Non-Real Time Reliable Multicast Protocol Using Sub Sub-Casting

Ali Alsaih[1] and Tariq Alahdal[2]
[1]Department of Electrical and Computer Engineering, Sana'a University, Yemen
[2]Department of Communication Technology and Networks, University Putra Malaysia, Malaysia

Abstract: *In this paper, a Reliable Multicast transport protocol over combined (fixed/mobile) networks using Sub Sub-casting called (RMSS) is proposed. RMSS is based on a hierarchical structure where receivers are grouped into local regions. In each local region there are special receivers, which are called Designated Receivers (DRs) and Mobile Agents (MAs). Each DR or MA is responsible for retransmission of requested packets to the receivers which belong to its local region. DRs and MAs send their acknowledgments periodically to the sender. They also process the acknowledgments sent to them from the Fixed Hosts (FHs) and Mobile Hosts (MHs) and they retransmit lost packets to the requesting receivers. A sub sub-casting technique is applied within these relatively smaller regions, where the repaired packets are retransmitted only to the requested receivers of the local group. These receivers form a sub group of the local group which itself is a subgroup of the global multicast group. By applying sub sub-casting technique the number of duplicated packets due to retransmission of packets drops to zero which improves the system performance. Simulations (using C++) have demonstrated the scalability of RMSS with sub sub-casting technique.*

Keywords: *Reliable multicast, designated receiver, mobile agent, sub sub-casting.*

Received October 5, 2005; accepted January 17, 2006

1. Introduction

Multicasting is the ability of a communication network to accept a single message from an application and to deliver copies of the message to multiple recipients at different locations. With the emergence of mobile users, many existing Internet protocols, including those with multicast support, need to be adapted in order to offer support to this increasingly growing class of users. Mobile access is currently characterized by lower access speeds, the relatively limited processing power of portable units, higher bit and packet error rates, and user mobility.

Multicast represents an efficient mechanism that implements point-to-multipoint communications. Applications that utilize it fall into two classes, which are soft real-time and fully reliable multicast applications [17]. The first of these handles delay sensitive applications such as video conferencing, service discovery and distance learning while, the latter class includes applications that expect reliable data transmission such as software distribution and access to distributed databases. In this work, a new reliable multicast protocol, specially tailored for wireless access, is presented.

Reliable multicast transport is considerably more complex than reliable unicast. It is generally difficult to build a generic reliable transport protocol for multicast, such as TCP which is a generic transport protocol for unicast. It is a good idea to implement different kinds of protocols to serve different kinds of applications instead of having one protocol to serve all kinds of applications. Doing so will make the protocols simpler to implement and more efficient.

This paper is organized as follows. Section 2 gives an overview of related works, followed by a description of the reliable multicast transport protocol using sub sub-casting RMSS in section 3. The simulation performance analysis is described in section 4, followed by the results and discussions in section 5. Conclusion is presented in section 6.

2. Related Work

A number of approaches have been designed to improve the scalability of reliable multicast protocols. One approach is Negative Acknowledgment (NAK) suppression [1, 15]. Another way to improve scalability is via hierarchy [2, 3, 4, 13, 18]. A tree for the reliable multicast session is made up of ordinary and special receivers which are called the DRs [2, 3, 4, 13, 18]. Commonly used reliable protocols include the Scalable Reliable Multicast (SRM) [10] and the Reliable Multicast Transport Protocol (RMTP) [9, 17, 19]. SRM is based on an application level framework where it is the application's responsibility to guarantee packet sequencing. RMTP defines a hierarchy of DRs, a concept also used in Reliable Mobile Multicast (RM2). RM2 [11] is a reliable multicast protocol that can be used for both wired and wireless environments.

RM2 divides a multicast tree into subtrees where subcasting within these subtrees is applied using a tree of Retransmission Servers (RSs). Furthermore, RM2 guarantees sequential packet delivery with no packet loss to all of its multicast members. RM2 relies on the Internet Group Management Protocol (IGMP) [15] to manage group membership, and on the Internet Engineering Task Force (IETF's) mobile IP to support user mobility through a Care-of-Address (CoA). Each RS has a retransmission subcast address shared by its members and which may be dynamically configured using the IETF's Multicast Address Dynamic Client Allocation Protocol (MADCAP) [13].

Although each DR is responsible for handling error recovery within a region of the multicast tree, it does not say whether this is done in multicast or unicast and therefore it does not concern itself with the control of retransmission overhead. This is a serious drawback when considering emerging mobile environments.

