Performance of SCTP in Wi-Fi and WiMAX networks with multi-homed mobiles

Vicuña Nelson
INRIA and LIA/Université d'Avignon
339, chemin des Meinajaries, 84911 Avignon, France
nelson.vicuna@etd.univ-avignon.fr

Jiménez Tania
LIA/Université d'Avignon
339, chemin des Meinajaries, 84911 Avignon, France
tania.jimenez@univ-avignon.fr

Hayel Yezekael
LIA/Université d'Avignon
339, chemin des Meinajaries, 84911 Avignon, France
yezekael.hayel@univ-avignon.fr

ABSTRACT

This paper provides an exhaustive performance analysis by simulation of the SCTP transfer protocol in WiMAX and Wi-Fi networks. We provide also a comparison of SCTP with both transfer protocols UDP for VoIP applications and TCP for FTP sessions, as SCTP can support these two types (elastic and non-elastic) of traffic. Finally, we study how SCTP performs when a mobile is multi-homed, i.e. connected simultaneously to two wireless networks (Wi-Fi and WiMAX).

Categories and Subject Descriptors

C.2.5 [**Computer-Communication Networks**]: Local and Wide-Area Networks; C.2.6 [**Computer-Communica-tions Networks**]: Standards; C.4 [**Performance of Systems**]: Performance attributes

General Terms

Performance

Keywords

Wireless, SCTP, NS-2, VoIP, Multi-homing.

1. INTRODUCTION

The Stream Control Transmission Protocol (SCTP) has similar congestion control and retransmission mechanisms to those of TCP, which were designed for wired networks.

Wireless networks have some particularities that can create problems when adaptive protocols as TCP and SCTP are used.

We study the performance of SCTP in wireless networks to see the behavior of this protocol with different parameters and compare it with other transport protocols, like TCP and UDP. We used two types of traffic: elastic (i.e, FTP transfers or HTTP) and non-elastic (VoIP).

We use NS-2 to study SCTP through extensive simulations. We compare the behavior of the three transport protocols (SCTP, UDP, and TCP) over three different technologies (Wired, IEEE 802.11 and IEEE 802.16) as well as multi-homing between the two wireless technologies.

Flexibility and diversity of modules in NS-2 allows us to design heterogeneous scenarios evaluating their performance. We made the necessary adjustments in the source code to achieve the interconnection of the modules and operation between them. Modules like SCTP of Protocol Engineering Laboratory (PEL) at the University of Delaware, WiMAX and Wi-Fi extentions developed at the National Institute of Standards and Technology (NIST) of the United States, have been used at the same time with different kinds of traffic and topologies to do the performance evaluation study.

The rest of this paper is organized as follows. In section 2, we have a short description of SCTP with an emphasis on the differences it has with respect to TCP and UDP. In section 3, the simulation scenarii are explained and in section 4 the performance analysis is presented. In section 5, we provide an overview of the related work in SCTP performance evaluation and compare it with our work. Finally, section 6 concludes the paper.

2. SCTP

SCTP is a transport protocol defined in RFC4960 [14]. It was designed by the Signaling Transport (SIGTRAN) group of the Internet Engineering Task Force (IETF). Initially, it was introduced to serve as a reliable signaling and control transport protocol for telecommunications traffic running over IP networks via a number of proposed adaptation layers, but has since evolved for more general use to satisfy the needs of applications that require a message-oriented protocol with all the necessary TCP-like mechanisms [4]. SCTP provides sequencing, flow control, reliability and full-duplex data transfer like TCP. In addition, SCTP has unique features including *multi-homing* and *multi-streaming*. Based on these two features, SCTP was originally designed to provide a reliable transport between two end hosts using multiple, independent control of streams. SCTP belongs to the transport layer in the IP architecture, like TCP and UDP.

SCTP is also richer in functionality and more tolerant to network and component failures than TCP [15]. Like TCP,

WNS2, October 24, 2008, Athens, Greece

Protocol Feature	SCTP	TCP	UDP
State required at each endpoint	yes	yes	no[1]
Reliable data transfer	yes	yes	no
Congestion control and avoidance	yes	yes	no
Message boundary conservation	yes	no[2]	yes
Path MTU discovery and message fragmentation	yes	yes[2]	no
Message bundling	yes	yes[2]	no
Multi-homed hosts support	yes	no	no
Multi-stream support	yes	no	no
Unordered data delivery	yes	no	yes
Security cookie against SYN flood attack	yes	no	no
Built-in heartbeat (reachability check)	yes	no[3]	no

Table 1: Comparison of SCTP, TCP, and UDP[15]

SCTP provides a reliable, full-duplex connection and mechanisms to control network congestion. Unlike both TCP and UDP, SCTP offers new delivery options that are particularly desirable for telephony signaling and multimedia applications. Table 1 compares SCTP's services and features with those of TCP and UDP.

An SCTP connection (association) provides novel services such as multi-homing, which allows the end points of a single association to have multiple IP addresses, and multi-streaming, that allows for independent delivery among data streams. At the bottom of the Figure 1, we can see an architecture that includes two network interfaces per host. Two paths are provided through the independent networks, these two paths would be collected into an association. At the top is a TCP connection. Each host includes a single network interface; a connection is created between a single interface on each node. Upon establishment, the connection is bound to each interface.

2.1 SCTP Congestion Control

The congestion control algorithms used by SCTP are based on TCP Congestion Control described in RFC2581 [1]. SCTP congestion control is always applied to the entire association, and not to individual streams.

The advanced congestion control mechanism of SCTP consists of slow-start, congestion avoidance and fast retransmit algorithms. The endpoints maintain three variables to regulate data transmission rate: receiver advertised ($rwnd$), congestion window ($cwnd$) and slow-start threshold ($ssthresh$). SCTP requires an additional control variable, $partial\_bytes\_acked$ (pba), that is used during congestion avoidance to facilitate $cwnd$ adjustment [14].

[1]With UDP a node can communicate with another node without going through a setup procedure or changing any state information. This is sometimes called connection-less, but in reality each UDP packet has the needed state within it to form a connection so that no ongoing state needs to be maintained at each endpoint.

[2]Because TCP treats all the data passed from its upper layer as a formatless stream of data bytes, it does not preserve any message boundaries. However, due to its byte-stream-based nature, TCP can automatically resize all the data into new TCP segments suitable for the Path MTU before transmitting them.

