USING MDP FOR TELEMETRY DATA TRANSFERS

Anirban Chakraborti
Klipsch School of Electrical Engineering
New Mexico State University

ABSTRACT

The current challenge has been to develop and adapt commercial Internet protocols for usage in space communications. Commercialized solutions, rather than Customized ones are cheaper, have low turnaround time and offer higher flexibility in deployment and operation. The focus of the study was to modify and develop UDP/IP based protocols commonly used in commercial Internet for reliable data transfers in space environment.

Multicast Dissemination Protocol was designed by Naval Research Laboratory to provide reliable multicast data and file transfer delivery on the top of general UDP/IP platform. It is very suited for bulk data transfer over the Internet. We have extended its usage in space channels and evaluated it as a solution to meet key challenges in space communications like high bit error rates and asymmetric channels. We have also tried to optimize the performance of the protocol in the terms of throughput, reliability, integrity and security of data. The evaluation test were carried on our Space to Ground Link Simulator which uses PPP to model point to point satellite links and correspond to low capacity systems as found in small satellite systems.

KEY WORDS
Multicasting, MDP, Space communication Protocols, Space Internet

INTRODUCTION

In an effort to study the entire spectrum of both connectionless and connection oriented transport layer protocols in space communications and with the Transmission Control Protocol (TCP) being already extensively researched has led to the investigation of User Datagram Protocol (UDP) based application layer protocols [1]. The focus is on evaluating the ones capable of meeting challenges of space environment, i.e. relatively high bit error rates, long delays, intermittent communication and propagation delays [2]. Traditionally TCP based protocols with their positive acknowledgement mechanism suffer limitations in satellite data transfers and hence UDP based protocols may have an advantage due to UDP's connectionless nature and may be more suitable in handling intermittent connections and packet drops. That is the reason we have directed our investigation into one of UDP based negative acknowledgement feedback protocol known as Multicast Dissemination Protocol (MDP) [3].

WHY MULTICAST DISEMMINATION PROTOCOL

MDP is an UDP based protocol capable of reliable data and file delivery transfers. It has advertised mechanisms to re-establish links after link outages. It doesn't have a slow-start mechanism or other flow controls unlike TCP based protocols. It can be used in both unicast and multicast mode .Its selective negative acknowledgement mechanisms made it viable to be evaluated in space channels because of very high bit error rates and long propagation delays which don't allow a positive acknowledgement scheme to work efficiently. The protocol is available in Linux and Windows environments to allow easier testing.

QUESTIONS TO BE EXPLORED

The objective of the paper is to explore and probe the following questions in the light of operational characteristics of MDP

- Can UDP-based Protocols like MDP be effectively used in small satellite type links for data file transport?
- How well connectionless UDP compare with connection oriented TCP in space environments.
- How well do negative acknowledgement work at the application layer in providing a reliable data and file delivery services.

MULTICAST DISSEMINATION PROTOCOL

The multicast dissemination protocol is designed to provide reliable multicast data and file delivery services on the top of generic UDP/IP transport. MDP is well suited for reliable multicast bulk transfer of data across a heterogeneous internetwork. Additionally, the protocol is adaptable to range of network environments including space channels. At its core, MDP is an efficient negative acknowledgement based reliable multicast protocol [4] that leverages erasure based coding in the ways to improve protocol efficiency and robusteness. The core MDP framework makes no design assumptions about network structure, hierarchy or reciprocal routing paths. MDP also includes adaptive end-to-end rate based congestion control mode that is to operate with competing flows. MDP also dynamically collects group timing information and uses it to further improve its data delivery efficiency in terms of latency, overhead and minimal redundant transmissions.

CONSTRAINTS IN SPACE ENVIRONMENT

The space environment poses challenges potentially different from the current ground-based wide area network links where TCP is normally used. Space links often have characteristics such as

- High Bit Error rates
- Longer Propogatinal Delays
- Intermittent Connectivity
- Asymmetric Channels

TCP is known to have limitations in network environments with long operational delays and high error rates. TCP does provide reliable transfer of data but it needs positive acknowledgement from the receiver for transfer of each data packet, which in turn reduces the throughput. It is also not very efficient in recovering lost links and handling intermittent connections. Finally it's slow start mechanism is not ideally suited for space to ground data transfer where there are few competing flows. TCP treats channel errors as congestion hence it takes longer time to complete the transmission than for a protocol based on UDP where the application provides for a reliable feedback loop.

UDP, on the other hand send the data packet with checksum attached to its lower layer and forgets about it. The application needs to provide for a feedback mechanism for error detection and correction for reliable data transfer.

