

IMPACT OF STATUS RETRY TIMER VALUES ON MFTP PERFORMANCE OVER BGAN SATELLITE MULTICAST

Shyamalie Thilakawardana¹, Guray Acar¹, Ben Hale² and Rahim Tafazolli¹

¹Mobile Communications Research Group, Centre for Communications Systems Research, University of Surrey, Guildford, UK, ²Inmarsat UK

ABSTRACT: MFTP is a multicast transport protocol optimized for reliable file transfers. MFTP is rate based, which means that the transmission rate can be configured. By limiting the transmission rate one can reserve bandwidth for other protocols. MFTP uses a NACK based error correction scheme where the server is almost constantly sending data and very seldom has to wait for acknowledgements from the clients. MFTP sends a file in passes. The first pass represents the complete file while subsequent passes represent data identified as not received by clients in NACK messages. This makes MFTP specially suited for transmissions over links with long delays, for example satellite links. This paper analyses MFTP multicast performance over Inmarsat BGAN multicast satellite system. The full interaction between the MFTP server and MFTP clients deployed over the Inmarsat BGAN down link multicast delivery mechanism along with BGAN uplink access mechanism is considered. The performance evaluation is based on a BGAN multicast system level simulator. The analyses aims to evaluate and optimize MFTP performance in Inmarsat BGAN system in terms of file transfer delay and system throughput as a function of available capacity, client population size, data product size, channel error characteristics, and MFTP protocol settings.

I. INTRODUCTION

Multicast transport protocols have to simultaneously deal with different channel conditions that are experienced by several receivers. This is in contrast to the requirements on a unicast transport protocol. The most significant difference from a server point of view is the large amount of returning traffic packets or status updates that a multicast transport protocol has to deal with. When using a unicast transport protocol the back traffic represents a small part of the communication. However, a multicast transport protocol receives back traffic from a potentially large number of clients and the amount of back traffic messages can be significant. Thus, design of multicast transport protocols needs to minimise the amount of back traffic. This is the reason behind why most multicast transport protocols use negative acknowledgements (NACKs) instead of positive acknowledgements (ACKs). When negative acknowledgements are used, the receivers inform the sender only about which packets were lost during transmission (A loss is detected when there is a missing sequence number in the receive traffic stream). Since the number of packets lost is normally a low percentage e.g. 10^{-3} , the back traffic becomes much smaller when NACKs are used compared with the back traffic caused by ACKs.

There are two types of multicast transport protocols: reliable and unreliable. Reliable protocols provide an error free transfer and can be used for example to transfer files to a number of clients. Multicast File Transfer Protocol (MFTP) is an example of a reliable multicast transport protocol. Unreliable protocols are mainly used by real time applications, which are more sensitive to delays than packet errors. RTP, RTCP, RTSP and RSVP are examples of unreliable multicast protocols. Note that these protocols are not primarily multicast protocols. They can be used over multicast connections but their main purpose is to provide different real-time services (as in the case of RTP, RTCP, RTSP) and resource allocation functionalities as in the case of RSVP ([3], [4], [6], [8]).

This work investigates the MFTP performance over an Inmarsat BGAN multicast system. Based on current and expected future usage potential, we have selected two applications of interest for Inmarsat BGAN multicast. They are (a) Content distribution and (b) Net-radio type services. The main difference of interest to our study between these two applications is that the content distribution application will in most cases require guaranteed delivery (reliability and loose or non-real-time delay requirements), whereas net-radio requires real-time transmission with very stringent requirements in terms of bandwidth and delay but delivery need not be guaranteed. MFTP consists of two parts; the one part is the Multicast Control Protocol which deals with the multicast group management and the other part is the Multicast Data Protocol (MDP) which handles the data transfer within MFTP. In our study only the MDP is considered. Section II of this document gives an overview of MFTP with emphasis on simulation analysis of the Multicast Data Protocol (MDP) part of MFTP. Section III describes the simulation scenarios selected for the investigation of MFTP over BGAN multicast. Section IV presents the performance evaluation of MFTP over BGAN multicast. Conclusions of this study can be found in Section V.

