JWSN 2015, 2, 1-0011

Journal of Wireless Sensor Networks
ISSN: 2001-6417
www.wsn-journal.com

Article

Analysis of TCP Performance over a Low-Delay MAC Protocol Designed for Satellite-based Sensor Networks

A. Y. Tambuwal*, C. O. Iwendi**

* Electronics Research Group, University of Aberdeen, UK

** WSN Consults Ltd, United Kingdom

Correspondence:

A.Y. Tambuwal, Electronics Research Group, University of Aberdeen AB243FX, Aberdeen, UK
Email: ahmed@erg.abdn.ac.uk

Received: 12th June 2015 / Accepted: 20th July 2015 / Published: 27th July 2015

Abstract: Advances in terrestrial network technology such as fibre optic cables have significantly increased data rates and reduced cost, making it highly attractive for high-speed data networks. However, satellite communication remains competitive for certain applications where it has clear advantages over other technologies including fibre optic cables. The point to multipoint broadcast capability of a satellite is an important characteristic that allows multiple sub-networks or nodes to be controlled simultaneously by a single transmission. Similarly, multiple sub-networks or nodes can send data to a central point through a common channel, instead of using multiple point-to-point channels. This facilitates implementation of unique supervisory control and data acquisition systems such as a sensor network to monitor oil and gas pipelines or for agricultural purposes. One important problem in design of a satellite data network is how uncoordinated sources can share the common satellite channel. A multiple access control protocol is required to achieve efficient sharing of the channel while meeting the user traffic constraints. This paper investigates effects TCP performance when used with a new low-delay protocol that integrates Random Access and Bandwidth-on-Demand techniques.

Keywords: Random Access; Bandwidth-on-demand; Startup Latency; Packet Reordering

---

1. Introduction

Many people rely on the Internet for communications every day. It supports a diverse range of applications, each of which generates traffic with distinct characteristics and needs. The performance of these applications depends to a large extent on capacity [1][3], packet loss [1][2], and end-to-end delay [4] characteristics of the underlying network links.

Network capacity is a measure of how much data can be transferred instantaneously, and is an important factor for applications that send large amounts of data i.e. long-lived flows to send large files, emails, film downloads, cloud storage or exchange datasets between applications. In recognition of this fact, major investments have been made to upgrade the core Internet infrastructure resulting in relatively cheap access to capacity, with global average peak access speeds above 18 Mbps [5]. Many complementary methods have also been proposed to increase throughput (i.e. bandwidth utilisation) of high-capacity paths [6][7] and to better utilise links with limited and/or variable bandwidth [8][9]. These are positive steps towards providing better support and higher performance for the majority of Internet traffic, especially during peak traffic periods.

In contrast, short-lived flows that send only a small amount of data at a time constitute a majority of Internet connections [5][10][11]. For this class of application, latency (i.e. time delay to complete a transaction) is the main performance metric. One of the top online-based commercial stores (Amazon.com) recently estimated that every 100ms delay reduces profit by 1% [12].

Marissa Mayer, former Google executive, also said that when the 'Google Maps' home page was reduced in size from 100KB to about 70KB, user traffic increased by 10% in the first week and 25% more in the following three weeks [13]. This improvement was due to faster page load time. Google then reported that increasing the retrieval time of its search page from 400ms (old page with 10 results) to 900ms (new page with 30 results), decreased traffic and ad revenues by 20% [13].

A different set of results shows that end-users are willing to wait no longer than 4-8 seconds for a web page to be displayed [15][16], or 10 seconds for an Internet video streaming session to startup [14][16], before quitting or restarting the connection.

These studies provide motivation for the research summarized in this paper, which seeks to address startup latency of satellite broadband for a growing sector of wireless sensor applications connecting to the Internet via satellite, and other long-haul wireless technologies.

