Sep. 30
International Journal of advanced studies in Computer Science and Engineering
IJASCSE Volume 4, Issue 9, 2015

REDUCING BUFFER OVERHEAD IN A RELIABLE MAODV PROTOCOL

Parul Nabiyal
Department of CSE,
National Institute of Technology, Jalandhar
Jalandhar, India

Abstract: As the technology of Mobile ad hoc networks (MANETs) progresses, many kinds of new applications merge demanding new and better services with each research being done. The major challenge for these ad hoc networks is to adapt the multicast service in an environment where failures are frequent. To perform a reliable multicast delivery of data packets, a feedback from each multicast receiver is necessary which in turn indicates whether or not is transmission is required.

A Reliable Multicasting AODV protocol provides reliability in MAODV by sending positive acknowledgement for each packet and maintaining are transmission queue. In this paper, we have proposed an algorithm which distributes the overhead of sender node into a set of its one-hop neighbours i.e Forwarding Nodes (FNs). This selected set is responsible for receiving the acknowledgements and retransmissions hence decreasing the overall. The proposed algorithm was tested in a simulation environment of 75 and 150 nodes using OPNET giving high delivery ratio and less packet dropping.

Keywords: **ad hoc network, reliability, multicasting, buffer, forwarding nodes (FNs)**

I. INTRODUCTION

A mobile ad hoc network is a network of mobile devices which communicates through wireless radio and do not have any fixed topology. These nodes can act as host as well as router.

Main characteristics of MANETs being infrastructure-less, wireless links, node movement, power limitation, self-configuring and dynamic topology. Due to obvious reasons, routing in Mobile ad hoc network is a challenging task and has received a remarkable amount of attention from researchers around the world. Reliable multicast delivery of packet refers to the delivery of multicast packets to all the multicast receivers. So, one or more members need to buffer these packets for possible error recovery. Though many protocols and algorithms have been proposed on multicast routing in MANETs, it is important to focus on controlling the overhead and retransmission traffic with reliable multicasting.

Reliable Multicasting AODV provides reliable transmissions of data from sender to all the receivers. It is an extension of AODV protocol. Sender breaks the data into packets of some fixed size except the last one. It assigns the sequence number to each packet starting from 0 in increasing order, so that receiver can distinguish between the duplicate packets. It also helps in detecting the lost packet. For every received packet, it sends a positive acknowledgement (ACK) to the sender. A lost packet is detected by the gap in sequence number. After sending the packet, each sender waits for a particular time (T_{ack}). If acknowledgement of the sent packet is not received till time T_{ack}, sender assumes the packet as lost and does the entry in its retransmission queue, retxQ, where each entry consists of sequence number of the packet and list of receiver nodes. The retransmission queue is examined every T_{retx} by the sender node. Retransmissions are done whenever the queue is non-empty.

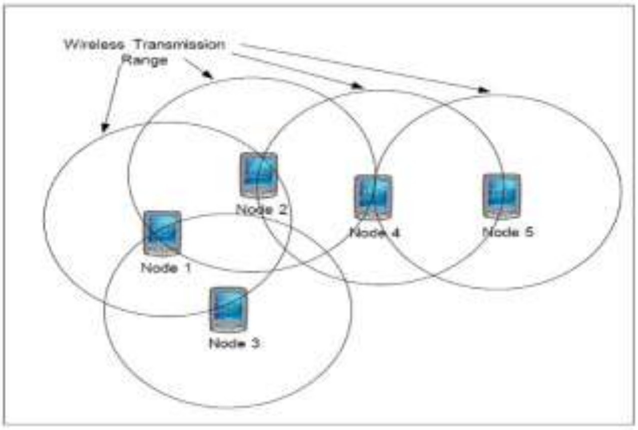

Fig. 1: Connectivity in a mobile ad hoc network

II. RELATED WORK

Many reliable multicast protocols for MANETs have been proposed in the recent years. These protocols use different approach to increase the packet delivery and their buffer management.

A Reliable Multicast Protocol for Adhoc (Remhoc) [2] follows receiver-initiated negative acknowledgement (NAK). This protocol relies on receivers to detect the lost packets and to initiate loss recovery process for such losses. Source send the packets with increasing sequence number so it'd be easy to detect packet lost through gap in sequence number. Every receiver has a request timer which is random for each receiver. As this timer expires, receiver can send its NAK. Source responds to this NAK by retransmitting the packet.