[3]Most TCP implementations do implement a "keep-alive" mechanism. This mechanism is very similar to the SCTP heartbeat, with the main difference being the time interval used. In TCP the "keep-alive" interval is, by default, set to two hours. The goal of this "keep-alive" is long-term state cleanup, which is in sharp contrast to SCTP's much more rapid heartbeat, which is used to aid in fast failover.

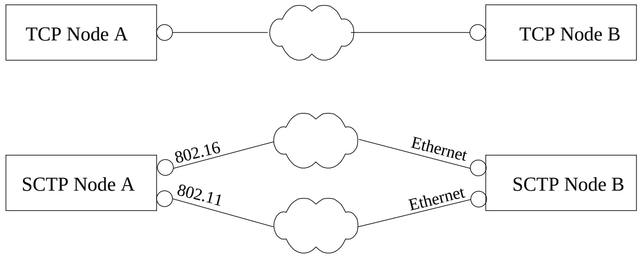


Figure 1: An SCTP Association vs. a TCP Connection

SCTP assigns a Transmission Sequence Number (TSN) to each user data fragment or unfragmented message. The TSN is independent of any Stream Sequence Number assigned at the stream level. The receiving end acknowledges all TSNs received (SACK), even if there are gaps in the sequence. In this way, reliable delivery is kept functionally separate from sequenced stream delivery. Each SACK acknowledges the Cumulative TSN, i.e., all the data chunks received before a break in the sequence, and can also contain one or more Gap ACK Blocks. They acknowledge a subsequence of TSNs received following a break in the sequence of received TSNs. By definition, all TSNs acknowledged by Gap ACK Blocks are greater than the value of the Cumulative TSN ACK.

Slow-start. The slow-start algorithm is used to probe the network in order to determine the available capacity at the beginning of a transfer, or after repairing loss detected by the retransmission timer. The idea is that the $cwnd$ is initially fixed to at most twice the value of the MTU of the address.

However, usually the network is able to carry much more than such quantity without major efforts. So, during the slow-start phase, when a SACK chunk is received, the value of $cwnd$ is increased by the total size of the acknowledged DATA chunks. The result is that $cwnd$ increases exponentially, doubling every RTT. The complete rules are a little bit more complicated, and can be check in section 7.2.1 of [14].

Congestion Avoidance. When $cwnd$ reaches the value of $ssthresh$, SCTP changes its behavior to the congestion avoidance algorithm. In this phase, the $cwnd$ is increased by at most one MTU per RTT, so it grows linearly. Again, the complete rules are written in section 7.2.2 of [14]. During congestion avoidance of SCTP, $cwnd$ can only be increased when the full $cwnd$ is utilized.

Fast Retransmit. To palliate the effects of a single packet drop, another algorithm called fast retransmit is used. It consist to retransmit the DATA chunk i when the SACKs show that several other DATA chunks sent after DATA chunk i have already arrived to the destination, while the DATA chunk i is still unacknowledged. In this way we can avoid the time-out of the retransmission timer.

In TCP, a data segment is fast retransmitted upon the arrival of 3 duplicate ACKs [1]. Due to the use of Delayed

ACKs (only used when there are no gaps in the incoming data), a data segment is fast retransmitted when the data receiver has gotten 3 or 4 later segments. This algorithm was defined for TCP before the use of selective acknowledgement was widely deployed. So, in SCTP, due to its compulsory use of Gap ACK Blocks, the algorithm is slightly different: if a TSN is not acknowledged in 4 consecutive received SACKs while any other newer TSN is acknowledged in any Gap ACK Block of those 4 SACKs, the TSN must be retransmitted. Moreover, both *cwnd* and *ssthresh* variables are set to one half of the value of *cwnd* in the moment of the fast retransmission.

Fast Recovery. The fourth algorithm used for congestion control is called fast recovery, also defined in [1] and used right after a fast retransmission. TCP without the Selective Acknowledgement option can not inform the data sender about anything else but the last data segment received in order. This implies many duplicated acks before that the selective ack arrives to destination and then a big advance in the Acknowledgement Number when received. In order to cope with this problem TCP can anticipate it already increasing the *cwnd* when the duplicate acknowledgements are still arriving, and this is basically the fast recovery algorithm.

SCTP, however, does not need that algorithm due to its use of Gap ACK Blocks, so the problem is elegantly solved.

2.2 Differences between SCTP and TCP

Gap ACK Blocks in the SCTP SACK carry the same semantic meaning as the TCP SACK in [12]. TCP considers the information carried in the SACK as advisory information only. SCTP considers the information carried in the Gap ACK Blocks in the SACK chunk as advisory. In SCTP, any DATA chunk that has been acknowledged by SACK is not considered fully delivered until the Cumulative TSN ACK Point passes the TSN of the DATA chunk. Consequently, the value of *cwnd* controls the amount of outstanding data, rather than (as in the case of non-SACK TCP) the upper bound between the highest acknowledged sequence number and the latest DATA chunk that can be sent within the congestion window. SCTP SACK leads to different implementations of Fast Retransmit and Fast Recovery than those in non-SACK TCP [14].

The major differences between SCTP and TCP congestion control algorithms are:

1. *cwnd*, the initial congestion window, is suggested to be $2*MTU$ in SCTP, which is usually one MTU in TCP.

2. In SCTP, the increase of the *cwnd* is controlled by the number of acknowledged bytes; while in TCP, it is controlled in general by the number of new acknowledgement received.

3. SCTP is required to be in slow-start phase when the slow-start threshold, *ssthresh*, is equal to the *cwnd*. It is optional in TCP to be either in the slow-start phase or in the congestion avoidance phase when the *ssthresh* is equal to the *cwnd*. In NS-2 when the slow-start threshold is equal to the *cwnd* the congestion avoidance phase is used.

4. In SCTP, fast retransmission is triggered by the fourth missing report of a chunk. This implies that at least $5*MTU$ of the received side window is required to trigger fast retransmission; while in TCP, the minimum receiver side window for fast retransmission is $4*MTU$ (three duplicate ACKs trigger fast retransmission).

5. SCTP has no explicit fast recovery algorithm that is used, in contrast to TCP. In SCTP the parameter $MaxBurst$ is used after the fast retransmissions to avoid flooding the network. $MaxBurst$ limits the number of SCTP packets that may be sent after processing the SACK, which acknowledges the data chunk that has been fast retransmitted.

