УДК

ACTUAL APPROACHES FOR - MULTICAST-BASED RELIABLE DATA TRANSPORT AND THEIR DEFICIENCIES

A.Bakharev; E.Siemens

Siberian State University of Telecommunication and Informatics Sciences (SibSUTIS)
630102 Russia, Novosibirsk Kirova street 86
Anhalt University of Applied Sciences (HSA)
06366 Germany, Köthen Bernburger Str.57
E-mail: a.bakharev@emw.hs-anhalt.de; e.siemens@emw.hs-anhalt.de

This paper describes the state of the art of reliable multicast data transport in the light of contemporary changing Internet technology requirements. Emerging cloud computing applications, distributed workflows with massive data transport demands and global services infrastructures require novel approaches in field of content distribution. So global infrastructures, like *Akamai CDN* require optimization of data transmission means across its networks. Multicast can potentially bring significant improvements in content distribution technologies - e.g. optimization of bandwidth utilization and improvement of interactivity. In this paper an overview of currently widely used multicast approaches is given along with the performance and usability constraints in its usability in contemporary CDN and cloud computing environments. A general overview of some additional protocols is given, which appear as a good basis for further work on multicast, but which are not yet ready to use widely. Also, new ideas for improvement and new investigation directions in reliable multicast approaches are proposed. Finally, most significant topics, at which improvements in the reliable multicast data transport have to be applied and three main issues are pointed out – there are: congestion control, offering of pure *time-bounded reliability* and an efficient sending and receiving buffer management for multicast protocol stacks.

Key words:

1. Multicast networking

Originally, IP multicast protocols have been designed as pure unreliable data transport solutions and were first standardized in 1986 [1]. One of the first worldwide unreliable multicast implementations was *Mbone* [2] with its affiliated stack of IP multicast protocols like IGMP, PIM, released in early 1990s. These protocol family fit pretty well needs of applications like multimedia conferencing, messaging and real time loss-tolerant applications. For a long time it was the only approach of multicast applications.

With rapid expansion of the internet community and the emergence of grid applications and later on cloud services, requirements for one-to-many networked data transport solutions, usually based on multicast, have caused a focus change of multicast data transport solutions. In these use cases users expect to get a high quality service within online conferences and presentations, high data rates while content distribution and essentially a bitwise copy of source data at each destination of the multicast tree. All these aspects assume existence of reliable transport, which was not proposed in multicast field for a long time. However, the unreliable approaches didn't fit these requirements. As shown in [3] three types or reliability are actually required:

- ***Total reliability***
- ***Semi-reliability***
- ***Time-bounded reliability***

Total reliability assumes that 100% of sent data will be delivered to all recipients and no one bit will be loosed. Hereby, the order of data transmission and delay as jitter of end-to-end data transport is irrelevant. This approach works pretty well in the area of file transport.

Semi-reliability offers retransmission of some of missed packets in combination with *Forward Error Correction* (FEC). As described in RFC 5740 [4], sender will determine critical erasure-filling needs for each sent block of data. Then, if sender will decide that error could be smoothed with FEC redundant blocks, FEC will be applied in reply to e.g. NACK. If error has too massive character, additionally block retransmission will be performed.

Time-bounded reliability is a specific type of reliability which is suitable for applications with strict jitter requirements. In this case, retransmissions have to be performed within certain, strictly bounded retransmission window, since the end-to-end delay or jitter must not exceed some pre-defined bounds. Examples for application with time-bounded reliability is online video streaming, online news release or and real-time text applications like one used to provide summary of quotations on the stock exchange trading. All these applications deal with information which is valuable only in a very short time range.

Reliability in multicast data transport can be offered and initiated by sender or receiver. So, in general reliable multicast protocols can be classified into ***sender-initiated*** and ***receiver-initiated*** multicast reliability. In the first class, the transport layer is dealing with acknowledgements (ACKs) being sent by the receiver as reply on each successfully received packet. It causes a problem, named ACK implosion, when continuous stream of ACKs locks the network up. In order to overcome this problem, receiver-initiated protocols have been proposed. In this case, reliability is based on negative acknowledgements (NACKs) instead of ACKs. Here, a reliable multicast receiver notifies the sender not about successfully received packets but only about missed ones. This significantly decreases intensity of service traffic within entire network and prevents against the implosion effect caused by the ACK flooding. The transition to receiver-initiated multicast reliability leads to a new challenge of NACK-based repair efficiency. In fact, it assumes buffering of NACKs on the sender site in order to find out the most optimal retransmission way. Calculation of this optimal time is still one of the actually most significant reliable multicast networking challenges, as discussed in RFC 3269 [5].

