# Implementation and Evaluation of Satellite Internet System

Noboru FUJII, Kazuhiro HARA, Makoto ISHII, Masateru AKACHI, and Ichiro KUBOTA*

**Abstract--Use of the Internet is growing explosively, not only for text-based applications such as e-mail and bulletin boards, but also for multimedia WWW applications involving video and audio, and multicast services. Recently, satellites have been used in communication networks as a means of avoiding terrestrial Internet traffic jams. Satellites are a very efficient means of delivering large volumes of content to multiple clients simultaneously. We show that fast throughput using TCP/IP connection is feasible in networks comprising both satellite and terrestrial links. Then we propose a UDLR (Uni-Directional Link Routing) mechanism for networks that include uni-directional links (such as satellite links), and show that general dynamic routing algorithms such as RIP and OSPF can be adopted even when using uni-directional links.**

**Index terms-- satellite Internet, TCP/IP, UDLR, NAT**

## I. INTRODUCTION

Communication networks that utilize satellites which provide large data transmission capacity and multicast capability have attracted increasing attention. Some multi-branch corporations have introduced communication networks with satellite links through which efficient networks can be established for high speed data transmission. Multicast or push type applications have increased traffic on terrestrial Internet links. Satellite links that feature multicast functions can reduce network traffic by bypassing the traffic of terrestrial links.

This has led to an increase in the use of satellites to deliver IP (Internet Protocol) datagrams. However, satellite-based networks currently have limited application due to limitations associated with protocols for satellite links. In addition, satellite-based networks can be cost prohibitive.

The introduction of digital satellite broadcasting services has popularized the application of MPEG2-based technologies standardized by DVB (Digital Video Broadcasting). Although DVB-based systems are one-way communication systems, they are cost effective because expensive terrestrial links such as cables and ATM lines are not required. Our group first introduced DVB compliant IP-based satellite communication systems three years ago, and have proposed specifications to handle IP datagrams on DVB-based systems at DVB, DAVIC (Digital Audio-Visual Council), and IETF (Internet Engineering Task Force).

To design a satellite-based system, high data throughput must be established, because a satellite link can have as long as a 240ms time delay for one link. We demonstrated that high data throughput can be achieved through a TCP/IP connection by establishing a large window size.

The mechanism for IP routing must also be considered. We have developed hybrid networks using two different links: terrestrial links for requesting data and satellite links for delivering requested data. A special mechanism is required for dynamic routing in hybrid networks, because current dynamic routing protocols are designed on the principle that the networks are bi-directional; however, the satellite link is uni-directional. Consequently, routing information cannot be exchanged between the transmission side and reception side of the satellite link. The router on the transmission side of the satellite link recognizes that it is not possible to reach the reception side because the router on the transmission side cannot receive routing information on the network of the reception side. To overcome these problems, we introduced UDLR (Uni-Directional Link Routing) which controls dynamic routing for networks using a uni-directional link. This approach enables dynamic routing to be realized. Standardization of UDLR is being conducted at UDLR WG of IETF.

The present report outlines the system architecture of our proposed satellite communication system. Furthermore, we examine data throughput of the TCP/IP connection. Finally, we describe the UDLR mechanism and evaluate the use of UDLR.

## II. SATELLITE INTERNET SYSTEM

### A. System architecture

The satellite Internet system is composed of a transmission system and reception system (Fig. 1).

The transmission system is composed of transmission routers (henceforth referred to as Feed), multiplexers, QPSK (Quadrature Phase Shift Keying) modulators, as well as other components. Feed is achieved using a PC with LAN interfaces, such as Ethernet and FDDI, as well as satellite link interfaces. The satellite interface is established using a PCI board (henceforth referred to as an encoder board), and the Feed performs routing at the IP level. The encoder board encrypts the IP datagram, encapsulates the IP datagram into the MPEG2 private section, and then divides the section into the Transport Stream Packet (henceforth referred to as the TSP). By this process, a high-speed transfer rate of approximately 30Mbps, which corresponds

* Sony Corporation Osaki West Technology Center, 2-10-14, Osaki, Shinagawa-ku, Tokyo, 141-0032 Japan
*e-mail: fujii@dct.sony.co.jp*

to the bandwidth of one transponder, can be achieved. The output of the encoder is connected directly to the multiplexer by the DVB ASI interface. The multiplexer multiplexes input TSP by timesharing (TDM: Time Division Multiplexing), and outputs the multiplexed TSP to the modulator.

