CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2007; **19**:1207–1218
Published online 3 April 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cpe.1159

SCTP performance evaluation over heterogeneous networks‡

Alberto Dainotti, Salvatore Loreto, Antonio Pescapé*,†
and Giorgio Ventre

Dipartimento di Informatica e Sistemistica, University of Napoli 'Federico II', Via Claudio 21, 80125 Napoli, Italy

SUMMARY

Since its definition in 2000, the Stream Control Transmission Protocol (SCTP) has attracted increasing interest. Several research works, often validated through analytical and simulative analysis, have attempted to evaluate the benefits of substituting TCP with SCTP, both for signaling and data transfer. In this work, we present a traffic generation and performance analysis tool to test SCTP on real networks. We study the performance of SCTP on real heterogeneous (wired/wireless) scenarios, providing results in terms of throughput and jitter, and comparing its performance against TCP and UDP over the same conditions. Our experimental analysis shows that the current performance of SCTP (operating on a Linux platform) does not justify the use of SCTP as a simple substitute for TCP. Copyright © 2007 John Wiley & Sons, Ltd.

Received 29 October 2006; Accepted 21 November 2006

KEY WORDS: SCTP; performance evaluation; active measurements; heterogeneous networks

1. INTRODUCTION

The Stream Control Transmission Protocol (SCTP) was originally designed to support public switched telephone network (PSTN) signaling messages over IP Networks. However, because many applications may gain advantages from its peculiarities, SCTP is now considered as a general-purpose transport protocol. Therefore, there is increasing interest in the performance of SCTP over real networks. In the last few years, several research efforts have been devoted to the study of both TCP and UDP performance over wired, wireless, and heterogeneous network scenarios. However, to date there has been a lack of studies in which the performance of SCTP is evaluated in real networks and compared

*Correspondence to: Antonio Pescapé, Dipartimento di Informatica e Sistemistica, University of Napoli 'Federico II', Via Claudio 21, 80125, Napoli, Italy.
†E-mail: pescape@unina.it
‡This paper represents an extended version of [1].

Contract/grant sponsor: MIUR (PRIN 2004 QUASAR Project)

with other protocols. Delay, jitter, loss, and throughput of network packets are fundamental parameters that affect how the quality of service (QoS) is perceived by the users of Internet applications. How their values change as the demands on an application vary depends on the network scenario, its conditions (nodes, links, etc.), and on the end-to-end transport protocol.

In this paper we propose an experimental analysis of the SCTP performance in several realistic wired, wireless, and heterogeneous network scenarios, by comparing it against UDP and TCP. The motivations for this work and related literature are discussed in Section 2, and in Section 3 we present a brief overview of the SCTP protocol. The analysis presented here is based on two well-known QoS parameters, throughput and jitter, and it has been carried out through D-ITG [2], a packet-level traffic generator and active measurement tool that we developed over the past few years, and to which we have recently added the SCTP support presented in this paper. In Section 4.1 we give details of the different network scenarios under which the experiments were conducted, while in Section 4.2 we describe the measurement approach we adopted, the software tools involved, and the test-bed used. In Section 5 we present the experimental results of our analysis by highlighting the most relevant aspects found for each different network scenario and the differences and similarities among different scenarios. Finally, in Section 6 we conclude the paper by discussing results and commenting on issues for future research.

2. MOTIVATIONS AND RELATED WORKS

There are several works in the literature that study SCTP performance by means of simulation, experimental tests, and analytical models. However, most of these studies are focused only on some specific protocol features, or do not examine protocol performance on different/heterogeneous network scenarios. In the case of wired scenarios, Nagamalai and Lee [3] present results related to the experimentation of SCTP congestion control in high-speed WANs as a mechanism for bulk transfer. In [4] Rajamani *et al.* compare the performance of SCTP and TCP with respect to Web traffic, highlighting how the new transport protocol fits the Web traffic characteristics better than TCP. In [5], using ns-2, Kang and Fields study the *multi-streaming* and the *multi-homing* SCTP features. They prove that these features, as claimed by the protocol's designers, really do give advantages over TCP in the scenarios considered. In particular, they define the optimal number of streams in multi-streaming, and show how it affects network performance. In the case of wireless networks, in [6] Ma *et al.* developed an analytical model which takes into account the congestion window, the round trip time, the slow start, and congestion avoidance processes when predicting SCTP performance. By comparing numerical results from the analytical model with simulation results, they demonstrate that the proposed model is able to accurately predict SCTP throughput. Fu *et al.* [7] present a simulation study of the 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 as a result of the support provided by SCTP SACK in the early detection of the lost segments to be retransmitted. Furthermore, they also demonstrated that a higher link bandwidth does not always increase the data throughput of SCTP, TCP Reno, and Eifel. Kumar *et al.* [8] provide a simulation-based performance comparison of SCTP versus TCP in mobile *ad-hoc* network (MANET) environments. They found that SCTP and TCP have similar behavior, but that TCP outperforms SCTP in most cases owing to the extra overhead present in SCTP. The simulation confirms the expected

