On how to Mitigate the Packet Reordering Issue in the Explicit Load Balancing Scheme

Tarik Taleb, Daisuke Mashimo, Kazuo Hashimoto, Nei Kato, and Yoshiaki Nemoto
Graduate School of Information Sciences, Tohoku University, Japan
{taleb, kh}@aiet.ecei.tohoku.ac.jp
{mashimo, kato}@it.ecei.tohoku.ac.jp
nemoto@nemoto.ecei.tohoku.ac.jp

***Abstract*—Efficient load balancing is an important requirement for intelligent engineering of traffic over all-IP satellite networks. In this regard, the authors have recently proposed an Explicit Load Balancing (ELB) scheme for next-generation LEO/MEO satellite systems. In ELB, a congesting satellite requests its neighboring satellites to forward a portion of data, originally destined to travel through the satellite, via alternative paths that do not involve the satellite. While this feature yields better traffic distribution and reduces the overall packet drops that may occur at the congesting satellite, it raises the so-called packet reordering issue. In connection-oriented protocols such as TCP, an out-of-order reception of packets generates duplicate acknowledgments that result in a gratuitous halving of the congestion window. This ultimately degrades the overall throughput of the network. To cope with packet reordering issue in ELB, we suggest some minor modifications to the TCP implementation at the receiver side to enable receivers to judge the actual reason beneath the out of order reception of packets. We compare the performance of our method to that of standard TCP and TCP-PR, a recently proposed scheme for persistent packet reordering. Depending on the traffic characteristics and the satellite constellation type, discussion on the advantages and pitfalls of each scheme is given.**

I. Introduction

Satellite communication systems are regaining ground at a tremendous pace. For network operators, they are envisioned as an important and scalable access technology to solve the last mile problem and to realize the vision of global ubiquitous systems. In academia, they have been the focus theme of many researchers and have generated a large library of researches in the recent literature.

Communications via satellites have first commenced with the use of satellites in geostationary orbits. Recent trends in satellite-related researches have been towards non-geostationary (NGEO) satellites [1] [2], known as Low Earth Orbit (LEO) or Medium Earth Orbit (MEO) satellites, and that is for a number of reasons (e.g., need for lower propagation delays, lower terminal power requirements, and high-quality coverage of high latitude regions).

Within the researchers' community, it has been almost agreed that next-generation NGEO satellite systems should be designed to support broadband applications similar to today's Internet [3]. Furthermore, the worldwide acceptance of the Internet and the universality of its core protocol, the Internet Protocol (IP), will drive satellite systems into all IP. One of the key challenges in realizing the vision of such all-IP satellite systems consists in the development of a load distribution-aware scheme for routing traffic over satellite constellations. Indeed, the high variance in the users' density and the frequent topological variations of NGEO satellite constellations lead to a non-uniform distribution of traffic over satellite constellations. Effectively, some satellites get overloaded with data packets while others remain underutilized. In the absence of an efficient routing algorithm, significant packet drops and excessive queuing delays may be experienced at the overloaded satellites.

In the recent literature, a number of routing protocols have been specifically proposed for satellite networks. A common feature among most of these protocols consists in their focus on searching for the shortest path with the minimum routing cost without any consideration of the total traffic distribution over the entire constellation. To reflect network conditions in the routing decision, and to hence guarantee a better distribution of traffic over the satellite constellations, the authors have recently proposed a routing protocol dubbed Explicit Load Balancing (ELB) [4]. The key idea behind ELB consists in the enabling of explicit exchange of current congestion state among only neighboring satellites. To avoid an imminent congestion, a satellite with high traffic load requests its neighboring satellites to forward a portion of data, originally destined to travel through the satellite, via alternative paths that do not involve the satellite. While this traffic detouring feature ensures better traffic distribution and reduces the overall packet drops that may occur at the congesting satellite, it raises the so-called packet reordering issue. While this phenomenon does not affect connectionless-oriented protocols such as User Data Protocol (UDP), in case of Transmission Control Protocol (TCP), it results in the transmission of duplicate acknowledgments, unnecessarily halves the congestion window of TCP, and ultimately degrades the throughput.

