TECHNICAL RESEARCH REPORT

Congestion management in access networks with long propagation delay

by Xiaoming Zhou, John S. Baras

CSHCN TR 2003-23
(ISR TR 2003-46)

The Center for Satellite and Hybrid Communication Networks is a NASA-sponsored Commercial Space Center also supported by the Department of Defense (DOD), industry, the State of Maryland, the University of Maryland and the Institute for Systems Research. This document is a technical report in the CSHCN series originating at the University of Maryland.

Web site http://www.isr.umd.edu/CSHCN/

Congestion management in access networks with long propagation delay

Xiaoming Zhou and John S. Baras
Institute of System Research
Center for Satellite and Hybrid Communication Networks
University of Maryland at College Park, MD, 20742
Email: xmzhou@isr.umd.edu

***Abstract*— Satellite networks are going to play an important role in the global information infrastructure. Satellites can be used to provide Internet services to fixed users and to mobile users. However, recent measurements show that the satellite link efficiency is only about 30%. In order to improve the performance of Internet over satellite, a new protocol called Receiver Window Backpressure Protocol (RWBP) is proposed. RWBP uses per-flow queuing, round robin scheduling and receiver window backpressure for congestion management. Simulation results show that RWBP can maintain high utilization of the satellite link and improve fairness among the competing connections.**

I. INTRODUCTION

Satellites can be used to provide Internet services to fixed users and to mobile users (Figure 1). About ninety five percent of the Internet traffic is TCP traffic. TCP works well in terrestrial networks. However, TCP performance degrades dramatically in satellite networks. Recent measurements show that the satellite link efficiency is only about 30% [1]. There are at least four aspects that cause the low efficiency:

First, the round trip time (RTT) is very large in satellite networks. The round trip time (RTT) in geo-synchronous orbit (GEO) satellite networks is about 500ms. The time taken by TCP slow start to reach the satellite bandwidth (SatBW) is about RTT*log_2(SatBW*RTT) when every TCP segment is acknowledged [16]. For a connection with large RTT, it spends a long time in slow start before reaching the available bandwidth. For short transfers, they could be finished in slow start, which obviously does not use the bandwidth efficiently. Some researchers propose to use a larger initial window [3] up to about 4K bytes rather than one maximum segment size (MSS) for slow start. So files less than 4K bytes can finish their transfers in one RTT rather than 2 or 3 RTTs. Another proposal [10] is to cancel the delayed acknowledgement mechanism in slow start so every packet is acknowledged and the sender can increase its congestion window (CWND) more quickly.

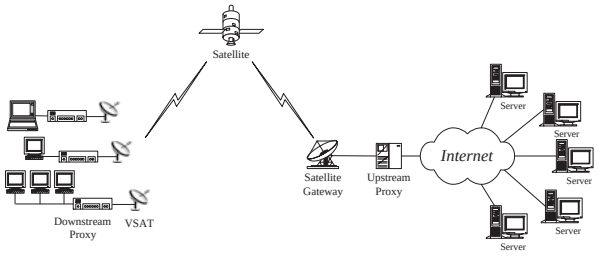

Fig. 1. Direct to user satellite networks (star topology)

Second, Satellite channel is noisier than fiber channel. Bit error rates (BER) of the order of 10^{-6} are often observed [12]. Under the bad weather or when the terminals are mobile, bit error rates can be even higher. Because TCP treats all losses as congestion in the network, this kind of link layer corruption can cause TCP to drop its window to a small size and lead to poor performance.

Third, congestion could happen in the return channel. The forward channel bandwidth from the satellite gateway to the ground terminals is much larger than the return channel bandwidth [4]. Most of the time, the users download data from the Internet and the return channel is used to transfer TCP ACKs. When the return channel traffic load is greater than its bandwidth, congestion could happen. The congestion in return channel may cause poor performance in the forward channel because TCP uses ACKs to clock out data. In the best case, the ACKs are not lost, but queued, waiting for available bandwidth. This has a direct consequence on the retransmission timer and slows down the dynamics of TCP window. To alleviate this problem, ACK filtering [6] is proposed to drop the ACKs in the front of the IP queue by taking advantage of the cumulative acknowledgement strategy in TCP.

Lastly, the satellite link is shared by the TCP traffic and high priority traffic. Usually the satellite network operators provide both Internet services and other services such as multicasting video or audio. Because of the higher revenue brought by the multicating traffic, it is given a higher priority. The leftover bandwidth is used by TCP traffic. To achieve high utilization of the satellite link, TCP traffic has to adapt its sending rate to fill in the leftover bandwidth. This problem has not been addressed in the literature. Usually what is assumed is that the satellite link is used exclusively by TCP traffic.

In addition to the low efficiency, the satellite networks lack an effective fairness control scheme. The fairness control policy of a real satellite network is total usage based [18]. If the total amount of data downloaded by a user exceeds a certain threshold for an extended period of time, the user's throughput could be throttled for several hours. Although this scheme can prevent abusive consumption of bandwidth by a small number of users, the time scale it operates on is too large and it cannot adapt to the traffic arrival pattern on small time scales.

Internet over satellite is more expensive than its terrestrial alternatives. In oder to provide competitive Internet services over satellite, a Receiver Window Backpressure Protocol (RWBP) is proposed to improve the performance in term of both efficiency and fairness in satellite networks.

The rest of this paper is organized as follows. Section II describes the system model of the satellite networks. Section III presents the congestion control, error control, buffer allocation and management in RWBP. Section IV gives the simulation results. Section V relates our work to other proposed schemes for improving TCP performance over satellite links. Finally, Section VI concludes this paper.