2. DRAWBACK OF SATELLITE BOD MECHANISM

Even as the Internet has become a fundamental commodity for many people around the globe, a significant number that reside in remote (35% in Europe) and developing regions (84% in Africa) still have poor Internet access, or none at all [17]. Getting high-speed (broadband) Internet access to many of these regions using fiber or other cable technologies is too expensive. However due to the importance of the Internet as a means of providing basic healthcare, education, and employment, it is now a common global policy to enable broadband Internet access for everyone. This requires enabling technologies such as long-range wireless or satellite networks. Satellites have special capabilities, which can be used to great advantage in this regard. A satellite can receive signals from, and transmit signals to all satellite terminals in its coverage area, allowing efficient distribution of data to very large number of users [18]. A mechanism is needed to share the transmit radio channel between geographically distributed users. This is called a satellite multiple access control (MAC) protocol [19].

Satellite equipment cost represents start-up cost for both the network operator and the service users. The cost of operation, which represents the dominant long-term cost, is determined by the bandwidth efficiency and complexity of the system [20]. Hence many satellite systems now use sophisticated bandwidth-on-demand (BoD) MAC protocols to optimize bandwidth efficiency. This follows a gradual evolution from early generation satellite systems that used frequency division multiple access (FDMA), random access (RA), and time division multiple access (TDMA) techniques. Unfortunately, BoD can add an extra delay component to the application latency. While this delay may be small for long-lived flows where the satellite terminal is continuously busy, it can be in the order of several seconds when the terminal requesting capacity was previously idle [21][22].

- Initial Access Delay

Initial access delay occurs when a return channel satellite terminal (RCST) is logged-on and has no previous allocations, or following a period with no traffic from an idle terminal (referring to a terminal that is logged-on but exchanging only control/signalling information with the network control centre (NCC)). If new traffic arrives while the RCST is in this state, it has to submit a capacity request (CR) and wait for timeslot allocations from the NCC before packets can be transmitted. This request-allocation cycle is called BoD access delay. Similarly, when traffic is still growing (e.g. following a DNS request or TCP SYN packet), it is difficult to accurately predict the capacity needed for the initial demand in terms of volume, rate and time of allocation (i.e. due to TCP slow start, or variable rate applications). This results in a BoD system introducing initial delay when optimizing for allocation efficiency, or bandwidth over-allocation when optimizing for application performance [22][23].

- Additional Capacity Delay

Additional capacity delay occurs for an active terminal (referring to a terminal that is logged-on and currently transmitting user data traffic) if the traffic arrival rate exceeds network allocation rate. This is normally due to an unpredicted increase in traffic (e.g. due to a scene-change in variable-bit-rate video, or the start of additional traffic flows), or when allocation reduces unexpectedly (e.g. due to rain-fade, movement of a mobile terminal to a location with a different channel condition, or loss of a capacity request). If any of these two events occur, this causes the terminal queues to buildup (or packets are dropped) due to insufficiency of transmission slots. To empty these queues or to prevent packet losses, a new request-allocation cycle is required to update the terminal burst time plan (TBTP).