To cope with the packet reordering issue in ELB, this paper argues the addition of a simple control mechanism to TCP receivers to enable them judge whether the out-of-order reception of packets is due to congestion or simply to changes in the communication path. The basic idea behind the suggested control mechanism consists in a simple monitoring of the Time-to-Live (TTL) field of packet headers at receivers. If the reception of a packet in an out-of-order

1-4244-1376-1/07/$25.00 ©2007 IEEE.

manner is followed by no change or an increase in the TTL value, the receiver judges the out-of-order reception as due to change in the communication path and accordingly does not send a duplication acknowledgment. If the out-of-order reception is preceded by a decrease in the TTL value, the receiver can consider it as a congestion indication and accordingly acknowledge the sender by sending it a duplicate acknowledgment. Receivers become thus capable of differentiating between out-of-order reception induced by congestion and that due to change in the communication path. Simulations are conducted to evaluate the performance of the proposed control mechanism against that of standard TCP. The packet reordering issue can be further solved by using the TCP-PR (Persistent Reordering) [5], a recently proposed scheme for persistent packet reordering. In light of the complexity and significant overhead of TCP-PR (compared to our proposed control mechanism), guidelines on which scheme to use are given while taking into account traffic characteristics, namely the ratio of delay-non sensitive traffic rate to that of delay-sensitive traffic, and the satellite constellation type (LEO or MEO).

The remainder of this paper is structured as follows. Section II highlights some research work in the field of routing over satellite networks. Section III briefly describes the ELB scheme and highlights the distinct features that are incorporated in the proposed control mechanism. Section IV portrays the simulation philosophy and discusses the simulation results. The paper concludes in Section V.

II. Related Work

Routing over mobile satellite networks has been an interesting area of research for a large population of researchers. A plethora of routing protocols has been thus proposed in the recent literature. A thorough discussion on the credits and pitfalls of notable routing protocols is given in [6]. A common feature of most conventional routing protocols consists in the fact that they base their routing decision on only propagation delay without paying attention to queuing delays that may be significant in case of heavy loads.

A. Load Balancing in Satellite Networks

To reflect queuing delays in the routing cost metric, several researchers have investigated the idea of incorporating load balancing functions in the routing procedure. In [7], Kucukates et al. propose a Minimum Flow Maximum Residual (MFMR) routing protocol that selects the minimum-hop path with the minimum number of flows. One of the main drawbacks of the MFMR protocol consists in the fact that it implies knowledge of the flows over the constellation and does not consider the case where the flows count increases along the selected path. Considering the fast motion of satellites, changes in flows count during the communication time is highly possible. This would lead to the congestion of the selected MFMR paths and ultimately unfavorable performance. The Probabilistic Routing Protocol (PRP) [8] uses a cost metric as a function of time and traffic load. The traffic load is assumed to be location homogeneous. The major drawback of the protocol consists in this assumption as it is far away from being realistic. Indeed, newly coming traffic can easily congest the chosen PRP path and leave other resources underutilized. In [9], Jianjun et al. propose a Compact Explicit Multi-path Routing (CEMR) algorithm that bases its cost metric on both propagation and queuing delays. Queuing delay is predicted by monitoring the number of the packets in an outgoing queue of each satellite every time interval. It is assumed that the network state over each time interval is updated before the routing calculation is carried out. However, this cost metric does not reflect the congestion state of the downstream satellite and consequently does not reflect the likelihood of packets to be dropped at downstream hops. To cope with the above mentioned limitations, the authors proposed the ELB scheme [4]. As previously said, ELB exhibits interesting features: better traffic distribution, congestion alleviation, and packet drop avoidance. It however leads to the packet reordering issue.

B. Impact of Packet Reordering on TCP

In [10], it is shown that packet reordering has a negative effect on TCP throughput. Indeed, current implementations of TCP work on the assumption that out-of-order packets indicate network congestion and unnecessarily cut their congestion window. They thus perform poorly when packets are reordered. Such packet reordering may occur in different networks, particularly in satellite communication systems where different paths can be involved in communication as in ELB. To cope with packet reordering, different schemes have been proposed in literature [11]-[13]. The most recent and most outperforming method is the TCP-PR protocol [5]. In TCP-PR, detection of packet losses is made through the use of timers rather than duplicate acknowledgments. Indeed, packets are assumed to be lost only if their corresponding acknowledgments do not arrive within a predefined time. In the design of TCP-PR, worst-case analysis and Internet traces are referred to for appropriate setting of timers. While TCP-PR does not require any modifications at the receiver side, and is therefore ”backward compatible” with any TCP receiver, it adds significant complexity and incurs important overheads, in terms of both computation and memory, at the sender side. In the following section, we suggest a simple modification (based on a simple comparison equation) to the TCP implementation at the receiver side to cope with packet reordering in ELB. The utility of TCP-PR and our proposed scheme is discussed based on traffic characteristics and satellite constellation type.

