One Source Multicast Model Using RTP in NS2

Milan Simek, Dan Komosny, **Radim Burget**

Brno University of Technology, Department of Telecommunications, Czech Republic

Summary
The simulation models in general are irreplaceable tools for testing and validating of the present or the proposed algorithms. This paper deals with the design of the multicast model with the one source using the RTP (Real Time Protocol) transport protocol for data delivering to the great number of receivers. The simulation results show the present situation in the sharing of control bandwidth for RTCP feedback reports among all multicast receivers. The main object is to propose the efficient multicast model in the ns2 for the new feedback algorithm implementation. Input here the part of summary.

Key words:
Multicast, RTP, ns2, feedback, simulation

1. Introduction

The issue of the control RTCP (Real Time Control Protocol) bandwidth sharing by the all multicast receivers is very popular subject at the present time, because with the increasing popularity of the Internet and the bandwidth capacity growth, the Internet comes to be the most appropriate solutions for the multimedia service appointment such as IPTV (Internet Protocol Television) or VoD (Video on Demand) and it is expected that more and more users is going to use these multimedia services. These services have to offer the QoS management to be interesting for the present and new clients expecting the 100% service availability and reliability. In the RTP multicast session, the receivers joined in multicast group are sending in the accurate intervals (to divide the control bandwidth among all participants) the feedback reports that contains the packet-loss and quality of the data delivery information. The service provider is taking advantage of these reports for the flexible end effective responding at the data delivery problems. By all means, with the growth of the participants count receiving data from the multicast group, the value of the report interval is increasing, thus the delay when the service provider is able to react at the data delivery problem is increasing also and if the customer's request is not handled in the expected time, the service becomes unreliable and uninteresting for the customer. At the present time, the number of IPTV customers is on the level that making this service still reliable, but with the growth of the Internet popularity it is expected more IPTV customers, whereon the present algorithm for the RTCP protocol is not ready for. The proposed multicast model represents the issue of the feedback report interval for the great number of receivers and in the future will serve for the implementation of the new algorithms that will be used for the feedback report interval reduction. The IPTV service takes advantage of the multicast model called SSM (Source Specific Multicast), but the cooperation of the RTP and SSM models in ns2 will be proposed in the future, thus we have used the classical internet multicast model using PIM-SM (Protocol Independent Multicast – Sparse Mode) [4].
The paper is organized as follows. Section 2 and 3 deal with the RTP/RTCP and multicast overview and their implementation in the ns2. In the fourth section, the basic simulation model together with the extended model and their results are presented. And the Section 5 describes the simulation model features and proposes the future work for the new algorithm implementation.

2. RTP/RTCP

The RTP (Real Time Protocol) is used by the application theirs data have the real-time character and the lower UDP protocol is applied for the transport, [1]. The RTP is in the most cases used in the IP multicast environment. This protocol is composed of the two parts, the RTP part, that cares about data delivery and the RTCP (Real Time Control Protocol) part for the transport and management of the feedback report from all participant of the RTP session. In accordance with [1], the main task of the RTCP is sending in periodic interval feedback reports that are important for the monitoring and maintaining of the quality RTP packets delivering. The format of the RTCP protocol is depending whether the participant is active sender or not. The SR (Sender Report) serves for sending and receiving statistics from the active sending participant and RR (Receiver Report) for the participants that stands as only the receivers. The each RTCP protocol contains the transport-level identifier of the RTP source referred to CNAME (Canonical Name) for the maintaining of the transport path from receivers toward the source. For calculating of the feedback report interval, let's call a rint (we are using it in the next sections), the applications needs to know the number of participants in the time of the report transport, thus this information is included in the each RTCP packet. According to [1], the 5 % of the RTP session bandwidth is reserved for the RTCP transmission,

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

let's call it control traffic, and the ¾ of this control traffic is used for the RTCP/RR packets and ¼ of control traffic for the RTCP/SR. These fractions are valid if the number of sources is less than ¼ of all receivers and this condition is valid for our situation, because we have proposed the multicast model with only one source. The [1] introduces the constant referred to "C" equals to size of the average RTCP packet divided the participant control traffic and constant referred to "n' equals numbers of participants. For the interval calculation rint = C * n, we are using the receiver control traffic (0,75*0,05*session bandwidth) and the n = number of receivers. It is in evidence, that with the increasing number of the receivers, the report interval will reach the undesirable values. The dependency of the report interval values at the number of receivers for the 1Mb/s and 100kbit/s session bandwidths is shown in the Figure 1. The achievement of the same results is the object of our proposed models.