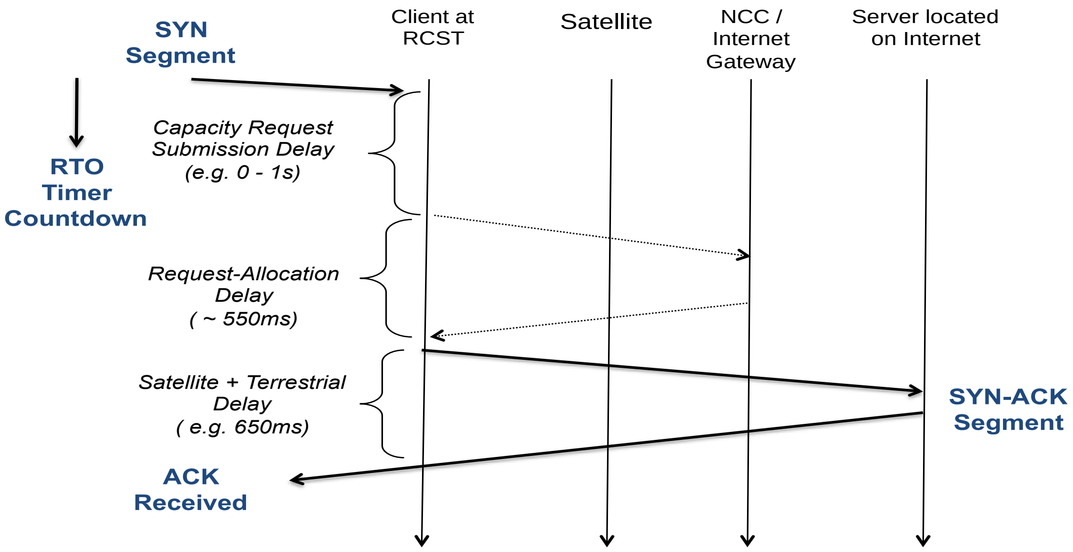

Fig. 2.1 DAMA Access Delay Components

3. PROPOSED RA-DAMA INTEGRATION TECHNIQUE

In [24][25] we proposed a new architecture for combining RA and BOD (herein referred to as DAMA) to reduce startup latency of satellite broadband Internet. Two techniques were recommended for use of the RA channel, referred to as RA Warm Start and RA Top-up.

A. RA Warm Start

RA Warm Start is a technique that avoids initial access delay for a satellite terminal initiating transmission after log-on or resuming transmission after a reasonably long idle period. In this period, DAMA allocations are not available (as there is no previous traffic to trigger a CR) and the terminal would normally wait an entire allocation cycle before receiving transmission slots for a new CR generated for the new traffic. In this state, RA Warm Start uses the RA channel to remove access delay before DAMA allocations are assigned. This reduces the response time for interactive transactions and real-time messaging. Two Warm Start modes are specified: Exclusive RA mode and Switched RA-DAMA mode.

Exclusive RA mode starts a 'RA-eligible flow' on the RA channel and aims to maintain the flow on this channel for all its duration. RA-eligible flows are defined as flows that benefit from fast packet transmission and are able to cope with the loss and low throughput characteristics of the RA channel. Switched RA-DAMA mode starts an RA-eligible flow on the RA channel, but aims to dynamically switch the flow from RA to DAMA channel as quickly as possible. Exclusive RA mode attempts to deliver the packets of a flow exclusively on the RA channel. It is based on the idea that a low-rate of RA channel is sufficient to serve an RA-eligible flow. Hence the aggregate flow is queued in a RA buffer, which does not solicit DAMA allocations (i.e. no CR is sent due to traffic arrivals on this queue).

Apart from removing access delay during startup, Exclusive RA mode helps to improve system efficiency by avoiding DAMA bandwidth over-allocation that occurs with low-rate bursty traffic flows. However, if unused DAMA slots become available before the completion of an RA-eligible flow (e.g. due to some variable rate DAMA flows), Exclusive RA mode does allow transmission of packets on the DAMA channel. This is useful because the DAMA channel is more reliable and removes possibility of further RA channel packet losses, thus improving network response time. Exclusive RA mode depends heavily on RA channel performance and does not always take advantage of DAMA channel. A high PLR could be experienced when demand on the RA channel is high even though DAMA channel capacity could be used.

Switched RA-DAMA mode starts an RA eligible flow on RA channel while also sending a CR for DAMA allocations. Hence the rational of this method is to optimize delay performance at the expense of low allocation efficiency. When DAMA allocations arrive, the flow is dynamically switched from the RA to DAMA channel depending on timeslot availability. Since Switched RA-DAMA mode actively takes advantage of both RA and DAMA transmission opportunities, it is more dynamic than Exclusive RA mode and allows a higher transmission rate.

B. RA Top Up

RA Top-up is a method to address extra capacity requirements of the terminal when real time conversations occasionally send sequence of packets at a speed faster than their average rate (e.g. videoconferencing and variable-bit-rate video). This may cause the terminal to be temporarily short of capacity allocations, requiring a request-allocation delay before obtaining timeslots for excess packets. RA Top-up may take advantage of the RA channel for instantaneous transmission of the excess real time packets to mitigate the impact of jitter, queuing delay, and backlogs at the terminal.