Another solution for implementing multicast in mobile environments was made in [12] and later on revised in [21]. The latter does not however make use of the IETF's mobile IP and offers limited scalability as discussed in [20]. Furthermore, it assumes that mobile groups are static. The global state is maintained by a manager and assumes that network components such as routers, links and hosts are totally reliable. This technique also considers that multicast routers have infinite buffering capabilities to store all multicast messages, hence guarantying end-to-end protocol reliability. This solution only focuses on host mobility merely considering multicast messages addressed to a Mobile Host (MH). In [6] a Reliable Multicast Delivery Protocol (RMDP) for wireless environments is described. It uses Forward Error Correction scheme (FEC) and Automatic Retransmission reQuest messages (ARQ). The main drawback of this approach is the computation of the erasure codes by software which implies an overhead in protocol execution. Furthermore, it suffers from the problem of ARQ explosion due to burst packet loss in wireless environments.

In [5], the multicast message is flooded to all base station over channels that are assumed to be reliable. The base station then collectively ensure that all mobile nodes belonging to the multicast group get the message and the wireless channels between the base station and the mobile node are unreliable. The protocol is fully decentralized where each base station can independently decide when to flush a message from its buffer. The authors stated that the delivery of the message to the mobile host is guaranteed when the mobile host moves from one cell to another while the multicast is in progress.

More work on multicast at the transport level can be acquired from [8] and [14] where the basic idea consists of using mobile IP's Home Agent (HA) functionality to send multicast datagrams to the MHs. The authors introduced the concept of a Designated Multicast Service Provider (DMSP) in order to overcome the problems risen from the use of tunnels, such as the tunnel convergence problem. Within a given Foreign Network (FN), the proposal defines a DMSP responsible for each of the multicast groups. The DMSP sends multicast packets to MHs belonging to its own group and located in its FN. The proposed solution also presents some problems. Firstly, datagrams delivery is always performed through the DMSP, therefore also leading to an inefficient form of routing. In addition, this solution does not guarantee reliability and overlooks performance issues.

Observing that it is better not to push the responsibility of the session state or reliability mechanism to the network layer, the emphasis should be to design a minimal set of network primitives sufficient for upper layers to implement their own transport protocols. The proposed algorithms aim at achieving this. In the following section, the new proposed reliable multicast protocol will be explained.

3. RMSS

RMSS provides sequence, lossless delivery of bulk data from one sender to many receivers. Some of these receivers are FHs and others are MHs. RMSS has been designed to alleviate the ACK-implosion problem by using a tree-based hierarchical approach [3, 4, 13]. The key idea behind hierarchical approach is to group receivers into local regions and to use special receivers called the RSs as representatives of local regions of children receivers. In RMSS, there are two kinds of RSs. The first kind is called the DRs and these works as a representative of local areas of FHs and the second kind is called the MAs and these works as representatives of local areas of MHs. This protocol dedicated for those applications that are not delay sensitive but are loss sensitive.

The sender periodically multicasts a window of new data packets to all receivers in the multicast group (DRs, MAs, FHs, MHs) using the global multicast tree. Only first level DRs and MAs send their own status to the sender indicating which packets they have received correctly or not. The DRs and MAs send their status to the sender in the form of acknowledgment packets at periodic intervals. Also, each FH or MH sends its status to its parent DR or MA, respectively at regular intervals. The DRs or MAs use these status messages to perform local retransmissions to the receivers, thus, reducing the end to end delay significantly. Thus, the sender sees only the DRs and MAs and these see only the FHs and MHs, respectively. Processing of status messages is thus distributed among the sender, DRs and MAs, thereby avoiding the ACK-implosion problem. RMSS consists of two-level hierarchy as shown in Figure 1, the first level consists of DRs and MAs and these are called the RSs. The second level consists of FHs and MHs. Extension to multi-level hierarchy is a straightforward issue.

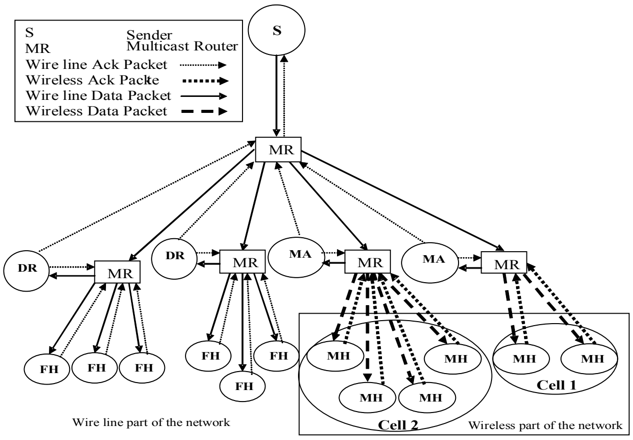


Figure 1. Topology of RMSS.

3.1. Protocol Details

The sender divides the message to be transmitted into fixed-size packets except for the last packet. The sender assigns each packet a sequence number starting from 0. Each receiver periodically sends ACK packet to its parent to inform it about the packets that the receiver has received correctly and those packets that are needed by the receiver.

