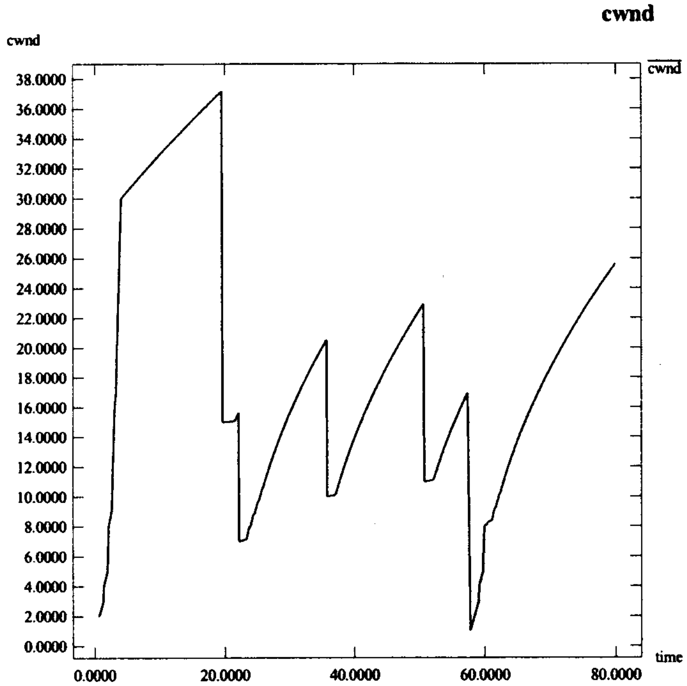

FIG. 3.2. *TCP congestion window size variation caused by packet losses*

Limited Transmit.

4.1.2. Selective Acknowledgements (SACKs). Using SACKs [32], the sender can be exactly informed which packets need to be retransmitted in the first RTT (Round Trip Time) following the loss event. SACK thus allows TCP to recover from multiple segment losses in a window of data within one RTT of loss detection. Although Fast Retransmit, Fast Recovery and SACK are generally able to rapidly recover from multiple packet losses, they reduce the congestion window to avoid further congestion. The above behavior, which is based on the assumption that packet losses are indicators of congestion, results in the degradation of throughput in the presence of non-congestion related packet losses (such as wireless link errors). Therefore, when they are applied to wireless links, where most of packet losses are due to link errors instead of congestion, TCP is unable to determine the available bandwidth.

4.1.3. Distinguishing Congestion Losses from Corruption Losses. This method makes the congestion window behave differently in the presence of congestion losses and corruption losses (due to link errors and hand-offs) by distinguishing the two types of losses. The various algorithms that have been proposed using this method are summarized in Table 4.1. Authors in [14] provide an comparison of some of the algorithms.

4.2. Wireless Unaware TCP. This approach is based on the intuition that since most of the wireless segments in the global Internet are close to local end users, the packet loss problem should be solved locally, and TCP should be independent of the behavior of individual links. We present below some schemes based on this approach.

TABLE 4.1
Distinguishing congestion losses from corruption losses

Algorithm	Method to Distinguish
TCP-Decoupling [38]	Sending TCP data packets and header packets in independent streams; congestion control is only applied to the header-packet stream.
TCP-Peach [2]	Sending dummy packets to probe the type of losses.
WTCP [33]	Measuring the inter-packet interval.
LEA [23]	Sender's receiving of either an acknowledgement packet, or an ICMP (Internet Control Message Protocol), or both.
ELN [9]	Explicitly setting the ELN bit in packet header whenever a non-congestion loss is detected.
Diff-C-TCP [6]	Optimally dimensioning ECN-capable RED gateway and notifying congestion losses with ECN.

4.2.1. Snoop and Delayed Duplicate Acknowledgements (DDA). The Snoop algorithm [11] assumes that the wireless link is the last hop in the TCP connection, and introduces a module, named the *snoop agent* at the base station. The agent caches TCP packets that have been sent across the link but have not yet been acknowledged by the receiver. The agent retransmits the lost packet (if it has been cached) and suppresses the duplicate ACKs for lost TCP packets. The lost packets are retransmitted locally, thereby avoiding unnecessary fast retransmissions and congestion controls by the sender. This scheme, however, needs a base station to maintain the state information, and cache the unacknowledged TCP packets, which results in scalability issues.

The DDA scheme [37] attempts to imitate the behavior of Snoop by using link-layer retransmissions. However, DDA tries to reduce the interference between TCP-layer retransmissions and link-layer retransmissions by delaying the third and subsequent duplicate packets for an interval of d. If the receiver receives out-of-order packets, it responds to the first two out-of-order packets by sending duplicate packets immediately.

4.2.2. Indirect-TCP (I-TCP). This scheme [8] breaks the connection between the fixed wired network and the wireless mobile host into two connections. One connection is between the fixed host and the base station; the other connection is between the base station and the wireless host. Data sent to the wireless host is first received by the base station. Upon receiving the data, the base station sends an acknowledgement to the fixed host and then the received data is forwarded to the wireless host. The base station and the wireless host does not need to use TCP for communication. Instead a specialized protocol that is optimized for mobile applications and for low speed and unreliable wireless medium can be used. This indirection helps shield the wired network from the uncertainties of the wireless network. However, I-TCP may violate the acknowledgement mechanism of the current TCP, because acknowledgements of data packets would possibly reach the original source before the data packets reach the wireless host.

