On Cloud-Based Multisource Reliable Multicast Transport in Broadband Multimedia Satellite Networks

Alberto Gotta, Nicola Tonellotto, and Emnet T. Abdo

CNR-ISTI, National Research Council of Italy, via G. Moruzzi 1, 56124 Pisa, Italy
{alberto.gotta,nicola.tonellotto}@isti.cnr.it,
emnettsadiku@gmail.com

Abstract. Multimedia synchronization, Software Over the Air, Personal Information Management on Cloud networks require new reliable protocols, which reduce the traffic load in the core and edge network. This work shows via simulations the performance of an efficient multicast file delivery, which advantage of the distributed file storage in Cloud computing. The performance evaluation focuses on the case of a personal satellite equipment with error prone channels.

Keywords: Satellite, Multicast, Cloud Computing, ASM.

1 Introduction

The *Cloud* is a novel distributed platform that provides an abstraction between the computing resource and its underlying technical architecture (e.g., servers, storage, networks), enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction. The increased demand of network resources can not sustain the exponential growth of data dissemination required by the next generation Web-enabled services, since the current Internet business models are shifting toward pervasive and ubiquitous mobile devices (e.g., GPS navigators, smart-phones, netbooks) and new socio-economic models arise from the Web 2.0 (e.g., social networking, context awareness, All-as-a-Service). Some of the current types of applications viable to be ported on Clouds will place very high demands on the network, often requiring the speedy delivery of the same data to multiple destinations (one-to-many communications). In cases of data dissemination from a single content provider to a group of consumers Reliable Multicast Transport (RMT) protocols can be a key factor fostering optimal resource allocation. If we look at the traffic load of a single user only that performs the synchronization of all his devices (smartphones, smart-TVs, laptops, NASs, set-top-boxes, etc.), the convenience of a RMT is outstanding. In particular, when all the personal devices are behind the home router (cable or satellite), only a single multicast flow crosses even the access network, and the replicas affect only the Local Area Network at home. In fact, since the current typical employment of Cloud services for home customers is

P. Pillai, R. Shorey, and E. Ferro (Eds.): PSATS 2012, LNICST 52, pp. 156–163, 2013.

devoted to personal devices synchronization of personal multimedia file, i.e. movies, music, photos, e-books, documents, notes, contacts, apps, bookmarks, etc., the main advantage of using an RMT in a multicast domain is the significant reduction of bandwidth on the backbones, that is, in a Cloud scenario, on the core and edge networks.

In the field of RMT protocols, the primary design goals are providing efficient, scalable, and robust bulk data transfer across possibly heterogeneous IP networks and topologies to a group of users, and, more specifically, to a set of devices. In particular, referring to reliable transport means either imposing a certain level of Quality of Service or providing the confirmed delivery of data, when applications requires the integrity of received data.

The Negative ACKnowledgment (NACK) Oriented Reliable Multicast (NORM) protocol [1] supports a reliable multicast session participation with a minimal coordination among senders and receivers. NORM allows senders and receivers to dynamically join and leave multicast sessions at will, with a marginal overhead for control information and timing synchronization among participants. To accommodate this capability, NORM message headers contain some common information allowing receivers to easily synchronize to senders throughout the lifetime of a reliable multicast session. NORM can self-adapt to a wide range of dynamic network conditions with little or no pre-configuration, e.g., in case of congestion situations on network bottlenecks due to traffic overload. The protocol is tolerant to inaccurate round-trip time estimations or loss conditions that can occur in mobile and wireless networks; it can correctly and efficiently operate even in situations of heavy packet loss and large queuing or transmission delays.

A NORM session is defined within the context of communicating participants over a network using pre-determined addresses and host port numbers in a *connectionless* fashion (e.g., UDP/IP). The participants communicate by using a common IP multicast group address and port number. NORM senders transmit data content in the form of *objects* to the session destination address and NORM receivers attempt to reliably receive the transmitted *object* using NACKs to repair requests, in a confirmed delivery fashion. Moreover, the sender logically segments a transmitted *object* into Forward Error Correction (FEC) coding block. The parity segments may be transmitted proactively, i.e., appended to the information part of the coding block, reactively, i.e., reacting to a repair request, or both, i.e., devoting part of the parity segments to proactive aim and the remaining to reactive one.