For the considered scenarios, it was shown that the Switched RA-DAMA methods outperform DAMA for a large majority of trials. When the CR period is 10 s, DAMA cannot meet the Quality of Service (QoS) delay targets for interactive transactions. In contrast, Exclusive RA and Switched RA-DAMA response times remain below the target in most of the cases showing considerable performance gains. Also, in this case, the two RA methods exhibit similar behaviour unless CRs are also sent using the RA channel. Switched RA-DAMA mode (with RA CR) achieves highest performance, with response times always below 4 seconds even in presence of high PLR.

4. TCP ANALYSIS OVER AN INTEGRATED RA-DAMA SYSTEM

The protocols that control communication of packets across the Internet are organized into a structure to form a layered design called the TCP/IP protocol stack [26]. The link (or MAC) layer transfers data between adjacent Internet nodes called routers or end-hosts while the transport layer provides end-to-end communication services for user applications. There are two common transport protocols: Transmission Control Protocol (TCP) [27][29] and User Datagram Protocol (UDP) [30]. Most common Internet applications, such as the Web, E-commerce, and E-mail, use services provided by TCP.

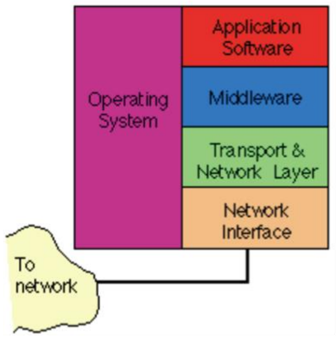

Fig. 4.1 Layering of TCP/IP Internet Protocols at a Host [26]

TCP is implemented at End-hosts for reliable data transfers, and managing Internet congestion over short time periods. When congestion is detected, TCP causes the end-host to reduce its current sending rate by half before data transmission continues. This helps to reduce utilisation and prevent unstable behaviour due to oversubscription of the network over short-time scales.

TCP congestion control has been very successful and remains fundamentally unchanged since it was invented almost 30 years ago [27]. However due to the varying transport needs of new data applications and changes to the underlying network technologies, TCP is continuously being challenged. One fundamental goal is to improve Internet latency without causing congestion problems.

Due to the complex nature of TCP and its extensive use, even a small change can potentially have serious consequences. This section investigates how TCP performance may be affected by the dynamic switching of traffic flows between parallel MAC channels with different characteristics. This potentially creates new transport challenges due to random packet losses, delay variation and packet reordering.

A. Loss of Control Segments During TCP Handshake

If a TCP segment suffers RA collision during the TCP 3WHS leading to loss of a SYN/ACK or the last ACK, the loss would only be detected after the RTO delay at the TCP sender. The SYN packet is retransmitted, but only after the cwnd and ssthresh values have been set to reflect the 'severe' congestion implied by the packet loss.

B. ACK Losses During Data Transfer

After completion of the 3WHS, the server delivers data requested by the client. Standard TCP sends an initial burst of 3 segments and waits for ACKs before further segments are sent. In the case of ACK losses on a return link (e.g. from forwarding some ACK packets over the RA channel using Top Up), there is no requirement for their retransmission. Therefore if specific ACKs are lost while the FlightSize (or cwnd) is small, a TCP sender may wrongly assume that data packets were lost on the forward link. This can cause RTO and a false congestion response as illustrated in Figure 3.