2.3 SCTP Multi-homing

TCP involves one source and one destination IP address during the connection. It means that even if the TCP sender or receiver contains more than one physical address with multiple IP address, only one of these IP addresses will be used. On the other hand a SCTP association supports multi-homed hosts. In this fault tolerant approach, when one path fails, another interface can be used for data delivery without interruption.

Multi-homing allows two end points to setup an association with multiple addresses for each end point. During association initialization, each end point lists its IP addresses as well as its port number. Hence, the SCTP sender or receiver has a list of transport addresses that share the same port number. The SCTP sender selects a primary destination address and transmits all data chunks through this address and the rest of the addresses are considered as alternate destination addresses. This built-in support for multi-homed endpoints allows high availability applications to perform switch over to an alternate destination address without interrupting the data transfer during link failure situation.

Multiple active interfaces also suggest the simultaneous existence of multiple paths between the multi-homed hosts. Currently, due to lack of research in Concurrent Multipath Transfer (CMT), RFC4960 does not allow a sender to simultaneously send new data on multiple paths; an SCTP sender maintains a primary destination to which all transmissions of new data are sent (Note: retransmissions are sent to alternate destinations). In this paper, in the case of multi-homing, we use these multiple paths between multi-homed source and destination hosts through CMT proposed by [8] to increase throughput for a network application. CMT is the simultaneous transfer of new data from a source host to a destination host via two or more end-to-end paths.

2.4 SCTP Multi-streaming

The multi-streaming feature separates and transmits user data on multiple SCTP streams. These streams are capable of independent and sequenced delivery. If message loss occurs in one stream, other streams are unaffected, that way, SCTP eliminates unnecessary blocking.

A stream in TCP is a sequence of bytes that affirms the delivery in strict sequence. The negative effect of this sequence delivery is that the bypass among streams is not permissible. But in SCTP, the stream can be bypassed upon prioritization.

3. SIMULATIONS

We used four different scenario to evaluate the performance of SCTP. All of them have clients in one side, and servers in the other side, and share a bottleneck link. The main difference is the technology used for the connection of the clients to the bottleneck.

1. A Wired scenario is proposed as reference. Both TCP and SCTP were designed for wired networks. We want to know the behavior of both protocols in its natural scenario.

2. IEEE 802.11 (Wi-Fi)

3. IEEE 802.16 (WiMAX)

4. Multi-homing Wi-Fi and WiMAX

The parameters used for the simulation are the following:

- Three types of traffic:
  - Elastic Traffic, which are File Transfer Protocol (FTP) flows. The size of downloaded files is generated according to a Pareto distribution with average of 80KB (small files or mouses) and 800KB (big files or elephants) and 1.18 of shape; The inter-request time per client is exponential with average of 90 seconds.
  - Non-elastic Traffic, is represented by Voice over IP (VoIP) data traffic: Bi-directional voice traffic generated on the basis of G.729. The holding time is exponentially distributed with an average of 30 seconds for short calls and 300 seconds for long calls. The interval between two calls is exponentially distributed with average of 60 seconds.
  - Noise (exogenous) on-off traffic: Exponentially distributed "on" and "off" durations with rate of 40Kbps per client during the "on" period. The "on" and "off" average durations are 100ms.
- Simulation Time: 3600 seconds
- Client Data Rate: 100 Mbps (Wired Network), 54 Mbps (Wi-Fi)
- Server Data Rate: 100 Mbps
- Routing protocol for wireless topologies: No Ad-Hoc Routing Agent (NOAH).
- The path MTU: 1500 bytes
- SCTP associations data chunk: 1468 bytes.

Three similar topologies have been used for simulating the proposed scenario. An initial topology for the wired network (see Fig. 2), another for wireless networks Wi-Fi and WiMAX (see Fig. 3), and another one for multi-homing wireless networks Wi-Fi and WiMAX (see Fig. 4).

3.1 Simulated Topologies

3.1.1 Wired Topology

We used the topology shown in Figure 2 to simulate a wired network. This topology supports n_1 FTP clients for TCP connections or SCTP associations, n_2 VoIP clients (UDP connections or SCTP associations) simultaneously and n_3 noise clients.

$N0_{i=1..3,j=0..n_i}$ are destination nodes, and $N1_{i=1..3,j=0..n_i}$ are the source nodes or servers. The link between nodes $n0$ and $n1$ is the bottleneck with a bandwidth of 2Mbps and 10ms of propagation delay. The clients and the servers, at each side of the bottleneck) act as a local area network. All the connections or associations have random RTT uniformly distributed between $42ms$ and $624ms$.

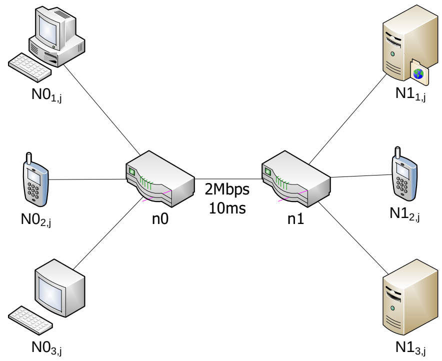


Figure 2: Wired Topology

3.1.2 Wireless Topology (802.11 and 802.16)

We used the topology shown in Figure 3 to simulate the wireless network. Like the previous topology, it supports n_1 TCP connections or SCTP associations simultaneously for FTP, n_2 UDP connections or SCTP associations simultaneously for VoIP data and n_3 noise clients.

$N0_{i=1..3,j=0..n_i}$ are destination nodes, and $N1_{i=1..3,j=0..n_i}$ are the source nodes or servers. BS node, is the base station to which client nodes are associated. The link between node BS and $n1$ is the bottleneck; the bandwidth of the link is 2Mbps and has 10ms of propagation delay. The wireless clients are uniformly distributed in the space. For the implementation in NS-2, we used a hierarchical structure based on 2 domains and one cluster for each domain.

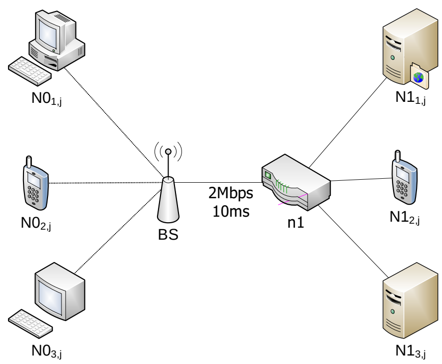


Figure 3: Wireless 802.11 and 802.16 Topology

Depending on the type of wireless networks, BS and the client nodes are configured to provide the physical environment adapted to Wi-Fi or WiMAX[1].