II. System Model

TCP connections in satellite networks need large windows to fully utilize the available bandwidth. However it takes much longer for satellite TCP connections than for terrestrial TCP connections to reach the target window size because of the long propagation delay and the slow start algorithm in TCP. And the multiplicative decrease strategy makes the hard gained TCP window very vulnerable to congestion. The misinterpretation of link layer corruption as congestion makes this situation even worse. In the best case, the packet loss does not cause timeout and TCP can stay in congestion avoidance phase rather than in slow start, the additive increase strategy makes the window to grow very slowly. Therefore TCP performance over satellite degrades dramatically.

Because the feedback information of the satellite networks is either delayed too long or too noisy, end-to-end schemes cannot solve these problems very effectively [12][11]. An alternative to end-to-end schemes is to keep the large window of packets in the network such as at the proxies between the satellite and terrestrial networks. Considering the interoperability issue, we adopts the connection splitting based scheme [18][5][7][12] which is currently used in the industry, and we design a new protocol for reliable data transfer over the satellite link.

In the network as shown in Figure 1, an end-to-end TCP connection is split into three connections at the proxies. The first one is set up between the server and the upstream proxy; the second is from upstream proxy to downstream proxy; and the third is from the downstream proxy to the client. Upstream proxy sends premature acknowledgements to the server and takes responsibility to relay all the acknowledged packets to the downstream proxy. Downstream proxy relays the packets to the client the same way as the upstream proxy relays the packets. Normal TCP is used for the server-proxy and proxy-client connections. Receiver Window Backpressure Protocol (RWBP) is designed for the proxy-proxy connection to transfer data over the satellite link. RWBP has newly designed congestion control and error control algorithms, which can achieve high utilization of the satellite link and improve fairness among competing connections.

A. Queuing Model at the Satellite Gateway

The satellite gateway and the very small aperture terminals (VSAT) are connected to the local proxies (Figure 1) through high-speed links whose bandwidth is much larger than the satellite link bandwidth. Therefore between the upstream and downstream proxies, the satellite link is the bottleneck link. The satellite link is used to transfer TCP traffic as well as multicasting video or audio traffic. At the satellite gateway, we assume that a high priority queue is used for multicasting traffic and a low priority queue is used for TCP traffic. These two queues are link layer queues at the terrestrial-satellite output interface (Figure 2).

B. Queuing Model at the Proxies

For a normal router, only those packets waiting for transmission are buffered in the IP output queue. However, the proxies have to buffer the packets waiting for transmission as well as those packets that have been

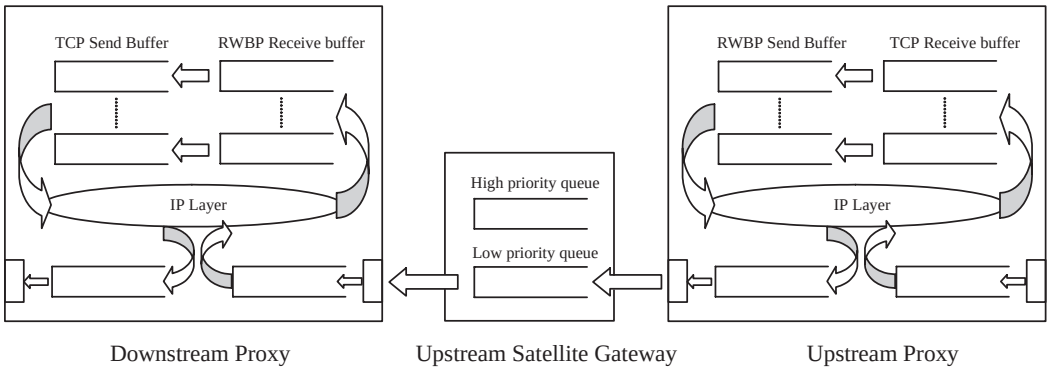

Fig. 2. Queuing model for the satellite gateway and proxies. Flow control is done between downstream proxy and upstream proxy, and between satellite gateway and upstream proxy. It is also done between link layer and the IP layer, between the IP layer and transport layer, and inside transport layer.

transmitted but not acknowledged. A normal router keeps all the packets in a FIFO queue while the proxies have a queue for each connection. From Figure 2, we can see that the input queues at IP layer and link layer should be almost always empty if we assume that the processing rate is not the bottleneck. Therefore the possible queuing points at the proxies are transport layer receive/send buffer, IP output queue and link layer output queue.

III. Receiver Window Backpressure Protocol

Receiver window backpressure protocol (RWBP) is based on TCP with newly designed congestion control and error control algorithms. Although TCP congestion control algorithms can achieve network stability and fairness among TCP connections in terrestrial networks, it is not efficient and effective in satellite networks. Besides the inefficient congestion control algorithms, TCP windowing scheme ties the congestion control and error control together and errors can stop the window from sliding until they are recovered. The above observations motivate us to decouple error control from congestion control in TCP first and then design more efficient and effective congestion and error control schemes with the specific characteristics of satellite networks in mind. Our design goals of RWBP are to: 1) achieve high utilization of the satellite link; 2) improve fairness among competing connections; 3) reduce response time of short transfers such as web browsing; 4) reduce the return channel bandwidth requirement without the degradation of forward channel performance.

A. Congestion Control in RWBP

TCP uses slow start to probe the bandwidth at the beginning of a connection and uses additive increase and multiplicative decrease (AIMD) congestion avoidance to converge to fairness in a distributed manner. RWBP is based on TCP; however RWBP cancels all the congestion control algorithms in TCP and uses per-flow queuing, round robin scheduling [8] and receiver window backpressure for congestion control (Figure 2).