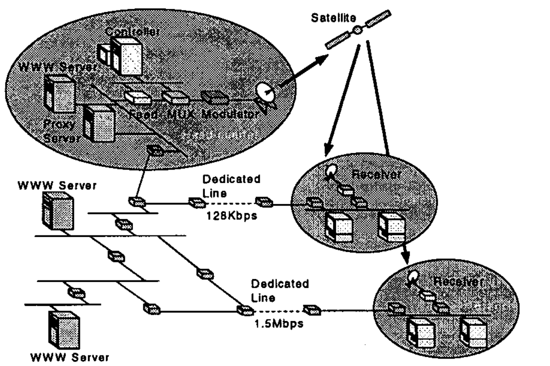
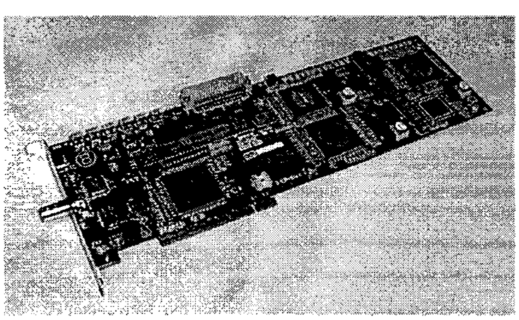



Fig. 1: Topology of Satellite Internet System

The reception system is composed only of the reception router (henceforth referred to as the Receiver) using a PC in which the reception board with the PCI interface is built. Reassembly of the IP datagram from the TSP occurs at the reception board, and the IP datagram is then forwarded to the main memory of the PC through the PCI bus. Throughput of at least 30Mbps can be achieved in this reception board. Device drivers for Windows NT Ver.4.0 and FreeBSD-2.2.6R for PC UNIX have been developed for this reception board.

The external charts of the encoder board and reception boards are shown in Fig.2. These are full-sized PCI boards. However, the size of these boards can be reduced dramatically using technology that enables the front-end to be miniaturiezed and by incorporating the data processing part into the ASIC.



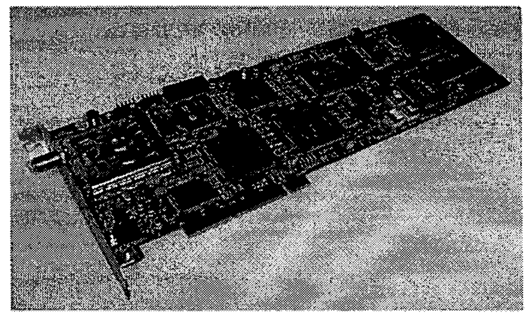


Fig. 2: Encoder Board and Reception Board

### *B. Data link format*

In this system, a method called Multiprotocol Encapsulation, which was standardized by DVB and DAVIC, has been adopted as the data link format to encapsulate IP into the private section of MPEG2. This method not only transports the IP datagram but also encrypts the payload and MAC address part.

### *C. CA (Conditional Access)*

CA plays an important functional role in the operation of a communications system. We have implemented two functions for encrypting data on satellite links: one at the TSP level by MULTI2 specified by the 74th telecommunication technology council report, and the other at the IP level. Encryption at the TSP level is currently being used by digital satellite broadcasting services such as SKYPerfecTV!, and the function is implemented in the multiplexer produced by Sony.

Encryption at the IP level occurs in the encoder board of Feed when IP datagrams are encapsulated into private sections. A common key system is adopted for the IP level encryption method. It is necessary to provide session key (Ks) for IP datagram decryption to the Receiver side before communication starts. These keys can be provided either by satellite links or by terrestrial links.

### *D. Features of satellite links*

Compared with other types of communication links, satellite links have some unique features. First, satellite links are uni-directional. Although some systems, such as VSAT (Very Small Aperture Terminal) are equipped with interfaces for both a transmitting and receiving, DVB-based Receivers are only equipped with a data receiving function to make them more cost effective. Another characteristic of satellite links is a large transmission delay. With geo-stationary satellites, which are located approximately 36,000 km above the surface of the earth, the delay from Feed to Receiver is approximately 240ms. A means of achieving high data throughput must be achieved to overcome this large transmission delay.