Although the protocol provides high reliability to the multicasting scheme, it increases the overhead as there is only one node (sender) which has to receive all the acknowledgements and buffer all packets until the retransmission is done for lost packets. So, due to these reasons there are chances of buffer overflow where buffer starts dropping the packets due to lack of space. Problem with NAK approach is long end to end delays where source has to wait for a long time. An algorithm named Reliable multicasting algorithm (RMA) was proposed [18] which supports sender initiation. Connectivity between reachable nodes is maintained by periodically exchanging HELLO messages between neighbors. RMA considers path with longer lifetime as the best path and the path must have more of the group members. Source creates sequence table for each multicast session where it stores the sequence number of message which is sent. Sequence entry gets deleted each time all the group members receives the packet and in case packet is not received by all the receivers, it is resend and a special field is marked in the table. Here, as all receivers send the acknowledgement to the source, it may suffer from feedback implosion problem.

Reliable multicasting transfer protocol (RMTP) [19] guarantees higher reliability and lossless delivery of data packets even in bulk transfer. Receivers are grouped into local regions where each region has a representative called designated receiver (DR). DRs are chosen according to the approximate distance of all receivers. Only DRs are permitted to send status of the packets to the sender about its whole local region. DRs handle the retransmissions, process the received acknowledgements and buffer the received data. But for long sessions where they may have shortage of space, the caching part can be impractically large.

Bimodal Multicast Protocol (BMP) [15] is an extension of gossip protocol. . It is based on probabilistic broadcast and has two phases: First phase detects message loss and second phase correct such losses but second phase runs only when needed. Nodes send their message summary history to randomly selected nodes and hence they themselves detect the missing packets and send the retransmission request. If a message is there for recoveryfrom too long, the protocol give up and the packet is marked as lost. Gossip about a message continues for a fixed number of rounds and it is necessary to send the retransmission request within the same round where the message has lost otherwise the request would be dropped. Drawback of BMP is that each node buffers a packet only for a fixed amount of time.

Improved version of BMP, named as Lightweight Randomized reliable multicast protocol (RRMP) [20], has two kinds of buffers: short term buffers and long term buffers. Short term buffers can keep the message for a fixed interval of time whereas long term buffer stores the message for quite a long time and discards the message only when there is no retransmission request for it in a long time. Lost packets are recovered in this protocol either using local or remote recovery. As the number of participants increase, this process is very time-consuming for the receiver to search and find the correct repair nodes.

Stepwise probabilistic buffering algorithm [21] provides reliability along with scalability and it is based on epidemic algorithms. The stepwise probabilistic buffering reduces the Congestion controlled adaptive lightweight multicast transport protocol (CALM) [3] where congestion in the network is indicated through NAK. As source gets the NAK, source adds the receiver in the list and sends the data packet with an indicator which indicates the receiver an indicator which indicates the receiver to send the acknowledgement.
On receiving the acknowledgement, receiver is received from the list. When a node receives a buffering request message for a particular data, the request is accepted with probability (1 – BF). Otherwise, the node forwards the message to a randomly selected node from its partial view with a probability equal to BF. Frequent exchange of history messages to determine the buffeters of a data message causes high traffic, resulting to higher delays.

The aforementioned algorithms indicate that several existing approaches are not sufficient to guarantee an efficient buffer management and reliable multicasting in a mobile ad hoc network.

III. PROPOSED APPROACH

In the proposed approach, an algorithm has been developed which selects a set of one-hop neighbors from the sender, named as Forwarding Nodes (FNs). These FNs receive the packets from the sender so as to forward it to the further receivers. When FNs receive the packet from sender, it buffers the packet for retransmission purpose, in case if there is any retransmission request. Receivers in turn send the ACK packet to the FNs and not to the sender. Key idea behind selecting the subset of one hop neighbors is to retransmit the requested data lessening buffer overhead and receiving the acknowledgements from different receivers avoiding ACK-implosion problem.

