Simulation of Large-Scale IPTV Systems for Fixed and Mobile Networks

Radim Burget[1], Dan Komosny[1], Milan Simek[1]

[1] Department of Telecommunications,
Faculty of Electrical Engineering and Communication, UT Brno,
602 00 Brno, Czech Republic
{burgetrm, komosny}@feec.vutbr.cz, xsimek12@stud.feec.vutbr.cz

Abstract. The Internet Protocol Television has received much attention in recent years. It is a service that delivers media contents to costumers using the Real-time Transport Protocol and RTP Control Protocol. It brings many benefits for consumer and even for media distributors. IPTV is a specific application. It has only one sender and it should be able to offer its content to a huge number of receivers. As it is a new kind of service, there are still some problems to be solved. One of the most important is the Quality of Service (QoS) reporting delay and its dependence on the number of receivers in the session. The conventional RTP/RTCP architectures exceed acceptable range of the reporting interval with a relatively small number of receivers. This paper surveys the newest RTP/RTCP topologies and proposes a simulation model for further optimizations of algorithms and mathematical models for IPTV systems.

Keywords: RTP, RTCP, Real-time Transport Protocol, RTP Control protocol, Feedback, hierarchical aggregation, IPTV, Internet Protocol Television, Receiver Summary Packets, Sender reports, Receiver reports, Petri net..

1 Introduction

IPTV (Internet Protocol Television) is a service that delivers media contents (most commonly audio & video) to receivers. Each receiver (R) receives RTP data with audio and video contents and sends feedback reports, so called Receiver-Reports, in an accurate calculated interval that contains information about the Quality of Service (QoS). The calculation of this interval is described in detail in RFC 3550 [1]. Each member in this session communicates using the Any-Source Multicast in the many-to-many fashion (see Fig. 1). It is therefore really simple to inform other the members in
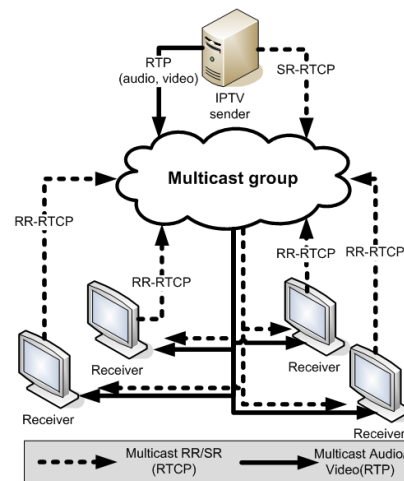
Fig. 1: IPTV session using Any-Source Multicast

the session about the member's state. A disadvantage of the many-to-many communication is that it produces a great routing complexity, especially for huge number of receivers. Therefore a new routing protocol, the Source-Specific Multicast (SSM), has been introduced.

2 Source-Specific Multicast

A major advantage of the Source-Specific Multicast is that routing is much simpler than the older Any-Source Multicast. But it also has an unpleasant restriction. In the Source-Specific Multicast it is possible to communicate only in one-to-many fashion. Therefore the one and the only member in a whole session which is capable of sending data via multicast channel is the sender. Therefore it is not possible for any receiver to send its Receiver-Reports (RR) via the multicast channel. To bypass this restriction so called Reflection method is used. Each receiver sends its receiver reports to the sender via a unicast channel and subsequently the sender retransmits the received Receiver-Reports (RR) into the multicast channel(see Fig. 2).

As mentioned above, the interval for transmitting Receiver-Reports is computed according to equations described in RFC 3550 [1]. These folowing session parameters has an impact on the interval length: average length of packets (SR, RR), bandwidth, number of senders in the session (in case of Source-Specific multicast it equals to 1) and finally the number of members in the session.

The length of packets is limited by the protocol used, the bandwidth and the number of senders are not dynamically changing during the session time. Hence, these two could rarely be the origin of an unacceptable length of the reporting interval.

