Journal of Universal Computer Science, vol. 11, no. 6 (2005), 874-897
submitted: 17/4/04, accepted: 1/2/05, appeared: 28/6/05 © J.UCS

Analysis, Design, and Performance Evaluation of MS-RTCP: More Scalable Scheme for the Real-Time Control Protocol

N. A. Elramly, A. S. Habib, O. S. Essa
(Computer Science Dep., Faculty of Science, Menoufia University, Egypt
omarsaid_essa2001@yahoo.com)

H. M. Harb
(Computer Science Dep., Faculty of Science, Emirates University, UAE
h.harb@uaeu.ac.ae)

Abstract: Recently, some problems related to the use of the Real-time Control Protocol (RTCP) in very large dynamic groups 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 recent one, which was introduced by EL-Marakby and was named Scalable RTCP (S-RTCP), still has several drawbacks. This paper will evaluate the previous model by introducing all its drawbacks. Consequently, we will demonstrate a design of More Scalable RTCP (MS-RTCP) scheme based on hierarchical structure, distributed management, and EL-Marakby scheme. Also, we will show how our scheme will alleviate all the drawbacks found in the S-RTCP. Finally, we will introduce our scheme implementation to analyze and evaluate its performance.

Keywords: RTCP Scalability, Distributed Management, Multimedia Protocols, Multimedia Communication
Category: C.2.2, C.2.6

1 Introduction

Today, the Real-time Transport Protocol (RTP) is widely deployed in most Multicast Backbone (MBone) applications over the Internet involving multiple senders and receivers. The RTCP, which is RTP's control protocol, is used mainly in adaptive applications where the sender changes its rate of data transmission in order to suit current state of the network [1].

Some problems have arisen when RTCP has been used in very large dynamic multicast groups [2]. Firstly, because the RTCP reporting interval grows linearly with the group size, a feedback delay occurs. Consequently, infrequent feedback reports are sent and timely reporting does not occur. Second, each member has to keep track of every other members in the group, thus a storage scalability problem may appear [3]. Third, a flood of initial RTCP reports multicast to the whole group can occur when large number of members joins at the same time. As a result, members can be flooded with these reports, especially the ones connected to the network through low bandwidth links, and the network may be congested [4]. This problem occurs if the reporting members are not implementing the reconsideration algorithms that are described in [2], [3]. Fourth, for receivers connected through low bandwidth links, the RTCP bandwidth available could be used more effectively than is presently the case.

More models have been appeared to alleviate the RTCP problems [2], [3]. The recent one is proposed by El_Marakby [5]. This model is based on hierarchical scheme, whose group members are organized in local regions. Members in each region send their Receiver Reports (RRs[1]) [1] locally to an Aggregator (AG[2]) [5] in the same region. The AG summarizes important information in the RRs and derives some statistics, then sends its information to the manager. The manager does some monitoring and diagnosis functions to estimate which regions are highly suffering form congestion and to evaluate the quality of the transmitted data [5].

This paper proceeds as follows: In Section 2, some background information about El_Marakby scheme and its evaluation is presented. In Section 3, we describe the MS-RTCP. Section 4 presents how our scheme solves El_Marakby scheme problems. In section 5, analysis and performance of the MS-RTCP are introduced. Finally, we summarize the current status and outline future work.

2 El_Marakby Scheme and its Evaluation

The basic idea of El_Marakby scheme is to develop a tree-based hierarchy of report summarizers. At the lowest level of the tree, session participants send RRs. A node, acting as an AG, collects these reports, and summarizes them. These summaries are then passed up towards the next highest node in the tree, until finally they reach the sender or some other appropriate feedback point. The summarization scheme is most useful when the nodes at one level of a sub-tree see similar network performance. El-Marakby uses network hop counts to a summarizer, measured through Time To Live (TTL[3]), to group hosts together in the tree. She also proposed a dynamic scheme for building the tree [5].

The summarizer approach appears to be quite attractive; the hierarchical nature of the system allows for scalability, less information would need to be maintained at any given node, and convergence times would be reduced. However, the scheme has a number of drawbacks for Internet Protocol (IP) telephony and management basics:

1- Fault tolerance is not guaranteed. When any AG (or any component) in the model is crashed or has left the RTP session, all the children[4] in its region search for other AG. This will make an old child to a new one (with its problems). Also, this will affect the convergence time during this interval.

2- Load balancing is not guaranteed. The load balancing depends on the maximum number of children with which the AGs deals. This is not sufficient, because if we suppose that the maximum number of each AG is 100 children, we may find AG has 90 children and another has 10 children.

1- **RR** is a report for reception statistics from participants that are not active senders.
2- **AG** is an expression used in the S-RTCP, and acts as a manager. It summarizes the reports sent by its children.
3- **TTL** specifies how long, in seconds, the packet is allowed to remain in the Internet system.
4- **Children** are the session participants.

3- The structure of the model depends on the central processing unit (manager). Hence, if the manager is crashed there is no other unit that can take place. The model in this case will become unstable (failed in worst case).
4- Overkill the small groups, which the IP telephone is mainly dealing with. This is due to the condition of low children number per AG that was put after the model evaluation [6].
5- Didn't explain a technique or a protocol to handle the multiple sender case.
6- Didn't define the new Aggregation Report (AGR[5]) [5] which was sent by AGs.
7- Didn't determine a protocol to explain how all AGs will connect to the manager to send their AGRs?
8- Election of LAN Aggregator (LAG[6]) is not sufficient.
9- Non-persistent data which is collected by the AGs and may be stored by the manager.
10- Source Description (SDES[7]) items [1] about any multicast group member will take a long time to access it.
11- The load on the AG is calculated with children of the same parent, or can scale with the number of end nodes in the sub-tree rooted in the summarizer (this is not clear).

3 MS-RTCP

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 data which may be aggregated from any small group manager.
4- Flat tolerance in the system.
5- Determining of the tree levels and branches (scalability is wide or deep).
6- All the comments mentioned above in the S-RTCP.

To satisfy these features, we should build 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 (for the details of each MS-RTCP component functions, see Appendix A.1).

We will describe the MS-RTCP by demonstrating the following subsections:
1- How does the scheme work?
2- How is the multiple sender case handled?

