SCALABILITY SOLUTIONS FOR MULTIMEDIA REAL-TIME CONTROL PROTOCOL (PDPTA'06)

Omar Essa and Nabaway El-Ramly
Faculty of Science, Menoufia university

Hany Harb
Faculty of Engineering, Al-Azhar University

Khaled kamel
College of Science and Technology, Texas Southern university
kamelka@tsu.edu

Keywords: Internet services, Internet telephony, Real-time Control Protocol, Datagram Protocol/Internet Protocol, Quality of Service.

Abstract: Recently, some problems related to using the Real-time Control Protocol (RTCP) in a very large dynamic group have arisen. Some of these problems are: feedback delay, increasing storage state at every member, and ineffective RTCP bandwidth usage, especially for the receivers that obtain incoming RTCP reports through low bandwidth links. More schemes are proposed to alleviate the RTCP problems. The famous and the recent one, which was introduced by EL-Marakby and was named Scalable RTCP (S-RTCP), still have some drawbacks. This paper demonstrates a brief discussion and evaluation of existing schemes and algorithms. Comments on the proposed scheme and algorithm are introduced. Consequently, we demonstrate the design of More Scalable RTCP (MS-RTCP) scheme based on hierarchical structure, distributed management, and EL-Marakby scheme. Also, we show how our scheme will alleviate all the drawbacks that found in the S-RTCP. Finally, we will highlight this scheme implementation, analysis, and its performance evaluation.

1. INTRODUCTION

When we describe media as real-time, we mean simply that the receiver is playing out the media stream as it is received, rather than simply storing the complete stream in a file for later playback. In the ideal case, play out at the receiver is immediate and synchronous; although in practice some unavoidable transmission delay is imposed by the network.

The primary requirement that real time media places on the transport protocol is for predictable variation in network transit time. Consider, for example, an IP telephony system transporting encoded voice in 20-milliseconds frames. The source will transmit one packet every frame, and ideally we would like those to arrive with the same spacing so that the speech they contain can be played out immediately. Some variation in transit time can be accommodated by the insertion of additional buffering delay at the receiver, but this is possible only if that variation can be characterized and the receiver can adapt to match the variation [1], [2].

A lesser requirement is reliable delivery of all packets by the network. Clearly reliable delivery is desirable, but many audio and video applications can tolerate some loss: In our IP telephony example, loss of a single packet will result in a dropout of one fiftieth of a second, which, with suitable error concealment, is barley noticeable. Because of the time-varying nature of media streams, some loss is usually acceptable because its effects are quickly corrected by the arrival of new data. The amount of loss that is acceptable depends on the application, the encoding method used, and the pattern of loss [3].

These requirements drive the choice of transport protocol. It should be clear that TCP/IP is not appropriate because it favours reliability over timeliness, and our applications require timely delivery. A User Datagram Protocol (UDP)/IP-based transport should be suitable, provided that the variation in transit time of the network can be characterized and loss rates are acceptable.

The standard Real-time Transport Protocol (RTP) builds on UDP/IP, and provides timing recovery and loss detection, to enable the development of robust systems.

2. OVERVIEW OF RTP AND RTCP

RTP provides transport services for real time applications, including IP telephony. It consists of two parts, which are the transport part itself (RTP), and a control and feedback component, called the Real Time Control Protocol (RTCP). Both RTP and RTCP are engineered for multicast multimedia conferences.

RTP is generally used in conjunction with the UDP, but can make use of any packet-based lower-layer protocol. When a host wishes to send a media packet, it takes the media, formats it for packetization, adds any media-specific packet headers, pre-appends the RTP header, and places it in a lower-layer payload. It is then sent into the network, either to a multicast group or uni-cast to another participant.

