Transmitting Hierarchical Aggregation Information Using RTCP Protocol

Radim Burget , Dan Komosny, and Milan Simek

Department of Telecommunications, Faculty of Electrical Engineering and Communication,
UT Brno, 602 00 Brno, Czech Republic

Summary
The current state of the RFC 3550 RTP/RTCP standard is not optimal for large-scale streaming sessions employing source-specific multicast and is currently the subject of research in many research laboratories. The most promising optimization for the RTP/RTCP protocol seems to be hierarchical aggregation [2], [3], [5], [6], [10], [11]. However, hierarchical aggregation has not yet completely solved the problem of hierarchical structure organization. The answer can be a Tree Transmission Protocol (TTP), which seems to be flexible and powerful but quite demanding to implement. This text deals with a different approach that tries to maximally utilize the features of current standards.

Key words:
RTCP, Hierarchical aggregation, Sub-report Block, Receiver Summary Information

1. Introduction

RTP and RTCP [1] are protocols designed for data delivery in real-time and to measure the quality of service (QoS). The RTCP protocol uses receiver reports (RR-RTCP) and sender reports (SR-RTCP), which are necessary to make it functional. According to the latest IETF draft for RTCP extension of Source-Specific Multicast sessions with unicast feedback [2], receiver reports (RR-RTCP) can be aggregated by a feedback target into a so-called Receiver Summary Information packet (RSI-RTCP). These RSI-RTCP packets remove an irrelevant (e.g. IP & UDP headers) part of information and reduce data redundancy (e.g. IP & UDP headers) in reports being transmitted. Furthermore, this extension reduces the total consumed bandwidth of RTCP protocol and thus allows the deployment of a greater number of receivers in a single session. In Fig. 1 the basic schema of the summarized topology is depicted. As presented here, receivers send RR-RTCP packets; the feedback target aggregates these messages into RSI-RTCP packets and transmits them to a data source; the data source transmits media data (e.g. audio & video in the case of IPTV) and reflects the RSI-RTCP packets received from the feedback target into a multicast channel. Although the data source and the feedback target are represented as standalone machines, they are usually deployed on a single machine with a single IP address.

Manuscript received November 5, 2007
Manuscript revised November 20, 2007

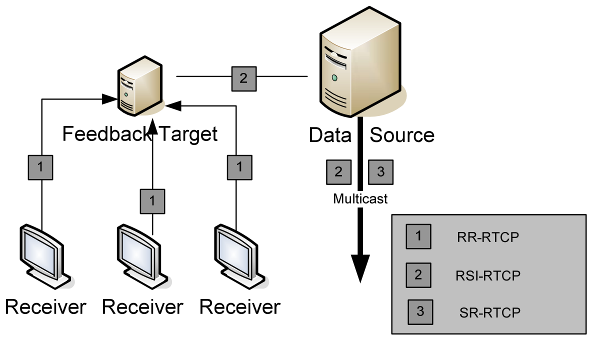

Fig. 1: Schema of summarization of receivers reports into Receiver Summary Information packet

The RSI-RTCP packet is proposed [2] to be very flexible and it can be used for various purposes and types of QoS measurement. Its basic block structure is shown using the notation of UML class diagram [4] in Fig. 2. There the basic block called “RSI packet” can aggregate the basic block fields. These fields can be of several types: generic basic sub-report block, feedback address target sub-report block, collision sub-report block, RTCP bandwidth sub-report block and RTCP group and average packet size sub-report block. In a single RSI packet there can be an unlimited number of these sub-report blocks, starting from zero (no occurrence) to several. However, most common occurrence is just a single occurrence in an RSI-RTCP packet.