5- **AGR** is an expression used in the S-RTCP. It indicates the summarized message sent to the system administrator by the AG.
6- **LAG** is a component that acts as a manager in relation to local area network.
7- **SDES** is a source description items that contains some personal data about RTCP packet originator.

3.1 How does the MS-RTCP work?

When starting the RTP session, two multicast addresses are announced; the first address is for the delivery of RTP data packets, while the second one is for transporting RTCP control packets. Then, the MS-RTCP managers join the control multicast group, while the MS-RTCP children join the data multicast group. Each group of children constructs a region which is controlled by a manager. Each child should know its region before sharing the RTP session. The Load Balancing Manager (LBM) accomplishes this target by testing the real position of each child and the real number of children per region. In MS-RTCP, each shared child should possess two addresses; the IP address referring to the child Internet position, and the tree address referring to the position of the child in the MS-RTCP. After the child finds its region, the Address Resolution Manager (ARM) will take the child IP address and will provide this child with its tree address. At any time, if the child is turned to a manager, the Fault Tolerance Manager (FTM) should find a spare component for this new manager. The scheme has to collect the personal data about any one sharing the RTP session. So, our scheme should contain a manager to store this personal data in its database. The RTP session participants personal data is collected by SDES Manager (SDES-M).

After each child is settled in its region, it will send RRs to its region manager. Thereafter, the manager collects the different RRs and summarizes them using any analytic function(s). Hence; the manager can extract a report and send it to the General Manager (GM) (or the up-level managers in the MS-RTCP). The GM collects all the managers' reports and evaluates the RTP session performance. For data persistency and fault tolerance considerations, all stored data should be saved in a spare copy. The Collection Database Manager (CDM) is responsible for connecting any scheme managers and taking a copy of the stored data (For more details, see Appendices A.2 and A.4. For the communication form, see Fig. A.5).

3.2 Handling of the multiple sender case

The multiple sender case is described as follows. More than one sender can send its Sender Report (SR[8]) [1] simultaneously to one child or more in the RTP session. To handle the multiple sender case, we will demonstrate 4 algorithms, each algorithm responsible for one sub-process. These algorithms are running sequentially because each one works after the result of the previous algorithm. The result extracted from the last algorithm could be considered as a packet that will be sent from any manager to another up-level manager or GM in the management tree. The basic idea of multiple sender algorithms is built on how to analyze and filter the data for each stand-alone sender. Consequently, this sender data will be parsed by some management functions. Accordingly, the manager can extract the locations of the problems and send a report to the GM.

The first algorithm is responsible for collecting the Synchronization Source (SSRC[9]) identifier [1] of all senders in the RTP session. Also, the duplicated SSRCs are eliminated by using the first algorithm. So, we can extract and distinguish the senders from other children.

8- **SR** is a report for transmission and reception statistics from participants that are active senders.
9- **SSRC** is a field used in the RTCP different packets' header to identify the packet originator.

The second algorithm uses the extracted SSRCs to collect the different reports. These reports contain all the data that are important for each manager to evaluate its region senders.

It is obvious that the result of the second algorithm is in form of different reports from different senders. Till now, the overview of the collected reports is not suitable for the manager to evaluate each sender individually. So, the data of each sender should be filtered such that the manager can view and summarize all the individual sender reports. The third algorithm can separate the reports of each sender using the extracted sender's SSRC [7].

Now the data becomes ready for the manager to analyze it. The forth algorithm will provide the manager with some diagnostic functions (like Average, Median, and Standard Deviation (SD)) to help him with the analysis operations (for the algorithms' steps, see Appendix A.3).

4 How did MS-RTCP Solve the El_Marakby Scheme Drawbacks?

1- Fault tolerance guarantee: any control component in the MS-RTCP has a spare one. If one or more scheme component fail, we can replace the failed one with its spare one by decision from the GM (or FTM) until the failed one is fixed and the normal situation is re-established.

2- Load balancing guarantee: the decision for receiving a new child in the RTP session depends on the real number of children per scheme manager. Consequently, if any new child joins a session, it is told with the best manager taking in consideration the manager load (real children number) and the new child position.

3- The basic idea of the MS-RTCP is based on the management distribution. So, the central management processing is eliminated, as we have one manager for each management process.

4- The maximum number of children per manager reaches the upper limit at which the RTP session is working safely (some hundreds or more). Hence, if a small group joins the RTP session, the MS-RTCP will be transformed to the simple RTCP view with one manager and its spare. The other MS-RTCP managers will be found, but with minimal overhead (neglected values).

5- The structure of our scheme messages is introduced (see Appendix A.4).

6- The algorithms, which handled the multiple sender case, are demonstrated.

7- There is no direct relation between the GM and any child, so we are not in need of a protocol to handle the GM to the children relation.

8- The LAN Manager (LAN-M) has its pre-determined spare component. So, in our scheme, we don't need to elect another LAN-M when the basic one fails.

9- Our scheme introduced a spare copy for any management data which is controlled by CDM. Hence, the management database can be considered as persistent database.

10- The GM can access any data about any MS-RTCP entity by requesting the SDES-M (or by CDM in case of any problem happened for the SDES-M).

11- We determined the maximum tree depth which our scheme can reach (equal 3). So, the scalability of the scheme became wide (not deep). This gave us more than one advantage such as alleviating the feedback delay to the GM and decreasing the size of the summery message.

12- We calculated the manager load by the real number of children plus its sub-managers (the managers which were found in its region in the next levels of the tree). This method makes our calculations more accurate.

Remark: GM determines if a new child is a manager or ordinary child using the same technique of El_Marakby scheme [5].

5 Implementation and Simulation Results for MS-RTCP

5.1 Simulation setup

Thereafter, we will implement the system to demonstrate how the MS-RTCP alleviated the three RTCP problems [2], [4] similar to the S-RTCP scheme. Unlike the S-RTCP, the MS-RTCP can adapt the system management and guarantee the fault tolerance, scalability, and load balancing between the MS-RTCP components. The implementation includes also the effects of our scheme uploaded components (FTM, LBM,…) on performance and scalability of the system.

