Performance Evaluation of SCTP with Adaptive Multistreaming over LEO Satellite Networks

Hiroshi Tsunoda†, Nei Kato†, Abbas Jamalipour‡, and Yoshiaki Nemoto†
†Graduate School of Information Sciences, Tohoku University, Sendai, Japan 980-8579
E-mail:{tsuno, nemoto}@nemoto.ecei.tohoku.ac.jp, kato@it.ecei.tohoku.ac.jp
‡University of Sydney, Australia
E-mail: a.jamalipour@ieee.org

Abstract—**In this paper, we evaluate the performance of Stream Control Transmission Protocol (SCTP) with adaptive multistreaming which we had previously proposed. This proposed modification can be useful for the resource limited mobile terminals in LEO satellite networks. In the proposed modification, a SCTP sender adaptively enables or disables multistreaming based on the comparison between the estimate of available bandwidth and current congestion window size. By doing this, user terminals use multistreaming feature appropriately and can avoid waste of resource caused by unnecessary use of multistreaming. In this paper, we propose a further modification to the adaptive multistreaming and evaluate its performance.**

I. Introduction

Satellite networks have global coverage and can provide connectivity to end users at anytime and anywhere in the world. Global coverage helps mobile users and users in the areas with no terrestrial network infrastructure to connect the Internet. Satellite networks are categorized as GEO (Geosynchronous), MEO (Medium Earth Orbit) and LEO (Low Earth Orbit), based on the altitude of satellite. Among them, more research attention has been focused on LEO satellite networks connected with ISLs (Inter Satellite Links), because LEO satellite networks have several advantages: lower propagation delay, fewer risk of propagation losses, coverage for high latitude region [1]. A global network can be developed if LEO satellite networks are integrated to the existing IP networks through TCP/IP protocols. Such IP/LEO satellite integrated networks will be useful to provide a wide variety of IP-based applications, such as teleconferencing and contents-delivery services. Moreover, considering the independence from terrestrial networks, LEO satellite networks have a great role to support emergency communication, information gathering, and information provision systems in the emergency situations. Supporting such systems require reliable and efficient transport protocols for IP/LEO satellite networks.

Transmission Control Protocol (TCP) is the dominant protocol in the current Internet. However, it is not suitable for LEO satellite networks, which are characterized by sudden change of route and the number of flows due to frequent handover and corruption losses. Therefore, several enhancements for TCP over LEO satellite networks have been proposed [2]–[4].

Meanwhile, in recent years, new transport protocol called Stream Control Transmission Protocol (SCTP) [5] has been standardized by the Internet Engineering Task Force (IETF). The most distinguishing features of SCTP are multihoming and multistreaming. These new features are expected to help improving the performance of SCTP in satellite networks, and Fu *et al.* [6], [7] investigated the suitability of some features and showed that SCTP is a suitable transport protocol for satellite networks.

Recently the authors have proposed an modification of SCTP to efficiently utilize multistreaming feature for resource limited user terminals in LEO satellite networks [8]. In the proposed modification, a SCTP sender adaptively enables or disables multistreaming based on the comparison between the estimate of available bandwidth and current congestion window size. By doing this, user terminals use multistreaming feature appropriately and can avoid waste of resource caused by unnecessary use of multistreaming. However, the performance of the proposed modification was evaluated by the limited set of simulations. Thus, we give extensive performance evaluation of the proposed modification of SCTP in this paper.

The reminder of the paper is structured as follows. Section II gives a brief description of standard SCTP and explains the mechanism of the proposed SCTP with adaptive streaming. In Sec. III, we evaluate the effect of the number streams on the goodput of SCTP communication and propose a modification of the algorithm for setting the number of streams. The performance of the proposed method is evaluated in Sec. IV. Concluding remarks are in Sec. V.

II. SCTP with Adaptive Multistreaming

A. Overview of SCTP

SCTP is the next-generation transport protocol and provides in-order and reliable data delivery to upper layers as TCP. SCTP has the core feature of TCP and additional unique features which do not exist in TCP. Congestion control of SCTP is based on the window-based congestion control scheme of the TCP. Slow start, congestion avoidance, fast retransmission, and fast recovery [9] are supported in SCTP. Although selective acknowledgement (SACK) [10] is optional in TCP, use of SACK is mandatory in SCTP.

