EFFECTIVE EXTENSIONS OF INTERNET IN HYBRID SATELLITE-TERRESTRIAL NETWORKS

Vivek Arora, Narin Suphasindhu, and John S. Baras
Center for Satellite and Hybrid Communication Networks
Institute for Systems Research
University of Maryland at College Park, MD 20742
(301) 405-6610

Douglas Dillon
Hughes Network Systems, Inc.
11717 Exploration Lane
Germantown, MD 20876
(301) 428-5500

Abstract

DirecPC™'s Turbo Internet is a low-cost hybrid (satellite-terrestrial) high-speed digital transmission system developed as a collaborative effort between the Center for Satellite and Hybrid Communication Networks and Hughes Network Systems. The system uses receive-only VSAT satellite links for downstream data delivery and public telephone networks at modem speeds to provide the upstream communications path. One of the services provided is high speed Internet access based on an asymmetric TCP/IP protocol. In the initial protocol implementation, we achieved four times higher throughput than that of today high-speed modems (28.8 Kbps) alone (Falk 1995). This throughput can be further enhanced. The mismatch in bandwidth and delay in this hybrid network prevents the full use of the satellite link bandwidth (1 Mbps). This paper presents two techniques, TCP spoofing and selective acknowledgment dropping, which significantly increase the overall throughput of the hybrid network. Our approach does not require any modification to the TCP/IP protocol stacks on the end hosts. The solutions proposed in this paper could be used to improve TCP/IP performance of other hybrid networks which have the disadvantage of high bandwidth-delay products and/or low bandwidth return paths.

INTRODUCTION

A rapid and feasible (both technologically and financially) development of the NII and GII will follow the following scenario. We can capitalize on the existing installed base of the vast entertainment network(including cable and satellite delivery) and enhance it at no additional cost with many value-added services allowing information browsing and interactivity by the utilization of asymmetric channels. Then modify the end-user devices and service provision with small additional cost, so that the bandwidth differences in the asymmetric channel (and thus the asymmetry) can be variable and modifiable. Then we can let the market and services (to be developed) to determine the actual connectivity requirements on the basis of individual user need. The rest is traffic engineering in the network. This scenario has gained wide spread support recently. For instance, several such products have been offered as one being tested involving either cable or satellite entertainment and Internet type services.

The Center for Satellite and Hybrid Communication Networks and Hughes Network Systems have been working together to develop inexpensive hybrid (satellite and terrestrial) terminals that can provide a variety of services to the user and to foster hybrid communications as the most promising path to the Global Information Infrastructure.

Indeed Internet access is either too slow (SLIP dial-up) or too expensive (switched 56 Kbps frame relay) for individual users or small enterprises. Our solution is what we have called "Hybrid Internet Access". Using hybrid networking, the hybrid terminal merges two connections, a bidirectional terrestrial link using a modem and a receive-only satellite link, so that the TCP/IP software above the device driver sees only one ''virtual'' device.

This design exploits three concepts:

- Satellites are able to offer high bandwidth connections to a large geographical area.
- A receive-only VSAT is cheap to manufacture and easier to install than one which can also transmit.

CONF 960109 © 1996 American Institute of Physics

- Most computer users, especially those in a home environment, will want to consume much more data than they will generate ("asymmetric" computer use).

In order for this hybrid TCP/IP network to be commercially deployable, it must seamlessly interoperate with existing TCP/IP networks. In other words, the following requirements must be satisfied:

- The system must work with *any* Commercial-Off-The-Shelf (COTS) TCP/IP protocol stacks.
- The system must work with *any* SLIP Internet service provider.
- The system must work with *any* hosts on the Internet.

We accomplished the above in the initial prototype of the system (Falk 1995). However, the TCP throughput was not satisfactory. This lead us to consider the performance bottlenecks associated with this type of hybrid network. We have come up with techniques to significantly enhance the TCP throughput of the system.

SYSTEM DESCRIPTION

Figure 1 describes the hybrid Internet access in detail. A hybrid terminal has two network interfaces. One interface is attached to a receive-only VSAT via a special ISA bus PC adapter. The other is a modem attached to a serial port. The hybrid terminal uses a modem connection for outgoing traffic while receiving incoming information through the VSAT. A special NDIS compliant driver combines the two interfaces and make them appear as one virtual interface to upper layer TCP/IP protocol stacks (Falk 1995). The hybrid terminal is attached to the Internet through any Internet service provider who supports Serial Line Internet Protocol (SLIP). The traffic from the hybrid terminal is transmitted to the hybrid gateway through IP-within-IP encapsulation. This encapsulation is needed in order to accomplish asymmetric routing in the Internet and still maintain interoperability with the rest of the Internet without having to modify any existing protocols/routers in the Internet. The hybrid gateway is responsible for decapsulation of traffic from hybrid terminals. It is also responsible for formatting data to suite the satellite transmission. As we shall see in a later section, because all traffic in and out of hybrid terminals must pass through the hybrid gateway, the hybrid gateway can perform certain tasks on behalf of hybrid terminals to achieve better throughput.

