Multicast Feedback Control Protocol for Hierarchical Aggregation in Fixed and Mobile Networks

Dan Komosny, Radim Burget

Dept. of Telecommunications, Brno University of Technology
Brno, Czech Republic
{komosny, burgetrm}@feec.vutbr.cz

Abstract. For large-scale multimedia distributions, multicast is the preferred method of communication. ASM (Any Source Multicast) and SSM (Source-Specific Multicast) are the two types of multicast used. ASM is designed for either many-to-many or one-to-many communication. SSM is derived from ASM. SSM is used when only one session member is allowed to send data. An example use of SSM could be an IPTV broadcasting system over fixed or mobile network. The paper deals with describing hierarchical aggregation for feedback transmission in SSM. For the purpose of hierarchical aggregation, multicast receivers are organized into a tree structure. We present a tree structure consisting of end and summarization nodes. End nodes act as multicast receivers and summarization nodes perform feedback aggregation. The proposed MFCP (Multicast Feedback Control Protocol) is used to establish the tree structure and to exchange signalization needed for the feedback hierarchical aggregation.

Keywords: multicast, feedback, hierarchical aggregation, tree, feedback target, IPTV

1 Introduction

ASM (Any Source Multicast) and SSM (Source-Specific Multicast) are the two types of multicast used in IP-based networks [1] [2]. SSM is expected to cover all types of multimedia sessions with many receivers and only one source, such as IPTV broadcasting. The feedback could be transmitted with either ASM or SSM. The feedback which comes from receivers is used by the media source, for example, for the parameterization of a multicast forward error correction (FEC) algorithm or the tuning of audio suppression algorithms. The feedback transmitted usually contains information about the synchronization of transmitted media (audio, video), packet loss, packet delays, and jitter. RTP (Real-time Transport Protocol) and the accompanying RTCP (Real-time Control Protocol) [3] [4] are typically used for multimedia real-time transmissions. RTP is designed to transmit the multimedia (video/audio) whereas RTPC is used for the feedback transmission. Two main types

of packets are used within RTCP - SR (Sender Report) transmitted from the source to receivers, and RR (Receiver Report) transmitted from receivers to the source. For the purpose of communication quality monitoring, the RR and SR packets can be also distributed to end nodes not actually involved in the multimedia reception, i.e. to a dedicated monitoring application. H.323 and SIP (Session Initiation Protocol) [5] are some of the architectures working on the RTP/RTCP protocol stack.

An SSM session is described by the multicast group address and the source unicast address. SSM is much simpler than ASM as regards the protocol complexity. Unlike ASM, there is no need to deploy complex routing trees for bidirectional communication among all participants. Therefore, SSM is more suitable for large-scale conferences than ASM. However, SSM lacks the support for communication among session members (i.e. many-to-many). Therefore, RR packets cannot be transmitted directly via multicast. The existing solutions employ unicast connections from receivers to the source and a summarization method [6] [7] is used to distribute the feedback data back from the source to the receivers via multicast. The method is based on aggregating the received data from RR packets in the source. When the aggregation is finished, a summary packet called RSI (Receiver Summary Information) is assembled and sent to all receivers. In addition, the aggregated values can be compressed up to a factor of 16. The compression significance grows when there are large sessions. The RSI packet is sent from the source together with the sender SR packets.

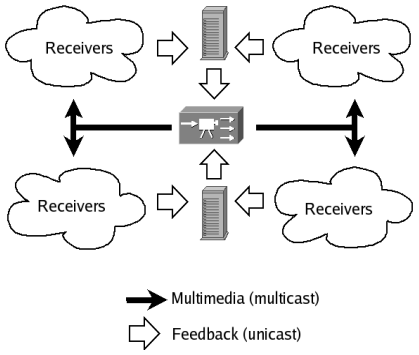

Fig. 1. Feedback hierarchical aggregation

2 Tree Structure for Hierarchical Aggregation

With the hierarchical aggregation, some nodes behave as summarization servers for a group of receivers, see Fig. 1. Members from a group report their feedback to the summarization node and the node puts the feedback received into a single summary. This summary is sent using RSI packets. Summarization nodes are organized

hierarchically, thus a node produces a summary for another node of higher level, all the way up to the source. It is supposed that a summarization node is a dedicated server not involved in the media reception.

