Networking using Direct Broadcast Satellite[1]

Venkata N. Padmanabhan, Hari Balakrishnan, Keith Sklower, Elan Amir, and Randy H. Katz
{padmanab,hari,sklower,elan,randy}@cs.berkeley.edu

Department of Electrical Engineering and Computer Sciences,
University of California at Berkeley, Berkeley, CA 94720-1776.

Abstract

In this paper, we present a networking architecture based on the DirecPC Direct Broadcast Satellite (DBS) system that supports rapid deployment and easy relocation. A fundamental characteristic of this network is its asymmetry — the downlink has an aggregate bandwidth of 12 Mbps while the reverse channel, such as a dialup line, has a relatively low bandwidth. We describe our extension of the DirecPC routing scheme, which is designed for a single host, to support a subnet with multiple hosts connected to it. We also discuss performance results from UDP and TCP bulk transfers, and Web-like transfers. Finally, we describe how the hybrid subnet has been integrated with the MBone to support real-time audio/video dissemination.

1. Introduction

With the surge in interest in the Internet in recent years, there have been several proposals for providing high-speed network connectivity to homes and offices. These include Hybrid Fiber Coax (HFC), such as those based on cable modem technology, Digital Subscriber Loop (DSL), such as ADSL (Asymmetric DSL), and Direct Broadcast Satellite (DBS), such as Hughes' DirecPC system [8]. Cable modems and ADSL operate over the existing cable TV and telephone networks respectively. Despite the high degree of penetration of these technologies, the uneven quality of wiring and other network components has so far hindered their large scale deployment. In contrast, DBS requires minimal terrestrial support infrastructure and provides universal coverage over a large geographical region.

In this paper, we present a networking architecture using the DirecPC DBS system and discuss the results of several experiments. The DBS system provides asymmetric connectivity with a high-speed DBS downlink and a separate low-speed reverse channel. The original system from Hughes provides connectivity to a single end host and support for dialup return over a telephone line. We have extended it to support multiple hosts and arbitrary reverse connectivity (including an Ethernet that gateways into the Internet). Our TCP bulk transfer experiments indicate that the socket buffer sizes typically used by Internet hosts result in suboptimal performance due to the large delay-bandwidth product of the DBS network. Because the connectivity of the earth station to the Internet is limited by a 1.5 Mbps T1 line, the bandwidth asymmetry is adequately overcome by techniques such as SLIP compression [13]. However, the long latency and the limited reverse channel bandwidth increase latency for Web-like accesses; we show that techniques such as persistent connections and pipelined requests (e.g., [24]) lead to significant performance improvements.

The rest of this paper is organized as follows. Section 2 presents an overview of the DirecPC DBS system. In Section 3, we present an architecture for supporting easy relocation using DBS. In Section 4, we motivate the need for asymmetric routing and discuss various schemes for achieving it. In Section 5, we discuss the performance of reliable and unreliable bulk data transport, and shorter Web-like transfers. In Section 6, we describe an extension of the Internet MBone connectivity to hosts on a DirecPC subnet, to support real-time audio and video dissemination. We summarize our work in Section 7 and discuss plans for future work in Section 8.

2. DBS Overview

Figure 1 depicts the architecture of the DirecPC DBS system, which is designed to provide *asymmetric* network connectivity to users. The system has a geostationary satellite that receives data from an earth station and broadcasts it *directly* to the user's premises. The bandwidth of this broadcast channel is about 12 Mbps and is shared among all users in a large geographical region via packet-level multiplexing. The user is equipped with a 24-inch wide dish antenna which is pointed towards the geostationary satellite. A coaxial cable connects the antenna to an ISA bus adapter card installed in a PC called the *hybrid host*. The reverse connectivity from the user's premises to the Internet is provided via

1. Daedalus project URL: http://daedalus.cs.berkeley.edu

This work was supported in part by DARPA Contract DAAB07-C-D154.

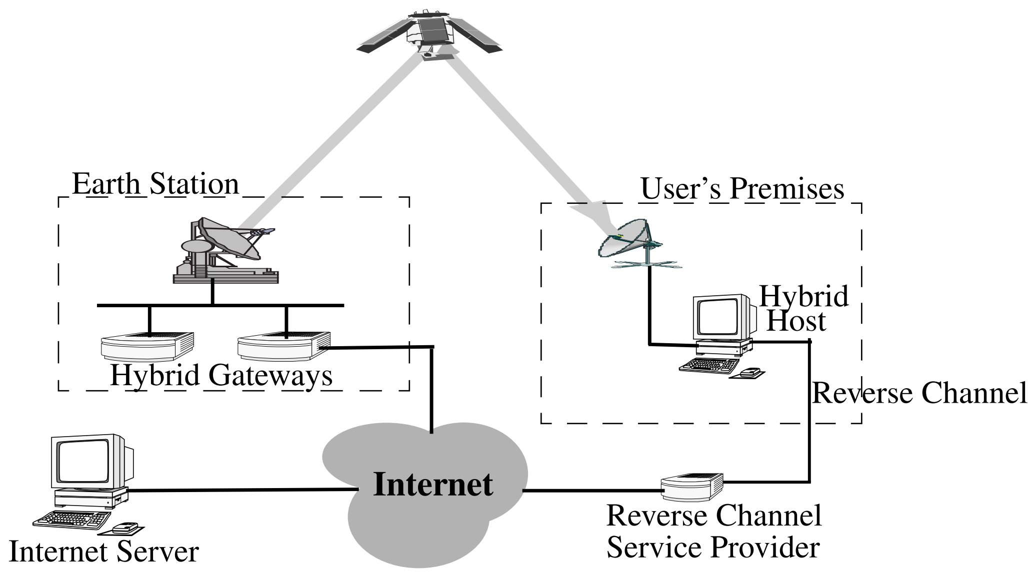


Figure 1. The architecture of the DirecPC DBS system.

one of several different means: a SLIP or PPP dialup line, a cellular modem, a Cellular Digital Packet Data (CDPD) modem, etc. The reverse channel bandwidth is typically much lower than the aggregate downlink bandwidth (15-30 Kbps versus 12 Mbps), resulting in asymmetric ("hybrid") Internet connectivity.