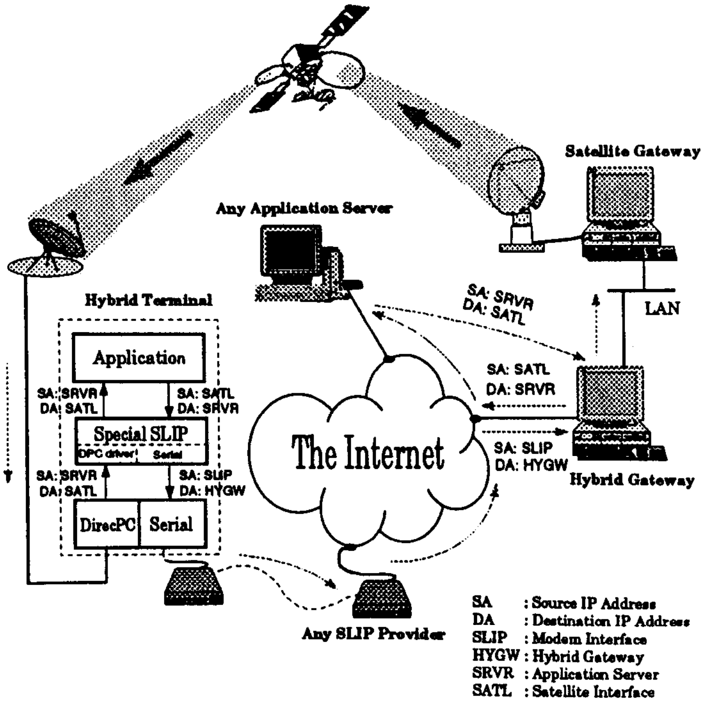

FIGURE 1. Overview of the Hybrid Internet Access.

PERFORMANCE BOTTLENECKS

The TCP protocol uses end-to-end flow, congestion and error control mechanisms to provide reliable delivery over an internetwork. The flow control mechanism depends upon the window size and the round-trip-time (*RTT*). Because a segment of this hybrid network involves a geostationary satellite, the *RTT* grows many times larger than that of terrestrial communications. This means the end hosts must wait longer time for an acknowledgment from the other end. Large value of *RTT* can be compensated by increasing the TCP transmit window size. However, most TCP/IP implementations have small default window size. This window size, although it can be changed to a larger value, it cannot be changed by the hybrid gateway or hybrid terminals. Because our primary design objective is interoperability without modifying existing protocols/parameters at end hosts, we must consider the TCP window size on the Internet host as unchangeable. Thus, the communication channel cannot be efficiently used without other enhancements.

Another factor which decreases the overall throughput is the large amount of acknowledgment traffic generated by the hybrid terminal. These acknowledgment packets are triggered by the high-bandwidth satellite channel. Because our return path is merely a low-bandwidth modem connection, these packets cause congestion in the return path. The congestion leads to longer time for an acknowledgment packet to reach the Internet server. Thus, the overall TCP throughput is reduced.

SOLUTIONS

Figure 2 illustrates the conceptual understanding of the two enhancement techniques we have developed and implemented.

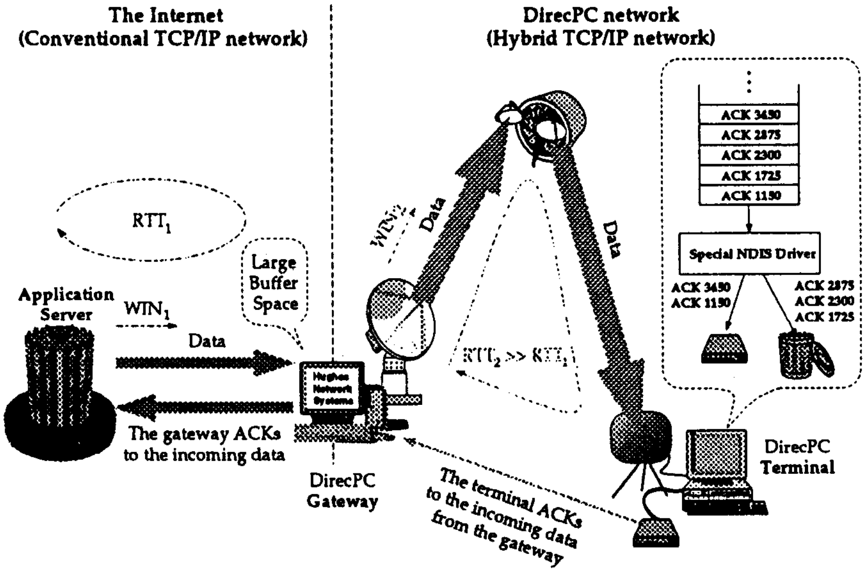

FIGURE 2. TCP Spoofing and Selective Acknowledgment Dropping.

Effects Due to High Round-Trip-Time

We solve the bandwidth-delay problem by transparently splitting the end-to-end TCP connections into two parts: the conventional terrestrial portion and the hybrid portion. Since the hybrid gateway processes both upstream and downstream traffic to and from the hybrid terminal, it will also connect these two networks.

Because of the long delay in geosynchronous satellite communications and the small default window size on an Internet host, the more we isolate the satellite link from Internet hosts, the higher the throughput. Thus, by splitting

TCP connections into two portions, we can isolate the satellite channel from Internet hosts. This isolation is done at the hybrid gateway. The optimum TCP throughput occurs when an Internet host sends out the first TCP segment in a window and receives the acknowledgment for that segment back just when it finishes transmitting the last segment in that window (Steven 1994). What this means is that if the round-trip-time is large, the window size should also be large. Most implementations of the TCP/IP protocol stacks set the default window size to only 4096 bytes (Steven 1994). This window size is considered very small in long-delay high-bandwidth environments such as this hybrid network. We avoid the long-delay effect of the satellite channel by allowing the hybrid gateway to acknowledge incoming data from Internet hosts on behalf of hybrid terminals; thus, reducing the round-trip-time values that Internet hosts expect. According to (Jacobson 1992), the effective bandwidth $B = TCP\_window\_size/RTT$. If *TCP_window_size* is held constant, by reducing the round-trip time, the effective bandwidth will increase. For this particular hybrid network, the overall throughput is now dominated by the minimum throughput of either the conventional TCP/IP portion of the network or the hybrid portion of the network.

To fully utilize the satellite bandwidth, the hybrid gateway is configured to forward data to hybrid terminals as fast as its resources (buffer space) will allow. In other words, the hybrid gateway advertises large transmit windows to hybrid terminals. Return acknowledgment packets from hybrid terminals are used to remove transmitted data from the hybrid gateway's buffer.

Effects Due to Low Bandwidth Return Path

Upon receiving data from Internet hosts through the hybrid gateway, the hybrid terminal generates acknowledgments to the data. Because the downstream path has very large bandwidth, many acknowledgment packets will be generated. These acknowledgment packets can cause congestion in the low-bandwidth upstream path. We solve this problem by selectively dropping redundant acknowledgment packets. This technique works because the TCP acknowledgment scheme is *cumulative* (Comer 1991). This means that the receiver does not acknowledge per packet received but per octet received instead. The special SLIP driver at the hybrid terminal maintains a serial transmit queue. When the queue is building up, it can empty every acknowledgment in the queue but the last one. This last acknowledgment packet acknowledges all octets that the previous acknowledgment packets do. Thus, there is no need to transmit every acknowledgment packet.

IMPLEMENTATION

Unlike traditional implementation of routers or bridges, our implementation also touches a transport layer namely the TCP layer. The hybrid gateway software consists of three major portions (HNS 1994):

- *Hybrid Gateway Environment*: is responsible for interfacing to network adapter.
- *IP Handler*: does segmentation and reassembly of IP packets, handles ARP, encapsulates IP traffic to suite satellite transmission and decapsulates IP-within-IP traffic received from hybrid terminals.
- *TCP Spoofer Kernel*: isolates the hybrid TCP/IP network from conventional TCP/IP network and handles the TCP performance enhancement.

THE TCP SPOOFER KERNEL

The TCP Spoofer operates as a Finite State Machine which manages TCP connections between hybrid hosts and Internet hosts.

Data Structures

• Connection Control Block: A connection control block (CCB) will be kept for each TCP connection being spoofed. A CCB is allocated when a new connection is detected and it remains active as long as the connection is active. It is freed when the connection is terminated normally or is aborted by a TCP Reset or has been idle for a long time.

- CCB Hash Table: To enable fast searching for the CCB of a received segment a hash table is maintained and each CCB is hashed to a bucket based on the tuple *<HH_addr, HH_port, IH_addr, IH_port>*. Chaining is used to resolve collisions in the hash table.

Connection States

A CCB goes through the following states:

- *Closed state*: connection does not exist.
- *Connection-wait state*: in the process of setting up an end-to-end connection between an Internet host and a hybrid host.
- *Connected state*: connection up and data can be transferred.
- *FIN-wait state*: in the process of taking down an end-to-end connection between an Internet host and a hybrid host.