[1]For Wi-Fi and WiMAX implementation in NS-2, we used the module developed by the National Institute of Standards and Technology (NIST).

3.1.3 Wireless Multi-homing Topology (802.11 and 802.16)

We used the topology shown in Figure 4 to simulate the wireless multi-homing network. This topology supports n_1 SCTP associations simultaneously for FTP clients, n_2 SCTP associations simultaneously for VoIP data and n_3 noise clients.

$N0_{i=1..3,j=0..n_i}$ are destination nodes, and $N1_{i=1..3,j=0..n_i}$ are the source nodes or servers. Each $N0_{i,j}$ node has 2 interfaces, $N0WiFi_i$ and $N0WiMAX_i$; one in order to establish the connection with Wi-Fi and the other one for WiMAX. The $BSWiFi$ and $BSWiMAX$ nodes are the base stations to which client nodes are associated. The link between node $n0$ and $n1$ is the bottleneck link with a bandwidth of 2Mbps and 10ms of propagation delay. Wireless clients are uniformly distributed in the space. A hierarchical structure was used in the NS-2 implementation with three domains. One cluster for the servers nodes, one cluster for the multi-homing client nodes and three clusters for the access points and multi-homing interfaces of the clients nodes.

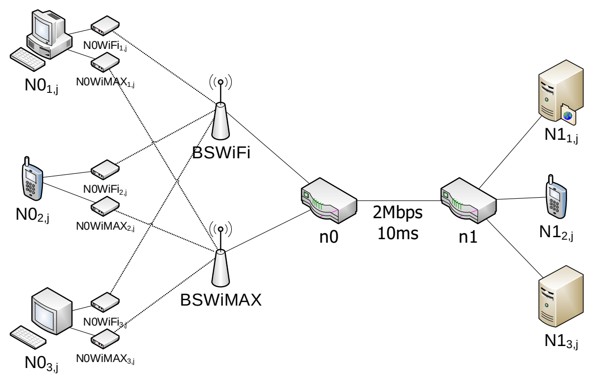


Figure 4: Wireless IEEE 802.11 and IEEE 802.16 Multi-homing topology

3.2 Performance Measures

We consider here three performance measures: Throughput, Delay, and Packet loss.

3.2.1 Throughput

We measure the average amount of data per second per client that is delivered over the bottleneck from the node $n1$ to the node $n0$.

3.2.2 Delay

We measure the average time that a packet of data takes in the queue of the bottleneck from the node $n1$ to the node $n0$.

3.2.3 Packet loss

Packet loss is due to network congestion. Packets are dropped in the bottleneck when the packet queue is full. We measure the rate of packets dropped in the bottleneck from the node $n1$ to the node $n0$.

3.3 Simulation scenarios

We simulate each transport protocol in each technology (Wired, Wi-Fi and WiMAX). We used TCP or SCTP for FTP and UDP or SCTP for VoIP data. We have hence the following combinations of protocols and applications:

- **TCP/UDP**: TCP for FTP and UDP for VoIP data,
- **TCP/SCTP**: TCP for FTP and SCTP for VoIP data,
- **SCTP/UDP**: SCTP for FTP and UDP for VoIP data,
- **SCTP/SCTP**: SCTP for FTP and SCTP for VoIP data.

In multi-homing Wireless Wi-Fi and WiMAX topology we used only SCTP as transport protocol for both applications, FTP and VoIP data (SCTP/SCTP).

We simulate each combination 10 times with different random seeds. The results showed are the average of these ten replications.

In the Table 2 we show the values of the parameters used for the different simulation scenario, for each combination of transport protocol in the different topologies.

In the scenario A the number of FTP clients varies from 0 to 50 by steps of 5 clients. The number of clients of VoIP varies from 0 to 50 in the scenario B also by steps of 5. We introduced noise traffic in scenario C and varied its number from 0 to 100 by steps of 10. These clients sent traffic in the ACK sense. In the scenario D we vary the ratio small/big file size, by steps of 10%. In the scenario E we vary the proportion of short duration calls and long duration calls by steps of 10% as well. In scenario F the buffer queue size varies from 5 to 50 by steps of 5. In the last scenario (G), we took as bandwidth of the bottleneck the values 250kbps, 500kbps, 1000kbps and 2000kbps.

4. PERFORMANCE EVALUATION

We compare in all the scenarios (A to G), the three performance measures: throughput, delay and packet loss.

4.1 Throughput for elastic traffic

We observe on figure 5(d) that as TCP is more aggressive than SCTP, the mean throughput of one FTP session is better with TCP in all topologies when the number of FTP clients is small (less than 20). Conversely, the SCTP protocol gives a better throughput when the number of FTP clients is high (more than 20). We explain this phenomenon by, in a low charge network, aggressiveness of TCP allows to obtain more bandwidth but, in a high charge network, this aggressiveness implies less bandwidth because there is too much packet drops (see figure 7(a)).

When the number of VoIP clients is important (see figure 5(e)), the UDP protocol for VoIP application overloads the Wi-Fi access channel and produces a fall of the TCP or SCTP throughput for FTP application. By using SCTP protocol for VoIP application, this phenomenon doesn't appear and the throughput of FTP application is stable as the number of VoIP clients increases. Moreover the packet drops and the mean delay for a VoIP client are not so much degraded using SCTP instead of UDP in every topology (see figure 6(a) for Wi-Fi topology and figure 6(d) for WiMAX topology).

Then we can conclude with this study on the throughput for elastic traffic, that SCTP is less aggressive when the network supports more and more traffic than the protocol UDP when it is used for VoIP application.

Simulation Scenario	FTP Clients	VoIP Clients	Noise Clients	FTP average File Size (KB)	VoIP average call duration (sec.)	Bottleneck Queue Size	Bottleneck Bandwidth (Kbps)
A	0-50	10	0	50% 80	50% 30	25	2000
				50% 800	50% 300		
B	10	0-50	0	50% 80	50% 30	25	2000
				50% 800	50% 300		
C	10	10	0-100	50% 80	50% 30	25	2000
				50% 800	50% 300		
D	10	10	0	0%-100% 800	50% 30	25	2000
				100%-0% 80	50% 300		
E	10	10	0	50% 80	0%-100% 300	25	2000
				50% 800	100%-0% 30		
F	10	10	0	50% 80	50% 30	5-50	2000
				50% 800	50% 300		
G	10	10	0	50% 80	50% 30	25	250-2000
				50% 800	50% 300		

Table 2: Values of parameters in each simulation scenario

4.2 Delay for non-elastic traffic

In both wireless topologies Wi-Fi (see figure 6(b)) and WiMAX (see figure 6(e)), the average per packet delay of VoIP application increases with the file size of FTP transfers. But we observe that this mean delay is lower when the non-elastic traffic uses SCTP instead of UDP. This difference is up to 30% less when all FTP transfer has 800 KB in the WiMAX topology and still 25% in the Wi-Fi one.