In addition to having an uplink to the satellite, the earth station has a gateway to the Internet. There is a host called the *hybrid gateway*, which acts as an intermediary between the hybrid hosts and other Internet hosts. The earth station has several hybrid gateways, one for each group of more than a hundred hybrid hosts. Each hybrid gateway is connected to the Internet via a 1.5 Mbps T1 line. The downstream data rate to each hybrid host advertised in the system specification is 400 Kbps. However, several hybrid hosts share the T1 capacity associated with the hybrid gateway to which they are connected. Moreover, the 12 Mbps satellite downlink is shared by all hybrid hosts. Thus, depending on the number of active traffic streams, the bandwidth available to a single hybrid host could be lower than the advertised rate, or could actually exceed it.

3. Easy Relocation: "LAN in a Van"

In this section, we describe a networking architecture based on the DirecPC Direct Broadcast Satellite (DBS) system that supports rapid deployment and easy relocation. The motivation for this is to enable an *instant infrastructure* capability.

In our architecture, the hybrid host at the user's premises is connected to a subnet with multiple hosts, which we call the *hybrid subnet*. Figure 2 shows the organization of the local network at the user's premises. As before, there is a hybrid host, with an associated DBS address, which receives data from the satellite's broadcast channel. The hybrid host is also connected to a local area network such as an Ethernet. In general, the reverse channel gateway could be different from the hybrid host. This is useful if the hybrid host cannot be configured as the gateway for the desired reverse channel for any reason, such as hardware or software limitations.

Figure 2 also shows multiple hybrid subnets in the vicinity of the hybrid host. In our testbed, these include a radio-frequency wireless LAN (WaveLAN [2]) and an infrared wireless LAN (IBM Infrared), both of which form a part of the BARWAN wireless overlay testbed [17]. The base stations for both these LANs are connected to the hybrid host and the reverse channel gateway via a 10 Mbps Ethernet.

The entire collection of hybrid subnets is easy to relocate. Establishing the downlink only requires the satellite dish to be oriented in the correct direction[2], while setting up the reverse channel requires low speed connectivity such as a wired phone line or cellular coverage. Because of the ease of relocation, we colloquially refer to this networking architecture as a "LAN in a Van". Many practical scenarios map onto this architec-

2. The current DirecPC DBS antenna must be pointed manually; Hughes is developing a satellite antenna that can orient itself automatically.

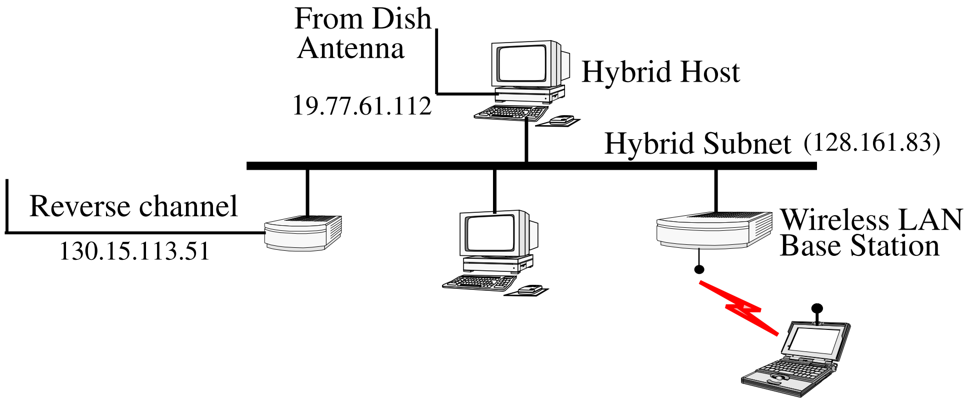


Figure 2. A hybrid subnet ("LAN in a Van"). Note that the gateway for the reverse channel need not necessarily be different from the hybrid host. As shown, the DBS address, the address of the reverse channel gateway and the addresses used on the hybrid subnet are independent.

ture. For example, soldiers might access battlefield information via portable computers connected to base stations located in a mobile command center, or medical personnel may view medical images from inside a repositioning ambulance.

4. Asymmetric Routing

Asymmetric connectivity has implications for the routing of packets to the hybrid host and the hybrid subnet. Outgoing packets from the subnet must traverse the bandwidth-constrained reverse channel. For many types of applications, including Web access, the volume of outgoing traffic is small compared to the incoming traffic. This mitigates the effect of the limited reverse channel bandwidth. While the reverse channel might actually be able to support duplex traffic[3], its limited bandwidth would make it undesirable to receive incoming traffic over it. It would be better to receive such traffic via the high-speed DBS downlink. In Section 4.1, we discuss how this asymmetric routing is achieved in the DirecPC system. (As mentioned in Section 1, the original DirecPC software assumes a dialup reverse channel, but our software can be configured for any form of IP connectivity.) In Section 4.2, we describe our extensions for the more general hybrid subnet case.

4.1 Tunnel-based routing

Figure 3(a) illustrates a tunnel-based solution designed for a hybrid host that is not part of a hybrid subnet. The hybrid host typically has two different IP addresses: one for the DBS downlink (HH_{DBS}) and one for the reverse channel (HH_R) (such as the address assigned by a SLIP service provider). If the outgoing packets sent by the hybrid host to an Internet host (S_{IP}) use HH_{DBS} as the source address in the IP header, then the reply from the Internet host will be sent to the address HH_{DBS}. This is exactly the desired routing behavior.

However, security mechanisms employed by the reverse channel service provider could pose difficulties for this simple solution. Packets with an IP source address in a foreign domain, such as the DBS domain, may be disallowed from traversing the link ("IP-spoofing" protection). In order to solve this problem, the hybrid host encapsulates each outgoing packet in an outer packet with the reverse channel IP address, HH_R, as the source address, and tunnels it to the hybrid gateway (with address HG_{IP}), which decapsulates it. The hybrid gateway routes the inner packet, with HH_{DBS} as the source address and S_{IP} as the destination address, to the Internet host. As before, the Internet host sends any reply packet to the address HH_{DBS}, which gets routed via the DBS downlink. This tunneling mechanism also facilitates the use of techniques such as TCP acknowledgement spoofing [9] by forcing acknowledgement packets from the hybrid host to be routed via the hybrid gateway.