The main drawback of a reliable multicast delivery is that the sending agent adapts the throughput in order to not penalize the most impaired receiver (e.g. a personal mobile satellite equipment in severe fading conditions). The other members of a group see this as performance degradation, since the delivery may last for longer. The main overhead in terms of data delivery delay, in case of disruptive channels, is due to the error recovery phase. This is particularly harmful when long latencies are experienced on the channel, as in the case of GEO satellite links, since the retransmission request requires at least twice the channel latency in time units to be satisfied.

Even if no standard protocols have still been defined for RMT, NORM is the best candidate and is going to be finalized as reference standard by IETF-RMT-WG [2].

This work presents the performance evaluation of an Any Source Multicast (ASM) multi-source multicast session, which relies on the architecture of Cloud/Grid distributed parallel storage servers, and which aims at reducing the data delivery delay with a specific advantage for those Cloud applications that would require to achieve a quasi-real-time playing of multimedia files.

2 State of the Art on File Delivery in Grid/Cloud Networks

In the case of unicast communications in Grid networking, GRIDFTP is an extension of the standard File Transfer Protocol (FTP) for use with Grid computing [3]. It is defined as part of the Globus toolkit, by the GRIDFTP working group [4]; GRIDFTP can be used independently, to provide high-speed file transfers. An FTP data transfer is limited by the maximum size of TCP/IP packet and the acknowledged (ACK) reception of each data packet. In WANs, or in Satellite Networks, a simple file transfer based on FTP is affected by latency, as the transmission delay of data packet and the acknowledgements reduces the data transfer rate. GRIDFTP supports parallel data transfer through FTP command extensions and data channel extensions. The use of multiple streams in parallel (see Figure 1) (even between the same source and destination) improves the aggregate bandwidth over a single stream, according to [5]. Data may be striped or interleaved across multiple servers, as in the network disk cache of a distributed parallel storage server or a striped file system.

GRIDFTP includes extensions that initiate striped transfers, which use multiple TCP streams to transfer data that is partitioned among multiple servers. Striped transfers provide further bandwidth improvements over those achieved with parallel transfers.

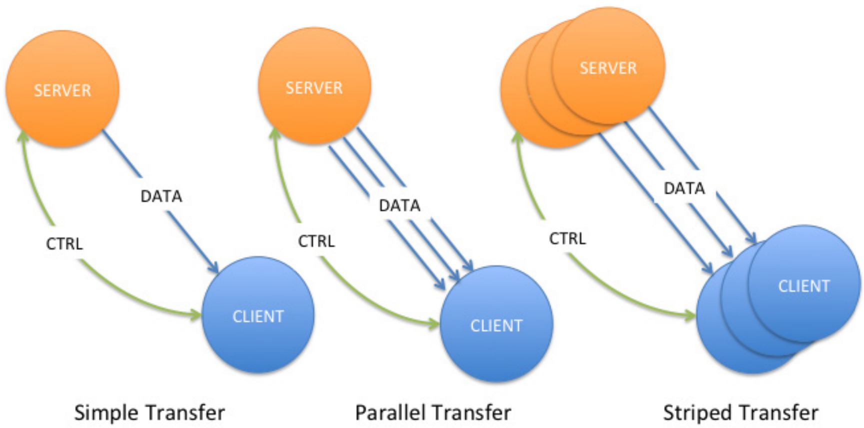

Fig. 1. Grid FTP transfers

The main concepts that are behind parallel and striped distributions can be summarized as follows:

1. In the case of a single server shared among a large number of receivers, the bottleneck rate may undergo saturation. This leads to a scalability problem that can be only solved by redunding the number of servers.
2. When multiple servers offer different performance levels in term of reliability, due to the network they are respectively behind, two receivers, belonging to the same class of service, may experience different delivery times according to the selected server. This is unfair, in terms of service level agreement, with the respect to customers that have subscribed for the same level of service.
3. With striped transfers from multiple servers, two receivers belonging to the same class should experience at least the same performance, apart from the performance of the different access networks, which are not in charge of the Cloud service provider, but rather of the Internet provider.

