Evaluation of Reliable Multicast Implementations with Proposed Adaptation for Delivery of Big Data in Wide Area Networks

Aleksandr Bakharev
Siberian State University of Telecommunication and Information Sciences
Novosibirsk, Russia
a.bakharev@emw.hs-anhalt.de

Eduard Siemens
Anhalt University of Applied Sciences
Koethen, Germany
e.siemens@emw.hs-anhalt.de

***Abstract*—This paper describes the state of contemporary open source reliable multicast solutions and reveals deficiencies regarding their use for massive data transport in Content Delivery Networks (CDN). A performance evaluation of the three most popular open-source implementations - *UDP-based File Transport Protocol, NACK-oriented Reliable Multicast* and *Pragmatic General Multicast* in multi-gigabit IP-based networks was performed in the 10Gigabit-WAN laboratory of the Communications Group of Anhalt University of Applied Sciences. This evaluation was completed under the real-world scenario of heavy-weight content distribution in Wide Area Networks. The performance evaluation presented in this paper reveals bottlenecks and deficiencies in current approaches and the paper proposes ideas for improvements and further development of the reliable multicast data delivery family. The defined test scenario was limited to three recipients for the following two reasons: Big data distribution does not imply a large number of recipients, and the goal of this work was to determine upper performance bounds even in a quite simple scenario as a starting point for further investigations. This investigation identified three main challenges: congestion control, losses recovery management and send/receive buffer management. The investigations presented have been performed in the course of a research project in which reliable point-to-multipoint IP-based data transport solution will be proposed. The goal is to achieve data rates of up to 1 Gbit/s per stream with up to ten simultaneous streams from one content server, even in presence of high RTT delays and packet losses in the network.**

Keywords-CDN; reliable multicast; network performance; cloud computing; big data

I. INTRODUCTION

According to a report of the IEEE Ethernet Working Group in [1], in the time period from 2013 to 2018, world traffic will grow by a factor of ten in comparison to the 2010 value. Such a rapid growth in network traffic means improving existing networking technologies and seeking new approaches to data distribution in the core IP network. Challenges such as effective utilization of available bandwidth become crucial. One of the technologies that addresses this issue is multicast networking [2].

In general, the idea of reliable multicast networking aims at achieving maximum utilization of bandwidth whilst avoiding unnecessary duplication of data. In classic unicast networking, each IP packet is sent by a host to exactly one recipient. In the case of multicast networking, data sources deal with groups of recipients and always send only one packet to the entire group. The packets are then duplicated by intermediate network devices such as IP routers and switches. This packet duplication is only performed when the network device knows that it is no longer possible to use one packet for the entire recipient group. Consequently, on all common parts of a network path between a sender and a receiver, the number of packet duplications is minimized.

First standardized in 1986, IP multicast protocols were originally an unreliable data transport solution [2]. One of the first worldwide multicast implementations was *Mbone* [3] with its multicast protocol family such as IGMP or PIM, released in the early 1990s. This protocol family was fairly well adapted to the needs of multimedia applications such as conferencing and live messaging, and, for a long time, multimedia communication was the only application of multicast data transmission. However current use of multicast communication has significantly widened. With the rapid growth of the amount of Internet traffic around the globe, simultaneous point-to-multipoint data delivery is becoming crucial in large Content Delivery Networks (CDNs) and cloud infrastructures. Therefore, distribution of large amounts of content is an ongoing task for most large CDNs. Replications of databases, HD-video delivery, online gaming etc. require high network performance and transmission efficiency. For example, Felix Baumgartner's recent ultrasonic jump was watched in nearly real-time by more than 8 million people; a world record for the number of simultaneous video streams.

Such simultaneous data delivery is one of the big challenges in reliable multicast networking. Raising efficiency of content distribution within CDNs is one of the purposes of reliable multicast communication. For example, the Akamai CDN uses IP multicast technology to provide subscription-based media streaming for consumers. The Amazon cloud constantly receives customers' requests to enable multicast on its EC2 clouds and is currently planning to implement it. The emergence of enormous online gaming services such as the PlayStation Network with over 90 million [4] connected unique consoles (members) must also maintain reliable multicast sessions. These cases clearly demonstrate the need for modern networking in terms of transmission session management for multiple recipients.