A limitation of this scheme is that it works only for a single hybrid host. Other hosts on the subnet cannot use the address HH_{DBS} to receive incoming traffic via the DBS downlink because the identities of the original senders would be lost in the process, making it impossible to deliver the received packets to the correct host. Another limitation of this scheme is that outgoing packets are always routed via the hybrid gateway which might be far from the most direct route to the destination.

3. In the DirecPC system, subscription keys are exchanged over the duplex reverse channel.

4.2 Home agent-based routing

In order to extend the asymmetric routing mechanism to all hosts in the hybrid subnet, we use a home agent-based routing scheme similar to that employed by the IETF Mobile IP protocol [25]. Each host in the hybrid subnet is assumed to have a home subnet which is well-connected to the Internet, and a corresponding home address, H_{IP}. When a host is away from home and part of the hybrid subnet, it informs a software agent (the *home agent*) in its home subnet of its new *care-of address*, which is the hybrid subnet's DBS gateway (HH_{DBS}).

With this association established, routing works as shown in Figure 3(b). A host on the hybrid subnet uses its home address, H_{IP}, as the source address for outgoing packets. As before, it tunnels the packets to the hybrid gateway, which decapsulates and forwards them on to the final destination. This host sends back its response packets to the address H_{IP}, which get routed to the receiver's home subnet. As in Mobile IP, the home agent (with address HA_{IP}) intercepts these packets and tunnels them via the Internet and the satellite downlink to the hybrid host, HH_{DBS}. The hybrid host decapsulates the packets and delivers them to the destination host. The hybrid host acts like a Mobile IP *foreign agent*, and uses either link-layer connectivity or a host route to deliver packets to the destination. Note that we have assumed that a host uses a local address (in the hybrid subnet) that is the same as its home IP address thereby facilitating local delivery of the decapsulated packet. This requirement can be eliminated by having the home agent do two levels of encapsulation: first with the host's new address in the hybrid subnet and then with the DBS address of the hybrid host.

From Figure 3(b) we see that packets forwarded by the hybrid gateway to the Internet have a source address, H_{IP}, which is outside the DirecPC domain. This may be a problem in the presence of security-conscious routers[4]. An intermediate level of encapsulation solves this problem, as depicted in Figure 3(c). The source address in this intermediate header is HH_{DBS}, which avoids problems with security-conscious routers in both the DirecPC domain and the home domain.

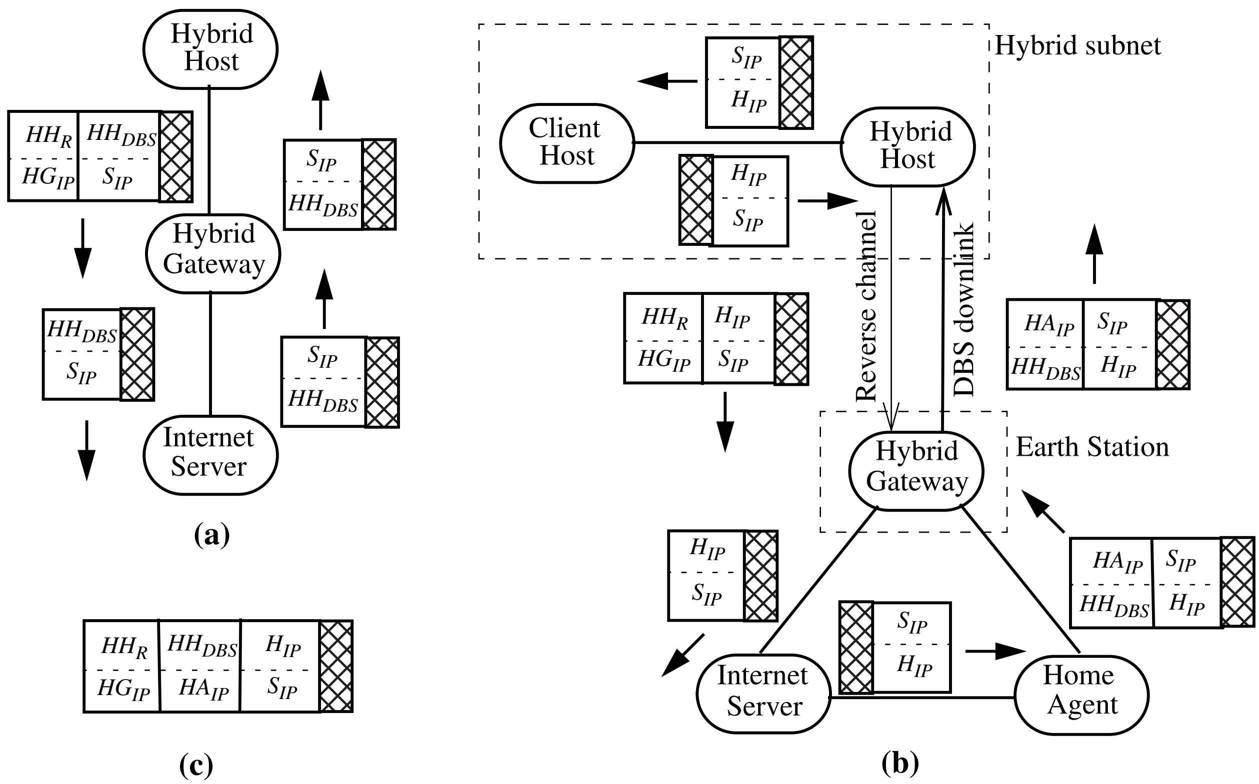


Figure 3. Two schemes for asymmetric routing: (a) tunnel-based, and (b) home agent-based. The rectangular boxes show the headers of an outgoing packet sent to the Internet server and the corresponding reply. The source IP address is in the upper box and the destination IP address in the lower one. The vertical lines separate multiple IP headers when there is encapsulation and the shaded boxes denote data. The hybrid host is assumed to be the gateway for the reverse channel. (c) shows the additional level of encapsulation in the header of an outgoing packet to avoid filtering by security-conscious routers.