II. MFTP OVERVIEW

MFTP is a multicast transport protocol optimized for reliable multicast file transfers. The protocol has been designed by StarBurst Corp. and is specified in an Internet-draft ([1], [2]). This section provides an overview of MFTP. MFTP uses a NACK-based error correction scheme where the server is almost constantly sending data and very seldom has to stop transmission to wait for feedback from the clients. This makes MFTP specially suited for transmissions over links with long delays such as satellite links. However, this behaviour also means that MFTP is unsuitable for real time applications.

MFTP runs over UDP and is able to manage a large number of simultaneous receivers/clients. Experiments have been performed at StarBurst to emulate up to 10,000 receivers in one closed group transmission and no significance performance degradation was observed [1]. MFTP consists of two protocols: MCP and MDP. MCP (Multicast Control Protocol) is used for managing multicast groups and MDP (Multicast Data Protocol) handles the actual data transfer [Figure 1].

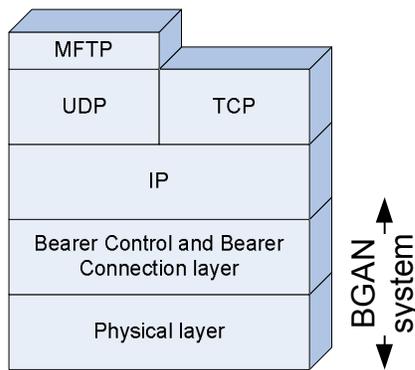


Figure 1 Operation of MFTP over BGAN system

There are three basic concepts within MFTP; they are MFTP pass, MFTP block, and Data Transmission Unit (DTU). Before the file is transmitted to the clients (multicast receivers) by multicast delivery it is divided into a number of MFTP blocks. The amount of bytes in a MFTP block can be configured by the MFTP server. Each block is comprised of number of DTUs and the DTU size can be configured in terms of bytes. In the case of MFTP over BGAN multicast, DTU size is specified as 526 bytes [Figure 2].

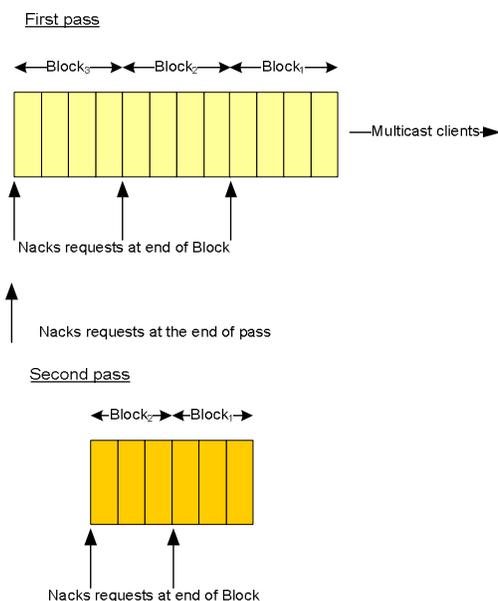


Figure 2 MFTP Pass, MFTP Block and Data Transfer Unit

A. Data transfer in MFTP

The file is transferred in a sequence of passes. In the first pass, the whole file is transferred. After each block, the unsuccessful clients may send a bit map NACK message to the server containing information about which DTUs in the block that were not successfully received. Within the bit map NACK, one bit represents one DTU and the number of bits is equal to the number of DTUs in an MFTP block. For example in [Figure 3] the multicast file is divided into three MFTP blocks with each block carrying four DTUs. Therefore the bit map NACK for each MFTP block is four bits long. In each NACK by keeping the value of bit as 1 (corresponding DTU in error) or 0 (corresponding DTU not in error) the DTUs which are in error can be identified at the MFTP server. After the first pass the MFTP server waits for a predetermined waiting time or a Status Retry Timer (SRT) value before sending the next pass. The NACKs received before the SRT expires are used to construct the corresponding DTUs of the next pass. Once NACKs are received at the server using the OR operation on received bitmap NACKs the relevant DTUs for the second pass are identified.