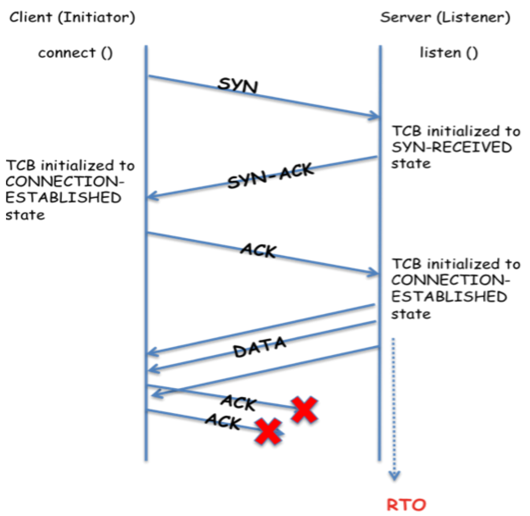

Fig. 4.2 Loss of ACK packets during TCP data transfer phase

A. Data Losses During Data Transfer

TCP flows may be bi-directional, and may exhibit different traffic patterns in the forward and return directions. RA packet loss (even of small control packets) can result in a long-term reduction in the performance of TCP flows that continue to become medium-sized bulk transfers. These flows are constrained by the combination of a limited ssthresh (after the initial loss) and a large round trip path delay, since it can require many more RTTs to complete a transfer as the flow linearly increases the cwnd towards a value that is sufficient to sustain a high transfer rate. However, RA systems may not be optimal for transmission of large data segments and a well-designed RA-DAMA system should avoid using the RA channel to send data segments of medium-sized or long-lived TCP flows.

B. Packet Reordering

Packet Reordering is a phenomenon where the order of packets is inverted in the Internet [31], which is mostly caused due to parallelism in a device or logical link. Integration of RA-DAMA may also cause packet reordering in the following ways:

I. *Link Layer Retransmissions*

Packets lost due to RA collisions could be recovered by link-layer retransmissions [32]. This would hide the packet loss from TCP, hence avoiding the negative consequences including RTO and congestion control. However in this case, the retransmitted packets may be delivered out-of-order with respect to packets belonging to the same traffic flow if the ARQ does not also provide a reordering buffer. This would be bad with a long delay, requiring a method (e.g. F-RTO [33] and the Eifel algorithm [34][36]) or similar stack improvements to recover from the disordered retransmissions.

II. *QoS Traffic Classification and Scheduling*

Traffic classification techniques allow a separate IP queue behaviour to be assigned for different applications or traffic types. DiffServ and IntServ are designed to avoid *significant* re-ordering of packets within a flow. That is, they attempt to preserve per-flow order. The modification of this architecture may alter this behaviour if not properly managed.

III. *Traffic Switching between RA and DAMA Channels*

The switch of traffic (or unbalanced loading) from a RA to DAMA channel could result in reordering of packets at a TCP receiver when the return path implements a queuing method that does not preserve per-flow order. Such methods could be motivated by the desire to use the RA channel for short packets, where it has the best possibility of reducing the queuing delay. Examples of such methods include size-based queuing algorithms [38] (e.g. ACKs-first scheduling and variants of the shortest-packet first algorithm). These methods have been used in early routers [37] to reduce the queuing delay that can result when ACKs are queued behind larger data segments sent over low capacity links. ACK-first based scheduling [3] can produce unusual pathologies when used with TCP, and their impact depends on the traffic pattern, as described in the examples below. In these examples size-based network queuing is considered for a path that comprises a capacity-constrained return link and high-speed forward link using standard queuing:

- ACKs only RL - When the return link carries only ACKs, the forward path does not benefit from these methods. This also has issues when the TCP flow is in part bi-directional and hence some segments carry data. The problem is less when only ACKs are sent since ACK reordering does not trigger issues it simply slows slow-start.
- ACKs and Data RL - When the return link predominantly carries ACKs with occasional data segments and ACKs, the forward link will observe decreased delay. Bi-directional flows may experience slower cwnd growth (disordered ACKs in data segments do not inflate cwnd in modern TCP). Overall there may be benefit for forward transfers, which decreases as the volume of data increases. The return link data performance may not be appreciably impacted, since ACKs are generally much smaller than data. Small packet reordering relative to larger packets is a dangerous approach that can trigger congestion control and loss recovery and may even trigger RTO.
- Data and ACKs RL - When the return link predominantly carries data segments with occasional ACKs, the forward link will benefit. But Shortest-First queuing can result in reordering of bursts when there are variable-sized data segments, which can trigger fast retransmission and reduce performance.
- Data only RL – When the return link carries only data, the forward path does not benefit from these methods, and the return path is not significantly impacted by ACK-First queuing. The impact of Shortest-First queuing depends on the viability of data segments, but is not recommended, since it can result in unpredictable behaviour with specific applications (e.g.

block-oriented data transfers that typically send full-sized segments but periodically send small segments at end of each block).