This routing scheme works for multiple hosts with possibly different home subnets. However, the involvement of the home agent makes routing more complicated than in the previous scheme. Techniques such as Mobile IP route optimization [22] should mitigate this problem, but we have not yet investigated them.

We also considered using the IP loose source routing (LSRR) option [6] to obtain the desired routing behavior. In this scheme, a host in the hybrid subnet sends out packets with the LSRR option specifying HH_{DBS} as the intermediate node to visit. The destination host is required to reverse the addresses listed in the LSRR option when sending back its reply [6], causing it to be routed via HH_{DBS}. However, two practical problems forced us to abandon this approach: some operating systems (including certain flavors of UNIX) do not perform the route reversal required by [6]; more importantly, the hybrid gateway is configured to filter out source-routed TCP packets for administrative reasons.

5. Transport Protocol Performance

In this section, we discuss some issues pertaining to end-to-end data transport over the DBS system. We begin by pointing out two distinguishing features of this system as far as data transport is concerned: large delay-bandwidth product and asymmetric connectivity. We then present performance results for different types of transfers: UDP transfers, TCP bulk transfers and Web-like accesses.

5.1 Fundamental Issues

Reliable data transport over the DBS system has to contend with the large delay-bandwidth product because of the satellite hop and with the asymmetry between the upstream and downstream bandwidths available to the hybrid host and other hosts on the hybrid subnet. We discuss these in turn.

5.1.1 Large Delay-Bandwidth Product

The delay-bandwidth product of a network path determines its capacity to hold data packets. Reliable transport protocols attempt to match the amount of outstanding data at the sender to this capacity[5]. For instance, a TCP sender constantly probes the network, adjusting its window size to best match the available capacity.

In the DBS system, the long latency of the satellite hop (about 280 ms) coupled with the latency of a typical reverse channel (about 180 ms one-way for a SLIP dialup line) results in a much larger end-to-end delay than for regular Internet paths. This also leads to a large delay-bandwidth product. For instance, a T1 downstream capacity with a round trip delay of 1 second yields a delay-bandwidth product of about 180 KB. This implies the need for large windows and the ability to handle multiple packet losses in a single TCP window. Several extensions to TCP, including window scaling and selective acknowledgements, have been proposed to improve performance over such networks [14, 15, 18], and implementations are available (e.g., [3]).

5.1.2 Asymmetric Paths

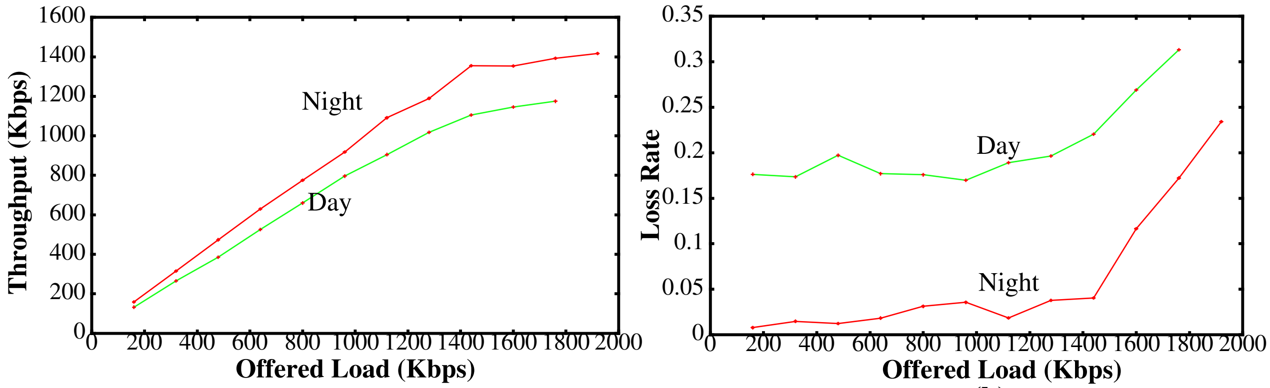
Reliable transport protocols require bidirectional flow of traffic to ensure reliability. In the case of TCP, the acknowledgements that flow from the receiver to the sender are used both for loss recovery and flow control. Therefore, a steady flow of acknowledgments is needed to sustain the flow of data segments in the forward direction. In steady state, a TCP receiver acknowledges every incoming, in-sequence data segment (or every other one with *delayed acks* [26]) and the sender sends new data segments as it receives these acknowledgements (*ack clocking* [12]).

With asymmetric bandwidth, the flow of acknowledgements over the slow reverse channel could throttle the flow of data packets. One way of alleviating this problem is to reduce the size of the acknowledgement packets using techniques such as SLIP header compression [13]. Another is to send TCP acknowledgements less frequently (e.g., [16]), although care must be taken to ensure that the sender does not become very bursty as a result. This may be achieved by regulating the transmission of data at a consistent rate within the TCP congestion window, using the ratio of window size to smoothed round-trip time as an estimate of this rate.

Another problem arises because of the way TCP chooses the maximum segment size (MSS) for a connection. During connection setup (SYN exchange), the peers exchange the MSS they are willing to support and

4. In our experience, the routers in the DirecPC domain do not filter out packets based on their source addresses. However, the routers in our home domain at UC Berkeley do so.

5. Routers along the path are provisioned with buffers to accommodate temporary fluctuations in the flow of data, so the sizes of these buffers should also be added to determine the capacity of the path.