### E. *Routing control in satellite link*

Because satellite links are uni-directional, we must develop a new control method for routing IP datagrams in a UDL (uni-directional link) environment.

The routing program on the Receiver cannot transmit route information to the subnet of the UDL. Since existing routing protocols are designed based on a bi-directional link , information on the adjacent network connected to the Receiver cannot be transmitted to the Feed. Moreover, address solution cannot be performed because the ARP packet cannot be transmitted from the Receiver to the Feed through a UDL. Thus, ARP, which solves the address of the data link, cannot operate if two-way communication is not possible through the same subnet.

Consequently, the influence of uni-directional satellite links on the IP network is substantial. We have introduced the UDLR to overcome the problem of routing control on such UDLs.

## III. TCP/IP THROUGHPUT ON SATELLITE LINK

### A. *TCP throughput*

While satellite circuits provide wide bandwidth, they also have an inherent transmission delay of approximately 250ms between Feed and Receiver. Further, if IP datagrams are sent via terrestrial links as outgoing packets and received from the satellite as incoming packets, the round-trip time is over 400 ms. Therefore, this transmission delay greatly influences the throughput when using TCP, which is a positive acknowledgement method, as the transmission protocol. TCP throughput can be calculated by the following equation:

$$\text{Throughput [Byte/s]} = \frac{\text{TCP Window Size [Byte]}}{\text{RTT (Round Trip Time) [s]}}$$

This equation is applicable when the TCP window size is at its maximum. Therefore, actual throughput tends to be less than the calculated value because data transmission generally occurs at a half-opened window size when small files are transmitted.

### B. *Evaluation of TCP performance*

We examined the relationship between the size of the window and the throughput of TCP in an environment that uses a satellite link with a transmission delay. Experiments were performed using the FreeBSD-2.2.6 platform for both the server and client. Fig.3 shows the measurement result. We found that throughput increased as the size of the TCP window was increased.

Based on these results, high-speed throughput of a large file appears to be attainable using an enlarged TCP window.

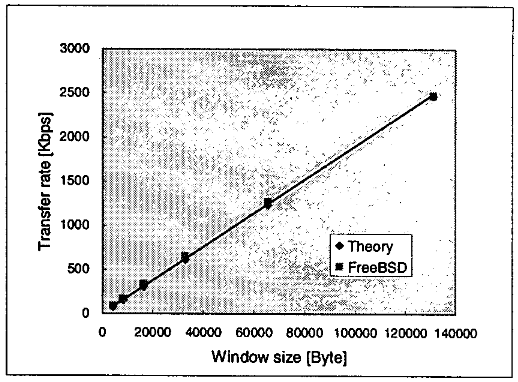



Fig. 3: Throughput of File Transfer

## IV. UDLR

### A. *Problems handled by UDLR*

UDLR is a mechanism that enables routing protocols to work on UDL. This can be achieved by two approaches: through the development of a new routing protocol that can handle UDL. This approach is time intensive for the development and standardization of such a routing protocol or modification of current routing protocols.

The other approach is link layer emulation, which enables upper protocols to handle the UDL as well as a BDL.

Although our goal is to develop a practical system, due to time constraints, we have begun development of link layer emulation as a short-term solution.

### B. *Broadcast emulation*

We considered that a UDL could be handled similar to a BDL at the IP layer in which routing protocols occur. We have called this mechanism "Broadcast Emulation (BCE)". BCE emulates a BDL which is a broadcast network. The BCE layer is located between the IP layer and link layer as shown in Fig. 4.

The BCE is designed to enable a packet be sent from a receiver to a feed. The method of achieving this function is described in the next chapter.

### C. *BCE with IP tunnel*

To achieve BCE, packets need to be transmitted from a receiver to a feed and the other receivers. We used "IP tunnel" in order to achieve this.

The following describes how routing programs on a feed and a receiver exchange the packets. A routing program sends routing information packets to neighboring routers and receives packets from neighboring routers. Fig. 4 shows how a receiver that has a receive-only UDL interface sends a routing information packet to a neighboring feed.

The routing program on the receiver attempts to send a routing information packet. At this time, the BCE layer hooks the packet and encapsulates it in another IP datagram, and sends it from the BDL interface. When the IP datagram sent from the receiver arrives at the BDL interface of the feed, decapsulation is processed in the BCE layer and the decapsulated packet is handled as if it were received from the UDL interface. In addition, if the packet is a broadcast packet or a multicast packet, the BCE layer forwards it to the UDL for the other receivers. In this way, the BCE layer achieves complete broadcast emulation.