III. Proposed Packet-Reordering Recovery Mechanism

Before delving into details of the proposed packet-reordering recovery mechanism, there is firstly a brief description of the ELB scheme.

A. ELB Overview

ELB is exclusively designed for multi-hop NGEO satellite constellations. Depending on its queue ratio (Q_r: current queue occupancy to the total queue size), an ELB-implementing satellite resides in one of the three following states:

1) Free State (FS): when Q_r is inferior to a predefined threshold α.
2) Fairly Busy State (FBS): when Q_r is between the threshold α and another predetermined threshold β.
3) Busy State (BS): when Q_r exceeds the threshold β.

The key idea behind the ELB scheme consists in enabling neighboring satellites to constantly exchange explicit information on the states of their queue occupancies. Indeed, when a satellite A experiences a state transition from free to fairly busy, it sends a warning message to its neighboring satellites informing them that it is about to get congested. The neighboring satellites are then requested to update their routing tables and start searching for alternate paths that do not include satellite A. When the satellite enters the busy state, it transmits a Busy State Advertisement (BSA) signaling packet requesting the neighboring satellites to reduce their sending rates of traffic destined to satellite A by a Traffic Reduction Ratio (TRR) χ. The $(1 - \chi)$ portion of traffic data will be transmitted via the alternate paths retrieved earlier. BSA packets are broadcasted merely upon a state transition and only to the neighboring satellites (not over the entire connection path). They thus do not incur any significant overhead, in terms of neither bandwidth consumption nor scalability.

The key philosophy behind the setting of the queue ratio thresholds, α and β, is to reflect the packet discarding probability in these two parameters so as to avoid packet drops when a satellite is running under heavy loads. Indeed, when traffic load gets heavy at a given satellite and the packet drop probability gets a high value, α and β are set to small values so as the satellite would quickly transit to the busy state and neighboring satellites would promptly reduce their sending rates to avoid possible congestion and packet drops at the satellite. As for the setting of χ, it aims at ensuring a long enough recovery time for satellites before they enter again the busy state and request again their neighboring satellites to further reduce their sending rates. It also ensures that the detoured portion of traffic does not experience further detouring along the selected path till the destination, an issue referred to as traffic redistribution cascading. Details on the settings of α, β, and χ can be found in [4].

B. Packet-Reordering Recovery Mechanism

TCP usually misinterprets packet reordering as an indication of network congestion and unnecessarily cuts its congestion window. Indeed, when a packet arrives at the receiver out of order, the receiver immediately sends back an ACK to inform the sender that a packet with a certain sequence number is missing. Such an ACK is referred to as a duplicate ACK (DupACK). The sender retransmits the missing packet when it receives more than three DupACKs with the same sequence number. After retransmission, the sender reduces its window size to half and enters the congestion avoidance phase. Being unaware of the underlying reason beneath the out-of-order reception of packets, the sender misinterprets the event as an indication of network congestion and gratuitously throttles its transmission rate. In case of Newreno based TCP variant [14], Partial ACKs (ParACKs) are used to indicate the occurrence of multiple losses in a single window. Upon reception of a ParACK, the sender retransmits the lost packet and waits for an ACK to come back. To retransmit multiple lost packets, multiple Round Trip Times (RTTs) are thus required. This, coupled with the fact that satellite links exhibit relatively long delays, means that the TCP sender may necessitate a long time to increase its congestion window to its value before entering the fast retransmit phase. This leads to a drastic under-utilization of the network resources.

Algorithm 1 Pseudo code of the proposed packet reordering recovery mechanism.

```
Upon packet arrival
if Packet arrival in order then
    Store $TTL = TTL_{in-order}$
    Reset timer
    Send back ACK
else
    Check new $TTL$
    if $TTL \geq TTL_{in-order}$ then
        Set a timer
        if Timer expires then
            Send DupACK
        else
            Send normal ACK
        end if
    else
        Send DupACK
    end if
end if
```

To avoid such an unnecessary shrinkage of transmission rate due to packet reordering, we suggest that receivers refer to the TTL field of packets to judge whether the out-of-order in the reception of packets is due to congestion or simply to changes in the communication path. In case of IPv6, the use of the TTL field can be substituted by the Hop Limit field. Algorithm 1 portrays the pseudo code of the proposed packet reordering recovery mechanism.