(a) (b)

Figure 4. The throughput (a) and packet loss rate (b), as observed by the receiver, for UDP transfers with different levels of offered load during the day and at night.

pick the minimum of the two as the MSS for the connection. Typically, TCP computes its advertised MSS value as the maximum transfer unit (MTU) of the outgoing network used to communicate with the peer, less the sizes of the TCP and IP headers. This could lead to inefficiencies in the presence of asymmetry. Consider a hybrid host, with a satellite link downlink with an MTU of 1500 bytes and a SLIP reverse channel with an MTU of 300 bytes, receiving data from an Internet host. The MSS chosen for the connection will be 300 bytes, although all data segments will traverse the satellite downlink which has a much larger MTU.

This problem can be solved by suitable modifications to the way TCP at the hybrid host chooses the MSS to advertise. When a connection is initiated by an Internet host, the hybrid host uses the MTU of the interface on which the SYN packet arrives to compute the advertised MSS. In the more common case when the hybrid host initiates a connection, it assumes that the incoming packets will arrive via the DBS downlink and uses the corresponding MTU. (Ideally, the MSS should be renegotiated once the actual incoming interface is known, but TCP implementations typically do not allow this.) In the case of other hosts on the hybrid subnet, the incoming and outgoing interfaces are presumably the same, so this problem is not present. However, to avoid fragmentation along the network path, the peers should perform path MTU discovery [21].

5.2 UDP experiments

We describe a set of experiments involving the transfer of data from an Internet host to the hybrid host using UDP. The path traverses the Internet from UC Berkeley to the earth station in Germantown, Maryland and includes a satellite hop, for a total of 16 network hops. Our goals in performing these experiments are:

1. To determine the characteristics of the bottleneck link in the path between the Internet host and the hybrid host. Since we would like to characterize this link independent of other traffic in the network, we conducted experiments late in the night to minimize the effect of interfering traffic.
2. To determine the performance of UDP transfers in terms of the end-to-end delay and delay jitter. This is of interest in the context of real-time applications such as Internet audio and video, which use UDP as the underlying protocol.

To characterize the bottleneck link, the sender clocks out UDP packets at a constant rate, which determines the *offered load*. Figure 4(a) shows the throughput observed by the receiver as the offered load is varied. The “night” curve, corresponding to low Internet load, levels off beyond an offered load of about 1.4 Mbps. Furthermore, the packet loss rate (Figure 4(b)) also shows a sharp upswing at the point. Thus we estimate the bottleneck bandwidth to be about 1.4 Mbps, which closely matches the bandwidth of the T1 link that connects the hybrid gateway to the Internet (Section 2). It is very likely that this link is the bottleneck in these experiments.

We also gather additional information from the above experiments to help estimate the buffer size at the bottleneck router. Suppose the offered load, l, is greater than the bottleneck link bandwidth, b. For some initial period after the UDP transfer begins, the buffer will be able to absorb the excess inflow of packets. Once the buffer fills up, there will be sustained losses, with a loss rate of $(l-b)/l$. By observing how long it is before sustained losses begin, we attempt to estimate the buffer size[6] as we now explain.

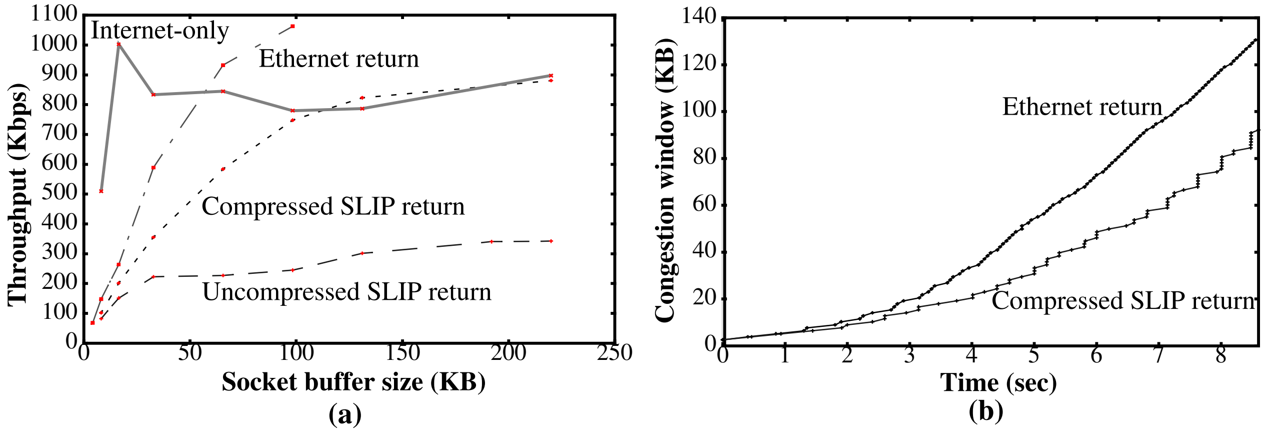


Figure 5. (a) shows the variation of TCP throughput with socket buffer size for several configurations: (i) a DBS downlink with one of these reverse channels: a 14.4 Kbps dialup line with or without SLIP header compression or a 10 Mbps Ethernet reverse channel, and (ii) an Internet-only path involving 19 hops. The experiments were conducted at night when the Internet is presumably lightly loaded. (b) shows the initial growth of the TCP congestion window during slow start for two configurations. The total duration of the 2 MB transfers is 19.3 seconds with an Ethernet reverse channel and 27.4 seconds with a compressed SLIP reverse channel.

Suppose packets of size *s* are clocked out by the sender every *t* seconds. Let *i* be the index of the packet that marks the beginning of the period of sustained losses[7]. At this point in time, the amount of data the sender has sent is *si*. The amount of data that has been drained out of the bottleneck buffer in the same period can be estimated as *itb*. The bottleneck buffer size, *B*, can then be estimated as *si - itb*.

The offered load, *l*, must be greater than *b* to avoid the possibility of the bottleneck link being idle at times. However, it should not be too much greater because that might cause a link which is earlier in the path than the bottleneck link and has a bandwidth that lies between *b* and *l*, to cause packet drops. We conducted several experiments with offered loads in the range 1.6 Mbps to 2.25 Mbps, and consistently estimated the bottleneck buffer size to be between 56 and 64 KB.