Nevertheless, if a number of members in the session is too big, the length of computed interval could be unacceptably large (minutes or even hours). If the interval exceeds some particular value, the reports might not be relevant to the current session state and the sender could form an inaccurate view. In this case, the optimization that sender could make according to its view could even have a negative effect on the session performance.
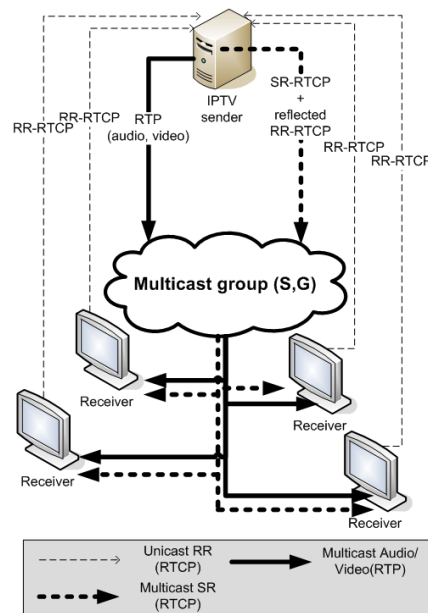
Fig. 2: IPTV session using Single-Source Multicast and reflecting method

3 Hierarchical Aggregation

There are many ways how to reduce the interval delay. Detailed information can be found in "Real-time control protocol and its improvements for Internet Protocol Television" [2]. One of the most advanced and most promising improvements for the future use is the so-called Hierarchical aggregation (HA). This method is based on an idea where a new member type is inserted between the sender and the receivers, the so-called feedback target. Each receiver sends its reports to one of the feedback targets via a unicast channel. The feedback targets gather these reports from their related receivers and they create summaries of parameter receiver reports for each quality of service (QoS) parameter being measured. These histograms are retransmitted to the sender (using the so-called RSI packets) and it is up to the sender to decide how to optimize the audio and video contents to maximize the Quality of Service (see Fig. 3).
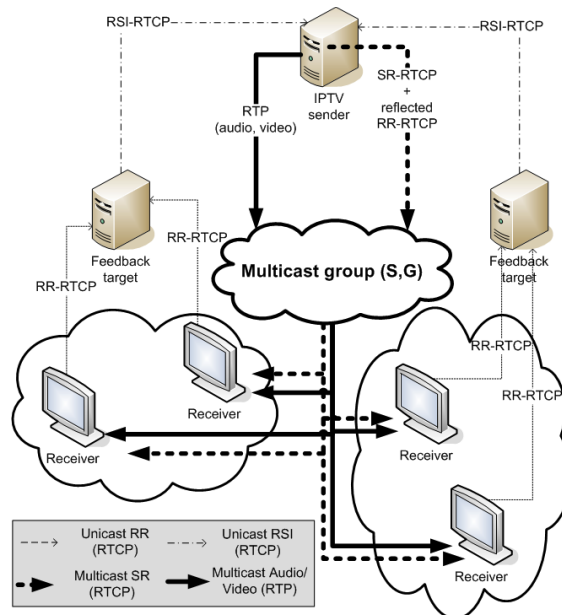
Fig. 3: IPTV session wit Single-Source Multicast using hierarchical aggregation

4 Member types in Hierarchical aggregation

In the architecture of hierarchical aggregation there are 3 types of members in the session: **sender**, **receiver** and **feedback target**. Each of them has a different functionality. The sender transmits audio and video data via the multicast channel, sends Sender-Reports and reflects Receiver-Reports received from receivers to the multicast group. The sender is the only member in a session which can send data into the multicast channel whereas receivers can no do that. They can only receive data

from the multicast channel and therefore they have to bypass transmitting the Receiver-Reports messages using the unicast channel.

5 Data flow in Hierarchical aggregation

When we were dealing with how to verify experimentally the hierarchical aggregation model, we were asking several questions:

- How to simulate the behaviour of a network with a huge number of users (2 millions)? It would be really expensive to build an experimental network with such a huge number of nodes, routers and network devices.