3.2. Protocol Entities

There are five main entities in our protocol:

- *Sender*: The sender has a controller component which decides whether the sender should transmit new packets; retransmit lost packets or send messages to advertise itself as an ACK processor.
- *FHs and MHs Receivers*: Each receiver entity has a controller component which decides whether the receiver should receive data packets or send an ACK packet.
- *DRs and MAs*: These entities are a combination of the sender entity and receiver entity.

3.3. Transmission

The sender multicasts new data packets at regular intervals defined by a configuration parameter, *Tsend*. The number of packets transmitted during each interval normally depends on the space available in the sender send window. The sender can transmit at most one full Window of packets (*Ws*) during *Tsend* period, thereby limiting the sender's maximum transmission rate to $W_s * Packet\_Size/T_{send}$. Where W_s is the size of the sender sending window.

3.4. Acknowledgment

Receivers send acknowledgment packets periodically every *tack* to their parents indicating the status of their receive windows. Each of these ACK packets consists of two parts. The first part contains information about the receiver kind i. e., DR or MA and receiver number. The second part consists of a bitmap B of 0's and 1's and the last bit of this bitmap contains the number of the lowest unstable packet N i. e., the packet that is all packets with sequence numbers lower than this packet have been received correctly. Receivers use a bit vector of *Wr* bits to record the existence of correctly received packets stored in their buffers. For example, if the ACK contains N = 29 and B = 01011101. This indicates that the receiver has correctly received packets with sequence numbers less than 29 and that, it is requesting the retransmission of packets 29, 31 and 35 as indicated by zeros present in the bitmap B.

3.5. Round Trip Time Measurement and Tack Calculation

The ACK period, T_{ack}, the interval between two consecutive ACKs is a design parameter and should be chosen correctly in order to prevent redundant retransmissions. In order to send the ACK packet at the right time, delay sending the ACK packet until almost all data packets reaches their destinations. At the same time the ACK packet should reach the parent in time such that the parent is able to process this ACK message and have up to date information about the status of each receiver before the parent resend required packets. If the ACK message reaches the parent after it resends repair packets, then the parent resends packets that most of them have been received correctly by receivers. Also, if the ACK message is sent too early to the parent before some packets reaches their destinations, these packets will be resent by the parent even though most of these packets might be received correctly. Thus, the ACK message is sent to the parent before each transmission or retransmission message by an amount of time which is just enough for the ACK packet to travel to the parent and be processed there.

3.6. Sub Sub-Casting Technique

In the previous protocols such as RMTP or RM2 protocols [3, 4, 13, 14], when the number of requesting receivers for a certain packet is greater than a specific threshold, the sender or RS multicast the packet to all receivers that are located under its local group even though some of these receivers have received this packet correctly.

In this research, a new algorithm where the packet is multicast only to the requesting receivers in the local group. Multicasting of the packet only to receivers which request it would decrease the number of duplicated packets thus; reduces the number of packets transmitted through the network which in turn would increase the system performance. Retransmission of a packet by multicasting it only to receivers that request it. Subsequently, this requires creating a dynamic (temporary) multicast group by the sender or RSs whose receiver's members are only those which request the packet as shown in Figure 2.

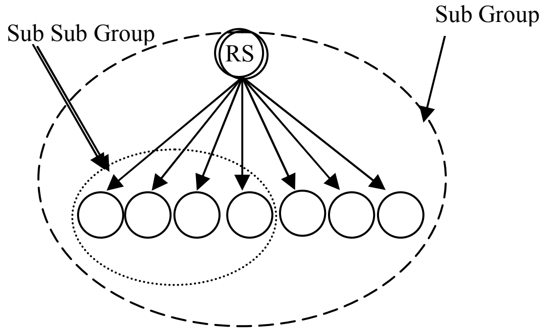


Figure 2. An example of sub sub-casting tree.

The temporary multicast group to be created is a sub group of the local group which itself is a sub group of the global multicast group. Thus, this group is a sub group of a sub group and we can call multicasting to this group sub sub-casting. In order to build the sub sub-casting, the topology information available at routers is used. When any receiver sends an ACK packet to its RS, the Id of the receiver is registered at the multicast router, which is connected directly to the RS. When the next retransmission time comes the multicast router knows the receivers that request the packet and if their number is greater than multicast threshold, a temporary multicast group is created whose members are those receivers which ask for the packet. If the number of receivers which request the packet is smaller than the multicast threshold, the packet is unicast to each requesting receiver as shown in sub sub-casting algorithm in Figure 3. This method can not be implemented using the NAK based protocol with NAK suppression because the idea of this method is to reduce the number of receivers that send NAK packets to the parent. However, sub sub-casting technique needs each receiver to send an ACK packet such that the multicast router would know all receivers that need the packet.

