Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 92 (2016) 222 – 232
Procedia Computer Science

2nd International Conference on Intelligent Computing, Communication & Convergence

(ICCC-2016)

Srikanta Patnaik, Editor in Chief

Conference Organized by Interscience Institute of Management and Technology

Bhubaneswar, Odisha, India

Efficient Buffer Management Protocol for Multicast Streaming in MANET

[1]Kavitha Subramaniam* , [1]Latha Tamilselvan

[1]Department of Information Technology ,School of Computer and Information Sciences
B.S. Abdur Rahman University, Chennai, Tamilnadu, India

Abstract

Buffer management techniques are essential while handling multicast streaming in MANET since real-time data will involve playback delay and jitter. In this paper, an efficient buffer management protocol is developed for streaming data in multicast groups. The frequently requested video data can be buffered in any intermediate nodes along the multicast tree from the source to the receivers. When packets are received, they are classified as real-time or non-real-time and placed into respective queues. Cumulative weight of the packets in the real-time buffer is then estimated based on number of hops, deadline and waiting time. Based on the estimated weight value, transmission priorities are assigned. The buffer space is dynamically adjusted depending on the number of intermediate nodes along the multicast tree from the source to the receivers. Simulation results show that the proposed buffer management protocol reduces the latency and energy consumption while increasing the packet delivery ratio.

Peer-review under responsibility of the Organizing Committee of ICCC 2016

Keywords: Buffer Management Protocol, MANET, real-time video data buffering, Mobile Ad-Hoc Network.

* Corresponding author. Tel.: 91-9884730376
*E-mail address:*skavitha.jegan@gmail.com

Peer-review under responsibility of the Organizing Committee of ICCC 2016
doi:10.1016/j.procs.2016.07.349

1. Introduction

1.1. Mobile Ad-Hoc Network

Mobile Ad hoc Network (MANET) is a network with dynamic topology and mobile nodes. Due to the dynamic nature of network, there is no central control. Hence, nodes communicate with other nodes through intermediate nodes. The intermediate nodes are normal nodes in the same network and assume the responsibility of forwarding packets on the route from source to destination [1]. MANET consists of a group of wireless nodes, which help each other in forwarding packets to enable communications. This type of network requires no fixed network infrastructure [11]. Since the node may change their position unpredictably, the data transmission can be temporarily disturbed if any link on the path fails. Ad hoc routing protocols have been designed to reroute traffic when opposed with network congestion, faulty nodes, and dynamically changing topologies [2].

1.2. Buffer Management for Video Streaming

An increasing number of ad-hoc network applications require a sender to distribute the same data to a large group of receivers. These applications fall in the category of group communication, as opposed to the traditional one-to-one communication. Accordingly, multi-point communication (i.e., multicast) offers the most efficient way to support this application by delivering a single message to multiple recipients. For supporting multicast protocol reliability, the error recovery mechanism is achieved via efficient buffer management design. An efficient buffer management algorithm is an indispensable part of an error recovery mechanism. The existing buffer management algorithms are classified into (i) Reducing the buffer usage (ii) Flow control and (iii) Providing packet stability [3].

Reducing Buffer Size

Upon receiving a packet, a receiver node determines whether it should buffer the packet using a hash function based on network address and the packet number. Receivers that lost packets use a hash function to select the set of members that have the packet in the buffer and request a packet retransmission.

Flow Control

Flow control is an adaptive mechanism that deals with varying resources such as CPU speed and receiver node bandwidth. Buffer optimization techniques in this category adjust the network rate to minimize buffer overflows at the receiver nodes.

Packet Stability

A packet is stable when it is delivered to all group members. Buffer management approaches that explicitly take stability into account exist. The members are partitioned into groups, and every node is included in the error recovery. This process is achieved by letting the receivers periodically exchange history information about the received sets of packets [3].

1.3. Problem Identification

In our first paper [6], we have proposed a Predictive Energy efficient and Reliable Multicast Routing Protocol. The technique uses PSO algorithm to construct energy efficient and reliable multicast tree. The fitness function of PSO algorithm is designed considering path delay, expected path energy and path stability.

While handling real-time traffic like video streaming in multicasting, playback delay and jitter will occur. If more users are requesting a same video file, then they have to join the respective multicast group. If the users are joining at different time intervals, then the source has to send the entire data repeatedly for the users, which will increase the overhead and energy consumption of the source.

