This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2008 proceedings.

cmpSCTP: An Extension of SCTP to Support Concurrent Multi-path Transfer

Jianxin Liao, Jingyu Wang, Xiaomin Zhu
State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications
P.O. Box 296, Beijing University of Posts and Telecommunications, Beijing, P.R. China, P.C.: 100876
Email: { liaojianxin, wangjingyu, zhuxiaomin}@ebupt.com

***Abstract*—This paper introduced cmpSCTP, a transport layer solution for concurrent multi-path transfer that modifies the standard Stream Control Transmission Protocol (SCTP). The cmpSCTP aims at exploiting SCTP's multi-homing capability by selecting several best paths among multiple available network interfaces to improve data transfer rate to the same multi-homed device. Through the use of path monitoring and packet allotment techniques, cmpSCTP tries to transmit given amount of packets at corresponding path as its ability. At the same time, cmpSCTP updates the transmission strategy based on the real-time information of all of paths. Using cmpSCTP's flexible path management capability, we may switch the flow between multiple paths automatically to realize seamless path handover. Extensive simulations under different scenarios using OPNET verified that cmpSCTP can effectively enhance transmission efficiency and highlighted the superiority of cmpSCTP against the other SCTP's extension implementations under performance indexes such as throughput, handover latency, packet delay, and packet loss.**

***Keywords*—Multi-homing, Concurrent multi-path transfer, Traffic control, SCTP.**

I. INTRODUCTION

A host is multi-homed if it can be addressed by multiple IP addresses, as is the common case when the host has multiple network interfaces. Multi-homing is increasingly economically feasible and can be expected to be the rule rather than the exception in the near future. A Multi-homing host may be simultaneously connected through multiple access technologies, and even multiple end-to-end paths to increase resilience to path failure [1]. For instance, a mobile user could have simultaneous Internet connectivity via a wireless local area network using 802.11 and a wireless wide area network using UMTS.

As of today, the only transport protocol supporting Multi-homing is the Stream Control Transmission Protocol (SCTP) [2]; yet, SCTP can only exploit at most one of the available paths at any given time. That is, SCTP uses multiple interfaces only for redundancy: every host chooses a primary destination address, normally used for the transmission of the data units, "data chunks" in SCTP terminology, whereas the alternate addresses are considered as secondary, whose conditions are periodically monitored with the transmission of probe chunks called Heartbeat. The backup path is used only (i) to retransmit lost data chunks, in order to increase the probability of successful retransmissions, (ii) to transmit new data chunks when, due to the excess of the number of (e.g., five) consecutive timeouts on the primary path, the default interface is declared as "inactive". In the latter case, SCTP transmits new data chunks toward the backup interface and Heartbeat chunks toward the primary one. As soon as the Heartbeats reception on the primary interface is confirmed, its state is toggled to "active" and the transmission is resumed.

In this paper, we proposed and designed cmpSCTP, which is designed to use all the available paths for the association at the same time, instead of using only the primary path like SCTP. Therefore cmpSCTP provides full multi-homing support through the simultaneous utilization of all available paths. Furthermore, all available paths are classified as active path and monitor path, and introduced different strategies to be used for load sharing and redundant transmission according to their connection states. In addition, cmpSCTP monitors the states of the available paths and as their states change, i.e., new paths become active or existing paths break, it updates the transmission strategy, which can be used to switch the flow between multiple paths smoothly. As such, this work naturally leads to another fundamental issue of end-to-end support for seamless handover.

The remainder of this paper is organized as follows. Section II surveys related work. Section III proposed the concurrent multi-path SCTP, with original contribution presented in detail. In Section IV, simulation models are described, and numerical results are presented to investigate the performance. Finally, conclusions and possible future work are described in Section V.

II. RELATED WORK

Researches on extending SCTP to support concurrent multi-path transfer, e.g. simultaneously sending data over multiple available paths to increase the association bandwidth, are currently in progress [3]–[7].

The work in [3], [4], by the original SCTP proposers, suggests to change the SCTP sender operation to compensate for the problems introduced by using a unique sequence-number space for tracking packets sent over multiple paths. The sender maintains a set of per-destination virtual queues and spreads the packets across all available paths as soon as the congestion window allows it. Retransmissions are triggered only when several Selective ACKnowledgments (SACKs) report missing chunks (SCTP protocol data units) from the same virtual queue.