DOI: 10.1002/cpe

worse behavior when SCTP does not use multi-streaming and unordered delivery. Argyriou and Madisetti [9] present their simulation results regarding the performance of SCTP in a wireless *ad-hoc* network environment. They showed the advantages of SCTP multi-homing and heartbeat messages in the detection and recovery from path failures. Finally, Fu and Atiquzzaman [10] introduce the main features of SCTP and discuss the state of the art in SCTP research and development activities, as well as providing a useful survey of the available products that use SCTP. The lack of networking tools supporting SCTP was also noted during the course of this work. In the literature we found a lack of experimental studies focusing on the performance evaluation of real scenarios where a comparison among SCTP, TCP, and UDP is presented. Our work extends the results contained in literature in the following ways: (i) we introduce an open-source software tool supporting SCTP traffic generation; (ii) we provide an evaluation of a wide range of heterogeneous network scenarios in terms of throughput and jitter; (iii) by applying a per-packet analysis, we compare SCTP performance against TCP and UDP; (iv) we make the measurement data traces related to our experiments freely available to the research community.

3. A BRIEF OVERVIEW OF SCTP

SCTP was defined by the IETF in late 2000 [11]. SCTP has several features that make it comparable to TCP: it offers a reliable transport service (error-free and in-sequence data delivery), and it is a session-oriented transport protocol. Moreover, in common with TCP, SCTP is rate adaptive, and is indeed designed to behave cooperatively with TCP sessions, sharing the same bandwidth. SCTP also provides a number of new functionalities with respect to TCP and UDP. These new functionalities are critical for signaling in telephony applications, and have also proven to be useful for various other tasks. In contrast to TCP, which is byte oriented, SCTP is message oriented. An SCTP session is called association. The association establishment procedure relies on four-way handshaking, where data can be already included in the third and fourth message of the handshake, as these messages are sent when the association has already been validated. A cookie mechanism has also been incorporated into the handshake as a protection against some types of denial of service attacks. Among the most peculiar features of SCTP are multi-streaming and multi-homing. With multi-streaming the data transmitted within a single SCTP association can be partitioned into multiple streams. The delivery sequence of these streams is managed independently. It is also possible to independently set, within each stream, the ordered or unordered delivery. Packet loss in one stream does not affect the remaining streams, which helps to avoid the well-known TCP Head of Line Blocking problem. When the multi-streaming feature is used, the in-sequence delivery of the transmitted data can be ensured only within each stream, not within the whole association. At the same time, transport is performed within a single association, so that all of the streams are subject to a common flow and congestion control mechanism. SCTP *multi-homing* supports nodes multi-homing. Indeed, any endpoint of an SCTP association may support multiple IP addresses, exchanging lists of addresses during the initiation of the association. At each endpoint, for a specific session, a single port number is used across the entire address list. Despite the multiple IP addresses, only one is active for data exchange, while all of the others exchange only heartbeat packets, so that one of them can be activated in the case of a failure of the active IP. The SCTP packet is made of two main parts: a common header and one or more chunks. The common header contains the following elements: (i) source and destination port numbers; (ii) a 32-bit tag used to avoid

DOI: 10.1002/cpe

the insertion of out-of-date or false messages into the association; (iii) a 32-bit checksum for error detection. The chunks included in the message may be control chunks or data chunks. Each chunk contains the following information: (a) an indication of its type (data or control); (b) a flag, which in data chunks is used to control the segmentation and reassembly process, whereas in control chunks it may assume different meanings; (c) chunk length; and (d) a chunk value. In data chunks the value field contains the following information: (i) a TSN that is a unique sequence number within the association; (ii) a stream identifier that identifies the stream to which the following user data belongs; (iii) a stream sequence number that represents the stream sequence number of the following user data within the stream; (iv) a payload protocol identifier that represents the upper layer specified protocol identifier; (v) user data that is the payload user data. The SCTP message format natively supports bundling of multiple data and control chunks into a single message (controllable by the application), in order to improve transport efficiency.