In our analysis we will use a network simulator called NS2 [8] that works with the following characteristics:

1- The RR size equals 60 byte.
2- Session bandwidth equals 250 Kbps.
3- Delay time between each subsequent user RR equals one second.
4- The simulation is done over the session seize equals (10, 20, 30, 40, 50,………….) as a start.
5- The simulation time equals 100 seconds.
6- The session size is dynamic.
7- The simulation contains the multiple sender case.
8- The tree level equals three.
9- The timer and user number (inputs and outputs) are generated by a random number generator function.
10- The packet formation time is neglected. It is supposed that the packet is formed during the delay time (one second).

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

5.2 Storage state evaluation

If any manager in the scheme is overloaded with data (exceeding its capacity), new parameters (time of search and data persistency) will develop and should be taken in consideration. So, the storage state should be tested for each control component in the MS-RTCP.

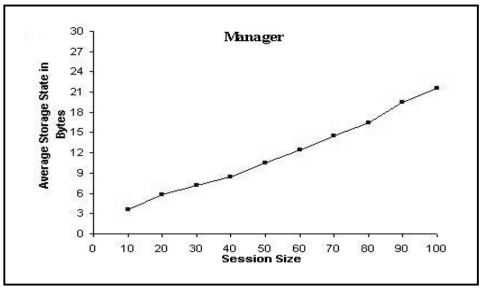

Figure 1: Average storage state of the Manager.

In the MS-RTCP, it is obvious that each manager stores all its own children SSRC only. Consequently, the number of stored bytes depends on the SSRC unit capacity. Each manager is responsible for approximately the same number of children. Hence, the increase in the session size is accompanied by an increase in the manager storage area. In Fig. 1, the simple oscillation found in the curve resulted from the slight variability in the number of children in each region. This curve also demonstrates the efficiency of LBM to keep a balanced managers' load.

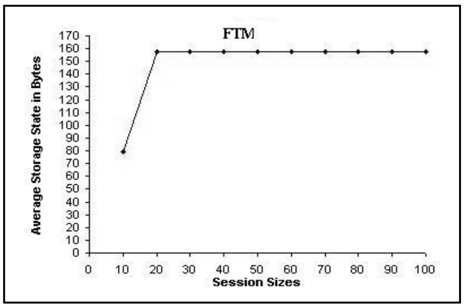

Figure 2: Average storage state of the FTM.

The session containing approximately 10 children is considered a simple session. The simple session contains only one manager. Hence, the FTM stores a little data about this manager and its spare one. When the session accepts more children, the regions will be constructed and the number of managers will increase. Consequently, the FTM will store more data to define all the session managers and their spares. Therefore, the stability in the curve, which is found in Fig. 2, results from the fixed number of the managers when the session size is between 20 and 100 children.

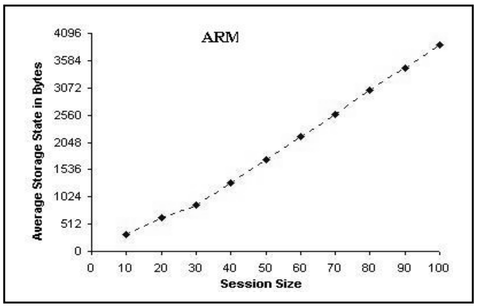

Figure 3: Average storage state of the ARM.

Fig. 3 explains the ARM storage state in bytes. The ARM storage area is increased following an increase in the session size. The subsequent increase in the ARM storage area is justified by the fact that the ARM is responsible for storing the IP and tree addresses for all the MS-RTCP components. We can notice a minimal oscillation in the curve when the session size is between 10 and 30 children. The minimal oscillation occurs since we have supposed that the session size is dynamic, i.e. more children are joining or leaving the session suddenly during a small time interval. When the session size exceeds 30 children, the sudden change in the session size doesn't occur, and the curve becomes smooth (the oscillation disappears).

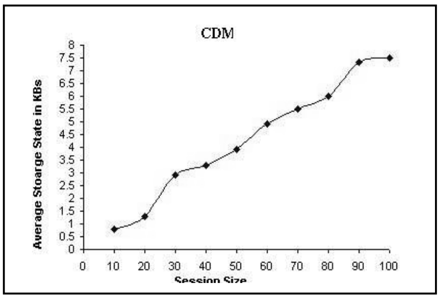

Figure 4: Average storage state of the CDM.

The CDM storage area is increased following an increase in the session size. The increase in the CDM storage area is concerned with storing the CDM for the IP and tree addresses about each MS-RTCP component. In Fig. 4, the notable oscillations are due to rapid joining and leaving the RTP session by a group of children, i.e. a group of children stays in the session for a short time interval without any active sharing.

The LBM stores the maximum and real numbers of each region together with the tree address of each manager in the session. Therefore, the storage area is approximately constant while the session size is increasing. In Fig. 5, the oscillations found in the curve are due to gradual creation of regions. While a new region is being created, a little data is added to the LBM database.

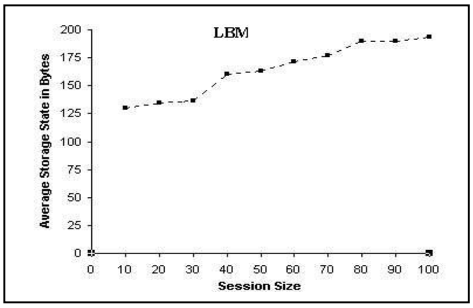

Figure 5: Average storage state of the LBM.

It was formerly documented that the main function of the SDES-M is to store the personal data about each child sharing the RTP session. So, if the number of children increases, the SDES-M stored data will increase consequently. In Fig. 6, the oscillations found in the curve results from the sudden changes in the session size.

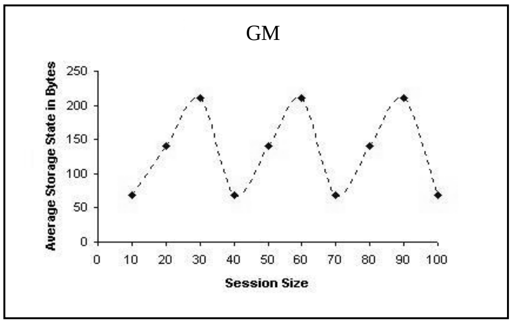

Figure 6: Average Storage State of the SDES-M.