AL et al [5] propose Load-Sharing SCTP (LS-SCTP), a

This work was jointly supported by: (1) National Science Fund for Distinguished Young Scholars (No. 60525110); (2) National 973 Program (No. 2007CB307100, 2007CB307103); (3) Program for New Century Excellent Talents in University (No. NCET-04-0111); (4) Development Fund Project for Electronic and Information Industry (Mobile Service and Application System Based on 3G); (5) National Specific Project for Hi-tech Industrialization and Information Equipments (Mobile Intelligent Network Supporting Value-added Data Services).

mechanism to aggregate the bandwidth of all the paths connecting the endpoints and dynamically adds new paths as they become available. The key idea is to introduce a per-association, per-path data-unit sequence numbering that extends the per-association SCTP congestion control to a finergrained, per-path congestion control.

Hsieh et al. [6] propose pTCP (parallel TCP) based on the Transmission Control Protocol (TCP). pTCP has two components - Striped connection Manager (SM) and TCP-virtual (TCP-v). This decoupling of functionality allows for intelligent scheduling of transmissions and retransmissions. Similarly, mTCP [7] also significantly modifies TCP to use multiple paths provided by a special routing layer that implements from a Resilient Overlay Network (RON).

However, none of the previous proposals fully addresses the case in which the paths comprised in the SCTP association exhibit widely-different bandwidths and round trip times (RTTs). In such scenario, the packets sent by the source reach the destination out of order, triggering a lot of retransmissions on the underlying TCP SACK on which SCTP is based. The most promising solutions proposed to mitigate packet reordering are based on: i) estimating the available bandwidth and the RTT on each path, ii) using an appropriate packet scheduling algorithm to stripe packets across all paths, so that they reach the destination almost in order.

III. A CONCURRENT MULTI-PATH SCTP

As a general remark, we found our modifications to be similar to those recommended by other Multi-homing related papers [3]-[7]. In this section, we will refer to the features of cmpSCTP different with those techniques and present the key design elements of the cmpSCTP protocol.

A. cmpSCTP Design

Similar to LS-SCTP[5]，cmpSCTP is also based on the idea of separating the association flow control from congestion control. In cmpSCTP the flow control is on association basis, thus both the sender and receiver endpoints use their association buffer to hold the data chunks regardless of the path these data chunks were sent or received. On the other hand, congestion control is performed on per path basis, thus the sender has a separate congestion control for each path. Especially, the congestion control mechanism on each path can follow the standard SCTP [2], TCP Friendly Rate Control (TFRC) [8] and other congestion control algorithms, so as to insure fair integration with other traffic in the network.

To support the decoupling of functionalities, cmpSCTP uses several novel mechanisms including multi-buffer structure, multi-state management, two-level sequence number, and cooperative SACK strategy to realize effective bandwidth aggregation. Also cmpSCTP includes an overall retransmission technique that prevents the side effects of simultaneous transmission of data on paths with different characteristics, including unnecessary fast retransmissions, which ensures fast delivery of lost data chunks to prevent stalling the association.

Through extending dynamic address reconfiguration [10], as we show in Figure 1, cmpSCTP keeps ongoing end-to-end paths alive and provides adaptive load sharing in multiple paths. In addition, cmpSCTP extends SCTP path-monitoring feature, through regular transmission of actual effective data chunks, to update the list of unstable paths suitable for load sharing.

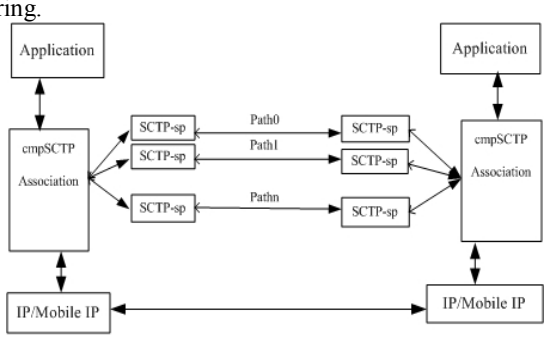

Fig.1. Multiple paths in a cmpSCTP Association

B. Multi-Buffer Structure

The single buffer architecture of SCTP has been replaced by a multi-buffer structure in cmpSCTP, meaning that each connection now has its own send buffer, and the total association has a single, shared send buffer. When a new chunk is received from the upper layer, the traffic scheduler is invoked to determine the path it will be sent over, the chunk is then queued into the chosen path send buffer.