4. MEASUREMENT SCENARIO

4.1. Network scenarios

Several different network scenarios have been considered (see Table I). To have a first look at both the performance of our implementation and of SCTP, and to trace a performance reference that may also be useful for people working on protocol performance improvements, in all of the considered scenarios we used two hosts: the first acting as a traffic source, the second acting as a traffic sink. In order to investigate the characteristics of the current implementation of SCTP under 'quasi-ideal' conditions, we first connected the two hosts via an Ethernet crossover cable (*eth2eth*). This network configuration obviously offers the highest throughput, the lowest loss rate, and the lowest delay among all of the scenarios we studied. Then, we considered two wireless scenarios. In the first (*w2w-ah*) the two hosts are directly connected in '*ad-hoc* mode'. In the second (*w2w-ap*) the two hosts communicate via an access point (AP). Finally, we evaluated SCTP in two wired-to-wireless scenarios. In these scenarios one host is connected via an Ethernet cable to an AP, whereas the other is connected to the AP via an 802.11 connection. We first considered the 'wired host' acting as a traffic source and the 'wireless host' acting as a traffic sink (*e2w*). Then we exchanged the roles of the two hosts, using the wireless host as a traffic source and the wired host as a traffic sink (*w2e*). We call these last two scenarios 'heterogeneous' because the two communicating hosts use totally different access network technologies. From a performance evaluation point of view it is very interesting to study such a mix: first, because it reproduces realistic scenarios; second, as will be shown in the following section, because it presents different behaviors from other scenarios.

4.2. Measurement approach

We evaluated protocol performance by adopting an active measurement approach. We injected synthetic traffic into our test-bed and measured throughput and jitter by means of probe traffic. This approach allows a high level of control over our experimental setup. More precisely, in order to limit the number of variable parameters that must be taken into account in our analysis, we generated constant traffic (that is constant packet rate (*PR*) and constant payload size (*PS*)) using UDP,

DOI: 10.1002/cpe

Table I. Description of the network scenarios.

Scenario name	Scenario description	Medium transmission rate	Traffic profile
eth2eth	Two host connected via an Ethernet crossover cable	100 Mbps	$PS = 512$ bytes $PR \in \{9766\ \ 12\,207\ \ 14\,648\ \ 17\,090\ \ 19\,531\ \ 20\,752\ \ 21\,973\ \ 23\,193\}$ pps
w2w-ah	Two wireless hosts connected in *ad-hoc* mode	11 Mbps	$PS \in \{32\ \ 64\ \ 128\ \ 256\ \ 512\ \ 1024\ \ 1460\}$ bytes $PR \in \{100\ \ 1000\ \ 10\,000\}$ pps
w2w-ap	Two wireless hosts connected via an access point (AP)	11 Mbps	$PS \in \{32\ \ 64\ \ 128\ \ 256\ \ 512\ \ 1024\ \ 1460\}$ bytes $PR \in \{100\ \ 1000\ \ 10\,000\}$ pps
e2w	Two hosts communicating via an AP. The first (traffic source) is connected to the AP via an Ethernet cable. The second by wireless.	11 Mbps	$PS \in \{32\ \ 64\ \ 128\ \ 256\ \ 512\ \ 1024\ \ 1460\}$ bytes $PR \in \{100\ \ 1000\ \ 10\,000\}$ pps
w2e	Similar to the *e2w* scenario. In this case the wireless connected host acts like a traffic source	11 Mbps	$PS \in \{32\ 64\ 128\ 256\ 512\ 1024\ 1460\}$ bytes $PR \in \{100\ 1000\ 10\,000\}$ pps

TCP, and SCTP. In all of the scenarios, except in *eth2eth*, we considered three working conditions: low, medium, and high packet rate in which, 100, 1000, and 10 000 packets per second (pps) were respectively sent. As for the packet size, we used a payload size ranging from 32 to 1460 bytes for the low and medium packet rate conditions. In the high packet rate condition, in order to not exceed the nominal throughput of the wireless channels, we instead considered only three packet sizes: 32, 64, and 128 bytes. As mentioned above, we examined *eth2eth* in order to evaluate the behavior of SCTP in a reference scenario characterized by high available bandwidth, low delay, and low jitter, and to test the protocol implementation. Therefore, in this scenario we fixed *PS* at 512 bytes and we varied *PR* in order to obtain different load conditions ranging from about 40 to about 95 Mbps. We used D-ITG [2] to generate traffic flows and to perform measurements (details on the hardware and operating systems employed in our test-bed are reported in [1]). D-ITG is a packet-level traffic generator we have developed over the last few years. To perform the study presented in this work we extended the current publicly available release, which is labeled 2.4, by adding SCTP support based on the LK-SCTP implementation [12]. The LK-SCTP project started in 2001, and it currently supports the Linux kernel 2.6 family. This implementation supports all of the SCTP standard features and is fully compliant with RFC 2960 [11] and RFC 4460 [13]. D-ITG can be used to analyze throughput and jitter at the packet level, and also to measure the *round trip time* delay and the *one-way* delay. In our analysis, we have not considered the delay. Indeed, the delay that characterizes some of the scenarios we investigated (i.e. *eth2eth*) is too small to be correctly studied without the support of special hardware to synchronize the components of the test-bed, and to capture the traffic with a sufficient precision. Anyway, throughput and jitter are sufficient properties with which to study the performance of SCTP (compared with TCP and UDP) and they are among the main QoS parameters that should be considered when delivering many services [14].

