SCALABLE AND ROBUST RELIABLE MULTICAST FOR SATELLITE NETWORKS

Kyriakos Manousakis
John S. Baras

Center for Satellite and Hybrid Communication Networks
University of Maryland, College Park
College Park, MD, 20742

ABSTRACT

The evolution of satellite applications has created new traffic patterns on the satellite links. That is because the number of users demanding service from the satellite links has increased tremendously. The latter results in a more sophisticated usage of resources. One way of accomplishing that is the use of multicasting, which is a natural extension of the broadcast nature of satellite medium. Numerous of the existing satellite applications demand reliable delivery of the transmitted data, which is opposed to the unreliable nature of the multicast transmission due to the error prone satellite medium. In this work we have designed a numerous of reliable multicasting protocols which are based on forward error correction (FEC) and air caching. The combination of those two techniques, results on a significant boost in the performance of reliable multicasting protocols.

INTRODUCTION

Due to the increase of applications and services that offered through the satellite links, the number of users demanding service has multiplied. The latter observation leads us to the thought that the available resources have to be spent efficiently. One way of accomplishing that is by using multicast communication between the satellite and the receiving ends. The problem with that is the unreliable nature of multicasting. So, for applications that demand reliable delivery of the transmitted data, multicasting seems not a very appropriate solution. The solution is to build multicast transmission of data on a platform that ensures reliability in the end-to-end delivery. Putting reliability in multicasting is not coming for free, because of the resulting increase in the usage of network resources. Also, taking into consideration the increased delay characteristics of the satellite links, we have to find a way of using the link as less as possible. The latter two observations show the difficulty of addressing the problem of designing efficient reliable multicasting protocols for this kind of networks. In the process of designing protocols that fit the flat network hierarchy, as the networks that we address have, we have found out that bandwidth usage and performance increase are coming into conflict. So, we had to increase bandwidth usage to decrease the end-to-end delay. As we are going to show, this extra bandwidth usage is compensated from the boost in performance, and we prove it by using the Normalized Gain factor, which is a relative measure of delay in terms of the extra bandwidth usage.

The reliable multicasting protocols we designed are based on two fundamental techniques, forward error correction and Air Caching. The combination of those two results in very efficient, robust and extremely scalable protocols. Our work is not focusing only in delay minimization but also in designing customized protocols based on the available processing power of the receivers (i.e. portable devices) and the buffering capabilities of them (i.e. devices that cannot have the appropriate memory to accommodate protocols that need excessive amounts of buffering). So, we designed also protocols that are light weighted in terms of buffering and processing requirements. Of course, the latter protocols do not have the performance of the protocols that use combined Air Cache and FEC, but have better delay characteristics than the reliable multicasting protocols that just use ARQ for the corrupted packets correction.

In the next section of the paper we revisit some of the work that has been done on reliable multicasting the recent years. In section 3, we briefly describe the techniques of FEC [4] and Air Caching [5]. Section 4, introduces the use of the latter two techniques and in what ways we can combine them in order to design protocols that are efficient in terms of end-to-end delay and in terms of network resources usage. In section 5, we describe some of the designed protocols. We have studied many reliable multicasting protocols based on different combinations of Air Caching and FEC but here we present the protocols with the best performance in terms of end-to-end delay. Finally, we give a brief introduction about the simulator that we used and give out some detailed results about our protocols' performance. We close this paper with the conclusion section, where we wrap up the whole idea and give out our main concerns.

0-7803-7225-5/01/$17.00 (c) 2001 IEEE

RELATED WORK

The reliable multicasting protocols that have been proposed up to now are targeting networks that have a certain degree of depth in hierarchy, so effective methods like the distributed error correction (DEC)[1] can be applied. Basic techniques that are worth mentioning are the local recovery and the forward error correction (FEC). Some of the existing reliable multicasting protocols are:

- SRM[1] belongs to the category of distributed error correction protocols and is based on local recovery. Although, SRM is very flexible, it suffers from several flaws including its incompatibility with asymmetric networks.
- RMTP [2] also, belongs to the DEC category and is very scalable. The basic characteristic of this protocol is the use of intermediate designated nodes that are responsible for a certain group of receivers. These nodes are collecting the requests for retransmissions and in such a way a very successful NACK suppression mechanism is achieved. This mechanism copes with the problem of the NACK implosion problem. The major drawback of this protocol is that it cannot be used in flat hierarchy networks.
- Also there are protocols like the RAMP, the MFTP, the APES [3] and numerous other protocols. The majority of them is based on distributed error correction and is using techniques like ARQ, local recovery, FEC and combinations.