2. NACK-Oriented Reliable Multicast (NORM)

The *NORM* protocol is described within RFC 5740 [4] in year 2009. The source code of a reference implementation of NORM is maintained by the Naval Research Laboratory. The protocol is based on NACKs, so it is a receiver-initiated reliable multicast protocol. It is fully compatible with both IPv4 and IPv6 and offers ready-to-use application, which can be compiled from available source code. The NORM application, based on typical UDP sockets offers features like TCP friendly congestion control which provides fair sharing of available bandwidth between multiple data streams. *NORM* can also be used in conjunction with FEC, which is actually an on-demand feature. The FEC usage represents *semi-reliability*, described above. If switched on, NORM sends redundant symbols in reply to NACK or within data stream itself, accordingly to chosen option.

NORM source code provides a very flexible application programming interface (API) for networked applications development based on reliable multicast. It operates with four levels of instances: API initialization, Session Creation and Control, Data Transport, API event notification.

The packet loss recovery algorithm of NORM is quite tolerant to RTT in the network, but very sensitive to packets losses. Fig. 1 represents dependency of data rate on RTT and packet loss for *NORM*. Hereby, as well as in the following measurement results, the testbed was represented by one server and three recipients. Data were transmitted via a 1 Gbit-Ethernet with emulated network impairments like packet loss and jitter as well as transmission delay. Emulation was done using a *Netropy 10G* impairment emulator. The overall amount of transmitted data was in all tests 10 GBytes.

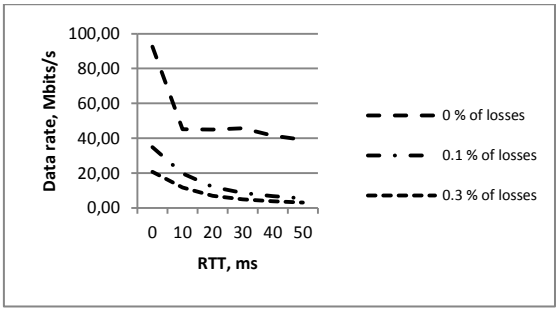

Fig. 1. Dependency of *NORM* data rate on RTT and packet losses

3. UDP-based file transfer protocol with multicast (UFTP)

UFTP is a reliable multicast protocol as well as correspondent end-user application and can be considered as a successor of *Starburst Multicast FTP* (*MFTP)* [6] proposed in 2004 and offering reliable multicast file transfer by means of typical UDP transport. The protocol is currently in use in production of the Wall Street Journal to send WSJ pages over satellite to their remote printing plants [7].

UFTP uses specific scheme of data transmission organization. First of all, the protocol decides how to divide input data set. It is going to be divided by blocks (one block is always sent within one UDP packet), while blocks, in turn, going to be grouped into sections. Afterwards, the sender just sends a section to multicast group. As soon transmission of a section is finished, the sender requests current status of received data from each multicast receiver and gets a batch of NACKs from recipients. On reception of all NACKs, missed blocks are retransmitted in a unicast way. A new section will begin only after the reception of all blocks of the previous section at each recipient in the multicast group. Such data transmission organization leads to significant increase of protocol performance compared to *NORM*. Data rate evaluation results for the same testbed as for *NORM*, showcased at Fig. 2.

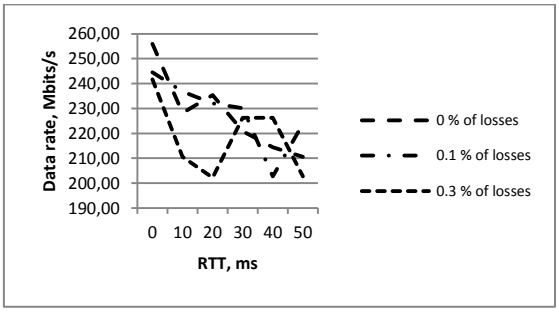

Fig. 2. Dependency of *UFTP* data rate on RTT and packet losses

Obtained results reveal that UFTP is quite loss tolerant protocol and recovery of lost packets does not lead to significant reduction of the overall data rate as NORM does. However, in both cases, a significant data rate reduction with an increased RTT can be observed.