5. EXPERIMENTAL RESULTS

In this section we explore the performance of SCTP with respect to jitter and throughput, considering several network scenarios, with different traffic loads. Moreover, we provide a comparative analysis between SCTP and both TCP and UDP. Owing to space constraints, the tables containing the complete results of our experimental analysis are available in [15]. In this work we do not use SCTP's multi-streaming or multi-homing features. All of the traffic between the two SCTP endpoints is delivered only on the stream 0 in an ordered way. SCTP adopts the same congestion/flow control scheme of TCP, with the exception of the fast recovery mechanism, and therefore we expect a comparable behavior between TCP and SCTP, even if an inefficiency in the current specification of SCTP's congestion control causes a performance degradation when there are multiple packet losses in a single window [16]. Before considering any specific experimental details, it is worth mentioning that we repeated each test several times. The mean values across 20 test repetitions are reported in the following figures.

5.1. Wired scenario

Figure 1 shows the throughput and jitter values measured in the *eth2eth* scenario. In this scenario the SCTP performance is comparable to that of UDP and TCP. Both UDP and SCTP are able to reach a bit rate of 87 Mbps. TCP seems to have a better control of congestion and it can serve a maximum bit rate of about 94 Mbps. This can be explained by comparing the basic congestion control algorithm implemented in SCTP against that implemented in TCP, which has all of the latest optimizations. As for the jitter, the three protocols are characterized by similar behavior. There is an initial decrease in the values we measured until the communication becomes congested. During the link congestion we observed a sudden increase of the jitter. UDP is the protocol that presents the lowest jitter in all of the non-congested traffic conditions. Conversely, TCP presents the highest jitter when the link is not congested. Finally, SCTP presents intermediate jitter values. Considering that both TCP and SCTP are protocols carrying out flow control, it is worth noting that, when the link is not subject to a heavy load, the jitter values related to SCTP are approximately half of those of TCP.

5.2. Wireless and heterogeneous scenarios

With regards to measurements conducted under the low packet rate condition, we found similar results for all of the following network scenarios. SCTP has been able to follow the throughput increment we imposed by increasing *PS* from 32 to 1460 bytes. More precisely, it reached a maximum throughput of 1.17×10^3 kbps ($PS = 1460$ bytes). As for the jitter, we recorded the following measurements: (i) mean values smaller than 1 ms; (ii) standard deviations and median values comparable with jitter mean values. It follows that the jitter samples are heavily scattered around their mean values. However, the maximum jitter values that we measured were always of the order of a few fractions of a millisecond. Therefore, we may argue that such dispersion should have a small impact on those SCTP-based applications that have stringent jitter constraints. Both UDP and TCP showed a throughput behavior similar to that of SCTP. Indeed they have been able to correctly serve all of the transmission rates we imposed. As for the jitter, the behavior of UDP is very similar to that of SCTP. Also the TCP jitter behavior can be assimilated to that of SCTP. Looking at measurements made using the medium packet rate, in all of the network scenarios that we considered the protocols were not able to

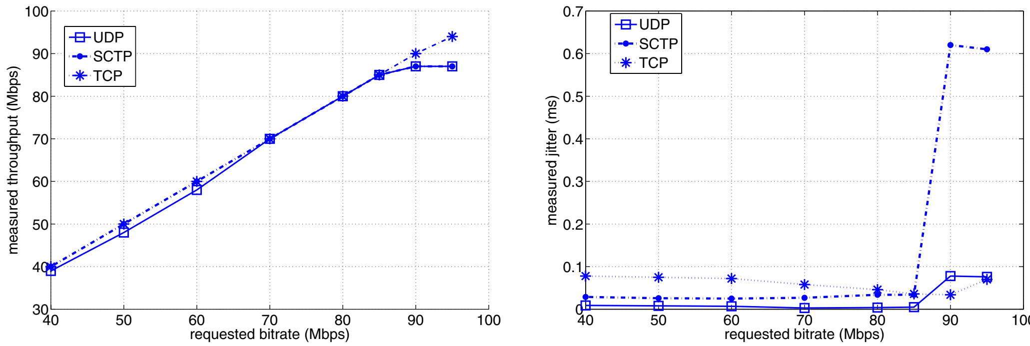

Figure 1. Throughput and jitter measured in the *eth2eth* scenario.