Although, each network's architecture and each different application have different demands, the most favorable schemes are those that are based on DEC and are using combinations of techniques like ARQ, FEC and local recovery. Schemes that are based solely on centralized error recovery or ARQ are the least favorable among the existing reliable multicasting protocols.

In this work we are targeting networks with flat hierarchy, so the protocols that are based on DEC cannot be used, no matter how successful those protocols turn to be in terms of scalability, performance and latency.

Satellite networks are one-hop networks and are based on the broadcasting of information. Also, the broadcasting medium has higher delay characteristics than the corresponding media used in terrestrial communications. Those characteristics seem to limit flexibility for designing a very efficient reliable multicasting protocol. But this is not necessarily the case, as we will show in subsequent sections. Our design is based on two very successful and attractive techniques like the FEC and Air Caching. A brief presentation of them is given in the following two sections.

[1] The term DEC refers to the protocols, where nodes which are not necessarily the source of the transmitted data, have the privileges to retransmit requested data.

FEC AND AIR CACHING TECHNIQUES

In the following paragraphs we introduce the concepts of Forward Error Correction and Air Caching.

A. Forward Error Correction

Forward Error Correction is a proactive error recovery mechanism and is based on the transmission of parity packets along with the data packets. The generation of the parity packets is based on the data packets that belong in the transmission group (TG). One of the most important and attractive characteristics of FEC is that each one of the parity packets can correct each one of the data packets at the receiving end. Assume that the TG has size k and h parity packets are generated. Then, the TG is delivered reliably to the receiving end when k out of the $k+h$ packets have been received correctly, independently if these packets are parity or data packets, because each one of the parity packets can compensate for any of the data packets. In the case, where less than k packets have been received then the receiving end issues NACKs and the source or a designated retransmitter sends parity or data packets based on the corresponding protocol.

B. Air Caching

We can characterize Air Cache [5] as a continuous broadcast or continuous push of data. The immediate result of broadcasting the data continuously is that these data can be accessed from the end hosts more frequently and with less average end-to-end latency compared to data that do not belong to the Air Cache. The continuous broadcast of the cache results in low latency access to its contents. The latter is the basic characteristic especially in the case of the satellite links, because the data that are continuously broadcasted can proactively serve a request from any end host. Let us assume that we broadcast data with period T over a channel of bandwidth B. This broadcast can be considered to form a memory space of size $B \times T$ with some special characteristics:

- Any number of clients can access it concurrently i.e. there is no access contention.
- It can be accessed only sequentially, so the access time depends on the period T.

The technique of Air Cache has not been used before in designing reliable multicasting protocols. The performance gain of this technique's usage is significant

C. The Use of Air Cache

Assume that we want to transmit a number of packets (TG-Transmission Group) to a multicast group. We use two channels to transmit, the channel C_1 and C_2. The former is used for the transmission of the data packets in the TG, and the latter is used as Air Cache. The ongoing transmissions in the channels happen in parallel. This simultaneous transmission aims to the proactive correction of the lost or corrupted packets at the receiving ends, in order to avoid as much as possible the use of the satellite link for retransmissions. The effectiveness of the Air Cache depends on its content and its size. The content can be parity packets (FEC) or just data packets depending on the receivers' requirements (e.g. when we have receivers that do not have the needed processing power to use the parity packets, the air cache should contain data packets). The size depends on how much extra bandwidth we can dedicate to the transmission of the TG data packets. The content and the size of the air cache can be constant throughout the transmission or can vary depending on the feedback. There are many combinations for the evolution of content and size throughout the complete reliable transmission of data.

NETWORK TOPOLOGY

The assumed network topology in the design and evaluation of the proposed protocols is shown in the following figure. The important characteristics of this kind of networks are the increased propagation delay of the satellite link and the flatness of the communication hierarchy. The disadvantage of the former characteristic is that we cannot expect small end-to-end delay if we access this link many times and the

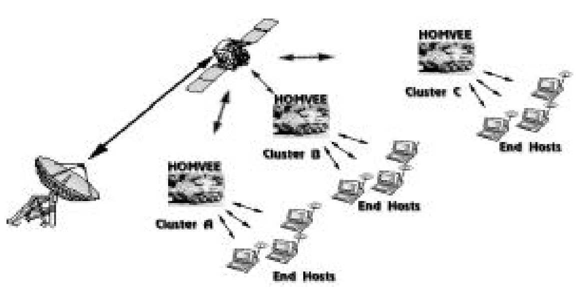


Figure 1: Network Topology. Basically, we focus on the communication part between the satellite and the HOMVEEs.

disadvantage of the latter characteristic is that we cannot use intermediate nodes in the hierarchy to cache some of the transmitted packets in order to use local recovery. Obviously, the flatness in hierarchy results in accessing the satellite link more, because the only way to recover the missing data is to request them through the satellite link.