While sender has to send a packet to the multicast group, it sends a message to all its Forwarding Nodes (FNs) indicating them to find a route to the destination. After the route-finding, FNs update their table and reply back to the sender with their table entry. Sender selects a FN on the basis of freshest sequence number and number of hop routes. Now it is the responsibility of the FN to send, receive acknowledgement and retransmit for the packet.Following is the algorithm:

Algorithm for selecting the FNs

1. H_One = One-hop neighbours
2. H_Two = Two-hop neighbours
3. FN = Nodes in H_One with unique neighbours in H_Two
4. H_One = H_One – FN
5. H_Two = H_Two – N(FN)
6. while(H_One ≠ null or H_Two ≠ null)
7.   for each node in H_One
8.   Check B_f for each node
9.   n = highest B_f node
10.   FN = FN + n
11. H_Two = H_Two – N(n)
12. H_One = H_One – n
13. end while
14. return FN.

International Journal of advanced studies in Computer Science and Engineering
IJASCSE Volume 4, Issue 9, 2015

amount of buffering by distributing the buffering load to the entire system where every node does not have the complete view of the entire receiver group. In every receiver group, all peers only have partial knowledge of the participants. In this algorithm, only a small subset of the node keeps a data message in its long-term buffer.

The buffering request messageis sent by source to randomly selected nodes in its partial view to determine the bufferers of a data message. Buffer fullness (BF) node ratio is the ratio of the number of messages stored in the node buffer to its long-term buffer capacity. Steps-to-live (STL) value attached to a buffering request message indicates the maximum number oftimes that request messages can be forwarded among nodes.

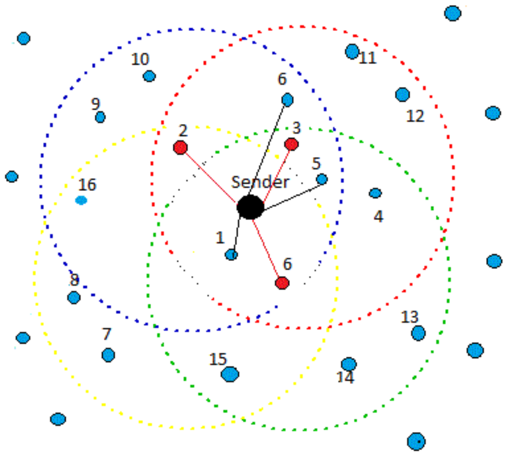

Fig. 2: Sample network using the proposed algorithm

This algorithm also avoids the ACK implosion problem as it requires only the selected FNs to send ACKs to the sender, instead of all receivers sending the ACKs to the sender.
Fig. 2 represents the sample network for proposed algorithm where black node represents the sender, blue node represents the mobile nodes and red nodes represent the selected FNs.

IV. PERFORMANCE EVALUATION

In the simulation, we have considered the scenario of 75 and 150 nodes using OPNET as the simulation tool. Simulation time for each scenario is taken as 30 minutes. The nodes are randomly placed with certain gap from each other over 1500*1500 meters campus environment. Each node is allowed to generate and forward packets as defined in its parameter “MANET Traffic Generation Parameter” as mentioned in table 1 & 2, with a transmission range of 250m.

Out of all the nodes, almost half of the nodes have been made the source of generating packets in various intervals. Two kinds of traffic generation parameters are defined, each node following any one of these two:

Here, B_f represents the buffer factor:

$$B_f = \frac{\text{Remaining buffer space}}{\text{Total buffer space}}$$

We use $N_k(u)$ to represent the neighbor set of u, where nodes in the set are not further than k−hops from u. $N_1(u)$ can also be represented as N(u).
The proposed approach also supports mobility in a different way by dynamically selecting the forwarding nodes.These selected FNs must cover all the two-hop neighbors of sender.

Table II: Traffic Generation Parameter_2

Attrbiute	Value
Start time (seconds)	100
Packet Inter-arrival time (seconds)	Constant (25)
Packet size (bits)	Constant (1024)
Destination IP address	Random
Stop time (seconds)	1800

For node movement, vector trajectory is being used. Movement is defined via bearing, ground speed, and ascent rate of attributes of a mobile node.
New protocol is named as “Improved AODV” and is compared with the original (reliable AODV) protocol using the following performance metrics:

- **Total Packets dropped:** It refers to the total packets dropped by the buffer when there is no more space in it to keep new packets.
- **Packet Delivery Ratio (PDR):** PDR is taken as the ratio of total packets received to the total packets sent. It basically determines the reliability of a protocol.
- **Average Retransmitted Packets:** It is taken as the ratio of number of retransmission packets transmitted to the total number of original data packets transmitted.
- **Number of Route Requests:** Total number of route requests has been calculated by measuring the total number of RREQ messages sent over the network.