We also used UDP packet streams to measure the end-to-end delay jitter. The sender clocks out packets with a constant inter-packet spacing, and the receiver records the spacing between the received packets. The jitter is then computed as the mean deviation of the inter-packet spacing at the receiver. Our measurements show that the jitter of the satellite downlink (measured at night) is 25-30 ms.

6. Other traffic at the bottleneck link will lead us to underestimate the buffer size. We conducted these experiments late at night to minimize the chances of such interference.

7. The bottleneck buffer must employ a drop-tail policy for *i* to be useful. Since successive experiments with the same offered load yield similar values for *i*, we can (at least) deduce that the gateway does *not* employ a random-drop policy.

5.3 TCP experiments

We conducted a series of TCP bulk transfers between the same pair of hosts as before using three kinds of reverse channels for acknowledgements: (i) a 14.4 Kbps dialup line without SLIP compression, (ii) the same with SLIP compression, and (iii) a 10 Mbps Ethernet link that directly connected the receiver to the sender. We also conducted transfers between two hosts connected by a 19-hop Internet path not involving either the DBS downlink or the SLIP link. Figure 5 shows the results obtained for 2 MB long TCP transfers.

5.3.1 Effect of asymmetric paths

Figure 5(a) shows the throughput over the DBS downlink with the three different kinds of reverse channels. Without SLIP header compression, each acknowledgement packet is 40 bytes long. Since the line speed is 14.4 Kbps, a maximum of 45 acknowledgement packets can flow over the SLIP line every second. In our experiments, the MSS used by the sender is 1280 bytes but that used by the receiver (the hybrid host) is only about 300 bytes. This causes the receiver to acknowledge every incoming segment which is larger than twice *its* MSS of 300 bytes, thereby defeating the TCP delayed acknowledgements scheme. This limits the data rate to 45 segments per second, yielding an upper bound of 460 Kbps on the throughput. The best case throughput observed in our experiments is only about 350 Kbps; one of the reasons for the shortfall is that it

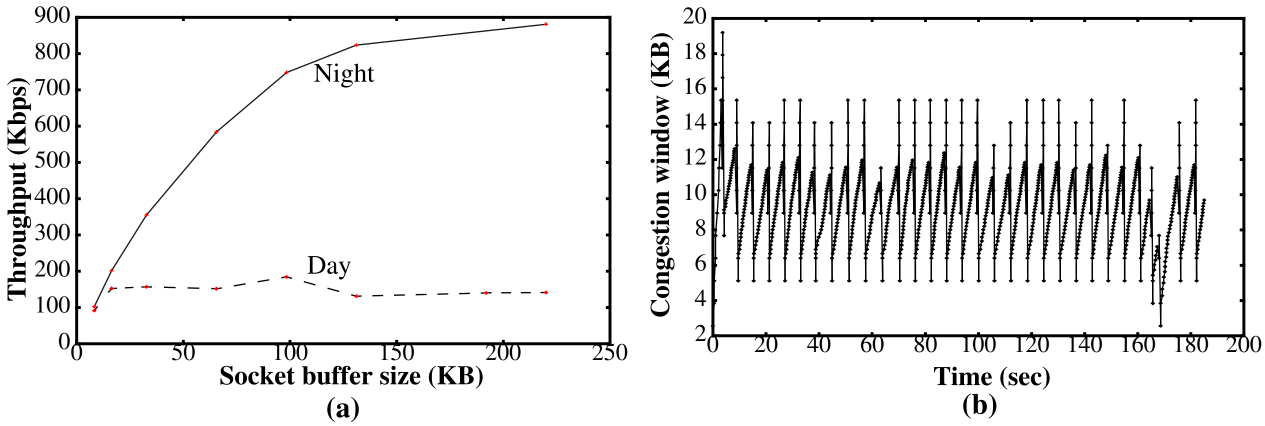


Figure 6. (a) shows the TCP throughput over the DBS downlink with a 14.4 Kbps dialup line as the reverse channel. Curves are shown for both for experiments conducted during the day and at night. (b) shows the variation of the TCP congestion window for a transfer during the day. The rapid fluctuations due to fast retransmissions and coarse timeouts result in a small average window size.

takes several seconds for the sender's congestion window grow to a large enough size during the TCP *slow start* phase [12].

Header compression reduces the size of TCP acknowledgements to about 6 bytes, which yields an upper bound is 3.1 Mbps on throughput. As mentioned in Section 5.2, the bandwidth of the bottleneck link in the system is about 1.4 Mbps, so the flow of acks over the compressed SLIP line is not the limiting factor for throughput. The best case throughput is about 880 Kbps. This is lower than the throughput obtained with an Ethernet reverse channel because the SLIP line introduces an additional latency of about 180 ms, resulting in an increased delay-bandwidth product. The larger delay also causes the window size to grow more slowly during slow start (Figure 5(b)), which reduces throughput further.

5.3.2 Effect of large delay-bandwidth product

As Figure 6(a) shows, the throughput at night (with few Internet losses) over the satellite downlink increases steadily for small buffer sizes and levels off beyond 130 KB. The large delay-bandwidth product of the network path necessitates such large buffers to achieve optimal throughput. During the day, the trend in throughput is less marked. This is because Internet losses cause the sender's congestion window size to fluctuate, resulting in a small average window size (Figure 6(b)), which nullifies the benefits of large socket buffers. Preliminary experiments of TCP enhanced with selective acknowledgements [3] indicate a 30-50% improvement in throughput.

Figure 6(a) also shows the variation of throughput with socket buffer size for Internet-only transfers. A buffer size of about 16 KB achieves the best throughput, and there is a 10-20% degradation for larger buffer sizes. This is so because the delay-bandwidth product of the Internet-only path is small (about 16 KB), and larger buffer sizes cause the sender to frequently exceed this capacity, resulting in losses (and consequently, degraded throughput).