In the RTCP protocol the maximal consumed bandwidth is limited to 5% of the total service reserved bandwidth. To meet this limitation the frequency of transmitting RTCP messages must fulfill exactly the given equations. These equations compute the period for transmitting RR-RTCP messages (T_{RR}), SR-RTCP messages (T_{SR}) and RSI-RTCP messages (T_{RSI}). All of them are described by equations (1), (2), (3) and (4):

$$T_{RR} = \frac{PL_{RR} \cdot n}{25\% \cdot BW_{RTCP}} \quad (1)$$

$$T_{SR} = \frac{PL_{SR}}{75\% \cdot BW_{RTCP}} \quad (2)$$

$$T_{RSI} = 1.5 \cdot T_{SR} \tag{3}$$

$$BW_{RTCP} = 5\% \cdot BW \tag{4}$$

PL stands for the packet length of message, *BW* stands for the total bandwidth, BW_{RTCP} for the bandwidth reserved for RTCP protocol, and *n* is the total number of receivers in the whole session. As follows form equation (1), for a great number of receivers *n* the period T_{RR} can become really long and this leads to inaccurate measuring and makes the values measured useless.

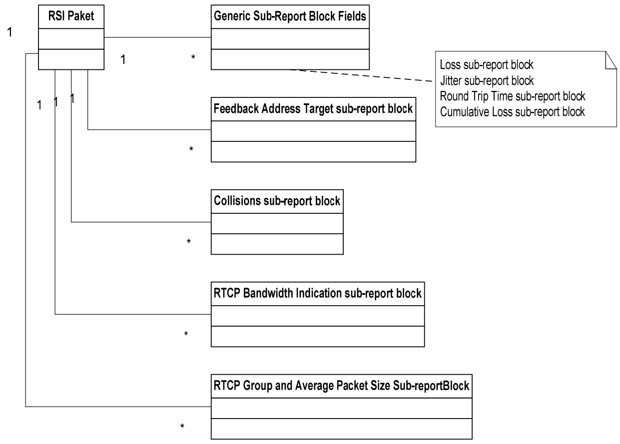

Fig. 2: Diagram of RSI packet structure expressed in UML class diagram.

What should be emphasized on a summarized topology is the fact that only a single feedback target exists in the whole session and therefore RSI-RTCPs describes the session as a whole. This is the main difference in comparison with a hierarchical aggregation that will be presented in the next section.

2. Hierarchical Aggreagation

The hierarchical aggregation [3] (HA) is another improvement of the RTCP protocol that has been recently introduced. Thanks to HA the idea of redundant data flow reduction was been advanced even further. It uses more than a single feedback target and these feedback targets may be organized hierarchically. With their help data redundancy can be removed in a short distance from the receiver and this gives us the ability to construct topologies ready for large-scale deployment where a huge number of receivers can be connected at the same time with a lower bandwidth consumption.

As described in detail in “Tree Structure for Source-Specific Multicast with feedback Aggregation” [3], HA can give up to 100 times better results in comparison to the RFC 3550 RTP/RTCP standard [1].

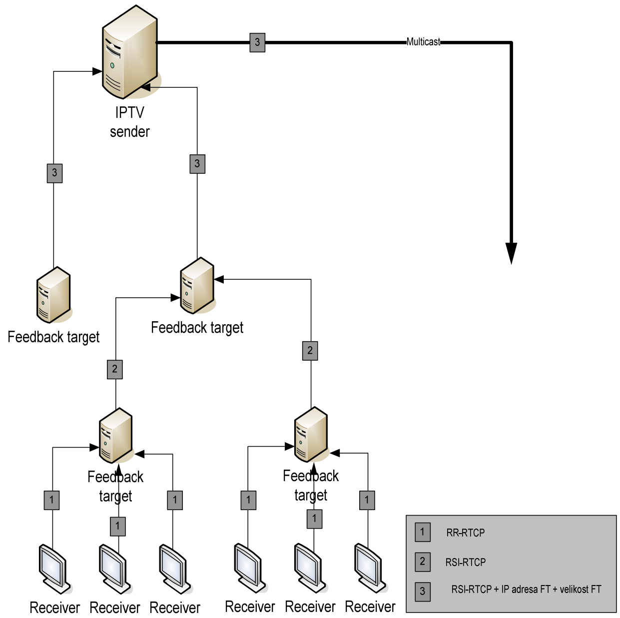

Fig. 3: Hierarchical aggregation scheme with many feedback targets

Unfortunately, new problems with HA have emerged - there must be a way how to organize the hierarchical tree structure and how to inform receivers about the size of each subgroup. In other words, how many members share the BW_{RTCP} bandwidth. This is necessary to know to be able to calculate T_{RR}, T_{RSI} time intervals as shown in equations (5), (6), (7) and (8):