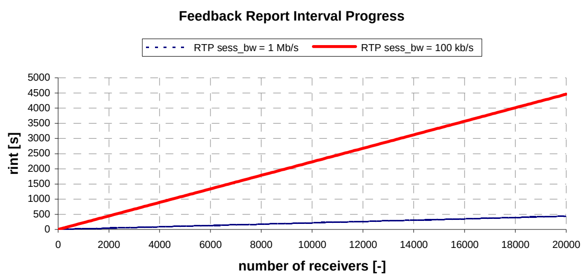

Fig. 1 The dependency of the feedback report interval at the number of receivers.

3. Network Simulator v2

For the model design, we have used the ns2 simulator. The ns2 is popular simulation tool for the network behaviour modelling proposed by UC Berkeley for the educational and the research needs [3]. It is the object simulator written in C++ with the Otcl language for the object definition. Users define required protocols in C++ and Otcl that are represented by object inherited from the Agent class. The ns2 use NAM (Network Animator) utility for the animation of the correct protocol design and the XGRAPH tool for the representation of the achievement results. The ns2 simulator supports the great scale of the protocols (TCP, UDP, RTP, i.e.) and the technologies (LAN, MAN, sensor networks, sessions, multicasting i.e.) in the wired and also in the wireless networks.

3.1. Multicast in Ns2

The ns2 support the basic multicast protocols such as the PIM-DM (Protocol Independent Multicast-Dense Mode), PIM-SM (Sparse Mode) [4] and the simulation with the PIM-SSM (Source Specific Mode) after installation of the required patches is possible also [7]. In the next sections, the basic of the ns2 multicast simulations are described. For the computation of the delivery tree and the specification of the routing protocol, the "*mrtproto {}*" process should be used [4].

PIM-Dense Mode.

This model of multicast applies so called "push" model of data delivering. The active source multicast data to the whole network and the particular receivers have to announce their interest by means of the Graft message or in case the lack of interest with the Prune messages [6]. This multicast model is supported in ns2 with the: "*$ns mrtproto DM*" command [4].

PIM - Sparse Mode

In the Sparse mode, the multicast data are delivered only toward the participants, that are requesting for this traffic and this model is referred to "pull" model. The PIM-SM takes advantage of the RP (Rendezvous Point), the specific router in the network where all participants of the multicast session register to it. The data from the source towards the RP are delivered by the unicast way and subsequently, the RP forwards data via the Shared Path Tree referred to (*,G). After the first packet is received, the switch-over process is performed for the data delivering via the Shortest Path Tree [5]. For the application of this multicast model should be used the following command: "*$ns mrtproto CtrMcast*". The "*mrthandle set_c_rp $node*" command configures the particular RP in the network.
The Ns2 also implements simplified model of the sparse mode with the "*$ns mrtproto ST*" command and the model of the Bi-directional Shared Tree Mode with the "*$ns mrtproto BST*" command. The main multicast core files are included in the "*ns2/tcl/mcast*" folder.

PIM-Source Specific Mode

The current version of the ns2 simulator ns-2.31 (March, 2007) does not include the classes for the SSM (Source Specific Multicast) support. However, the interest users may use the SSM implementation proposed by the research group at the Coimbra University [7]. The function of the SSM model is derived from the CtrMcast class with the particular modifications. In the SSM model, no RP is needed any more and the participants invoke join-group method with the two arguments (S-source address, G- multicast group address) in the contrast to CtrMcast class that defines join-group method with the one argument (G-multicast group). The simulation with the PIM-SSM protocol is configured by the "$*ns mrtproto SSMMcast*" command.