reach the maximum throughput that we asked for. Starting from a threshold value (this value varies for the protocol and network scenario considered), the measured throughput was less than the requested throughput. TCP reaches congestion for higher bit rates, but as the requested bit rate increases even more, it is surpassed in performance by UDP. Indeed, for all of the scenarios, at the highest bit rates that we considered for the medium packet rate condition (that is, during congestion) the following results were obtained: (i) UDP has the best throughput and jitter performance; (ii) SCTP reaches the lowest throughput values, but performs better than TCP with respect to the jitter. Such behavior can be explained by the lack of congestion control in the UDP protocol, which makes it ignore congestion situations at medium packet rates and also makes jitter values smaller. The high packet rate results are quite homogeneous when comparing different scenarios, and the behavior does not change when the packet size, and consequently the bit rate, increases. TCP always obtains the best performance, both in terms of throughput and jitter, followed by SCTP. UDP has the worst values. This can be explained with the fact that at high packet rates the congestion control mechanisms can dramatically improve performance.

5.2.1. Wireless scenarios (w2w-ah and w2w-ap)

With regards to jitter under the low packet rate condition, *w2w-ah* exhibits significantly different behavior from all of the other scenarios. Both the mean and the maximum jitter values we measured are higher than those observed elsewhere. The maximum jitter value is approximately equal to 6.8 ms, one order of magnitude higher than those of the other scenarios. With regard to the medium packet rate tests made in *w2w-ah*, Figure 2 shows that, starting from $PS = 512$ bytes, the SCTP throughput is remarkably less than the requested throughput (2.02×10^3 kbps instead of 4.10×10^3 kbps). Also, in Figure 2 it can be seen that SCTP reaches mean jitter values for the largest packet size ($PS = 1460$ bytes) that are considerably higher than those of the other scenarios (1.24×10^1 ms). With regards to *w2w-ap*, at the medium packet rate this scenario has the worst throughput performance (see Figure 2). In all of the other scenarios, the three protocols are able to at least satisfy the requested bit rate at least until 2.05×10^3 kbps. In *w2w-ap*, UDP cannot even follow the bit rate imposed by

DOI: 10.1002/cpe

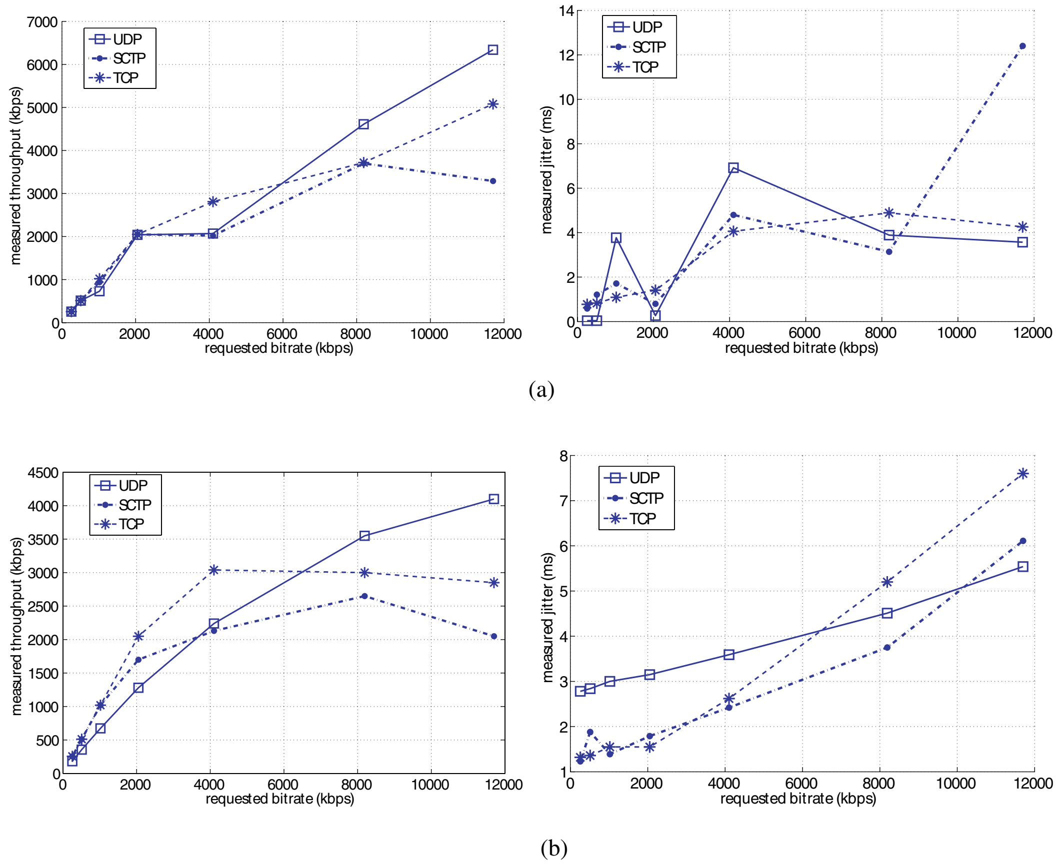

Figure 2. Throughput and jitter of the (a) *w2w-ah* and (b) *w2w-ap* scenarios at medium packet rate.