Algorithm SubSub_Casting()

```
Request_i ← nodes that request the packet
If Request_i > Threshold
    For all Request_i ∈ MobileAgent
        If Request_i Status == 0
            Sendpacket(MobileAgent (Request_i))
    For all Request_i ∈ DesignatedReceiver
        If Request_i Status == 0
            SendPacket (DesignatedReceiver (Request_i))
```

Figure 3. Sub sub-casting algorithm.

3.7. Reliability of the Protocol

The sender or RS keeps the packet in its buffer until it is sure that the packet has been received correctly from all of its receivers. The sender discards a packet from its buffer after receiving positive acknowledgment from all DRs and MAs of the first level. Also, each RS discards a packet from its buffer after it is sure that the packet has been received correctly from all receivers of the RS. This algorithm is good and enough for reliability as all RSs in the multicast group are working properly. However, if a receiver misses a certain packet and its RS died before it can retransmit the missing packet, then the receiver may not be able to recover the missing packet if the receiver rejoins a new parent which has already discarded the packet. To solve this problem the sender should keep each packet for a certain period of time which is enough for a receiver that loses connection with its parent to join a new parent and receive the missed packets.

4. Simulation

The simulator is developed based on event-driven. There are 5 events: Tsend, Tretrx, Tack, Hand_off_Begin, Hand_off_End, and Quit. All the events are scheduled per sender and receivers; the sender periodically sends a window of data packets. The first data packet of this window has the value of the number of the packets the sender will send in the current window and their sequence numbers. Every receiver upon receiving this value waits until it receives these packets and sends with the next ACK packet to its RS the time it received the last packet of these data packets. The RS uses this value to calculate the time difference between the transmission and retransmission messages where this time difference should be greater than or equal to the time taken by these data packets to reach farthest receiver and processed there added to the time needed for the ACK packet to reach the RS and processed there.

4.1. Description of the Simulation Program

In order to evaluate the performance of RMSS, we implemented the simulation program by using the C++ language. We assume static memberships where all nodes in the network join the multicast group at time 0 and no node joins the group during the simulation time. We do not evaluate membership's dynamics in this paper. We also assume that all nodes and links work properly and none of them fail during the simulation time.

4.2. Network Model

For each simulation run, a sample network topology is generated and for the same topology and configuration parameters the simulation program is run 15 times and the result is taken as the average among these iterations in order to have a stable result. Table 1 shows a summary of the simulation parameters.

Table 1. Simulation parameters.

Definition	Units	Value
PACKET_SIZE	Kbytes	1
WIRELINE_LINK_BW	Kbps	2000
WIRELESS_LINK_BW	Kbps	70
CELL_DWELL_TIME	ms	variable
TRANS_RETRANS_TIME_DIFFERENCE	ms	variable
WIRELINE_LOST_RATIO	-	0.05
WIRELESS_LOST_RATIO	-	0.1
DESIGNATED_RECEIVER_WINDOW_SIZE	packets	16
FIXED_RECEIVER_WINDOW_SIZE	packets	16
FIXED_RECEIVER_BITMAP_LENGTH	bits	32
DESIGNATED_RECEIVER_BITMAP_LENGTH	bits	32
MOBILE_AGENT_BITMAP_LENGTH	bits	40
MOBILE_HOST_BITMAP_LENGTH	bits	40
MOBILE_AGENT_RECEIVER_WINDOW_SIZE	packet	40
MOBILE_HOST_RECEIVER_WINDOW_SIZE	packet	40
WIRELINE_MAX_QUEUING_DELAY	ms	60
WIRELINE_MIN_QUEUING_DELAY	ms	20
WIRELESS_MAX_QUEUING_DELAY	ms	80
WIRELESS_MIN_QUEUING_DELAY	ms	20
WIRELINE_PROPAGATION_DELAY	ms	5
WIRELESS_ PROPAGATION_DELAY	ms	10

4.3. Error Model

Two reasons related to packet dropping. The first is due to the probability of error inherent to each link. The packets dropping probability due to this kind of errors is small. It is assumed that the packet loss probability due to this reason equal to 10^{-6} at each wired link and 10^{-3} at each wireless link.

The second reason of errors is a result of buffer overflow in routers in the network and this represents the dominant source for packet error. The value of packet loss probability due to the second reason is generated according to a uniform distribution U [0, 1]. Thus, the packet loss probability in the link can be represented by the following equation:

$$Err.Pro. = link\ error\ rate + buffer\ overflow \quad (1)$$

4.4. Link Delay Model

Three reasons related to delay in a link. The first is due to the propagation delay and this is represent a very small value and is the same for the same kind of links. The second reason depends on the link bandwidth and packet size. The packet will encounter a smaller delay when pass a link with higher bandwidth. Also the packet of the smaller size will encounter a smaller delay when pass the same link than a packet of larger size. Thus, delay in the link is decreased by increasing the link bandwidth and decreasing packet size. The third reason of delay in the link is due to congestion in the link and this cause the major part of delay. Thus, the link delay can be represented by the following equation:

$$Link\ Delay = propagation\ delay + s/bw + queuing\ delay \quad (2)$$

Where s is the packet size and *bw* is the link bandwidth. The first and second parts are constant for each link and the third part is obtained from a uniform distribution.