The use of size-based queuing was common in early packet networks, but is not recommended for use in the general Internet, since it can lead to erratic application performance [3]. Packet reordering causes a number of problems for both TCP and the underlying network [39][40]. These are briefly summarized below:

- When actual packet losses occur, fast retransmit speeds up recovery. But it can have negative effects if packet reordering occurs. Excessive packet reordering beyond the SuPACKThreshold [41] will trigger spurious fast retransmission possibly causing network congestion.
- A spurious fast retransmit is followed by fast recovery with resulting impact on the cwnd and ssthresh. TCP congestion response in case of ACK based recovery is not as severe as RTO based recovery. However if this happens frequently, long-term TCP performance can be affected.
- Packet reordering can cause loss of TCP's ACK clock. This causes bursty transmission and may lead to packet loss. This problem is caused when new segments cannot be sent in response to duplicate ACKs before fast retransmit is triggered. When the ACK finally arrives, all accumulated packets at the transmit buffer will be sent in a burst.
- When a packet is retransmitted due to packet reordering, then measured RTT samples must be discarded due to the ambiguity introduced [40]. The TCP sender may not be able to know which packet triggered the eventual ACK received. This could cause long retransmission delay especially in satellite scenarios.
- TCP Friendly Rate Control (TFRC) [42] is used by real-time applications to compete for bandwidth fairly with TCP flows. TFRC emulates TCP using the throughput equation below to calculate a sending rate, X. This equation means packet loss due to reordering will cause the sending rate to be kept low.

$$X = \frac{s}{R\sqrt{\left(\left(\frac{2bp}{3}\right) + RTO * \sqrt[3]{\left(\frac{bp}{8}\right) * p(1+32p^2)}\right)}}$$

C. Packet Jitter

A system that switches traffic flows from RA to DAMA channel could result in a change of the delay, or introduce variation of delay. This is especially the case for size-based network queuing. These considerations are in fact most important for scenarios where the traffic may be divided between multiple physical layer transmission queues. Sudden changes in delay could adversely impact the TCP RTT measurements, potentially resulting in expiry of the RTO and hence an unwanted congestion response. This is not expected to be a significant effect when using a modern TCP implementation.

The impact of delay variation occurs at the application layer. Most TCP applications, such as web browsing, are tolerant to small (<RTT) delay variations [43]. On the other hand, performance of real-time applications such as VoIP can be affected by delay variation [44][14], resulting in loss at the input to a speech codec, or adversely impacting the round-trip estimator used to scale the play-out buffer.

5. SUMMARY AND CONCLUSION

In the world today, majority of users still access the Internet using terrestrial wired technologies. However there is a growing sector of wireless sensor applications that need to connect to the Internet via satellite, and other long-haul wireless technologies, that cannot be disregarded. Therefore support for short flows needs to be addressed at both TCP and link layers. Updating one without the other is only a partial solution, but together these can provide effective support for all Internet users. This paper analyzed the performance of TCP over an integrated RA-DAMA system. Three main consequences of RA-DAMA integration include startup losses, packet delay variation, and packet reordering. Of these, packet reordering was recognized as the main problem that needs to be addressed in future work.

ACKNOWLEDGEMENTS

I wish to acknowledge the important technical contributions of Prof. Gorry Fairhurst and Dr. Raffaello Secchi to this work.

References

[1] T. Lakshman, and U. Madhow, “The performance of TCP/IP for networks with high bandwidth-delay product and random loss,” *IEEE/ACM Transactions on Networking*, vol. 5, (3), pp. 336-350, 1997.

[2] S. Dawkins, G. Montenegro, M. Kojo, V. Magret, and N. Vaidya, “End-to-End Performance Implications of Links with Errors,” IETF RFC 3155, August 2001.

[3] H. Balakrishnan, V. Padmanabhan, G. Fairhurst, and M. Sooriyabandara, “TCP Performance Implications of Network Asymmetry,” IETF RFC 3449, December 2002.

[4] O3b Networks, “What is Network Latency and Why Does it Matter,” [online]. Available: http://www.o3bnetworks.com/media/40980/white%20paper_latency%20matters.pdf (Accessed: 8 December 2013).