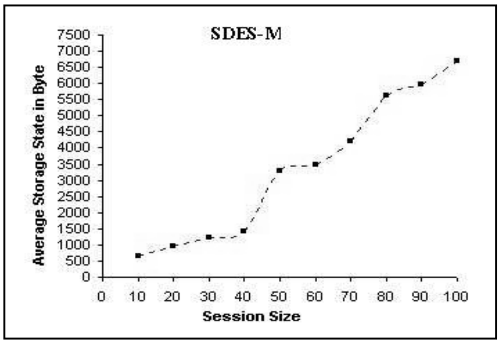

Figure 7 : Average storage state of the GM.

The GM stores the processed data for a period of time to compare such data with another data collected during the subsequent periods. The time taken by the GM to keep the summarized data depends on two parameters: 1) The GM properties which determine its storage capacity. 2) The session size which determines the quantity of data that will be summarized and sent to the GM. In our simulation, the GM can store the evaluated data that coming from under-level managers for 30 seconds, then delete it. The deletion of the data is followed by instantaneous resorting of another evaluation data. The process of storage, deletion, and restoring are occurred consequently. These consequent operations cause the big oscillations in the curve presented in Fig. 7.

5.3 Feedback delay evaluation

Fig. 8 shows the average feedback delay, i.e. the average time interval between two consecutive feedback reports. It is obvious that the feedback delay is constant (one second) in the MS-RTCP. As the children are divided into regions, the feedback delay is calculated depending on the region size (approximately equal at all regions) and regardless the session size.

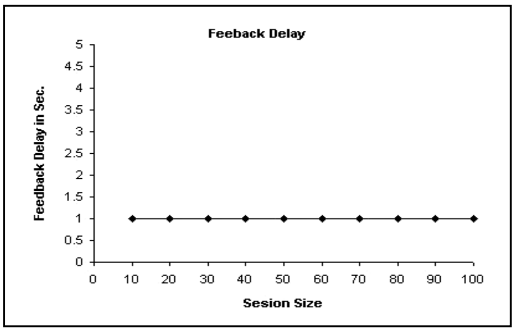

Figure 8 : Feedback delay between two consecutive RRs for each user.

5.4 Bandwidth utilization evaluation

To analyze our scheme in relation to the bandwidth utilization, we should determine the average number of feedback RRs which are sent during the simulation time. If the number of RRs increases with notable values during any time interval, this may lead to decrease the bandwidth for the multimedia data. Consequently, the congestion problem probably occurs. In Fig. 9, it is obvious that the number of RRs sent in the MS-RTCP and the S-RTP is approximately equal. Also, it is notable that the number of RRs sent in the old RTCP is large between 25s and 55s. In the old RTCP scheme, large number of RRs occurs because the whole RTP session can be considered as one region.

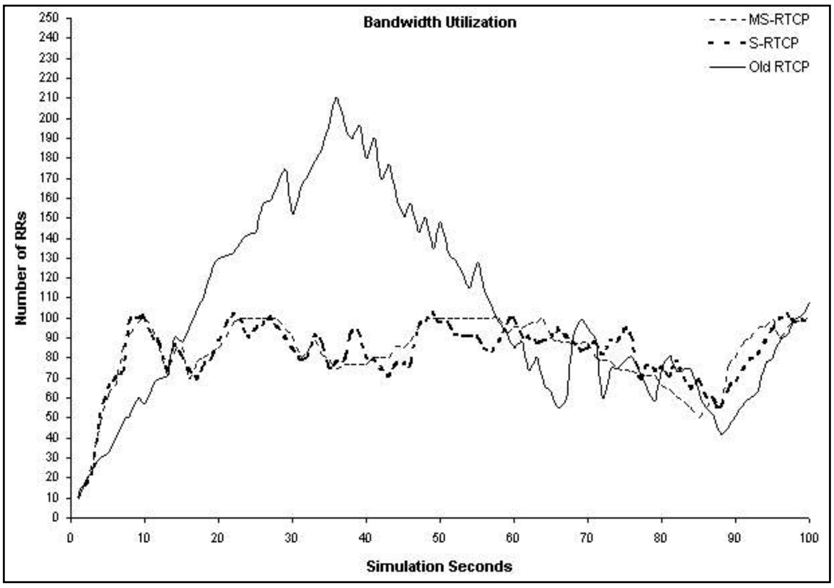

Figure 9 : Bandwidth utilization in MS-RTCP, S-RTCP, and RTCP.

Also, in the MS-RTCP curve, it is obvious that the number of RRs during the time interval 80s-90s decreases significantly. The significant decrease in the RRs is justified by the event that a large number of old children left the session and new children were gradually joining to the session.

5.5 Comparison of the S-RTCP and the MS-RTCP in relation to scalability

In this subsection, we will study the scalability of the MS-RTCP and the S-RTCP. To accomplish this scalability study, a session size should be increased up to 1000 children. This number of children makes our test more accurate. To demonstrate the attitude of the MS-RTCP and the S-RTCP while the session size is increasing both gradually and suddenly, two tests must be performed:

1-Scalability test 1: this test scales how the children are distributed among the regions?

2-Scalability test 2: this test demonstrates the average number of RRs in each region.

The plot, which is illustrated in Fig. 10, shows the comparison between the S-RTCP and the MS-RTCP as regards to the children distribution among the regions. In the S-RTCP curve, at the session size between 350 and 650 children, it is obvious that the curve is stable. The curve stability means that any new child which joins the session will settle in the old four regions. Also, we can notice that about 7 regions are created when the session size is between 650 and 1000. This indicates that the distribution of children in the regions is abnormal, and a notable diversion will be found in the managers' load. Unlike S-RTCP, the MS-RTCP has a normal curve with minimal oscillations. When the MS-RTCP session size increases (gradually or suddenly), new regions are created. This curve also demonstrates the efficiency of the

LBM in children distribution over the regions. We can also observe that the number of the regions in the MS-RTCP is more than in the S-RTCP. This is because the LBM tests the real number of children per region every time. When the LBM finds a notable increase in a region size, it stops the children entry in this region, and creates a new one.