- How to find out the real network conditions. The sender's view of the conditions a network according to the Receiver-Reports is almost every time affected by some diversion or a time-shift. The diversion is due to the trade-off between the effort to reduce the length of these RR packets and its precision. To monitor the the real network conditions would need an extra protocol that would communicate faster than RTCP does. Unfortunately, even this could change the network conditions and would not be accurate.

- How to visualize the behaviour of session members?

The answer to all these questions could consist in a simulation where we can relatively cost-effective by fulfil all these requirements.

The first thing we need to know is the behaviour of each member type. Each of them has a different functionality and each has different data available.

5.1 Sender data flow

The sender is the only member that can send data into the multicast channel. It sends RTP packets, Sender-Report RTCP packets (SR-RTCP) and Receiver Summary Information packets (RSI-RTCP). RTP packets are **Real-time Transport Protocol** packets that carry media contents, most commonly audio or video data. They are described in detail in RFC 3550 [1]. The **Sender Report** packets (SR-RTCP) are Real-time control protocol packets that tell each member how many packets have been sent. Each receiver makes use of it to evaluate the quality of reception and subsequently to create **Receiver-Reports** (RR-RTCP). The last one are **Receiver Summary Information** packets (RSI-RTCP). Sender creates them on the basis of information received from feedback targets and Receiver-reports received from receivers (see Fig. 4).
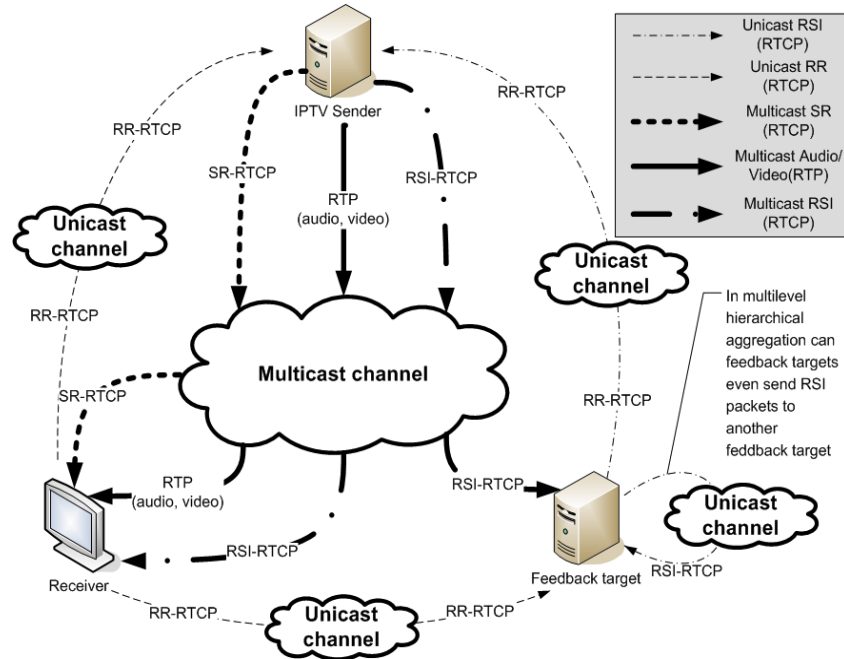
Fig. 4: Data flow between member types in Source-Specific Multicast and hierarchically aggregated session

5.1 Receiver data flow

Receivers receive RTP packets, RSI-RTCP packets and SR-RTCP packets. RSI-RTCP packets stand for Receiver Summary Information and they contain summarised statistics about network status, in particular about the sender or feedback target view of the network status of its related subgroup. And finally SR-RTCP packets that stand for Sender-report packets. They are transmitted by the sender and contain some important parameters that are used by receivers to evaluate receiver's quality of reception (see Fig. 4).

5.1 Feedback target data flow

As shown in Fig. 4, each feedback target receives **Receiver-Report** packets (RR-RTCP) from its related group of receivers. It gathers the received packets and in pre-calculated interval, transmits **summary information packets** (RSI-RTCP) to the sender or to other feedback targets if the level of hierarchy is greater than one. Each receiver chooses a feedback target by itself.