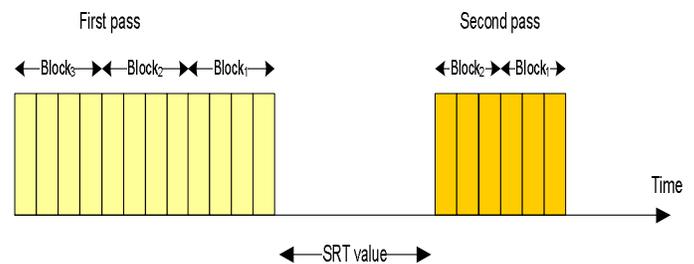


Figure 3 Status Retry Timer in MFTP

Therefore in the second pass the server transfers the DTUs that were unsuccessfully received in the first pass. The DTUs that are retransmitted in the second pass belong to the same block as in the first pass. In the first pass each block has a fixed length. In the subsequent passes the size of a block depends on how many DTUs in that block had to be retransmitted.

The server keeps sending the file in passes until all clients have received an error free copy of the file [Figure 3]. The performance of MFTP is highly dependent on the Status Retry Timer or SRT which is an external parameter of the protocol [Figure 3]. Once the SRT expires the MFTP server starts the next pass by transmitting the repair blocks of the received NACKs. Therefore SRT value needs to be long enough to collect/capture maximum number of NACKs within a given duration. At the same time unnecessarily longer SRT values may result in having longer file transfer delays. In other words longer the SRT timer larger the amount of collected NACKs and thus resulting bigger pass sizes. Also at the same time if SRT timer is unnecessarily longer than the required duration to collect almost all the NACKs this will lead to longer file transfer delays. Therefore it is important to

come up with the preferred values for SRT timer for efficient performance. The SRT timer variations needs to look into different multicast group sizes (number of clients), different file sizes (ranging from Kbytes to Mbytes) and also the amount of contention slots allocated in the return link for NACK transmission.

III. SIMULATION SCENARIO

This section describes the main evaluation scenarios defined to study the MFTP performance over BGAN multicast. These scenarios are based on varying SRT timers, PER, and multicast group sizes.

In simulation analyses two BGAN bearer types are selected on the return link or uplink transmission. These are R5T1X-L1 and R20T1X-R bearers, where R5T1X generates 5-ms contention slots and R20T1X generates 20-ms reservation slots. Return link frames are of 80ms, each consisting of contention and reservation slots according to the return schedule. For instance, according to (4C, 3R) allocation pattern, a return schedule 80-ms return frame consists of four 5-ms contention slots (C slots) and three 20-ms reservation slots (R slots). Only one reservation slot (R slot) is needed to send an MFTP NACK. The reservation of R slots is performed via sending the requests via the contention slots (C slots) in the return schedule.

In the forward link or downlink, F80T4.5X-R bearer type is selected. F80T4.5X-R carries 375 bytes of data in each 10ms FEC block which results a forward link data rate of 300kbts/s. There are 8 FEC blocks in each 80ms forward frames.

Figure 4 illustrates an example of IP data encapsulation in MFTP over BGAN multicast. MFTP DTU size is considered as 526 bytes. At each layer MFTP/UDP/IP overhead of total 50 bytes are added which is being transmitted over BGAN. As seen, one IP packet (576 bytes = 526 bytes IP size + 50 bytes of overhead) is segmented over three BGAN PDUs. Each PDU payload is 251bytes, 108bytes and 217bytes consecutively. Since IP segment is carried in multiple PDUs the first PDU carries 9 bytes BGAN overhead and the rest of the PDUs carry 7 bytes of BGAN overhead. Since each FEC block carry only 375 bytes the first FEC block carries only two PDU segments.