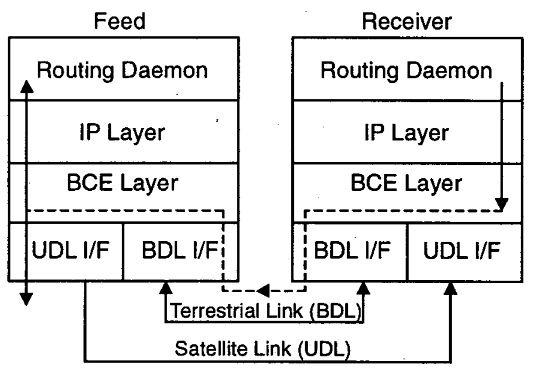



Fig. 4: Link Layer Tunneling Mechanism

*D. DTCP*

Another important function to achieve UDLR is DTCP (Dynamic Tunneling Configuration Protocol). DTCP is used when a feed sends information required by receivers in order to make an IP tunnel; for example, an IP address of the end point of the IP tunnel, time-out value, or protocol number of the IP tunnel. Receivers can create or remove an IP tunnel dynamically using DTCP, thus enabling the network topology to be updated dynamically.

## V. Implementation of UDLR

*A. Method of implementation*

We implemented UDLR, and confirmed the appropriate operation of existing routing protocols. This chapter explains the implementation method on FreeBSD-2.2.6.

*B. Broadcast emulation*

*1) Outline*

The BCE layer is located in the middle of the IP and data link layers. The layer can be implemented either by the kernel method or by implementation into the user space. We implemented the BCE layer by the kernel method.

GRE (Generic Routing Encapsulation) which can encapsulate the packets of several protocols is used for the protocol of the IP tunnel to allow not only the IP datagram, which contains the route control information, but also the packet of ARP to be carried.

In FreeBSD, processing in the IP layer is achieved with the ip_input() and ip_output(), processing in the data link layer is achieved with the ether_input() and ether_output(), and the interface is controlled with the eth_intr() and eth_start(). The implementation processes BCE in the Receiver with gre_output(), and processes BCE in the Feed with gre_input().

*2) Receiver Processing*

IP datagram output from a routing program such as Gated is processed so that it is output from the interface on the UDL side via the ip_output() and ether_output(). Next, the gre_output() is called from the ether_output(). The gre_output() encapsulates the Ethernet frame to the GRE, and then calls another degree ip_output(). At this point, the IP address in the BDL interface of the Feed is set in the destination address of the IP datagram by which the GRE packet is carried. The route is selected so that IP datagram passed to the ip_output() is sent from the BDL interface because the destination address is in the BDL interface of the Feed this time.

*3) Feed Processing*

Upon receiving the GRE IP datagram output from the Receiver, the Feed processes it in the order of ether_input(), ip_input(), and gre_input(). In the gre_input(), the packet in the GRE is removed. In this case, IP datagrams that contain the route control information data are removed from the GRE packets. Here, another ip_input() call is made. Furthermore, for the packet to other Receivers, the broadcast packet or the multicast packet, the ether_output() is called with gre_input() and the IP datagram is sent to the UDL.

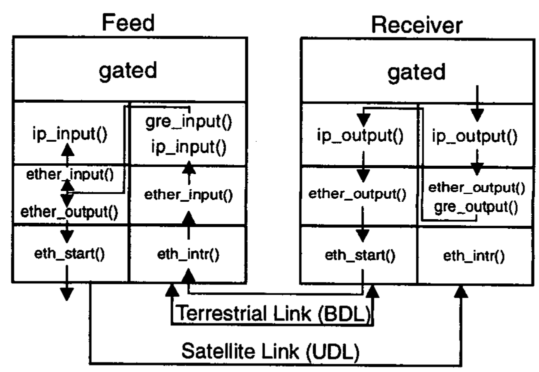



Fig. 5: Implementation of BCE layer

When the IP layer receives data, it is important that the UDL interface identifier be set in the parameter that indicates the interface from which the datagram entered. If the input interface is not correctly processed, Split-Horizon processing that uses RIP will not be performed properly. Moreover, it is necessary to prevent the Feed from sending ICMP redirects when an ICMP echo request is sent from one Receiver to another. When data must be sent to the UDL, it is necessary for TTL not to be decremented by 1.