4. Pragmatic General Multicast (PGM)

The *PGM* protocol is described within RFC 3208 [8]. This protocol has been developed with the ultimate goal to provide reliable data transmission service for as many recipients as possible. This design focus leads to the necessity to dispense with ACKs in favor of NACKs, while using ACKs as a mechanism of reliability significantly reduces scalability of end application and entire protocol due to the well-known problem with ACK implosion. *PGM* provides *time-bounded reliability*. Time-bounded reliability assumes reliability within some certain retransmission window. The retransmission window has to be defined by the user within the configuration of the reliable multicast session. As an option, window size can be configured for automatic adjustment based on NAK-silence. *PGM* operates over classic IP multicast stack and does not deal with group management and delegates this tasks directly to IGMP, while previously described protocols deal with group instances on themselves and are able to manage it. So, it works as superstructure (in form of raw socket), over UDP and IP multicast. Since *PGM* is intended to operate with a time-bounded reliability, it accelerates sending of NACKs as much as possible in order to get rid of unrecoverable losses.

An open source implementation of *PGM* is *openPGM*, which in fact is a framework for development of new reliable multicast applications. openPGM does not provide a ready-to-use application, however it gives a lot of development opportunities. Since any performance evaluation within this paper has been performed with ready-to-use applications, performance results haven't been compiled in the same lab environment. Instead, we use the performance results, presented in the press release of *MIRU development studio* [9]. According to this, openPGM test application offers a maximum sustained data rate of about 540 Mbits/s on a shortcut - without network delay and packet

loss and jitter. Investigation of the behavior of openPGM in presence of packet losses and high packet delay is subject to further work of our group.

5. Other approaches and protocols

Some other contemporary approaches besides the described ones are also to be mentioned. Of interest is *Reliable Data Center Multicast* (RDCM) [10], proposed by Microsoft Research Asia. It offers reliable multicast service for local environments with high link density and so predestined for extended local area network environments (e.g. data centers and metro nets). Non-typical feature of *RDCM* is ability to organize retransmissions of missed packets by means of neighbor recipients. This feature in fact causes a strict requirement regarding the link density – so the receivers must be located geographically close to each other. Interesting approach in RDCM is that acknowledgments are sent over multicast in order to notify each member about missed packet and to find out, which intermediate of a particular end node (recipient) has to retransmit missed data to recipient by means of unicast, in turn. This protocol is not available as open source project, so there was no ability to test it in own lab environment.

Another solution is *Reliable Overlay Multicast with Loosely Coupled TCP Connections (ROMA)* [11], proposed by Boston University. ROMA uses a concept, which fundamentally differs from the abovementioned solutions. It deals with TCP instead of UDP, some performance measurements results are published in [11]. Bandwidth of *ROMA*, in accordance with tests, performed by inventors, is not more than 98 Mbits/s. This protocol is also not available publically, so it could not be tested in here.

The *Scalable Reliable Multicast* framework *(SRM)* [12], proposed in year 1995 and proved as a protocol for serving light-weight reliable multicast sessions, like networked whiteboard application. This approach is out of interest here, since it initially has been designed as a solution for maintaining of light-weight sessions with all consequences like focus on huge number of recipients instead of achieving high data rates.

One more approach is the *Reliable Multicast Transport Protocol (RMTP)* [13] proposed by *Alcatel-Lucent*, which assumes to have a set of intermediate designated receivers across the network in order to minimize probability of ACKs implosion. RMTP proposed in year 1997 and not maintained any more for about a decade.