However, a single receiver buffer is presented in cmpSCTP, like in standard SCTP, though each connection in the association is assigned a virtual buffer from the unique receiver buffer for per-path congestion control. In fact, all the chunks from different path are collected by a single association receiver buffer. At the receiver, the data chunks do not compete on association buffer, as the sender controls the amount of data injected on all the paths, based on its view of the free space in the receiver's association buffer, as well as the total outstanding data on all the available paths.

Since the beginning of the association, each single connection proceeds without interference from other connections, handling only those packets in its send buffer. When a new data chunk needs to be transmitted, it is inserted in the send buffer of the connection indicated by scheduling algorithm (Round-Robin, Bandwidth-aware, or other more optimal algorithm). From then on, it is the connection's responsibility to ensure that it is delivered: the data chunk remains in the send buffer until acknowledged; it causes Head-of-the-Line blocking to its own connection, but not to the other connections. When a retransmission timeout occur, a retransmission process is declared. The cmpSCTP will use an alternative path with lowest packet drop probability. In this case, lost chunks queued in the send buffer of the failed path are shifted into the send buffer of the path chosen for the retransmission.

C. Two-Level Sequence Number

The standard SCTP uses Transmission Sequence Number (TSN) as sequence number in its congestion control algorithms. However TSN might uses only one path

although it is used by entire association. Therefore we still continue to use the TSN as the sequence number that can be used for independent congestion control over each path.

The proposed cmpSCTP operates in two levels, involved in connection level and association level. Every level has its own sequence number. At connection level, we still use the TSN, which is not used for entire association, but still represents the sequence of SCTP DATA chunk transmitted through per path, used for reliability and congestion control on each path. TSN may split the first 4-bit as path ID (PID) denoted the transmission path, the remain 12-bit part represents per-path-sequential TSN. Data chunks sent to the same path are assigned same path ID (PID) and per-path-sequential TSN. At the same time, data chunks sent to different path carry different PID. Since TSN is used only by SPM to keep track of the states of data chunks sent through each path and per-path congestion control is activated.

At association level, the Association Sequence Number (ASN) of each data chunk, which is a per association sequence number, is used to reassemble all received data chunks from different paths to an integrated file. In other words, we use ASN to reorder the received data chunks at the receiver association buffer, regardless the path from which they have been received.

Such, we defined a new modified data chunk as Fig.2, by adding two new parameters to the standard SCTP data chunk [2]. The first parameter is a 4 bits Path Identifier (PID), which identifies the path used for the data chunk transmission. The second parameter is 16 bits Association Sequence Number (ASN), which is a monotonically increasing sequence number for the data chunks transmitted over the association. In addition, cmpSCTP continues to use the SSN for ordering the data chunks within the association streams.

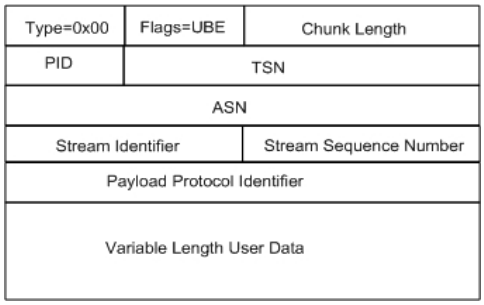

Fig.2. Modified cmpSCTP Data Chunk

D. Handling of cmp-SACK chunk

In order to acknowledge the received data chunks, cmpSCTP defines concurrent multi-path SACK (cmp-SACK) as Figure 3. The cmp-SACK chunks, which can also be called ASN-based SACK, received by a cmpSCTP source endpoint actually reflect the reception of ASNs sent through all the paths used by a cmpSCTP association. The cmp-SACK chunk carries several pieces of information includes four different parameters than the standard SCTP SACK, Cumulative ASN ACK, Time Stamp, Path ID and Advertised Receiver Window Credit.

• The Cumulative ASN Ack is a per association cumulative acknowledgement instead of Cumulative TSN ACK in the standard SCTP SACK, which records the highest sequential ASN received.

• Time Stamp is used to order the cmp-SACKs received from the different paths.

• Every Path IDs and their correspondent Advertised Receiver Window Credit, each of them reflects the current capacity of each receiver's inbound virtual buffer.

• A sequence of Gap Ack Blocks, which record any out of sequence ASNs received.