5. Results and Discussion

Interesting performance measures in the experiments are average number of duplicated packets, average number of acknowledgment packet, average delay, and average stability time. The number of DRs, MAs, FHs and MHs are taken as 5, 5, 50, and 50, respectively. The multicast threshold is equal to 2. The sending period and the time difference between the transmission and retransmission messages, (*Tr_Ret_Diff)* are taken as 340 ms. The cell dwell time is taken as 10000 ms. The following sections discuses the results.

5.1. The Relationship Between the Sending Period and the (Tr_Ret_Diff)

The purpose of this study is to show the relationship between the sending period and the average number of duplicated packets at different ratio between the sending period and (*Tr_Ret_Diff*). Figure 4 shows that the minimum number of duplicated packets occur when the *Tr_Ret_Diff* equals to the sending message period. The figure also shows that the number of duplicated packets is increased as the *Tr_Ret_Diff* differs from the sending message period. However, the duplicated packets increment is larger when the *Tr_Ret_Diff* is larger than the sending message period. The figure shows other important information which says that the sending period should be greater than or equal to a specific threshold value, 340 ms according to the given parameters, to have a minimum value of the number of duplicated packet.

5.2. The Effect of Sending Period on the System Performance

This section presents the simulation results of the effect of changing the sending period on the system performance. Figure 5 shows the effect of increasing the sending period on the average stability time, the average recovery latency time and the average delay time. The figure shows that the average stability time and the average recovery latency time are increased linearly with increasing the sending period. However, the average delays time taken by the packet to travel from the sender to the receiver which has no relation with the sending period. The effect of increasing the sending period on the average number of duplicated packets is presented in Figure 6. The average number of duplicated packets is decreased by increasing the sending period up to certain threshold and then becomes independent of the sending period.

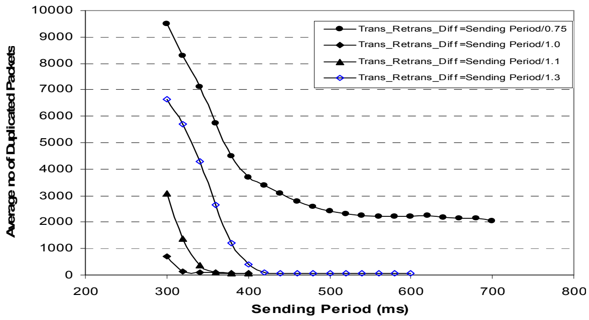


Figure 4. The relationship between the sending period and the average number of duplicated packets at different time difference between the transmission and retransmission messages.

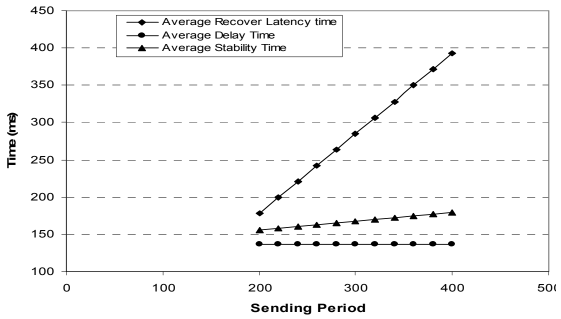


Figure 5. The effect of the sending period on the average (recovery latency, delay, and stability) times.

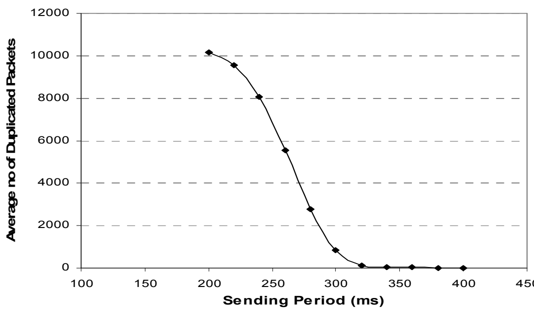


Figure 6. The effect of sending period on the average no. of duplicated packets.

5.3. The Effect of Sub Sub-Casting Technique on the System Performance

Figure 7 shows the relationship between the average number of duplicated packets and the multicast threshold with and without using the sub sub-casting technique. The figure shows that the average number of duplicated packets is independent from the multicast threshold when using the sub sub-casting technique which is very good advantage because one can make the multicast threshold small which increases the number of packets retransmitted using multicast technique which is much more efficient than the unicast technique. Whereas the figure shows that the average number of duplicated packets is very high when the multicast threshold is small for the protocols without sub sub-casting feature and in order to decrease the number of duplicated packets, one has to increase the multicast threshold which causes most of the packets to be retransmitted using unicast technique which is not efficient. The reason that the number of duplicated packets independent of the multicast threshold when the sub sub-casting technique is used is that the required packets are retransmitted to the requesting receivers only of the sub-group.