This paper is organized as follows. Section I gave an introduction to the research field. Section II gives a brief description of the setup for testing the performance

ISBN: 978-1-61208-256-1

measurements of the selected multicast solutions. Section III describes the approaches of the evaluated protocols and presents data related to the results. Section IV describes the revealed deficiencies and Section V proposes an improvements' plan for the solutions considered. In Section VI, we conclude and propose an agenda for further investigations and implementation of higher-speed reliable multicast data transport.

II. Testbed Definition

The chosen testbed is based on facilities of the 10 Gbit/s test lab installed at Anhalt University of Applied Sciences in Koethen, Germany. All test cases were performed on 64-bit OpenSuSE Linux PC systems. The test network comprises one sending server and three recipients that belong to one multicast group. The work was performed using only three recipients because the ultimate goal of tests was to evaluate upper maximal data rate limit for current reliable multicast approaches. Because this was the main goal, there was no sense in deploying a larger and more complex topology with multiple recipients at this stage. The test network is very well-scalable due to the use of the hardware-based 10G WAN impairment emulator Netropy 10G [5], with which a total data throughput of up to 21 Gbit/s can be achieved. With this device, WAN-sized networks can be emulated and network parameters of the emulated channels - delay, jitter, packet loss – adjusted with an accuracy of about 20 ns. Our test scenarios assume a transmission of one 10 Gbyte file over the emulated WAN to the tree recipients under different network conditions, whereby Round Trip Time (RTT) and packet loss rates are increased up to 50 ms and 0.3% respectively. The topology of the test setup is shown in Figure 1. In general, this topology assumes inhomogeneous delays among emulated links. However, for current tests, we emulated a simplified case with similar RTT values and packet loss rates on each emulated link.

III. Protocols Overview

The following three solutions were considered: *NACK-oriented Reliable Multicast (NORM)*, *UDP-based File Transport Protocol (UFTP)*, and *Pragmatic General Multicast* (PGM). These solutions were chosen due to their high popularity and the availability of a ready to use transport application built upon the respective reliable multicast transport protocol. PGM does not contribute a ready-to-use data transport application, though the protocol stack is used in different production environments such as the one at TIBCO [6], which uses PGM for discovering new members of computing cluster.

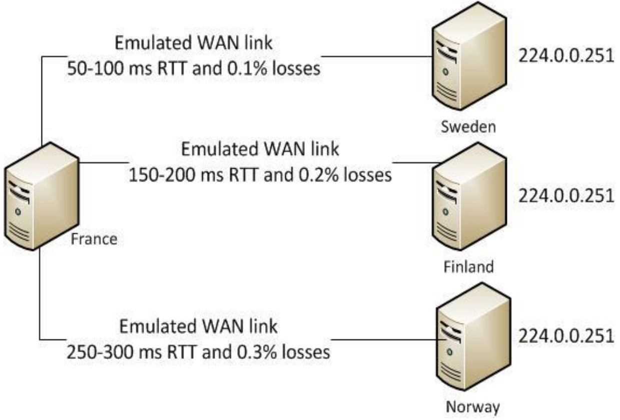

Figure 1. HSA test installation

A. NACK-Oriented Reliable Multicast (NORM)

The *NORM* protocol was defined within RFC 5740 [7] in 2009. The source code of a reference implementation of NORM is maintained by the Naval Research Laboratory [8]. As well as being a transport protocol, the protocol provides a ready-to-use application that can be compiled from available C source code on Linux. Based on Berkeley UDP sockets, the NORM application offers features such as TCP friendly congestion control, which provides fair sharing of available bandwidth between multiple data streams. NORM uses selective negative acknowledgements (NACKs) to provide reliability. *NORM* can also be used in conjunction with Forward Error Correction (FEC), which is currently only an on-demand feature.