Sending to the hybrid host

The segments are extracted from the front of the queue and transmitted towards the hybrid host over the satellite gateway. The window size advertized by the hybrid host kept in the *snd_win* variable limits the amount if data outstanding. Only the segments falling within the window can be transmitted. A retransmission timer is set using *rrt_estimate*. On each transmission the timer can be set to a higher (double) value compared to the previous value. A *rexmt_count* is kept and when it is exceeded the connection is aborted and the CCB is deallocated.

Zero Window Size Program

When the window advertized by the hybrid host (*snd_win*) goes down to zero the TCP Spoofer will stop sending further data until a non-zero window update is received. A deadlock can occur if the non-zero window update from the hybrid host is lost and the hybrid host is waiting to receive data from the TCP Spoofer.

To avoid this problem the TCP protocol uses a *persist timer* which causes it to periodically send *window probes* to the receiver to find out if the window has been changed. We cannot rely on the Internet host to send the window probes in this situation because in general the window size seen by it (*rcv_win*) can be non-zero when the window seen by the hybrid host (*snd_win*) is zero. The Internet host may have finished transmitting its data at this stage. We need a *persist timer* within the TCP Spoofer to trigger periodic window probes.

Idle Connection

If the CCB has been idle without any unacknowledged segments in the queue and without any traffic in either direction in the CONNECTION-WAIT or CONNECTED state, the CCB can be silently deallocated. An idle timer needs to be set for this which should get reset after segment transmission/reception.

Timer Management

Currently three timers are needed per connection. They are the idle timer, retransmission timer, and the persist timer. There are two ways to manage these timers. One approach is to simply store the timer value in the CCB and at each timer tick scan all the CCB's decrement the timer values and check if any timer has expired. The other approach is to store the timer events in what is known as a *delta list* where they are ordered by relative time. At each timer tick only the first item on the list needs to be decremented and checked for expiration. The former approach is used in the first implementation.

RESULTS

The initial TCP throughput of the system before the enhancement modules were added was around 120 Kbps. Although this throughput is four times higher than that of today high-speed modems (~28.8 Kbps), it is still much less than the throughput of the satellite channel (1 Mbps). With the enhancement modules, the TCP throughput is now peaking at 400 Kbps. This throughput is not optimum. It is a trade-off between throughput and compatibility with existing TCP/IP protocol stacks. Even though the hybrid gateway forwards data with its largest window size (64 KBytes) to the hybrid terminal, the window size is still small in high-bandwidth long-delay environment. Internet RFC 1323 (Jacobson 1992) suggests extensions to TCP protocol which includes TCP window scaling option to allow windows larger than 64 KBytes. These TCP extensions, however, are not widely implemented in COTS TCP/IP protocol stacks.

CONCLUSION

Many issues, which do not appear in conventional networks, emerge when splicing two networks with different characteristics together. In order to make full use of the networks, sometimes unconventional techniques such as the ones presented in this paper must be used. We hope that hybrid networks such as the DirecPC™ network will pave the way towards high-quality and cost-efficient means to connect individual users to the NII/GII.

Acknowledgments

This work was supported by the Center for Satellite & Hybrid Communication Networks, a NASA Center for the Commercial Development of Space (CCDS) under NASA contract NAGW-2777, Hughes Network Systems and the State of Maryland under a cooperative industry-University contract from the Maryland Industrial Partnerships Program (MIPS).

References

Comer, D.E. (1991) *Internetworking with TCP/IP Vol. I: Principals, Protocols, and Architecture*, 2nd edition, Prentice Hall, Englewood Cliffs, New Jersey.

Falk, A.D., Arora, V., Suphasindhu, N., Dillon D., and Baras, J.S. (1995) "Hybrid Internet Access," in *Proc. Conference on NASA Centers for Commercial Development of Space*, M. S. El-Genk and R. P. Whitten, eds, American Institute of Physics, New York, AIP Conf. Proc. No. 325, 1:69-74 (See also: CSHCN TR 95-7).

Jacobson, V., Braden, R., and Borman, D. (1992) "TCP extensions for high performance; RFC-1323," *Internet Request For Comments*, No. 1323, Network Information Center, May 1992.

Hughes Network Systems, Inc. (1994) "DirecPC™ Hybrid Internet Technical Specification," Rev.1.2., No. 8050744, Hughes Network Systems, Inc. Germantown, Maryland., November 1994.

Stevens, W.R. (1994) *TCP/IP Illustrated Volume I - The Protocols*, Addison-Wesley, Reading, Massachusetts.