DESCRIPTION OF THE PROPOSED PROTOCOLS

In section IV.C above, we presented briefly, numerous ways of how the size and the content of the Air Cache can vary. In this paper we cover two cases. The case where the size of the air cache is constant and the content is parity packets and the case where the size is adapted based on the feedback and the content is also parity packets. The former protocol is called PPAC-EB (Parity Packets in Air Cache – Extending Buffering, the extending buffering has added because of the buffering requirements for the receivers). The latter protocol is called ASPAC (Adaptive Size with Parity packets in Air Cache). ASPAC is based on the same principal idea as PPAC-EB but it tries to take advantage of the feedback in order to minimize the extra bandwidth usage from the Air Cache. Observing the protocols' names we can extract their characteristics. Both of them use a combination of Air Cache and FEC (i.e. due to the use of parity packets). ASPAC differs from PPAC-EB due to the adaptivity of Air Cache's size. The algorithmic presentation of PPAC-EB and ASPAC is given next.

<u>PPAC-EB (Parity Packets in Air Cache – Extending Buffering)</u>

<u>*Round 1:*</u> In the initial round PPAC-EB acts as follows. There are two parallel ongoing transmissions on the two channels (C_1 and C_2). On the first channel C_1 there is the transmission of the TG data packets and on the second channel C_2, there is the transmission of the contents of the air cache. The air cache contains ACRsize parity packets.

<u>*Round k (k>1):*</u> if there is not request for retransmission from any of the members of the multicasting group. That means that each one of them has received the TG data packets correctly or recovered them from the parity packets. So, the transmission of the TG data packets ends here, with transmission rounds equal to k-1. If there is request for retransmission for any of the members then there are two parallel ongoing transmissions on the two channels (C_1 and C_2). On the first channel C_1 there is the transmission of the TG data packets and on the second channel C_2, there is the transmission of the contents of the air cache (e.g. parity packets).

<u>*Round k+1:*</u> if there is not request for retransmission from any of the members of the multicasting group. That means that each one of them has received the TG data packets correctly or recovered them from the parity packets. So, the transmission of the TG data packets ends here, with transmission rounds equal to k. If there is request for retransmission, then round k is repeated with k=k+1.

<u>ASPAC (Adaptive Size Parity Air Cache)</u>

<u>*Round 1*</u>: The behavior of ASPAC in the initial round is the same as in PPAC-EB protocol. We have two parallel

transmissions going on. In C_1 we have the transmission of the TG data packets and in C_2 we have the transmission of the ACRsize parity packets. In this initial round we fill the air cache with the maximum number of parity packets (e.g. ACR size (Air Cache Round size)).

Round k (k>1): If there are no requests for retransmission then all the members of the multicasting group have received the TG data packets correctly and the transmission rounds are k-1. Otherwise, if there are requests for retransmission then depending on the number of different requested packets we adapt the size of the air cache.

$$ACRsize = \min(DDPR, \max ACRsize)$$

Where DDPR stands for Different Data Packets Requested, and the second term in min function, represents the maximum allowed Air Cache Size. After filling the air cache with the appropriate number of parity packets, there is again a parallel retransmission going on the two channels C_1 and C_2, the transmission of the TG data packets and the transmission of the air cache, respectively.

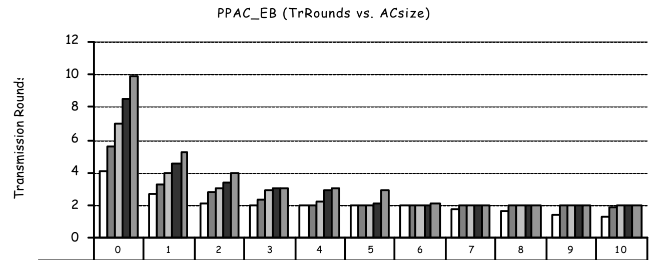
Round k+1: If there are no requests for retransmission, then all the participants in the multicasting group have received the TG data packets correctly. So, the transmission of the TG stops and the transmission rounds are k. Otherwise, repeat round k with k=k+1.