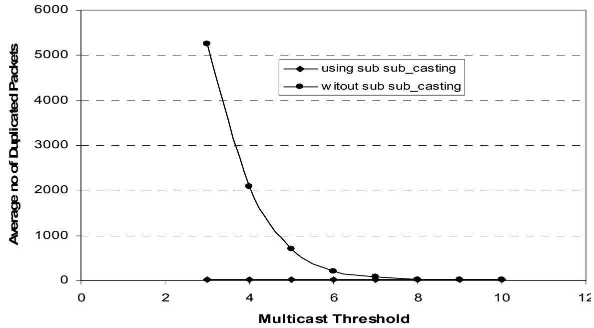


Figure 7. The relationship between the multicast threshold and average no. of duplicated packets.

5.4. Scalability of the Protocol

The purpose of this study is to show the effect of increasing the number of receivers on the system performance. First, we will study the effect of increasing the number of mobile hosts on the system performance with and without using the sub sub-casting technique. Figure 8 shows that the relationship between the number of mobile hosts and the average number of duplicated packets with and without using sub sub-casting technique, when the sending period is equal to 300 ms. The figure shows that the average number of duplicated packets is much smaller when the sub sub-casting technique is used. The duplicated packets that appear in the case when the sub sub-casting technique is used are because the Tr_Ret_Diff of 300 ms is not enough such that the next retransmission message comes before some of the ACK packets have not arrived to their parent. Thus, the data packets that these ACK packets acknowledge will be retransmitted again even most of them might be received correctly.

Figure 9 shows the same relationship but now the sending period is equal to 340 ms which is large enough for probably all ACK packets to reach their destinations. Thus, now the number of duplicated packets is approximately zero when using the sub sub-casting technique whereas the number of duplicated packets is still very high when the sub sub-casting technique is not used.

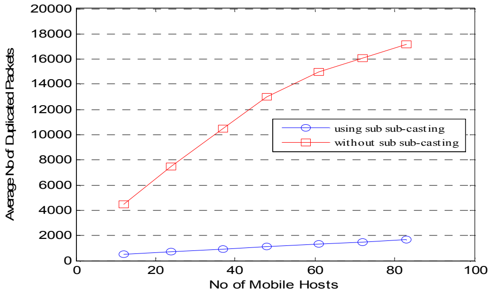


Figure 8. The relationship between the average no. of duplicated packets and the no. of mobile hosts when the sending period is equal to 300 ms.

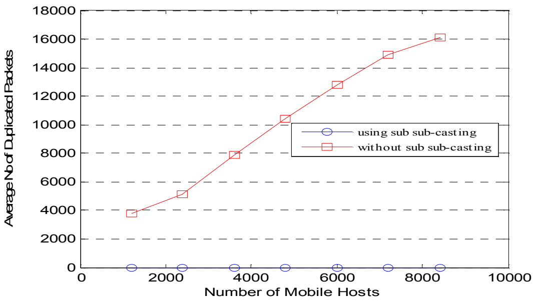


Figure 9. The relationship between the average no. of duplicated packets and the no. of mobile hosts when the sending period is equal to 340 ms.

Figure 10 shows the effect of the multicast group size on the average implosion, the figure shows that the average implosion is increased linearly with increasing the number of receivers. However, the rate of increasing is very small, which is an advantage of RMSS protocol and it proves that this protocol can serve a large number of receivers.

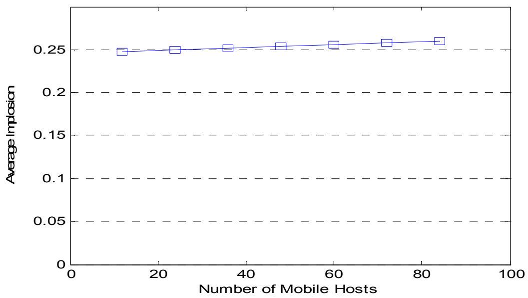


Figure 10. The relationship between the average implosion and the no. of receiver when the sending period is equal to 340 ms.

Figure 11 shows the effect of increasing the number of receivers on the average delay time, average stability time and average recovery latency time. The figure also shows that the effect of increasing the number of receivers on the different average time is very small and the average recovery latency time is approximately independent from the number of receivers in the network. These results also show that RMSS is scalable and can be used for large networks.

