Adaptive Multimedia Communication over Satellite Routed IP

Michael Welzl
Telecooperation Department
University of Linz, Austria
michael@tk.uni-linz.ac.at

Abstract

Since the IETF "TCPSAT" Working Group was chartered in 1997, numerous issues with TCP over satellite have been addressed. Although the difficulties encountered are the same, it seems as though adaptive multimedia communication over satellite has been neglected. We identify the common problems, discuss their impact and introduce a method to deal with some of them — a protocol that queries routers for QoS related information.

1 Introduction

The QoS perceived by users who watch streaming video or talk to each other over the Internet mainly depends on the service interruptions they experience. On the Internet, "interruption" means dropped or severely delayed packets — something that can be avoided via resource reservation and reduced via flow control mechanisms. Since the typical Internet user has no means to demand special treatment from lower layers, Internet multimedia applications need to adapt to the network's condition. Although TCP provides us with well designed flow control, it is not the protocol of choice for streaming multimedia, where fast is more important than reliable transport and compression mechanisms limit the applications' capability of varying the data rate. Programmers of adaptive multimedia applications need to implement their own flow control based on UDP.

Since UDP provides the same unreliable, packet oriented service as IP, these applications' perspective on how satellite links affect throughput is just the same as TCP's:

- GEOs [1], satellites in geostationary orbit, introduce a delay between 239.6 and 279.0 ms [2]. If the network is symmetrical, which means that the path between two nodes does not depend on a packet's direction, a delay of approximately half a second is imposed on any end-to-end feedback mechanism.
- A satellite link may exhibit high bit-error rates. In some cases, this problem may already be solved at lower layers; we will assume that it is not. Since practically all flow control mechanisms on the Internet interpret packet loss as a sign of congestion, this can pose a real problem.
- Satellites have a large bandwidth * delay product. It defines the amount of data a protocol should have "in flight" (transmitted, but not yet acknowledged) to fully utilize the link [2].
- More and more Internet providers offer a satellite downlink, but the end user's outgoing traffic is still sent over a slow terrestrial link (or a satellite uplink with reduced capacity). This highly asymmetric usage brings about a problem called "ACK starvation" or "ACK congestion", in which the sender can not fill the satellite channel in a timely fashion due to slow acknowledgments on the return path [6].

There are a number of other relevant factors such as variable round trip times, intermittent connectivity in non-GEO configurations and a satellite's natural broadcasting capability which will not be discussed here.

2 Common solutions

Commonly used or recommended solutions on top of IP generally fall into four categories:

2.1 Deal with the large bandwidth * delay product

Most of the solutions in this category have to do with TCP specifics as older implementations show some faults when it comes to links with a large bandwidth * delay product: The standard congestion window size is too small, slow-start as well as congestion avoidance will take very long due to the long RTT. Some proposed changes are a *larger initial window size*, *window scaling* and the *Selective ACKnowledgment (SACK)* extension which allows a TCP sender to properly adapt if multiple packets are lost per RTT [6] [5]. Since increasing TCP's congestion window is segment based rather than byte based, using *Path MTU Discovery*, a mechanism to determine the largest possible packet size, is recommended [2] [8]. Using the biggest possible packet also reduces communication overhead. Traditional Path MTU Discovery can take long because its duration depends on the RTT; we will describe a faster method.

[1] Throughout this document, "satellite" refers to a GEO satellite.

2.2 Relieve the backward path

TCP-specifc methods to avoid ACK congestion in the case of asymmetric satellite usage are *TCP Header Compression*, *ACK Suppresion* and *ACK Compaction*. Details on how these methods work can be found in [11]. In the case of adaptive multimedia applications, feedback must be reduced or given less frequently.

2.3 Overcome transmission errors

The only widely deployed method to deal with transmission errors on satellite links is Forward Error Correction (FEC). It should be noted that FEC cannot always be expected to fix link noise which is caused by weather conditions, deep space missions, etc. FEC not only requires additional bandwidth but needs processing and thus may add delay and jitter [2]. Future research into congestion control with TCP-SACK might also turn out to be helpful in this area [5]. Additionally, [1] mentions a new "Corruption Experienced" ICMP error message for transmission error notification.