For the feedback aggregation, we assume the structure of a tree consisting of both end and summarization nodes. The summarization node highest in the tree hierarchy is the multicast source. On other levels, summarization nodes are presented except the lowest level, which consists of end nodes only. A set of end nodes represent multicast receivers of the media being transmitted via one-to-many multicast. The tree structure is depicted in Fig. 2. The feedback is transmitted from end nodes using RR packets to the summarization node higher in the tree branch. The summarization node aggregates the received feedback values into one RSI packet as described above. The RSI packet is then sent to the next higher summarization node. Note that a higher summarization node aggregates values only from the received RSI packets. Finally, the highest summarization node in the tree receives the summary feedback from all end nodes in a session. Then the highest summarization node (also the multicast source) sends the summary feedback back to all session members via one-to-many multicast. In addition the multicast source sends its own feedback to all session members in the SR packets.

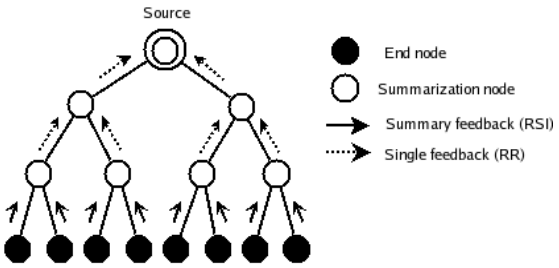

Fig. 2. Tree structure for feedback transmission using hierarchical aggregation

The tree structure should be formed in a way that keeps the round-trip delay in feedback transmission RI_{SS} as low as possible. The session feedback reporting interval could be expressed as

$$RI_{SS} = RI_{RR} + RI_{RSI} \times I, \quad (1)$$

where I is the number of tree levels, RI_{RR} is the reporting interval of RR packets and RI_{RSI} is the reporting interval of RSI packets provided that RI_{RSI} is of a constant value trough tree levels i=0,1,…,I-1. In order to keep the reporting interval as low as possible the reporting interval values should be substituted with RI_{min}, which is the lowest possible reporting interval (5 seconds). The reporting interval for RR packets is identified as

$$RI_{RR} = \frac{PS_{RR} \times n_{gend}}{0.75 \times BW_{RTCP}}, \quad (2)$$

and for RSI packets as

$$RI_{RSI} = RI_{RSI}(i) = \frac{PS_{RSI}(i) \times n_{gsmr}}{0.75 \times BW_{RTCP}}, \quad (3)$$

where $i \geq 0$ is the tree level, PS_{RR} is the size of the RR packet, n_{gend} is the number of end nodes below a summarization node, BW_{RTCP} is the session bandwidth used RTCP packet transmission (5% of the total allowed session bandwidth BW_{SS}), PS_{RSI} is the size of the RSI packet and n_{gsmr} is the number of summarization nodes in a group on a tree level i. In order to assure that the statement $PS_{RSI}=PS_{RSI}(i)$ from equation (3) holds, we need to keep the size of the RSI packet at the same value through all tree levels where aggregation is done, i.e. for $I \geq i \geq 0$. The RSI packet size can be calculated as

$$PS_{RSI} = RSI_{fix} + \sum_{k=1}^{K} (RBL_{fix} + RBL_{data}(k)), \quad (4)$$

where RSI_{fix} is the fixed part size of RSI packet, K is the number of report blocks, RBL_{fix} is the fixed part size of report blocks, and $RBL_{\mathrm{data}}(k)$ is the variable part size of report block k. For more information about the use of report blocks see [7]. Classic headers for IP, UDP, RSI can be used to calculate the fixed part size of the RSI packet. Then, we can identify the variable part size of the report block RBL_{data} as