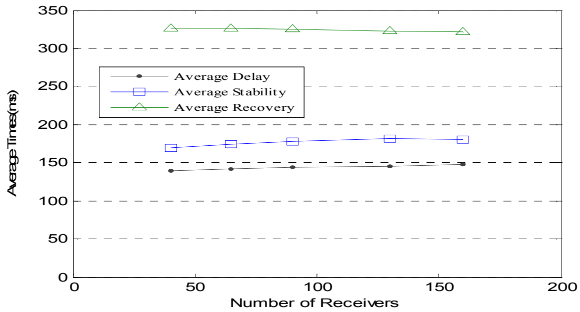


Figure 11. The effect of the no. of receivers on the different average times when the sending period is equal to 340 ms.

5.5. The Effect of Changing the Time (Tr_Ret_Diff) on the System Performance

Here, we want to study the effect of changing the value of the *Tr_Ret_Diff* on the performance of the system. Figure 12 shows the effect of changing the *Tr_Ret_Diff* on the average recovery latency time. The figure shows that the recovery latency time is increased linearly by increasing the *Tr_Ret_Diff*. The reason is that in the RMSS, the packets that are required to be retransmitted are not retransmitted immediately but the parent waits until the next retransmission message time comes and then retransmitted all required packets. Thus, the recovery latency time which is the time difference between the failure time and the successful time depends on the time of the next retransmission message which depends on the *Tr_Ret_Diff*. Figure 13 shows the effect of changing the *Tr_Ret_Diff* on the average number of duplicated packets. The figure shows that the average number of duplicated packets is decreased by increasing *Tr_Ret_Diff* up to a certain value above which the average number of duplicated packets remains constant. The reason is that *Tr_Ret_Diff* is smaller than the time required for the packets to travel to their destinations and processed plus the time required for the acknowledgment packets of these packets to travel to the parents and processed. Thus, the next retransmission message comes too early before the acknowledgment packets of some receivers reach the parents and thus these parents resend the packets that their acknowledgment packets have not arrived yet even some of these packets will be received correctly which results in duplication of packets. When the *Tr_Ret_Diff* is large enough, then the next retransmission message time comes after all acknowledgment packets reach parent and thus the number of duplicated packets due to this reason is equal to zero which is shown in the figure.

6. Conclusion

In this paper, a scalable reliable multicast protocol has been proposed which is mainly devoted for transmission of non real time information over a combined (fixed/mobile) network. In this protocol, by applying sub sub-casting technique, the number of duplicated packets due to retransmission of packets will be zero which will increase the system performance. Another benefit from this modification has been discovered from the results which show that the number of duplicated packets is independent from the multicast threshold result in that a large number of retransmitted packets will be retransmitted using multicast technique rather than using unicast technique which is less efficient.

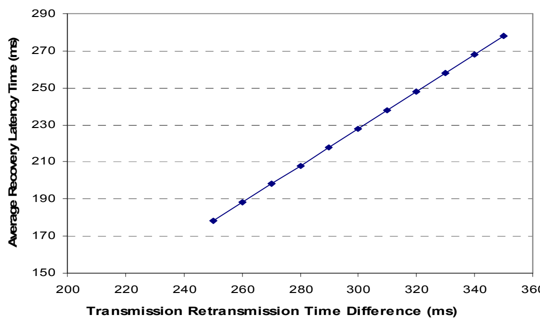


Figure 12. The effect of changing the time difference between the transmission and retransmission messages and the average recovery latency time.

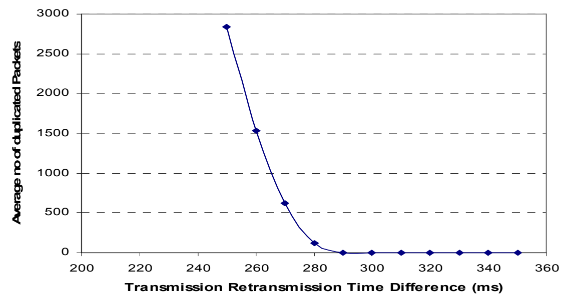


Figure 13. The effect of changing the period of time between the transmission and retransmission messages on the average no. of duplicated packets.

In addition, the results prove that the proposed protocol is scalable and can be used for large networks where the average implosion, average delay, average stability and average recovery latency times increments are small when increasing the multicast group size or the number of mobile hosts.

In future work, we will studying the complexity and overhead on the system due to including the sub sub-casting technique which requires creating temporary sub sub-group and routers assistance and studying the delay that will result due to this creation.

References

[1] Acharya A. and Badrinath B., "A Framework for the Delivery of Multicast Messages in Networks with Mobile Hosts," *Wireless Networks*, *Special Issue on Routing in Mobile Communication Networks*, vol. 1, no. 2, pp. 199-219, 1996.

[2] Acharya A. and Badrinath B., "Delivering Multicast Messages in Networks with Mobile Hosts," *in Proceedings of the 13th International Conference on Distributed Computer Systems*, May 1993.