6 Abstract model of hierarchical aggregation

It is possible to create an experimental network with a few items in the network and verify the theoretical background on it. However an IPTV session usually has the size of millions of users in a single session. Building an experimental network of that extent would be really expensive if not impossible. This is where simulation comes to play.

6.1 Abstract sender model

The sender in an IPTV session can send via the multicast channel and receive via a unicast channel. It sends Sender-Reports (SR-RTCP), Receiver Summarization (RSI-RTCP) packets and RTP packets with audio and video contents. And it receives Receiver Reports (RR-RTCP) from receivers and Receiver Summarization (RSI-RTCP) packets from feedback targets.

The corresponding model is designed using the Petri-net model. Firing the transition will start asynchronous processes that send sender report packets, receiver summary packets, and audio and video data (RTP packets). On the other hand, it receives Receiver-Reports from the receiver and Receiver-Summary packets from feedback targets (see Fig. 4 and Fig. 5).
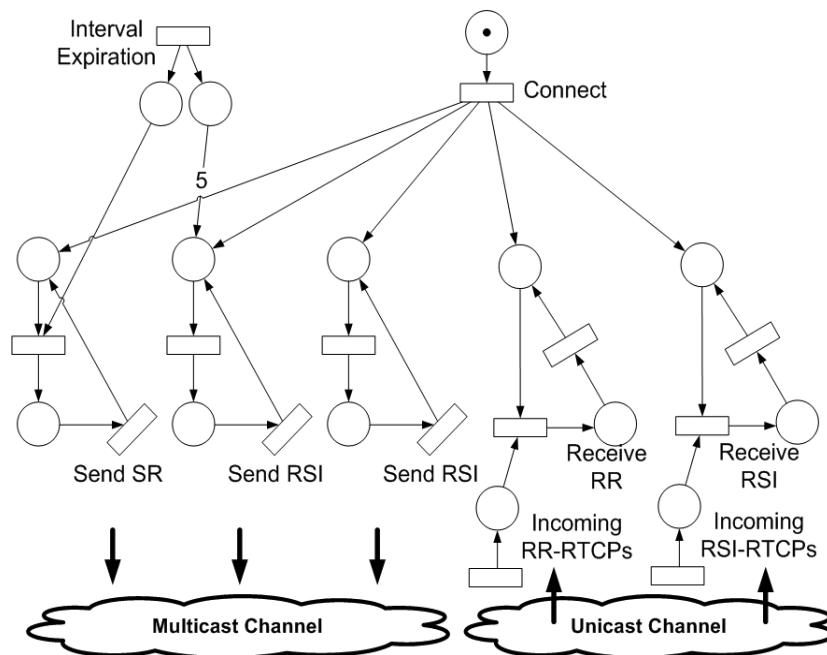

Fig 5: Petri-net model of sender behavior

6.2 Abstract receiver model

The receiver can send Receiver-Report (RR-RTCP) packets via a unicast channel and receive RTP data and Receiver Summarization (RSI-RTCP) packets via the multicast channel. The corresponding model is depicted in Fig. 5.
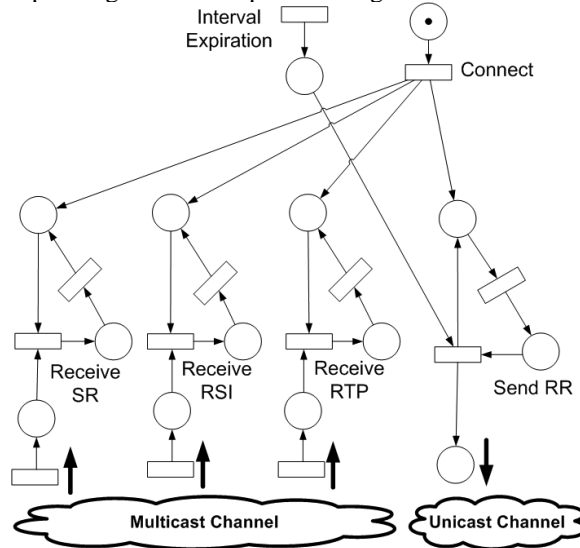