the smallest packet size (2.56×10^2 kbps with $PS = 32$ bytes, at the medium rate condition), whereas SCTP fails to deliver the requested bit rate for $PS = 256$ bytes: only 1.70×10^3 kbps are transfered instead of 2.05×10^3 kbps. Moreover, we observed that the wireless-only network scenarios are characterized by notably higher jitter values for all of the protocols. For example, SCTP mean jitter values start from 1.23 ms and reach 6.11 ms in *w2w-ap*, whereas in *w2e* and *e2w* jitter values are of the order of tenths of a millisecond under normal conditions, and reach a maximum value of 3.65 ms under congestion. Furthermore, throughput and jitter values in *w2w-ap* and *w2w-ah* scenarios are worse than those of the heterogeneous scenarios, when the high packet rate condition is considered. For example, in the first two cases SCTP maximum throughput values are both around 1500 kbps, whereas in *w2e* and *e2w* a bit rate of about 2500 kbps is reached.

DOI: 10.1002/cpe

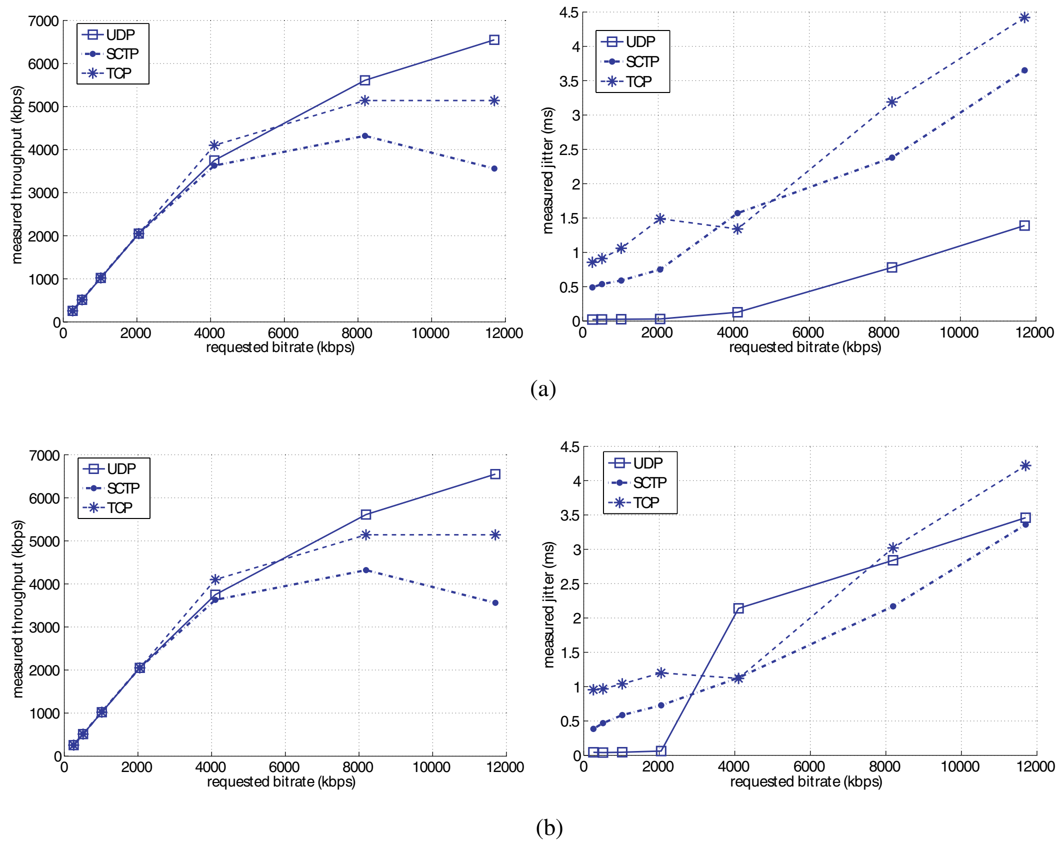

Figure 3. Throughput and jitter of the (a) *e2w* and (b) *w2e* scenarios at medium packet rate.