Multihoming and multistreaming are the most distinguishable features of SCTP from TCP. In TCP, a connection between two endpoints cannot use multiple IP address pairs (i.e. source and destination). However, multihoming feature enables an SCTP association, which is analogous to connection

1-4244-0939-X/07/$25.00 © 2007 IEEE

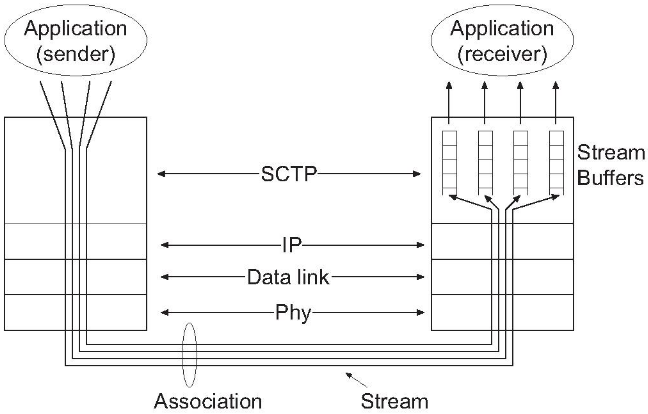

Fig. 1. Data transfer using multiple streams within a SCTP association

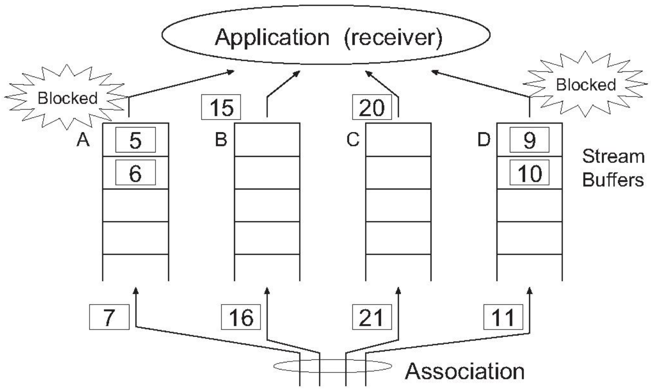

Fig. 2. Avoidance of head of line blocking using multiple streams

in TCP, to span across multiple IP address pairs. By using multiple IP address pairs for an association, endpoints can communicate through different paths.

Thanks to the multistreaming feature, a SCTP sender can prepare several streams in an association and split data from application layers into multiple streams. In Fig. 1, applications at a sender and a receiver are communicating through four streams. At the receiver side, a buffer is allocated for each stream and is used for maintaining data sequence in a stream. Since sequencing of data is performed within each stream, if a packet in a certain stream is dropped, other streams can continue to deliver packets and pass data to the application. By doing this, SCTP can mitigate the Head of Line (HoL) blocking problem found in TCP.

Fig. 2 explains the HoL blocking problem. There are four streams (A, B, C, and D) in an association in the figure. Squares with digits in the figure mean segments, and digits in squares indicate stream sequence numbers (SSNs). SSNs are unique number in a stream, and segments in a stream are identified by SSNs. Although both segment 15 in stream B and segment 20 in stream C are delivering to the application, two segments are queued in the buffers of streams A and D, respectively. This is because that segment 4 in stream A and segment 8 in stream D are lost in the network, and these streams are waiting for retransmission of the lost segments. Until the lost segments are arrived, these streams are remained blocked and cannot send segments to the application. On the other hand, stream B and C can continue to send segments to the application because of independence of streams. Since wireless link error in LEO satellite networks results in packet losses, this feature is useful for efficient communication in LEO satellite networks.

B. Adaptive Multistreaming based on the Available Bandwidth

In the proposed method called SCTP with adaptive multistreaming, the change of the number of streams is triggered by packet losses. Fig. 3 presents the pseudo code of the basic algorithm for setting the number of streams. When a sender notices packet losses by four duplicate ACKs or retransmission timeout, the sender compares congestion window size ($Cwnd$) and the estimated available bandwidth (RE) estimated by the TCP westwood rate estimation algorithm [11]. According to [11], if RE is significantly smaller than $Cwnd$, it is more likely that packet losses are due to congestion. The authors also showed that $Cwnd$ which is larger than $RE \times 1.4$ indicates congestion condition. Therefore, in our method, the sender disables multistreaming feature and sets the number of streams to one in the case that $Cwnd$ which is larger than $RE \times 1.4$. Otherwise, the sender enables the multistreaming feature and sets the number of streams to four.