Internet hosts, such as Web servers, typically use buffer sizes of 8-32 KB for TCP connections, which results in good performance for Internet-only transfers but poor performance for transfers to a DBS hybrid subnet. One solution is for the servers to use large socket buffers for all connections at the cost of increased memory requirements. Client hosts that do not need the large buffers could advertise smaller windows, thereby avoiding the possibility of performance degradation (as discussed above for Internet-only transfers). Another possible solution is to split the end-to-end connection for transfers to a hybrid subnet into separate parts for the Internet path and for the DBS downlink[8]. Only the sender of the latter connection needs to have a large buffer size. In the case of Web accesses, a (caching) proxy could be located at the satellite earth station to explicitly split the connection.

5.4 Web performance

The World Wide Web [4] uses TCP as the underlying transport protocol. The base HyperText Transfer Protocol (HTTP [5]) uses a separate TCP connection for each file retrieval (HTML, GIF, JPEG, etc.). Further-

8. The DirecPC hybrid gateway can be configured to spoof TCP acknowledgement packets to effect such a split.

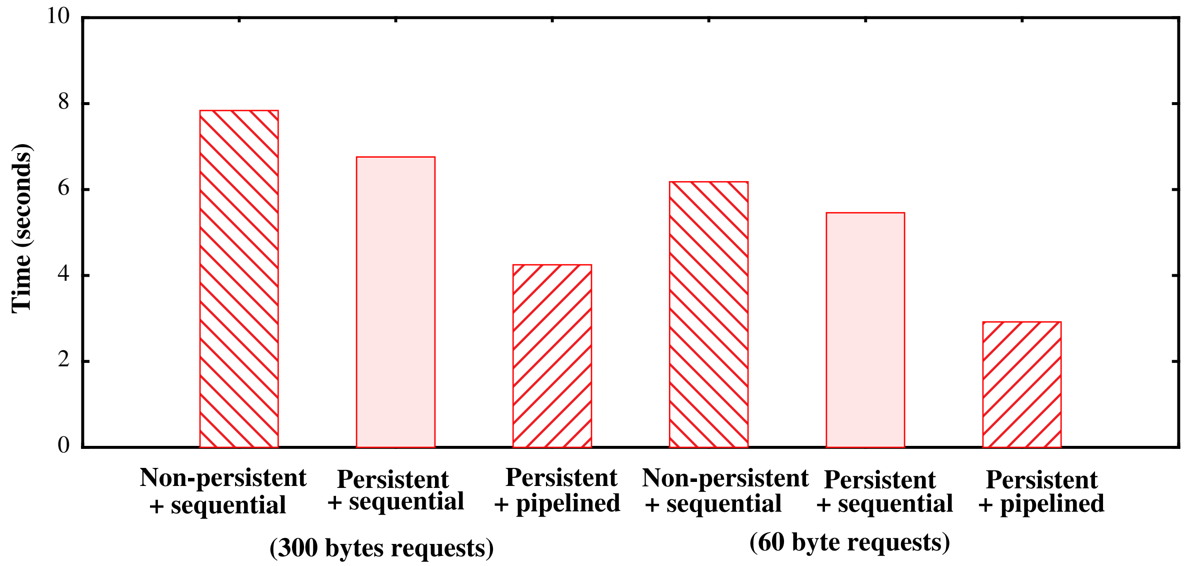


Figure 7. The time to complete the Web microbenchmark for various configurations and parameters settings. The forward path involves the DBS downlink and the reverse channel is a 14.4 Kbps dialup line with SLIP compression. For comparison, when the SLIP line is used as both the uplink and the downlink, the completion time is 25-30 seconds.

more, each transfer is typically quite short. These together result in increased retrieval latencies because of the large number of networks round trips involved and because each TCP connection does not last long enough to build up a substantial window. This is particularly so in the DBS system due to the large end-to-end latency. There have been proposals to mitigate these problems by making HTTP use more persistent connections [10,24].

The asymmetric nature of the DBS system also results in increased latencies. Web requests are sent over the bandwidth-constrained reverse channel. Our analysis of traces indicates that typical Web requests are 250-350 bytes long. Over a 14.4 Kbps SLIP line, this translates to a transmission time of 150-200 ms. In addition, the SLIP line has a one-way latency of about 180 ms. Techniques such as pipelined Web requests [24] or issuing requests in parallel [23] can help minimize the effect of this large latency. Furthermore, successive Web requests from a client to the same server have large portions (65-80%) in common, so techniques to coalesce multiple requests (e.g., [24]) could be also be useful.

We have developed a simple microbenchmark to help quantify the various effects discussed above. A simulated client sends a request of a certain size to a simulated server. In response, the server sends back a 5 KB long file (which is representative of the size of a single Web transfer). This process is repeated four times to simulate the download of a Web document comprising four files. Figure 7 shows the time to complete this transaction for various configurations. The baseline case is non-persistent TCP connections with sequential requests. Using one persistent TCP connection reduces the time for the transaction by about 12-14%. Combining persistent connections with pipelined requests reduces the time by about 50%. Finally, reducing the requests size from 300 bytes to 60 bytes (reflecting the coalescing of successive requests) reduces the time by 20-30% for each of the three configurations.

6. Real-time Audio/Video Dissemination

The Internet multicast backbone (MBone [19]) has been used to support audio/video teleconferencing applications for the past few years. Currently, our DirecPC installation does not support IP multicast [7]. We have developed techniques to extend MBone connectivity to a hybrid subnet using application-level agents integrated in a proxy architecture. The basic idea is to overcome the lack of multicast support by sending unicast packets to the hybrid host, which then uses IP multicast to redistribute packets receivers connected to the hybrid subnet, possibly via heterogeneous networking technologies (such as Ethernet, WaveLAN, infrared LAN, etc.). Another issue is the need to control the rate of the real-time streams in order not to saturate the satellite downlink.

The architecture includes RTP-aware transcoding agents and a control agent called the *proxy*. The RTP-aware agents are capable of transcoding between different audio/video formats and converting between unicast and multicast packets [1]. The proxy conveys MBone session directory protocol (SDP [11]) messages to the hybrid subnet, and configures and controls the transcoding agents based on requests from hosts on the hybrid subnet. Ideally, the proxy and the transcoders should run close to the satellite earth station. A transcoder is also run on the hybrid host to convert the incoming unicast packets to multicast packets.