Flow control is done between the downstream proxy and the upstream proxy at the transport layer by using the receiver window (Figure 2). For each RWBP connection, the downstream proxy advertises a receiver window based on the available buffer space for that connection just as in TCP. RWBP does not use Window scaling to advertise large windows to upstream proxy because large window scale factor can produce inaccurate values. In RWBP, the 16-bit receiver window field is still used but its unit is maximum segment size rather than byte. Similarly flow control is also done between the satellite gateway and the upstream proxy at the link layer (Figure 2). The low priority queue at the satellite gateway advertises a receiver window to the upstream proxy so that the low priority queue will not overflow.

In addition, flow control is done between the transport layer and the IP layer, and between the IP and the link layer. At the upstream proxy, a round-robin scheduler can send a packet for a RWBP connection only if the available advertised receiver window is greater than one and there is at least one packet buffer space available at the IP output queue. When there is no packets can be sent or the available advertised receiver window is zero, the scheduler goes on to serve the next connection. When the IP layer output queue sends packets to the link layer, it has to make sure that the link layer queue is not going to be overflowed. This allows the link layer congestion backpressure to propagate to IP layer and then to trans-

port layer. Inside the transport layer, when packets are moved from upstream connection receive buffer to the downstream send buffer, a blocking write is performed so that the send buffer will not overflow. This way the congestion is back pressured to the receive buffer of the upstream connection and a smaller receive window is going to be sent to the source. Finally the congestion is back pressured to the source. When the traffic load decreases, the buffers begin to be emptied faster and larger advertised receiver windows are sent to the sources so the sources can speed up. If some connections are bottlenecked upstream or are idle because the application layers do not have data to send, the scheduler can send packets from other connections and high satellite link efficiency can be achieved. The round-robin scheduler does not take into account the packet sizes. Connections with larger packet sizes can get more bandwidth than those with smaller packet sizes. This problem can be solved by a more sophisticated scheduler and is left as future work.

The above flow control scheme can guarantee that there is no buffer overflow in the downstream proxy queues, in the upstream proxy queues or in the low priority queue at the satellite gateway. Therefore if there is a packet loss at downstream proxy, it must be due to satellite link corruption rather than due to buffer overflow. Therefore RWBP decouples error control from congestion control.

B. Error Control in RWBP

TCP depends on duplicate acknowledgements and timer for error control. Because out of order packet arrivals are possible in the wide area networks, the fast retransmit algorithm is triggered after three rather than one or two duplicate acknowledgements are received. The high bit error rate of the satellite link can cause multiple packet losses in one window and may lead to timeout. Furthermore the loss probability over the satellite link is determined by the bit error rate and packet size, so the retransmission packets can be corrupted as probable as original packets when the error rate is high [17]. When the retransmitted packets are lost, timer could be the only means for error recovery in TCP. However, the timeout value is usually set much larger than the round trip delay to make sure the original packet does leave the networks to avoid false retransmission. The conservative loss detection and recovery schemes in TCP are not effective in satellite networks.

In RWBP, we explore the specific characteristics of the satellite networks. Firstly, because RWBP congestion control can eliminate packet losses due to buffer overflow, any loss must be caused by the link layer corruption. So the error control scheme can operate independently with the congestion control scheme. Secondly, the satellite link is a FIFO channel and out of order packet arrivals are impossible. RWBP error control algorithms explore the in order packet delivery characteristic for error detection and use selective acknowledgement (SACK) for error recovery.

All data packets including retransmission packets of a RWBP connection are sorted in their transmission order. RWBP sender keeps track of the right edge packet in sequence space of all acknowledged packets, i.e. cumulatively or selectively acknowledged packets. Whenever an acknowledgment packet is received, RWBP sender compares in sequence space the right edge packet acknowledged in the current ACK with that in the previous ACK. If the sequence number does not advance, RWBP error control algorithm does nothing. Whenever the sequence number advances, RWBP error recovery scheme is triggered. The first match of the current right edge packet in the sorted list must have arrived at the RWBP receiver. If a packet before the right edge packet in the sorted list is neither cumulatively acknowledged nor selectively acknowledged, RWBP assumes that the packet is lost and retransmits it. From the following example and the simulation results in section IV, we will see RWBP can recover not only the first time losses but also the retransmission losses very effectively. Timer is still used as the last resort for loss recovery. After timer expires, two copies of the lost packet are sent to increase redundancy.

Figure 3 gives an example of RWBP error control scheme. On the left of Figure 3, in sequence packets are not drawn and only out of order packets are drawn. Initially the right edge packet is set to packet 0. Packet 1, 2, ..., 6 are sent to the RWBP receiver. Packet 2 and packet 4 are corrupted. The receiver acknowledges packet 1, 3, 5 and 6. The sender compares the right edge packet 6 in the ACK to the initial right edge packet 0. The right edge packet advances in sequence space. And the sender checks the sorted list and finds out packet 1, 2, 3, 4 and 5 are sent before packet 6. Only packet 1, 3 and 5 are acknowledged, packet 2 and 4 should be lost. Packet 2 and packet 4 are retransmitted before new packet 7, 8, 9 and 10. Packet 2 arrives at the receiver successfully; however packet 4 is corrupted again. The receiver cumulatively acknowledges up to packet 3 and selectively acknowledges packet 7 to packet 10. The right edge packet advances in the sequence space from packet