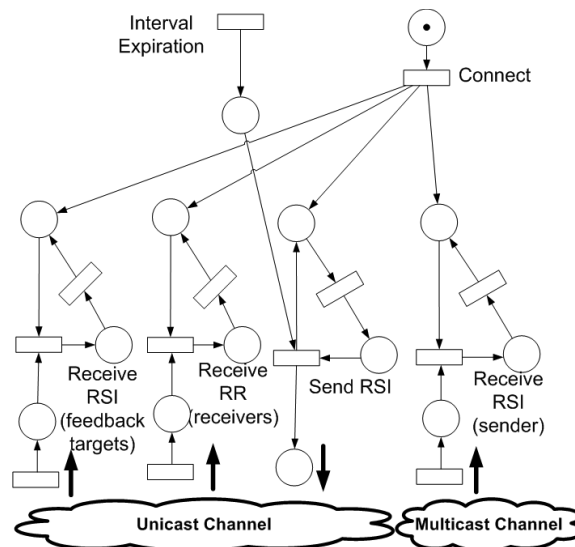**Fig 6**: Petri-net model of receiver behavior.

Fig 7: Petri-net model of feedback target behavior

6.3   Abstract feedback target model

The feedback target can send packets using a unicast channel and receive from both the unicast and the multicast channel. It sends Receiver Summarization (RSI-RTCP) packets to the other feedback targets. It receives Receiver Reports (RR-RTCP) from a related group of receivers, and Receiver Summarization (RSI-RTCP) packets from the other feedback targets from a unicast channel. And finally, it receives Receiver Summarization (RSI-RTCP) packets from sender. Thus it knows the number of members in a session, which is necessary to calculate the reporting interval T_d.

The corresponding model is depicted in Fig. 7.

7   Application for IPTV simulation

The simulation model allows obtaining results for large numbers of receivers. However, it lacks the real-network conditions. For that purpose, the IPTV client/server application has been developed. The main application goal is to broadcast IPTV via a real network and to evaluate the feedback interval sent from receivers to the source. For that purpose, two multimedia sessions were implemented for audio and video transmissions. The application creates a histogram and monitors the actual number of receivers. The IPTV server and client use separate streams with associated ports as depicted in Figure 8.
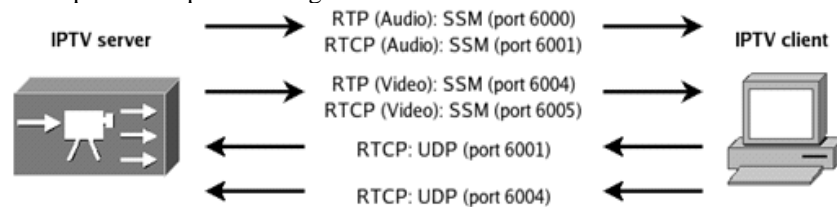

Figure 8: IPTV server media streams

The IPTV server works with one multicast socket and one unicast socket. The multicast socket is used for transmitting the media stream encapsulated in RTP packets and also for sending the SR-RTCP and RSI-RTCP packets to all receivers. The unicast socket is used for receiving RR-RTCP and RSI-RTCP packets which come from receivers and summarization nodes respectively. The receiver also works with two ports associated with RTP and RTCP packets. Besides the receivers and the sender, a new type of member was introduced – the feedback target. The feedback target can behave as a summarization node for a specific group of receivers. The implementation overview of the IPTV receiver is shown in Figure 9 using UML notation. The application has also a graphical interface to show detailed information about simulation outcomes, see Figure 11. It can show every transmitted and received packet. Furthermore it displays a chart with information about the reported number of