SIMULATION AND RESULTS

In order to simulate the two protocols (PPAC-EB and ASPAC) we build our simulator. The inputs were the Size of the multicast group, the Air Cache Size in packets and the Packet Error Probability (PEP). The results we present involve the end-to-end delay characteristics of both protocols, the relative performance gain compared to the extra usage of bandwidth (i.e. Normalized Gain) for the PPAC-EB and the robustness of ASPAC in error prone environments.

A. Transmission Rounds vs. Air Cache Size vs. Group Size

The measure we use to exploit the end-to-end delay is the number of Transmission Rounds needed in order the transmitted data to be delivered reliably to all the members of the multicast group. In the following graphs the number of packets (TG) that have to be delivered are 20. We run the simulation for different air cache sizes (0 to 10) and different multicast group sizes (10 to 100000). Figure 2 is referred to PPAC-EB and the figure 3 is referred to ASPAC. We assumed that PEP is 0.2 constant for every transmitted packet.

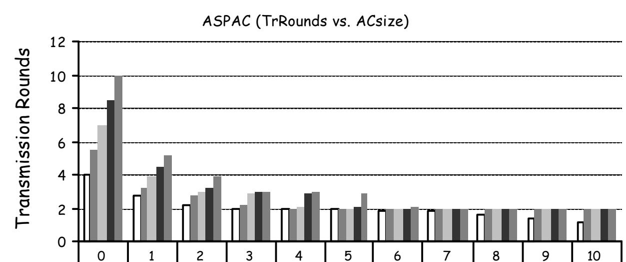


	0	1	2	3	4	5	6	7	8	9	10
10	4.03	2.72	2.14	2.05	2	1.98	1.98	1.8	1.6	1.36	1.24
100	5.55	3.29	2.75	2.28	2.01	2	2	2	2	1.99	1.9
1000	6.95	3.95	3.05	2.9	2.21	2.03	2.01	2	2	2	2
10000	8.48	4.52	3.33	3.01	2.89	2.15	2.03	2	2	2	2
100000	9.9	5.2	4	3	3	2.9	2.1	2	2	2	2

Air Cache Size

Figure 2: Transmission Rounds vs. Air Cache size vs. Group Size for PPAC-EB (Parity Packets in Air Cache – Extending Buffering). TG=20 and PEP=0.2

Analyzing the above graph, and comparing the different results for the various Air Cache sizes and group sizes, we can come into the conclusion that the designed protocols are very scalable and the delay results are very promising. Moving from air cache size 0 (i.e. we have just a simple ARQ going on) to the case where air cache size is 5 or 6, the improvement in performance is significant. The needed Transmission Rounds have decreased nearly by a factor of 5 in the case of 100000 group members.



	0	1	2	3	4	5	6	7	8	9	10
10	4.03	2.72	2.21	2.03	2	1.99	1.93	1.84	1.62	1.37	1.18
100	5.55	3.29	2.82	2.19	2.01	2	2	2	2	2	1.95
1000	6.95	3.95	3.07	2.87	2.16	2.01	2.02	2	2	2	2
10000	8.48	4.52	3.3	3.02	2.86	2.1	2.02	2	2	2	2
100000	9.9	5.2	3.9	3	3	2.9	2.1	2	2	2	2

Air Cache Size

Figure 3: Transmission Rounds vs. Air Cache size vs. Group Size for ASPAC (Adaptive Size Parity Air Cache). TG=20 and PEP=0.2

Also, one more of the observations concerning figure 2 is that we actually do not need a lot of extra bandwidth in order to reach stabilization in performance and getting a very scalable reliable multicasting protocol. The relative performance gain compared to protocols using only FEC is given in the next section and there we can get a feeling of the real improvement in performance, which compensates for the extra bandwidth usage (i.e. use of extra channel and extra bandwidth for Air Caching). Figure 3 contains the same information as figure 2 with the difference that this graph is for ASPAC protocol. Even though, ASPAC's air cache size in average is less than the corresponding Air Cache size in PPAC-EB, the performance of the two protocols is almost the same, and we expect the ASPAC's normalized gain to be better than PPAC-EB's, which is the case after simulating both protocols.

In the next paragraph we present one important result, which is the relative gain in transmission rounds when we take into consideration the extra bandwidth usage.