According to this backing provided by Cloud/Grid networking, a preliminary investigation on reliable Any Source Multicast (ASM) transmissions is provided in the follow in terms of performance evaluation of NORM protocol over large delay error prone satellite channels.

3 The Case Study

NORM is a protocol centered on the use of selective NACKs to request repairs of missing data. NORM provides for the use of packet-level forward error correction (FEC) techniques for efficient multicast repair and optional proactive transmission robustness [6]. FEC-based repair can be used to greatly reduce the quantity of reliable multicast repair requests and repair transmissions in a NACK-oriented protocol. The principal factor in NORM scalability is the volume of feedback traffic generated by the receiver set to facilitate reliability and congestion control. NORM uses probabilistic suppression of redundant feedback based on exponentially distributed random back-off (BO) timers. This allows NORM to scale well while maintaining reliable data delivery transport with low latency relative to the network topology over which it is operating. The NACKing procedure begins with a random BO timeout in order to avoid the possibility of NACK implosion in the case of sender or network failure. At the end of the BO time, the receiver will suppress its NACK message whether at least one of the following conditions is verified:

1. a NACK message, received from another receiver, equals or supersedes the receiver's repair needs;
2. the receiver detects the sender is sending ordinally earlier blocks (in response to earlier NACKs) than what it is currently pending repair.

Finally, the receiver enters a rest phase, according to another exponentially distributed random timer called hold-off (HO) timer, before starting a new recovery phase, when a new back-off timer will be sorted.

Both BO and HO are function of the round trip time (RTT) of the group (GRTT) of receivers, calculated by the sender/s, which collects the individual RTT of each receiver and assumes the maximum one as that of the group.

When the second condition is verified, the receiver will NACK again in a following repair cycle, after the senders ordinal transmission point will have exceeded the receivers pending repair needs. However, in case of large channel latency as in wireless and satellite technologies, this policy - that avoids multiple repair requests in contiguous repair cycles - stretches the time required to deliver a block of data to the receiving group. Figure 2 highlights through a dashed line the case of a NACK suppression due to the first repair request still pending of resolution.

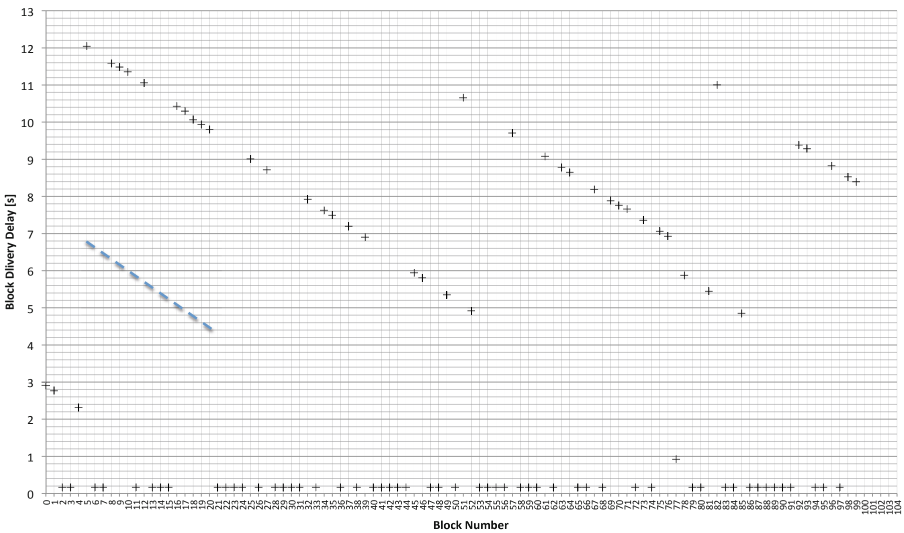

Fig. 2. Data block delivery delay

The simulation environment poses on NS2 v2.34 network simulator patched with the nrl-NORM code [7], which implement the reliable multicast transport according to [1]. The scenario is made up of a certain number of receivers and a variable number of repositories where files are stored. The test in Figure 2 is obtained with only one sender and two receivers, with a GEO satellite channel latency of 125 ms, an average loss probability of 5%, a transmission rate of 256 kbps. A data block, i.e., the FEC code information part, is made up of five segments of 1024 bytes each and one parity segment is devoted to the reactive redundancy.