receivers and senders in a session and the value of an RR-RTCP packet transmission interval.
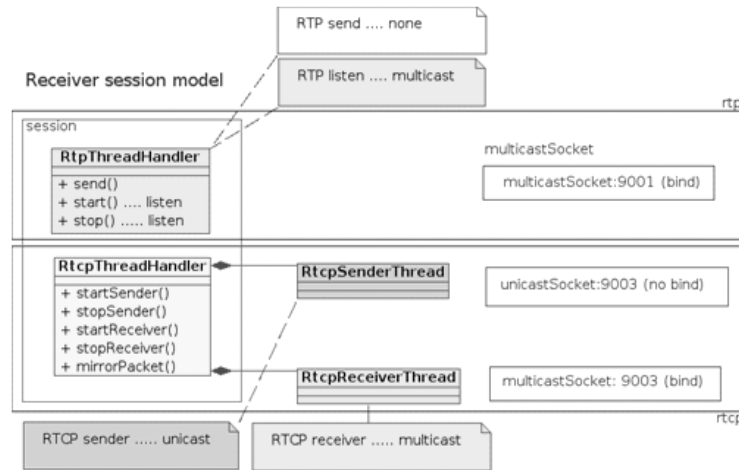
Fig 9: IPTV client structure

The large number of IPTV receivers is allowed due to multiple instances of IPTV application. A selected number of virtual IPTV clients behave as summarization servers for specified groups. The software has been developed in order to run multiple instances with minimal consumption of PC resources (memory, CPU). For example, a group of IPTV receivers can be simulated by one virtual IPTV client with the respective bandwidth consumption and generating multiple RR packets according to equation). Simulation results are gathered at the IPTV server and stored in the text files for further processing using additional software.
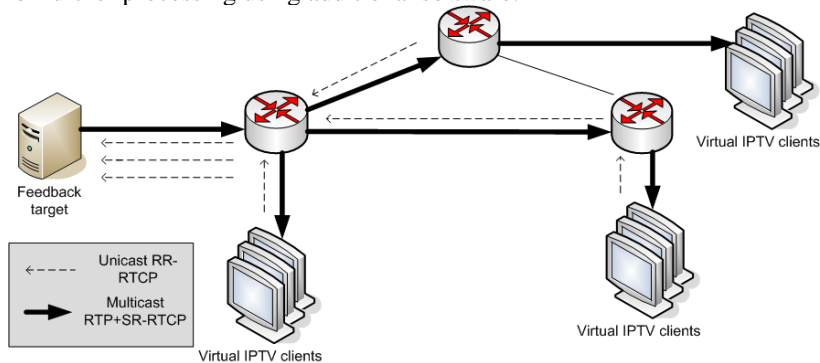
Figure 10: Experimental network overview

The application is used in an experimental network, whose structure is shown in Figure 9. The routers used create domains for a set of session members. The routers

support the necessary routing protocols to implement an SSM session - PIM-SSM (Protocol Independent Multicast) [An Overview of Source-Specific Multicast (SSM)] and IGMPv3 (Internet Group Management Protocol) [Using Internet Group Management Protocol Version 3 (IGMPv3)]. However, in order to provide real conditions, the session members could be assigned to hierarchical tree groups with no relation to the router position. Also, the IPTV client/server application could cooperate with commercial IPTV solutions. This provides for future analysis of IPTV broadcasting from the experimental network. The IPTV solution used in the experimental network covers three components Cisco IP/TV 3442 Broadcast Server, Cisco Content Engine CE-566A (including IPTV Program Manager) and IPTV Viewer.
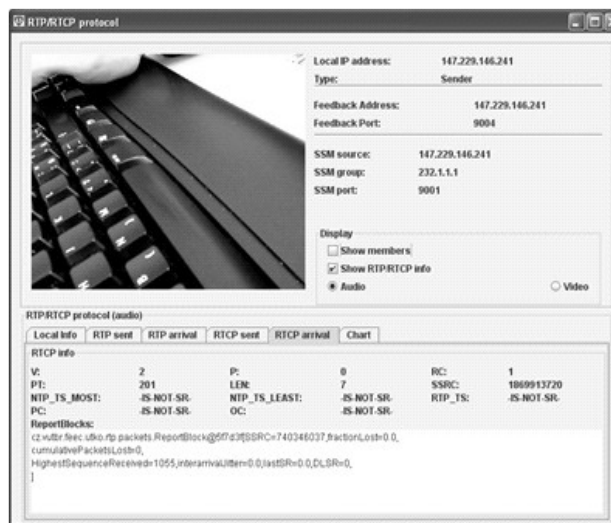