[5] Akamai, “State of the Internet,” [online]. Available: http://www.akamai.com/html/about/press/releases/2013/press_101613.html (Accessed: 14 October 2013).

[6] S. Ha, I. Rhee, and L. Xu, “CUBIC: a new TCP-friendly high-speed TCP variant,” *ACM SIGOPS Operating Systems Review - Research and developments in the Linux kernel*, vol. 42, (5), pp. 64-74.

[7] S. Floyd, “HighSpeed TCP for Large Congestion Windows,” IETF RFC 3649, December 2003.

[8] L. Brakmo, S. O’Malley, and L. Paterson, “TCP Vegas: new techniques for congestion detection and avoidance,” in *Proceedings of the conference on Communications architectures, protocols and applications*, London, UK, 1994, pp. 24-35.

[9] C. Casetti, M. Gerla, S. Mascolo, M. Sansadidi, and R. Wang, “TCP Westwood: end-to-end congestion control for wiredlwireless neworks.” *Wireless Network Journal*, vol. 8. pp. 467-479, 2002.

[10] P. Borgnat, G. Dewaele, K. Fukuda, P. Abry, and K. Cho, "Seven Years and One Day: Sketching the Evolution of Internet Traffic," *INFOCOM 2009, IEEE*, Rio de Janeiro, 2009, pp. 711-719.

[11] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian. "Internet inter-domain traffic." *ACM SIGCOMM Computer Communication Review*, vol. 40, (4) pp. 75-86, 2010.

[12] G. Linden, “Make Data Useful” [online]. Available: http://sites.google.com/site/glinden/Home/StanfordDataMining.2006-11-28.ppt (Accessed 7 December 2013).

[13] Nokia Siemens Network, http://br.nsn.com/file/2103/latencywhitepaperprintversion White paper. Retreieved 7 December 2013.

[14] B. Wang, J. Kurose, P. Shenoy, and D. Towsley, “Multimedia streaming via tcp: an analytic performance study,” in *MULTIMEDIA ’04: Proceedings of the 12th annual ACM international conference on Multimedia*, New York, 2004, pp. 908-915.

[15] Estimating end-to-end performance in IP networks for data applications, ITU-T Rec. G.1030, 2005.

[16] G. Giambene, “QOS requirements for multimedia services,” *in Resource Management in Satellite Networks Optimization and Cross-Layer Design*, New York: Springer, 2007, ch. 3, pp. 67-92.

[17] W. R. Stevens, *TCP/IP Illustrated, Volume 1: The Protocols*, Massachusetts: Addison-Wesley, 1994.

[18] N. Abramson., “Packet switching with satellites,” in *Proceedings of the National Computer Conference*, New York, USA, 1973, vol. 42, pp. 695-702.

[19] SS. Lam, “Satellite Multiaccess Schemes for Data Traffic,” *Conf Rec. International Conference on Communications*, Chicago, USA, 1977, pp. 37.1-19 to 37.1-24.

[20] H. Skinnemoen, R. Leirvik, J. Hetland, H. Fanebust, and V. Paxal, "Interactive IP-network via satellite DVB-RCS," *IEEE Journal on Selected Areas in Communications*, vol. 22, (3), pp. 508-517, 2004.

[21] M. Allman, D. Glover, and L. Sanchez, “Enhancing TCP over Satellite Channels using Standard Mechanisms”, IETF RFC 2488, January 1999.

[22] M. Sooriyabandara, and G. Fairhurst, ” Dynamics of TCP over BoD satellite networks,” *International Journal of Satellite Communications and Networking*, vol. 21, (4-5), pp. 427-449, 2003.

[23] G. Faihurst, A. Sathiaseelan, H. Cruickshank, and C. Baudoin, "Transport challenges facing a next-generation hybrid satellite Internet," *International Journal of Satellite Communications and Networking*, vol. 29, (3), pp. 249-268, 2011.

[24] A. Tambuwal, R. Secchi, and G. Fairhurst, “Exploration of Random Access in DVB-RCS,” *in 12th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting*, Liverpool, UK, 2011.