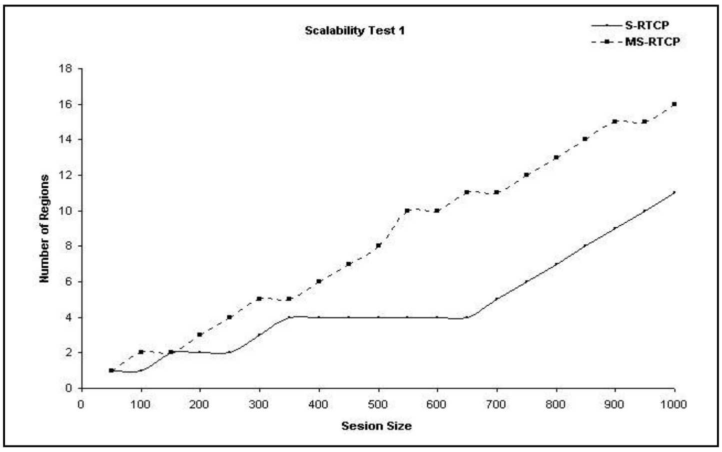

Figure 10: Scalability test 1 (distribution of children among the

The plot found in Fig. 11 determines the number of RRs that are sent to each S-RTCP region and MS-RTCP region. In the S-RTCP curve, we can notice that the number of RRs sent to the region number 4 is large. This results from the abnormal distribution of the children which makes region number 4 overloaded. An important notice must be taken in our consideration viz.: if there is a notable increase in the session size (e.g. thousands), the S-RTCP system administrator has only two possibility to deal with this situation. The first possibility is to reject the joining of extra children when the region size reaches its maximum, and this is against the scalability definition. The second possibility is to increase the maximum size that each region can load. This second possibility may lead us to the first case of the RTCP (huge number of children with one manager) with its three problems. In the MS-RTCP curve, it is observed that the number of RRs in each region is approximated due to the LBM efficiency.

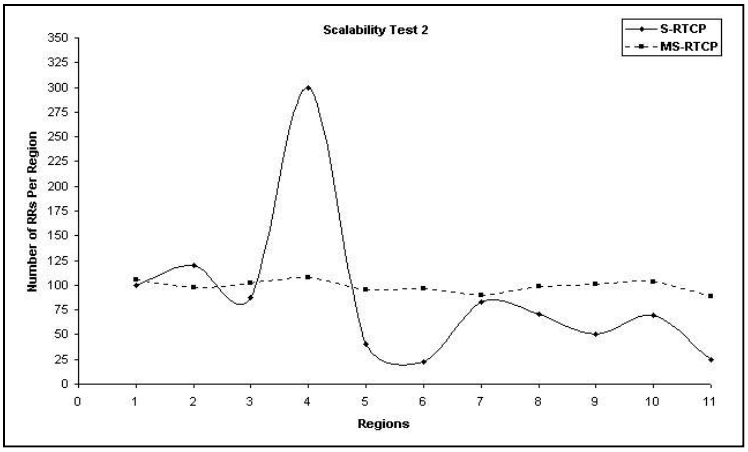

Figure 11: Scalability test 2 (average number of RRs in each region).

6 Conclusion

We have presented the problems encountered in the deployment of RTCP in large dynamic groups. Also, we have introduced a brief discussion of the previous RTCP scalability scheme (El_Marakby) and its drawbacks. 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 a 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. So, we have a manager for each essential management function. Also, we have introduced how our scheme overcame all the previous scheme drawbacks to be more scalable. Finally, we have evaluated the MS-RTCP performance in relation to all RTCP problems and system scalability by using a network simulator called NS2.

7 Future Work

The next phase of our work is divided into two approaches:

1- We will construct MS-RTCP real system, and monitor its connection efficiency for a long period (one year or more). So, we are in need of collecting some data about the MS-RTCP connections and keeping this data in a database. Thereafter, we will analyze the stored data using any data mining algorithm. We will predict the connections that repeatedly failed, i.e. the connections that use their spares permanently. Consequently, we will keep the spare connections of the failed ones and eliminate the spare connections of the others. Therefore, the number of connections will be decreased. Accordingly, the overhead communication will also be decreased.

2- We will try to construct an intelligent agent for each management task in the MS-RTCP. Also, we will try to communicate these agents with synchronized events. Finally, we have to scale the new agent system performance, and compare MS-RTCP with the new agent system.

Acknowledgements

We would like to express our deep thanks to the people who have helped during the research preparation. Also, we greatly indebted and grateful to the Professor Mohamed M. El-kafrawy, for his efforts, advice, and continuous encouragement.

References

[1] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," *IETF Internet Draft*, March 2, 2003.

[2] J. Rosenberg and H. Schulzrinne, "Timer reconsideration for enhanced RTP scalability," *Proceeding of the Conference on Computer Communication (IEEE Infocom)*, San Francisco, California, March/April. 1997.

[3] J. Rosenberg and H. Schulzrinne, "Sampling of the group membership in RTP," *IETF Request for Comments* 2762, Feb. 2000.

[4] B. Aboba, "Alternative for enhancing RTP scalability," *IETF Internet Draft*, Jan. 1997.

[5] R. El-Marakby and D. Hutchison, "A scalability scheme for the real-time control protocol," *Proceeding of IFIP Conference on High Performance Networking (HPN'98)*, Vienna, Austria, Sept. 1998.

[6] R. El-Marakby and D. Hutchison, "Design and performance of a scalable real-time control protocol simulations and evaluations," *Proceeding of IFIP Conference on High Performance Networking (HPN'98)*, Vienna, Austria, Sept. 2000.

[7] Y. Langsam, J. Augenstein, M. Tenenbaum, J. Mushe " Data Structures Using C and C++ (2nd Edition)" *Prentice Hall, Upper Saddle River NJ, USA*. 1996.

[8] S. McCanne and Floyd, (McCanne, 1998) "LBNL Network Simulator," 2003 version, 2003.