We have demonstrated real-time dissemination of audio and video streams using the MBone tools, *vic* [20] and *vat*, with the transcoding agents running in our home environment at UC Berkeley. In the current system the transcoders are configured manually. In the future, these configurations will be performed dynamically and adaptively by the proxy in response to changing link conditions.

7. Summary

In this paper, we have presented a networking architecture based on the DirecPC Direct Broadcast Satellite system that supports rapid deployment and easy relocation, colloquially referred to as a "LAN in a Van'. A fundamental characteristic of this network is its asymmetry; the downlink has an aggregate bandwidth of 12 Mbps while the reverse channel has a relatively low bandwidth. We have presented an asymmetric routing scheme that can support an entire hybrid subnet rather than just a single host. We have also discussed performance results from UDP and TCP bulk transfers, and Web-like transfers. Our results indicate that the TCP socket buffer sizes typically used by Internet hosts result in sub-optimal performance due to the large delay-bandwidth product of the DBS network. Because the connectivity of the earth station to the Internet is limited by a 1.5 Mbps T1 line, the bandwidth asymmetry is adequately overcome by techniques such as SLIP compression. However, the long latency and the limited reverse channel bandwidth increase latency for Web-like accesses; our experiments show that persistent connections and pipelined requests lead to significant performance improvements. Finally, we have described how the hybrid subnet has been integrated with the MBone to support real-time audio/video dissemination.

8. Future Work

The experiments described in this paper were conducted between an Internet host at UC Berkeley and a DBS hybrid host. While these end-to-end measurements are useful, the characteristics of the Internet path (including the T1 line going in to the hybrid gateway) tend to obscure the DBS downlink. We plan to conduct experiments, with an additional point of control at the earth station, to isolate the downlink. These measurements would be very useful for Web caching and prefetching studies that we plan to do. (A cache at the earth station can bypass the Internet when sending data to hybrid hosts.)

We are investigating the benefits of decreasing the frequency of TCP acknowledgements in the presence of bandwidth asymmetry. We are also investigating the usefulness of techniques that combine window-based flow control and rate-based flow control. We plan to include other asymmetric networks, such as those based on cable modems, in our study.

Acknowledgements

Srinivasan Seshan provided several useful comments on a draft of this paper. Son Dao, Yongguang Zhang and Dante de Lucia helped set up the real-time audio/video dissemination demonstration. We would like to thank them all.

9. References

[1] E. Amir, S. McCanne, and H. Zhang. An Application Level Video Gateway. In *Proc. ACM Multimedia '95*, November 1995.

[2] AT&T. *WaveLAN: PC/AT Card Installation and Operation*, 1994.

[3] H. Balakrishnan. An Implementation of IETF Selective Acknowledgements for TCP. ftp://daedalus.cs.berkeley.edu/pub/tcpsack/, 1996.

[4] T. Berners-Lee, R. Cailliau, A. Luotonen, Frystyk H. N., and A. Secret. The World-Wide Web. *Communications of the ACM*, 37(8):76–82, Aug 1994.

[5] T. Berners-Lee, R. Fielding, and H. Frystyk. *Hypertext Transfer Protocol – HTTP/1.0*. RFC, May 1996. RFC-1945.

[6] R. T. Braden. *Requirements for Internet Hosts – Communication Layers*. Information Sciences Institute, Marina del Rey, CA, October 1989. RFC-1122.

[7] S. E. Deering. *Multicast Routing in a Datagram Internetwork*. PhD thesis, Stanford University, December 1991.

[8] DirecPC Web page. http://www.direcpc.com, 1996.

[9] A. D. Falk. A System Design for a Hybrid Network Data Communications Terminal Using Asymmetric TCP/IP to Support Internet Applications. M.S. Thesis, University of Maryland, 1994.

[10] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. Internet Draft, IETF, August 1996. Expires Feb 1997.

[11] M. Handley and V. Jacobson. SDP: Session Directory Protocol. Internet Draft, IETF, November 1995. Expires May 1996.

[12] V. Jacobson. Congestion Avoidance and Control. In *Proc. ACM SIGCOMM 88*, August 1988.

[13] V. Jacobson. *Compressing TCP/IP Headers for Low-Speed Serial Links*, February 1990. RFC 1144.

[14] V. Jacobson and R. T. Braden. *TCP Extensions for Long Delay Paths*. RFC, Oct 1988. RFC-1072.

[15] V. Jacobson, R. T. Braden, and D. A. Borman. *TCP Extensions for High Performance*. RFC, May 1992. RFC-1323.

[16] P. Karn. Dropping TCP acks. Mail to the end-to-end mailing list, February 1996.

[17] R. H. Katz et al. The Bay Area Research Wireless Access Network(BARWAN). In *Proceedings of the Spring COMPCON Conference*, San Jose, CA, 1996.

[18] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. TCP Selective Acknowledgments Options. Internet draft, Draft-ietf-tcplw-sack-00.txt, January 1996. Expires July 1996.

[19] The MBONE Information Web. http://www.mbone.com/, 1996.

[20] S. McCanne and V. Jacobson. vic: A Flexible Framework for Packet Video. In *Proc. ACM Multimedia '95*, November 1995.

[21] J. C. Mogul and S. E. Deering. *Path MTU Discovery*, April 1990. RFC 1191.

[22] A. Myles, D.B. Johnson, and C. Perkins. A Mobile Host Protocol Supporting Route Optimization and Authentication. *IEEE Journal on Selected Areas in Communications*, 13(5), June 1995.

[23] Netscape Communications Corporation. http://home.netscape.com/, 1996.

[24] V. N. Padmanabhan and J. C. Mogul. Improving HTTP Latency. In *Proc. Second International World Wide Web Conference*, October 1994.

[25] C. Perkins (editor). IP Mobility Support. Internet Draft, IETF, May 1996. Expires Nov 1996.

[26] W. R. Stevens. *TCP/IP Illustrated, Volume 1*. Addison-Wesley, Reading, MA, Nov 1994.