$$RBL_{data} = DBN \times DBL, \quad (5)$$

where DBN is the number of distribution buckets (also see [7]), and DBL is the size of the distribution bucket. For the purpose of identifying the bucket size, we need to consider the worst case i.e. all end nodes reports feedback values which belong in one bucket. In other words, we need to have enough bits to express the number of all end nodes in a session. Therefore, the worst-case bucket size DBL is

$$DBL(i) = \log_2(n_{gend} \times n_{gsmr}^{(I-i-1)}), \quad (6)$$

where $DBL(i)$ is the bucket size on tree level i. It can be seen from the equation above that on the higher tree levels, the distribution bucket size is growing (more end nodes are involved in aggregation). However, we need to keep the RSI packet size of a constant value as we proposed above. To assure this, we use the multiplicative factor MF defined in [8]. Utilizing the following equation

$$DBL = DBL(i) = \log_2\left(\frac{n_{gend} \times n_{gsmr}^{(I-i-1)}}{D(i)}\right) \quad (7)$$

we are able to calculate the divisor D for the specific tree level i as

$$D(i) = n_{gsmr}^{(I-i-1)} \tag{8}$$

and with the definition $D = 2^{MF}$ in [8], we are able to identify the multiplicative factor *MF* for each tree level as

$$MF(i) = \log_2 n_{gsmr}^{(I-i-1)}. \tag{9}$$

Now, the number of end nodes in a group n_{gend} and the number of summarization nodes in a group n_{gsmr} can be identified for the purpose of a tree establishment. Utilizing equation (2), we can express n_{gend} as

$$n_{gend} = \frac{RI_{\min} \times 0.75 \times BW_{RTCP}}{PS_{RR}}, \tag{10}$$

and similarly, utilizing equation (3), we can identify the n_{gsmr} as

$$n_{gsmr} = \frac{RI_{\min} \times 0.75 \times BW_{RTCP}}{PS_{RSI}}, \tag{11}$$

where RI_{min} is the lowest possible reporting interval (5 seconds).

3 Multicast Receivers Clustering into Tree Groups

Using a defined number of group members n_{gend} and n_{gsmr} from equations (10) (11) only to set the tree structure is not adequate in terms of data transmission in IP networks. In order to achieve a proper routing performance, session members should also be organized in groups considering the relative distance between them and the group summarization node. This relative distance should be kept as small as possible. In large-scale multimedia distributions, the uncontrolled members partitioning into groups could lead to an inefficient IP-level routing. For example, for European IPTV broadcasting, it could happen that a member located in Russia reports its feedback to a summarization node located in Spain, whereas the next higher summarization node is situated in Italy, see Fig. 3. This leads to a difference in the feedback-level (or application-level) overlaying topology and the IP-level underlying topology. For more information about the problem see paper [9]. In a case involving European-scale or even word-scale IPTV broadcasting with feedback transmission, this could seriously degrade the overall network performance.

The intuitive solution of the feedback-level routing problem is to integrate the exact IP-level routing topology into the tree establishment algorithm. This solution is however quite tricky since a close cooperation with all involved routing protocols is required. Instead, we think of another known solution that uses the network latency to find the appropriate closest nodes see [10]. The algorithm called "binning" partitions nodes into bins and nodes within a bin are thought to be relatively close. Results presented in [10] show, that only approximate tree structure information offered by the binning algorithm allows significant routing complexity improvements. The

algorithm is based on a set of landmark nodes (LM) placed on the Internet in a certain way. Then, a node evaluates its round-trip time (RTT) to these landmarks in a specified period of time. On the basis of the measured RTTs, nodes join bins as follows: Each node creates a vector consisting of landmarks ordered by increasing RTTs. The resulting landmark vector then defines a bin, and nodes with the same vector belong to the same bin. Furthermore, nodes can be partitioned into bins using the vector similarity. An important feature of this algorithm is that nodes assign themselves into bins without any communication with other nodes. This is ideal for large-scale multimedia sessions where communication among all nodes would be harmful to the network bandwidth load.