3.2. RTP Support in Ns2

The specific protocols are in the ns2 implemented as the Agents that are predefined in the particular classes. The RTP and RTCP protocols are implemented as the RTPAgent class (/ns2/common/rtp.cc) and the RTCP Agent class (/ns2/apps/rtcp.cc), both classes are implemented in C++. These agents stand for the transport agents serving for the generating, the sending and the receiving of the packets. For the management of the mentioned procedures, the RTPSession (/ns2/tcl/rtp/session-rtp.tcl) class standing for the managing of whole RTP session is required. RTPSession class defines the procedures for the report interval calculation, it maintains the participant's tables etc. The steps of the creating of the new RTP source by means of the RTPSession class are described in the Figure 2. With the new session, the four objects are created. They are Agent/CBR/RTP, Agent/RTCP, RTPTimer and RTPSource. Then, the session is joined to the multicast group, the RTP agent and RTCP agent are joined to the separate multicast groups. It is the adequate solution for usage of the same multicast group with the different ports. The start procedure initializes the RTCP agent whilst, the transmit procedure launches the RTP agent. The RTPSession class is implemented in accordance with the obsolete RTP standard from the 1996 [2]. The present standard [1] differs from [2] in the algorithm for report interval calculation, above all in the assessment of the RTCP average packet size, it introduces the timer reconsideration and the reverse reconsideration, the rules for the participant's tables maintaining etc. In our multicast model, we have used this class that implements the draft from 1996 [2], but in the future we will work at the implementation of the new RTP standard [1] into the ns2 environment.

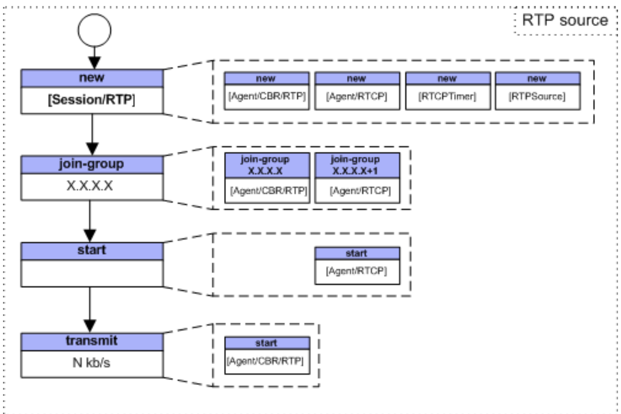

Fig. 2 The steps of the RTP source design and initialization

4. Experimantal Scenario

4.1. Basic Model Configuration

Our object is to propose the IPTV multicast model with the great number of customers and demonstrate the undesirable effect of the RTCP feedback reporting with this number of interesting receivers using the common unicast way for the feedback report announcement. At the begin of our work, we have proposed the basic multicast model (see Figure 3.) with the one source (attached to node0) and with the 8 participants (6 interest receivers) that are joining to the multicast group in the 12 second interval if the first receiver (node 8) is joining in the time of 18 sec, the second one in the 30 sec and so on.

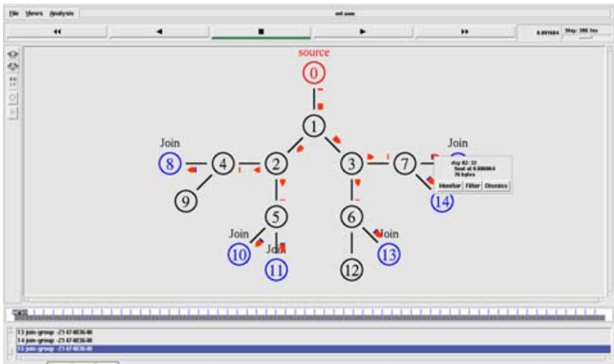

Fig.3 The basic model topology

The first receiver serves above all as the monitor unit, it means that all results in the charts are values that calculates this receiver. How was mentioned above in the section 1, the ns2 does not implement the SSM model, that's why the RTPSession class has no procedures for this multicast model. Hence, we have used the classical internet multicast model with the PIM-SM and the Rendezvou Point, that was configured on the node 0. The proposed testbed uses the link with the 100Mb/s banwidth and 10ms link-delay. The session bandwidth was set up at the 1Mb/s (51 kb/s for control traffic). The [1] defines the lowest interval boundary for the feedback reports to avoid having a burst of packets. It means, if the calculated interval is lower than this boundary, the report interval is set at the value corresponding with the boundary and we refer to this function as the ReducedLimit. Reduced limit equals the 360sec divided the session bandwidth, which means that this value equals 5 second for the session bandwidth of the 72 kbit/s. This value (5 seconds) is also recommended value for the minimum report interval [1], whereas the new participants after join to group can use the half interval for the faster determination of the parameters necessary for the report interval calculation. The default parameter in the RTPSession class is 1s and we have used this value also by reason of the small topology. The simulation time was set up at 100s. In addition, the [1] defines the final report interval value as the product of the rint value and the random value with the boundaries of [0.5, 1.5], but we have not implemented this function because of the possibility of the assessment of the precise value from the result charts.

4.2. Basic Model Results

In the Figure 4a), there are the results from the simulation model. We have focused on the receiver 8 (attached on the node 8) and displayed its calculated report interval during the whole simulation. The results indicates, that receivers used the half minimum interval of 0,5 sec after his joining. How the other participants were joining to the given group, the receiver in the periodic interval recalculates the time for the next report. The value was set up at the minimum interval of 1s because the ReducedLimit function was turned on. For the verification of the calculated report interval values that the model performs, we have reconfigured the simulation model without the ReducedLimit function. The results from this simulation are shown in the Figure 4b. In the time of 18 second (receiver 8 is alone in the network), the report interval is equal to 24 ms and subsequently the report interval is linearly increasing during the simulation process when the other receivers are joining. After the last receiver has joined to the group, the final value for the report interval was settled to the value around 140ms.

For the verification of the multicast model results we have compared them with the formula for the calculating of the report interval (1) that is defined in [1]. We have used the avgsize value from the simulation model also.

$$r\,\mathrm{int} = C \cdot n = \frac{8 \cdot avgsize}{0{,}75 \cdot 0{,}05 \cdot sessbw} \cdot n = \frac{8 \cdot 108{,}19}{0{,}75 \cdot 0{,}05 \cdot 1Mbps} \cdot 6 = 0{,}132s \qquad (1)$$

The results from the formula (1) shows, that the proposed multicast model gives the correct values of the calculated report interval. The little difference is given by the inaccuracy of value assessment from the chart.

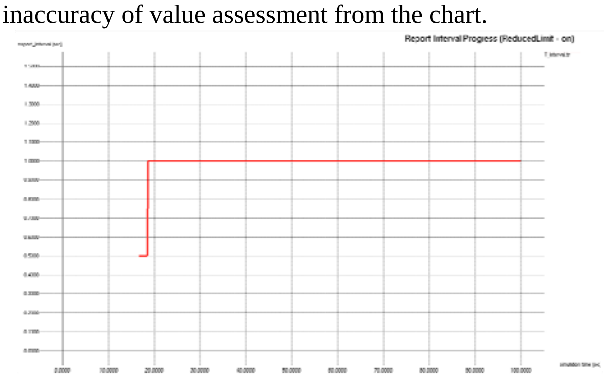

Fig.4a) The report interval progress with the ReducedLimit function

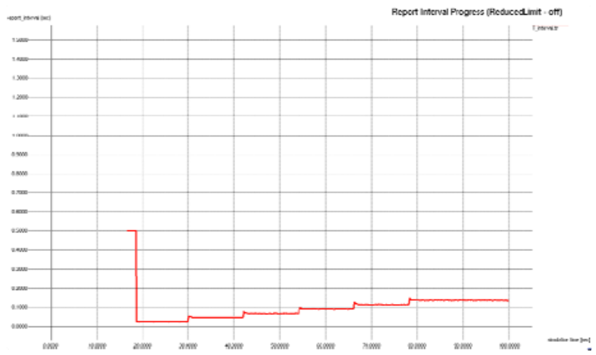

Fig.4a) The report interval progress without the ReducedLimit function

4.3. Extended Multicast Scenario

How was mentioned above, our object was to design the multicast model with the great number of receivers, thus we have proposed the new topology where the number of receivers is not statical, but the receivers are added to the topology dynamically. The new topology is shown in the Figure 5. The receivers being joined to the multicast group in the periodic interval of 1 second and the total number of them was set up at the 150. The ReducedLimit function was turned on and the minimal interval was changed at the 5 seconds in accordance with the recommendation in the [1]. The parameters used for the simulation are shown in the Table 1.

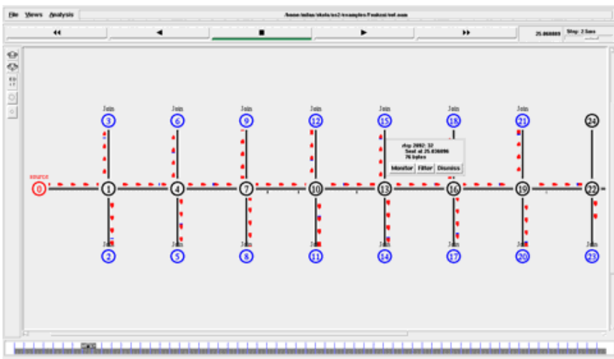

Fig.5 The extended model topology with the dynamical number of receivers

4.4. Extended Model Results

Progress of the report interval calculation is shown in the Figure 6. Because of the ReducedLimit function, the lowest value of the report interval was 5 seconds (except the half minimum interval at the start of the simulation) and after the last receiver has joined to the multicast group, the value of the report interval was determined at the 18,5 seconds, but in accordance with the formula (4-1) and the simulation conditions, the value around 16 seconds was expected. This difference is probably caused by the short interval of joining to the given group or the simulation strenuosity, but the real reasons will be object of the future work.

Table 1: The simulation parameters

Parameter	Value
Link bandwidth	100Mb/s
Link delay	10 ms
Session bandwidth	200kb/s
Join interval	1 sec (first receiver at 4 sec)
Number of receivers	150
Simulation time	200 s
Reduced Limit	on (5 sec)

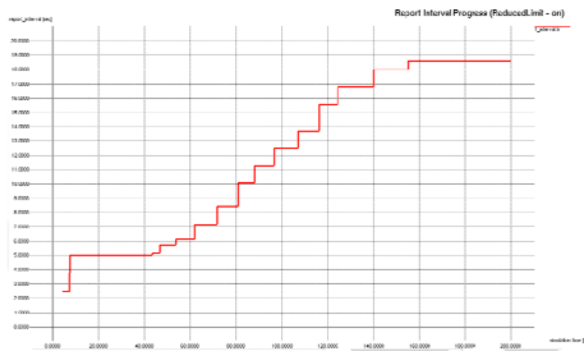

Fig.6 The progress of the report interval calculation for the model with the 150 receivers

5. Future Work and Conclusion

How was mentioned above, the next work at the ns2 multicast model will include:

(i) The clarification of the results in the chapter 4.4 and give the ns2 model precision for computation of the report intervals,
(ii) The implementation of the RTP standard accordance to [1] into the ns2 environment,
(iii) The implementation of the new RTPSession class (inherited from the RTPSession class) for the application with the SSM multicast model ,
(iv) The designing of the multicast model with the new disposed TTP protocol (Tree Transmission Protocol) for the reduction of the report interval.

We have considered the design issue of the multicast model with one source in ns2 environment. In this paper, we have presented a brief overview of the RTP/RTCP protocol and the multicast delivery models and presented the main issue with the using the common RTCP feddback reporting in the multicast environment. For the multicast model design we have used the RTPSession class of the ns2 and in the paper we have presented its main features. By reason of the demonstration of the strange impact of the increasing number of receivers on the calculated report interval, the basic and the extended multicast model were proposed. The proposed ns2 models offer the acceptable results, but into the future the RTP implementation in the ns2 environment has to be rebuilded for our purposes. At the end of our paper, we have summarized the future work for our research.

Acknowledgement
This work was supported by Grant Agency of the Czech Republic - project No. 102/07/1012.

References

[1] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, “RTP: A Transport Protocol For Real-Time Application”, Internet Draft, rfc1889, March 2, 2003.
[2] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, “RTP: A Transport Protocol For Real-Time Application”, Internet Draft, rfc 3550 January, 1996.
[3] Information Sciences Institute, “The Network Simulator – ns-2”, June 2004, <http://www.isi.edu/nsnam/ns/>
[4] K. Fall, K. Varadhan, “The ns Manual”, October 3, 2007, <http://www.isi.edu/nsnam/ns/ns-documentation.html>
[5] B.Fenner, M. Handley, H. Hoolbrook, I.Kouvelas,” Protocol Independent Multicast – Sparse Mode (PIM-SM)”, draft-ietf-pim-sm-v2-new-09.txt, February 2004.
[6] A. Adams, J. Nicholas, W. Siadak, “Protocol Independent Multicast – Dense Mode (PIM-DM)”, rfc3973, January 2006.
[7] T. Camilo, J. Silva, F. Boavida, “Enabling NS-2 with SSM Enviroments”, 2004, < http://eden.dei.uc.pt/~tandre/ssm_extension/index.htm >

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 the research focused on the analysis and the simulation of the multicast operations mainly in Ns2 and SDL environment.

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

Radim Burget was 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