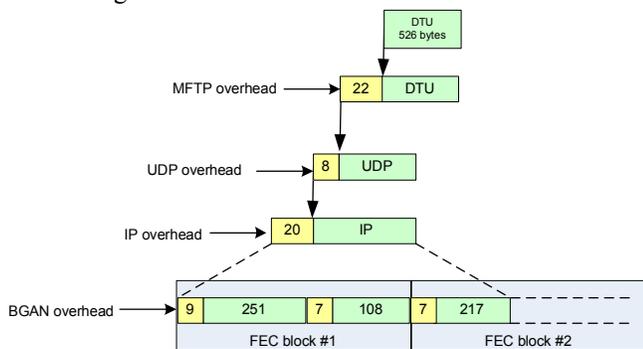


Figure 4 Data Transfer Mapping between MFTP over BGAN

Within the simulation scenario 1Mbytes multicast file size is transmitted over the BGAN forward link of data rate 300kbts/s. Three different multicast group sizes are considered with PER (Packet Error Rate) values of 0.001 and 0.005. During the first pass multicast receivers are expect to send NACKs once they received the whole file (at the end of the pass) and whereas in all other passes the receivers send NACK once at the end of the MFTP block.

A. Variations in SRT timer

In the following section the results are presented in the case of performance analysis based on variable SRT timer. The minimum practical SRT timer value is considered as the addition of MFTP block transmission time and the RTT (round trip time). This allows NACKs that do not suffer any collisions to reach the server. In the case of a downlink bearer of 300kbts/s data rate, 1Mbyte file size, DTU size of 526 bytes and MFTP block size of 488 DTUs, the MFTP block transmission time is 6.845s. The round trip time is considered as 1second. (The round-trip propagation delay plus packet encoding/decoding delays in BGAN system is 900 ms. In addition, we assumed a 100 ms processing delay at BGAN RNC). Thus, the minimum SRT timer value is assumed to be 7.845s. The actual time a NACK takes to reach the server will depend on the number of retransmissions caused by collision on the contention channel, if any. Each collision will induce a delay of approximately a RTT. SRT time is equal to the summation of MFTP block transmission time and $n*RTT$ value. The value of n is an integer. Simulations have been executed with the following values of n : 1, 2, 4, 6 and 10. The resulting SRT timer variation in (Figure 5 to Figure 10) along the x axis is as mentioned in [Table 1]. This results in a maximum SRT timer value of 16.845s. The packet error rate is varied from 0.001 and 0.005 with multicast group sizes of 20, 40 and 80 users [Table 2].

Table 1 SRT timer variation

n value	SRT Timer value (seconds)
1	7.845
2	8.845
4	10.845
6	12.845
10	16.845

Table 2 Scenario variables

Parameter	Value
Multicast group size	20,40,60
File size	1Mbytes, DTU size of 526 bytes, MFTP block size of 488 DTUs
PER	0.001, 0.005
SRT timer variation	MFTP block transmission time + (1,2,4,6,10)*RTT

IV. SIMULATION RESULTS AND PERFORMANCE ANALYSIS

Figure 5 presents the file transfer delay variations for varying SRT timer values for the case of (PER = 0.001, Group size = 20). Along with the average file transfer delay, average value + standard deviation and the average - standard deviation values are also presented. It can be decided that the case of small multicast group size (in this case 20 clients) and with PER of 10^{-3} , the best file transfer value is given at the optimum SRT timer of 10.845 seconds (SRT value = MFTP block transmission time + 4RTT). At this point file transfer delay is 54.1221seconds. Once the SRT value is increased beyond this point, the file transfer delay also increases due to idle periods of waiting between passes.

Main reason for having lower file transfer delays at optimum SRT values are two folds. The first reason is for sufficiently longer SRT values the number of NACKs collected at the end of the pass gets increased, thus resulting larger pass sizes in the next pass. These NACKs consist of NACK responses for the current pass and delayed NACK responses of the earlier passes. This may reduce the number of passes needed for complete file transfer. Reducing the number of passes means less occurrences of waiting time for SRT to expire as well as lower file transfer times. This is shown by the comparison of number of passes need to complete the file transfer is each case of varying SRT values.