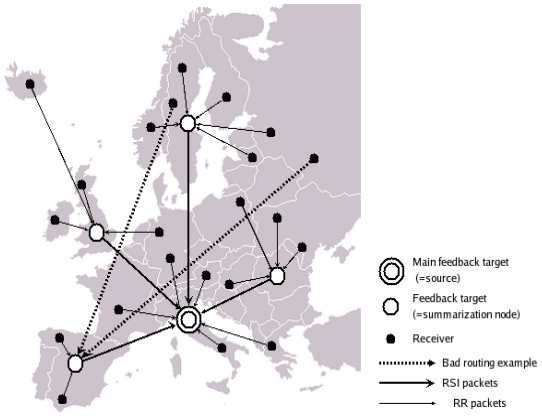

Fig. 3. Feedback transmission example in an European-scale IPTV broadcasting session

4 Multicast Feedback Control Protocol for Hierarchical Aggregation

To organize a tree structure for feedback hierarchical aggregation, every session member should be assigned to a previously defined group, which is represented by the summarization node (also feedback target). In other words, the source communicates with all session members to tell them to which summarization node they should send their feedback to. Since only a one-way multicast channel is provided from the source to session members, this information cannot be addressed to a particular member. Establishing new unicast connections from the source to every session member would greatly increase the network traffic. Therefore, we prefer solution based on a messages sent by the source to all session members, using the existing multicast channel.

For that purpose, we have proposed the multicast feedback control protocol for hierarchical aggregation (MFCP). Fig. 4 shows the protocol position in the protocol hierarchy. Note, that the protocol exploits both unicast and multicast (SSM) communication. The protocol should be considered as an enhancement to the standard protocol RTP/RTCP protocol stack.

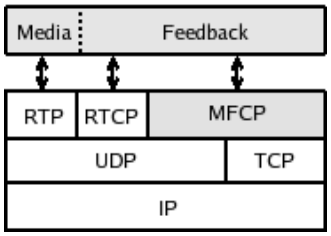

Fig. 4. Multicast feedback control protocol

The key idea used in the protocol is as follows: Let us suppose that the sender is aware of landmark RTT vectors of session members, including both summarization nodes and end nodes. The vector is formed as a list of landmarks starting with the landmark with the lowest RTT value measured. To assure that the source knows all vectors, each session member sends its vector to the sender periodically. We will discuss below how members perform this and how they obtain the landmark IP addresses for the purpose of RTT measurement. As the source is informed about all landmark RTT vectors, it is able to establish the tree structure in terms of relative distance between summarization nodes and group members. The next thing is to involve the required number of group members n_{gend} and n_{gsmr} into the tree structure. The source could meet this requirement by selecting suitable summarization nodes to form an available set of dedicated servers. The servers' availability should be managed by the service provider. After summarization nodes selection, the source calculates the number of group members belonging to a summarization node and compares it with the required values n_{gend} and n_{gsmr} from equation (10) and (11) respectively. Also, the ratio between the significance of relative distance and the number of group members could also be set at the source. In this way, the required tree structure is found. If the source cannot achieve this by selecting summarization nodes from the available set, it could also use fake landmark vectors of summarization nodes. Finally, the source sends the resulting tree structure using a MFCP message to all multicast members in order to put the tree structure in place.

For the purpose of data transmission via the multicast channel, we have proposed a general message shown in Fig. 5. The message consists of the following fields: version of the protocol (4bits), padding bit to signalize the use of padding bytes at the end of the packet (1bit), reserved bits for the future use (11bits), packet type (8bits), length in 32-bit words including the header (8bits) and SSRC value (32bits) carrying the synchronization source identifier of the originator of the packet within a RTP/RTCP session. Using the SSRC value, we are able to create different tree

structures for each synchronization source. This could be useful, for example, if we are interested only in feedback on particular stream within a session.