For avoiding frequent fluctuation of the number of streams, the proposed method gives a little strict condition to disable multistreaming in the case that multistreaming is already enabled. Once multistreaming is enabled, multistreaming state continues until $Cwnd$ becomes larger than $RE \times 1.6$.

If a link condition becomes better right after the sender enables multistreaming, the sender has to maintain unnecessary streams until the next packet loss occurs, which may not occur for a long time. Therefore, the sender performs this judgement at every five seconds while multistreaming is enabled. Then, the sender disables multistreaming if no retransmission occurs in the last five seconds.

III. Effect of the Number of Streams

In our previous proposal, the number of streams is predefined value. However, the appropriate number of streams can be different for different situations. Thus, in this section, we evaluate the effect of the number of streams on the goodput of the SCTP communication through computer simulation, using the Network Simulator, NS-2 [12]. Then, we propose the modified algorithm for setting the number of streams.

A. Simulation Environment

Fig. 4 shows the topology that we used in our simulations. Ten pairs of SCTP senders and receivers are in the network, and thus there are ten associations. The bandwidth of ISLs and GSLs (Ground-Satellite Links) are set to 10Mbps. The propagation delay of ISLs and GSLs are set to 13ms and 5ms, respectively. These delay settings are based on Iridium system

```
if ((4 duplicate ACKs are received)
    || (Retransmission Timeout))
    if (Multistreaming is enabled)
        weight = 1.6
    else
        weight = 1.4
    endif
    /* Cwnd: congestion  */
    /*       window size */
    /* RE: estimated available */
    /*     bandwidth           */
    Diff = Cwnd - RE * weight
    if (Diff > 0)
      /* Congestion condition. */
      /* Disable multistreaming */
      The number of streams = 1
    else
      /* Likely link error. */
      /* Enable multistreaming */
      The number of streams = 4
    endif
endif
```

Fig. 3. Algorithm for setting the number of streams

[13]. The RTT between a sender and a receiver depends on the number of hops (H). Queues on all nodes use Drop-Tail as packet-discarding policy, and its size equals to the bandwidth-delay product. For simplicity, packet losses due to link error occur only in the up-link of the senders' GSLs, and all other links (i.e. ISLs and the receiver's GSL) are assumed to be error-free.

We assume that a SCTP sender has a SCTP-aware application which provides infinite data, and thus SCTP-aware FTP is used throughout the evaluation. The data packet size is fixed to 1,500 bytes. Since multistreaming is useful for the case that receiver buffer size is limited, we set the buffer size of each receiver to 10Kbytes. The start time of each SCTP sender is a random variable uniformly distributed from 0 to 5 seconds to avoid bursty losses at the simulation launch time, and each sender sends data packets for 60 seconds.

B. Simulation Results

We measured the goodput of SCTP communication for different segment error rates in the senders' GSLs. Goodput is defined as the total number of segments (without considering the retransmitted segments) reaching the destination during the simulation time. In order to illustrate the effect of the number of streams on the goodput, we calculate the goodput increase ratio (δ_n) defined as follows:

$$\delta_n = \frac{\lambda_n - \lambda_1}{\lambda_1} \times 100 \qquad (1)$$

where n denotes the number of streams and λ_n is the average goodput of n streams.

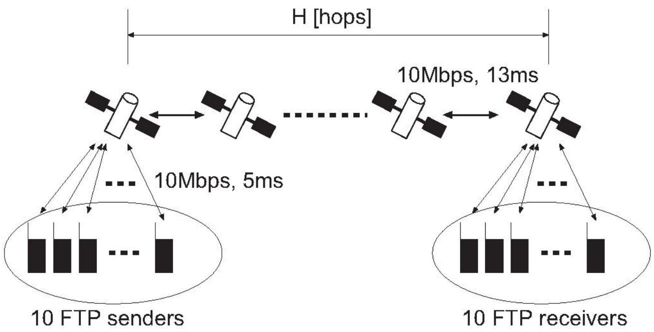

Fig. 4. Simulation environment.

Figures 5–10 summarize the goodput increase ratio for different numbers of hops H. From these figure, we can see that higher link error rates requires more streams for achieving the maximum goodput. In addition, the additional streams becomes useless as round trip time between a sender and a receiver increases. Although we do not show the evaluation result for the larger number of hops due to the space limitation, the omitted graphs are similar to Fig.10.