2.4 Give earlier feedback

A good way to deal with a satellite link's long delay is to actually avoid using it for feedback. Methods are Backwards *Explicit Congestion Notification* (ECN) — an ECN variant which uses the ICMP "Source Quench" message — and a so-called *TCP Proxy* which buffers and acknowledges packets in spite of the actual receiver [10]. The latter, sometimes also called "Link Splitting", is used by the well-known Internet provider "DirecPC".

3 Our solution

We propose to use the *Performance Transparency Protocol* (PTP) to give adaptive multimedia applications earlier feedback. To some extend, it satisfies all of the above four categories: In addition to reduced delay, it also addresses transmission errors, can be used to relieve the backward path and supports efficient Path MTU Discovery. PTP is a transport mechanism to query routers along a path for link information from the "Management Information Base" (MIB) such as the *Maximum Transmission Unit (MTU), nominal bandwidth* (defined as a link's bandwidth when there is no traffic at all) and traffic counters in conjunction with a timestamp and address for *available bandwidth* determination [7]. While these information requests are specified as so-called "Content Types" in [13], the protocol itself should only be regarded as a carrier for performance-specific data. Packets are addressed to the "normal" receiver and treated by routers on-the-fly; to facilitate this kind of usage, it is layered right on top of IP. The protocol is designed for scalability, so there is no per-flow state saving in routers — which also means no state merging as in RSVP and no need for multicast support (PTP packets could be distributed via IPv6 multicast, though).

Making such information available to the public may sound odd to network administrators, but we can argue that these are no secrets anymore: Timestamps and addresses can be retrieved via IP options or traceroute. Path MTU Discovery allows detecting a path's smallest MTU, the "packet-pair" approach presented in [3] determines the bottleneck (nominal) bandwidth. The difference is that PTP *asks* instead of performing bandwidth- or time-consuming measurements. There are two basic ways to use the protocol:

1. *Forward Packet Stamping:* Similar to ATMs "Explicit Rate Feedback" mechanism, where a so-called "Resource Managament" (RM) cell gets updated as it traverses the network, routers update a PTP packet. Instead of changing values, a new dataset is added at each hop.

2. *Direct Reply:* Here, a PTP packet contains only one dataset with values set by the sender according to its performance requirements. Each router along the path compares the values with its own; the first router that cannot support the specified service (e.g. shows a smaller MTU value) updates the dataset and directly returns the packet to the sender, thereby decreasing feedback delay — like that, using a satellite link might be avoided. Direct Reply also has a corresponding option in ATM's "Explicit Rate Feedback": RM cells may be generated on the backward path.

Since the requested information always concerns links and each router is connected to two links that will be used by the packet, one PTP-compliant router can make up for one non-PTP-compliant router. The protocol encompasses a rather simple mechanism to do this, based on the IP headers TTL field; an in-depth explanation is given in [14]. If the information is still incomplete, it can be thought of as an upper limit for the path's capabilities. For instance, the Path MTU will certainly not be bigger than the the minimum obtained from the PTP results table.

It should be noted that redirection as done in Direct Reply mode is regarded as being critical within the Internet community. BECN for the Internet and general usage of the ICMP "Source Quench" message has a long history of discussion, mainly due to problems with scaling and unnecessarily increased backwards traffic. Additionally, redirection cannot work for available bandwidth estimation unless the calculation takes place in the routers. Having routers do the work is a difficult issue; the definition of "available bandwidth", which is given by

$$nom.BW - (counter_2 - counter_1)/i \tag{1}$$

($counter_1$ and $counter_2$ being the traffic counters measured at the beginning and the end of the interval i, respectively) varies depending on i. Second, "available" does not necessarily mean "available to the application in question" — just think of a router with three incoming lines and one outgoing line. In

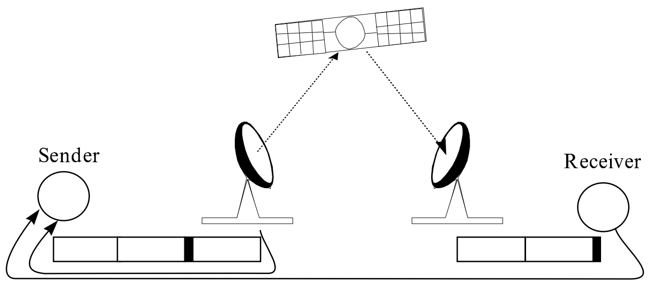

Fig. 1. PTP usage scenario

fact, PTP traffic measurements can be expected to give too high and hence not directly usable values. On the other hand, simulation studies have shown that applying these measurements to adaptive multimedia communication can lead to a significant decrease in the packet loss ratio if they are interpreted as a trend.

