

INTRODUCING AIR CACHING TECHNIQUES FOR BOOSTING RELIABLE MULTICASTING 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 emergence of satellite networks and the demand for reliability in transferring large files over the network via satellites has created the need for building effective reliable multicasting protocols, customized for this kind of networks. During the previous years, several protocols and techniques have been proposed to address the demanding problem of reliability in multicasting protocols. In our study we are trying to give a different perspective to the work that already has been done, with the use of Air Caching. The combined use of this technique along with the use of forward error correction (FEC) can boost substantially the performance of reliable protocols in terms of transmission rounds, by paying a small overhead in bandwidth usage. We present three protocols, the UDPAC (Unchanged Data Packets in Air Cache), the RDPAC (Replace Data Packets in Air Cache) and the PPAC (Parity Packets in Air Cache). The performance of these three protocols, in terms of Transmission Rounds and robustness in erroneous environments, is compared with two already known protocols, which are based on the simple ARQ and the FEC, respectively.

INTRODUCTION

Hybrid networks have emerged as ubiquitous over the past years, and as result there is a large number of applications designed for them. The basic entities of these networks are the satellites, which are used as data sources, as gateways or as routers of the information that flows around the existing hybrid network infrastructure. Many of today's applications are becoming more demanding in terms of bandwidth usage and the basic requirements are reliability and scalability.

ated Laboratory Program, Cooperative Agreement DAAL01-96-2-0002. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

Examples of such applications are the transfer of large medical files, telemedicine, video distribution and the virtual war games, which are important to military planners.

Multicasting is a solution to the demands of many modern applications, because it saves bandwidth, decreases latency and increases scalability. As we will describe in section 2, a lot of work has been done for wireline networks or networks that are based on point-to-point communications.

The problem of designing an efficient reliable multicasting protocol in the case of broadcast transmission based networks, which have flat hierarchical structure, is still emerging. So, the question is how we design a protocol for this kind of networks, that has high performance, is scalable and the buffering requirements are not prohibitive.

Two of the basic characteristics of this class of networks are their flat hierarchy and their broadcast nature. The former does not give much flexibility to the protocol's designer, and the latter can be used effectively for getting an efficient and interesting protocol. Our objective is to propose a protocol that fits well, flat hierarchy networks, which have broadcast nature, like satellite networks.

In the following section we briefly describe the work that already has been done for the design of reliable multicasting protocols over satellite links for flat hierarchy networks. In sections 3 we give a brief description of FEC techniques[7][9]. In section 4, we present the notion of the air caching and why this technique can be used for the design of a very efficient reliable multicasting protocol. In section 5, the network architecture and the proposed generic protocol for this infrastructure, are presented. The simulation results and the evaluation of the protocol's performance

*Prepared through collaborative participation in the Advanced Telecommunications & Information Distribution Research Program (ATIRP) Consortium sponsored by the U.S. Army Research Laboratory under the Feder-

are given in section 6. Finally, in section 7 we provide conclusions.

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). The former aims on the localization of retransmissions and the latter aims in the effective use of bandwidth and the decrease of latency. 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. The latter disadvantage keeps the protocol from scaling in satellite networks.
- RMTP[2][4] 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 that is one of the basic problems that arise in multicasting environments. The major drawback of this protocol is that it cannot be used in flat hierarchy networks.
- Also there are protocols like the RAMP[3][4], the MFTP[4], the APES[5] 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 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. The latter restriction is due to the lack of hierarchy levels in flat networks.

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.

FORWARD ERROR CORRECTION (FEC)

Forward Error Correction is a proactive mechanism compared to the ARQ mechanism. In ARQ, retransmissions are taking place, only when the receivers are requesting corrupted or erroneous data. Obviously, ARQ mechanisms are wasting little bandwidth at the cost of increased latency, due to feedback delay. FEC acts the other way around due to its proactive nature.

FEC 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[6].

There are two different approaches in implementing FEC. The one is the layered approach[7], where the reliable multicasting layer sits atop of the FEC layer in the protocol stack. The second is the integrated approach, where the reliable multicasting layer and the FEC layer are fused[7]. The reliable multicasting layer generates the TG, which consists of data packets and the FEC layer generates the parity packets, based on the TG's packets. The TG along with the parity packets is called the FEC multicast block.

Assume that the TG has size k and h parity packets are generated. Then, the TG is delivered reliably to the receiver.

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.

ing 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 re-transmitter sends parity or data packets based on the corresponding protocol.

AIR CACHING

We can characterize Air Cache[8] 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. Using the latter description of Air Cache provides the justification for the term; because it acts like a cache. The notion of a cache, as it appears in computer engineering, is a very fast memory that contains the most popular data. The speed of the cache results in low latency access to its contents. In the case of the Air Cache, the latter is the basic characteristic, 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. A direct consequence is that the average access time depends on the size of the memory, which in turn is determined by the period T .
- The server cannot have any information about the effectiveness of this memory space, i.e. which –if any– clients actually use it.