6. Contemporary deficiencies and fields for algorithm improvements

Most of the reliable multicast protocols discussed here, have been proposed before year 2005 and so are designed for relatively low data rates, that are is nowadays insufficient. Current implementations of these protocols do not fit requirements of contemporary content delivery networks and cloud infrastructures. Contemporary CDNs assume distribution of massive data sets with the amount of data of units of up to Petabytes like in case of Energy Sciences Network [14]. The CDN of *Akamai*, the world largest content delivery network provider, which claims to serve about 20% of world-wide web traffic [15] by means of 100 000 of servers around the world, what again represents data sets which contemporary CDNs have to deal with. In accordance with *Akamai* technical publication of year 2010 [16], delivery of 4 GB DVD image with unicast transmission nowadays takes about 2.2 hours in regional network (800-1600 kilometers). More detailed dependency of content delivery time on distance is shown on Fig. 3. The test setup of Akamai hereby is transmission of 4 GB DVD image by means of unicast transmission, not multicast. A packet loss rate varies in the range of 0.6 % to 1.4 %, depending on a distance. As a comparison, the network distance between Moscow and New York is about 12 400 kilometers, and delivery of 10 GB of content over *Akamai* network takes significantly more than 30 hours.

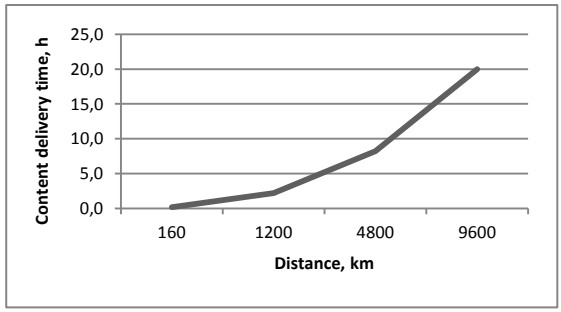

Fig. 3. Dependency of content delivery time on distance in *Akamai* CDN network.

Obviously, fulfilling requirements of contemporary point-to-multipoint reliable applications is still a wide field of research, and also a matter of development of new algorithms, optimization of old ones and proposing of fundamentally new reliable multicast transport protocols.

Analysis of considered reliable multicast approaches reveals some causes of performance weaknesses of each protocol. openPGM is potentially the fastest and the most effective solution, since it can deal with scenarios with active participation of intermediate nodes. Such scenarios assume that intermediate nodes will also deal with NACKs and be responsible for further retransmissions. It significantly reduces batch of NACKs which sender have to serve and amount of sender-retransmitted packets as well. In fact, this approach is a combination of generic approaches of both *RDCM* and *SRM*.

UFTP achieves relatively high data rates due to non-standard scheme of section-based data transmission. Also, it uses new schemes of retransmission, when retransmission is going to be performed not in context of certain packet, but in context of missed blocks from some certain session, size of each is calculated in accordance with each certain case.

NORM showcases the lowest performance among considered protocols. This can be explained by a very generic approach of the protocol. The main feature is the exploitation of FEC, but it is not really applicable well to file transmissions. Non-effective scheme of congestion control significantly reduces resulting data rate and, in fact, moves this protocol out of file transfer area.

The statements above point to the fact that some significant changes must be applied to the matter of reliable multicast data transport. We pointed out three significant fields of investigation here: congestion control, problems of time-bounded reliability and structure of sending and receiving buffer which leads to issues of software design for handling of high-speed point-to-multipoint (mostly based on multicast sessions) data transport within computing and operation systems.

Considering nowadays congestion control in reliable multicast approaches, it is easy to note imperfection of proposed algorithms. Regarding *NORM*, there is a problem of significant data rate reducing if congestion control is enabled. In fact, TCP friendly congestion control of *NORM* reduces maximal data transmission rate by the factor two. Valuable deficiency of *UFTP* protocol is manual configuration of congestion control in form of additional configuration file. It contents percentage of received NACKs in relation to number of sent packets. For each percent, it is possible to specify coefficient which resulting data transmission rate will be obtained with. Obviously that coefficient could be grater that “1” and at low percentage of NACKs, data rate could be increased. Generally speaking this is very flexible approach, but it pushes users to adjust configuration of congestion control for each environment and certain network conditions, what is very helpful in the phase of protocol development, but not usable for ready-to-use applications.

Coming back to three types of reliability, explained in section 1 and comparing with actual CDN- and cloud requirements and up-to-date trends, apparently, *time-bounded reliability* will become main trend in reliable multicast communications for next years. Jitter sensitive applications, e.g. online translations are very popular and take significant part of nowadays network approaches. *openPGM* focuses on the goal to deal a lot with *time-bounded reliability*. On this matter, it is necessary to have a very effective scheme of NACKs sending. NACKs should be sent to sender in a shortest time in order to notify sender about missed packet within retransmission window, which in turn should be as narrow as possible. Matter of retransmission windows width is another challenge in reliable multicast communications. In general, it should be adjusted in accordance with current transmission conditions in order to construct the possibly best stream of real-time traffic.