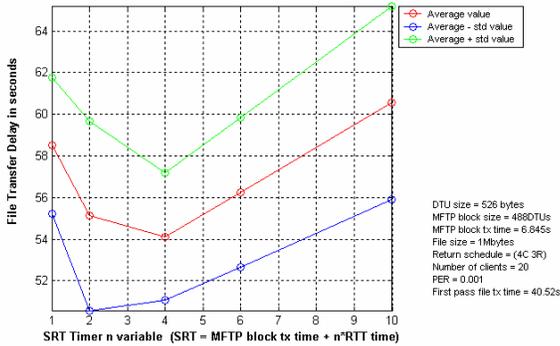


Figure 5 File Transfer Delay Vs SRT variation

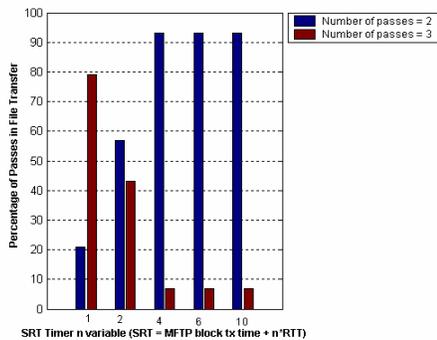


Figure 6 Percentage of passes for complete file transfer

Figure 6 shows the percentage of passes needs for complete file transfer in a data set of 100 samples of each SRT value.

As shown for the SRT value of 7.845 seconds (when $n = 1$), 79% of the samples took 3 passes for complete file transfer with only 21% samples completing in 2 passes. In the case of $n = 2$ only 43% of samples took 3 passes for complete file transfer with 57% completing in 2 passes. In the case of $n = 4$ and after words ($n = 6, n = 10$) only 7% of the samples took 3 passes for complete file transfer with 93% finishing file transfer in 2 passes. Thus at SRT value is at optimum at $n = 4$. Once SRT is increased beyond this point ($n > 4$) this does not have any impact on the number of passes. The file transfer time increase purely due to the increase of SRT value thus making the MFTP server wait longer to start the next pass. For this scenario (PER = 0.001, Group size = 20) minimum number of passes for complete file transfer is two passes whereas the maximum number of passes for complete file transfer is three passes.

Figure 7 presents the file transfer delay variations for varying SRT timer values for the case of (PER = 0.005, Group size = 40). Along the average file transfer delay the upper bound and the lower bound of the values is also presented. It can be decided that for the case of medium group size of 40 clients with PER of 0.005, the best file transfer value is given at the optimum SRT timer of 10.845 seconds where SRT value equals to MFTP block transmission time + 4RTT. At this point file transfer delay is 81.065 seconds. Once the SRT value is increased beyond this point file transfer delay also get increased.

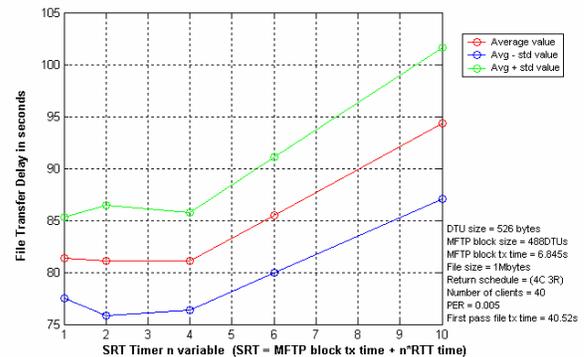


Figure 7 File Transfer Delay Vs SRT variation

Figure 8 shows with the comparison of number of passes needed to complete the file transfer is each case of varying SRT values. The percentage of passes needs for complete file transfer is evaluated from a data set of 100 simulation runs for each SRT value. It can be seen that higher the SRT values lesser number of passes. For SRT values when $n = 1$ and $n = 2$, each file transfer is a mix of 3, 4 and 5 passes. When $n = 1$, 7% of the samples complete in 3 passes, 79% completed in 4 passes and the rest (14%) took 5 passes to complete the file transfer. For the case of $n = 2$, 36% competed in 3 passes, 57% completed in 4 passes and only 7% went for 5 passes. In the case of $n = 4, n = 5$ and $n = 10$, 79% completed in 3 passes and 21% completed in 4 passes with none of the samples going for 5 passes. Once the SRT timer is longer ($n = 4, n = 6, n = 10$) the number of passes for complete file transfer decreases with