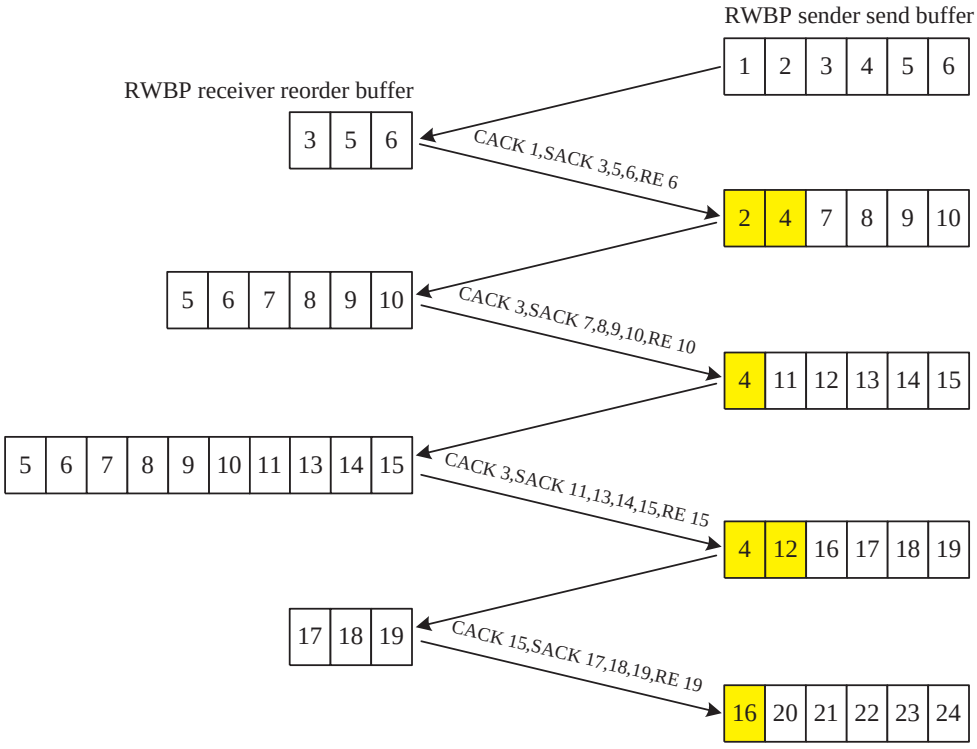

Fig. 3. Error control in RWBP. CACK : Cumulatively acknowledged; SACK : Selectively acknowledged; RE : Right edge

6 to packet 10. The sender checks the sorted list and finds out packet 4 is transmitted before packet 10. However packet 4 has not been acknowledged, therefore packet 4 should be lost again. Packet 4 is retransmitted again. The above procedure keeps on going until the sender finishes its transfer. It is not difficult to see from this example that RWBP can not only recover effectively the first time losses such as packet 2 and packet 12, but also the higher order packet losses such as packet 4, which has been corrupted three times.

When a packet is received correctly by the RWBP receiver but all the acknowledgments for it are lost, RWBP could retransmit this packet unnecessarily. In RWBP, one acknowledgement can carry up to four SACK blocks. As long as the acknowledgements are not sent very infrequently, this event should be rare.

C. Buffer Allocation and Management in RWBP

The buffer sizes allocated to each connection at the upstream and downstream proxies have a direct impact on the end-to-end throughput. Firstly, assume that there is only one end-to-end transfer in the system. The satellite link is error free and its bandwidth is SatBW. At the upstream proxy, the size of the TCP receive buffer is RecvBuf and the size of the RWBP send buffer is SndBuf. The round trip time of the satellite RWBP connection is SatRTT and the round trip time of the terrestrial TCP connection is TerrRTT. Then the maximum achievable throughput of the end-to-end transfer is MIN (SatBW, SndBuf/SatRTT, RecvBuf/TerrRTT). From this formula, we can see that SndBuf or RecvBuf can become the bottleneck of the end-to-end performance if it is less than the bandwidth delay product of its corresponding connection. However if the buffer size is greater than the bandwidth delay product, the satellite link becomes the bottleneck and packets could be backlogged at the proxy. The same analysis applies to the downstream proxy.

When there are multiple connections in the system, the bandwidth available to each connection depends on the number of connections and their activities. One possible buffer allocation scheme is adaptive buffer sharing [8] which dynamically allocates a buffer pool to all the connections based on their bandwidth usage. While this scheme can dramatically decrease the buffer requirement, it is complex to be implemented. In RWBP, each connection is assigned a static peak rate and the buffer size is set to the peak rate delay product (PRDP).

When the satellite link is error free, the buffer sizes allocated above are enough to achieve the target peak rate. However when the satellite link is not error free,

changes need to be made at both the downstream and the upstream proxies.

When a packet is corrupted, the downstream proxy has to buffer the out of order packets because RWBP receiver only forwards in sequence packets. For example, in Figure 3 the in sequence packet 1 is forwarded while the out of order packets 3, 5, and 6 are kept in the receive buffer. In order to keep the advertised receiver window open so that the RWBP sender can send new packets during the error recovery, the downstream proxy needs a buffer size larger than the peak rate delay product to achieve the peak rate. If the error rate of the satellite link is low and corrupted packets can be recovered in one RTT, receive buffer size about two times of the peak rate delay product should be provided. If the error rate is relatively high, retransmissions packets can be corrupted. Our simulation results in section IV show that receive buffer size should be about three to four times of the peak rate delay product to maintain high satellite link utilization.