The technique of Air Cache has not been used before in designing a reliable multicasting protocol, and as we are going to show there is significant boost in the performance of the reliable protocol by dedicating a portion of the available bandwidth for the Air Cache.

NETWORK INFRASTRUCTURE

The basic characteristic of the network topology that we assume is the flat hierarchy. We are interested basically on the part of the network between the satellite and the receivers that have the hardware to receive from it. Those receivers form a multicast group by requesting and receiving or dedicated to receive the same information from the

satellite. The received information afterwards can be forwarded to hosts that do not have the appropriate hardware to receive directly from the satellite. As we said, there is no much work done for this part of the network, which can end up to add the largest part on the end to end delay and this is the case because there is a lot of work done for the terrestrial links. There are sophisticated protocols that can achieve very good performance, so the delay from the satellite link dominates the end to end delay.

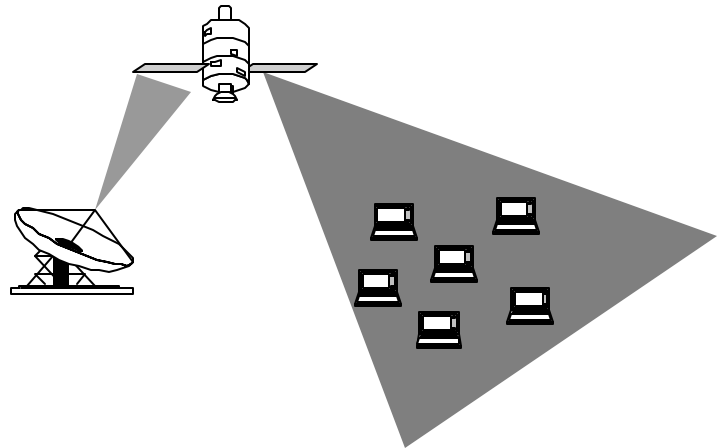


Figure 1: Network Infrastructure

Understanding the importance of the satellite link as a part of a largest hybrid network or as an individual communication link between a sender (e.g. a satellite) and a group of hosts that receive the same information.

Obviously, assuming the above network infrastructure there is no much flexibility in designing a reliable multicasting protocol for this kind of networks.

OPERATIONAL AND ALGORITHMIC VIEW OF THE PROTOCOLS

Having as a goal to improve the performance of reliable multicasting transmission over the satellite link we designed three protocols, the UDPAC (Unchanged Data Packets in Air Cache), the RDPAC (Replace Data Packets in Air Cache) and PPAC (Parity Packets in Air Cache). All of them are based primarily on the Air Caching techniques. This has as a result the use of an extra channel (e.g. more bandwidth) but the gain in performance compensates for this extra usage of bandwidth.

Before describing the schemes that we evaluated, we are giving some definitions and a descriptive figure, which presents the principal idea of the protocol.

Definitions

We are defining two coexisting time slots:

- **Transmission Group Transmission Round (TGTR):** Is defined as the time T_{TGTR} that it takes for all the packets in the TG to be transmitted.
- **Air Cache Transmission Round (ACTR):** Is defined as the time T_{ACTR} that it takes for all the packets in the Air Cache to be transmitted once.
- The relationship between TGTR and ACTR is:

$$ACTR \leq TGTR$$

(1)

- The ACTR and the TGTR depend on the Air Cache Size $|AC|$ and the Transmission Group Size $|TG|$ in packets, respectively. Also, another factor that affects these times is the dedicated bandwidth B_{TG} and B_{AC} for the transmission of the corresponding packets.
- The number of times K that the Air Cache contents are transmitted throughout the transmission of the TG is given by (4), using (1) and the following two relations:

$$ACTR = \frac{|AC|}{B_{AC}}$$

(2)

$$TGTR = \frac{|TG|}{B_{TG}}$$

(3)

$$\text{So, } K = \left\lfloor \frac{B_{TG}}{B_{AC}} \right\rfloor, K \in \mathbb{N}^* \quad \text{and} \quad |AC| = \left\lfloor \frac{|TG|}{K} \right\rfloor \quad (4)$$

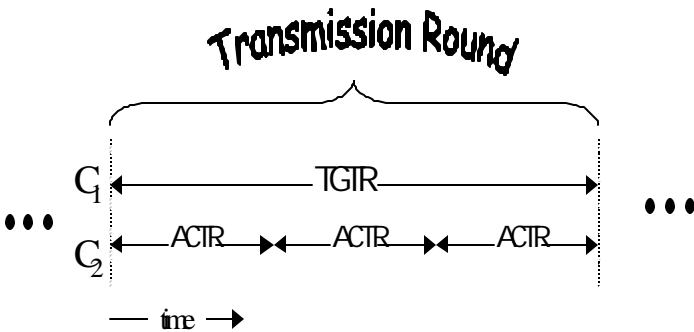


Figure 2: Principal Operation of the Proposed Protocol. C_1 channel is used for the TG and C_2 is used for the Air Cache. In this case Air Cache rounds are 3.