larger pass sizes in consecutive passes. For shorter SRT timer values ($n=1, n=2$) the consecutive pass sizes are smaller with resulting higher number of passes for file transfer completion. For this scenario (PER = 0.005, Group size = 40) minimum number of passes for complete file transfer is three passes whereas the maximum number of passes for complete file transfer is five passes.

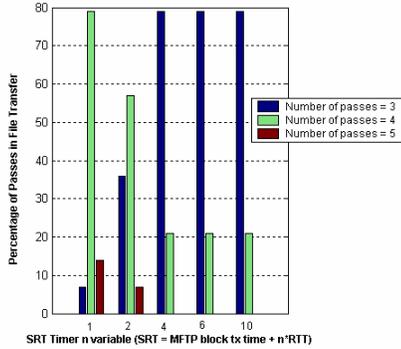


Figure 8 Percentage of passes for complete file transfer

Figure 9 presents the file transfer delay variations for varying SRT timer values for the case of (PER = 0.005, Group size = 80). Along the average file transfer delay the upper bound and the lower bound of the values is also presented. It can be decided that the case of larger multicast group size (in this case 80 clients) and with PER of 0.005, the best file transfer value is given at the optimum SRT timer of 10.845 seconds (SRT value = MFTP block transmission time + 4RTT). At this point file transfer delay is 91.0157seconds. Once the SRT value is increased beyond this point file transfer delay also get increased.

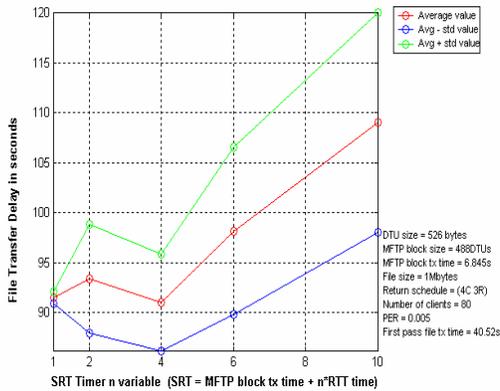


Figure 9 File Transfer Delay Vs SRT variation

Figure 10 shows the comparison of number of passes needed to complete the file transfer in each case of varying SRT values. The percentage of passes needs for complete file transfer is evaluated from a data set of 100 simulation runs for each SRT value. In the case of short SRT (SRT value of 7.845seconds, when $n=1$) all the samples completed file transfer in 4 passes. For longer SRT value (when $n=2$) only 64% completed file transfer in 4 passes, with 22% taking 3 passes and the rest 14% going for 5 passes for file completion.

Once the SRT timer is longer ($n=4, n=6, n=10$) the minimum number of passes for complete file transfer decreases (for $n=4$, approximately 78% of runs finishes in 3 passes where as for $n=6$, that is 64% and for $n=10$ only 57% of the simulations finished file transfer in 3 passes). For shorter SRT timer values ($n=1, n=2$) the consecutive pass sizes are smaller with resulting higher number of passes for file transfer completion. For this scenario (PER = 0.005, Group size = 40) minimum number of passes for complete file transfer is three passes whereas the maximum number of passes for complete file transfer is five passes.