$$T_{SR} = \frac{PL_{SR}}{75\% \cdot BW_{RTCP}} \tag{5}$$

$$T_{RSI\_S} = \frac{PL_{RSI}}{25\% \cdot BW_{RTCP}} \tag{6}$$

$$T_{RSI\_FT} = \frac{PL_{RSI} \cdot n_{FT}}{25\% \cdot BW_{RTCP}} \tag{7}$$

$$T_{RR} = \frac{PL_{RR} \cdot n_{G\_R}}{25\% \cdot BW_{RTCP}} \tag{8}$$

T_{RSI_S} stands for the interval of the RSI-RTCP packet transmission from the sender, T_{RSI_FT} stands for the interval of the RSI-RTCP packet transmission from the feedback targets, n_{FT}, n_{G_R}, give the number of neighbouring feedback targets or receivers that have a common feedback target in a single subgroup (see Fig.3).

3. Extension of Source-Specific Multicast Sessions

An answer to the tasks described in the section above can be the Tree Transmission Protocol (TTP) [5], [6]. This is quite a flexible and robust protocol for organizing the hierarchical tree overlays. Furthermore, it is a stand-alone protocol independent of RFC 3550 RTP/RTCP standard [1] and can be used for any hierarchically organized protocols. It can work for simple hierarchical tree overlays as well as for large-scale overlays with many hierarchical levels.

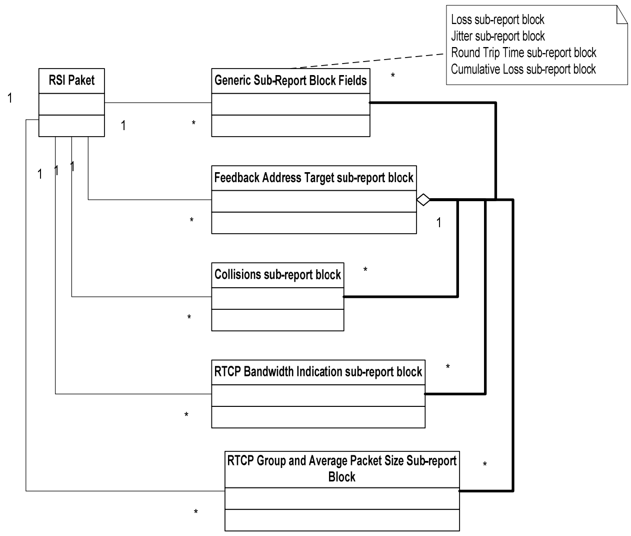

Fig. 4: Relation between group packet subreport block and other subreport blocks. Scheme designed in UML[4] notation.

The fact that the TTP is a stand-alone protocol is an advantage and a disadvantage at the same time. On the one hand it can be optimally designed for a particular purpose and it is not necessary to consider any previous standards. On the other hand, we cannot utilize implementations of other protocols and as we need to start from scratch, much more code will be necessary to implement. In this section another approach will be introduced, which tries to make maximum use of the current RTP/RTCP related standards.

As has been described above, the latest draft for source-specific multicast sessions [2] (see Fig. 2) defines the structure of an RSI packet and its sub-report blocks. By using them, we are able to describe the session as a whole. However, the designers have proposed the protocols taking into account only a single feedback target but in HA we need several feedback targets. For this purpose a new kind of sub-report block is introduced here – so-called feedback group packet sub-report block. This kind of sub-report block can include any other sub-report blocks and possibly even itself (see Fig. 4, Fig. 5) and thus separate an RSI message into logical sections that belongs to particular subgroups in a session. Thus all the receivers can be informed about the size of all subgroups and it is up to each receiver to make a decision where to connect. In general it should be a matter of feedback utilization and the distance from the receiver. Finding the best rate between these two properties is still an open question.