As shown in Figure 2, the data rate decreases fast even with very few impairments to the link. This rapid decrease means that the NORM maximal data rate is very sensitive to retransmissions caused by network losses. According to protocol specification, users have to enable FEC to minimize the amount of active NACKs in the network. It also means that the NORM algorithm does not focus on the improvement of NACK management efficiency. Instead, it focuses on improving reliability through the FEC mechanism. However, the problem is that FEC is a difficult approach for big data transmission because there are huge increases in the FEC overhead, even on links in good condition. FEC redundancy issues will be discussed in more detail in Section IV.A. NACK-based reliability, as implemented by default in NORM, enables the receiver to send NACKs at any time – in fact, as soon as a loss has been detected. For bulk data transmission, this causes an enormous batch of NACKs

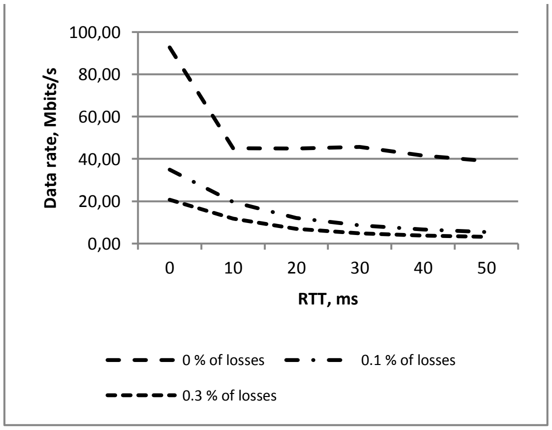

Figure 2. NORM performance dependency on RTT and losses

ISBN: 978-1-61208-256-1

And, therefore, leads to a decrease in data rate as well. Consequently, protocol parameterization, as currently used in NORM, requires significant tuning to raise transport data rate. NORM’s RFC would allow a suitable configuration of the transmission settings e.g. by using minimal inter-NACKS intervals or by consolidating NACKs from multiple packets into one packet.

B. *UDP-Based File Transfer Protocol (UFTP)*

UFTP is also a reliable multicast protocol with a corresponding end-user application and can be considered as a successor to the *Starburst Multicast FTP* (*MFTP)* [9] proposed in 2004. It provides reliable multicast file transfer through UDP transport. The protocol is currently in use in the production of the Wall Street Journal for transporting WSJ pages to their remote printing plants via satellite [9].

UFTP uses a specific scheme of data transmission organization. First of all, the protocol decides how to divide input data into data sets. Input data are split into blocks, whereby one block is always sent within one UDP packet. Since these blocks are, in turn, logically grouped into sections, the sender just sends a section to a multicast group. As soon as the transmission of a section is finished, the sender requests the current status of received data from each multicast receiver and receives a batch of packets containing a list of the packets missing at the site of each recipient. On reception of all NACKs, missed blocks are retransmitted in the unicast way to the requesting recipient. The sender begins to transmit a new section only after all the recipients in the multicast group have confirmed the reception of all blocks of the previous section. This type of data transmission organization results in protocol performance being significantly increased compared to *NORM*. Figure 3 shows the data rate evaluation results in the same testbed as for *NORM*. The results reveal that UFTP has a high loss tolerance and that recovery of lost packets does not reduce the overall data rate as significantly as does *NORM*. However, in both cases, a significant data rate reduction with an increased RTT can be observed.

The obtained results reveal a significantly more efficient sections-based data transmission method than the classic one in the NORM (NACK packet if reception failure revealed). However, when packet loss occurs, the long retransmission periods mean that section-based acknowledgment shows significant dependency on the RTT. Also, transmitting data in this way represents NACKs consolidation, since the UFTP receiver sends NACKs that contain information about multiple missed packets.