For the upstream proxy, the buffer management in RWBP is different from that in TCP SACK [14]. In TCP SACK, only packets cumulatively acknowledged are released from the send buffer. Packets selectively acknowledged are still kept in the send buffer because the TCP receiver may renege and discard the SACKed packets when it runs out of receive buffer. For example, in Figure 3 cumulatively acknowledged packet 1 is released from the send buffer, however selectively acknowledged packet 3, 5 and 6 are still kept in the TCP send buffer. The buffer occupied by these SACKed packets can cause the upstream proxy to advertise a smaller window to the source. This will slow down or even stall the source. After the error is recovered, the cumulative acknowledgement may clear a large number of packets from the proxy send buffer. The upstream proxy could run out of packets to send and it has to wait for new packets to arrive from the source. Therefore the terrestrial TCP connection could cause starvation of the upstream proxy queue. In RWBP, the receiver never reneges and the sender does not clear the SACK state information after timeout. So the SACKed packets can be released from the send buffer. Thus only those packets actually corrupted over the satellite link are still kept in the buffer. For example, in Figure 3, successfully received packet 3, 5 and 6 are released from the buffer and only the lost packet 2 and 4 are still buffered.

IV. Performance Evaluation

In this section, we evaluate the performance of RWBP with OPNET. The metrics we are interested in are return channel bandwidth requirement, end-to-end throughput and fairness for bulk transfers, response time for short transfers and satellite link utilization when there is high priority traffic as well as TCP traffic.

A. Return Channel Bandwidth and Proxy Buffer Size Requirement

In order to achieve high throughput in the forward channel, we need to find out the return channel bandwidth and proxy buffer size requirements. If the requirements are not satisfied, they could become the system bottlenecks.

In Figure 1, a single transfer in the system is set up between a client and an Internet server. The satellite link bandwidth is 600kbps. Server-proxy and proxy-client link bandwidth is 2Mbps. The RTT of the proxy-proxy connection is 500ms, the RTT of the server-proxy connection is 80ms and the RTT of the proxy-client is 10^{-4} ms. The packet size is 512bytes and the file size is 3M bytes. The peak rate is set to the satellite link bandwidth. To get the downstream proxy receive buffer size requirement, it is set to infinity and all the other proxy buffer sizes are set to the peak rate delay product.

In RWBP, an acknowledgement is sent when every N data packets are received. By changing N, we can change the acknowledgement frequency. Figure 4 shows that when N increases exponentially i.e. the ACK frequency decreases exponentially, the return channel bandwidth usage decreases exponentially. However the forward channel throughput is very insensitive to the return channel usage (Figure 5). Only when N increases up to 16, the forward channel throughput begins to decrease. This happens because of the following reason. Although ACKs are not used to clock out data packets in RWBP, they are still used to clear upstream buffers. Less frequent ACKs can cause the buffers to be filled up and to slow down the terrestrial connections. When N equals 8, the forward channel throughput is very close to that achieved when N equals 1. Therefore we set N to 8 in RWBP. In TCP, one ACK is sent every two data packets are received. So RWBP can reduce the return channel bandwidth requirement to one quarter of that in TCP without sacrificing the forward channel throughput.

Figure 6 shows the reorder buffer sizes at the downstream proxy for bit error rate equals 10^{-6} and 10^{-5}. For both cases, one acknowledgement is sent when every eight data packets are received. For BER equals 10^{-6},

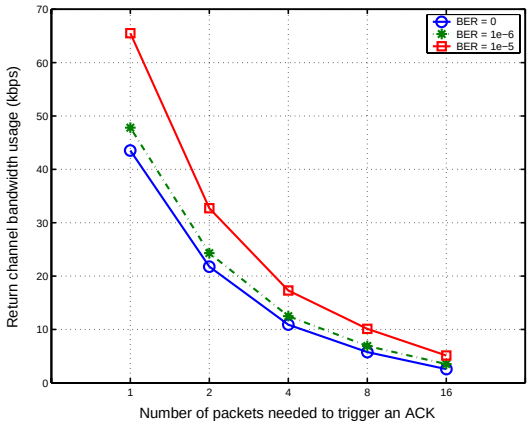

Fig. 4. Return channel usages for different acknowledgement frequency in RWBP

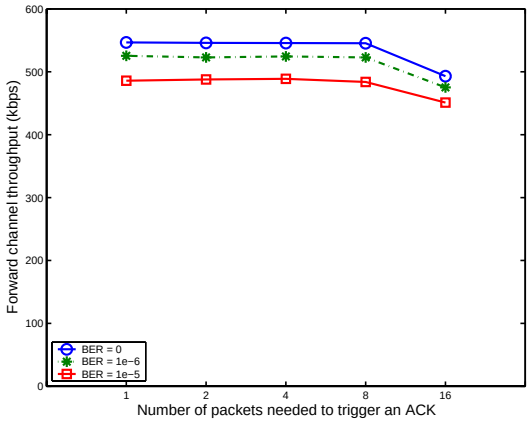

Fig. 5. Forward channel throughputs for different acknowledgement frequency in RWBP

occasionally there is about one peak rate delay product (PRDP) of packets in the reorder buffer (upper plot in Figure 6). This means that the errors can be recovered in one RTT. However when the error rate is increased to 10^{-5}, retransmissions can be lost too. The lower plot in Figure 6 shows that retransmissions could be lost twice because sometimes there are about three PRDP of packets in the reorder buffer. Therefore in order to achieve high forward channel throughput, downstream proxy receive buffer size larger than PRDP is needed so that the advertised window can remain open and the upstream proxy can continue to send new packets during error recovery. For low bit error rate, buffer size about two times of the PRDP is needed. While for high bit error rate, buffer size about four times of PRDP is needed (Figure 6).