•A sequence of Duplicate ASNs, which record any ASNs for which duplicates have been received.

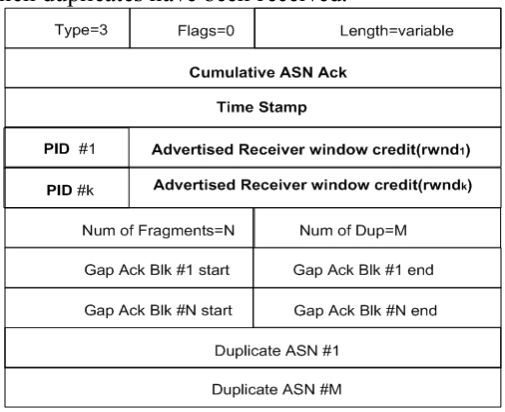

Fig.3. Modified cmpSCTP SACK

Considering the chunks of the same ASN is possible to receive from different paths on a certain redundant transmission strategy, acknowledgement mechanism should be design based on ASN to prevent unnecessary retransmissions unless data chunk losses ocurr on every transmission path. Fast retransmission is triggered by four consecutive duplicate SACKs at an association. Whenever a data chunk needs to retransmit, cmpSCTP will use an alternative path with packet lowest drop probability. Since in our concurrent multi-path transfer, current *cwnd*s of all paths are available, such modified retransmission scheme is more efficient than defined in RFC 2960: selected alternative path uses current *cwnd* to retransmit all lost packets. In such strategy, there are N disjointed paths between sender and receiver to transmit simultaneously and retransmit lost packets via alternative path with lowest packet drop probability.

Moreover, differing the standard SCTP SACKs run by each connection: when the SACK is received, it is processed only for those paths whose carried chunks were acknowledged by the SACK itself, the cmp-SACKs return on the path from which the last data chunk was received, but it is possible that a cmp-SACK is reporting information about other on-going connections of the same association. Thus, when a cmp-SACK arrives at the source, the information is processed on each interface and, consequently, all send buffers are refreshed. Since the information of the receiver window is required by sender per-path prior to the congestion window increase, this change avoids that a connection parameter out of date due to its *RTT* too large.

E. Flexible Multi-Path Management

The mobile SCTP (mSCTP) [9] is the SCTP with the ADDIP extension. It utilizes dynamic address reconfiguration [10] to manage the possible changes of IP addresses such as adding new addresses and deleting obsolete addresses while keeping ongoing end-to-end connections alive. However, the current SCTP specification designates only one path at each destination host as the primary path, and all new data is transmitted to the only primary path. If ever the primary path fails, new data transmission fails over to an alternate reachable destination path. Furthermore, the SCTP association experiences a reduction of traffic rates immediately after handover regardless of available traffic state of the new primary path.

In order to mitigate these negative effects, we refer to this new extension by cmpSCTP. It proposes to achieve higher throughput and seamless handover in an SCTP association by concurrently using all independent paths between a sender and receiver for data transfer.

By the cmpSCTP, we mean that the mobile host is maintaining connections with more than one path. These paths are classified by following two set:

The *Active_Set* includes the paths that form a cmpSCTP connection to the mobile host, which allows to transfer data packets for the MH.

The *Monitored_Set* is the list of candidate paths that the mobile hosts continuously measure, but their bandwidths is not enough to be added to the *Active_Set*.

The cmpSCTP handover support can be achieved by using modified Address Configuration Change (ASCONF) and Address Configuration Acknowledgement (ASCONF-ACK) control chunks of mSCTP, which may contain the four new modified and appendant request parameters for the paths. These parameters signal:

–0xC001-AddPath: the path specified is to be added to the *Monitored_Set*;

–0xC002-DeletePath: the path specified is to be removed from the *Monitored_Set*;

–0xC007-ActivePath: the path specified is to be added to the *Active_Set*;

–0xC008-DeactivePath: the path specified is to be removed from the *Active_Set*.

In a handover situation, MH sends CT ASCONF chunks with these four different types of parameters. To add new path, for example, the MH should send ASCONF chunk of 0xC001 type, which should be acknowledged by ASCONF-ACK chunk. Such through using the concurrent multi-path feature, cmpSCTP could reduce latency and increase throughput during performing vertical handover. This means that packets are not lost during the handover and there is no interruption to service, making it suitable for handover of real-time traffic.