Considering the scenario E when the proportion of long call increases, the mean packet delay of VoIP application is apparently surprising because it decreases, in both wireless topologies Wi-Fi (see figure 6(c)) and WiMAX (see figure 6(f)). Indeed, the number of FTP clients is fixed to 10 and as the proportion of long VoIP session increases, more non-elastic packets are present in the queue proportionally to elastic traffic packet generated by FTP applications. Moreover, the number of VoIP simultaneous sessions increases. Then, the mean packet delay of VoIP application consequently decreases.

4.3 Drop for elastic traffic

Comparing the packet drops for elastic traffic with SCTP or TCP, we observe on figure 7(b) for Wi-Fi topology, that the packet drop is in fact increasing with the queue size when using TCP and decreasing when using SCTP. This comes from the aggressiveness of the slow-start congestion avoidance mechanism. Indeed, in TCP, the congestion window is doubled in terms of packets at each acknowledgement whereas in SCTP it is doubled in terms of bytes as seen in section 2.1. Moreover, the used of delayed ACK by SCTP will reduce the number of ACKs, which in turn slows the *cwnd* growth rate. This implies that more TCP packets are dropped during burst of losses, because a burst of data (i.e, a file transfer) can potentially cause a large amount of segment loss during the slow-start congestion avoidance phase than using SCTP.

This implies also that the throughput of FTP application is better using TCP than SCTP (see figure 5(c) in a Wired topology or figure 5(f) in a Wi-Fi topology). Thus there exists a compromise between loss and throughput between TCP and SCTP for elastic traffic.

4.4 Drop for non-elastic traffic

Comparing the packet drops for non-elastic traffic with SCTP or TCP, we observe in wireless Wi-Fi topology (see figure 8(b)) that the packet drop is less when we used UDP as transport protocol for VoIP and SCTP as transport protocol for FTP application. When the number of VoIP client is high (more than 20) we observe a decrease of the packet drops when we used TCP as transport protocol in FTP applications. This behavior is due to the Wi-Fi's contention-based access.

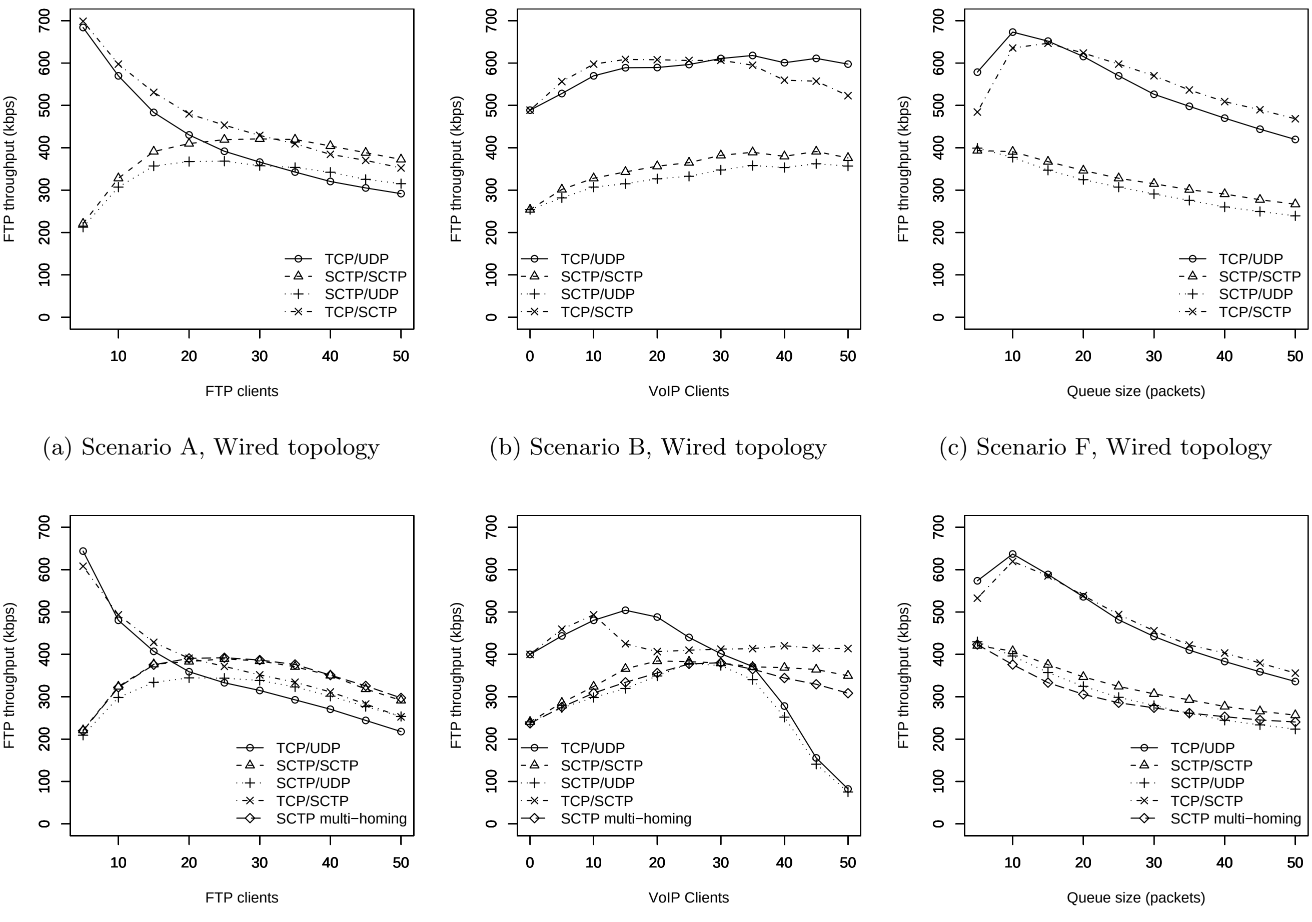
In the case of WiMAX (see figure 8(c)), UDP has a better behavior than SCTP as transport protocol of non-elastic traffic. Increasing VoIP clients, the increase in the percentage of losses is lower than when we used SCTP. In contrast to Wi-Fi, the MAC layer of WiMAX provides grant/request access, avoiding collisions, managing the resources of the wireless link in an efficient way.