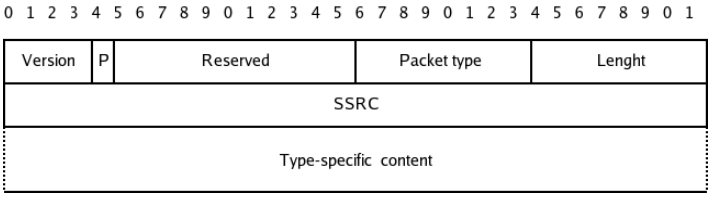

Fig. 5. General multicast feedback control packet (GMFC)

4.1 Transmission from Source to Receivers

The simplest way to transmit information about the required tree structure to the receivers is to format the data as a session member IP address and also its related summarization node IP addresses, which a session member should send feedback to. The IP address of a session member is needed since we use a multicast channel and we are not able to address a packet carrying this information to a particular member. However, provided that the source knows the landmark RTT vectors of all multicast members, instead of sending data containing two IP addresses, it can contain only IP addresses of the selected summarization nodes and their calculated (or fake) landmark RTT vectors. When a multicast member receives this data, it compares its own measured landmark RTT vector with the list of vectors provided and finds the closest summarization node. Then a session member joins the group by sending its feedback to this summarization node. In order to avoid the transmission of landmarks IP addresses in a vector, which would produce a great amount of data, these IP address could be replaced by a landmark ID number. The relation between ID and IP addressed of landmarks are sent to receivers in the landmark packet (LM). The packet also provides a list of landmarks IP addresses for receivers to evaluate the RTT vector. The landmark ID is set according to its position in the list, so the first landmark has a ID=0, second ID=1 and so on. The LM packet (Fig. 6) consists of the following fields: general message header with payload type=1, sequence number (32bit) which increments by one for each LM packet sent and the list of landmarks IP addresses. The purpose of the sequence number is to identify the current ID list for receivers. When a receiver receives the feedback target packet (FT) carrying information about the tree structure, it also includes the sequence number of the LM packet with the information for landmark IDs transformation to corresponding IP addresses. If the sequence number received from LM packet is different from the number received in FT packet, an error occurred during the communication and the receiver should start to send its feedback directly to the source.

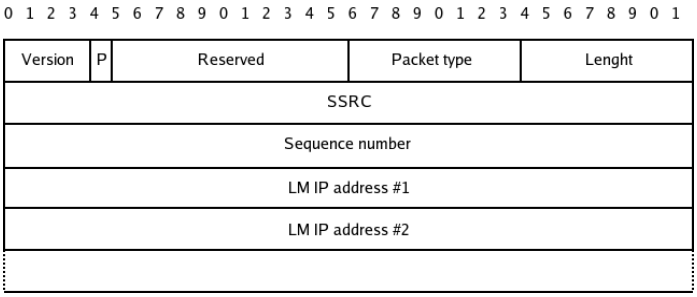

Fig. 6. Landmark packet (LM)

The FT is shown in Fig. 7. The message consists of the following fields: general header with payload type=2, sequence number of LM packet (32bits) which is used to identify the current list of ID for landmark IP addresses, first group feedback target (=summarization node) IP address (32bits), feedback target port (16bits), group size (32bits) allowing a receiver to calculate the proper RR packet transmission interval RI_{RR} from equation (2), vector specification (10bits) and length of landmark ID vector list in 32-bit words (6bits). The purpose of the vector specification field is to set the vector accuracy. The landmark vector could be reduced to only the several most significant values, i.e. to landmarks closest to the session member. The vector size then depends on accuracy required when identifying a session member position. Because the tree specification data can be too large to be sent in one FT packet, the information should be encapsulated into several packets.