Upon reception of a packet in order, a TCP receiver immediately sends back a normal ACK to the sender similar to the ordinary behavior of TCP. The receiver records then the TTL information available at the header of the received packet as $TTL_{in-order}$. When the receiver receives a packet in out-of-order, two cases can be envisioned. If the number of hops traversed by the received packet is the same or smaller compared to the previously received packet, in other words,

$$TTL \geq TTL_{in-order} \tag{1}$$

the receiver interprets the incident as due to changes in the communication path (Fig. 1). Acknowledgment packets are hold for a time interval. In this way, throughput degradation

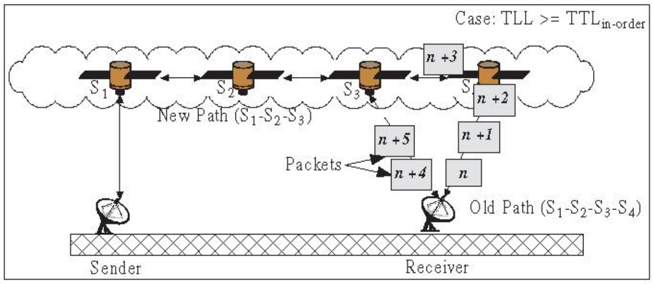

Case: TLL >= TTL$_{\text{in-order}}$
S_1
S_2
S_3
S_4
New Path (S_1-S_2-S_3)
$n+3$
$n+2$
$n+1$
n
$n+5$
$n+4$
Packets
Old Path (S_1-S_2-S_3-S_4)
Sender
Receiver

Fig. 1. When the communication path changes and the number of traversed hops decreases, packets traversing the new path may arrive earlier than some other packets that may be still in flight on the old path.

due to unnecessary transmission of duplicate ACKs can be prevented. If the missing packet does not arrive within the time interval, retained ACKs are returned requesting the TCP sender to retransmit the missing packets.

If Inequality (1) does not hold, the currently received packet was transmitted through a longer path than the previously received packet. Therefore, the receiver judges the out of order reception of packets as due to a packet discard and returns a duplicate acknowledgment. In other words, it proceeds in the same way as an ordinary TCP receiver. The sender retransmits the dropped packets and reduces its window size to half. Observe that the proposed operation can be accomplished without changing the protocol and requires a merely simple modification at only the receiving terminal. It is thus compatible with any TCP sender.

IV. Performance Evaluation

A. Simulation Setup

The performance evaluation is based on computer simulations using the Network Simulator (NS) [15]. In the performance evaluation, a multi-hop NGEO satellite constellation is envisioned. The constellation consists of S satellites with on-board processing capabilities, evenly and uniformly distributed over N orbits, forming a mesh network topology. Each satellite is able to set up a maximum of M links with its neighboring satellites. These links are called Inter Satellite Links (ISLs) and their delays are denoted as L. Satellites are assumed to be aware of their neighboring satellites. Different satellite constellations are considered by changing the parameters, S, N, M, and L (e.g., Iridium $S = 66, N = 6, M = 4, L = 15$ms). In the simulations, the four parameters (S, N, M, and L) are carefully chosen to ensure global coverage. Uplinks, downlinks, and ISLs are each given a capacity equal to 25Mbps. The average packet size is set to 1KB. Drop-Tail based buffers of lengths equal to 200 packets are used. Simulations are run for 60s. 600 non-persistent On-Off flows are used to generate background traffic. The On/Off periods of the connections are derived from a Pareto distribution with a shape equal to 1.2. The average burst time and the average idle time are set to 50ms. The source and destination end-terminals are dispersed all over the Earth, divided into six continental regions, following the same traffic distribution used in [16]. The background sources send data at constant bit rates from within the range of 0.8Mbps to 1.5Mbps.