4.5 Multi-homing

We observe on figures 6(e) and 6(f) that the non-elastic traffic delay is lower when each mobile is connected simultaneously to WiMAX and Wi-Fi (multi-homed) compared to a connection with a single wireless technology. Concurrent connections between two different wireless technologies allows mobiles to access also Wi-Fi which has better performance than WiMAX. Then a multi-homed mobile connected simultaneously to Wi-Fi and WiMAX has better performance (delay, drop and throughput) than connected to only WiMAX. We observe also that multi-homing does not perform well compared with only Wi-Fi (see figure 7(b) and 5(f) for examples). On the other hand, when the Wi-Fi access channel is overload (i.e, when the clients who sent traffic in the ack sense, are upper than 50, see figure 9(a)) the throughput in the multi-homing topology remains stable.

5. RELATED WORK

In the last few years, many studies have been done in evaluating the performance of many aspects of SCTP. For example, a study of the coexistence of SCTP and TCP in the Internet has shown that SCTP traffic is TCP friendly in the sense that it has the same impact on the congestion control of other TCP connections as normal TCP traffic [9]. This study is different than ours in two ways: i. it is an



(a) Scenario A, Wired topology (b) Scenario B, Wired topology (c) Scenario F, Wired topology

(d) Scenario A, Wi-Fi topology (e) Scenario B, Wi-Fi topology (f) Scenario F, Wi-Fi topology

Figure 5: Elastic traffic throughput

experimental study, constrained to a small number of clients and only wired technology; ii. The authors only use elastic traffic. We too observe in our simulations a TCP-friendly behavior of SCTP.

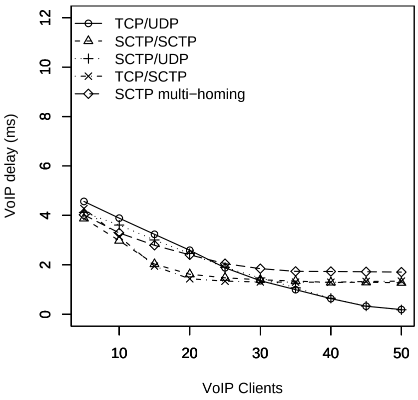
In [3] the authors focus on SCTP multi-streaming for reducing the latency of streaming multimedia in high-loss environments. They show that multi-streaming results in slower degradation in the network throughput as the loss rate increases than in TCP. Moreover, user satisfaction is increased with the improved multimedia quality provided by this feature. Similar results were obtained in our simulations when the loss rate increases (i.e, when we increase the number of FTP clients) despite using a single stream.

Using NS-2, in [10], the authors study the multi-streaming and the multi-homing SCTP features. They prove that these features have advantages over TCP in their scenario. They define the optimal number of streams in multi-streaming and explain how it affects network performance. In our work, multi-homing's advantage is observed when we have a high number of packets losses in a network (i.e, Wi-Fi) because the alternative pathway (i.e, WiMAX) minimize the impact of packets drops.

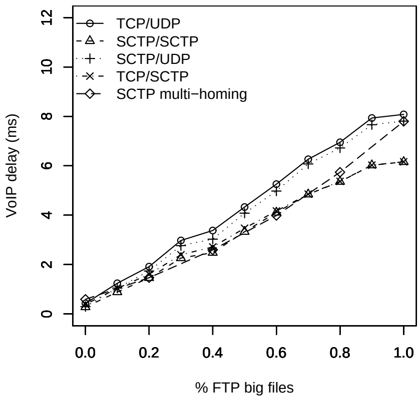
In [13] the authors compare the performance of SCTP and TCP with respect to Web traffic concluding that SCTP can help to reduce the latency and improve the throughput. This is also true in our scenario when the number of clients is larger than 20.

In the wireless networking area, the performance of SCTP in mobile networks [5] and wireless multi-hop networks [16] has been studied. The performance of SCTP in Mobile IP was investigated in [5] using NS-2 simulations, and it was shown that the support of a large number of SCTP SACK blocks result in better performance than TCP-Reno and TCP-SACK. In [16] the authors have shown that the throughput of an SCTP association degrades when the number of hops between the sender and receiver increases, mainly due to the hidden node and exposed node problems. These studies are different of our in the sense that we don't have a multihop scenario.

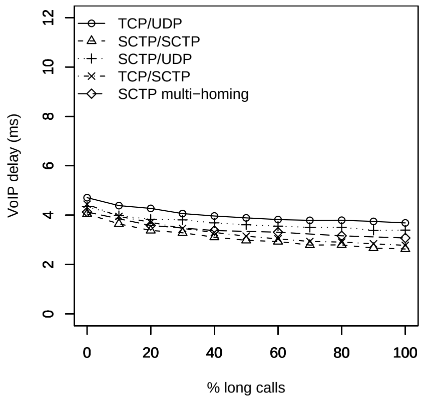
In [7] the authors presented a simulation study of delay spike of SCTP, TCP-Reno, and Eifel over wireless links. They found that Eifel performs better than TCP-Reno and SCTP when there are no packet losses. However, the opposite happens when packets are lost in the presence of delay spikes. Also they showed that a higher link bandwidth does not always increase the data throughput of SCTP, TCP-