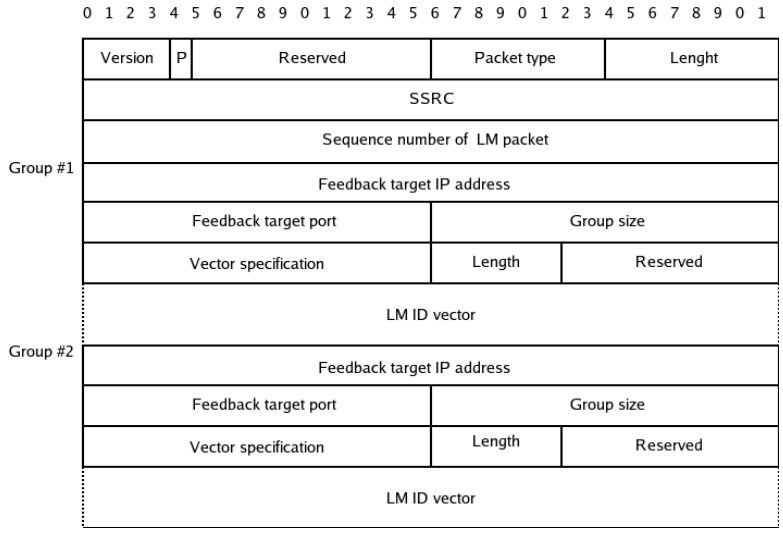

Fig. 7. Feedback target packet (FT)

4.2 Transmission form Receivers to Source

Every session receiver sends its evaluated landmark IDs vector to the sender periodically in the landmark vector packet (LMV) shown in Fig. 8. The LMV packet is sent using a unicast connection. The sequence number included is the number receiver from the last LM packet. As mentioned above, this allows the source to check whether a receiver use the current set of landmarks and their corresponding IDs.

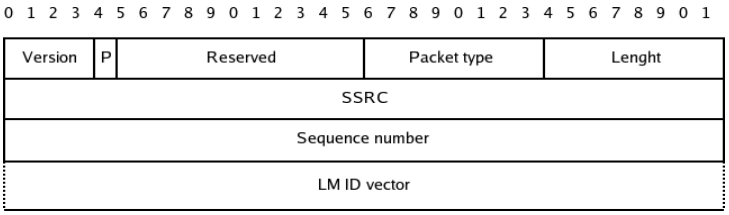

Fig. 8. Landmark vector packet (LMV)

4.3 Initiation of Feedback Transmission

This section describes how a new receiver initiates the feedback transmission in a previously established multicast session, for example using SDP (Session Description Protocol) [11]. When a new receiver joins a session, it starts to transmit its feedback directly to the source, i.e. it immediately belongs to the group where the source behaves as a summarization node. This scenario means that no immediate communication is required. This feature is quite important in case of massive session joining, for example, when an interesting IPTV program is beginning to start broadcast. After a specific time which depends on the LM packet transmission period, the new receiver will receive information about available landmark set and their corresponding IDs, see Fig. 9. Using this information, the receiver evaluates its landmark IDs vector and sends it to the sender in the LMV packet. After another time interval, the receiver will receive the FT packet with information identifying a new summarization node to send feedback to. Then the process continues by sending the LM, FT and LMV packet in specific periods of time. The LMV packet periodic transmission allows the source to check whether the receiver is still participating in the session.

5 Conclusion and Future Work

The most advanced method known for SSM feedback transmission is hierarchical aggregation, which uses a tree consisting end and summation nodes. A summarization nodes acts as a feedback target for a group of end nodes. For the purpose of end nodes clustering into groups, we use the binning algorithm. The algorithm partitions end nodes into bins and end nodes within a bin are thought to be relatively close. The binning algorithm uses a set of landmark nodes placed on the network and each end node evaluates RTT values of these landmarks. Then, end nodes are assigned to bins on the basis of a vector consisting of landmarks ordered by increasing RTT values and nodes from a bin send their feedback to the closest summarization node.