Hence as an extension to the previous works, we propose to develop efficient buffer management technique for streaming data in multicast groups.

2. Related Works

Muhammad Aamir et al [1] have introduced a new scheme of buffer management to handle packet queues in Mobile Ad hoc Networks (MANETs) for fixed and mobile nodes. In this scheme, they have tried to achieve efficient queuing in the buffer of a centrally communicating MANET node through an active queue management strategy by assigning dynamic buffer space to all neighboring nodes in proportion to the number of packets received from neighbors and hence controlling packet drop probabilities.

Nishu Garg et al [2] have proposed the new design for enhancing the performance of MANET by buffering the data and providing an alternate path for the packet to reach their destination. A number of neighbor-monitoring, trust-building, and cluster-based voting schemes have been proposed in their research to enable the detection and reporting of nasty activity in ad hoc networks. The resources consumed by ad hoc network member nodes to monitor, detect, report, and diagnose nasty activity, however, may be greater than simply rerouting packets through a different available path.

Tariq Alahdal et al [3] have proposed two algorithms to improve the performance of the source tree reliable multicast (STRM) protocol. The first algorithm was developed to avoid buffer overflow in the sender node as the forward server (FS) nodes of STRM. This reduction is achieved by managing the buffer of the FS nodes, i.e., selecting the FS nodes depending on its empty buffer size and reducing the feedback sent from the receiver nodes to their FS node. The second algorithm was developed to decrease duplicated packets in the multicast members of the local group, which may be achieved by sending the repair packets to the requesting member. The FS in the local group should create a dynamic and temporary subgroup whose members are only those that requested the repair packet retransmission.

Wei Kuang Lai et al [4] have proposed a hop-aware and energy-based buffer management scheme (HEB).HEB can provide better quality of service to packets with real-time requirements and improve MANET power efficiency. In their algorithm, the buffer is divided into real-time and non-real-time partitions. They have consider the number of hops passed, the power levels of the transmitting node, the predicted number of remaining hops, and waiting time in the buffer to determine packet transmission priority. In addition, specialized queue management and a probabilistic scheduling algorithm were proposed to decrease retransmissions caused by packet losses.

Jani Lakkakorpi et al [5] have proposed a mechanism that advertises buffer occupancy information to adjacent nodes and avoids forwarding through nodes with high buffer occupancy. The nodes then achieve global congestion avoidance simply based on locally available information. The proposed mechanism works independent of the routing protocol and is thus applicable to wide array of scenarios.

3. Efficient Buffer Management Protocol

3.1. Overview

In this paper, we propose to develop efficient buffer management technique for streaming data in multicast groups. The frequently requested video data can be buffered in any intermediate nodes along the multicast tree from the source to the receivers. The buffer space is divided into two partitions for real-time and non-real-time traffic. When packets are received, they are classified into real-time packets (video or audio packets) or non-real-time packets and placed into respective partitions [4]. Then cumulative weight of the packets arrived in the real-time buffer is then estimated based on number of hops, deadline and waiting time. Based on the estimated weight value, transmission priorities are assigned. The source is notified before the buffer is to be completely filled, so that it can stop sending data or lower the rate of data transmission [1]. The buffer space is dynamically adjusted depending on the number of multicast receivers, requesting data.

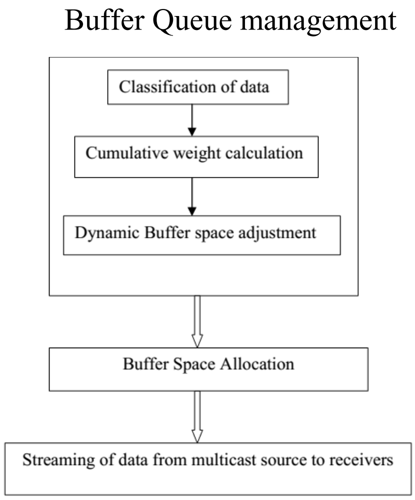

Fig. 1: Block Diagram

3.2. Classification of data

At every intermediate node in the multicast tree, a buffer is created and the buffer space is divided into real time and non real time buffer space. When a data packet arrives at an intermediate node, is buffered and divided into real time and non real time traffic and placed in the respective buffer space [4].