Protocol UDPAC (Unchanged Data Packets in Air Cache)

In this protocol, the contents of the Air Cache are data packets, which have been selected randomly from the TG. Throughout a single transmission of the TG, the Air Cache contents are being transmitted K times. These data packets can be used from the receivers to correct the corresponding corrupted data packets, without issuing request for retransmission. The meaning of the repetitive transmission of the Air Cache contents, which are also subject to errors, is that they can replace themselves in a subsequent Air Cache round. This iterative transmission of Air Cache makes the protocol more robust in highly error prone environments as the satellite links are.

The protocol's algorithm is:

Step I: Transmit the TG and the AC. The latter has been filled with random data packets from the TG.

Step II: If there are errors, retransmit the TG and the AC, else the TG has been transmitted successfully.

Contents of the Air Cache for (UDPAC)

The contents of the Air Cache are chosen randomly from the data packets that belong in the corresponding TG. The contents of the air cache are changing per TGTR. This means that the air cache remains the same throughout the Air Cache rounds per TG rounds. The visual relation between the ACTR and the TGTR is given in Figure 2.

Protocol RDPAC (Replace Data Packets in Air Cache)

The only difference of this protocol from the one mentioned previously, is that the Air Cache contents are changing every ACTR. But how we choose the contents of the air cache each time we refresh it? Again the contents are chosen randomly from the data packets that belong in the TG. This random filling of the air cache is based on the fact that we want to eliminate the feedback from the receivers to the satellite. So the latency is going to be decreased significantly compared to protocols like ARQ, as we are going to demonstrate to the following section.

Protocol PPAC (Parity Packets in Air Cache)

The previous two protocols do not use FEC (forward error correction), and they are useful when we do not want to use extra memory for the storing of parity packets and extra processing power for coding and decoding. The latter two advantages are not coming for free, because by selecting not to use parity packets we sacrifice performance.

In this protocol, the Air Cache contains parity packets. So, upon reception of each one of the parity packets, any of the corrupted data packets can be corrected. So, assuming that we have a large number of parity packets we choose in each TGTR to send a different set of parity packets in C_2 during the transmission of the TG in the first channel C_1 . If after the TGTR, there are requests for retransmission, then we repeat the transmission of the TG and the AC.

SIMULATION RESULTS

In this section we demonstrate some of the basic results that we got by simulating the above mentioned protocols. In Figure 3 we compare the performance (Transmission Rounds per TGTR) of the protocols UDPAC and RDPAC for different cache sizes versus the performance of ARQ protocol for constant group size (e.g. 10000).

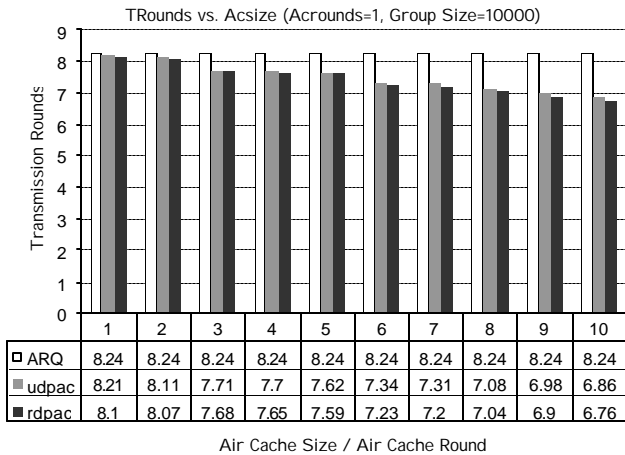


Figure 3: Impact of the Size of Air Cache for UDPAC and RDPAC compared to simple ARQ protocol.

Obviously, from the above figure the improvement in performance of UDPAC and RDPAC increases as we increase the air cache size. The improvement in this case is not as large in order to compensate for the extra bandwidth usage, but for those protocols we do not need feedback in contrast to ARQ. The slight increase in gain, results from the introduced randomness in filling the air cache.

The following figure presents the comparison between two of the most known protocols (e.g. ARQ and FEC) and the three proposed protocols.