[3] Bagnall P., Briscoe B., and Poppitt A., "Taxonomy of Communications Requirements for Large-Scale Multicast Applications," available at: http://www.labs.bt.com/people/briscorj/projects/lsma/taxonomy-reqs.txt, November 1997.

[4] Buskens R., Paul S, and Siddiqui M., "Reliable Multicast of Continuous Data Streams," *Bell Labs Technical Journa*l, vol. 2, no. 2, pp.151-174, Spring 1997.

[5] Chikarmane V., Williamson C., Bunt R., and Mackrell W., "Multicast Support for Mobile Hosts Using Mobile IP: Design Issues and Proposed Architecture," *ACM/Baltzer Mobile Networking and Applications*, vol. 3, no. 4, 1999.

[6] Cordeiro C., Kelner J., and Sadok D., "Performance Analysis of a Multicast Protocol for Wireless Environments," *Simulation*, vol. 75, no. 1, July 2000.

[7] Farinacci D., Speakman T., Lin S., and Tweedly A., "PGM Reliable Transport Protocol Specification," August 1998.

[8] Fenner W., "Internet Group Management Protocol, Version 2," Internet Draft, April 1996.

[9] Floyd S., Jacobson V., Liu C., McCanne S., and Zhang L., "A Reliable Multicast Framework for Light-Weight Sessions and Application Level Framing," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 784-803, December 1997.

[10] Floyd S., Jacobson V., McCanne S., Zhang L., and Liu C., "A Reliable Multicast Framework for Light-Weight Sessions and Application Level Framing," *in Proceedings of the ACM (SIGCOMM'95) Conference on Applications, Technologies, Architectures and Protocols for Computer Communications*, August 1995.

[11] Johnson D. B., "Scalable and Robust Internetwork Routing for Mobile Hosts," *in Proceedings of the 14th International Conference on Distributed Computing Systems*, 1994.

[12] Harrison T., Williamson C., Bunt R., and Mackrell W., "Mobile Multicast (MoM) Protocol: Multicast Support for Mobile Hosts," *in Proceedings of the ACM Mobicom'97*, Budapest, Hungary, pp. 151-160, September 1997.

[13] Hanna S., Patel B., and Shah M., "MADCAP Protocol (MADCAP)," RFC 2730, December 1999.

[14] Lin J. and Paul S., "RMTP: A Reliable Multicast Transport Protocol," *in Proceedings of the IEEE INFOCOMM'96*, pp. 1414-1424, March 1996.

[15] Paul S., Sabnani K. K., Lin J. C., Bhattacharyya S., "Reliable Multicast Transport Protocol (RMTP)," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, April 1997.

[16] Prakash R., Schiper A., and Mohsin M., "Reliable Multicast in Mobile Networks," *in Proceeding of IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 16-20, March 2003.

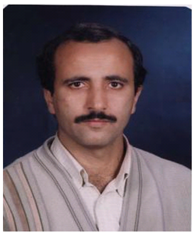
[17] Rizzo L. and Vicisano L., "RMDP: An FEC-Based Reliable Multicast Protocol for Wireless Environments," *Mobile Computing and Communications Review*, vol. 2, no. 2, 1998.

[18] Sadok D. H., Cordeiro C. d. M., and Kelner J., "A Reliable Subcasting Protocol for Wireless Environments," *in Proceedings of the 2nd International Conference Mobile Wireless Communications Networks*, Paris, France, pp. 174-185, May 2000.

[19] Whetten B. and Taskale G., "An Overview of Reliable Multicast Transport Protocol II," *IEEE Network Magazine*, vol. 14, no. 1, pp. 37-47, Jan/Feb 2000.

[20] Whetten B. and Taskale G., "The RMTP-II Protocol," available at: http://www.talarian.com/rmtp-ii, April 1998.

[21] Yoon W., Lee D., Yu C., and Kim M., "Tree-Based Reliable Multicast in Combined Fixed/Mobile IP Networks," *in Proceedings of the 25th Annual IEEE Conference on Local Computer Networks (LCN'00)*, Tampa, FL, USA, pp. 478-487, November 2000.



Ali Alsaih received his BS degree in electrical and computer engineering from Sana'a University, Yemen, and the MS and PhD degrees in electrical engineering from the University of Alexandria, Egypt. Currently, he is an assistant professor of computer engineering at Sana'a University, Yemen. His research interests include computer networks, multicast routing protocol design and evaluation in wireline, and wireless networks.

Tariq Alahdal is a PhD Student at the Department of Communication Technology and Networks, Faculty of Computer Science and Information Technology, University Putra Malaysia. His research area focuses on reliable multicasting protocol, congestion control and reliable protocol for multicast protocol, and reliable multicast routing protocol for ad-hoc networks.