*C. DTCP*

In the DTCP, the Feed regularly sends information (DTCP JOIN message) which the Receiver needs to establish the IP tunnel such as an IP address in the BDL interface of the Feed. In addition, the function of sending the DTCP LEAVE message to tell the Receiver that the UDL will be down and the function of processing the selection of sequence numbers when the Feed reboots are provided. The DTCP was implemented in the user space. This program simply sends the multicast packet provided with the Feed and Receiver.

## VI. EVALUATION OF UDLR

### A. *Evaluation method*

We tested for correct whether the route control by using an existing routing protocol to evaluate implementation of the Feed and Receiver. The evaluated system comprised one Feed, two Receivers, and a router (Fig. 6). In the experiment, we confirmed whether the broadcast emulation and DTCP were operating correctly, and whether each protocol performed the intended operation using RIP, RIP2, OSPF, and DVMRP routing protocols. Gated ver. 3.5.8 was used for the unicast routing to set the attribute of the route in detail, and mrouted ver. 3.8 was used for the multicast routing.

### B. *Broadcast emulation*

We first examined whether the broadcast emulation was achieved, and whether unicast, broadcast, and multicast packets were processed correctly. The following procedures were used for confirmation.

1. For unicasting

Unicasting was tested by using the ping application to check addressing from Receiver1 to the Feed and to Receiver2. ARP (the resolution of data link addressing) was executed prior to ping ICMP processing. Next, ICMP Echo and the Reply were transmitted. For transmissions addressed to the Feed, the Feed performed reception processing of ICMP Echo packets sent from Receiver1 and ICMP Echo Reply packets were sent in reply. On the other hand, the Feed performed decapsulation processing of ICMP Echo packets addressed to Receiver2, then forwarded those packets on the UDL. Receiver2 performed reception processing on these packets, then responded with an ICMP Echo Reply to Receiver1 via the Feed.

2. For broadcasting

As with broadcasting and unicasting, ICMP Echo Replies were returned from all nodes on the UDL subnet in response to ping execution directed to the UDL subnet broadcast.

3. For multicasting

Multicasting was confirmed using a home-made application. GRE packets output from Receiver1 to the UDL were processed by the Feed in the same manner as with broadcasting, then were output on the UDL. All nodes on the UDL subnet performed reception processing on these packets.

For broadcasting and multicasting, when the Receiver sends the packet, the packet is sent on the UDL via the Feed. In this case, the Receiver cannot receive its own packet output on the UDL, because the routing program cannot correctly process packets from a Receiver to itself. Therefore, the source MAC address of the packet on the UDL must be confirmed, and if it has been output by the Receiver, the Receiver must not receive the packet.

Moreover, when an IP datagram becomes larger than the MTU size during Receiver construction of the GRE packet for each packet, the IP datagram must be fragmented.

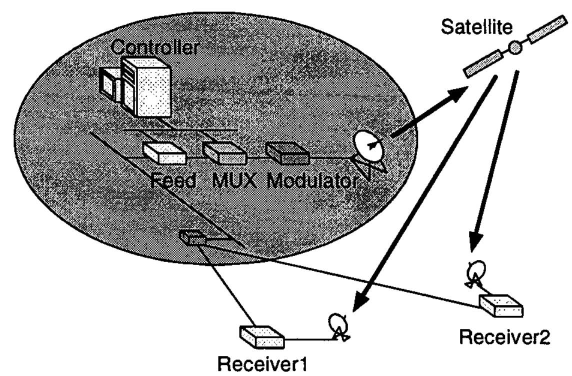



Fig. 6: Composition of system for evaluation

### C. *DTCP*

Processing of DTCP is intended to establish the IP tunnel from the Receiver to the Feed, and to invalidate the IP tunnel immediately when the Feed goes down or informs the Receiver that the Feed does not use UDL. To examine the routing protocols, data were sent up and down through the UDL to confirm that the DTCP was working correctly based on changes in the routing table of the router.