4.2.3. M-TCP. This architecture [12] was proposed for cellular networks to support high bandwidth and frequent hand-offs. The architecture can be viewed as a three-level hierarchy. Mobile hosts which communicate with mobile stations in each cell are at the lowest level. Several mobile stations are controlled by a supervisor host at the second level. Supervisor hosts are connected to the high-speed wired network at the highest level and handles most of the routing and other protocol details for mobile users. M-TCP is used for the communication between mobile hosts and mobile stations. When the mobile station receives data from the sender, it forwards it to the wireless host but defers the ACK to the sender until it receives an ACK from the mobile host. If a mobile host undergoes a hand-off or a period of data losses, the mobile station sends the deferred ACK and advertises a window size of zero, which leads the sender to a *persist state*. During this period, all timers are frozen until the mobile host regains the connection. This algorithm provides a solution to the problem of frequent and periodic disconnection.

4.2.4. Freeze-TCP. The main design goal of Freeze-TCP [24] is to handle hand-off disconnections. It is easy for a mobile host to monitor signal strengths, detect an impending handoff, and even predict a temporary disconnection. Therefore, the idea of Freeze-TCP is to modify the TCP algorithm at the mobile host so that the base station can be prevented from sending packets during hand-offs. If a handoff occurs, the mobile host sets advertises a zero receiver window size to force the sender to enter a frozen mode and preventing it from dropping its congestion window size.

4.3. Comparison of TCP variants. Table 4.2 [12] compares the performance of the major TCP enhancement schemes, in terms of the following criteria:

- Is end-to-end semantics maintained?
- Is it able to handle high BER?
- Is it able to handle hand-off disconnections?
- Is it a loss-distinguishing scheme?
- Is it a modification of existing TCP?

As seen in Table 4.2, only I-TCP, M-TCP, and Freeze-TCP are able to handle hand-offs which frequently occur in mobile environments. Furthermore, only M-TCP is able to handle both high BER and hand-offs.

5. SCTP: A New Transport Layer Protocol. Until late 2000, the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) have been the only available standard transport layer protocols in the TCP/IP protocol suite. A new transport layer protocol, called Stream Control Transmission Protocol (SCTP) was first standardized by the IETF SIGTRAN (Signaling Transport) working group (founded in November 1998). It was soon noticed that SCTP should be useful in a wider range of applications instead of just for signaling transport, resulting in moving the standardization work of SCTP from SIGTRAN to the Transport Area Working Group (TSVWG) of IETF in February 2001.

The design of SCTP absorbed many of the strengths of TCP, such as the window based congestion control, error detection and retransmission, that led to its success during the explosive growth of the Internet. Moreover, SCTP incorporated several new features that are not available in TCP. The two most prominent of the the features are multi-homing and multi-streaming, which will be discussed in more detail in Sec. 5.1.

Due to its new attractive features, SCTP has received much attention from the research community, and has become one of the hot topics in networking technology [36, 13, 18]. In this section, we provide the readers with a brief review on the main features of SCTP.

5.1. Main Features of SCTP. Like TCP, SCTP resides in the transport layer of the Internet protocol stack as shown in Fig. 5.1 which illustrates an SCTP association using multi-homing and multi-streaming.

5.1.1. Multi-homing. Multi-homing allows an association between two end points to span across multiple IP addresses or network interface cards. An example of SCTP multi-

TABLE 4.2
Comparison of different TCP enhancement schemes

TCP Enhancement Schemes	End-to-end Semantics	Handle High BER	Handle Hand-off Disconnects	Distinguish Losses	Modify Current TCP
Limited Transmit	√	√			√
SACK	√	√			√
TCP-Decoupling	√	√		√	√
TCP-Peach	√	√		√	√
WTCP	√	√		√	√
LEA	√	√		√	√
ELN	√	√		√	√
Diff-C-TCP	√	√		√	√
Snoop	√	√			
DDA	√	√			
I-TCP		√	May run out of buffer		
M-TCP	√	√	√		
Freeze-TCP	√		√		√