TEST ENVIRONMENT

The Space to ground link simulator (SGLS) channel simulator is used to perform the error generation and link delay used to test the protocol suite performance. In this section, we describe the simulator and the standard tools for gathering data to measure the performance. The SGLS has been developed at NMSU to model space channel characteristics experienced in transmitting data. The simulator is described fully in [4], [5], and [6]. Basically, the SGLS configuration allows the user to configure the simulated channel to

1. Allow for simultaneous bi-directional data flow (forward and return channels),
2. Allow user-selectable error rates and statistical descriptions of the channel,
3. Allow different data rates on the forward and return links as would be found in satellite links, E.g. 2400-baud forward, 115200- baud return.
4. Provide for a simulated delay up to 5 seconds on each link.
5. Allow for forward and return channel link disruptions and select the duration of the disruption along with the time interval at which link outage takes place from start of data transmission.

The error generation methodology used in the simulator is based performing an Exclusive-Or operation on the protocol data stream with a pre-stored channel error vector having the error statistics desired by the user. The channel error vector is typically on the order of 1 Mbit in length and repeats when the end of the vector is reached. The vector is selected by the user as part of the simulation process from a list of available error vectors with differing statistical characteristics.

The channel error delays are generated using a software buffer to store the data until the desired elapsed time to model the link propagation delay has passed. The SGLS utilizes the LabVIEW programming language to control data flow through the simulator, mix the base band data stream with the user-selected error vector, and provide for the user-selectable link delay value. The hardware configuration is illustrated in Figure 1. The LabVIEW software is run as an application on each of the SGLS computers. Typically, the LabVIEW modules are the only applications software running on the computers. This configuration was developed to model point-to-point satellite links in its current configuration. The bandwidth delay product for the system under an 115200bps symmetric link with no imposed channel delay is 671bytes. As a comparison, a T-1 line crossing the

United States has an estimated bandwidth delay product of 11,580 bytes [7]. Therefore, this simulator corresponds to a relatively low capacity system.

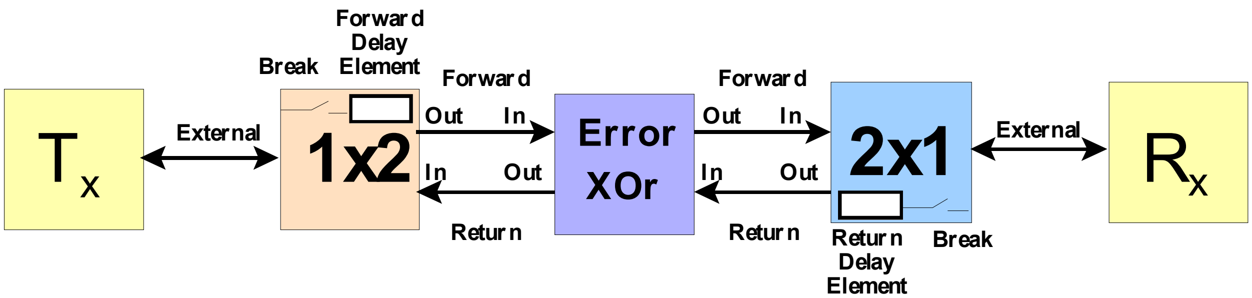

Figure 1-Block diagram of space channel simulator

TEST METRICS AND DATA PAYLOAD

Our design space for MDP consists of optimizing protocol performance in satellite to ground bulk data transfers. Hence we have only evaluated the parameters, which affect MDP at that level and tried to reach a solution.

Transmission lifetime is an important metric that used by the protocol. It is defined as the time taken by a file to transfer under zero error (0 Bit Error Rate [11]) and no delay conditions. The time is measured at the server and under the above conditions it is the same at the receiver too. This metric has been used to evaluate the effect of delays and link cuts on the duration of the file transfer.

The measurement and analysis made are based on the fact that in each of the end-to-end file transfers under varying error and link conditions, the file integrity was never compromised and only the duration of transfer varied accordingly.

The payload for our tests was files and directories and transfer characteristics for each were observed. The directories consisted of files with individual characteristics known and hence attention was paid towards their behavior as collective whole. The individual files tested were of 100KB, 1MB and 10MB each.

OPERATIONAL CHARACTERISTICS OF MDP

OPERATING SYSTEM REQUIREMENTS

MDP is compatible and interoperable with Windows and variants of Linux operating system. To illustrate this point we have interchangeably used Windows98, Redhat Linux and Suse Linux on our logical ground station and logical satellite. The performance in terms of throughput doesn't show any perceptible variance irrespective of whether the transmission is from linux host to windows receiver or linux to Linux. This indicates that MDP is not affected by the underlying operating system. This property is important because of the fact that MDP does not need any custom built operating system and can work any with operating system.

