

Path Selection of SCTP Fast Retransmission in Multi-homed Wireless Environments[†]

Yuansong Qiao^{1,2}, Enda Fallon¹, John Murphy³, Liam Murphy³, Austin Hanley¹

¹ Applied Software Research Centre, Athlone Institute of Technology, Ireland

² Institute of Software, Chinese Academy of Sciences, China

³ Performance Engineering Laboratory, University College Dublin, Ireland
{ysqiao, efallon}@ait.ie, {j.murphy, Liam.Murphy}@ucd.ie, ahanley@ait.ie

Abstract. In heterogeneous wireless network environments, network connections of a device may have a significant differential. This paper studies the effect of path bandwidth differential on the performance of fast retransmission strategies in Multi-homing environments. It identifies that fast retransmission on an alternate path may cause receive buffer blocking when path bandwidth differential is significant and the receive buffer is limited. A theoretical model is proposed for selecting retransmission path during the fast retransmission phase, which is based on receive buffer and path conditions. The model is verified through simulations with various path differentials.

1 Introduction

Multi-homing technologies, where a host can be addressed by multiple IP addresses, are increasingly being considered by developers implementing mobile applications. An enabling factor reinforcing this adoption is the trend towards mobile devices supporting a hybrid of networking capabilities such as 802.11 and UMTS. The characteristics of mobile environments, with the possibility of frequent disconnections and fluctuating bandwidth, pose significant issues for mobile application developers and therefore the path redundancy offered by multi-homing protocols has a clear attraction.

There is a significant standardization effort focusing on mobility and Multi-homing at various layers in the OSI stack. This is illustrated by the interest in tech-

[†] The authors wish to recognize the assistance of Enterprise Ireland through its Innovation Partnership fund in the financing of this Research programme.

nologies such as Media Independent Handover Framework (IEEE 802.21) [1], Mobile IP (MIP) [2], IP layer Multi-homing (SHIM6) [3], Datagram Congestion Control Protocol (DCCP) [4] and SCTP Dynamic Address Reconfiguration [5]. While it is possible to implement mobility at the network layer and the application layer, it is the Transport Layer through a technology such as SCTP [6] which enables an end-to-end and user centric implementation of mobility and at the same time provides transparent switch over services to the application layer.

Stream Control Transmission Protocol (SCTP) is a reliable TCP-friendly transport layer protocol defined by IETF, which was originally designed for signalling data. Due to its attractive features such as multi-homing and multi-streaming, SCTP has received much attention from the network community. SCTP support link backup through transmitting and receiving data from multiple IP addresses. It sends new data to the primary address and lost data could be transmitted to the current address or to an alternate address. SCTP continually monitors path availability. If the primary address fails, it will switch over to an alternate address.

This paper discusses the performance of SCTP retransmission policies in mobile environments where the paths used in an SCTP association have a significant transmission differential. According to path selection strategies during the fast retransmission phase, two retransmission policies have been proposed in [7]. They are:

FRtxOnAlt - fast retransmission on an alternate path.

FRtxOnSame - fast retransmission on the same path.

FRtxOnSame is a safe solution undoubtedly because it returns to single path scenarios, while FRtxOnAlt may benefit by exploring multi-homing features. The basic idea behind FRtxOnAlt is that it assumes sending data on different paths is similar to sending data on a single path with network anomalies, such as reordering or delay spikes. This paper studies the effects of path bandwidth differential and a limited receive buffer on the performance of the two retransmission strategies. The features of WLAN, 3G and GPRS are considered because they are broadly used in the current Internet.

This paper identifies and illustrates FRtxOnAlt may cause receive buffer blocking which degrades performance significantly. It presents a theoretical model for selecting retransmission path based on receive buffer size and path bandwidth differential.

This paper is organized as follows. Section 2 summarizes related work. Section 3 illustrates the simulation setup. Section 4 describes the SCTP performance degradation problem in detail. Section 5 proposes a theoretical model for analyzing receiving buffer blocking problem and presents suggestions about implementation of the model. The conclusions are presented in Section 6.

2 Related Work

Stream Control Transmission Protocol (SCTP) [6] originated as a protocol called Multi-Network Datagram Transmission Protocol (MDTP). The motivation for MDTP arose from the fact that TCP had inherent weaknesses in relation to the control of telecommunication sessions. MDTP was designed to transfer call control signalling on “carefully engineered” networks [8]. When one analyses the origins of SCTP it is interesting to note that its initial target environment was vastly different from that experienced in current day mobile networks. Given its origin as a fixed line oriented protocol, and in particular a protocol designed towards links with roughly equivalent transmission capabilities, the transition towards a mobile enabled protocol has raised a number of design issues. Many related works have raised issues in relation to the design of SCTP.

In [9] two SCTP stall scenarios are presented, the authors identify that the stalls occur as a result of SCTP coupling the logic for data acknowledgment and path monitoring. In [7] different SCTP retransmission policies are investigated for a lossy environment, a retransmission strategy which sends the fast retransmission packets on the same path and the timeout retransmission packets on an alternate path are suggested. In [10] SCTP is extended for Concurrent Multi-path Transfer (CMT-SCTP) while in [11] the authors identify that a finite receiver buffer will block CMT-SCTP transmission when the quality of one path is lower than others. Several retransmission policies are studied which can alleviate receiver buffer blocking. In [12] the authors focus on making SCTP more robust to packet reordering and delay spikes.

3 Simulation Setup

The simulations focus on the situation where a mobile node has a fixed GPRS or 3G connection along with various high bandwidth connections and analyze the SCTP performance degradation. The high bandwidth connection is set to the primary path in SCTP. All simulations in this paper are carried out by running Delaware University's SCTP module [13, 14] for NS-2.

The simulation topology is shown in Fig. 1. Node S and Node R are the SCTP sender and receiver respectively. Both SCTP endpoints have two addresses. R1,1, R1,2, R2,1 and R2,2 are routers. It is configured with no overlap between the two paths. As only the effect of bandwidth is considered in this paper the loss rate is set to zero. Node S begins to send 20MB ftp data to Node R at the 5th second. The MTU of each path is 1500B. The queue lengths of bottleneck links in both paths are 50 packets. The queue lengths of other links are set to 10000 packets. The bandwidth of the secondary path bottleneck link is 36Kbps or 384Kbps, the delay is 300ms. The bandwidth of the primary path bottleneck link changes from

36Kbps to 30Mbps, the delay is a constant 50ms. SCTP parameters are all default except those mentioned. Initially the receiver window is set to 1MB. The initial slow start threshold is set large enough to ensure that the full primary path bandwidth is used. Only one SCTP stream is used and the data is delivered to the upper layer in order.

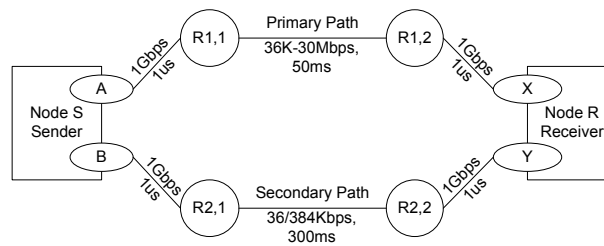


Fig. 1. Simulation network topology.

4 A Case Study for Receive Buffer Blocking

This section describes the FRtxOnAlt performance issues in detail for the simulation configuration where the primary path bandwidth is 10Mbps and the secondary path bandwidth is 36Kbps. Node S starts to send 20MB data to Node R at the 5th second. The initial slow start threshold is 350KB. The data transmission time for retransmission on the secondary path and retransmission on the primary path are 38.1697s and 18.1355s respectively. The secondary path affects the performance dramatically in this situation.

The simulation results for FRtxOnAlt are shown in Fig. 2. The sending process begins in slow start mode. From A to B (Fig. 2a), 68 packets are dropped on the primary path because of congestion. At C (Fig. 2b,2c), the sender finds the packets lost by duplicate SACKs, and it reduces the congestion window by half to 147534B. From C to E (Fig. 2b), the sender sends lost packets on the secondary path using the fast retransmission algorithm. Within these retransmitted packets, 18 are lost as a result of the queue of the secondary path being full due to its low bandwidth. These 18 lost packets will be retransmitted again on the primary path after their retransmission timeout.

The queuing delay of the secondary path is very large (Fig. 2b). This delay is in proportion to the retransmission packet number since the data enqueueing interval is shorter than the data sending time. In this simulation, the 68 packets are enqueued on the secondary path in $6.761304s - 6.579304s = 182ms$ (E-C), whereas the time for sending one packet on the secondary path is $1500B / 36Kb = 333.3ms$. At D (Fig. 2a), the primary outstanding data size is less than the congestion window, so the sender begins to send new data on the primary path at a speed of around

147534B per RTT (around 100ms). At F (Fig. 2a,2d), the receiver window maintained at the sender side drops to 292B and at this moment only the first retransmitted packet is received on the secondary path. Since SCTP provides ordered delivery of data, the receiver can not deliver the received data to upper layer. It takes 0.664396s (F-C) from the beginning of the fast retransmission phase to fill the receiver's buffer. After point F, the sender can only send new data chunks on the primary path after it receives a SACK from the secondary path even though the cwnd of the primary path is not fully used. Then the sender will recover from this blocking situation after all packets transmitted on the secondary path are received.

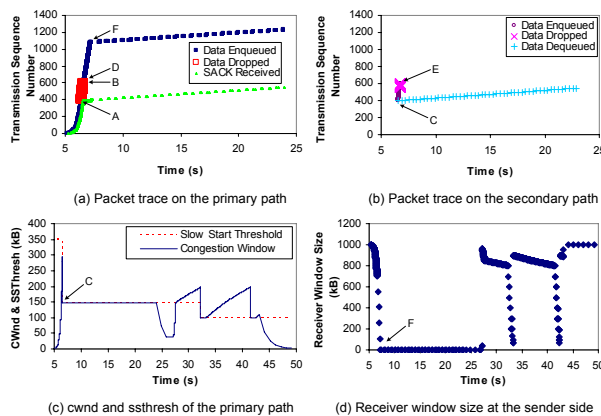


Fig. 2. Retransmission on the secondary path. A: 6.414934s, B: 6.575737s, C: 6.579304s, D: 6.658611s, E: 6.761304s F: 7.24370s.

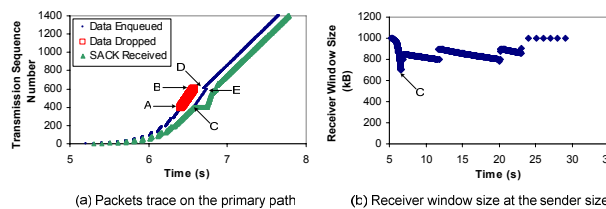


Fig. 3. Retransmission on the primary path. A: 6.414934s, B: 6.575737s, C: 6.579304s, D: 6.699467s, E: 6.740312s.

As a comparison with FRTxOnAlt, the simulation results for retransmission on the primary path are shown in Fig. 3. Fig. 3a shows the packets trace on the primary path. The sender starts transmission at the 5th second in slow start mode. From A to B, 68 packets are dropped as a result of the sending speed reaching the maximum bandwidth. At C, the sender finds the packets lost by duplicate SACKs, and the congestion window is reduced by half. From C to E, each dropped packet is fast retransmitted on the primary path immediately when the sender receives 4 consecutive loss reports for it. After D, the sender begins to transmit new data be-

cause the outstanding data size is smaller than the congestion window size. At C (Fig. 3b), the receiver window size maintained in the sender side drops to the lowest point 701996KB.

4.1 Result Analysis

The above tests illustrate that FRtxOnAlt becomes inefficient when the secondary path has a relatively low bandwidth and the receiver's buffer is finite. The sender injects large numbers of packets into a path with uncertain condition during fast retransmission. Then the receiver's buffer is filled up when the receiver is waiting for the retransmitted packets arrival from the secondary path. It is important to note that slow start phase happens not only in the beginning stage but also after a transmission timeout. Furthermore, fast retransmission can be triggered for many reasons, such as random loss, not only in slow start phase.

This performance degradation comes from the SCTP design rationale. SCTP is not a load sharing protocol, so it does not send data on multiple paths simultaneously. It assumes sending data on different paths is similar to sending data on a single path with network anomalies, such as reordering, delay spikes. Consequently, it adopts current TCP's congestion control and fast retransmission algorithms without significant modifications.

In single path configurations, network anomalies exist but happen randomly. In multi-homed environments, besides network anomalies, the paths differences are usually constants. Every time an alternate path is used, it will affect performance, and therefore performance degradation occurs frequently. Accordingly, path differences should be considered in the algorithm.

If the receiver's buffer is infinite, retransmission on a slow secondary path will not decrease throughput obviously. After the sender finishes retransmitting data on the secondary path, it can continue sending new data with a constant speed which is an estimation of the bottleneck link speed.

5 Modelling Receive Buffer Blocking

This section analyzes the relationship between receive buffer blocking and bandwidth difference to provide method for path selection during the fast retransmission phase.

The symbols used for this section are listed as follows:

- B1 – Bandwidth of the primary path;
- R1 – RTT of the primary path;
- Q1 – Queue length of the primary path;
- B2 – Bandwidth of the secondary path;

R_2 – RTT of the secondary path;

Q_2 – Queue length of the secondary path;

A – Receive buffer size;

W – Data length which has been sent on the primary path at the beginning moment of the fast retransmission phase; suppose these data has been received or lost when the fast retransmit occurs.

V_1 – Data arriving rate from the primary path;

V_2 – Receive buffer releasing rate when the retransmitted data arrives at the receive;

D_1 – The delay between the moment for the sender entering the fast retransmit phase and the moment the sender starting to transmit new data on the primary path.

$cwnd1_Final$ – Congestion window size before the sender enters the fast retransmit phase.

$cwnd1_Previous$ – Congestion window size of the last second round before the sender enters the fast retransmit phase.

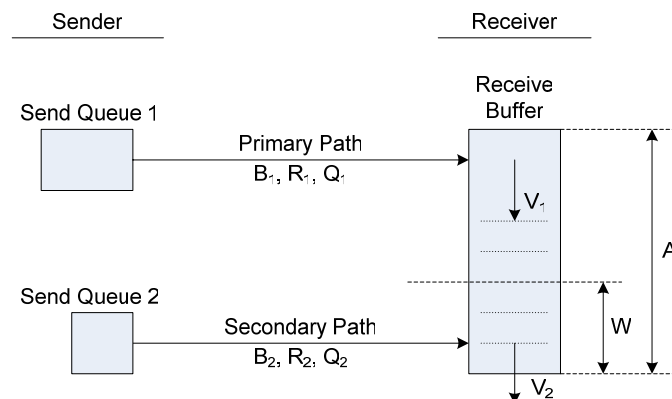


Fig. 4. Model for receive buffer blocking

Suppose the receive buffer size is large enough to fully use the bandwidth in the bottleneck link. When the sender enters the fast retransmit phase, W bytes data has been transmitted on the primary path. The sender will fast retransmit lost data on the secondary path. As the congestion window of the primary path has been decreased, the sender will wait for D_1 seconds before transmitting data on the primary path. In order to avoid receive buffer blocking, W bytes data should be released before the data arrived from the primary path fills up the receive buffer. The first data packet transmitted on each path takes half of the path RTT to arrive at the receiver. The following packets arrive at the speed of V_1 and V_2 for the primary and secondary path respectively. Therefore, the relationship of these variables could be described as (1):

$$\frac{W}{V_2} + \frac{R_2}{2} \leq \frac{A}{V_1} + \frac{R_1}{2} + D_1 \quad (1)$$

Next, we need to estimate the variables in (1). In the slow start phase, the final transmission round of the sender begins from the first lost packet is detected. The congestion window size is $cwnd1_Final$. The congestion window size before the final RTT round is $cwnd1_Previous$. As the receiver uses delayed acknowledgements, the congestion window grows by 50% every RTT round. Therefore:

$$cwnd1_Final = \frac{3}{2} cwnd1_Previous \quad (2)$$

In the final transmission round before the fast retransmit phase begins, $cwnd1_Previous$ bytes data arrives at the receiver and $cwnd1_Previous/2$ bytes data is dropped by network. Consequently, at the moment when the final transmission round begins, the buffer in the bottleneck link must be full. Therefore the maximum value for $cwnd1_Previous$ is:

$$cwnd1_Previous \leq B_1 R_1 + Q_1 \quad (3)$$

As the fast retransmit phase is triggered by 4 gap reports, the sender sends 3 new data packets for the first 3 gap reports. At the moment when the sender enters the fast retransmit phase, it has transmitted $cwnd1_Final$ bytes data plus 3 packets (4):

$$W \leq \frac{3}{2} (B_1 R_1 + Q_1) + 3 \times MTU \quad (4)$$

The receive buffer releasing rate can be estimated as follows. In the final transmission round before the fast retransmit phase, the sender transmits $1.5 \times cwnd1_Previous$ bytes data and $0.5 \times cwnd1_Previous$ bytes data are lost evenly. Consequently, 3 packets are released for every packet received on the secondary path. Since the data arriving rate of the secondary path equals to the secondary path bandwidth, the receive buffer releasing rate is 3 times of the secondary path bandwidth (5).

$$V_2 = 3B_2 \quad (5)$$

When the sender enters the fast retransmit phase, the congestion window is reduced by half. The transmission speed of the primary path can be estimated as in (6).

$$V_1 \leq \min(B_1, \frac{cwnd1_Final/2}{R_1}) \quad (6)$$

Since the primary path congestion window of the sender is reduced by half when the sender enters the fast retransmit phase, the sender need to wait for half of the congestion window data being acknowledged before sending new data. The waiting time is greater than half of the primary path RTT because the queue at the bottleneck link is full and the end to end RTT is greater than the path RTT (7).

$$D_1 > \frac{R_1}{2} \quad (7)$$

Substitute (4), (5), (6) and (7) into (1), receive buffer blocking can be avoid if (8) is satisfied.

$$B_2 > \frac{B_1 R_1 + Q_1 + 2 \times MTU}{\frac{2A}{B_1} + (2R_1 - R_2)} \quad (8)$$

5.1 Verification of the Model

Two groups of simulations are executed for the fixed secondary path bandwidths of 36Kbps and 384Kbps. The primary path bandwidth changes from 36Kbps to 30Mbps in both tests. The simulation topology is shown in Fig. 1. The initial slow start threshold is 1MB. 20MB data is transmitted via FTP. For each path configuration, the data transmission time for retransmission on the secondary path and retransmission on the primary path are computed. The data transmission time and the percentage change of two retransmission strategies for the 36Kbps secondary path bandwidth are shown in Table 1. The same results for the 384Kbps secondary path bandwidth are shown in Table 2. The percentage change P is calculated as in (9).

$$P = \frac{RtxOnSec - RtxOnPri}{RtxOnSec} \times 100\% \quad (9)$$

Table 1. Data Transmission Time for Primary Paths of Various Bandwidths and Secondary Path of 36kbps.

Primary Bandwidth	FRtxOnAlt (s)	FRtxOnSame (s)	Percentage Change (%)
30 Mbps	40.7766	9.02946	77.85
25 Mbps	40.821	9.06883	77.78
20 Mbps	30.0813	9.76529	67.53
15 Mbps	40.1722	12.8919	67.90
10 Mbps	38.1697	18.1355	52.48
5 Mbps	49.0648	34.1332	30.43
2 Mbps	91.5079	83.8049	8.41
1 Mbps	167.976	166.792	0.70
900 kbps	185.301	185.241	0.03
800 kbps	207.632	208.307	-0.32
512 kbps	324.038	325.116	-0.33
384 kbps	431.673	433.298	-0.37
36 kbps	4596.8	4617.78	-0.45

Table 2. Data Transmission Time for Primary Paths of Various Bandwidths and Secondary Path of 384kbps.

Primary Bandwidth	FRtxOnAlt (s)	FRtxOnSame (s)	Percentage Change (%)
30 Mbps	24.1781	9.02946	62.65
25 Mbps	25.577	9.06883	64.54
20 Mbps	21.841	9.76529	55.28
15 Mbps	18.7894	12.8919	31.38
10 Mbps	26.4738	18.1355	31.49
6Mbps	29.0827	28.6448	1.50
5.1Mbps	33.5139	33.4865	0.08
5 Mbps	34.0862	34.1332	-0.13
2 Mbps	83.5958	83.8049	-0.25
1 Mbps	166.252	166.792	-0.32
512 kbps	324.014	325.116	-0.34
384 kbps	431.673	433.298	-0.37
36 kbps	4591.13	4617.78	-0.58

The results show that when the primary path bandwidth is below a certain threshold, approximately 900Kbps for the 36Kbps secondary path and 5.1Mbps for the 384Kbps secondary path, retransmission on the secondary path can improve performance. When the primary path bandwidth is above this threshold, re-

transmission on the secondary path will decrease performance significantly. Higher primary bandwidth will suffer more performance degradation because the higher bandwidth will retransmit more data on the secondary path during congestion, and therefore the receiver buffer blocking time will be longer. Retransmission on the primary path will increase performance for this situation.

Equation (8) can be verified by the simulation results in Table I & Table II. According to (8), when the secondary path bandwidth is 36kbps, the primary path bandwidth should be less than 801.6kbps to avoid receive buffer blocking. Table I shows that this threshold is between 800kbps and 900kbps. When the secondary path bandwidth is 384kbps, the primary path bandwidth should be less than 4.86Mbps according to (8) and it is between 5Mbps and 5.1Mbps according to Table II.

The results of (8) are less than the simulation results in Table I and II because equation (5) underestimates the receive buffer releasing rate (V_2) when the retransmitted data arrives at the receiver. In the final transmission round before the fast retransmit phase starts, if the lost data size of the primary path is less than $cwnd1_Final/3$, the receive buffer releasing speed will be greater than 3 times of the secondary path bandwidth.

5.2 *Implementation of the Model*

This section presents some suggestions for implementing (8). The primary path bandwidth could be estimated through the bandwidth estimation algorithm in Westwood TCP [15]. The primary path RTT could come from the SCTP internal RTT measurement. The queue length in the primary path bottleneck link could be estimated from the maximum RTT and the minimum RTT. Suppose the minimum RTT is the transmission time for a null system and the maximum RTT is the transmission time for a full-loaded system, the estimated queue length is the product of the primary path bandwidth and the difference of the maximum and minimum RTT. The secondary path RTT can be acquired from SCTP internal RTT measurement. The secondary path bandwidth can be estimated from local link information or from active detection, such Packet-Pair Bandwidth Estimation (PPBE) [16].

6 **Conclusions and Future Work**

This paper studies the effects of path bandwidth difference on the fast retransmission strategies performance in a heterogeneous context. It illustrates that fast retransmission on an alternate path will decrease performance when the bandwidth of the secondary path is relatively low. The transmission will be blocked in this

scenario due to packet re-ordering filling the receiver's buffer. A model for estimating receive buffer blocking is proposed, which is based on path bandwidth, delay and receive buffer size. An implementation suggestion is given for using the model in path selection during the fast retransmit phase.

We plan to implement the receive buffer blocking model in SCTP for helping path selection and study path selection algorithms in experimental wireless and mobile environments.

References

1. Dutta, A., Das, S., Famolari, D., Ohba, Y., Taniuchi, K., Kodama, T., Schulzrinne, H.: 'Seamless Handoff across Heterogeneous Networks - An 802.21 Centric Approach', Proc. IEEE WPMC 2005, Aalborg Denmark, September 2005.
2. C. Perkins Ed., "IP Mobility Support for IPv4", IETF RFC 3344, August 2002.
3. E. Nordmark, M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", IETF draft, Feb. 2008, <http://www.ietf.org/internet-drafts/draft-ietf-shim6-proto-10.txt>.
4. E. Kohler, M. Handley, S. Floyd, "Datagram Congestion Control Protocol (DCCP)", IETF RFC 4340, Mar. 2006.
5. R. Stewart, Q. Xie, M. Tuexen, S. Maruyama, M. Kozuka, "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration", IETF RFC 5061, Sep. 2007.
6. R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang and V. Paxson, "Stream Control Transmission Protocol", IETF RFC 2960, Oct. 2000.
7. Caro Jr., P. Amer and R. Stewart, "Retransmission Schemes for End-to-end Failover with Transport Layer Multihoming", GLOBECOM 2004, November 2004.
8. R. Stewart and Q. Xie, "Stream Control Transmission Protocol (SCTP), A Reference Guide", Addison-Wesley, ISBN 0-201-72186-4, Jan. 2006.
9. J. Noonan, P. Perry, S. Murphy and J. Murphy, "Stall and Path Monitoring Issues in SCTP", Proc. of IEEE Infocom, Conference on Computer Communications, Barcelona, April 2006.
10. J. Iyengar, K. Shah and P. Amer, "Concurrent multipath transfer using sctp multihoming", SPECTS'04, San Jose, USA, July 2004.
11. J. Iyengar, P. Amer and R. Stewart, "Receive buffer blocking in concurrent multipath transfer", IEEE Globecom 2005, St. Louis, Nov. 2005.
12. S. Ladha, S. Baucke, R. Ludwig and P. Amer, "On Making SCTP Robust to Spurious Retransmissions", ACM Computer Communication Review, 34(2), Apr. 2004.
13. Caro and J. Iyengar, "ns-2 SCTP module, Version 3.5", <http://www.armandocaro.net/software/ns2sctp/>.
14. UC Berkeley, LBL, USC/ISI, and Xerox Parc, "ns-2 documentation and software, Version 2.29", Oct. 2005, www.isi.edu/nsnam/ns.
15. C. Casetti, M. Gerla, S. Mascolo, M. Sanadidi and R. Wang, "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links", ACM Wireless Networks, 2002, 8:467-479.
16. R.L. Carter and M.E. Crovella, "Measuring Bottleneck Link Speed in Packet-Switched Networks", Performance Evaluation, 1996, 27: 297-318.