List of Abbreviations

AG	Aggregator.
AGR	Aggregator Report.
ARM	Address Resolution Manager.
CDM	Collection Database Manager.
FTM	Fault Tolerance Manager.
GM	General Manager
IP	Internet Protocol.
LAG	LAN Aggregator.
LAN	Local Area Network.
LAN-M	LAN Manager.
LBM	Load Balancing Manager.
MBone	Multicast Bone.
MS-RTCP	More Scalable Real-time Control Protocol.
RR	Receiver Report.
RTCP	Real-time Control Protocol.
RTP	Real-time Transport Protocol.
SD	Standard Deviation.
SDES	Source Description.
SDES-M	SDES Manager.
SR	Sender Report.
S-RTCP	Scalable Real-time Control Protocol.
SSRC	Synchronization Source.
TTL	Time To Live.

Appendix A
Details of the Protocol from the Implementation Prospective

A.1 MS-RTCP components' functions

A.1.1 Manager (Tree address depends on its location)

1- Determining the number of children with packet loss exceeding the maximum threshold and those with packet loss laying between maximum and minimum thresholds.
2- Determining the child which has the worst quality (i.e. has the highest packet loss).
3- Determining the number of children which sent the RRs.

It collects and parses the information received from the children (RRs), then performs the following functions:

1- Obtaining the number of children with packet loss exceeding the maximum threshold as regards the total number of children in the region. Also, the percentage of the children suffering form the packet loss is derived.
2- Calculating the median, the average, and the SD.
3- Storing the name of each starved RR sender.
4- If some applications still insist on using sender-based adaptive schemes, then sender can be adapted. The region manager sends the packet loss fraction of the receiver suffering from the highest packet loss incurred in the current interval to the sender. Then, the sender may decrease its rate of the data transmission (if necessary).

Remark: If the region manager cannot reach the sender, it sends a message to the direct up-level managers and so on, till it has reached the sender.

A.1.2 FTM (Tree address = 1)

This manager is responsible for defining a spare component for each scheme component. So if any trouble has occurred in any component, the GM consults the FTM to extract the spare component of the failed one. The fields which are used by FTM are: 1) Basic manager address: the address of each scheme manager in the normal state (no problem state). 2) Spare manager address: the address of the manager which should substitute the failed manager (each component in the MS-RTCP have a spare one except the session children. We explained the manager as an example).

A.1.3 LBM (Tree address = 2)

This manager is responsible for load adjustment between managers in the MS-RTCP by storing a real number of children per region. When a new child tries to join the RTP session, the LBM takes into consideration not only the maximum number of children but also the real children number. The fields which are used by LBM are: 1) Basic manager address: the same address stored in the FTM. 2) The maximum number of children: the number of children that the manager can load without exceeding this number. 3) The real number of children: the number of children which the manager has loaded at each interval.

A.1.4 ARM (Tree address = 3)

This manager is responsible for storing the tree address (the address which indicates the component real location in the management tree) of any MS-RTCP component. This tree address corresponds to the IP address of the component. The fields which are used by ARM are: 1) Component IP address. 2) Tree address.

A.1.5 SDES-M (Tree address = 4)

This manager is responsible for defining all the data about the RTP session participants like name, address, geographical location, e-mail,….etc.

A.1.6 CDM (Tree address = 5)

This manager is responsible for storing all the management data which is stored by all the scheme managers (can be considered as a data spare copy).

A.1.7 Children (Tree address depends on its location)

Children are RTP session participants that send SRs or RRs.

A.1.8 LAN-M

LAN-M executes the same functions and collects the same data as a pre-explained manager but from a Local Area Network (LAN).

A.1.9 GM (Tree address =0)

1- Collecting the messages that are sent from the first under-level managers.
2- Parsing the collected data to determine the performance of the whole RTP session.
3- Determining the relation between senders and receivers suffering from huge packet loss.
4- Connecting to all other distributed managers (FTM, LBM,..) to solve any problem that may be occurred during the RTP session like crashed or left manager.
5- Storing the collected data and extracted results in the previous interval to make a comparison between the current network performance and the previous one. This is useful for determining if the network performance has increased or decreased.

Remark: The general view of our scheme that describes the communications of the scheme components is shown in Fig. A.5.

A.2 Details of the MS-RTCP work idea

In this subsection, we will describe how the model works by demonstrating the following:
1- How can a new child join the RTP session?
2- How can an old child can leave the RTP session?
3- The relations between the scheme entities.

A.2.1 How can a new child join the RTP session?
When a new child connects to the RTP session, the following steps are executed:

1- A new child sends a message to the GM asking him to join the RTP session.
2- The GM sends a message to the LBM asking about the best place for this child.
3- The LBM sends a reply message with the best place taking into consideration both the child's IP address and the load balancing between the scheme managers.
4- The GM replies to the new child with its tree address. The tree address contains the name of the child and its up-level managers (ex. 0.1.2.10).
5- The GM sends a multicast message to the FTM, LBM, ARM, and CDM informing them about the new child.

A.2.2 How can an old child leave the RTP session?
When an old child leaves the RTP session, the following steps are executed:

1- A child should send a BY[10] message [1] indirectly to the GM.
2- Consequently, the GM multicasts the BY message to all the distributed MS-RTCP managers (FTM, LBM, ARM, SDES-M, and CDM) to delete the old child from their database.
3- If the leaving child was basic or spare manager, the FTM will recover this status.

A.2.3 The basic relations between the GM and other model entities
1- GM and FTM: this relation is firing when any control component in the model such as LAN-M is crashed or left the RTP session.
2- GM and LBM: this relation is firing when any new or old scheme component is joining or leaving the RTP session respectively.
3- GM and ARM: this relation is firing when a new component is sharing the RTP session.
4- GM and SDES-M: this relation is firing when the manager is in need of any data like name or e-mail about any RTP session participant.
5- GM and CDM: this relation is firing when any control manager (FTM, LBM, ARM, SDES-M, and CDM) failed. By using this relation the data became more persistent.
6- GM and Manager: this relation is firing when the GM needs to evaluate the whole network (RTP session) performance during some interval (the interval in which children send their reports).

10- **BY** is RTCP packet indicates end of participation.