### D. *Routing protocol*

#### *1) Setting the route*

When UDL is used as a single network, our policy was to select it preferentially due to its high data transfer speed. On the other hand, traffic in the IP tunnel from the Receiver to the Feed allows only the passage of route control information in this direction, so that the usual data packet is set not to be sent. This decreases the load on the network. When the UDL is used in such a set up, the routing requirements in the Feed and Receiver must be correctly established in order to prevent unnecessary load demands on the network. For example, if the correct route is not selected such that UDL rather than BDL is selected in the IP tunnel from the Receiver to Feed, a loop is generated in the IP tunnel where GRE is processed. Thus, extreme care must be taken to prevent this. The following section describes the results of our tested routing protocols.

#### *2) RIP*

RIP is a Distance Vector routing protocol. Therefore, it is necessary to set the metric of an individual link appropriately. In order to make receivers recognize that UDL is not available, a large value was assigned to *metricout* in the UDL interface of the Feed. To prevent the Receiver from receiving route information from the Feed and other Receivers, the *noripin* attribute was set in the UDL interface, or a large value was assigned to *metricin.* Moreover, a small value was set to *metricout* in UDL interface.

*3) RIP2*

RIP2 uses multicasting to transmit and receive route control information. The configuration was same as that in the RIP except the version number.

*4) OSPF*

Because OSPF is a link state routing protocol, it is necessary to set the cost and the priority of the link in the attribute of each network. In setting the Feed, high router priority was set so as to become the Designated Router (DR). Conversely, the Receiver should be configured so as not to become the DR.

*5) DVMRP*

DVMRP is a Distance-Vector protocol. Since route selection was performed by RPF (Reverse Path Forwarding), the *metricout* on the UDL side was set low in the Feed and high in the Receiver, opposite to the settings for RIP.

## VII. Conclusions and Research Directions

In this article, we first evaluated the performance of the TCP/IP connection in networks comprising combined satellite and terrestrial links, and showed that high throughput can be achieved by having a large window size. We then introduced the UDLR and discussed the design and implementation of UDLR, as well as implementing BCE and DTCP. We then applied RIP, RIP2, OSPF and DVMRP in a network equipped with UDL, and confirmed that routing information is correctly exchanged between the Feed and Receiver.

This research and development have been performed in collaboration with the WIDE (Widely Integrated Distributed Environments) project in Japan. Furthermore, we are active participants in IETF and in the standardization of UDLR.

Our standardization of UDLR makes the assumption that the Feed and Receiver are routers. However, UDLR may also be achieved on a bridge, and the generalized system architecture can be realized if the function of UDLR and routing functions are separated. We implemented UDLR function on a bridge and confirmed that the routing protocols can be fully supported in the same way as that observed with the UDLR function on a router.

Some protocols, such as PIM-SM (Protocol Independent Multicast Sparse Mode) cannot be applied by the proposed IP tunneling approach. Consequently, a routing protocol that considers a one-way link should be developed. Furthermore, as multicasting is the key feature of satellite links, scalable routing protocols for multicasting should be developed.

## VIII. Acknowledgment

The authors would like to acknowledge Prof. Jun Murai of Keio University, General Chair of WIDE project and Mr. Hidetaka Izumiyama of NTT Satellite Communications Inc., a member of WIDE project, for insightful discussions.

## IX. References


[1] S. Hanks, "Generic Routing Encapsulation", Oct 1994. RFC 1701.

[2] S. Hanks, "Generic Routing Encapsulation over Ipv4 networks", Oct 1994. RFC 1702.

[3] E. Duros, W. Dabbous, H. Izumiyama, N. Fujii, "A Link Layer Tunneling Mechanism for Unidirectional Links", Mar 1998. I-D draft-ietf-udlr-lltunnel-00.txt (work in progress).

[4] V. Jacobson, R. Braden and D. Borman. "TCP Extensions for High Performance", May 1992. RFC 1323.

[5] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", January 1997. RFC 2001.

[6] W. Stevens, "TCP/IP Illustrated, Volume 2", Addison-Wesley, Reading, Massachusetts, 1994.

[7] R. Braden. "Requirement for Internet Hosts – Communication Layers", Oct 1989. RFC 1122.

[8] Mark Allman, Shawn Ostermann, "Multiple Data Connection FTP Extensions", Technical Report, Ohio University, Feb 1996.

[9] Mark Allman, Chris Hayes, Hans Kruse, Shawn Ostermann, "TCP Performance over Satellite Links", 5th International Conference on Telecommunication Systems, 1997.