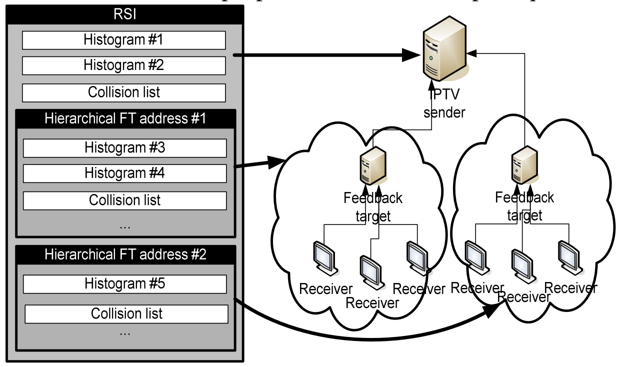

Fig. 5: Relation between group packet subreport block and other subreport blocks. Scheme designed in UML[4] notation.

The structure of packet is depicted in Fig. 5 and Fig. 6. It consists of fields: sub-report block type, length, port number, address and composition of other sub-report blocks.

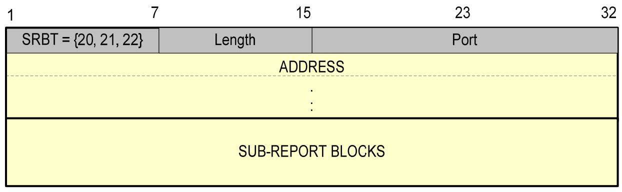

Fig. 6: Feedback group packet sub-report block

Length: 8 bits

Length is the length of a sub-report in 32-bit words. For an IPv4 address it equals 2 (e.g.., total length = 4 + 4 = 2*4 octets) + variable length of included sub-report blocks. For an IPv6 address his value equals 5 + length of included sub-report blocks. For a DNS name the length equals to number of 32-bits words that represents string finished with “\0” value (null value).

Port: 2 octets (optional)

Number of port where FTs or receivers should send its reports. The zero value is invalid and cannot be used.

Address: 4 octets (IPv4), 16 octets (IPv6), or *n* octets(DNS name)

The address is an IP address to which receivers send feedback reports. For IPv4 and IPv6 fixed-length address fields are used. A DNS name is an arbitrary length string that is padded with null bytes to the next 32 bit boundary. The string MUST be UTF-8 encoded [7]. For IPv4, SRBT=20. For IPv6, SRBT=21. For usage of the DNS name, SRBT=22.

4. Comparison with TTP protocol

Both the TTP protocols and here introduced extension gives the possibility to deploy hierarchical aggregation with the RTCP protocol. The main difference between them is a fact the TTP is a standalone protocol that can work with any hierarchically aggregated session (see Fig. 7).

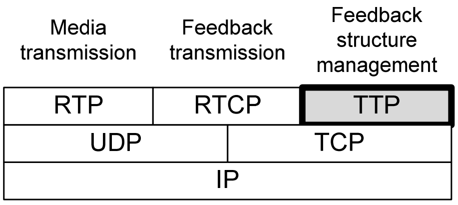

Fig. 7: Position of TTP to related protocols.

On the other hand the TTP protocol cannot utilize the features of the RTCP protocol and therefore the implementation will be more complex.
Using the feedback group packet sub-report block we can fully utilize the algorithms of the RTCP protocol because the sub-report block is a part of the protocol. Therefore we can are automatically limited to 5% of total RTP/RTCP as the RTCP standard describes [1]. The position of the feedback group packet sub-report block is depicted in Fig. 8.