The reason for these results is that the multistreaming feature improves the goodput by mitigating the HoL blocking at a receiver. Obviously, the HoL blocking occurs frequently in the case that the link condition is bad and many segments are lost. If the link error rate is same, the occurrence frequency of the HoL blocking depends on the sending rate from a sender. Since a sender can quickly increase its sending rate if the association has small RTT, the association is suffered from the segment losses and the HoL blocking. Hence, increase of the number of streams is effective for higher link error rates and small RTT.

These results indicate that it is difficult to decide the appropriate number of streams in advance of communication because the appropriate number varies depending on environment. Thus, we modify the previously proposed algorithm to increment or decrement the number of streams according to bandwidth utilization in a wireless link.

C. Modified Algorithm for Setting the Number of Streams

Figure 11 is the modified algorithm for setting the number of streams. Unlike the previous algorithm, the number of streams is not pre-defined value. When a sender detects a packet loss, the sender increments or decrements the number of streams by one. If the sender judges the cause of the packet loss is link error, the sender begin to use a new stream. If the packet loss occurs due to congestion in the network, the sender stops using one of the streams. This is because the large number of streams do not help to improve goodput under congestion.

In the next section, we analyze the performance of the proposed adaptive stream allocation method in detail.

IV. Performance Evaluation

In this evaluation, we use the same simulation environment explained in Sec. III-A. For validating the adaptiveness of the

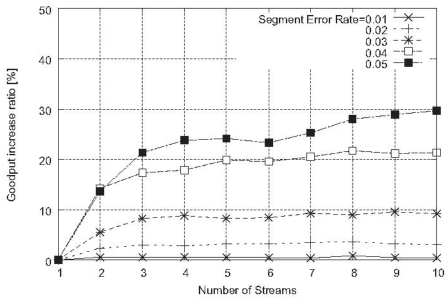

Fig. 5. Goodput increase ratio (H=1)

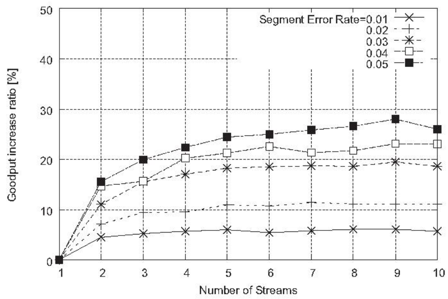

Fig. 6. Goodput increase ratio (H=2)

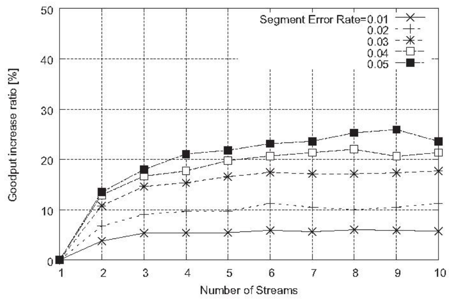

Fig. 7. Goodput increase ratio (H=3)

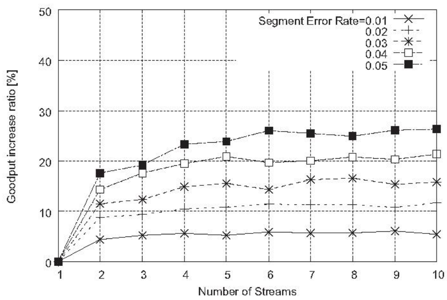

Fig. 8. Goodput increase ratio (H=4)

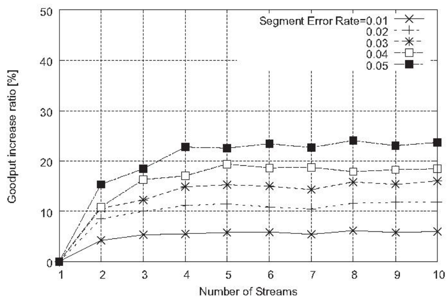

Fig. 9. Goodput increase ratio (H=5)