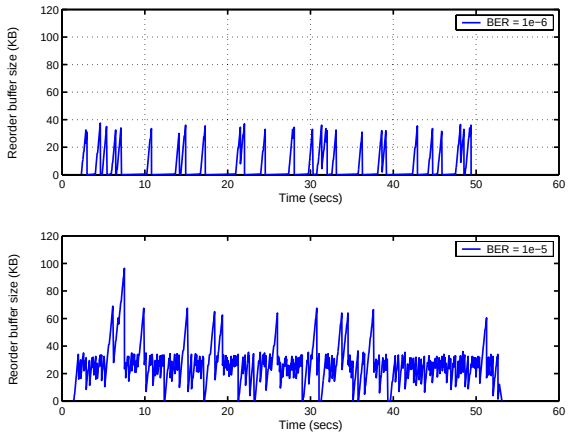

Fig. 6. Reorder buffer size for BER = 10^{-6} and BER = 10^{-5}

B. End-to-end Throughput and Fairness for Bulk Transfers

In Figure 1, fifteen clients download large files from fifteen Internet servers. The satellite bandwidth is 9Mbps. The link bandwidth from each server to the upstream proxy is 2Mbps. The link bandwidth from downstream proxy to each client is also 2Mbps. The RTTs for all the fifteen proxy-proxy connections are 500ms. The RTTs of the fifteen proxy-client connections are all set to 10^{-4}ms. The RTT of the server-proxy connection corresponding to end-to-end transfer i is (10*i-8)ms. Therefore the end-to-end round trip time for transfer i is 500.1 +(10*i-8)ms, i.e. in the range [502.1ms, 642.1ms]. The peak rate is set to 1.2Mbps. The downstream proxy receive buffer size is set to two times peak rate delay product(PRDP) and all the other proxy buffer sizes are set to one PRDP. The satellite gateway buffer size is set to the satellite bandwidth delay product.

1) End-to-end throughput: In Figure 7, we compare the end-to-end aggregate throughput of RWBP and TCP SACK for different bit error rates when they are used for the proxy-proxy connections. All the terrestrial connections use TCP Reno. The simulation time is 1000 seconds.

When the bit error rate is very low, both schemes can achieve very high throughput. For TCP SACK when the bit error rate increases up to 10^{-6}, the link layer corruption causes the upstream proxy TCP to drop its congestion window and leads to degraded performance. When the loss rate is increased further to 10^{-5}, the retransmitted packets can get lost again and TCP SACK may have to wait for the timeout to recover the losses. After timeout, the congestion window is set to one

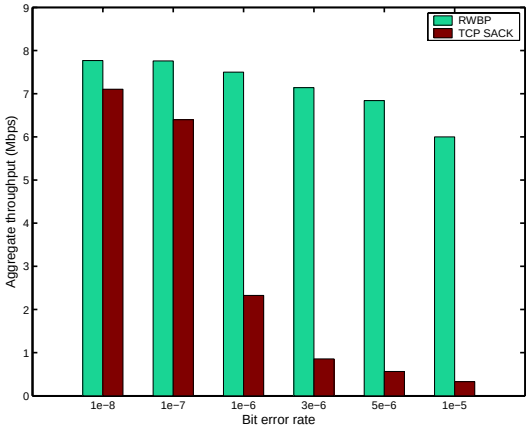

Fig. 7. Aggregate throughput for different bit error rates in RWBP and TCP

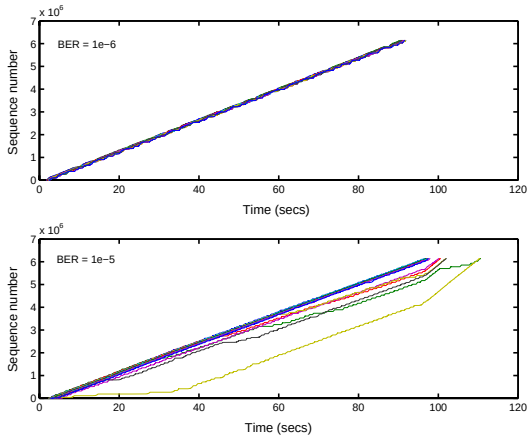

Fig. 8. Received sequence number progress at the fifteen clients for BER = 10^{-6} and BER = 10^{-5} in RWBP

and TCP enters slow start. Therefore the satellite link utilization is very low for high loss rate when TCP is used. In RWBP, the congestion control is decoupled from the error control. Because the upstream proxy can schedule new packets to be sent during error recovery and RWBP error control can recover effectively first time as well as higher order losses, RWBP can achieve higher throughput for both low and high bit error rates (Figure 7).

2) Fairness: We use the same parameters as in section B.1. A 6M bytes file is downloaded from Internet server i to client i. Figure 8 plots the received packet sequence number growth at the clients for the fifteen transfers. For BER equals 10^{-6}, the upper plot in figure 8 shows that the sequence numbers of the fifteen transfers grow almost at the same rate because they are overlapping with each other. Therefore data packets arrive at the clients almost at the same rate and each transfer gets a fair share of the satellite link bandwidth. When BER increases to 10^{-5}, the sequence numbers still grow at close rates. For the lowest curve in figure 8, at the beginning of this transfer, the sequence number grows at a much lower rate than those of other transfers. The reason is that its errors cannot be recovered by RWBP error control algorithm and the upstream proxy has to wait for the timer to expire. After about 35 seconds, this transfer recovers and its packets arriving rate becomes close to those of other transfers.