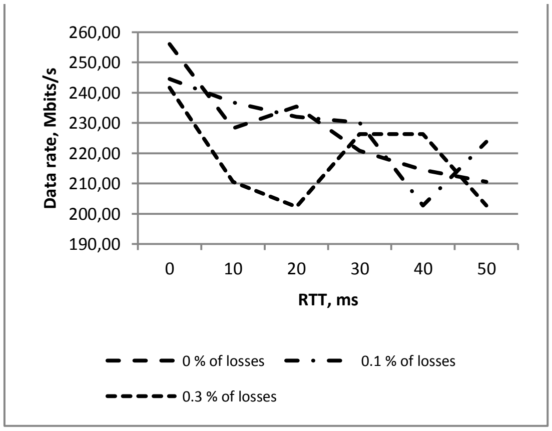

Figure 3. UFTP performance dependency on RTT and losses

C. *Pragmatic Reliable Multicast (PGM)*

The Pragmatic Reliable Multicast (*PGM)* protocol is described in RFC 3208 [10] and is officially supported by IP routers of Cisco Systems (beginning from Cisco IOS Software Releases 12.0 T). This protocol has been developed with the ultimate goal of providing reliable data transmission service for as many recipients as possible. This design automatically means dispensing with ACKs in favor of NACKs, since using ACKs implosion [11] significantly reduces the scalability of the end application and the entire protocol. The retransmission window has to be defined by the user within the configuration of the reliable multicast session. *PGM* assumes allocated disk space (in the form of a buffer) as a window size with a default value of 10 MB. As an option, the retransmission window size can be configured for dynamic adjustment based on NACK-silence times. *PGM* operates over classic IP multicast stack and does not deal with group management, delegating this tasks directly to IGMP, By comparison, the previously described protocols deal with group instances themselves. So, PGM works as a superstructure (in form of raw socket), over UDP and IP multicast stack.

An open source implementation of *PGM* is *openPGM*, which is a framework for the development of new reliable multicast applications. Since there is no ready-to-use *openPGM* application, we had to develop our own test application for sending and receiving files via *PGM*. Through contact with a *PGM* development and maintenance team, we were told that *openPGM* is not designed to be a file transfer protocol. Suggestions for adapting *openPGM* for big data transmission depended on using FEC and Lower-Density-Parity-Codes (LDPC). However it was important for us to get some exact values on possible data rate with the *openPGM* solution. Simple tests revealed an end-to-end data throughput of 27.1 Mbits/s without the packet loss and emulated packet delays in the 1-to-3 multicast scenario that had been found in the two previous tests. Even on RTTs of greater than 10 ms, the data transmission almost stalls. Initially, the idea of the protocol was to multicast very short data blocks such as market quotes and trades. Because we had very specific demands on big data transmission, we decided not to perform further exhaustive tests with openPGM. However, for our research agenda, the protocol provides interesting algorithms and possibilities for session management and dealing with NACKs. This information could be valuable for future work, at least in terms of quick NACKs processing.

IV. PROBLEMS AND SOLUTIONS

Summarizing, it can be stated that on networks with no packet loss and with low round trip delays of up to 20 ms, UFTP provides reliable data delivery in a multicast fashion with up to 250 Mbit/s. However, the data rate is significantly impacted by increasing network impairments such as delay and, especially, packet loss. For a further increase of data transfer rates using multicast, at least three significant problems must be addressed:

A. Congestion control schemes used for the rate control
B. Improvement of packet loss recovery algorithms
C. Send and receive buffer management (Section V)

A. *Congestion control*

Regarding congestion control, the main consideration is whether the receiver or the sender should be responsible for congestion control. For instance, the *Source Adaptive Multi-layered Multicast* (SOAMM) [12] algorithm proposes adjusting video encoding settings at source as a reaction to continuous congestion control feedback. *Receiver-driven Layered Control* (RLC) [12] represents receiver based congestion control. It functions completely source-independent and a participant joins the multicast group accordingly to its own available resources. Such an approach assumes multi-layered multicast with different subscription levels. Available subscriptions - which in IP multicast refers to a multicast group - are to be advertised by the sender, which uses special Synchronization Points (SPs) for this purpose.

Another important challenge here is how to decide between window-based and rate based congestion control schemas.