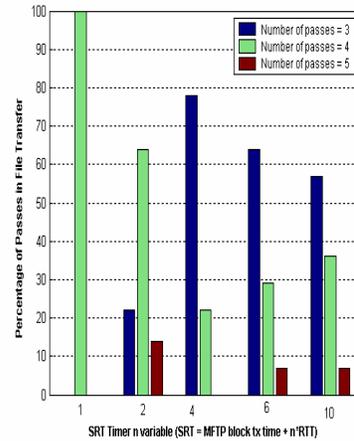


Figure 10 Percentage of passes for complete file transfer

V. CONCLUSIONS

Simulations were executed with various SRT timer values for performance comparison. RTT is assumed as 1second throughout the simulation study. SRT time is equal to the summation of MFTP block transmission time and $n*RTT$ value. The value of n is an integer. The value of n is varied as 1, 2, 4, 6 and 10. For the case of varying SRT values for different multicast group sizes reveal following findings. It can be decided that in the case of small multicast group size of 20 clients with PER = 0.001, medium group size of 40 clients with PER = 0.005 and larger multicast group size of 80 clients with PER = 0.005 the best file transfer value is given at the optimum SRT timer value of 10.845 seconds. For the case of 20 clients file transfer delay is 54.1221seconds whereas for the case of 40 clients the file transfer delay is 81.065 seconds and once the multicast group size is increased to 80 clients file transfer delay is 91.0157seconds. Once the SRT value is increased beyond this point file transfer delay also get increased. Therefore it is necessary to fine tune SRT values for better file transfer delay performance. Having lower file transfer delays at optimum SRT values are two folds. The first reason is that sufficiently longer SRT values increase the likelihood of a greater percentage of the generated NACKs being collected before the next pass, thus resulting larger pass sizes in the next pass. This can reduce the number of passes needed for complete file transfer. Reducing the number of passes means less waiting time for

SRT to expire as well as lower file transfer times. Also sufficiently longer SRT timers have more time to collect delayed NACKs of the earlier passes and construct repair blocks including DTUs signalled by delayed NACKs in the next pass. For the case of 20 users minimum number of passes for complete file transfer is two passes whereas the maximum number of passes for complete file transfer is three passes. For the case of 40 and 80 users the minimum number of passes for complete file transfer is three passes and the maximum number of passes for complete file transfer is five passes.

VI. REFERENCES

- [1] "Starburst Multicast File Transfer Protocol (MFTP) Specification", K. Miller, K. Robertson, A. Tweedly and M. White, Internet draft, (<ftp://ftp.ietf.org/internet-drafts/draft-miller-mftp-spec-03.txt>), July 1998
- [2] "Starburst MFTP: An Efficient, Scalable Method for Distributing Information Using IP Multicast", Starburst Technologies.
- [3] S. Paul and K. Sabnani, "Reliable Multicast Transport Protocol (RMTP)", IEEE J. Selected Areas of Communications, Vol. 15, April 1997
- [4] M. Koyabe and G. Fairhurst, "Reliable Multicast via Satellite: A Comparison Survey and Taxonomy". International Journal of Satellite Communications, Vol. 19, Issue 1, January/February 2001, pp. 3-28.
- [5] I. F. Akyildiz, J. Fang, "TCP-Peachtree: A Multicast Transport Protocol for Satellite IP Networks", IEEE Journal on Selected Areas in Communications, Vol. 22, No. 2, February 2004.
- [6] C. K. Miller, "Reliable Multicast Protocols: A Practical View," Local Computer Networks, 1997, Proc. 22nd Annual Conference, pp. 369-378, 1997.
- [7] C. Wang and V.C.M. Leung, "Performance Evaluations of SRMTP for Reliable Multicasting over Satellite Networks", in Proc. IEEE WCNC'03, New Orleans, LA, Mar. 2003
- [8] I. F. Akyildiz, G. Morabito, and S. Palazzo, "TCP-Peach: A New Congestion Control Scheme for Satellite IP Networks", IEEE/ACM Transactions on Networking, vol. 9, no. 3, June 2001.
- [9] S. Thilakawardana, G. Acar, A. Widiawan, R. Tafazolli "Transport Protocols for Reliability in Satellite Multicast", 23rd AIAA International Communications Satellite Systems Conference, September 2005.
- [10] Guray Acar, S. Thilakawardana, Rahim Tafazolli, and Barry Evans "Performance of Multicast File Transfer Protocol (MFTP) over geostationary satellite systems with DAMA uplinks" 24th AIAA International Communications Satellite Systems Conference, 2006.