Fig. 3 and Fig. 4shows that the packet dropping has been decreased in the improved protocol. As the overhead has been distributed, each buffer has lesser number of packets to deal with and hence less packet dropping. Hence, total packets dropped have decreased.

Packet delivery ratio has increased from 90% to 94% for 75 nodes and from 93% to 95.2% for 150 nodes as Fig. 5 and Fig. 6 depict.

Table I: Traffic Generation Parameter_1

Attribute	Value
1. Start time (seconds)	100
Packet Inter-arrival time (seconds)	Constant (24)
Packet size (bits)	Constant (1024)
Destination IP address	Random
Stop time (seconds)	700

Attribute	Value
2. Start time (seconds)	700
Packet Inter-arrival time (seconds)	Constant (28)
Packet size (bits)	Constant (1024)
Destination IP address	Random
Stop time (seconds)	1800

V. CONCLUSION AND FUTURE WORK

In this paper we have proposed an algorithm into a reliable multicast AODV protocol so as to decrease its overhead while sending the multicast packets. As the comparison result shows, reliability has increased and overhead has decreased. While analyzing more deeply, it is realized that while running the algorithm at each sender before sending the data packets, end to end delay has increased. Original protocol runs simply by sending and receiving RREQs/RREPs, but the improved protocol has to run the algorithm in addition to the existing process. This affects the throughput too. So, throughput and end to end delay shows a little negative result for the new improved protocol.

Future work of the paper can be overcoming the increased end to end delay as delay plays an important role in the network simulation of MANETs. Overall throughput can also be considered so that there are more packets sent per unit time.

Fig.7and Fig. 8 depicts that the average retransmission ratio keeps on increasing with the simulation time in the case of both the protocols. At the end of simulation, ratio has been decreased from 12 to 8.5 in case of 75 nodes and from 24 to 15 for 150 nodes, which give the positive sign for the improved protocol.

Total number of RREQ messages determines the total number of route requests. Fig. 9 and Fig. 10 show that there is a decrease in total route requests sent. At the end of simulation, for 75 nodes, protocol has decreased RREQs from 3300 to around 2800. Changes are similar in case of 150 nodes, as number of RREQs has decreased from 6800 to 5400.