B. Normalized Gain vs. Group Size vs. Air Cache Size

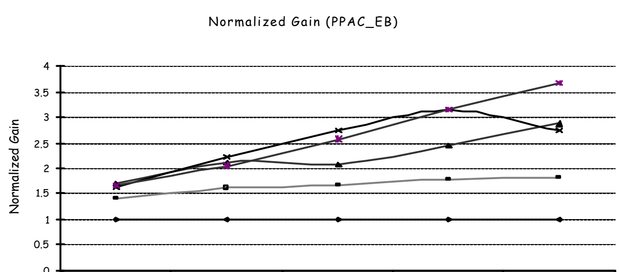
The Normalized Gain is defined as follows:

$$NormalizedGain = \frac{TransmissionRounds_{FEC} * TGsize}{TransmissionRounds_{PPAC} * (TGsize + ACsize)}$$

Using the above factor we get figure 4. In order the performance of PPAC-EB to compensate for the extra bandwidth usage, the Normalized Gain has to satisfy the following relation:

$$NormalizedGain \geq 1$$

The comparison is done using results where we do not use Air Cache but only parity packets (FEC) for the correction of lost or corrupted packets at the receiving ends.



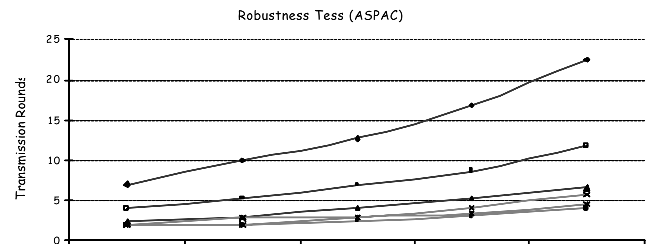
GroupSize	10	100	1000	10000	1E+05
AC=0	1	1	1	1	1
AC=1	1.411	1.607	1.676	1.787	1.813
AC=3	1.709	2.117	2.084	2.45	2.869
AC=4	1.628	2.22	2.739	3.155	2.731
AC=5	1.658	2.055	2.574	3.141	3.667

Figure 4: Normalized Gain vs. Group Size vs. Air Cache Size for PPAC-EB with TG=20 and PEP=0.2

The results show that the performance of the protocol PPAC-EB in terms of the needed transmission rounds compensates the extra bandwidth usage, even in the cases where the multicast group is small.

C. Transmission Rounds vs. PEP vs. Air Cache Size

The last graph presents the robustness of the ASPAC protocol. Obviously, the number of transmission rounds decreasing with the increase of Air Cache size. For example, comparing the case where we do not apply Air Caching with the case where the Air Cache size is 6, we can conclude the robust behavior of ASPAC protocol. The transmission rounds do not increase significantly even in the cases where the PEP is high, in contrast with the case, where there is no application of Air Cache.



Packet Error Probability	0.1	0.2	0.3	0.4	0.5
AC=0	7	9.9	12.7	16.8	22.4
AC=1	4	5.2	6.9	8.7	11.8
AC=3	2.4	3	4.1	5.3	6.6
AC=5	2	2.9	3	4.1	5.7
AC=7	2	2	3	3.3	4.5
AC=9	2	2	2.5	3.1	4

Figure 5: Transmission Rounds vs. Packet Error Probability vs. Air Cache Size for ASPAC with TG=20 and Group Size = 10000 members.

CONCLUSIONS

The goal of this work is to reduce the satellite link accesses during the reliable multicast transmission of a TG. That is because each time we use the satellite link we have to pay the cost of propagation delay. So based on this, we choose to use some extra bandwidth in order to achieve this. PPAC-EB's and ASPAC's performance compensates for the extra bandwidth usage. Beyond the promising results in terms of end-to-end delay, the protocols are characterized by scalability and robustness, which are crucial factors for the delivery of the expected QoS in today's networks.

REFERENCES

[1] S. Floyd, V. Jacobson, C-G. Liu, S. McCanne, L. Zhang, "A Reliable Multicast Framework for Light-weight Sessions and Application Layer Framing", IEEE/ACM Transactions on Networking, Nov. 1996.

[2] S. Paul, K. Sabnani, R. Buskens, S. Muhammad, J. Lin, S. Bhattacharyya, "RMTP: A Reliable Multicast Protocol for High-Speed Networks", 10th Annual IEEE Workshop on Computer Communications, Sept. 1995.

[3] D. Rubenstein, S. Kasera, D. Towsley, J. Kurose, "Improving Reliable Multicast Using Active Parity Encoding Services", Proc. of INFOCOM '99.

[4] J. Nonnenmacher, E.W. Biersack, D. Towsley, "Parity-Based Loss Recovery for Reliable Multicast Transmission", IEEE/ACM Transactions on Networking, Vol. 6, No. 4, August 1998.

[5] K. Stathatos, N. Roussopoulos, J.S. Baras, "Adaptive Data Broadcasting Using Air Cache", WOSBIS '96.