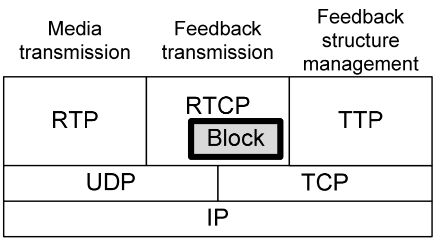

Fig. 8: Position TTP

5. Conclusion

The current RTP/RTCP standard is not prepared for large deployment. The hierarchical aggregation is the method how the RTP/RTCP protocols can be deployed on large-scale session with huge number of receivers, however it still has not solved all the problems. One of them is how the hierarchical tree should be organized and managed. In order to deal with it a new type of message was proposed, so-called feedback group packet sub-report block. It can be a part of RSI packet and gives the packet a possibility to describe its hierarchically divided groups and subgroups (see Fig. 3). This gives the possibility to use current RTP/RTCP standard together with hierarchical aggregation with very little additional coding and with quite high utilization of features of the RTCP protocol.
This protocol has been tested in laboratory network consisting of two routers and 3 stations. This is not enough to reveal all the potential problems that can occur in real network. In further work it is planned to deploy the protocol on some bigger experimental network where the PlanetLab[12] seems to be the suitable solutions for our requirements.

Acknowledgments

This work was supported by the Academy of Sciences of the Czech Republic, project
1ET301710508.

References

[1] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP - A Transport Protocol for Real-time Applications," RFC 3550 (STD 64), July 2003.
[2] J. Ott, J. Chesterfield, E. Schooler, "RTCP Extensions for Single-Source Multicast Sessions with Unicast Feedback", IETF draft, AVT-RTCP-SSM, March 2007.
[3] KOMOSNY D., NOVOTNY V. Tree Structure for Source-Specific Multicast with feedback Aggregation, in ICN07 - The Sixth International Conference on Networking . Martinique, 2007, ISBN 0-7695-2805-8
[4] Object Management Group, <http://www.omg.org/gettingstarted/what_is_uml.htm>
[5] KOMOSNY, D., BURGET, R., Feedback Distribution in Source-Specific Multicast using Tree Transmission Protocol
[6] NOVOTNY, V., KOMOSNY, D. Optimization of Large-Scale RTCP Feedback Reporting in ICWMC 2007. ICWMC 2007 - The Third International Conference on Wireless and Mobile Communications. Guadeloupe, 2007, ISBN: 0-7695-2796-5
[7] F. Yergeau, "UTF-8, a transformation format of ISO 10646", RFC 3629, November 2003.
[8] Control Protocol (RTCP) Leading to Management Facilities in the Internet," iscc, p. 125, Third IEEE Symposium on Computers & Communications, 1998.
[9] J. Chesterfield, E.M. Schooler "An extensible RTCP control framework for large multimedia distributions", 2003
[10] J. Chesterfield, J. Ott, E.M. Schooler "RTCP Extensions for Single-Source Multicast Sessions with Unicast Feedback", Internet draft, IETF draft-ietf-avt-rtcpssm-11.txt, 2006
[11] M. Castro, P. Druschel, A. Kermarrec, A. Rowstron "A large-scale and decentralized application-level multicast infrastructure", IEEE Journal on Selected Areas in Communications, vol. 20, no. 8, pp. 1489-1499, 2002
[12] PlanetLab, An open platform for developing, deploying, and accessing planetary-scale services, <http://www.planet-lab.org/>

Radim Burget Graduated from Brno University of Technology, Faculty of Information Technology. He is engaged in research focused on IPTV. He has experience with development of J2EE applications.

Dan Komosny
He is engaged in research focused on transmission of voice over IP network (VoIP). He also focuses on development of e-learning tools using formal and visual languages – the SDL object oriented design language and the MSC trace language.

Milan Simek
Milan Simek received the M.S. degree in Electrical Engineering from Department of Telecommunications of Brno University of Technology in 2006. He is engaged in research focused on analysis and simulation of multicast operations mainly in Ns2 and SDL environment.