IV. SIMULATION AND RESULT

The proposed cmpSCTP protocol has been implemented in the network simulator OPNET [11], and tested with various network configurations. The purpose of the extensive simulations is two-fold: first to investigate the performance of the proposed cmpSCTP with various network parameters, and second to compare the cmpSCTP protocol with other multi-path transport protocol.

In our simulation, we created the network topology consisting two hosts. To investigate the impact of various network parameters on the performance of the cmpSCTP, the multiple overlapping cells are also varied by using different simulation configurations including the number of overlapping cells, and available bandwidth in the cell the mobile host is entering. Available bandwidth in a cell is varied by changing the average of the Poisson distribution used to generate background traffic in all of cells.

A. Impact of Bandwidth Disparity

In this experiment we tested the robustness of the cmpSCTP protocol in dynamic conditions. We assumed that we have two paths, namely path 1 and path 2 in order to examine the performance of cmpSCTP under the condition of paths with diverse capacities. The speed of mobile host movement is assumed to be 15 m/s (meters per second) and the *RTT* is assumed to be 60 ms). In case 1, the mobile host moves from a cell with a larger amount of available bandwidth (2Mbps) to a cell with a smaller amount of available bandwidth (1Mbps). In case 2, a mobile host moves between two homogeneous cells with the same amount of available bandwidth (2Mbps). In case 3, a mobile host moves from a cell with a smaller amount of available bandwidth (2Mbps) to a cell with a larger amount of available bandwidth (5Mbps). In our performance study we used the association throughput as a performance metrics, which is defined as the amount of data delivered to the receiver's application layer per second.

Figure 4 shows the throughput during handover. The x-axis in the figure 6 represents the time, while the y-axis represents the effective association throughput excluding the duplicate packets. As can be seen from figure 9, that despite the difference in the bandwidths of the paths, the association throughput achieved by cmpSCTP is close to the ideal throughput during handover. The high throughput achieved by cmpSCTP is due to its striping mechanism that is based on the rate of the bandwidth of the paths.

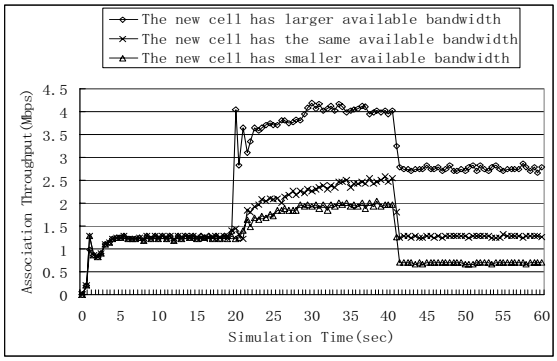

Fig.4. Association Throughput during handover between heterogeneous cells

B. Sensitivity to the Schedule Strategy

In our simulation, we created a cmpSCTP association

consisting two paths between the two multi-homed cmpSCTP hosts: cmpSCTP source and destination, which available bandwidth is 1Mbps and 2Mbps individually. For the sake of performance comparison, we also simulate the LS-SCTP and pTCP protocols in the same scenario. The simulation result represents the association throughput and the average delay of every packet.

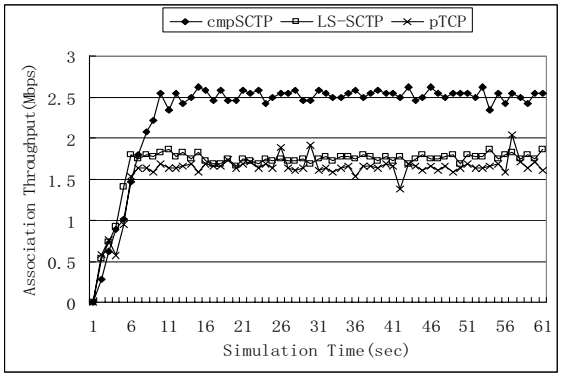

Fig.5. Throughput performance using various multi-path transport protocol

As can be seen from Figure 5, all three schemes show relatively small fluctuations in the throughput. The proposed cmpSCTP shows more high association throughput due to a series of efficient improvement mechanisms