OPTIMUM TRANSMISSION RATE

The Transmission rate parameter of MDP can be set according to the channel conditions, the maximum limited only be the channel capacity and the hardware. Its default is set at 64kbps, but we have set it up to10Mbps on the Ethernet. On our simulator though the channel speed is set at 115200 bps, any rate set above 88,000 bps leads to higher throughput time than at 88,000 bps itself. It has been observed that any speed above 88,000 causes interface loss on the channel and as a result the file transfer duration increases.

THROUGHPUT

Throughput as the measure of file transfer duration under varying conditions of Bit Error Rate is presented. We have specifically calculated the transmission times of variety of file sizes. We observed that MDP has a consistent data throughput and the achieved data transfer rate is varies within 90% of the set transmission rate of 88,000 bps. The channel capacity is at 115200bps.

Table 1. Transfer Duration at Different Bit Error Rates

File Size	Bit Error Rate	Transfer Duration (sec)	Achieved Throughput (Bits per sec)
100KB	0	10	81,290
100KB	.000001	11	74,427
100KB	.0001	36	22,755
1MB	0	96	87,381
1MB	.000001	99	84,733
1MB	.0001	323	25,970
10MB	0	956	87,746
10MB	.000001	1027	81,680
10MB	.0001	3225	26,011

In a zero error environment the transfer duration of MDP is slightly higher than TCP based protocols. But in case of high bit error rates like 10^{-4} MDP is able to complete the data transfer whereas TCP based protocols like File Transfer Protocol (FTP) ceases transmission. The fact that MDP is capable of maintaining data transmission even at relatively high error rates makes it suitable for space applications.

PROPOGATION DELAYS

Tests were conducted to ascertain the effect of propagation delays on the file transfer duration for MDP. We primarily concerned ourselves with 5 sec bi-directional delay (longest delay obtainable on our simulator) at zero Bit Error rate to analyze the effect of propagation delay on the throughput without any channel errors being simulated.

Table 2.Effect of 5 sec propagation delay on transfer duration.

File Size	Bit Error Rate	Propagation Delay (sec)	Transfer time (sec)
100KB	0	5	14
1MB	0	5	136
10MB	0	5	1387

TCP works on the positive acknowledge mechanism that it waits for receiving acknowledgement for the last block of data sent before it continues transmission. If the acknowledgement is not received before the timeout interval, it retransmits the last block again and again until it is acknowledged. Under high propagation delay of 5 sec TCP has to wait for longtime for the acknowledgement for each block of data and thus the data transfer times are higher than that of MDP which being based on negative acknowledgement, does not need to acknowledge every block received successfully.

UPDATE OF FILES

At the start of transmission of a file MDP sends a packet known as MDP_FILE_INFO concerning the particulars of the to be transmitted file including it's size. The file may not be updated until its transmission is complete or it is rendered useless due to format errors. If the size of the file is increased during transmission, up to original file size is transmitted and if its decreased during the transmission MDP assumes the file to be corrupted and discards the transmitted file on the receiver.

PARITY BASED REPAIRS

By default MDP sends no parity packets with the data and follows a reactive policy, that is, parity packets are automatically inserted during error correcting sessions and not in the original transmission. Transmissions in high error environments generate large number of retransmission requests. Hence sometimes it pays to follow a proactive policy, deliberately inserting parity packets in per MDP coding block and thus number of NACKS are generated are few along with fewer retransmissions. It is particularly useful where the channel is asymmetric though it adds to the overhead and leads to higher transfer duration. In few cases it actually decreases the transfer duration than it would be without the parity packets. TCP based protocol like FTP ceases data transfer at high error environments.

To illustrate our point we have chosen an mp3 file of 3659913 bits and tested it in a .0001 BER environment

Table 3. Effect of parity packets on transfer duration in high error environment

# Parity Packets	Transfer Duration (sec)
0	2395
20	2373
30	2292
60	1444

This validates our point that proactively inserting parity packets in MDP coding blocks helps to achieve lower file transfer duration and hence higher throughput than following a reactive policy in especially high error environments.

.

INTERMITTENT LINK CUTS

Links cut satellite communications can be intermittent, i.e. happen without warning or may be due to the time the satellite is not seen above the ground station. The ability to handle intermittent cuts and restore them is an important characteristic of any space communication protocol.