In Figure 3, 10 consecutive files of 512000 bytes size have been delivered according to the single delivery in Figure 2, in order to depict individual file (object) delivery over time. Each rectangle represents the start and completion of a delivered file. The rectangles overlap because the file delivery of multiple files overlaps each other, i.e., as new data is sent while repairs of the previous file are still occurring.

Looking at the "per file" goodput, it is about 160-180 kbps but the overlapping delivery of the series of multiple file objects comes an average goodput of about 220 kbps, which is not bad for 256 kbps sender rate and 5% packet loss. In fact, each file delivery takes about 24-25 seconds to execute, in each run.

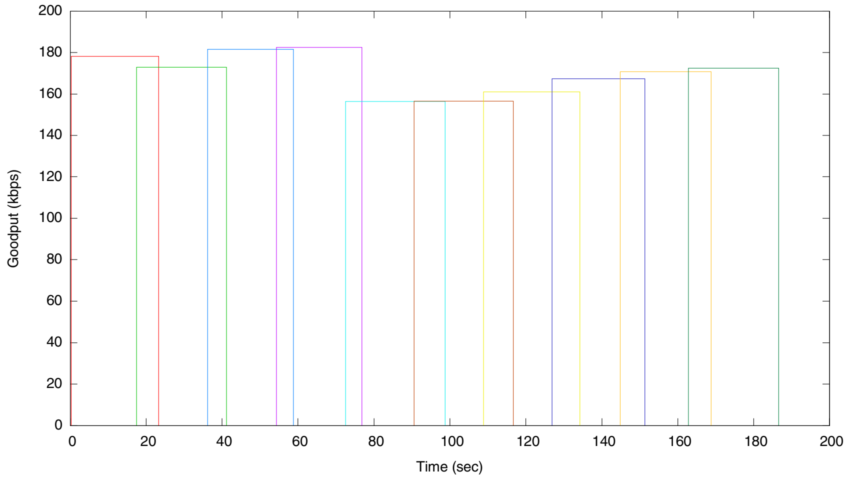

Fig. 3. Object delivery performance

In case of best effort file transfer, the sequential delivery of blocks is not a requirement, and NORM performs quite well. However, in case of on-demand multimedia playing from the Cloud, an out of order delivery of subsequent data block will cause the freezing of the playing. In Figure 2, e.g., block #5 is delayed of about 12 s; this means that after playing 20 KB of the file, the player freezes for more than 12 s, in addition to the 3 s required to sequentially play the earlier four blocks of the sequence.

The only first suppression causes a shifting in time of the following recovery cycles for the pending blocks. This effect is degenerative and incremental with the file size.

Proactive FEC might be a solution, at expense of an overhead of bandwidth that reduces the overall goodput. However NORM does not foresee a control mechanism for proactive FEC adaptation, and, even in case of that, the advantage would stand if the delivery takes a much longer time than that required to the algorithm to adapt to the channel impairments.

A different solution relies on the chance of segmenting the original file into chunks and deliver the object from multiple servers, each of them accounts for delivering a single chunk or few of them. This policy was formerly foreseen by the multicast Internet protocol, which accounts for the ASM as in the case of the Internet Group Management Protocol (IGMP) [8] for IPv4 networks, i.e., one or more sources and multiple receivers; in addition, it stems at the basis of Grid file transfer [3] policies for large file delivery, and has been being inherited by Cloud computing. In fact, over the past few years, Cloud computing has emerged as a new paradigm in advanced computing as a flexible, on demand infrastructure aiming at transparently sharing data, calculations, and services among users of a massive Grid [9].