7- GM and LAN-M: this relation likes the relation number 6 but in relation to the LAN.
8- GM and Children: no direct relation in a large RTP session.

A.2.4 The basic relations between the Manager (or LAN-M) and other model entities
1- Manager and its children: the relation is firing during each interval in which children can send their RRs.
2- Manager and the CDM: this relation is firing when the GM is replaced with its spare manager (in case of GM failed).

A.3 Multiple sender case algorithms

In the next subsections, we will demonstrate a prototype for the variables which should be used in our algorithms. Thereafter, we will introduce our algorithms according to the processes arrangement.

A.3.1 Prototype
We define some dynamic arrays as follows**:**

1- **N_RR** is the number of RR
2- **M_SSRC** is the number of blocks per RR
3- **H_Sender** is the number of senders.
4- **B[M_SSRC]** is an array of structures, where each structure in this array represents a part from RR for one sender [Report Block].
5- **C_H[N]** is an array containing the names of children which sending the RRs in the MS-RTCP.

Remark*:* N_RR and M_SSRC can be counted by any counter variable in the manager code.

A.3.2 Algorithms
Algorithm 1 extracts all the SSRCs for the senders in the RTP session. This can be accomplished by accurate scanning for each sender report using *For loop*. Consequently, the algorithm keeps the extracted SSRCs in stand-alone array. The array resulted from algorithm 1 is called SSRC_Array. To extract the individual source identification for each sender in the RTP session, algorithm 1 will delete the duplicated SSRCs that may be found.

A. <u>Algorithm 1 (SSRCs Extraction Algorithm)</u>

1- For I= 1 to N_RR
    1-1 Begin
    1-2 C_H[I]= C_Name[I].SSRC_Packet_Sender
    1-3 For J= 1 to M_SSRC
        1-3.1 Begin
        1-3-2 Push (SSRC) [To create the SSRC array (the sources' Identifications in each RR without duplication)]
        1-3.3 End J For Loop.
    1-4 Searching for the duplicated SSRCs.

1-4.1 If found duplicated SSRC delete it.
1-4.2 Else complete the algorithm
1-5 End I For Loop
2- End Algorithm.

By using the SSRCs resulted from algorithm 1, we can scan the RRs and extract the data of the session senders. Algorithm 2 can achieve this target by comparing the SSRC_Array entries with the SSRCs of the reports. If they are equal, the algorithm keeps the data of the scanned reports, and if not, these reports will be neglected.

B. <u>Algorithm 2 (Data Extraction Algorithm)</u>

1- For K= 1 to H_Sender
2- For I= 1 to N_RR
 2-1 Begin
 2-2 Read M_SSRC =N_of_Blocks_Per_Report
 2-3 For J= 1 to M_SSRC
  2-3.1 Begin
  2-3.2 If SSRC_Array [K] = B [J].SSRC Then
   2-3-2.1 Begin
   2-3-2.2 SSRC_Final [K].P_Lost [D] = B[J].Block_Packet_Lost.
   2-3-2.3 SSRC_Final [K].Name [D] = B[J].Name.
   2-3-2.4 End J For Loop
   2-3-2.5 D= D+1
   2-3-2.6 End If
  2-3.3 End J For Loop
 2-4 SSRC_Final [K] .Child_Name [I] = C_H[I]
 2-5 End I For Loop.
3- End Algorithm.

Now, after running algorithm 2, we have an array of structures called SSRC_Finally, each structure in this array represents a total report for a specific sender in the manager region. Consequently, we can extract any summarized data for any sender by running the two algorithms above. Each manager in the system can accept more than one array as SSRC_Final that is taken from the under-level managers or from the children in its region. So, we should filter the data for each sender, but till now, we have different arrays' structures. Algorithm 3 will be introduced to accomplish this process.

C. <u>Algorithm 3 (Separation Algorithm)</u>

1- Merging all the reports which come from the under-level managers (on array form) in one report [7].
2- Arranging the resulting array depending on the SSRC of each sender.
3- Merging all array members which have the same SSRC (we view the array member as a structure).
4- Storing the result in a new array called **MLA**.

After each sender data which is found in MLA has been filtered, we can parse it and extract the problems by applying some diagnostic functions. Running algorithm 4 can perform this target.

D. <u>Algorithm 4 (Diagnostic Algorithm)</u>

1- Calculating the number of children that packet loss exceeds the maximum threshold.
2- Storing names of the children which have the maximum packet loss (worst children).
3- Calculating the average, the median, and the SD for its data.
4- Extracting other information, Fig A.1.

A.4 Scheme messages (types and structure)

Some of new messages are added to our new scheme for connecting the scheme components with each other. Scheme messages should be defined according to both type (who send or receive it) and structure.

A.4.1 Types of the MS-RTCP Messages.

1- Message from Manager to Manager (or GM) in the RTP session
2- Fault tolerance message from GM to FTM. (Request and Reply).
3- Load balancing message from GM to LBM (Request and Reply).
4- Address resolution message.

A.4.2 Structure of the MS-RTCP Messages.

1- Message from Manager to Manager (or GM) in the RTP Session (Message Type = 0)

All fields of this message are previously stated above. The output of running above four algorithms is the Manager-to-Manager message, Fig. A.1.

Message Type	0	Up-M Tree Address	Up-M Tree Address	Up-M Tree Address	Manager Tree Address

<table>
<tr><td colspan="3">Sender SSRC_1</td><td rowspan="8">Block 1</td></tr>
<tr><td>Average</td><td>Median</td><td>Standard Deviation</td></tr>
<tr><td>Number of RRs</td><td colspan="2">Number of children exceeds the maximum threshold</td></tr>
<tr><td>Child Tree Address</td><td colspan="2">Packet Loss Number</td></tr>
<tr><td>1.2.3</td><td colspan="2">200</td></tr>
<tr><td>1.2.6</td><td colspan="2">300</td></tr>
<tr><td>.............</td><td colspan="2">.............</td></tr>
<tr><td colspan="3">Other extracted data</td></tr>
<tr><td colspan="3">Sender SSRC_2</td><td rowspan="8">Block 2</td></tr>
<tr><td>Average</td><td>Median</td><td>Standard Deviation</td></tr>
<tr><td>Number of RRs</td><td colspan="2">Number of children exceeds the maximum threshold</td></tr>
<tr><td>Child Tree Address</td><td colspan="2">Packet Loss Number</td></tr>
<tr><td>1.2.3</td><td colspan="2">200</td></tr>
<tr><td>1.2.6</td><td colspan="2">300</td></tr>
<tr><td>.............</td><td colspan="2">.............</td></tr>
<tr><td colspan="3">Other extracted data</td></tr>
<tr><td colspan="3">-
-
-</td><td>-
-
-</td></tr>
</table>

Figure A.1: Manager (or GM)-to-Manager message.