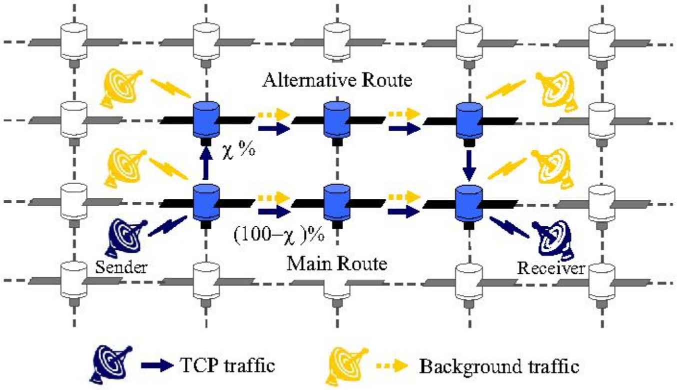


Fig. 2. A simplified simulation topology: a single TCP connection over the most congested area in the constellation (USA region).

While different TCP connections can be simulated on the entire constellation, the behavior of our proposed packet-reordering recovery mechanism is best understood by considering a single TCP connection. We set one TCP connection whose minimal-hop is three over the United States region, the most congested area in the constellation. When the satellite in the middle of the main route gets congested (Fig.2), a portion of the connection flow is forced (by the use of ELB) to change its path and traverse two more additional hops. In the implementation of the proposed packet-reordering recovery mechanism, the time-out interval to send back DupACKs in case of an out-of-order reception of packets is set to ($2L+$ 10ms). This is equal to the propagation delay of two hops, which is the minimum extra delay when a packet is detoured, added to some minimal queuing delays roughly estimated at 10ms. In [5] it is confirmed by simulations that TCP-PR outperforms most packet reordering solutions proposed in recent literature [11]-[13]. Standard TCP and TCP-PR are thus used as comparison terms. In the conducted simulations, parameters of TCP-PR are the same as in [5].

B. Simulation Results and Discussion

Firstly, it should be noted that in the original design of ELB, the setting of the traffic reduction ratio χ is instantly done as a function of the inbound and outbound traffic at a given satellite as described in [4]. To investigate the interaction of the three schemes in case of different values of χ, we plot the achieved goodput of the simulated TCP connection as a function of χ. We consider different satellite constellations by varying the ISL value (i.e., L = 15ms, 20ms, and 25ms).

Fig.3 shows the obtained results. The results demonstrate how the performance of standard TCP gets improved when adding our simple modifications to the receiver terminals. Indeed the proposed packet-reordering recovery mechanism exhibits higher goodput than standard TCP and that is in all the simulated scenarios. The reason beneath this good performance intuitively underlies behind the fact that in the proposed scheme DupACKs are not immediately sent back to the sender upon an out-of-order reception of packets and are rather hold for a time interval.