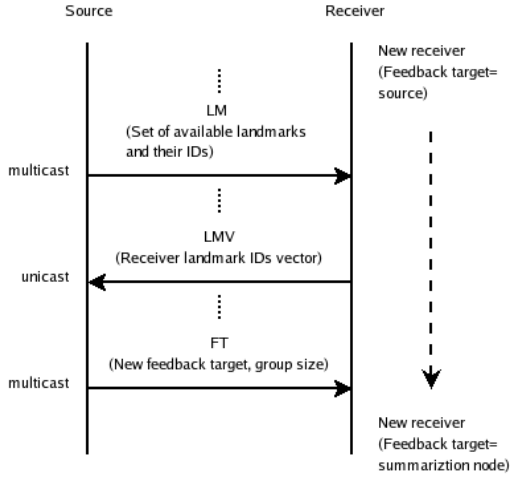

Fig. 9. New receiver joins a session

In this paper, we proposed the MFCP protocol which is used to establish and manage a tree structure for hierarchical aggregation. The main idea of the MFCP protocol is that end and summarization nodes send their landmark vectors to the source periodically. A vector consists of a list of landmark IDs starting with the landmark with the lowest RTT value. When the source knows all landmark vectors, it can calculate the required tree structure for hierarchical aggregation. The tree structure establishment covers two conditions as follows: 1) It has to meet the identified number of end nodes in groups n_{gend} and the identified number of summation nodes in groups n_{gsmr} 2) It has to assure low routing complexity for the feedback transmission. The source finds the tree structure by changing summarization nodes from an available set of dedicated servers. If this method does not work, it can create a fake vector of selected summarization nodes. Then the source sends the calculated tree structure information to receivers.

The MFCP protocol is simple since it uses only three packet types (LM-landmark packet, LMV-landmark vector packet and FT-feedback target packet). The protocol is scalable in terms of massive number of receivers joining a session since no immediate communication is needed. The convergence time for changes in the tree structure is strongly affected by transmission intervals of MFCP packets. Small transmission intervals could cause unnecessary network load. For the purpose of proper interval settings, we plan to test the protocol in the PlanetLab network. A similar RTT measurement method among PlaneLab nodes is presented in [12]. As this network consists of over 700 hundred nodes spread throughout the world, the test results should give an accurate overview of the MFCP protocol world-scale use.

Acknowledgement. This work was supported by the Academy of Sciences of the Czech Republic project 1ET301710510.

References

1. HOLBROOK H., CAIN B. Source-Specific Multicast for IP. Request for Comments: 4607. Internet Engineering Task Force. 2004.
2. BHATTACHARYYA, S. An Overview of Source-Specific Multicast (SSM). Request for Comments 3569. Internet Engineering Task Force. 2003.
3. SCHULZRINNE H., CASNER S., FREDERICK R., JACOBSON V. RTP: A Transport Protocol for Real-Time Applications. Request for Comments 3550. Internet Engineering Task Force. 2003.
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. HANDLEY M., SCHULZRINNE H., SCHOOLER E., ROSENBERG J. SIP: Session Initiation Protocol. Request for Comments 2543. Internet Engineering Task Force. 1999.
6. CHESTERFIELD J., SCHOOLER E. An Extensible RTCP Control Framework for Large Multimedia Distributions. Proceedings of the Second IEEE International Symposium on Network Computing and Applications. IEEE Computer Society. 2003.
7. CHESTERFIELD J., SCHOOLER E., OTT J. RTCP Extensions for Single-Source Multicast Sessions with Unicast Feedback. Internet Draft, work in progress. Internet Engineering Task Force. 2007.
8. CHESTERFIELD, J., SCHOOLER, E., OTT, J. RTCP Extensions for Single-Source Multicast Sessions with Unicast Feedback. Internet Draft, work in progress. Internet Engineering Task Force. 2004.
9. CASTRO M., DRUSCHEL P., KERMARREC A. , ROWSTRON A. A large-scale and decentralized application-level multicast infrastructure. IEEE Journal on Selected Areas in Communications. IEEE. 2002.
10. RATNASAMY S, HANDLEY M, KARP R, SHENKER S Topologically-Aware Overlay Construction and Server Selection. Proceedings of 21rd Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE. 2002.
11. HANDLEY M., JACOBSON V., PERKINS C. SDP: Session Description Protocol. Request for Comments 4566. Internet Engineering Task Force. 2006.
12. TANG L., CHEN Y., FEI L., ZHANG H., JUN L. Empirical Study on the Evolution of PlanetLab. Sixth International Conference on Networking - ICN 07. IEEE. 2007