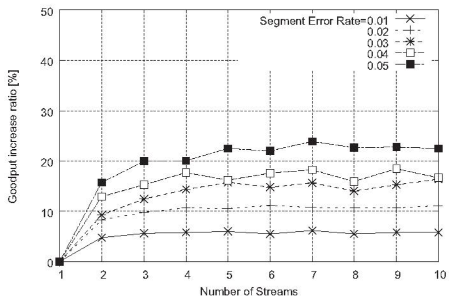

Fig. 10. Goodput increase ratio (H=6)

```
if ((4 duplicate ACKs are received)
    || (Retransmission Timeout))
    if (Multistreaming is enabled)
        weight = 1.6
    else
        weight = 1.4
    endif
    Diff = Cwnd - RE * weight
    if (Diff > 0)
      /* Congestion condition. */
      number of streams
        = number of streams - 1
    else
      /* Likely link error. */
      number of streams
        = number of streams + 1
    endif
endif
```

Fig. 11. Modified algorithm for setting the number of streams

proposed method, the link quality of senders' GSLs is changed at 15 seconds and 45 seconds, respectively. At first, the link qualities of senders' GSLs are good and each link is error-free. Then the segment error ratio becomes 0.05 at 15 seconds. At 45 seconds, the senders' GSLs become error-free again.

Fig. 12 shows the average goodput for different values of the number of hops. When the number of hops is small, the goodput of the proposed method (SCTP with adaptive streaming) is larger than that of the SCTP using four streams. As the number of hops increases, the difference between goodput of both method becomes small. This is because four streams are enough for getting sufficient goodput when the RTT between a sender and a receiver is large, as mentioned in Sec. III-B. Note that the goodput of SCTP using one stream never overwhelms that of SCTP using four streams and the proposed method. This result explains the advantage of multistreaming.

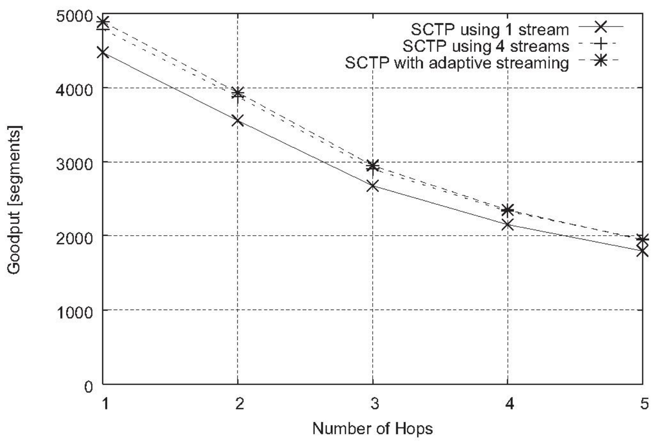

Fig. 12. Goodput.

In order to demonstrate that the adaptive streaming of the proposed method is useful for getting larger goodput, we show the variation of the number of streams in the proposed method in Fig. 13. Then, we compare the variation of the cumulative acknowledgement number between the proposed method and SCTP using four streams in Fig. 14. In these figures, the number of hops is one and the segment error ratio from 15 seconds to 45 seconds is 0.05.

In Fig. 13, all senders start to increase the number of streams after 15 seconds. As a result, the number of streams of all senders become ten. Since a sender of the proposed method disables multistreaming if no retransmission occurs in the last five seconds and there are no segment losses due to link error after 45 seconds, all senders reset the number of streams after 45 seconds.

The typical effect of the adaptive streaming is illustrated in Fig. 14. In the case of SCTP using four streams, the increase of the cumulative acknowledgement number becomes slowly from 15 seconds to 45 seconds. On the other hand, in the case of the proposed method, the slope of the cumulative acknowledgement number does not show drastic change during the segment error rate is 0.05. This is because the proposed method can effectively mitigate the HoL blocking by using many streams in the timely manner.

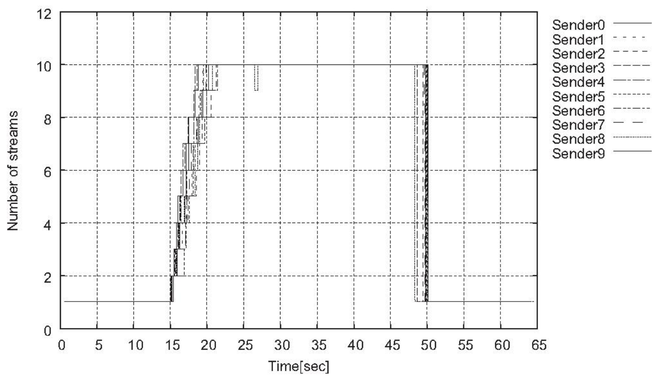

Fig. 13. Variation of the Number of Streams.