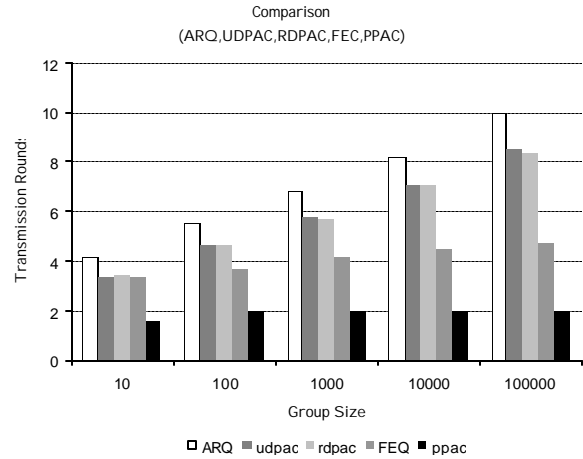


Figure 4: Performance Comparison between ARQ, UDPAC, RDPAC, FEC, PPAC for Air Cache Size = 8.

Obviously the performance of PPAC is very promising taking into consideration the scalability factor. PPAC seems to perform very well even for large number of users. This increase in performance compared to the other protocols is not coming for free, but to the existence of the Air Cache. Our question now is if the increase in performance compensates for the extra bandwidth usage

We introduce the term of Normalized Gain, which we define as follows:

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

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

$$NormalizedGain \geq 1$$

As the following figure demonstrates this is the case for various group sizes between 10 and 100000 multicast group members.

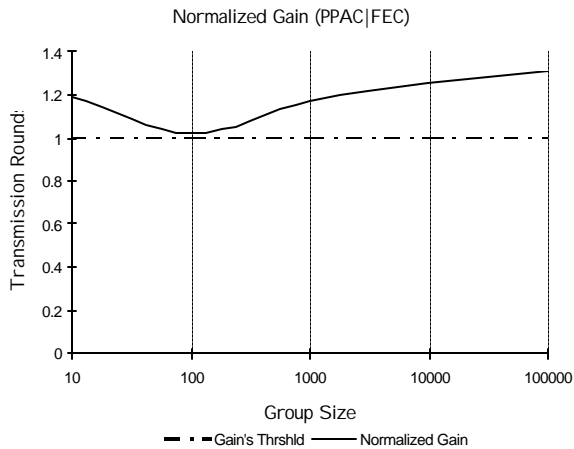


Figure 5: Normalized Gain of PPAC compared to FEC

So, the PPAC protocol is very promising as long as we interested in scalability and in elimination of ACK implosion problem due to the fact that there is no need for feedback and by using the air cache we act proactively for the upcoming packet errors at the recipients.

CONCLUSIONS AND FUTURE WORK

The goal of this work is to reduce the satellite link access during the reliable 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's performance compensates for the extra bandwidth usage. Also, UDPAC and RDPAC seem to be very useful in cases where the receiving nodes do not have enough buffering space or processing power for the coding/decoding of the parity packets. All of these three protocols seem to be very promising in terms of performance (e.g. PPAC) and system requirements of the receiving entities (e.g. UDPAC and RDPAC). The latter is going to be very useful, for devices that we want to be inexpensive, lightweight and having communication capabilities.

The basic disadvantage in terms of resource utilization for all the above-mentioned protocols is the fact that the size of the air cache remains constant and the contents of it are chosen randomly. Although, we eliminate the needed feedback, we can do better in performance and resource utilization by using effectively the requests for retransmission in order to shrink or expand the air cache by filling it with the most requested packets.

*The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied of the Army Research Laboratory or the U.S. Government.

REFERENCES

- [1] S. Floyd, V. Jacobson, C-G. Liu, S. McCanne, L. Zhang, "A Reliable Multicast Framework for Lightweight Sessions and Application Layer Framing", IEEE/ACM Transactions on Networking, November 1996.
- [2] S. Paul, K. Sabnani, R. Buskens, S. Muhammad, J. Lin, S. Bhattacharyya, "RMTP: A Reliable Multicast Protocol for High-Speed Networks", Proc. of the Tenth Annual IEEE Workshop on Computer Communications, September 1995.
- [3] R. Braudes, S. Zabele, "Requirements for Multicast Protocols", Network Working Group, RFC 1458, May 1993.
- [4] C. K. Miller, "Multicast Networking and Applications", Addison Wesley Longman, Inc., 1999.
- [5] D. Rubenstein, S. Kaser, D. Towsley, J. Kurose, "Improving Reliable Multicast Using Active Parity Encoding Services", Proc. of INFOCOM '99.
- [6] S. Lin, D.J. Costello, "Error Correcting Coding: Fundamentals and Applications", Englewood Cliffs, NJ: Prentice Hall, 1983.
- [7] 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.
- [8] K. Stathatos, N. Roussopoulos, J.S. Baras, "Adaptive Data Broadcasting Using Air Cache", in WOSBIS, 1996.
- [9] S. M. Payne, J. S. Baras, "Reliable Multicasting via Satellite: Delay Considerations", Proc. of ATIRP, 4th Annual Symposium, March 21-23, 2000.