Fig. 3: Total Packets dropped for 75 nodes

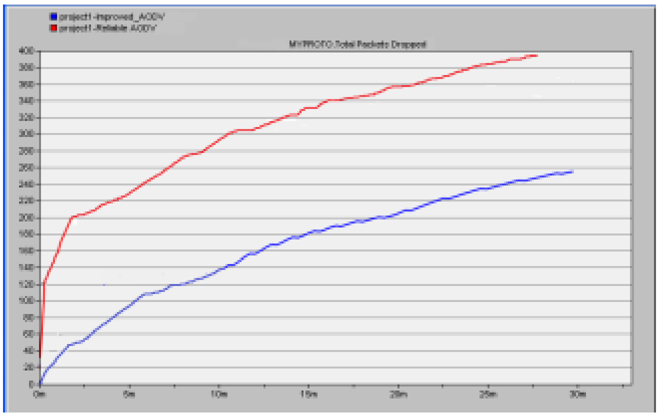

Fig. 4: Total Packets Dropped for 150 nodes

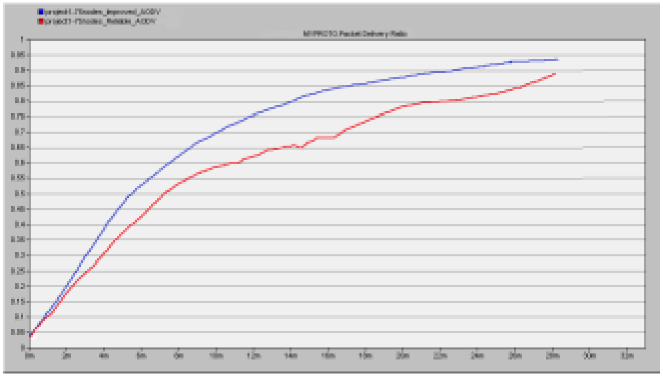

Fig. 5: Packet Delivery Ratio for 75 nodes

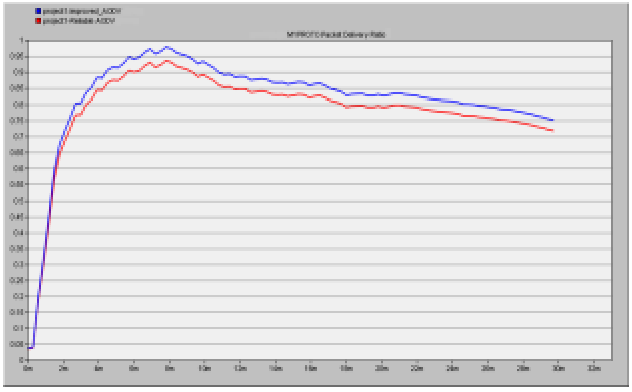

Fig. 6: Packet Delivery Ratio for 150 nodes

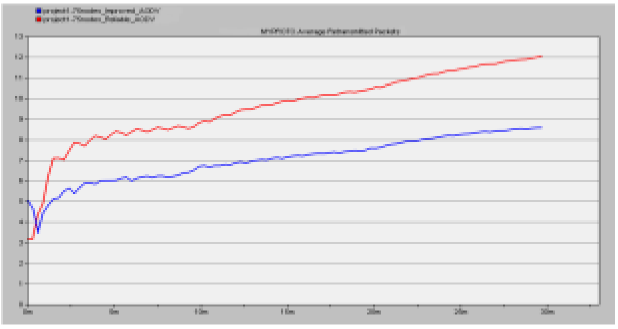

Fig. 7: Average Retransmitted Packets for 75 nodes

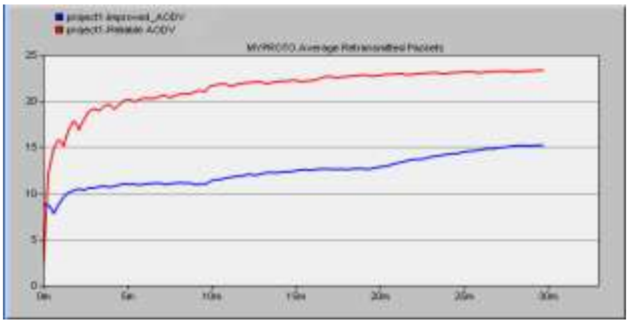

Fig. 8: Average Retransmitted Packets for 150 nodes

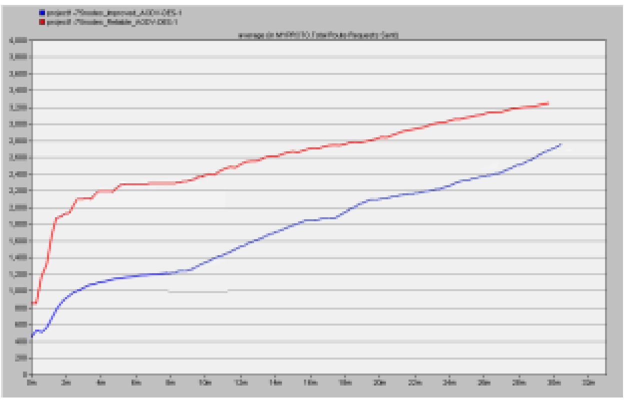

Fig. 9: Total RREQs in 75 nodes

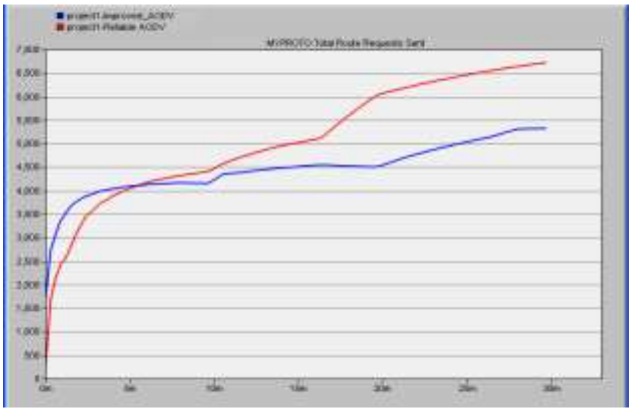

Fig. 10: Total RREQs in 150 nodes

REFERENCES

[1] X. Bangnan, S. Hischke, B. Walke, "The role of ad hoc networking in future wireless communications", International Conference on Communication Technology (ICCT 2003).