Due to scalability issues, the classic idea of window-based schemes, such as ones used in TCP, do not fit the requirements of modern reliable multicast communications. With increasing multicast group size, the probability of an acknowledgements' implosion problem also rises. Such an implosion can itself significantly slow down a multicast session. In this scenario, bottlenecks will be on the sender site, and this effect is known as "crying baby problem" [13]. Due to the mentioned ACKs implosion in window-based schemes, most of the contemporary reliable multicast implementations deal with rate-based schemes. However, there is a big difference in comparison to the unicast case. The system of metrics used in a reliable unicast transmission with rate based congestion control is fairly easy - upper data rate limit and appropriate adoption of the data rate on ARQ. However, in multicast transmission, we deal with fairly difficult network paths with couples of branches in which we have to evaluate the entire pattern of multicast tree efficiency. For this purpose, at least two special prediction metrics are proposed [14]:

1. Analysis of multicast tree shape with computation of graph edges weights.
2. Group size as a determining factor [15].

B. *Error recovery*

Three basic schemes are widely used for error correction today:

1. ARQ-based ones with acknowledgements of received data packets, retransmission schemes and timers for retransmissions.
2. The well-known FEC schemes with redundancy in each data packet
3. Error Resilient Source Coding (ERSC), which, in fact, just conceal losses at the receiver site.

Each scheme is used in special cases. Thus, ARQ-based schemes are mainly targeted at delaying insensitive applications, while FEC is mainly used in delay-sensitive applications. It is worth noticing that FEC could be implemented in two different ways: redundant symbols are either transmitted in a separate data packet or within regular data packets. However, for redundancy reasons, FEC is often disabled or even not implemented in contemporary multicast protocols, since redundant packets often make transmission very bulky. Since packet losses in packet-switched networks come in bursts and affect hundreds or thousands of packets, the FEC algorithm will generate so much redundant data that it will aggravate network conditions. As shown in [16], even at a link with a 0.1 % loss rate, the number of required redundant symbols grows exponentially. This result was found for HDTV streaming with a data rate of 1.5 Gbits/s. This work was done as a laboratory case, while real-environment conditions assume loss rates of up to 5% for intercontinental links [17]. The most popular codes for FEC are the Red-Solomon Code and the Tornado Code [16]. ERSC, in turn, is well suited to live video streaming but does not provide full reliability for each sent bit and is, therefore, not suitable for static data transmission.

V. Improvements Plan

The problems and findings described in Section IV point to ideas for optimizing existing solutions and developing entirely new solutions for big data transmission.

In the future, we initially propose dealing with effective data transmission; for instance, by separating entire data array by packets with a further grouping of packets to sections, similar to UFTP implementation with a fairly high upper data rate limit. This mechanism would work fairly well with a buffering of NACKs. The problem of NACKs buffering was initially raised in RFC 3269 [17]. NACKs buffering was aimed at minimizing the amount of NACKs in the network without increasing transmission latency. This challenge is like "walking on the razor's edge", but we are convinced that exhaustive tests and precise adjustments will help us find the most effective NACKs' buffer size.

In the field of congestion control, we are working on multicast-adapted rate-based congestion control with the prediction of network behavior by defined metrics (tree shape and group size).

The error recovery scheme shall be kept NACK-based in order to avoid the ACK implosion problem. We have also decided to dispense with FEC due to the high FEC overhead when losses come in bursts.

As shown in [18], losses in L2 and L3 caused by buffers' overflow prevail over BER-caused losses. For efficient buffer management implementation, we propose designing a novel send and receive buffer implementation adapted to reliable multicast constraints. We are planning to reach data rates of 1Gbit/s per stream in presence of up to 10 destinations within a session. At such high data rates, the ability to read and write data in the most effective fashion becomes crucial. Generally, the idea is to assume dynamic memory allocation for each stream with further re-allocation of available memory among other streams.