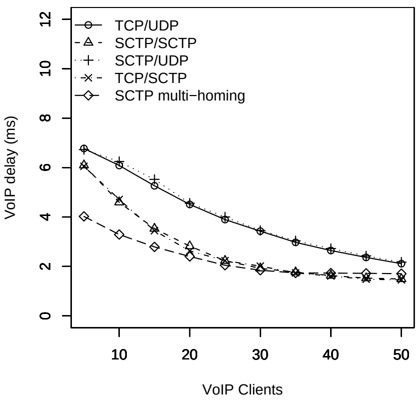
(a) Scenario B, Wi-Fi topology



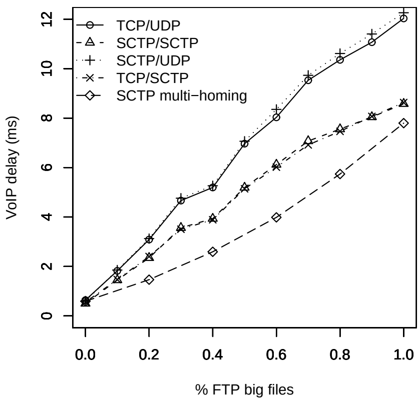
(b) Scenario D, Wi-Fi topology



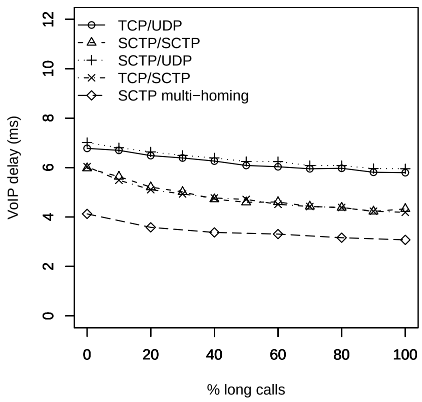
(c) Scenario E, Wi-Fi topology



(d) Scenario B, WiMAX topology



(e) Scenario D, WiMAX topology



(f) Scenario E, WiMAX topology

Figure 6: Non-elastic traffic delay

Reno, and Eifel.

In [11] the authors provide a simulation-based performance comparison of SCTP vs TCP in MANET environments. They found that SCTP and TCP have similar behavior in MANETs environment, but TCP outperforms SCTP in most cases because of the extra overhead present in SCTP. Certainly the size of the header is an important factor, especially when we use applications such as VoIP. SCTP header size is bigger than the header used by UDP. As a result, we have a greater use of resources by SCTP and therefore a non-optimal use of them.

In [2] the authors presented their simulation results regarding the performance of SCTP in a wireless ad-hoc network environment under two routing protocols: DSR and AODV. They proposed a set of modifications to the SCTP protocol for handling pro-actively route failures in mobile ad-hoc networks and they showed that the transport layer allows for faster path selection, in the case that a number of paths exist, leading thus to improve overall throughput. We used NOAH as routing protocol, but we didn't investigate the route failures.

In [17], the authors have shown that SCTP multi-homing can provide better throughput performance and more robustness in the wireless multi-access scenario, based on the Linux kernel experimentation. Similar results were obtained in our simulations studies.

In [6] the authors introduce the main features of SCTP and discuss the state of the art in SCTP research and development activities. They also provide a useful survey of the available products that use SCTP.

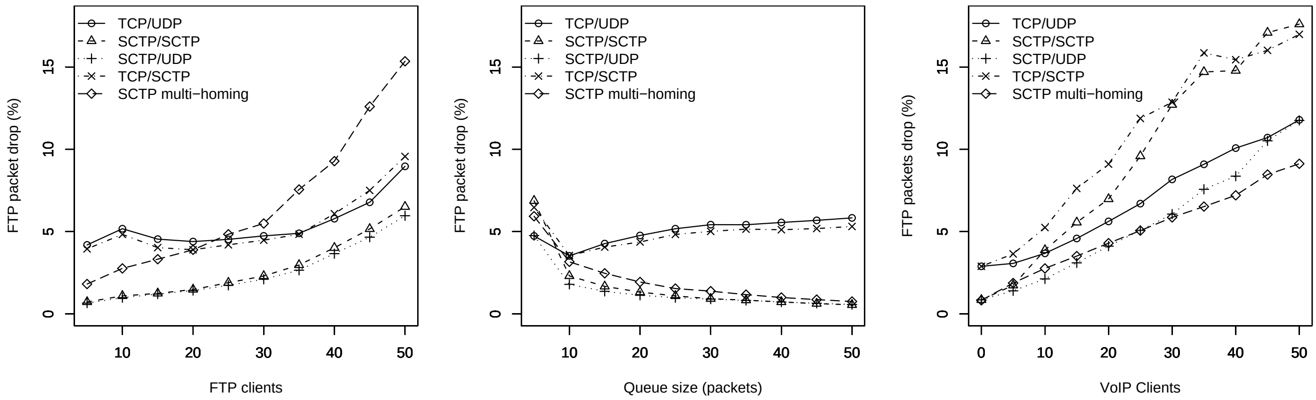
As far as we know, there is no reference on the use of SCTP over WiMAX.

In this article we give some initial ideas on the behavior of SCTP over WiMAX. Further work is being done with more emphasis on WiMAX.

6. CONCLUSIONS

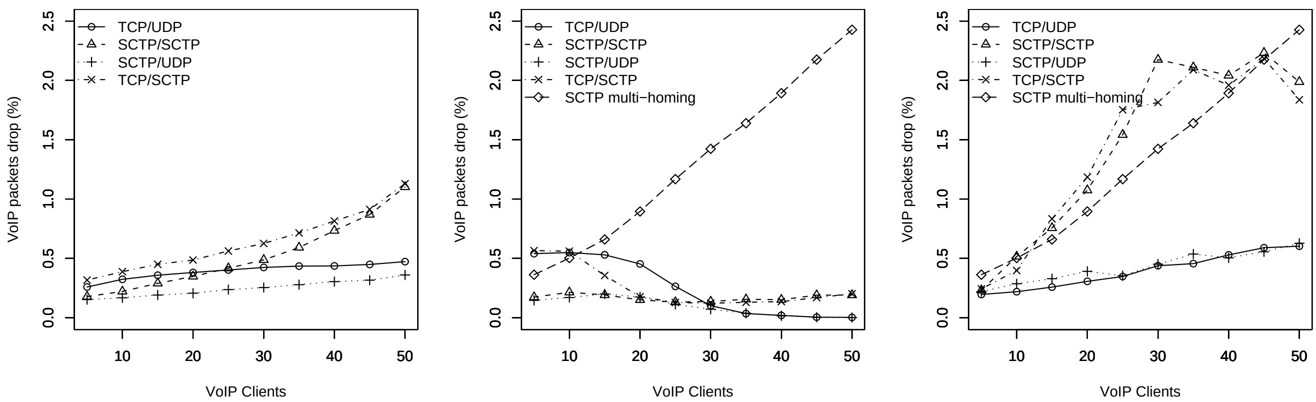
In this paper we presented a comparative study of SCTP with TCP over three different technologies: wired, Wi-Fi and WiMAX as well as multi-homing between the two wireless technologies. We simulate seven different scenarii in each technology, varying the parameters by small steps. In total we executed more than 80 different simulations. Each of them was executed ten times with different random seeds.