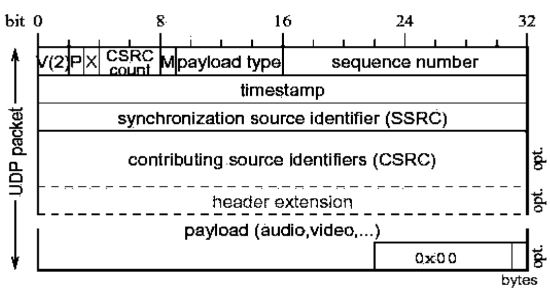

Figure1: RTP fixed header format.

The RTP header, in Figure 1, is 12 bytes long. The V field indicates the protocol version. The X flag signals the presence of a header extension between the fixed header and the payload. If the P bit is set, the payload is padded with zeroes to ensure proper alignment for encryption.

Participants (senders or listeners) in a multicast group are distinguished by a random 32-bit Synchronization Source (SSRC) identifier. Having an application-layer identifier allows to easily distinguish streams coming from the same translator or mixer and associate receiver reports with sources. In the rare event that two users happen to choose the same identifier, they redraw their SSRCs.

RTP supports the notion of media-dependent framing to assist in the reconstruction and play out process. The marker bit, M, provides information for this purpose. For audio, the first packet in a voice talk spurt can be scheduled for play out independently of those in the previous talk spurt. The bit is used in this case to indicate the first packet in a talk spurt. For video, a video frame can only be rendered when its last packet has arrived. Here, the marker bit is used to indicate the last packet in a video frame.

The Payload Type (PT) identifies the media encoding used in the packet. The Sequence Number (SN) increments sequentially from one packet to the next, and is used to detect losses and restore packet order. The Timestamp (TS), incremented with the media sampling frequency, indicates when the media frame was generated.

RTP supports the notion of a mixer. A mixer is a device that takes RTP streams from many users, and combines them into a single stream containing a mix of those streams. However, it is still desirable to indicate which participants have their audio mixed in a particular packet. This is the purpose of the Contributing SSRC or CSRC field. It contains a list of SSRC for those users mixed in a packet. The field is optional. The payload itself may contain headers specific for the media [4], [5].

3. RTCP MANAGEMENT

RTCP is the companion control protocol for RTP. Media senders (sources) and receivers (sinks) periodically send RTCP packets to the same multicast group (but different ports) as are used to distribute RTP packets. Each RTCP packet contains a number of elements, usually a Sender Report (SR) or receiver report followed by Source Descriptions (SDES). Each serves a different function.

SR is generated by users who are sending media (RTP sources). They describe the amount of data sent so far, as well as correlating the RTP sampling timestamp and absolute ("wall clock") time to allow synchronization between different media.

Receiver Reports (RR) are sent by RTP session participants who are receiving media (RTP sinks). Each such report contains one block for each RTP source in the group. Each block describes the instantaneous and cumulative loss rate and jitter from that source. The block also indicates the last timestamp and delay since receiving a sender report, allowing sources to estimate the Round Trip Time (RTT) to RTP sinks.

SDES packets are used for session control. They contain the Canonical Name (CNAME), a globally unique identifier similar in format to an email address. The CNAME is used for resolving conflicts in the SSRC value and to associate different media streams generated by the same user. SDES packets also identify the participant through their name, email, and phone number. Client applications can display the name and email information in the user interface. This allows session participants to learn about the other participants in the session. It also allows them to obtain contact information (such as email and phone) to enable other forms of communication (such as initiation of a separate conference using Session Initiation Protocol (SIP). This also makes it easier to contact a user should he, for example, have left his camera running [5].

If a user leaves an RTP session, they send a BYE RTCP message. Finally, Application (APP) elements can be used to add application-specific information to RTCP packets.

RTCP is the companion control protocol for RTP. Media senders (sources) and receivers (sinks) periodically send RTCP packets to the same multicast group (but different ports) as are used to distribute RTP packets. Each RTCP packet contains a number of elements, usually a Sender Report (SR) or receiver report followed by Source Descriptions (SDES). Each serves a different function.

SR is generated by users who are also sending media (RTP sources). They describe the amount of data sent so far, as well as correlating the RTP sampling timestamp and absolute ("wall clock") time to allow synchronization between different media.

Receiver Reports (RR) are sent by RTP session participants who are receiving media (RTP sinks). Each such report contains one block for each RTP source in the group. Each block describes the instantaneous and cumulative loss rate and jitter from that source. The block also indicates the last timestamp and delay since receiving a sender report, allowing sources to estimate the Round Trip Time (RTT) to RTP sinks.

SDES packets are used for session control. They contain the Canonical Name (CNAME), a globally unique identifier similar in format to an email address. The CNAME is used for resolving conflicts in the SSRC value and to associate different media streams generated by the same user. SDES packets also identify the participant through their name, email, and phone number. Client applications can display the name and email information in the user interface. This allows session participants to learn about the other participants in the session. It also allows them to obtain contact information (such as email and phone) to enable other forms of communication (such as initiation of a separate conference using Session Initiation Protocol (SIP)). This also makes it easier to contact a user should he, for example, have left his camera running [5].

If a user leaves an RTP session, they send a BYE RTCP message. Finally, Application (APP) elements can be used to add application-specific information to RTCP packets.

4. SCALING RTP

Scaling RTP to large groups requires scaling both RTP and RTCP. The use of silence suppression for audio enables scalability of the media transport. In normal conversations, only one or two people can be talking at a time, providing a natural scaling factor. For video, scalability can be provided by a similar means: only the person that is talking sends video updates. Alternatively, a mixer can be used, and the mixer distributes video streams only for those participants currently talking.

Scaling RTCP is more complex. Since it contains timely feedback information, it must be sent constantly. As such, the principal difficulty in achieving RTCP scalability to large group sizes is the rate of RTCP packet transmissions from a host. If each host sends packets at some fixed interval, the total packet rate sent to the multicast group increases linearly with the group size, N. This traffic would quickly congest the network, and be particularly problematic for hosts connected through low-speed dialup modems. To counter this, the RTP specification requires that end systems utilizing RTP listen to the multicast group, and count the number of distinct RTP participants which have sent an

RTCP packet. These results in a group size estimate, L computed locally at each host. The interval between packet transmissions is then set to scale linearly with L. This has the effect of giving each group member (independent of group size) a fair share of some fixed RTCP packet rate in the multicast group.

Specifically, each user *i* in a multicast group using RTP maintain a single state variable, the learning curve, which we denote as L(t). This variable represents the number of other users that have been heard from by *i* at time t. "Heard from" means validated, as specified in RTP. A participant is validated once an RTCP packet from them is received, or once two RTP data packets have been received. The state is initialized to L(0) = 1 when the user *i* joins the group.

Each user multicasts RTCP reports periodically to the group. These reports contain information such as QoS feedback, user information, and messages for management of loosely controlled multimedia sessions. In order to avoid network congestion, the total amount of RTCP reports multicast to the group is set at 5% of the total multicast session bandwidth. The multicast session bandwidth is a fixed parameter, known to all participants. It is normally advertised through an announcement protocol, such as the Session Announcement Protocol (SAP) [6], or can be entered manually. The session bandwidth is chosen by the session creator. To meet these criteria, each user increases the period of their reporting as the group size increases. Specifically, let C be desired RTCP packet inter-arrival time between RTCP packets from any user, computed as the average RTCP packet size divided by 5% of the session bandwidth. Each user computes their deterministic interval as in the following equation:

$$T_d = \max(T_{\min}, CL(t))$$

Where T_{min} is 2.5s for the initial packet from the user, and 5s for all other packets (this allows the initial packet to be sent quickly, to help speed up the validation process). The deterministic interval is the target average interval between RTCP transmissions from a specific user. To avoid synchronization, the actual interval is then computed as a random number uniformly distributed between 0.5 and 1.5 times T_d [7]. Figure 2 summarizes the RTCP algorithm for currently used schemes.

```
New interval := C * current group size estimate;
New interval := max(new interval, Tmin);
New interval := new interval * random factor;
Send packet();
Schedule timer(current time + new interval);
```

Figure 2: RTCP Algorithm

The above scaling mechanism works well for small to medium sized groups (up to perhaps a few hundred members). However, it suffers from problems when applied to larger groups, particularly ones whose group membership is dynamic. These problems can be classified as congestion, state storage, and delay. We note that the problems with RTP scalability are common to any application that uses distributed feedback. Similar problems have been observed for generating feedback for reliable multicast [8], and for counting participants in video conferencing sessions [9].

5. SOLUTIONS AND PROBLEMS

A number of solutions for scaling feedback have been proposed, both in general and in the context of RTP. These solutions include polling and summarization as general solutions, and for RTCP specifically, they include adding summarizers to aggregate reports from regional or administrative areas, turning off RTCP reporting entirely, TTL scoping, using separate multicast groups for RTCP reports, and using the BYE message for QoS reporting, among others [4, 5, 9, 10, 11, 12, and 13].

RTCP scalability problems are due to the dynamic increase of the group size on the fixed low bandwidth. The RTCP scalability problems don't appear on the small RTP session. So the solution idea for all RTCP scalability problems will be based on how to transform the large group into small groups. This can be achieved by transformation of the RTP session to tree graph taking in consideration some factors that are important to be present, like:

1) Managing of the small groups.
2) Load balancing between small groups.
3) Persistency of the aggregated manager's data.
4) Flat tolerance in the system.
5) Determining of the tree levels and branches (scalability is wide or deep).

To satisfy the above features, we designed and implemented another scheme called MS-RTCP. The MS-RTCP has nine components, all of them are control entities except the managers and the children support for data and control functions. Each of which is responsible for one or more of the management tasks. These components are working in harmony to raise the quality of the scheme. A general overview of the MS-RTCP [14] scheme is shown in Figure 3 and Table 1.

6. SIMULATION RESULTS

In our analysis we used a network simulator called NS2 [15] that work with the following characteristics:

1- The RR size starts from 128 to 1Kb byte.

2- Session bandwidth equals 250 Kb/s.

3- Delay time between each subsequent user RR equals one second.

4- The simulation is done over the session seize equals (10, 20, ...) as a start.

5- The session size is dynamic.

6- The simulation contains some multiple sender cases.

7- The tree level equals three.

8- The timer and user number (inputs and output) are generated by a random number generator function.

9- The packet formation time is neglected. It is supposed that the packet is formed during the delay time (one second).

In the following discussion, we demonstrate the implementation results in relation to each RTCP problem. Concerned with scalability, our implementation will comprise a comparison between S-RTCP and MS-RTCP.

Figure 4 shows an average constant feedback delay of one second regardless of region and session size. Figure 5 shows the bandwidth utilization in Receiver Reports units (RRs). It is obvious that the number of RRs sent in the MS-RTCP and the S-RTP is approximately equal. Also, in the MS-RTCP curve, the number of RRs during the time interval 80s-90s decreases significantly. The significant decrease is caused by the fact that a large number of old children left the session and new children were gradually joining the session. Figure 6 and 7 show the scalability results using children distribution among the regions and the average number of RRs in each region respectively.

7. CONCLUSIONS

We have considered the problem of feedback for multicast multimedia sessions. We demonstrated the problems of congestion, latency, and state storage for the feedback mechanism currently defined in RTP. After an examination of other proposed solutions, we have designed a more scalable scheme for RTCP called MS-RTCP, in which members are organized dynamically in local regions. Every region has a basic manager and spare one. The manager monitors its region and sends the result to the up-level manager. At the end, the GM receives the data from the under-level managers, which represent the evaluation of the whole network (session). The basic idea of our scheme is built on the management distribution, dedicated manager for each essential function. Also our scheme overcomes all the previous schemes scalability drawbacks. Finally, we have evaluated the MS-RTCP performance in relation to all RTCP problems and system scalability by using a network simulator NS2.

8. REFERENCES

1. James F. Durkin., “Voice-Enabling the Data Network: H323, MGCP, SIP, QoS, SLAs, and Security,” Cisco Press, Indianapolis, 2003.

2. Behrouz A., Forouzan, "TCP/IP Protocol Suite," McGraw-Hill, London, 2000.

3. H. Sanneck and N. T. L. Le, “Speech property-based FEC for Internet Telephony applications,” in Proceedings of the SPIE/ACM SIGMM Multimedia Computing and Networking (MMCN), Jan. 2000.

4. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson., “RTP: A Transport Protocol for Real-Time Applications,” Internet Draft, March 2, 2003.

5. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson.,”RTP: A Transport Protocol for Real-Time Applications,” RFC 1889, January, 1996.

6. M. Handley, C. Perkins, E. Whelan, "Session Announcement Protocol," RFC 2974, October 2000.

7. S. Floyd and V. Jacobson, “The synchronization of Periodic Routing Messages,” in SIGCOMM Symposium on Communication Architectures and

Protocols (D. P. Sidhu, ed.), (San Francisco, California), PP. 33-44, ACM, Sept. 1993.

8. J. Nonnenmacher and E.W. Biersack, “Optimal multicast feedback,” in Proceedings of the Conference on Computer Communications (IEEE Infocom), March 1998.

9. J.C. Bolot, T. Turletti, and I. Wakeman, “Scalable feedback control for multicast video distribution in the internet,” in SIGCOMM Symposium on Communications Architectures and Protocols, (London, England), pp. 58–67, ACM, Aug. 1994.

10. B. Aboba, “Alternatives for enhancing RTCP scalability,” Internet Draft, Internet Engineering Task Force, Jan, 1997.

11. R. El-Marakby and D. Hutchison, “Design and performance of a scalable real-time control protocol Simulations and evaluations,” in Proc. of IFIP Conference on high Performance Networking (HPN’98), (Vienna Austria), Sept. 2000.

12. J. Rosenberg and H. Schulzrinne., ”Timer Reconsideration for Enhanced RTP Scalability.,” in Proceedings IEEE Info-com’98, San Francisco, California, March/April 1998.

13. R. Rivest, “The MD5 message-digest algorithm,” Request for Comments 1321, Internet Engineering Task Force, Apr. 1992.

14. O. Essa, H. Harb, and N. El-Ramly, “More Scalable Scheme based on management distribution for the Real Time Control Protocol.,” in IEEE International Symposium on Communication Systems and Digital signal Processing, July 2004.

15. McCanne, S. and Floyd, S. (McCanne, 1998) “LBNL Network Simulator,” 2003 version.

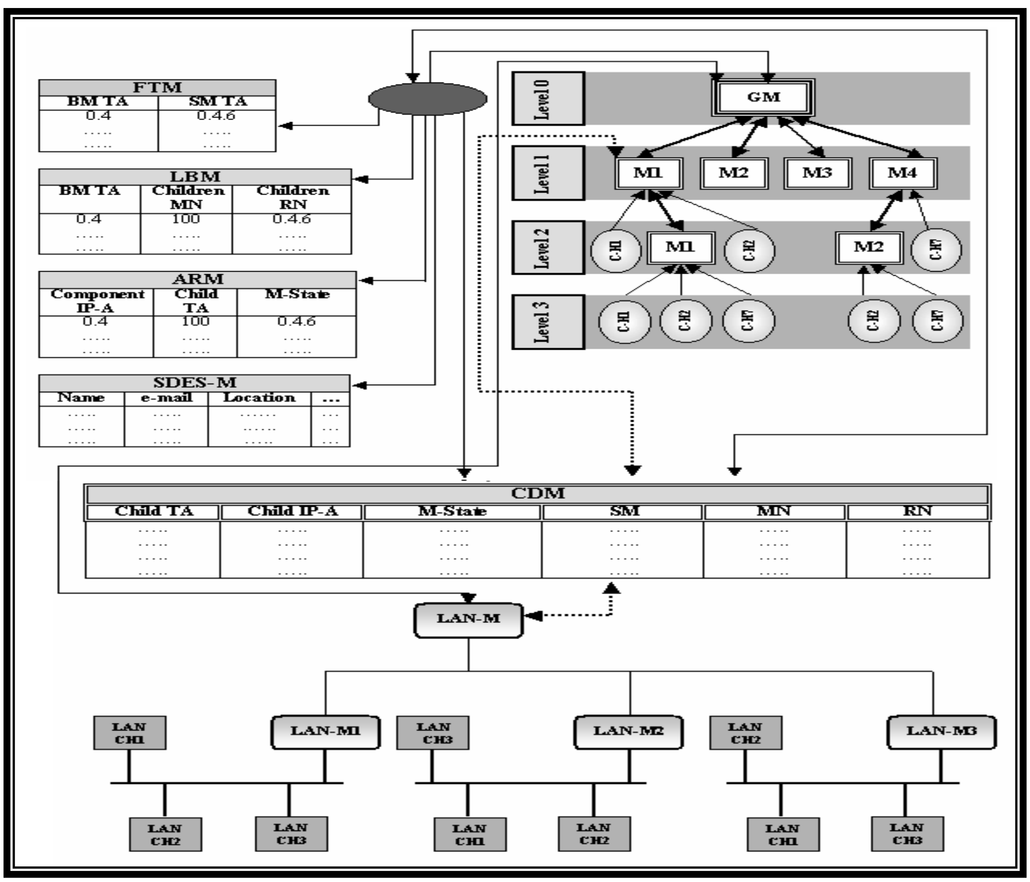

Figure 3: General View for the MS-RTCP

Symbol	Description	Symbol	Description
BM	Basic Manager	RN	Real Number
SM	Spare Manager	CH	Child
TA	Tree Address	→	Basic Connection
IP-A	IP Address	--------------	Spare Connection
MN	Maximum Number	→	M to M Connection
M	Manager		

Table 1: MS-RTCP General View Symbols

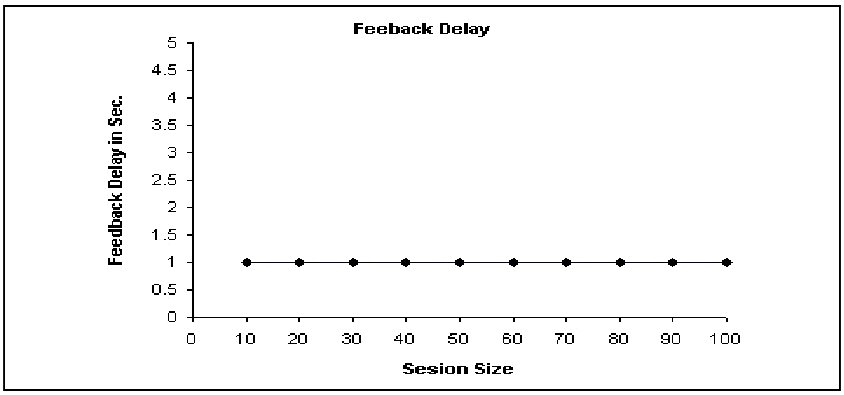

Figure 4: Feedback delay between consecutive RRs

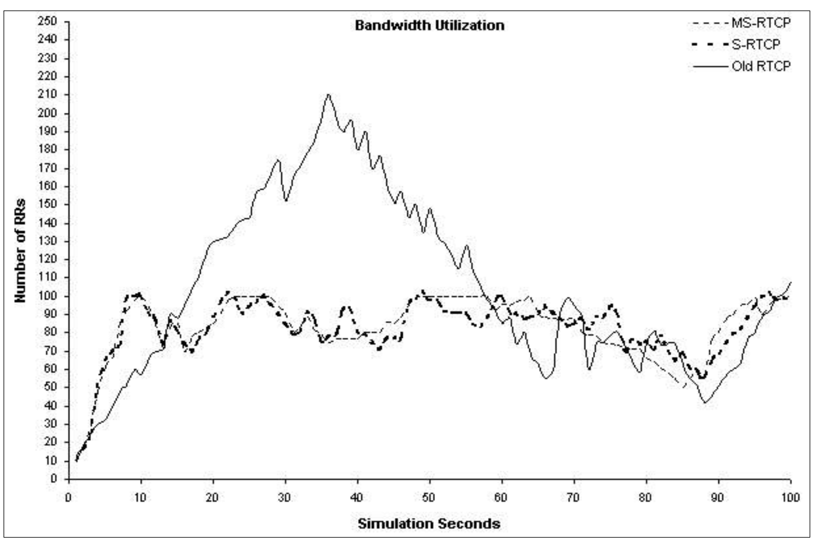

Figure 5: Bandwidth Utilization Comparison

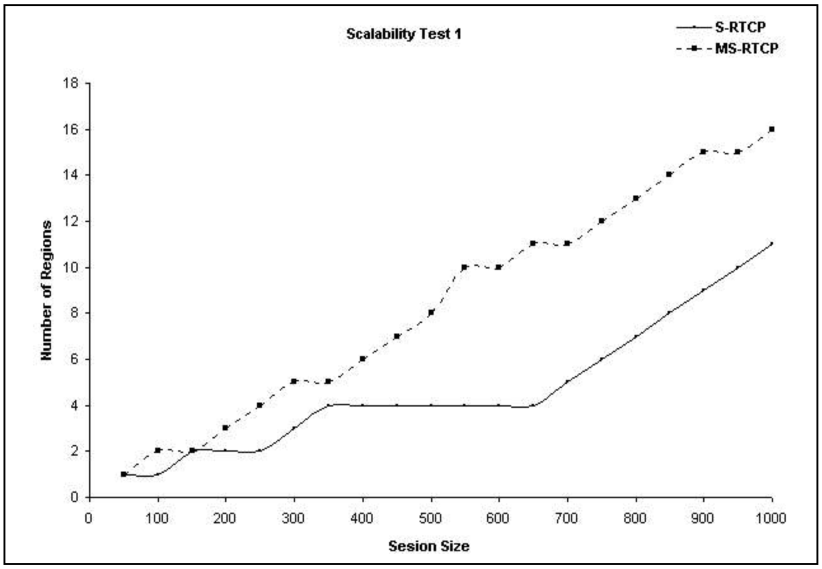

Figure 6: Scalability, children distribution among regions

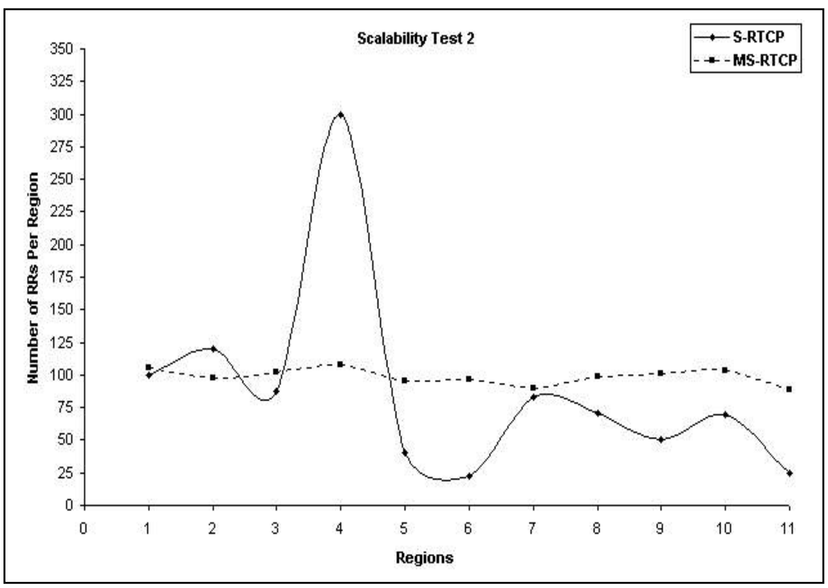

Figure 7: Scalability, average number of RRs per region