[25] A. Tambuwal, R. Secchi, and G. Fairhurst, "System Design for Second-Generation DVB-RCS RA-DAMA Integration," *in Proceedings of International Conference on Satellite and Space Communications*, Rome, Italy, 2012.

[26] G. Fairhurst, "Protocol Layers," http://www.erg.abdn.ac.uk/~gorry/eg3567/intro-pages/layer.html Lecture notes. Retrieved 8 December 2013.

[27] V. Cerf, and R. Kahn, "A protocol for packet network intercommunication," *IEEE Transactions on Communications*, vol. 22, (5), pp. 637-648, 1974.

[28] L. Garlick, R. Rom, and J. Postel, "Reliable host-to-host protocols: Problems and techniques", in *Proceedings of the fifth symposium on Data communications*, Snowbird, Utah, 1977, pp. 4.58-4.65.

[29] J. Postel, "Transmission Control Protocol," IETF RFC 793, September, 1981.

[30] J. Postel, "User Datagram protocol", IETF RFC 768, Aug. 1980.

[31] J. Bennett, C. Partridge, and N. Shectman, "Packet Reordering is Not Pathological Network Behavior,' *IEEE/ACM Trans. on Networking*, vol. 7, (6), pp. 789-798, Dec. 1999.

[32] H. Balakrishnan et al., "A Comparison of Mechanisms for Improving TCP Performance Over Wireless Links," in Proceedings of the ACM SIGCOMM '96 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, Stanford, CA, 1996, pp. 256-269.

[33] P. Sarolahti, M. Kojo, K. Yamamoto, and M. Hata, "Forward RTO-Recovery (F-RTO): An Algorithm for Detecting Spurious Retransmission Timeouts with TCP," IETF RFC 5682, September 2009.

[34] G. Fairhurst, and L. Wood, "Advice to link designers to link Automatic Repeat reQuest (ARQ)," IETF RFC 3366, August 2002.

[35] R. Ludwig and R. Katz, "The Eifel algorithm: Making TCP robust against spurious retransmissions," ACM Computer Commununications Review, vol. 30, (21), pp. 30–36, 2000.

[36] R. Ludwig, M. Meyer, "The Eifel Detection Algorithm for TCP," IETF RFC 3522, 2003.

[37] M. Choi, H. Owen, and J. Sokol, "Packet-Size Based Queuing Algorithm for QoS Support," in Proceedings of the First International Conference on Networking, pp. 137-147, 2001.

[38] M. Harchol-Balter, B. Schroeder, N. Bansal, and M. Agrawal, "Size-based Scheduling to Improve Web Performance", *ACM Transaction on Computer Systems*, vol. 21, (2), 2003.

[39] K. Leung, V. Li, and D. Yang, "An Overview of Packet Reordering in Transmission Control Protocol (TCP): Problems, Solutions, and Challenges," *IEEE Trans. on Parallel and Distributed Systems*, vol. 18, no. 4, pp. 522-535, Apr. 2007.

[40] E. Blanton, and M. Allman, "On Making TCP More Robust to Packet Reordering," *ACM SIGCOMM Computer Comm. Rev*, vol. 32, no. 1, pp. 20-30, Jan. 2002.

[41] S. Bhandarkar, A. Reddy, M. Allman, and E. Blanton "Improving the Robustness of TCP to Non-Congestion Events," IETF RFC 4653, August 2006.

[42] S. Floyd, M. Hanley, J. Padhye, and J. Widmer, "TCP Friendly Rate Control (TRFC): Protocol Specification," IETF RFC 5348, Jan. 2003.

[43] M. Fiedler, T. Hossfeld, and P. Tran-Gia, "A Generic Quantitative Relationship between Quality of Experience and Quality of Service," *IEEE Network*, vol. 24, (2), pp. 36–41, Apr. 2010.

[44] N. Markovich, and U. Krieger, "Statistical analysis and modeling of Skype VoIP flows," *Elsevier Computer Communications*, vol. 33, no. 1, pp. S11–S21, Nov. 2010.

[45] B. Wang, J. Kurose, P. Shenoy, and D. Towsley, "Multimedia streaming via tcp: an analytic performance study," in *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, New York, 2004, pp. 908-915