C. Response Time for Short Transfers

In addition to bulk file transfers, another popular application is web browsing, which is characterized by the clients send small requests and the servers reply with small files. The same network configuration is used as

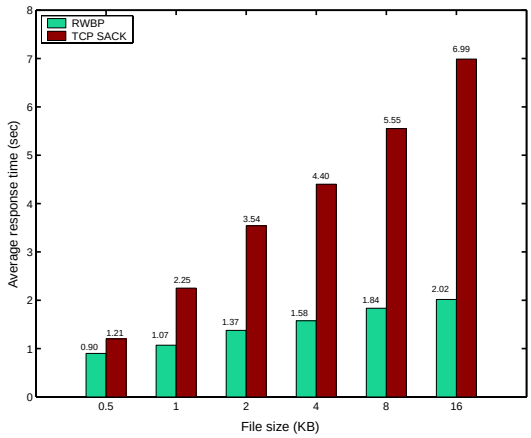

Fig. 9. Response time for short transfers in RWBP and TCP

in section B.1 and the BER is 10^{-6}. All the transfers between servers and clients are still bulk transfers except that the bulk transfer between server five and client five is replaced by short file transfers. Client five randomly requests small files of fixed size from Internet server five. Figure 9 shows the average response time for different file sizes. The average response time is calculated over 1000 samples. RWBP performs better than TCP SACK for the following reasons. Firstly, RWBP does not need to go through the slow start phase and packets can be sent as long as the link is available. Secondly, because RWBP provides per-flow queuing, packets of short transfers do not need to wait after packets of bulk transfers in the FIFO queue at satellite gateway. Therefore, RWBP isolates short transfers from bulk transfers and decreases the queuing delay of short transfers.

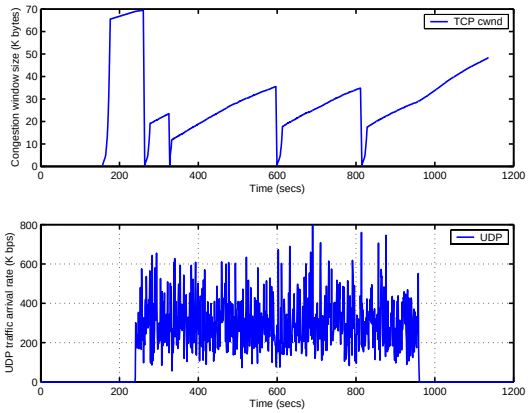

Fig. 10. TCP congestion window size at upstream proxy and the arrival rate of UDP traffic

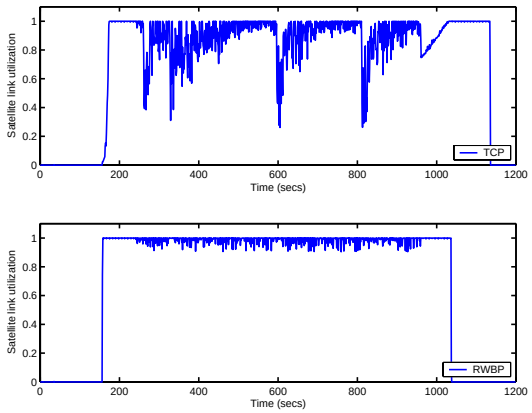

Fig. 11. Satellite link utilization for TCP and RWP with high priority UDP traffic

D. Satellite Link Utilization with Competing High Priority Traffic

We use the same network set up as in section B.1. However only the transfer between server five and client five is activated. The satellite link bandwidth is set to 600kbps. The peak rate is set to 600kbps. The downstream proxy receive buffer size is set to two times PRDP and all the other proxy buffer sizes are set to one PRDP. The satellite gateway buffer size for low priority TCP traffic is 75 packets, which is about the satellite bandwidth delay product and the buffer size for high priority UDP traffic is 50 packets. The bit error rate of the satellite link is set to zero. We compare the satellite link utilization when RWBP and TCP SACK are used for the proxy-proxy connection.

TCP transfer starts at the 150th second and a file of 36 M bytes is sent from server five to client five. UDP traffic begins at the 240th second and ends at the 960th second. Firstly when only TCP traffic is active, the upper plot in figure 10 shows that TCP can increase its congestion window large enough so that the satellite link bandwidth is fully utilized (upper plot in figure 11). However after a high priority UDP flow with average arrival rate 300kbps (lower plot in figure 10) begins, its dynamic changed traffic demand causes low priority TCP periodically timeout. After timeout, TCP goes into the slow start phase. Because of the long propagation delay of the satellite link, it takes a long time for TCP to increase its window large enough to fully utilize the satellite bandwidth. The upper plot in figure 11 shows that the satellite link utilization is low when there is competing UDP traffic with TCP traffic. However, RWBP can adapt to the high priority traffic load very well and the satellite link utilization is kept very high as shown in the lower plot of figure 11. Because link layer flow control is used between the satellite gateway and the upstream proxy, the congestion at the satellite gateway caused by the increase demand of high priority traffic is back pressured to the upstream proxy by advertising a smaller window. Because of this, the increasing rate of the high priority UDP traffic will not cause RWBP packets dropped at the satellite gateway. When the traffic demand of the UDP traffic decreases, a larger window is advertised to the upstream proxy so that RWBP can speed up to fill in the left bandwidth.

V. Related Work

TCP Peach [2] is an end-to-end scheme and it has two new algorithms sudden start and rapid recovery, which replace the slow start and fast recovery algorithms in TCP Reno respectively. Essentially TCP Peach has two logical channels, one is for the data transmission and another one is for bandwidth probing. TCP Peach uses low priority dummy segments to probe the bandwidth in sudden start and rapid recovery. One problem with TCP Peach is that dummy segments do not carry any information and they are overhead to the data. Another problem is that all the routers need to implement some kind of priority mechanism, which makes it difficult to deploy.

Satellite transport protocol (STP) [12] adapts an ATM-based protocol for use as a transport protocol in satellite data networks. STP uses a modified version of TCP slow start and congestion avoidance algorithms for its congestion control. STP can get comparable performance to TCP SACK in the forward channel with significantly less bandwidth requirement in the return channel. The transmitter sends POLL packets periodically to the

receiver, the receiver sends STAT packet as acknowledgements. The return channel bandwidth requirement depends mainly on the polling period, not on the forward channel data transmission rate. Therefore the bandwidth demand for the return channel decreases dramatically.

Space communication protocol standards-transport protocol (SCPS-TP) [9] is a set of TCP extensions for space communications. This protocol adopts the timestamps and window scaling options in RFC1323 [13]. It also uses TCP Vegas low-loss congestion avoidance mechanism. SCPS-TP receiver doesn't acknowledge every data packet. Acknowledgements are sent periodically based on the RTT. The traffic demand for the return channel is much lower than that in TCP. However it is difficult to determine the appropriate acknowledgement rate. SCPS-TP does not use acknowledgements to clock out the data and it uses an open-loop rate control mechanism to meter out data smoothly. SCPS-TP uses selective negative acknowledgement (SNACK) for error recovery. SNACK is a negative acknowledgement and it can specify a large number of holes in a bit-efficient manner.

VI. Conclusion

In order to improve the efficiency of satellite link and provide effective fairness control, a new protocol RWBP is proposed. RWBP is designed for the satellite connections by taking advantage of the specific characteristics of the satellite network. It uses per-flow queuing, round robin scheduling and receiver window backpressure for congestion management. The congestion management algorithms in RWBP can eliminate buffer overflows inside the satellite network even when there is high priority traffic competing with TCP traffic. Therefore any loss inside the satellite network must be caused by link layer corruption rather than by buffer overflows. So the error control in RWBP can operate independently with its congestion management. The newly designed error control scheme in RWBP can effectively recover not only first time losses but also higher order losses. Our results show that RWBP can improve the performance of both bulk and short transfers over the GEO satellites.

The error recovery in RWBP is ARQ based. It is interesting to investigate how an adaptive forward error correction (FEC) scheme interacts with the ARQ scheme in RWBP. The return channel bandwidth is managed by a multiple access control (MAC) protocol. It is also interesting to investigate the different MAC protocols and their effect on RWBP or TCP performance.

References

[1] N. Abramson. Web Access: Past, Present and Future. In *IEEE GLOBECOM'02 key note speech*, December 2002.

[2] I.F. Akyildiz, G. Morabito, and S. Palazzo. TCP Peach: A new congestion control scheme for satellite IP networks. *IEEE/ACM Trans. on Networking*, 9(3), June 2001.

[3] M. Allman, S. Floyd, and C. Partridge. Increasing TCP's initial window. *RFC 2414*, September 1998.

[4] V. Arora, N. Suphasindhu, J.S. Baras, and D. Dillon. Asymmetric Internet Access over Satellite-Terrestrial Networks. Technical Report 96-10, UMD CSHCN, July 1996.

[5] A. Bakre and B. R. Badrinath. Implementation and performance evaluation of indirect TCP. *IEEE Transactions on Computers*, 46(3), March 1997.

[6] H. Balakrishnan, V. Padmanabhan, and R. Katz. The effects of asymmetry on TCP performance. In *3rd ACM/IEEE MobiCom Conf.*, pages 77–89, September 1997.

[7] K. Brown and S. Singh. M-TCP: TCP for mobile cellular networks. *ACM Comput. Commun. Rev.*, 27:19–43, October 1997.

[8] Kongling Chang. IP layer per-flow queuing and credit flow control. Technical report, Ph.D. Thesis Harvard University, Division of engineering and applied science, January 1998.

[9] R. C. Durst, G. Miller, and E. J. Travis. TCP extensions for space communications. In *ACM MobiCom Conf.*, November 1996.

[10] M. Allman et al. Ongoing TCP research related to satellites. *RFC 2760*, February 2000.

[11] T. Henderson, E. Sahouria, S. McCanne, and R. Katz. On improving the fairness of TCP congestion avoidance. In *Proc. IEEE GLOBECOM'98*, 1998.

[12] T. R. Henderson and R. H. Katz. Transport protocols for Internet-compatible satellite networks. *IEEE J. Select. Areas Comm.*, 17:326–344, February 1999.

[13] V. Jacobson, R. Braden, and D. Borman. TCP extensions for high performance. *RFC 1323*, 1992.

[14] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. TCP selective acknowledgment options. *RFC 2018*, 1996.

[15] J. Padhye, V. Firoiu, D.F. Towsley, and J.F. Kurose. Modeling TCP Reno performance: a simple model and its empirical validation. *IEEE/ACM Trans. on Networking*, April 2000.

[16] C. Partridge and T. Shepard. TCP performance over satellitelinks. *IEEE Network*, 11:44–49, September 1997.

[17] N. Samaraweera and G Fairhurst. Reinforcement of TCP/IP Error Recovery for Wireless Communications. *Computer Communications Review (CCR)*, 28(2):30–38, February 1999.

[18] Hughes Network System. DirecPC web page. In *http://www.direcpc.com*, December 2002.