[2] H. B. A. Sobeih and A. Fahmy. "Remhoc: A reliable multicast protocol for wireless mobile multi hop ad hoc networks", In IEEE Consumer Communications and Networking Conference (IEEE CCNC 2004), Las,Vegas,NV, USA.

[3] Ken Tang, Katia Obraczka, Sung-Ju Lee, and Mario Gerla, "Congestion controlled adaptive lightweight multicast in wireless mobile ad hoc networks", In Proceedings of IEEE ISCC, July 2002

[4] Xiaoyan Hong, KaixinXu, and Mario Gerla, "Scalable routing protocols for mobile ad hoc networks", Proc. 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2002.

[5] A. M. Costello and S. McCanne, "Search party: Using randomcast for reliable multicast with local recovery," in Proceedings of the18th IEEE Conference on Computer Communications (INFOCOM'99), New York, NY, 1999.

[6] J. Pereira, U. do Minho, L. Rodrigues, U. De Lisboa, M. Monteiro, R. Oliveira, and A. M. Kermarrec, "Neem: Network-friendly epidemic multicast " in Proceedings on 22nd InternationalSymposium on Reliable Distributed Systems (SRDS'03), Florence, Italy, 2003.

[7] M.Maleki,M. Pedram. Lifetime-AwareMulticast Routing inWireless Ad Hoc Networks", in IEEE Wireless Communications and Networking Conference (WCNC), Mar. 2004.

[8] Lee, S.J., M. Gerla and C.C. Chiang., 1999. On-demand multicast routing protocol. In Proceedings of the IEEE Wireless Communications and Networking Conference, WCNC '99, September 1999.

[9] Alsaqour, R., M. Abdlehaq and T. Abdullah, "Modelling the position information inaccuracy in MANET position based routing protocols", Res. J. Appl. Sc. Eng. Tech., 2011.

[10] Floyd, S., V. Jacobson, C. S. Liu, S. McCanne and L. Zhang, "A reliable multicast framework for lightweight sessions and application level framing", IEEE/ACM T. Network, 1997.

[11] Tariq Alahdal, ShamalaSubramaniam, Mohamad Othman and ZuriatiZukarnain, "An adaptive reliable multicast protocol in ad hoc networks", Proceedings of the 2007 IEEE International Conference on Telecommunications and Malaysia International Conference on Communications, 14-17 May 2007, Penang, Malaysia

[12] Tang, K. and M. Gerla, "Reliable multicast of the On-demand multicast routing protocol" in The 5th World Multi-Conference on Systemics, Cybernetics and Informatics, 2002, Orlando, FL.

[13] D. B. Johnson and D. A. Maltz, "Dynamic source routing in adhoc wireless networks," Mobile Computing, T. Imielinski and H. Korth, Eds., Kluwer (1996) , pp.153–171.

[14] C.E. Perkins and T.J. Watson, "Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers", In ACMSIGCOMM'94 Conference on Communications Architectures, London, UK, 1994, 234-244.

[15] K. P. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, and Y. Minsky, "Bimodal multicast," ACM Transactions on ComputerSystems (TOCS), 2008.

[16] S.J. Lee, E.M Royer. & C.E Perkins, "Scalability study of the Ad Hoc On-Demand Distance Vector Routing Protocol", ACM Wiley International Journal of Network Management, Volume-13 Issue-2,, 97-114, March/April 2003.

[17] S. Sethi and S.K. Udgata, "A Reliable Multicasting AODV for Mobile Ad Hoc Network", in International Conference and Workshop on Emerging Trends inTechnology (ICWET), April 2012.

[18] ThiagarajaGopalsamy, MukeshSinghal, D. Panda, and P. Sadayappan, "A reliable multicast algorithm for mobile ad hoc networks", 22nd IEEE International Conference on Distributed Computing Systems, 1998.

[19] J. C. Lin and S. Paul, "RMTP: A reliable multicast transport protocol", Proc. of IEEE INFOCOM'96, Apr. 1996.

[20] Z. Xiao, K. P. Birman, and R. Van Renesse, "Optimizing buffer management for reliable multicast", Washington, DC, 2002.

[21] E. Ahi, M. Caglar, and O. Ozkasap, "Stepwise probabilistic buffering for epidemic information dissemination," in International Conference on Bio-Inspired Models of Network,Information, and Computing Systems, Cavalese, Italy, 2010, pp. 1-8.