5.2.2. *Heterogeneous scenarios (e2w and w2e)*

Figure 3 shows that in both of these scenarios SCTP throughput values at the medium packet rate are smaller than those of both TCP and UDP. In *e2w*, SCTP has been able to follow the requested throughput until 2.05×10^3 kbps. Starting from this value there is an increasing difference between the throughput we tried to impose and the measured throughput. On the other hand, TCP and UDP can satisfy the requested throughput up to 4.10×10^3 kbps. However, SCTP has reached a maximum throughput equal to 4.14×10^3 kbps for $PS = 1024$ bytes (trying to impose a throughput of 8.19×10^3 kbps). As for the jitter, SCTP presents the worst values in the *e2w* scenario, and values comparable to those of TCP in the *w2e* scenario (see Figure 3). Considering only the cases in which SCTP has been able to serve the required bit rate, we have measured mean values always smaller than

DOI: 10.1002/cpe

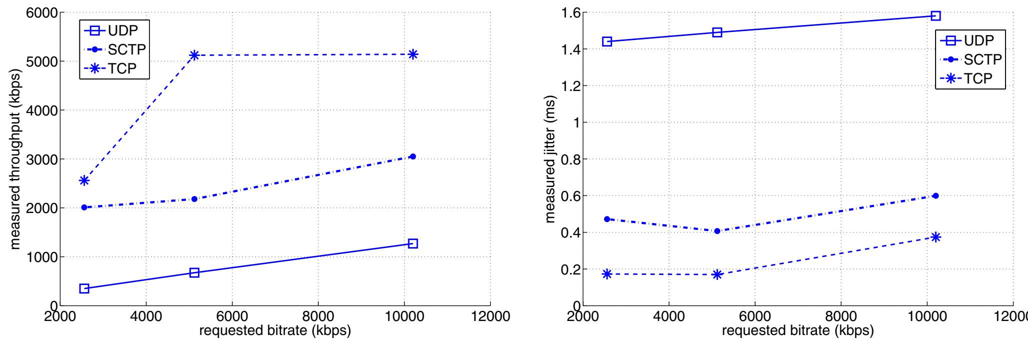

Figure 4. Throughput and jitter of the *w2e* scenario at the high packet rate.

2 ms and standard deviations at least one order of magnitude smaller than the mean values. Therefore, in this working condition, considering only the values of *PS* for which the SCTP association was not saturated, the jitter is quite small and is characterized by regular behavior. Moreover, jitter mean values do not dramatically grow when the SCTP channel becomes overloaded. As anticipated, the behavior at the high packet rate condition is similar for all of the scenarios considered. As an example, we only report diagrams for *w2e* (Figure 4), where it is visible that there are clear performance gaps among the protocols. For both throughput and jitter the best performing protocol is TCP, UDP achieves the lowest throughputs and has the largest jitter values, and the performance of SCTP lies between TCP and UDP. It is also interesting to note that such behavior is the same for all three packet sizes (and thus bit rates) considered.

6. DISCUSSION AND CONCLUSIONS

In this paper we have introduced a traffic generator that supports SCTP traffic generation and presented a packet-level experimental analysis of SCTP in wired, wireless, and heterogeneous scenarios in terms of throughput and jitter. We have also compared SCTP with TCP and UDP and the data used in this work are freely available [17]. In the wired scenario, UDP and SCTP present almost the same level of performance while TCP slightly outperforms them. It is worth noting that, under such quasi-ideal conditions, SCTP jitter values are slightly better. However, in wireless-only and heterogeneous scenarios the performance of SCTP is comparable to the other protocols only under low packet rate conditions. When the packet rate increases, TCP reaches congestion later than the other two protocols. Furthermore, at the highest packet rates, TCP rather outperforms the other protocols, while SCTP is located in the middle between UDP and TCP. As for a comparison between the heterogeneous and the wireless scenarios we found some minor differences in the behavior of SCTP. In both *e2w* and *w2e*, with a $PR = 1000$ pps, SCTP has been able to support a maximum bit rate equal to 2.05×10^3 kbps. This value is higher than that reached by SCTP in the *w2w-ap* scenario (1.02×10^3 kbps), and equal