Intermittent link cuts occur due to atmospheric disturbances and may last for few seconds to several minutes. These cuts may occur in between the transfer of a file. We have studied the effect of these cuts on 1 MB and 10 MB files.

Table 4. Effect of Intermittent Link Cuts on Transfer Duration

File Size	%	Gap Duration (sec)	Total transmission time (sec)
1MB	25	15	110
1MB	50	15	110
1MB	75	15	110
1MB	25	30	126
1MB	50	25	120
10MB	50	120	1076

Here % denotes the total percent of the file transferred when the link cut occurred. Total transmission time is the total time taking for the end-to-end file transfer and includes the gap duration time.

The following hypothesis was validated:

If the transmission lifetime of a file is *t sec* and link is cut when *% p* of the transmission is complete or *z sec* of transmission have taken place and link cut duration was *l sec* where $(l+z)<t$ *sec* then total time taken for complete transmission would be about $(t+l)$ *sec* and is not dependent on the *%p*.

For directories is too, if there is a link cut and link is restored within the lifetime of the server transmission, the time taken for total transfer is about the usual time + link cut duration. This was expected because individual files behave in that way**.**

SCHEDULED LINK CUTS

Scheduled links cuts are the duration during which the satellite is not seen over the ground station and hence no data transfer takes place. We have categorized scheduled link cuts as link cuts which of much larger duration than the transmission lifetime of the files**.** In such cases there is certain amount of resynchronization time required after the link is restored and data transfer commences again. The time taken for resync is independent of the transmission life time of the file and % of the file transmitted when the link cut occurred and also happens to in fairly constant range for link cut durations of 15mins to 1.5 hrs.

For 1MB files

Table 5. Resynchronization time for Scheduled Link Cuts

%	Gap Duration	Resynchronization time (sec)
25	900	120-159
25	1800	105-158
25	5400	150
50	5400	150-165

.

ASYMMETRIC CHANNELS

Unlike TCP based protocols, which rely on positive acknowledgements MDP can work in complete asymmetric channels. By asymmetric channels we mean that only the forward channel is operational and the return channel is closed Thus in absence of any feedback from the client, the server continues on transmitting in a best effort service fashion until all the files and directory have been transmitted. When the return link is again available the client may send aggregated NACKS for the transmitted files and server goes into repair session and completes the data transfer.

CONCLUSION

The presence of negative acknowledgement scheme at the application layer and using connectionless UDP at the transport gives it advantages in link management and reliable transfers in high error environments. The studies shows that MDP has desirable features, which makes it attractive for bulk data transfer in asymmetric and space Internet work applications. Proactive FRC based repairing improves protocol performance in very high bit error rate scenarios. The ability of the server to hold state information is very useful in handling link cuts and initiate repair sessions long after initial transmission is over. It is capable of operating independent of the network structure which makes it attractive for real time data transfer in space like environments like the wireless systems because of high error rates link disruptions and frequent packet drops.

ACKNOWLEDGEMENTS

Dr.Stephen Horan, Thesis advisor and Mr. Sandeep Muddassani, Teammate

REFERENCES

1. S.Deering."Host Extensions for IP Multicasting". Internet RFC 1112,August 1989
2. D.Grossink and J.Macker, "Reliable Multicast and Integrated Parity Retransmission with Channel Estimation." IEEE GLOBECOM 98,1998
3. Macker, J. and Adamson, Brian R.,"The Multicast Dissemination Protocol", Naval Research laboratory, October 1999
4. Horan, S and Wang, R., "Design of a Channel Error simulator using Virtual Instrumentation Techniques for Initial testing of TCP/IP and SCPS protocols", NMSU-ECE-99-002, Las cruces1st April 99.
5. Horan, S and Wang, R, "Enhancement of NMSU channel simulator to Provide unbalanced forward and return Transmission rates", NMSU-ECE-99-003, Las cruces, April 99.
6. Horan, S and Wang, R, "Enhancement of NMSU channel simulator to Provide user selectable link delays", NMSU-ECE-00-001, Las Cruces, May 2000.
7. Stevens, W., TCP/IP Illustrated Vol 1,chapter 20,Addison Wessely, Reading, MA, 1994,pg .289.
8. Goncalves, Marcus and Niles, Kitty,"Implementing IP Multicast in Different Network Infrastructure", IP Multicasting, concepts and applications, McGrawHill, New York, New York state, 1999,459-470.
9. Tanenbaum, Andrew S., Computer Networks, Prentice Hall PTR, Upper Saddle River, NJ, 1989.
10. Sklar, Bernard, Digital Communications, Prentice Hall, Upper Saddle River, NJ, 2001,pg 311