A file delivery is performed by choosing the number of sources (repositories) according to the number of chunks, in which the file is split. Between the core network, where there are the repositories, and the receivers a satellite bottleneck link play the role of the access network. The satellite link bit rate is set to 1 MB/s and the latency is 125 ms.

Table 1. Performance Evaluation of ASM delivery to 2, 5, and 10 Receivers

Number of Receivers	Chunk Size [KB]	Allocated Servers	Average FDD [s]	Average BDD [s]
2	100	32	7.13	4.30
	800	4	11.97	3.75
	1600	2	14.67	4.44
	3200	1	22.40	7.58
5	100	32	7.68	4.41
	800	4	12.19	4.28
	1600	2	15.23	5.54
	3200	1	24.49	9.43
10	100	32	8.32	4.48
	800	4	12.77	4.37
	1600	2	15.47	5.74
	3200	1	24.75	9.75

In Table 1 we investigate a 3.2 MB file delivery delay (FDD) – a typical MP3 song – segmented into a certain number of chunks, given the number of receivers. The NORM sending agent is configured, in order to organize the chunks into blocks of 10 data segments and 5 additional parity segments are generated for reactive recovery, i.e., no proactive parities are sent, appended to information. In addition, block delivery delay (BDD) is provided in Table 1. The simulations are performed with 2, 5, and 10 receivers respectively. We assume that information segments are released to the application layer by the transport agent through a de-jitter buffer, when all the segments of a received block are successfully retrieved, either when the decoding is required or not. For this reason BDD plays the main role in place of the segment delivery delay.

The simulations reveal that there is an optimal choice of the chunk size, and hence, of the number of allocated servers, independently from the number of receivers. This size is shown in Table 1 approximately between 100 and 800 KB. The reason has to be found in relation to the GRTT, which determines the timings for the BO and HO timers. From the log files, we have experienced that, when the chunk is too short (e.g. 100KB as in Table 1), the recovery phase after a suppression happens when the file delivery is already finished and only the missed packets are waited from recovery phases, before the relative blocks of data are passed to the application layer. By choosing an opportune chunk size (e.g. 800 KB) the number of server is optimized and the BDD can gain even more than 70%, by reducing the average delivery delay per block.

We remark that keeping the BDD as low as possible is one of the main goals in case of on-demand multimedia application, in order to match the user agreement and avoiding the suspension of a live application.

4 Conclusions and Future Works

These preliminary simulations show the benefit of ASM delivery in the context of Cloud networking, when the access network presents long latency. This technique allows reducing both the network load and the delivery delay, adopting a reliable multicast protocol. According to the parallel striped transfer, provided by Cloud computing, the future activities will deepen this investigation and will account for other challenging scenarios with particular interest to the enhancement that network coding could provide in the case of file delivery and broadband multimedia services.

Acknowledgments. The authors thank B. Adamson for his valuable support provided here and in our ongoing research activity on reliable multicast.

References

[1] Adamson, B., Bormann, C., Handley, M., Macker, J.: Negative-acknowledgment (NACK) Oriented Reliable Multicast (NORM), RFC 5740 (November 2009), http://tools.ietf.org/html/rfc5740
[2] http://datatracker.ietf.org/wg/rmt/charter/
[3] Taylor, I.J.: From P2P to Web Services and Grids - Peers in a Client/Server World. Springer (2005)
[4] http://forge.gridforum.org/projects/gridftp-wg/
[5] Butler, D.: GRIDFT Server Simple Performance Measurements, BBC R&D White Paper WHP178, http://www.bbc.co.uk/rd/pubs/whp/whp178.shtml
[6] [RFC3453] The Use of Forward Error Correction (FEC) in Reliable Multicast
[7] http://cs.itd.nrl.navy.mil/work/norm
[8] [RFC4604] IGMP protocol Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast
[9] Tuan-Viet DINH. Cloud Data Management, Bibliography Report, ENS de Cachan (February 2010), ftp://ftp.irisa.fr/local/caps/DEPOTS/BIBLIO2010/Dinh_Viet-Tuan.pdf