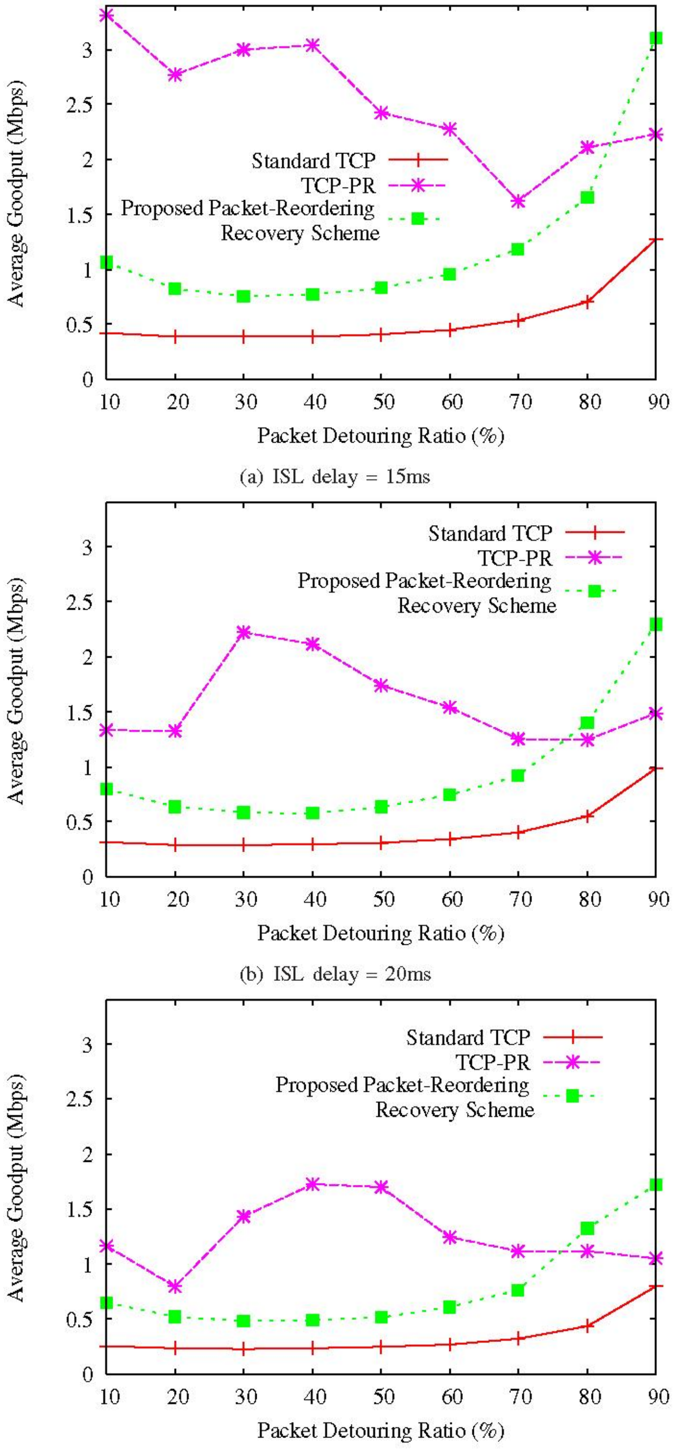
Compared with TCP-PR, the proposed packet reordering recovery mechanism shows much lower goodput in case of low values of χ. However, the performance of TCP-PR degrades as χ gets high values. In the vicinity of (χ=80%), the proposed scheme outperforms the TCP-PR as it achieves higher goodput. The good performance of the proposed scheme becomes more noticeable in constellations with high ISL values. The poor performance of TCP-PR in constellations with large ISL delays and high values of χ is attributable to its contingency on an estimate of the RTT and the bandwidth availability in the setting of its timer. For this reason, when ISL is set to high values, errors take place in the estimation of timers, due in turn to errors in the RTT estimations made before and after the packet detouring operation. Similarly, when χ takes large values, the available bandwidth in the alternative route becomes scarce and errors occur in the setting of timers.

From the observations that 1) today's Internet traffic is characterized by the dominance (more than 80% [17]) of delay-nonsensitive traffic, and that 2) in ELB delay-nonsensitive (e.g., data and non real-time video) packets are first detoured upon an imminent congestion of a satellite, setting χ to values larger than 80% is practical. In this case, the value of the ISL delay, in other words, the constellation type will be the main factor in the decision of which scheme should be used to cope with the packet-reordering issue. Indeed, for MEO systems, the proposed packet reordering scheme is seen more suitable given its simplicity and its good performance in large-ISL constellations. In case of LEO systems with ISL delays smaller than 20ms, TCP-PR can be used only if χ is set to values smaller than 80%. In this case, it should be guaranteed that the good performance of TCP-PR advocates for its complexity and its significant overhead in terms of both computation and memory at the sender side.

V. Conclusion

When ELB is in use, packets of the same flow are transmitted over multiple paths prior to an imminent congestion. While this multi-path routing of ELB has many advantages (e.g., better distribution of traffic, congestion alleviation, and packet drops avoidance), it makes packets of same application experience different latencies resulting in packet reordering.

As a remedy to packet reordering in ELB, we suggest simple modifications to the TCP implementation at TCP receivers. These modifications enable receivers to refer to the TTL field of packet headers to judge the reason beneath the packet reordering. The performance of the proposed packet reordering recovery mechanism is compared to that of Standard TCP and TCP-PR via computer simulations. In addition to its simplicity, simulation results demonstrated the utility of the proposed scheme in environments with high dominance of delay-nonsensitive traffic, a notable characteristic of today's



(a) ISL delay = 15ms

(b) ISL delay = 20ms

(c) ISL delay = 25ms

Fig. 3. Performance evaluation of the three schemes in terms of the achieved goodput for different ISL values.

Internet traffic. The suitability of the proposed scheme is further elucidated in case of constellations with high ISL values.