4 Applying it to satellite communication

Fig. 1 shows how PTP should be used in a satellite communication context: The first ground station acts as a PTP receiver — it checks for PTP packets (small black packets in the figure) and sends PTP replies back to the sender (the short bent arrow). The sender can see the reply's origin from the packet's source address. While this functionality resembles Direct Reply, it is not; packets are sent using Forward Packet Stamping. The disadvantage is that some per-flow state must be kept at the ground station to support available bandwidth determination. So how does it work?

- The sender sends a PTP packet requesting nominal bandwidth information. This packet may additionally contain a request for traffic counters. It always contains a request for the interface's address, and traffic counters always come with a timestamp.

- As it gets the packet, the ground station saves the information, determines the minimum nominal bandwidth and sends it back to the sender. If there is a need to relieve the backward path, the PTP packet is dropped now. Otherwise, it is forwarded to the receiver which will treat the packet in the same way.

- The sender now starts sending requests for packet counters at a regular interval which depends on the RTT, the nominal bandwidth and the application's QoS requirements; specifics on how to calculate this interval are given in [14].

- For each consecutive traffic counter request and for all datasets found in the packet, the ground station and the receiver calculate the available bandwidth as in (1).

- The path's available bandwidth, which is the minimum result of all datasets, is transmitted to the sender.

To benefit from the earlier feedback, an adaptive multimedia application would have to rely solely on PTP replies. Normally, such an application will base its bandwidth estimations on packet drops and/or delay. It will be able to adapt in certain discrete bandwidth steps with an upper and a lower limit. The primary goal of adaptation is to avoid packet drops for the possible cost of reduced throughput.

5 Evaluation

We simulated the bahaviour of such a "traditional" adaptive multimedia application and compared it with one using PTP. The scenario was a transit-stub network with 100 nodes and a 3/2 mixture of Pareto traffic and FTP connections, the simulator used was "ns-2" [15]. Fig. 2 shows an hour of bandwidth recorded by the receiver of a connection with an average RTT of three seconds. Every six seconds, the sender adapted according to the feedback: In the "normal" case, the bandwidth was changed to the next multiple of the adaptation step below the measured bandwidth or increased by one step if the results showed no bandwidth drop. In the PTP case, the application simply increased or decreased the bandwidth by one step according to the trend. We used a start bandwidth of 56 kbit/s, a step of 5 kbit/s and a lower limit of one step for the traces seen in fig. 2.

It is easy to see that "normal" adaptation shows a much more aggressive behaviour. The PTP trace may appear to show too many fluctuations, but we found that one could easily smooth the curve by reducing the adaptation frequency by at least four times the RTT (in our case, twelve seconds) without any drastic change in throughput or packet loss. In our case, PTP usage reduced the lost/received bytes ratio by 34.76%. Note that this ratio is not just a result of PTP using less bandwidth — an application sending at a constant bit rate of 5kbps only reduced the ratio by 2.3%, whereas starting the adaptive application at 5kbps reduced it by 18.37%.

6 Why not use RTP instead?

The *Real Time Protocol (RTP)* aids real-time applications mainly through synchronisation, payload type identification and quality feedback [12]. RTP directly supports adaptive multimedia applications through RTCP "Receiver Reports" which feed back the amount of lost packets to the sender, somewhat similar to our "traditional" adaptive multimedia application.

A ground station could not reliably send such reports instead of using PTP because RTP may be used on top of UDP, TCP or embedded in the application layer; the only real restriction with respect to port numbers is that only even numbers may be used for RTP and the next (odd) port number must be used for RTCP. In some cases, RTP detection due to frequently used port numbers or typical RTP behaviour may be possible. PTP packets, on the other hand, can easily be detected by examining the IP header's protocol field. Additionally, the "Router Alert Option" is used [4].