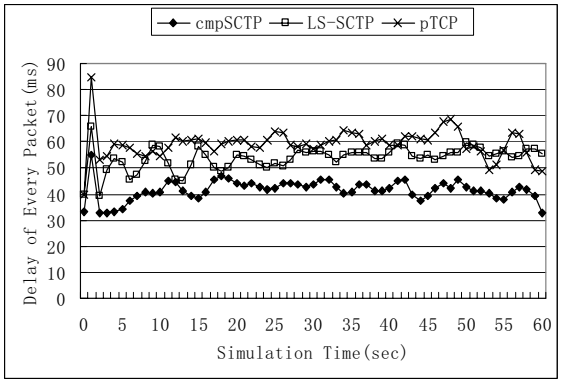

Fig.6. Delay performance using various multi-path transport protocol

Fig.6 shows the delay performance by performing the concurrent multi-path transfer feature. We can see that the end-to-end delay during performing cmpSCTP is much lower than the other multi-path transport protocol, while in most situations it brings noticeable improvements to jitter, packet loss percentage and reordering delays at the receiver.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a cmpSCTP protocol which highly coupled with Mobile IP to keep two or more end-to-end paths concurrent transferring new data from a source to a destination host. The cmpSCTP distributes the data on the available paths based on an estimation of the available bandwidth of each path. We presented the design and details of the proposed approach, and evaluated its performance through simulation experiments. Our simulation results demonstrated that the cmpSCTP can lead to satisfactory performance which is able to utilize the available bandwidth efficiently. We compared the performance of cmpSCTP with LS-SCTP and pTCP Results present that cmpSCTP dramatically outperforms LS-SCTP and pTCP in terms of throughput and delay especially in heterogeneous network environment.

Further investigation is planned to address some of the issues associated with impact studies on the other path factors such as packet loss rate, security and cost, and mechanisms to mitigate it. Also, the assumption of independent paths is being dropping off, we then plan to enable the sender to dynamically decide from either shared or distinct congestion control across paths through incorporating an end-to-end bottleneck detection mechanism [12]. The analysis and evaluation of these issues are our future work.

ACKNOWLEDGMENTS

We express our gratitude to the Research Institute of China Mobile. Thanks also to Yan Zhang, Cong Liu and the anonymous reviewers for their helpful suggestions which helped to improve the paper.

REFERENCES

[1] R. Chandra, P. Bahl, P. Bahl, "MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a Single Wireless Card", Proceedings of IEEE Infocom, 2004.

[2] R. Stewart, Q. Xie, K. Morneault, C. Sharp, et al, "Stream Control Transmission Protocol", RFC2960, Oct. 2000

[3] J. R. Iyengar, K. C. Shah, P. D. Amer, and R. Stewart, "Concurrent multipath transfer using SCTP multihoming," in Proceedings of SPECTS2004, San Jose, CA, USA, July 25–29, 2004.

[4] J. R. Iyengar, P. Amer, and R. Stewart, "Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths," IEEE/ACMTransactionsonNetworking, vol. 14, no. 5, pp. 951–964, Oct. 2006.

[5] A.A.E. Al, T. Saadawi, M. Lee, "LS-SCTP: a bandwidth aggregation technique for stream control transmission protocol", Computer Communications 27 (10) (2004) 1012–1024.

[6] H. Hsieh and R. Sivakumar, "A transport layer approach for achieving aggregate bandwidths on multi-homed mobile hosts". In Proc. ACM Mobicom, Atlanta, USA, Sept. 2002.

[7] M. Zhang, J. Lai, A. Krishnamurthy, L. Peterson, and R. Wang, "A transport layer approach for improving end-to-end performance and robustness using redundant paths," in Proceedings of the Usenix Annual Technical Conference,, Boston, MS, USA, June 27–July 2, 2004, pp. 99–112.

[8] M Handley, S Floyd, J Padhye and J Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification" RFC 3448, the Internet Society. Jan. 2003.

[9] M. Riegel and M. Tuexen, Ed., "Mobile SCTP", IETF Network Working Group, Internet Draft, Work in Progress, draft-riegel-tuexen-mobile-sctp-07.txt, October 22, 2006.

[10] R. Stewart, M. Ramalho, Q. Xie, M. Tuexen, and P. Conrad, "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration", draft-ietf-tsvwg-addip-sctp-17(work in progress), June 2007.

[11] OPNET simulator URL: http://www.opnet.com/

[12] D. Rubenstein, J. Kurose, and D. Towsley. "Detecting Shared Congestion of Flows Via End-to-end Measurement". IEEE/ACM Transactions on Networking, 10(3), June 2002.