DOI: 10.1002/cpe

to that of the *w2w-ah* scenario. Moreover, with $PR = 10\,000$ pps the maximum throughput reached by SCTP in the heterogeneous scenarios was greater than in the two wireless scenarios. With specific regard to the behavior of SCTP under heterogeneous scenarios the following observations can be made: (i) the throughput is smaller than those of both TCP and UDP; (ii) jitter values are worse or sometimes comparable to those of the other protocols. From the results of the presented experimental analysis, there are no advantages to be gained if SCTP is used as a simple substitute for TCP, without the unordered delivery and the multi-streaming features. Our results confirmed that the poor SCTP performance is mostly the result of the LK-SCTP implementation. SCTP is still a new protocol compared with TCP (which was created in the early 1980s). A great deal of work has been carried out with regards to performance in TCP. This is not the case with LK-SCTP. The priority in the LK-SCTP project has been to support all features. Only recently have developers turned their attention to performance issues. Thus, to correctly and fairly evaluate the performance benefits of using the new features introduced into SCTP (multi-homing, multi-streaming, etc.) compared with older protocols, the performance gap of current implementations, highlighted in our measurements, needs to be heavily reduced. Despite this, the results presented in this paper may define a reference framework for the development of throughput and jitter-aware SCTP-based network applications. We have learned that there is no advantage in using SCTP as a simple substitute for TCP (or UDP) and that the current poor performance of SCTP is the result of the poor performance of its implementation in the Linux operating system. With regards to our future work we plan to undertake the following: (i) apply the presented methodology to other QoS parameters, such as packet loss and delay; (ii) complete the statistical characterization of SCTP in heterogeneous scenarios (in terms of marginal distributions and temporal structures). Finally, we are also working on supporting SCTP multi-homing and multi-streaming.

REFERENCES

1. Emma D, Loreto S, Pescapé A, Ventre G. Measuring SCTP throughput and jitter over heterogeneous networks. *Proceedings of the IEEE AINA 2006 Workshop on Performance Analysis and Enhancement of Wireless Networks*. IEEE Press: Piscataway, NJ, 2006.
2. D-ITG Web Site. http://www.grid.unina.it/software/ITG/ [19 December 2006].
3. Nagamalai D, Lee J-K. Performance of SCTP over high speed wide area networks. *Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems*, vol. 2, December 2004. IEEE Press: Piscataway, NJ, 2004; 890–895.
4. Rajamani R, Kumar S, Gupta N. SCTP versus TCP: Comparing the performance of transport protocols for web traffic. *Technical Report*, Computer Sciences Department, University of Wisconsin–Madison, July 2002.
5. Kang S, Fields M. Experimental study of the SCTP compared to TCP. *Technical Report*, Department of Electrical Engineering, Texas A&M University, 2003.
6. Ma L, Yu F, Leung VCM. Modeling SCTP throughput in integrated WLAN/cellular networks. *Proceedings of the 2005 IEEE International Conference on Communications*, vol. 5, 16–20 May 2005. IEEE Press: Piscataway, NJ, 2005; 3445–3449.
7. Fu S, Atiquzzaman M, Ivancic W. Effect of delay spike on SCTP, TCP Reno, and Eifel in a wireless mobile environment. *Proceedings of the 2005 International Conference on Computer Communications and Networks*. IEEE Press: Piscataway, 2002; 575–578.
8. Kumar A, Jacob L, Ananda AL. SCTP vs TCP: Performance comparison in MANETS. *Proceedings of the 2004 IEEE International Conference on Local Computer Networks*, Washington, DC, 2004. IEEE Press: Piscataway, NJ, 2004.
9. Argyriou A, Madisetti V. Performance evaluation and optimization of SCTP in wireless *ad-hoc* networks. *Proceedings of the 2003 IEEE International Conference on Local Computer Networks*, October 2003. IEEE Press: Piscataway, NJ, 2003.
10. Fu S, Atiquzzaman M. SCTP: State of the art in research, products, and technical challenges. *IEEE Communications Magazine* 2004; **42**(4):64–76.
11. Stewart R, Xie Q, Morneault K, Sharp C, Schwarzbauer H, Taylor T, Rytina I, Kalla M, Zhang L, Paxson V. Stream Control Transmission Protocol. *Request for Comments 2960*, IETF, October 2000.

12. Linux kernel stream control transmission protocol (LK-SCTP). http://lksctp.sourceforge.net/ [19 December 2006].
13. Stewart R, Arias-Rodriguez I, Poon K, Caro A, Tuexen M. Stream Control Transmission Protocol (SCTP) Specification Errata and Issues. *Request for Comments 4460*, IETF, April 2006.
14. Alves M, Corsello L, Karrenberg D, Ogut C. New measurements with the RIPE NCC test traffic measurements setup. *Proceedings of the 2002 Passive and Active Measurement Conference*, March 2002.
15. Extended list of the experimental results. http://www.grid.unina.it/software/ITG/D-ITGpublications/SCTPresults.pdf [19 December 2006].
16. Caro AL, Shah K, Iyengar JR, Amer PD. SCTP and TCP variants: Congestion control under multiple losses. *Technical Report TR2003-04*, CISC Department, University of Delaware, February 2003.
17. Traffic Project Web site at the University of Napoli Federico II. http://www.grid.unina.it/Traffic/ [19 December 2006].

DOI: 10.1002/cpe