Figure 11: IPTV client graphical interface

Conclusion

The Internet is currently used for the distribution of classical broadcasting services such as TV (IPTV). The reason could be seen in the great number of possible subscribers and more features enabling the control of media transmission. Our research deals with the feedback transmission within IPTV sessions. The feedback is usually used by high-layer protocols to control and monitor the session behaviour. The research is mainly focused on the hierarchical feedback aggregation. The algorithm significantly reduces the feedback transmission interval sent in the receiver-to-source communication.

The paper gives an overview of the feedback interval transmission problem and proposes a simulation model of feedback transmission in large-scale IPTV sessions. The simulation model was proposed using Petri-net model and was implemented using a JAVA discrete event simulation library. However, this kind of simulation faces problems arising from neglecting some aspects of network properties. Therefore the IPTV client/server was developed and used in an experimental network. The required number of IPTV receivers is created using virtual clients. Special logging files are

created to store the simulation results, mainly at the IPTV server side. Also the application is able to cooperate with the IPTV Cisco Broadcast Server. This feature allows future deployment of the IPTV client/server application in a large real networks for further simulations and testing of new feedback transmission algorithms.

References

1. H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: Transport Protocol for Real-time Applications", RFC 3550, July 2003, ftp://ftp.rfc-editor.org/in-notes/rfc3550.txt
2. R. Burget, D. Komosny, "Improvements for RTCP Feddback for IPTV", GESTS, July 2006.
3. Chesterfield, J., E. Schooler, J. Ott, "RTCP Extensions for Single-Source Multicast Sessions with Unicast Feedback," Work in progress, October 2004.
4. SCHULZRINNE, H., CASNER, S., FREDERICK, R.. RTP Profile for Audio and Video Conferences with Minimal Control, Request for Comments 3551, Internet Engineering Task Force, 2003.
5. ROSENBERG, J, SCHULZRINNE, H. Timer reconsideration for enhanced RTP scalability, Proceedings of Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE, 1998.
6. BHATTACHARYYA, S.. An Overview of Source-Specific Multicast (SSM), Request for Comments 3569, Internet Engineering Task Force, 2003.
7. Chesterfield, J.; Schooler, E.M.. An extensible RTCP control framework for large multimedia distributions. Network Computing and Applications, 2003. NCA 2003.
8. KOMOSNY, D., NOVOTNY, V. Analysis of bandwidth redistribution algorithm for single source multicast In Proceedings of the Sixth International Network Conference. Sixth International Network Conference. United Kingdom: University of Plymouth, 2006, s. 45 - 52, ISBN 1-84102-157-1
9. 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
10. N. Baldo, U. Horn, M. Kampmann, F. Hartung "RTCP feedback based transmission rate control for 3G wireless multimedia streaming", 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2004 - PIMRC 2004, Volume: 3, page(s): 1817- 1821, ISBN: 0-7803-8523-3, 2004
11. Randa El-Marakby, David Hutchison "A Scalability Scheme for the Realtime Control Protocol". Proceedings of the IFIP TC-6 Eigth International Conference on High Performance Networking, Pages: 153 - 168, Kluwer, B.V, ISBN:0-412-84660-8, 1998
12. 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
13. R. El-Marakby, D. Hutchison, "Scalability Improvement of the Real-Time Control Protocol (RTCP) Leading to Management Facilities in the Internet," iscc, p. 125, Third IEEE Symposium on Computers & Communications, 1998.