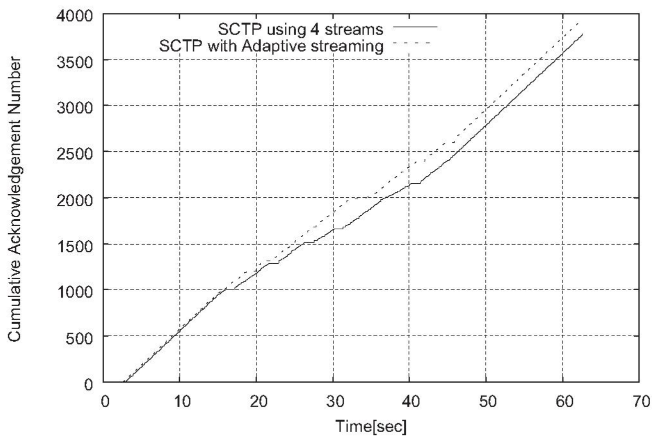

Fig. 14. Example of Cumulative Acknowledgement Number

V. Conclusions

In this paper, we focused on the performance evaluation of the Stream Control Transmission Protocol (SCTP) with adaptive multistreaming. The adaptive multistreaming which we had previously proposed is the required modification for the resource limited user terminals (e.g. PDAs and smart phones) in LEO satellite networks. In the our previous proposal, the number of streams when the multistreaming is enabled should be pre-defined. However, in the performance evaluation, we found that the appropriate number of streams is different for the different environment. Therefore, we propose a further modification enabling a sender to increment and decrement the number of streams according to the condition of wireless links. Through simulations, we confirmed that the validity of the proposed method.

Acknowledgment

The authors would like to thank Strategic Information and Communications R&D Promotion Programme (SCOPE), Ministry of Internal Affairs and Communications, Japan for supporting this work.

References

[1] A. Jamalipour, *Low Earth Orbital Satellites for Personal Communication Networks*, A. Jamalipour, Ed. Artech House Publishers, 1998.

[2] I. Akyildiz, G. Morabito, and S. Palazzo, "TCP-peach: A New Congestion Control Scheme for Satellite IP networks," *IEEE/ACM Transactions on Networking*, vol. 9, no. 3, pp. 307–321, Jun. 2001.

[3] H. Tsunoda, K. Ohta, N. Kato, and Y. Nemoto, "Improving TCP Performance Using Observed Hop Count Over LEO Satellite Networks," in *Proc. of 21st AIAA International Communications Satellite System Conference*, Apr. 2003.

[4] T. Taleb, N. Kato, and Y. Nemoto, "REFWA: An Efficient and Fair Congestion Control Scheme for LEO Satellite Networks," *IEEE/ACM Transaction on Networking*, vol. 14, no. 5, pp. 1031–1044, Oct. 2006.

[5] R. Steward, Q. Xie, M. Morneault, C. Sharp, H. Schewarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, "Stream Control Transmission Protocol," RFC2960, Oct. 2000.

[6] S. Fu, M. Atiquzzaman, and W. Ivancic, "SCTP over Satellite Networks," in *IEEE 18th Annual Workshop on Computer Communication*, Oct. 2003, pp. 112–116.

[7] ——, "Evaluation of SCTP for Space Networks," *IEEE Wireless Communications*, pp. 54–62, Oct. 2005.

[8] H. Tsunoda, A. Jamalipour, and Y. Nemoto, "An Adaptive Stream Allocation Method for Improving SCTP Performance over LEO Satellite Networks," in *In Proceedings of 25th AIAA International Communication Satellite System Conference (ICSSC2007)*, Apr. 2007.

[9] M. Allman, V. Paxson, and W. Stevens, "TCP Congestion Control," RFC2581, Apr. 1999.

[10] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP Selective Acknowledgement Options," RFC2018, Oct. 1996.

[11] R. Wang, M. Valla, M. Sanadidi, B. Ng, and M. Gerla, "Efficiency/Friendliness Tradeoffs in TCP Westwood," in *7th IEEE Symposium on Comupter and Communications*, Jul. 2002.

[12] S. McCanne and S. Floyd, *ns Network Simulator*, http://www.isi.edu/nsnam/ns/.

[13] "Iridium," http://www.iridium.com/.