Finally, it should be noted that while the proposed packet reordering recovery mechanism is investigated particularly in the context of the recently proposed ELB scheme, we do believe that it can also fare in more general multi-path routing schemes. This deserves further study and forms one of our future directions in this particular area of research.

REFERENCES

[1] T. Taleb, N. Kato, and Y. Nemoto, "REFWA: An Efficient and Fair Congestion Control Scheme for LEO Satellite Networks", IEEE/ACM Transactions on Networking Journal, Vol. 14, No. 5, Oct. 2006, pp. 1031-1044.

[2] T. Taleb, N. Kato, and Y. Nemoto, "Recent Trends in IP/NGEO Satellite Communication Systems: Transport, Routing, and Mobility Management", IEEE Wireless Communications Magazine, Vol. 12, No. 5, Oct. 2005, pp. 63-69.

[3] T. Taleb, N. Kato, and Y. Nemoto, "On-Demand Media Streaming to Hybrid Wired/Wireless Networks over Quasi-Geo Stationary Satellite Systems", Elsevier Journal on Computer Networks, Vol. 47, No. 2, Feb. 2005, pp 287-306.

[4] T. Taleb, D. Mashimo, A. Jamalipour, K. Hashimoto, Y. Nemoto, and N. Kato, "ELB: An Explicit Load Balancing Routing Protocol for Multi-Hop NGEO Satellite Constellations", in Proc. IEEE Globecom'06, San Francisco, USA, Nov. 2006.

[5] S. Bohacek, J.P. Hespanha, J. Lee, C. Lim, and K. Obraczka, "A New TCP for Persistent Packet Reordering", IEEE/ACM Transactions on Networking, Vol. 14, No. 2, Apr. 2006, pp. 369-382.

[6] T. Taleb, A. Jamalipour, N. Kato, and Y. Nemoto, "IP Traffic Load Distribution in NGEO Broadband Satellite Networks", in Proc. of 20th International Symposium on Computer & Information Sciences, Istanbul, Turkey, Oct. 2005.

[7] R. Kucukates, and C. Ersoy, "High Performance Routing in a LEO Satellite Network", in Proc. 8th IEEE International Symposium on Computers and Communications, Washington, DC, USA, Jun. 2003.

[8] H. Uzunalioglu, "Probabilistic Routing Protocol for Low Earth Orbit Satellite Networks", in Proc. of IEEE International Conference on Communications 1998. Atlanta, GA, USA, Jun. 1998.

[9] B. Jianjun, L. Xicheng, L. Zexin, and P. Wei, "Compact Explicit Multi-path Routing for LEO Satellite Networks", in Proc. of 2005 IEEE Workshop on High Performance Switching and Routing, Hong Kong, P.R. China, May. 2005.

[10] L. Wood, G. Pavlou, and B. Evans, "Effects on TCP of Routing Strategies in Satellite Constellations", IEEE Communications Magazine, Vol. 39, No. 3, Mar. 2001, pp. 172-181.

[11] E. Blanton and M. Allman, "On Making TCP More Robust to Packet Reordering", ACM SIGCOMM Computer Communications Review, Vol. 32, No. 1, Jan. 2002, pp. 20-30.

[12] F. Wang and Y. Zhang, "Improving TCP Performance over Mobile Ad-hoc Networks with Out-of-order Detection and Response," in Proc. ACM MOBIHOC'02, Lausanne, Switzerland, Jun. 2002.

[13] N. Zhang, B. Karp, S. Floyd, and L. Peterson, "RR-TCP: A Reordering Robust TCP with DSACK", Technical Report TR-02-006, ICSI, Berkeley, CA, July. 2002.

[14] S. Floyd and T. Henderson, "The NewReno Modifications on TCP's Fast Recovery Algorithm," RFC 2582, Apr. 1999.

[15] UCB/LBNL/VINT: Network Simulator - ns (version 2). http://www.isi.edu/nsnam/ns/

[16] M. Mohorcic, M. Werner, A. Svigelj, and G. Kandus, "Adaptive Routing for Packet-Oriented Intersatellite Link Networks: Performance in Various Traffic Scenarios", IEEE Trans. Wireless Commun., Vol. 1, No. 4, Oct. 2002, pp. 808-818.

[17] A.M. Odlyzko, "Internet Traffic Growth: Sources and Implications", in Proc. SPIE, San Diego, California, USA, Aug. 2003.