The third challenge, which has to be disclosed, is management and algorithm of sending and receiving buffer. At high data rates it will be quite valuable feature, to write and read information at the most effective and fast way. In case of multicast communications, buffer structure and mechanism of reading and writing of data have to be more complex, comparing to unicast. Dealing with NACKs buffering, described in section 1, dynamic adjustment of retransmission window will bring more and more complexity to buffer implementation.

7. Conclusion

The attention of internet community to reliable or even simple multicasting has been apparently reduced within last 10 years, but in this paper we have presented, that present-day networks need significant new ideas regarding content distribution and reliable real-time streaming. One of valuable and fundamental achievements for the last years in reliable multicasting is dispensing with ACKs because of strict limitations in protocols scalability. Relying on known facts and performed measurements, presented here, it is revealed that data rate limit for now is not more than 600 Mbits/s on shortcut connection, and *UFTP* shows not more than 260 Mbits/s in LAN over loss-free network links. All these numbers have been measured in local lab environments, while real-life rates showcased here are coming from *Akamai's* technical publications, according to which the actual data rate in contemporary CDN is not more than 44 Mbits/s in case of unicast. It is stable and working production environment, but not enough for fitting nowadays society needs. Set of interrelated changes and adjustments have to be done for known algorithms in order to achieve new level of real-time data transmission. Potentially, *RDCM* should provide significantly higher data rate, but it has strict requirements for links density and, other words, applicable only in closed environments of high performance data centers. Three trends of improvements are distinguished: congestion control, time-bounded reliability and sending and receiving buffer management within the senders and receivers of multicast sessions.

REFERENCES

[1] "IP Multicast," Available: http://en.wikipedia.org/wiki/IP_multicast. [Accessed 5 April 2012].

[2] "Mbone," Available: http://en.wikipedia.org/wiki/Mbone. [Accessed 10 April 2012].

[3] C. Coskun, "PERFORMANCE ANALYSIS OF RELIABLE MULTICAST, Master thesis," 2004. Available: http://etd.lib.metu.edu.tr/upload/12605656/index.pdf.

[4] "RFC-5740, NACK-Oriented Reliable Multicast (NORM) Transport Protocol," Available: http://tools.ietf.org/html/rfc5740.

[5] "Author Guidelines for Reliable Multicast Transport (RMT) Building Blocks," Available: http://tools.ietf.org/html/rfc3269.

[6] "Internet-draft, StarBurst Multicast File Transfer Protocol (MFTP) Specification," Available: http://tools.ietf.org/html/draft-miller-mftp-spec-02.

[7] "UFTP - Encrypted UDP based FTP with multicast," Available: http://www.tcnj.edu/~bush/uftp.html. [Accessed 10 April 2012].

[8] "RFC-3208, PGM Reliable Transport Protocol Specification," Available: http://tools.ietf.org/html/rfc3208.

[9] "MIRU development studio press release," 2010. Available: http://openpgmdev.blogspot.de/2010/09/miru-announces-openpgm-5.html.

[10] M. Xu, M.-c. Zhao and C. Guo, "RDCM: Reliable data center multicast," *INFOCOM, 2011 Proceedings IEEE*, pp. 56-60, 2011.

[11] G.-I. Kwon, "INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies," in *ROMA: reliable overlay multicast with loosely coupled TCP connections* , 2004.

[12] S. Floyd and V. Jacobson, "A reliable multicast framework for light-weight sessions and application level framing," *IEEE/ACM Transactions on Networking*, vol. Volume 5, no. Issue 6, 1997.

[13] "RMTP: A Reliable Multicast Transport Protocol," Available: http://www.bell-labs.com/project/rmtp/.

[14] "BER Network Requirements Workshop," 2010.

[15] "Content delivery network," Available: http://en.wikipedia.org/wiki/Content_delivery_network. [Accessed 6 April 2012].

[16] E. Nygren, R. K. Sitaraman and J. Sun, "The Akamai Network: A Platform for High-Performance Internet Applications," *ACM SIGOPS Operating Systems Review*, vol. vol.44, 2010.

Authors' information: