Influence of Jitter on Reliable Multicast Data Transmission Rate in Terms of CDN Networks

Aleksandr Bakharev, Eduard Siemens
Anhalt University of Applied Sciences - Department of Electrical, Mechanical and Industrial Engineering
Bernburger Str. 57, 06366 Koethen, Germany
E-mail: a.bakharev@emw.hs-anhalt.de , e.siemens@emw.hs-anhalt.de

***Abstract*— This paper devotes to evaluation of performance bottlenecks and algorithm deficiencies in the area of contemporary reliable multicast networking. Hereby, the impact of packet delay jitter on the end-to-end performance of multicast IP data transport is investigated. A series of tests with two most significant open-source implementations of reliable multicast is performed and analyzed. These are: *UDP-based File Transfer Protocol* (UFTP) and *NACK-oriented Reliable multicast* (NORM). Tests were targeted to simulate scenario of content distribution in WAN – sized Content Delivery Networks (CDN). Then, results were grouped and averaged, by round trip time and packet losses. This enabled us to see jitter influence independently on round trip time (RTT) and packet loss rates. Revealed jitter influence for different network conditions. Confirmed, that appearance of even small jitter causes significant data rate reduction.**

Keywords: jitter;reliable multicast; CDN

I. INTRODUCTION

A. Terms definition

We are defining the term "*jitter*" or "*delay variation*" in accordance to RFC-3393 [1]. Assume that "jitter" or "delay variation" in networking is a difference of one way network delay values for two consecutive packets with respect to some reference value. Practically, it means that: jitter value 10 ms corresponds to fact that RTT in network will vary in range of about 20 ms from certain value. Practical example shown below:

```
64 bytes from 192.168.11.1: icmp_seq=15 ttl=64 time=124 ms
64 bytes from 192.168.11.1: icmp_seq=16 ttl=64 time=150 ms
64 bytes from 192.168.11.1: icmp_seq=17 ttl=64 time=127 ms
64 bytes from 192.168.11.1: icmp_seq=18 ttl=64 time=132 ms
64 bytes from 192.168.11.1: icmp_seq=19 ttl=64 time=106 ms
64 bytes from 192.168.11.1: icmp_seq=20 ttl=64 time=124 ms
```

So, jitter affects inter-arrival gap and consequently leads to two effects, such as *clumping* (inter-arrival gap decrease), and *dispersion* (inter-arrival gap increase).

Main evaluation metric for the transmission performance is the achieved *data rate*. When calculating data rate, assume that data rate is the relation of amount of transmitted data over passed time:

$$R = \frac{A}{T}$$

Where R – data rate in Mbits/s, A – amount of transmitted bits, T – transmission duration in seconds.

It was also stated that results will be *averaged* by two parameters (packet loss rate and RTT). It means that we will have two additional specific types of achieved data rates:

$$R_{RTT\ independent} = \frac{R_{RTT_1} + R_{RTT_2} + \cdots + R_{RTT_i}}{i}$$

$$R_{losses\ independent} = \frac{R_{loss_1} + R_{loss_2} + \cdots + R_{loss_i}}{i}$$

By intention, calculation of e.g. $R_{losses\ independent}$ for different RTT values, we can see how jitter affects network performance without influence of losses on result. This metric is to be introduced due to probability of too high sensitiveness of protocols to RTT or losses. Averaging of results could minimize these effects and give more "pure" results.

B. Background

Jitter appears in any packet switched network and can significantly affect data transmission quality. Main issue is that jitter causes breaking of timing logics in transport protocols. If variation of inter-arrival interval exceeds some certain range, a situation occurs, that packets are considered dropped, though they are arriving just late. Consequently, receiver could be just not ready to accept packet, which is arrived not in time.

In real networks, significant jitter usually appears in cases, when OWD in network is relative high (practically it means long distance between network units). From this fact, occurs the first main reason of jitter – unexpected delays at intermediate devices (routers, switches, e.g.). This could be caused by buffer or performance issues at the intermediate device. In fact, unpredictable delay is the main part of jitter value.

However, jitter is caused not only by queuing within network nodes, but at the sender's side as well. Main source-related reasons of jitter at the data sender are sender's timer and scheduling issues. When using non-precise timers, we could not be sure about initial inter-packet gap and this uncertainty transforms in jitter effect at the receiver's side.

Approaches to minimize jitter influence coming from its reasons. Generally speaking, only what could be done are:

Implementation of *jitter buffers*, like in *Real Time Protocol* (RTP) [2] stacks of VoIP and Video-over IP applications. This is just an additional buffer, which stores certain amount of data packets on arrival and timely transmit them to the receiver, thereby, emulating correct inter-arrival gap for the client.

Adding additional interactivity to communication protocol, as it was done by *TCP-interactive* protocol [3]. Practically it means better analysis of network within data transmission and dynamic adjustment of transmission parameters.

The goal of this paper is to analyze the impact of jitter on actual data rate of reliable IP multicast transport protocols. Based on previous experience with evaluation of the data transport number contemporary reliable multicast approaches [4] and evaluation of multi-gigabit reliable transport protocols [5] we have expected that jitter influence would be not so significant in reliable multicast networks, at least in frame of our experiment, due to relative low data rates.

II. Experiment map

As the test scenario, a delivery of 4.1 Gbytes file among 3 receivers, by means of reliable multicast data transmission has been chosen. Such a scenario assumes quite easy topology and enables to observe more "pure" dependencies in the network due to minimizing of intermediate devices. One more reason for this topology is that our initial goal is to emulate distribution of heavy content in CDN network. Such a scenario often does not assume hundreds of recipients. We are dealing here with only three recipients and it enables us to observe jitter effects in not overloaded network. By means of such network conditions we got rid of additional network issues, such as congestions, receive/send buffers overflow and etc. Network impairements (delay, losses jitter) are managed by Netropy 10G network emulator, manufactored by Apposite Technologies [6]. The network has been set up in 10G laboratory of Anhalt University of Applied Research in Koethen (Germany). Topology is presented at Figure 1.

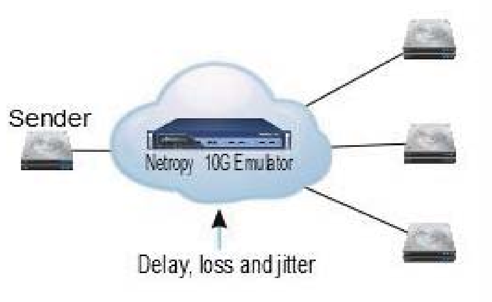

Figure 1. Test network setup

All participants of multicast data transmission session are Linux-based servers. Sender has Centos 6 installed at the Intel i7 machine, while all clients are running Open Suse 11.4 Intel Atom machines. All systems are running 64x editions of OSes. Network impairments have been set to the following values:

RTT: 60, 120 (ms)

Packet loss rate: 0.0, 0.1, 0.7 (%)

Jitter: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 (ms)

Protocols under consideration are *NORM* [7] and *UFTP* [8]. Though in the performance analysis paper [4] openPGM [9] was also taken into account, however initial performance tests have revealed inability to use this protocol for Big Data transmission.

III. Protocols

A. NACK oriented reliable multicast (NORM)

The *NORM* protocol was defined within RFC 5740 [7] in year 2009. The source code of a reference implementation of NORM is maintained by the Naval Research Laboratory [8]. It delivers along with the transport protocol a ready-to-use application, which can be compiled from available C source code on Linux. The NORM application, offers features like TCP-friendly congestion control which provides fair sharing of available bandwidth between multiple data streams. NORM uses selective NACKs to provide reliability. NORM can also be used in conjunction with FEC, which is actually an on-demand feature.

B. UDP-based File Transfer Protocol (UFTP)

UFTP is a reliable multicast protocol as well as a corresponding end-user application and can be considered as a successor of *Starburst Multicast FTP* (*MFTP)* [10] proposed in 2004 and offers 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 [11].

UFTP proposes a specific scheme of data transmission organization. First of all, the protocol decides how to divide an input data set. It is going to be split into blocks whereby one block is always sent within one UDP packet, while blocks, in turn are logically grouped into sections. Divided into blocks and sections, the sender just sends a section to a multicast group. As soon as a transmission of a section is finished, the sender requests current status of received data from each multicast receiver and gets a batch of packets which contain a list of missed packets at the site of each recipient. On reception of all NACKs, missed blocks are retransmitted in the unicast way. The sender will begin to transmit a new section only after the reception of all blocks of the previous section at each recipient in the multicast group.

IV. Results

Previous results (4) showcased that UFTP transfers data much faster than NORM and this fact enables us to observe jitter influence at very different data rates. The result of first run is presented in Figure 2. At the first glance is seen that data rate decrease, caused by appearance of the minimal jitter, is the most significant one. It is going to be referred to the phenomenon of *instant data rate decrease* (firstly introduced in this paper) at the moment when jitter appears – delay variation of 1 ms. This effect is especially visible at plot, corresponding to UFTP. Without jitter, it reaches data rate of more than 200 Mbits/s, but even with 1 ms of delay variation, data rate degrade till 150-175 Mbits/s and afterwards we observed quite low decrease – till 140 Mbits/s at 9 ms of delay variation. The same tendency can be observed with NORM, but it decreases more drastically and at low data rates.

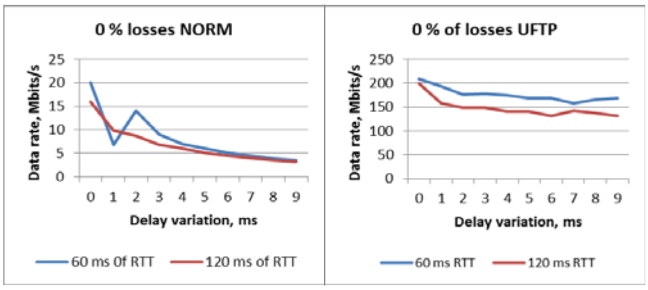

Figure 2. Data rate on delay variation dependency fot link with low loss

Figure 3 extends this exploration. Analysis of Figure 3 points to the fact, that with decreasing of data rate (caused by high delays and losses) data rates become more "stable". We do not see bursts (particularly a significant decrease of data rate). It means that jitter influence decreases with data rate decrease due to other impairments. This is contrastly visible at plot of 0.7 % of losses of the NORM protocol. NORM is pretty sensitive to network impairments, and 120 ms of RTT in conjunction with 0.7 % of losses gives high decrease of data rate (lower than 1 Mbit/s). And exactly at this level, it is obvious that jitter influence almost dissapeared. Table 1 and Table 2 demonstrate some numbers which describe real value of data rate reduction and confirm this statement. For losses-free link there is a reduction by the factor higher than 4. While in high-losses link same parameter is not more than 10% from initial value. Graphicaly, this fact presented at Figure 4.

TABLE I. DATA RATE VALUES, MBITS/S. FOR NORM AT 0.7 % LOSSES LINK WITH 120 MS OF RTT

RTT/ Jitter	0	1	2	3	4	5	6	7	8	9
120 ms	0,80	0,81	0,80	0,82	0,78	0,75	0,74	0,71	0,70	0,70

TABLE II. DATA RATE VALUES, MBITS/S. FOR NORM AT 0.0 % LOSSES LINK WITH 120 MS OF RTT

RTT/ Jitter	0	1	2	3	4	5	6	7	8	9
120 ms	16,00	9,76	8,70	6,84	6,02	5,10	4,52	4,08	3,56	3,25

The same test of UFTP protocol gives somewhat different results. As seen in Figure 5, one sees that all three lines of different packet loss rates are almost parallel. It means that influence of losses on data rate on jitter dependency is minimal for UFTP. Even in links with relatively high losses observe phenomenon of *instant data rate decrease*, mentioned above in the current section. Later on, at Figure 6 slow decrease of data rate take place. In case of UFTP, it was revealed that both, RTT and losses do not significantly influence the way, how jitter affects data transmission rate. However in case of NORM, the absolute value of RTT plays a role, but only on jitter values up to 4 ms. On higher jitter, lines go almost parallels. This dependency is demonstrated in Figure 6.

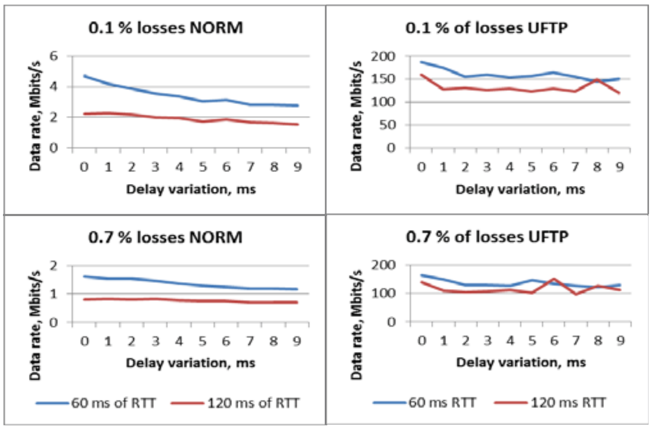

Figure 3. Data rate on delay variation dependency for link with different losses

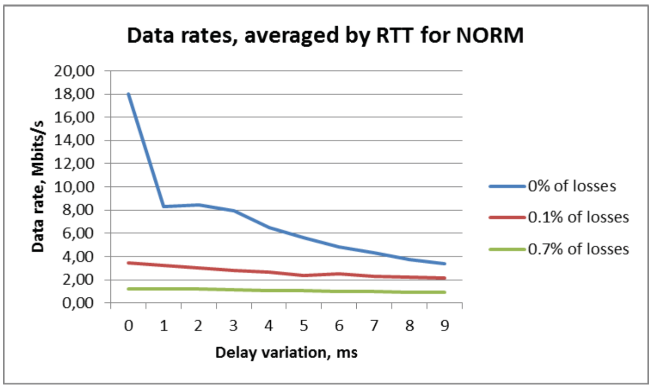

Figure 4. Data rate on delay variation dependency for link with different losses (with averaged value of RTT)

V. CONCLUSION

The reason of *instant data rate decrease* phenomenon is not certainly clear here, however obviously looks like its reasons are in the algorithms of protocols. The fact, that this phenomenon could be observed only at the upper data rate limit of both protocols points to the idea that no one of considered protocols is able to manage jitter issues properly at high data rates. Understanding of this phenomenon going to be discovered in future works of our team.

UFTP behavior looks very stable in any case and is apparently caused by very effective data transmission scheme. Contemporarily, this is the only protocol, which uses consolidated NACKs [12] in a bit concealed manner - division of input data by sections and blocks - within data transmission, and all our tests showcased its efficiency. Providing of reliability is very resource-consuming process and UFTP manages it very effectively. Consolidated NACKs minimizes amount of active NACKs in the network and lead to the provision of reliability in the most effective manner. This is especially crucial in multicast sessions, when sender has to deal with multiple (of even thousands) of receivers. In such scenario, the price of each NACK is very important.

NORM protocol uses the very common NACK-based retransmission scheme and it leads to relative low data rates, while amount of NACKs at the network is much more than in case of UFTP (due to test setup restrictions, we can't give certain numbers here, but this issue has to be covered by our future works). Processing of each NACK consumes more and more resources with increasing transfer data rates. And practically it makes the protocol very sensitive to any kind any network impairments.

A pretty new finding within this research work is the fact that jitter mostly affects data transmission rate only at high data rates. All test runs confirm that jitter influence at low data rates could be even neglected for both protocols. It could be explained by the fact that at low data rates, timing issues, such as changing inter-arrival gap do not affect receiver significantly. The packet rate is also lower and it enables both protocols to deal with jitter in a very good manner.

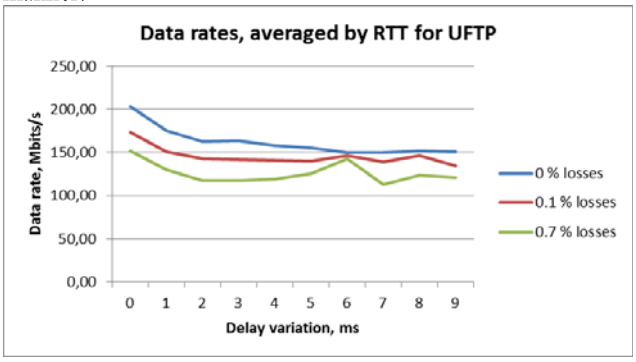

Figure 5. Data rate on delay variation dependency for link with different losses (with averaged value of RTT)

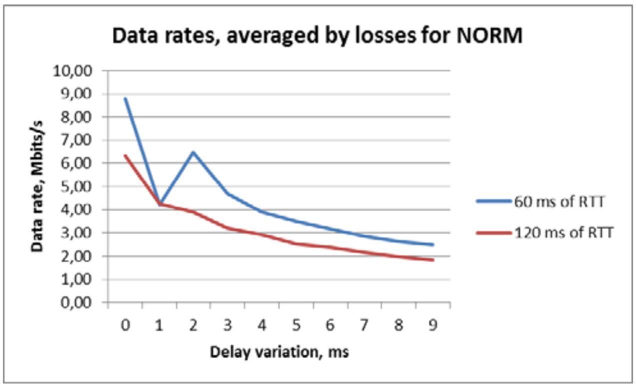

Figure 6. Data rate on delay variation dependency for link with different RTTs (with averaged value of losses)

REFERENCES

[1] C. Demichelis and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics", RFC 3393, 2002.

[2] H. Schulzrinne, S. Casne, and R. Frederick, "RTP: A Transport Protocol for Real-Time Applications," RFC-3550, 2003.

[3] Javed I. Khan and Raid Y. "Jitter and Delay Reduction for Time Sensitive Elastic Traffic for TCP-interactive based World Wide Video Streaming over ABone", Proceedings of the 12th IEEE International Conference on Computer Communications and Networks, ICCCN 2003, Dallas, Texas, October 2003, pp.311-318.

[4] E.Siemens and A. Bakharev, "Evaluation of Reliable Multicast Implementations with Proposed Adaptation for Delivery of Big Data in Wide Area Networks", Accepted for publishing in the proceedings of ICNS 2013, March, Lisbon, 2013.

[5] E.Siemens and D. Kachan, "Comparison of contemporary solutions for high speed data transport on WAN connections". Accepted for publishing in the proceedings of ICNS 2013, Lisbon, 2013.

[6] Apposite Technologies home page. http://www.apposite-tech.com. [retrieved: March 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: March 2013]

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

[10] Internet-draft, StarBurst Multicast File Transfer Protocol (MFTP) Specification. http://tools.ietf.org/html/draft-miller-mftp-spec-02.[retrieved: March 2013]

[11] UFTP - Encrypted UDP based FTP with multicast. http://www.tcnj.edu/~bush/uftp.html. [retrieved: March 2013]

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