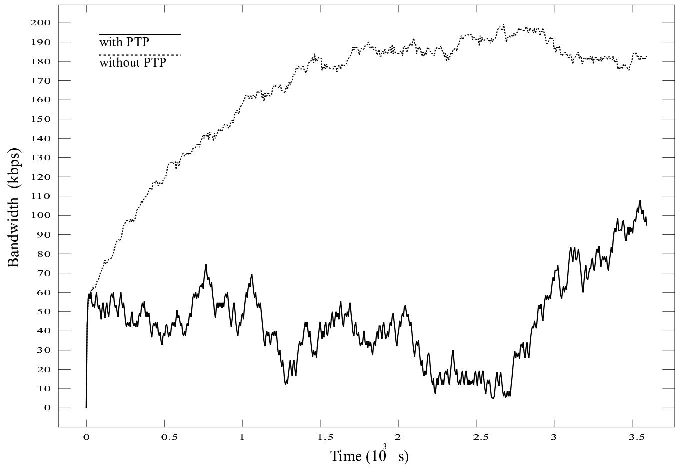

Fig. 2. Bandwidth with and without PTP

Since traffic is counted and not measured, transmission errors do not influence PTP based flow control. While RTP alone cannot satisfy this property, we expect that there would be an advantageous way to combine it with PTP.

7 Conclusion

We proposed to use the Performance Transparency Protocol to overcome some of the problems with adaptive multimedia communication over satellite. The benefits are reduced delay, robustness against transmission errors and support of faster Path MTU Discovery. PTP also has disadvantages: There is some additional communication overhead, PTP packets can be dropped if the network is heavily congested and it needs router support. This means that it must be widely deployed before it can be used by regular Internet applications; an event which may or may never happen.

The way PTP was used for the simulation studies described in this document is very simple. There are probably much better, more sophisticated methods — this is a subject of future research.

References

[1] Allman, M., Dawkins, S., Glover, D., Griner, J., Tran, D., Henderson, T., Heidemann, J., Touch, J., Kruse, H., Ostermann, S., Scott, K., and Semke, J., Ongoing TCP Research Related to Satellites, *RFC 2760*, February 2000.

[2] Allman, M., Glover, D., and Sanchez, L., Enhancing TCP Over Satellite Channels using Standard Mechanisms, *RFC 2488*, January 1999.

[3] Bolot, Jean-Chrysostome, End-to-End Packet Delay and Loss Behavior in the Internet, *Proceedings of SIGCOMM 1993*, pp. 289-298. also in *Computer Communication Review* 23 (4), Oct. 1992.

[4] Katz, D., IP Router Alert Option, *RFC 2113*, February 1997.

[5] Mathis, M., Mahdavi, J., Floyd, S., and Romanow, A., TCP Selective Acknowledgment Options, *RFC 2018*, October 1996.

[6] Metz, Christopher, TCP Over Satellite... The Final Frontier, *IEEE Internet Computing*, Januar / February 1999.

[7] McCloghrie, K., and Rose, M.T., Management Information Base for Network Management of TCP/IP-based internets: MIB-II, STD 17, RFC 1213, March 1991.

[8] Mogul, J.C., and Deering, S.E., Path MTU discovery, *RFC 1191*, November 1990.

[9] Paxson, V., End-to-End Internet Packet Dynamics, *IEEE/ACM Transactions on Networking* 7(4), pp. 1-16.

[10] Ramakrishnan, K., and Floyd, S., A Proposal to add Explicit Congestion Notification (ECN) to IP, *RFC 2481*, January 1999.

[11] Samaraweera, N., and Fairhurst, G., High Speed Internet Access Using Satellite-Based DVB Networks, *Proceedings of Intenational Network Confernence'98*, pp23-28, 1998.

[12] Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V., RTP: A Transport Protocol for Real-Time Applications, *RFC 1889*, January 1996.

[13] Welzl, M., The Performance Transparency Protocol (PTP), *Internet Draft*, draft-welzl-ptp-02.txt, available from http://www.ietf.org

[14] Welzl, M., A Stateless QoS Signaling Protocol for the Internet, *Proceedings of IEEE ICPADS'2000*, July 2000, in press.

[15] Zegura, Ellen W., Calvert, Ken, and Bhattacharjee, S., How to Model an Internetwork, *Proceedings of IEEE Infocom '96*, 1996, San Francisco, CA.