2- Fault tolerance message from GM to FTM. (Request and Reply) (Message Type = 1)

This message supports the relation between the GM and the FTM. When any problem occurs related to any scheme component, this relation between GM and FTM will be fired to solve this problem. This relation is accomplished by two different messages (request and reply). Request message contains four fields, viz. 1) Message type: 16-bit integer number to define the type of the message (for the request message = 11). 2) FTM tree address. 3) Basic manager tree address 4) State: If the manager crashed or left the RTP session. Reply message contains four fields also that can be stated as follows: 1) Message type (for the reply message = 12). 2) GM tree address 3) Basic manager tree address. 4) Spare manager tree address, Fig. A.2.

Request Message:

Message Type	F T M Tree Address	Basic Manager Tree Address	State
11	1	0.1.2	Crashed

Reply Message:

Message Type	G M Tree Address	Basic Manager Tree Address	Spare Manager Tree Address
12	0	0.1.2	0.1.2.1

Figure A.2: GM to FTM message.

3- Load balancing message from GM to LBM (Request and Reply) (Message Type = 2)

This message supports the relation between the GM and the LBM to adjust the number of children for each scheme manager. This relation includes request and reply messages. The request message contains three basic fields 1) Message type (equal 21). 2) LBM tree address. 3) IP address for the new children. The reply message contains four basic fields, 1) Message type (equal 22). 2) GM tree address. 3) Manager tree address. 4) IP addresses for the children which are related to the manager in the previous field, Fig. A.3.

Request Message:

Message Type	L B M Tree Address	Children IP Address
21	2	196.253.255.255, 180.240.255.255

Reply Message:

Message Type	G M Tree Address	Manager Tree Address	Child IP Address
22	0	0.1.2	196.253.255.255
		0.2.1	180.240.255.255
		------	------

Figure A.3: GM to LBM message.

4- Address resolution message (Message Type = 3)

This message supports the relation between the GM and the ARM. It is sent from the GM to the ARM after a new child joins the RTP session informing it about the child tree address. This message contains five fields, 1) Message type. 2) ARM tree address. 3) Child IP address. 4) Child tree address 5) Manager state: determines if the new child is a manager (field value = M) or not, Fig. A.4.

Message Type	A R M Tree Address	Children IP Address	Child Tree Address	Manager State
3	3	196.253.255.255	0.1.2.4	M

Figure A.4: GM to ARM message.

The request and reply messages between GM and SDES-M have the same structure of the FTM messages but with different required fields (name, location,---------).

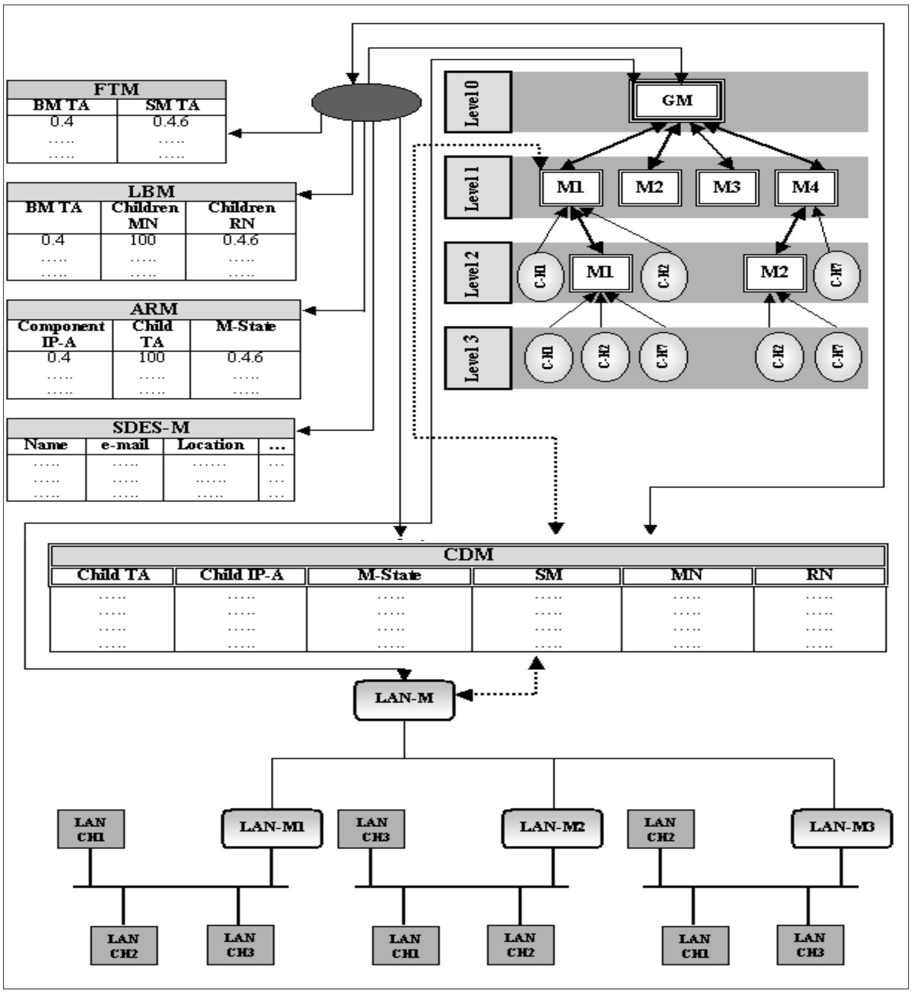

Figure A.5: General view of our scheme.

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 A.1: The symbols that are used in the scheme general view.