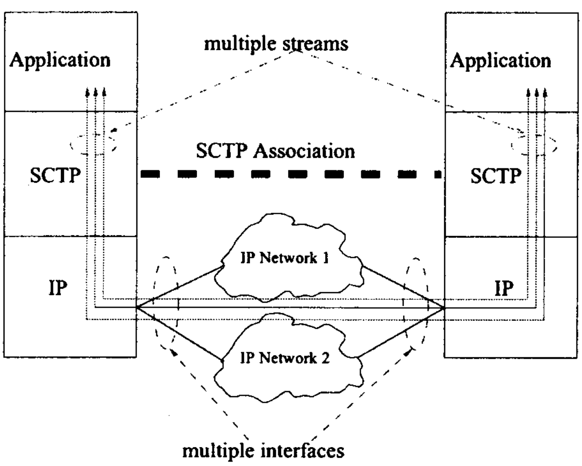

FIG. 5.1. *Schematic view of an SCTP association*

homing is shown in Fig. 5.2, where both endpoints A and B have two interfaces bound to an SCTP association. The two end points are connected through two types of links: satellite at the top and ATM at the bottom. One of the addresses is designated as the primary while the other can be used as a backup in the case of failure of the primary address, or when the upper layer application explicitly requests the use of the backup. Retransmission of lost packets can also be done over the secondary address. The built-in support for multi-homed endpoints by SCTP is especially useful in environments that require high-availability of the applications, such as SS7 signaling transport. A multi-homed SCTP association can speedup recovery from link failure situations without interrupting any ongoing data transfer.

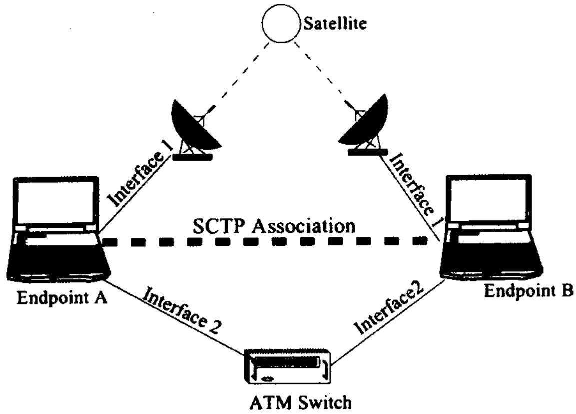

FIG. 5.2. *An SCTP association with multi-homed endpoints*

5.1.2. Multi-streaming. Multi-streaming allows data from the upper layer application to be multiplexed onto one channel (called association in SCTP) as shown in Fig. 5.3. Sequencing of data is done within a stream; if a segment belonging to a certain stream is lost, segments (from that stream) following the lost one will be stored in the receiver's stream buffer until the lost segment is retransmitted from the source. However, data from other streams can still be passed to the upper layer application. This avoids the head of line blocking found in TCP, where a single stream carries data from all the upper layer applications. In other words, the HOL effect is limited within the scope of individual streams, but does not affect the entire association.

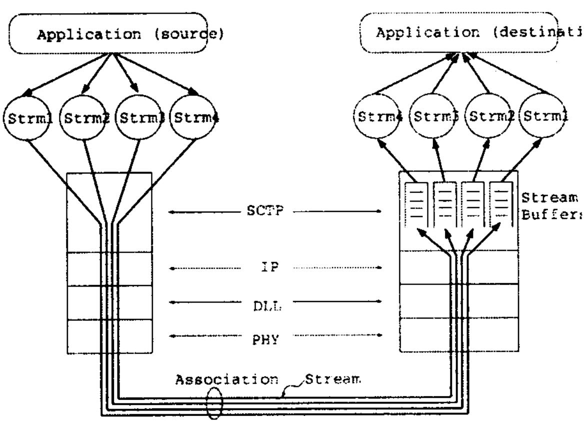

FIG. 5.3. *An SCTP association consisting of four streams carrying data from one upper layer application*

An example application of using SCTP multi-streaming in Web browsing is shown in Fig. 5.4. Here, an HTML page is split into five objects: a java applet, an ActiveX control, two images, and plain text. Instead of creating a separate connection for each object as in TCP, SCTP is making use of its multi-streaming feature to speedup the transfer of HTML

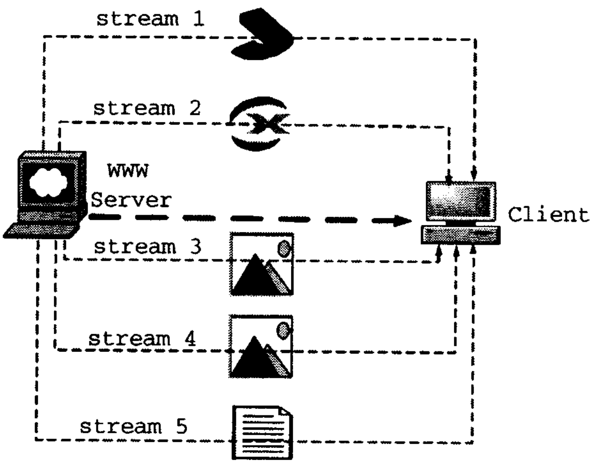

FIG. 5.4. *Multi-streaming in Web browsing*

pages. By transmitting each object in a separate steam, the HOL effect between different objects can be eliminated. If one object is lost during the transfer, the others can still be delivered to the Web browser at the upper layer, while the lost object is being retransmitted from the Web server. This results in a better response time to users while opening only one SCTP association for a particular HTML page.

5.1.3. Congestion Control. SCTP congestion control is based on the well proven rate-adaptive window-based congestion control scheme of TCP. This ensures that SCTP will reduce its sending rate during network congestion and prevent congestion collapse in a shared network. SCTP provides reliable transmission and detects lost, reordered, duplicated or corrupt packets. It provides reliability by retransmitting lost or corrupt packets. However, there are several major differences between TCP and SCTP as summarized below:

- SCTP incorporates a fast retransmit algorithm based on SACK gap reports similar to that of TCP. This mechanism speeds up loss detection and increases the bandwidth utilization. One of the major differences between SCTP and TCP is that SCTP doesn't have an explicit fast-recovery phase. SCTP achieves fast recovery automatically with the use of SACK [35].
- Compared to TCP, The use of SACK is mandatory in SCTP, which allows more robust reaction in the case of multiple losses from a single window of data. This avoids a time-consuming slow start stage after multiple segment losses, thus saving bandwidth and increasing throughput.
- During congestion avoidance of SCTP, *cwnd* can only be increased when the full *cwnd* is utilized; this restriction does not exist in TCP.
- TCP begins fast retransmission after the receipt of three DupACKs; SCTP begins after four DupACKs.

Alamgir et al. [3] compared the congestion control mechanisms of TCP and SCTP in a satellite environment. This work presented a detailed case study on the retransmission policies of the two protocols, and showed that under certain network scenarios, SCTP can achieve better performance than TCP even when both the protocols share a satellite path fairly. The throughput improvement, which was reported to be up to 30.6% resulted from the different retransmission mechanisms of TCP and SCTP during the congestion avoidance

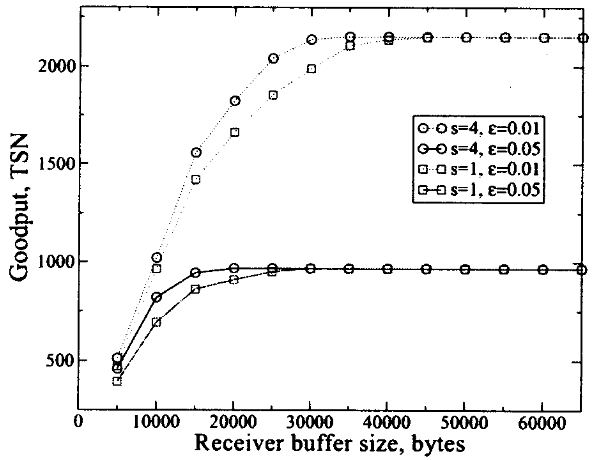

FIG. 6.1. *The effect of multi-streaming on goodput*

phase.

6. SCTP Over Mobile Wireless Networks. Although SCTP was initially designed primarily to transport signalling messages, the application of SCTP in wireless/mobile networks has become a hot spot in SCTP research. In this section, we discuss current research activities in this area, and cite recently published papers when applicable, with an attempt to providing readers with a clear vision on the state-of-the-art research on SCTP in wireless mobile environment.

6.1. Using SCTP Multi-streaming to Increase Goodput and Reduce Buffer Requirement. Mobile wireless environment is characterized by bandwidth-limited channels and buffer-limited user terminals. Atiquzzaman et al. [5] showed that multi-streaming results in higher goodput than a single stream when the receiver buffer is constrained, as in the case of wireless handheld devices. As illustrated in Fig. 6.1, it can be observed that for small receiver buffer sizes, multi-streaming (s indicates the number of streams within one association) can increase the SCTP goodput by eliminating HOL blocking in an error-prone (ϵ indicates the packet error rate) wireless environment.

In Fig. 6.1, we can also see that when the available receiver buffer increases, the goodput of SCTP association in the presence of packet errors becomes independent of the buffer size. We can determine the *optimal receiver buffer size* for different packet error rates, as shown in Fig. 6.2. The *optimal receiver buffer size* is defined as the threshold size above which the increased buffer does not contribute to increasing the goodput anymore, and hence is wasted. The figure demonstrated that the multi-streaming feature of SCTP (here s also indicates the number of streams within one association) results in reduced buffer requirements at the receiver in the presence of losses in wireless networks. The above two advantages (increased goodput, decreased buffer requirement) make SCTP an attractive transport protocol for wireless handheld devices.

6.2. Effect of Delay Spikes on SCTP. Like TCP, SCTP is also designed with wired network in mind. There are a number of problems in wireless communications, one of which is that wireless mobile networks encounter delay spikes more frequently than wireline networks. A *delay spike* is defined as a situation where the RTT suddenly increases and then

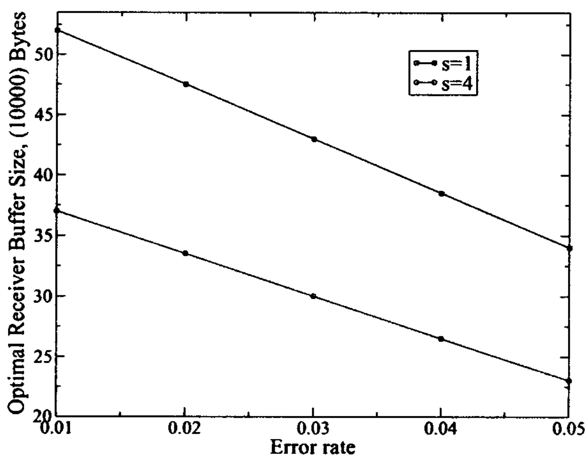

FIG. 6.2. *The effect of multi-streaming on buffer requirement*

drops sharply back to its previous value [26]. Delay spikes in a wireless mobile environment may occur due to mobile hand-off, temporary physical disconnection of the wireless link, link level recovery by the RLC layer, and preemption of data traffic by higher-priority traffic [25].

Delay spikes, resulting in *Spurious Timeout* (ST) and *Spurious Fast Retransmission* (SFR), can lead to serious end to end performance penalty in TCP [31]. The Eifel algorithm [31] has been proposed to alleviate the performance penalty in the case of TCP. Eifel requires both the sender and receiver to support TCP's timestamp option, which in turn requires an additional 12 bytes in the TCP header. Other alternative proposals to Eifel can be found in [30]. SCTP is based on congestion control and retransmission schemes which are similar to those of TCP; they assume all losses are caused by congestion, and RTT changes slowly and gradually. However, in the presence of frequent delay spikes in wireless networks. This will cause SCTP to back-off unnecessarily as TCP does and result in poor end-to-end throughput.

In a study by Fu et al. [19], the effect of delay spikes on SCTP in a wireless mobile environment is studied. It is shown that, like TCP Reno, SCTP also suffered a "go-back-N" behavior after a delay spike. It is further shown that SCTP SACK could be used to eliminate Spurious Fast Retransmission in SCTP. In the case of a lossy network with small bandwidth and receivers with large buffers, SCTP has been shown to perform better than TCP Reno and Eifel in the presence of delay spikes. Based on the simulation results in *ns-2* [1] simulator, the performance of the three protocols is ranked in *descending order* for different cases of link bandwidths, Receiver Window, and packet loss in Table 6.1.

6.3. SCTP over Mobile IP. Mobile IP [15] is the standard proposed by IETF to offer seamless mobile computing. During the handover period in Mobile IP, there is a very high probability of packet losses in wireless channels, which may eventually result in the transport protocol backing off in terms of data transmission rate. The performance of SCTP in Mobile-IP was investigated by Fu et al. in [17]. In this study, the main focus is the effect of improving end-to-end throughput by SCTP SACK compared to TCP SACK.

The TCP SACK option is defined in [16], the format of which is shown in Table 6.2, and the format of SCTP SACK chunk [35] is shown in Table 6.3. The use of SACK is mandatory in SCTP, which allows more robust reaction in the case of multiple losses from

TABLE 6.1
Relative performance of protocols in descending order (S. rwnd= *Small* rwnd, *L.* rwnd = *Large* rwnd*)*

	Low link BW		High link BW	
	S. *rwnd*	L. *rwnd*	S. *rwnd*	L. *rwnd*
Delay Spike	Eifel Reno SCTP	Eifel Reno SCTP	Eifel Reno SCTP	Eifel Reno SCTP
Delay Spike + Loss	Reno SCTP Eifel	SCTP Reno Eifel	Reno SCTP Eifel	Reno SCTP Eifel

TABLE 6.2
TCP SACK option format

0	15	23	31
	Kind=5		Chunk length
Left Edge of 1^{st} Block			
Right Edge of 1^{st} Block			
......			
Left Edge of n^{th} Block			
Right Edge of n^{th} Block			

a single window of data.

For TCP, the length of the *Options* field is limited to 40 bytes, while a SACK option specifying n blocks will have a length of $8 \times n + 2$ bytes. Therefore, the maximum number of SACK blocks that the TCP SACK option can have is limited to four. If the SACK is used together with time-stamp option (requiring 12 bytes), the maximum SACK blocks allowed would be three.

In contrast to TCP, SCTP allows a large number of blocks in its SACK chunk. The total available chunk space is determined by the *"Chunk Length"* field which is 2^{16} bytes. Subtracting the first 16 bytes required for description of a SACK chunk (see the first four rows in Table 6.3), the maximum length of space for gap blocks is $2^{16} - 16$. Every block needs 4 bytes; therefore the total number of blocks allowed is 16380, we can nearly regard it as an unlimited one. *When there are multiple non-consecutive segment losses in a single window, the number of available SACK blocks in TCP may not be sufficient for reporting all the segment losses.* The large number of SACK blocks makes SCTP more robust in case of multiple losses.

The behavior of TCP-Reno, TCP-SACK, and SCTP during the Mobile IP handover is compared based on *ns-2* [1] simulation results using the topology shown in Fig. 6.3. A router connects the CN to a HA and FA, and a MH moves back and forth between the HA and FA. The coverage of the HA and FA are shown by the dotted circles, where we assume overlapping coverage between the HA and FA.

Using *ns-2* simulation, it was shown that the support of a large number of SCTP GapACK blocks in its SACK chunks can expedite the error discovery and lost packet retransmission, and result in better performance than TCP-Reno and TCP-SACK. Simulation results have shown that the throughput improvement is especially prominent when the net-

TABLE 6.3
SCTP SACK chunk format

0	7	15	31
Type=3	Chunk Flags	Chunk length	
Cumulative TSN Ack			
Advertised Receiver Window Credit			
Number of GapAck Block		Number of Dup TSN	
Gap Ack Block #1 Start		Gap Ack Block #1 End	
......			
Gap Ack Block #N Start		Gap Ack Block #N End	
Duplicate TSN 1			
......			
Duplicate TSN X			

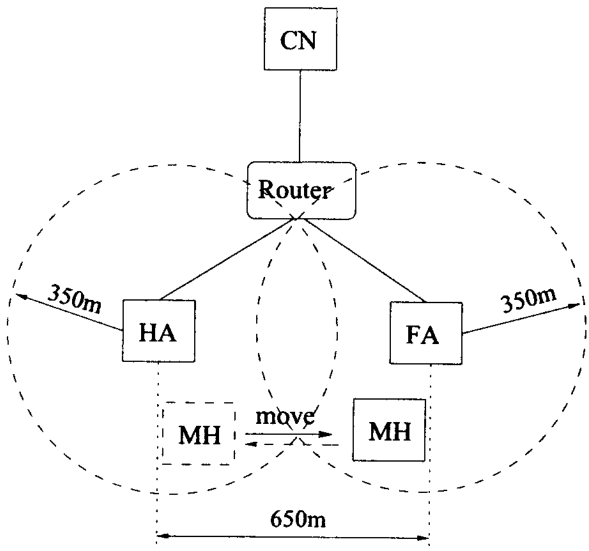

FIG. 6.3. *Simulation topology with Mobile IP handover*

work bandwidth is low.

6.4. Mobile Handover based on SCTP multi-homing. Using SCTP's multi-homing feature, researchers at University of Oklahoma and elsewhere are investigating new handover schemes in mobile computing. A new scheme, called **T**ransport Layer **S**eamless **H**andover (TraSH) [21, 22], is proposed in this context, where the handover is accomplished at the transport layer without requiring any modification to the IP infrastructure.

A typical handover scenario based on TraSH and using SCTP's multihoming feature is illustrated in Fig. 6.4. Initially the MH is in the coverage of previous IP domain's BS; as it enters the overlapping area while moving towards a BS belonging to a new IP domain, the MH can obtain a new IP address from the new domain, while the CN can still reach the MH using the previous IP address. The MH then notifies the CN about the availability of the new IP address. When the CN finds out that the MH's new IP address should be used as the primary destination address, it begins sending data through the MH's new IP address.

This eliminates the infamous triangular routing problem encountered by Mobile IP. Note that the retransmitted packets from CN in this scheme should be also directed to the MH's new IP address since the old IP address is very likely not any more reachable because of the MH's movement.

In contrast to Mobile IP, there are no Home or Foreign agents in TraSH; the scheme, however, requires a location manager for the CN to locate the current position of the MH when a new association is to be set up by the CN. In addition to the University of Oklahoma researchers, similar schemes are being explored by groups at ETRI (Korea), Technical University of Berlin (Germany), Georgia Institute of Technology (USA), and Siemens. They are all based on the use SCTP's multi-homing feature to assist in data transfer during the handover process.

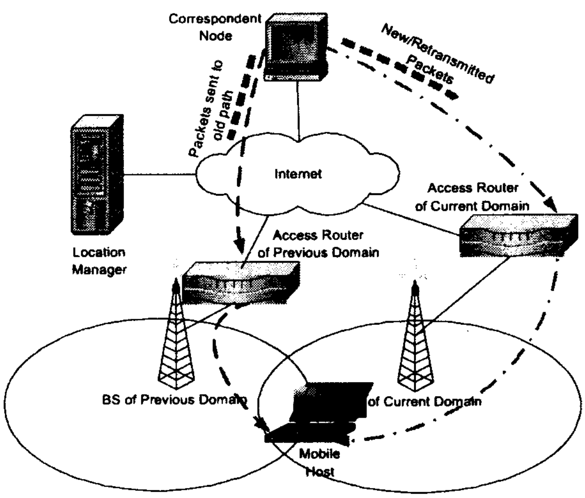

FIG. 6.4. *Handover in TraSH using SCTP's multi-homing*

TraSH based mobile handover has many potential advantages. For instance, TraSH does not require any infrastructure modifications in wireless base stations or Internet routers; it can achieve a seamless handover to reduce the packet loss and handover latency, thus improve QoS perceived by users; TraSH can also inter-operate with existing Internet security mechanisms like IPSec, ingress filtering, firewall, etc.

6.5. SCTP over wireless Ad-hoc networks. SCTP's performance in wireless Ad-hoc networks in the context of IEEE 802.11 WLAN was studied by Ye et al. [40]. One important 802.11 parameter investigated was the RTS threshold. Before sending data frames with sizes larger than the RTS threshold, the exchange of control frames (RTS/CTS sequence) is required. Generally speaking, a large RTS threshold will result in a high collision rate, whereas a small value will incur high signalling cost since virtually every data packet needs to use RTS/CTS signalling. Using the string simulation topology in Fig. 6.5 (where the dashed lines denote the radio coverage range), the authors have shown that the throughput of SCTP association degrades when the number of hops between the sender and receiver increases, mainly due to the hidden node and exposed node problems. The simulation results also show that when the hop count is less that three, the use of a low RTS threshold will reduce the collision occurring between SACK packets and RTS for DATA packets.

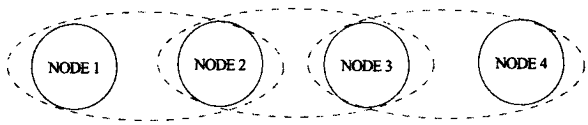

FIG. 6.5. *Simulation topology for SCTP over multi-hop networks*

The "small window syndrome" (SWS) that happens when the SCTP receiver window is too small was also illustrated in the above paper. When SWS happens, the sender can't get enough DupACKs to trigger a fast retransmit, and therefore, must wait for a coarse timeout. Thus, the SCTP sender will experience a long idle period. By assuming that most of the data losses are caused by the MAC layer collision instead of wireless random loss or network congestion, the authors proposed to transmit the data packets with the lowest unreceived TSN (reported in the SACK Gap Block) during the idle period. This algorithm can partially overcome the SWS problem, and speedup the error recovery caused by MAC collisions, at the risk of pumping more data into an already congested network when the above assumption is not valid.

To summarize, the research endeavors in SCTP over wireless networks are aiming at exploiting SCTP's current capabilities, or designing new features that can make SCTP more suitable for wireless channels and mobile scenarios arising from 3G and beyond wireless networks.

7. Summary. TCP is currently the dominant transport control protocol in the Internet. However, TCP was initially designed to perform well in networks with reliable wired links and stationary hosts. A number of characteristics of mobile wireless networks cause serious performance degradation of TCP. We have discussed a range of enhancements to the TCP protocol for wireless mobile networks, and compared them to give readers a clear picture on current status of this area.

A new emerging transport protocol, called SCTP, has also also discussed in this chapter. The initial aim of SCTP was to provide a transport protocol for transmitting SS7 signalling over an IP network. However, its attractive features have made it also useful for a wider range of applications. The main features and several possible application areas of SCTP are reviewed to give readers an alternative view on transport layer design in mobile wireless networks.

REFERENCES

[1] *The Network Simulator - ns-2*, http://www.isi.edu/nsnam/ns/.

[2] I. F. AKYILDIZ, G. MORABITO, AND S. PALAZZO, TCP-Peach: A new congestion control scheme for satellite IP networks, *IEEE/ACM Transactions on Networking*, 9 (2001), pp. 307–321.

[3] R. ALAMGIR, M. ATIQUZZAMAN, AND W. IVANCIC, Effect of congestion control on the performance of TCP and SCTP over satellite networks, in *Proceedings of the NASA Earth Science Technology Conference*, Pasadena, CA, June 2002.

[4] M. ALLMAN, H. BALAKRISHNAN, AND S. FLOYD, *Enhancing TCP's loss recovery using limited transmit.* RFC 3042, January 2001.

[5] M. ATIQUZZAMAN AND W. IVANCIC, Evaluation of SCTP multistreaming over wireless/satellite links, in *Proceedings of the 12th International Conference on Computer Communications and Networks*, Dallas, Texas, October 2003, pp. 591–594.

[6] H. BAI AND M. ATIQUZZAMAN, Enhancing TCP throughput over lossy links using ECN-capable RED gateway, in *Proc. IEEE VTC*, Orlando, FL, October 2003.

[7] H. Bai and M. Atiquzzaman, Error modeling schemes for fading channels in wireless communications: A survey, *IEEE Communications Surveys and Tutorials*, 5 (2003), pp. 2–9.

[8] A. Bakre and B. R. Badrinath, I-TCP: Indirect TCP for mobile hosts, in *Proceedings of the15th International Conference on Distributed Computing Systems*, Vancouver, Canada, June 1995, pp. 136–143.

[9] H. Balakrishnan and R. Katz, Explicit loss notification and wireless web performance, in *Proceedings of the IEEE Globecom Internet Mini Conference*, Sydney, Australia, November 1998.

[10] H. Balakrishnan, S. Seshan, E. Amir, and R. H. Katz, A comparison for improving TCP/IP performance over wireless links, in *Proceedings of the ACM SIGCOMM*, Palo Alto, CA, August 1996, pp. 256–269.

[11] H. Balakrishnan, S. Seshan, and R. Katz, Improving reliable transport and handoff performance in cellular wireless networks, *ACM Wireless Networks*, 1 (1995), pp. 469–481.

[12] K. Brown and S. Singh, M-TCP: TCP for mobile cellular networks, *Computer Communication Review*, 27 (1997), pp. 19–43.

[13] A. Caro, J. Iyengar, and P. A. et al, SCTP: a proposed standard for robust Internet data transport, *IEEE Computer*, 36 (2003), pp. 56–63.