The different simulations proposed in this paper show a similar behavior between SCTP and TCP. However, TCP



(a) Scenario A, Wi-Fi topology (b) Scenario F, Wi-Fi topology (c) Scenario B, WiMAX

Figure 7: Elastic traffic packet drop



(a) Scenario B, Wired Topology (b) Scenario B, Wi-Fi Topology (c) Scenario B, WiMAX

Figure 8: Non-elastic traffic packet drop

is more aggressive handling the congestion window. SCTP congestion control was designed similar to that of TCP with the goal to assure that SCTP does not behave more aggressively than TCP. When there are few competing flows TCP has better throughput, because it opens the window much quicker than SCTP, and so it take the available bandwidth quicker.

On the other hand, when the number of flows is high, SCTP has better throughput than TCP. SCTP has smaller delay and packet loss than TCP which results in better performance in throughput, despite the lack of aggressiveness in handling the window congestion.

In non-elastic traffic, as VoIP, SCTP's behavior is as expected. However, the header of SCTP, is much larger than that of UDP, and hence consumes much more resources.

Contrary to what was expected with multi-homing, the use of CMT didn't improve the throughput of the primary link. However, when we observe a collapse of the primary interface, there is no degradation in the throughput due to the use of a second link. Multi-homing in this case, behaves as a backup mechanism as originally proposed in RFC2960.

This study enables us to identify interesting problems to explore in future work.

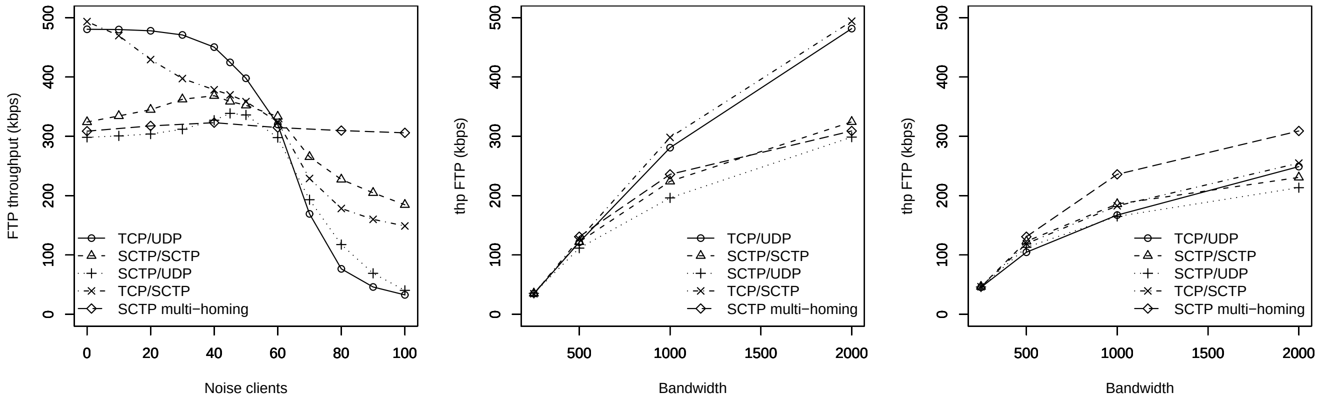
7. ACKNOWLEDGMENTS

The work of the first author was supported by the EuroFGI and EuroNF network of excellence. The work of the two other authors was partly supported by the French ANR project WiNEM: ANR-06-TCOM-05.

8. REFERENCES

[1] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control. RFC 2581 (Proposed Standard), Apr. 1999. Updated by RFC 3390.

[2] A. Argyriou and V. Madisetti. Performance evaluation and optimization of sctp in wireless ad-hoc networks. In *Proceedings of 28th Annual IEEE International Conference on Local Computer Networks*, pages 317–318, October 2003.



(a) Scenario C, Wi-Fi Topology (b) Scenario G, Wi-Fi Topology (c) Scenario G, WiMAX

Figure 9: Elastic traffic throughput

[3] A. Caro Jr and P. Amer. Improving Multimedia Performance over Lossy Networks via SCTP, 2000.

[4] L. Coene. Stream Control Transmission Protocol Applicability Statement. RFC 3257 (Informational), Apr. 2002.

[5] S. Fu and M. Atiquzzaman. Improving End-to-End Throughput of Mobile IP using SCTP. In *Wksp. High Perf Switching and Routing*, pages 171–176, Torino, Italy, June 2003.

[6] S. Fu and M. Atiquzzaman. SCTP: State of the Art in Research, Products, and Technical Challenge. *IEEE Communications Magazine*, 42(4):64–76, April 2004.

[7] 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 ICCCN2002*, pages 575–578, Miami, Florida, Octobre 2002.

[8] J. Iyengar, K. Shah, P. Amer, and R. Stewart. Concurrent multipath transfer using sctp multihoming. In *Proceedings of International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, July 2004.

[9] A. Jungmaier, M. Schopp, and M. Tüxen. Performance Evaluation of the Stream Control Transmission Protocol. In *Proceedings of the IEEE Conference on High Performance Switching and Routing*, pages 141–148, Heidelberg, Germany, June 2000.

[10] S. Kang and M. Fields. Experimental study of the sctp compared to tcp. Technical report, Department of Electrical Engineering of Texas AM University, 2003.

[11] A. Kumar, L. Jacob, and A. L. Ananda. Sctp vs tcp: Performance comparison in manets. In *Proceedings of 29th Annual IEEE International Conference on Local Computer Networks*, pages 431–432, November 2004.

[12] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. TCP Selective Acknowledgment Options. RFC 2018 (Proposed Standard), Oct. 1996.

[13] S. K. R. Rajamani and N. Gupta. Sctp versus tcp: Comparing the performance of transport protocols for web traffic. Technical report, Computer Sciences Department of University of Wisconsin Madison, July 2002.

[14] R. Stewart. Stream Control Transmission Protocol. RFC 4960 (Proposed Standard), Sept. 2007.

[15] R. Stewart and Q. Xie. *Stream Control Transmission Protocol (SCTP): A Reference Guide*. Addison-Wesley Professional, 2002.

[16] G. Ye, T. Saadawi, and M. Lee. SCTP Congestion Control Performance in Wireless Multi-hop Networks. In *MILCOM 2002*, pages 934–939, Anaheim, California, Octuber 2002.

[17] J. Yuehui and H. H. Dajiang. Experimental Performance Studies of SCTP in Wireless Access Networks. In *Proceedings of ICCT2003*, pages 392–395, Beijing, China, April 2003.