VI. Conclusion

A general overview of contemporary reliable multicast implementations is given in the paper. Our research reveals that, even with quite a small number of recipients, the upper limit of throughput on reliable data transport is currently not more than 250 Mbit/s. Performance results also revealed that packet loss causes the most significant decrease of transmission data rate. Thus, future work will focus more on improving error recovery schemes. Analysis of considered protocols revealed possible algorithm improvement for raising data rate performance. Our work reveals a few trends that could potentially be implemented in a reliable multicast scheme with the primary goal of achieving a data rate of 1 Gbit/s per reliable stream with at least up to 10 destinations.

References

[1] IEEE 802.3 Ethernet Working Group, "IEEE Industry Connections Ethernet Bandwidth Assessment". San Diego : IEEE 802.3 Plenary meeting, 2012.

[2] S. Deering, "Host Extensions for IP Multicasting", RFC-1112, 1989.

[3] K. Savetz, N. Randall, and Y. Lepage, "MBONE: Multicasting Tomorrow's Internet", John Wiley & Sons Inc (Computers), ISBN 978-1568847238, 1996.

[4] A. Osborn, "Number of Registered PlayStation Network Accounts Reaches 90 Million". March, 2012, http://www.playstationlifestyle.net/2012/03/07/number-of-registered-playstation-network-accounts-reaches-90-million/. [retrieved: January 2013]

[5] Apposite Technologies oficial web site, http://www.apposite-tech.com/index.html. [retrieved: January 2013]

[6] TIBCO official web site, http://www.tibco.com/. [retrieved: January 2013]

[7] B. Adamson and C. Bormann, "NACK-Oriented Reliable Multicast (NORM) Transport Protocol", RFC-5740, 2009.

[8] Naval Research Laboratory, http://www.nrl.navy.mil/. [retrieved: January 2013]

[9] K. Miller and K. Robertson, "StarBurst Multicast File Transfer Protocol (MFTP) specification", Internet-draft, 1997.

[10] T. Speakman and J. Crowcroft, "PGM Reliable Transport Protocol Specification" RFC-3208, 2001.

[11] D. S. Vijayakumar and S. V. Ram, "A network processor implementation for solving the ACK implosion problem", ACM-SE 44. Proceedings of the 44th annual Southeast regional conference, New York, March, 2006, pp. 732-733.

[12] D. Constantinescu, D. Erman, and D. Ilie, "Congestion and Error Control in Overlay Networks", Blekinge Institute of Technology, Sweden, 2007.

[13] H. Holbrook, S. Singhal, and D. R. Cheriton, "Log-Based Receiver-Reliable Multicast for Distributed Interactive Simulation". ACM SIG-COMM-95, Cambridge, August, 1995, pp. 328-341.

[14] R. C. Chalmers and K. C. Almeroth, "Developing a Multicast Metric. Global Telecommunications Conference", IEEE GLOBECOM '00, San Francisco, December, 2000, pp. 382-386.

[15] J. Chuang and M. Sirbu, "Pricing multicast communication: A cost based approach", INET’98, Geneva, July, 1998, pp. 281-297.

[16] S. Senda, H. Masuyama, S. Kasahara, and Y. Takahashi, "FEC Performance in Large File Transfer over Bursty Channels", Proceedings of the 4th International Working Conference on Performance Modelling and. Evaluation of Heterogeneous Networks (HET-NETs'06), D. Kouvatsos (ed), West Yorkshire, U.K., September 10-13, 2006, P07/1-10.

[17] Y. Angela Wang, C. Huang, J. Li, and K. W. Ross, "Queen: Estimating Packet Loss Rate between Arbitrary Internet Hosts". Proceedings of the 10th International Conference on Passive and Active Network Measurement, April, 2009, pp. 57-66.

[18] S. Dixit and T. Wu, "Content Networking in the Mobile Internet". s.l. : John Wiley and Sons, Inc., ISBN 0-471-46618-2, 2004.

[19] R. Kermode and L. Vicisano, "Author Guidelines for Reliable Multicast Transport (RMT) Building Blocks and Protocol Instantiation documents", RFC-3269, 2002.

[20] E. Siemens, R. Einhorn, and A. Aust, "Multi-Gigabit Challenges: Similarities between Scientific Environments and Media Production". ACIT - Information and Communication Technology, June, 2010.