[14] S. Cen, P. C. Cosman, and G. M. Voelker, End-to-end differentiation of congestion and wireless losses, *IEEE/ACM Transactions on Networking*, 11 (2003), pp. 703–717.

[15] C. P. editor, *IP Mobility Support*. IETF RFC 3344, August 2002.

[16] M. M. et al., *TCP selective acknowledgment options*. IETF RFC 2018, October 1996.

[17] S. Fu and M. Atiquzzaman, Improving end-to-end throughput of Mobile IP using SCTP, in *Proceedings of the Workshop on High Performance Switching and Routing*, Torino, Italy, June 2003, pp. 171–176.

[18] ———, SCTP: State of the art in research, products, and technical challenges, *IEEE Communication Magazine*, vol. 42, no. 4, pp. 64-76, April 2004.

[19] S. Fu, M. Atiquzzaman, and W. Ivancic, Effect of delay spike on SCTP, TCP Reno, and Eifel in a wireless mobile environment, in *Proceedings of the 11th International Conference on Computer Communications and Networks*, Miami, FL, October 2002, pp. 575–578.

[20] ———, Sctp over satellite networks, in *Proceedings of the IEEE 18th Annual Workshop on Computer Communications*, Dana Point, California, October 2003, pp. 112–116.

[21] S. Fu, M. Atiquzzaman, J. Jones, Y. Lee, S. Lu, and L. Ma, *TraSH: A* **Tr**a*nsport layer* **S**e*amless* **H**a*ndover scheme*, tech. report, Computer Science, University of Oklahoma, November 2003, http: www.cs.ou.edu/˜atiq.

[22] S. Fu, M. Atiquzzaman, L. Ma, and Y. Lee, "Signaling cost and performance of TraSH: A seamless handover scheme for data networks," Submitted to *Journal of Wireless Communications and Mobile Computing.*

[23] S. Goel and D. Sanghi, Improving performance of TCP over wireless links, in *Proceedings of the IEEE TENCON*, December 1998, pp. 332–335.

[24] T. Goff, J. Moronski, D. S. Phatak, and V. Gupta, Freeze-TCP: A true end-to-end enhancement mechanism for mobile environments, in *Proceedings of the IEEE INFOCOM*, Tel Aviv, Israel, March 2000, pp. 1537–1545.

[25] A. Gurtov, *Effect of delays on TCP performance*, in IFIP Personal Wireless Communications, August 2001.

[26] A. Gurtov and R. Ludwig, *Making TCP robust against delay spikes*. Internet Draft, draft-gurtov-tsvwg-tcp-delay-spikes-00.txt, February 2002.

[27] M. Hassan, A. Nayandoro, and M. Atiquzzaman, Internet telephony: services, technical challenges, and products, *IEEE Communications Magazine*, 38 (2000), pp. 96 –103.

[28] J. C. Hoe, Improving the start-up behavior of a congestion control scheme for TCP, in *SIGCOMM*, California, USA, 1996, ACM, pp. 270–280.

[29] H. Inamura, G. Montenegro, R. Ludwig, A. Gurtov, and F. Khafizov, *TCP over second (2.5G) and third (3G) generation wireless networks*, RFC 3481, February 2003.

[30] R. Ludwig, *The TCP retransmit (rxt) flag*. Internet Draft, draft-ludwig-tsvwg-tcp-rxt-flag-02.txt, November 2001.

[31] R. Ludwig and R. H. Katz, The Eifel algorithm: Making TCP robust against spurious retransmission, *ACM Computer Communications Review, 30 (2000)*, pp. 30–36.

[32] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, *TCP selective acknowledgment options*. RFC 2018, October 1996.

[33] P. Sinha, N. Venkitaraman, R. Sivakumar, and V. Bharghavan, WTCP: A reliable transport protocol for wireless wide-area networks, in *Proc. ACM/IEEE MOBICOM*, Seattle, WA, August 1999, pp. 231–241.

[34] W. R. Stevens, *TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithm.* RFC 2001, January 1997.

[35] R. Stewart and Q. X. et al., *Stream control transmission protocol.* IETF RFC 2960, October 2000.

[36] R. Stewart and C. Metz, SCTP: New transport protocol for TCP/IP, *IEEE Internet Computing*, 5 (2001), pp. 64–69.

[37] N. H. Vaidya, M. Mehta, C. Perkins, and G. Montenegro, *Delayer duplicate acknowledgements: A TCP-Unaware approach to improve performance of TCP over wireless*, Technical report 99-003, Computer Science, Texas A&M University, February 1999.

[38] S. Y. Wang and H. T. Kung, Use of TCP decoupling in improving TCP performance over wireless networks, *ACM Wireless Networks*, 7 (2001), pp. 221–236.

[39] G. Xylomenos, G. C. Polyzos, P. Mahonen, and M. Saaranen, TCP performance issues over wireless links, *IEEE Communication Magazine*, 39 (2001), pp. 52–58.

[40] G. Ye, T. Saadawi, and M. Lee, SCTP congestion control performance in wireless multi-hop networks, in *MILCOM2002*, Anaheim, California, October 2002, pp. 934–939.