The classification of the buffer data is described in algorithm 1

Algorithm 1

1. Initially when a multicast tree is developed, at every intermediate node, a buffer is created.
2. The buffer space is divided into two divisions: real time (RT) and non real time (NRT) buffer space.
3. When an incoming data arrives at an intermediate node, the classifier classifies into either real time or non real time traffic.
4. The buffer space is adjusted by the Queue size manager by borrow or push out technique.
5. When the RT buffer space is full and there exists free space in the NRT buffer space, then the RT buffer space temporarily borrows some free space from the NRT buffer space so as to avoid discarding the incoming RT traffic.
6. When the entire buffer is full, then the push out process is followed when a new data arrives.
7. The weighted value of the incoming data packets is determined by the priority manager and accordingly the data priority is assigned. The calculation of the cumulative weight is explained in Section 3.3.
8. When the partition is full and if a new packet arrives for RT, then as the used space for NRT is still larger than the threshold, NRT will push out lower-priority packets at the tail, thus generating space for RT to borrow.

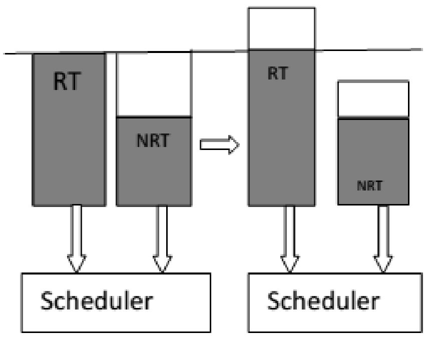

Fig.2 : Borrow Technique

. When the partition is full and the weighted value of the incoming packet is greater than at least one packet within the partition, the new packet will enter the partition, whereas the one with the lowest probability will be pushed out and discarded.

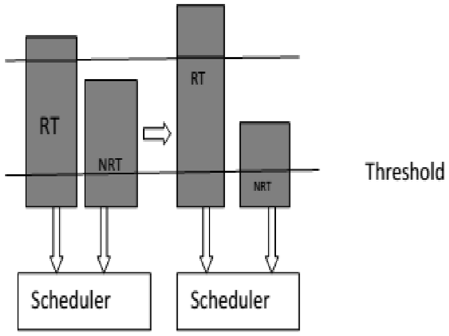

Fig.3: Push Out Technique

10. When the partition is full and if the weighted value of the new packet is smaller than the minimum weighted value within the partition, the new packet will be discarded.

Thus, the incoming data traffic is classified as RT or NRT traffic and assigned to the corresponding buffer space.

3.3. Cumulative Weight Calculation

The cumulative weight of the packets arrived in the real-time buffer is then estimated based on number of hops (H), deadline (D) and waiting time (WT) [3] within the multicast tree.
Let the cumulative weight be represented by *CW,* and it is estimated according to the equation (4)

$$CW = w_1.H + w_2.D + w_3.WT \qquad (1)$$

where w_1, w_2 and w_3 are the are the weight factors whose summation is equal to 1. Given $w_1 = 0.3, w_2 = 0.3$ and $w_3 = 0.4$.

3.4. Buffer Queue Management

When the buffer space is about to be filled, then the sender is informed that the buffer limit is about to be reached and is the determination of the buffer space is described [1] in algorithm 2.

Algorithm-2

1. The number of neighbors (nh) around each node j is determined.
2. The used buffer space for each neighbor of j is determined.
3. Then, the difference (Buf_D) between the pre defined buffer limit (Buf_L) and the used buffer space (Buf_U) is estimated .

$$Buf_D(i) = Buf_L(i) - Buf_U(i) \qquad (2)$$

Where i is the neighbor of node j, i=1,2...nh

4. Then the neighbors are arranged in the ascending order of Buf_D which is denoted as Ne(j).
5. Next the residual buffer space (rbs) which is the difference between the overall buffer space and the overall occupied buffer space is determined.

$$\text{rbs} = \sum_{i=1}^{n} Buf_L(i) - \sum_{i=1}^{n} Buf_U(i) \qquad (3)$$

6. The extended buffer space for node 1 of Ne(j) is determined as

$$\text{ebs}(1) = \frac{rbs}{\sum_{i=1}^{nh} i} * nh \qquad (4)$$

7. Then used buffer space for node 1 of Ne(j) can be extended by
   $Buf_U(1) = Buf_U(1) + ebs(1)$
8. Similarly, extended buffer space for node 2 of Ne(j) is determined as

$$ebs(2) = \frac{rbs}{\sum_{i=1}^{nh} i} * (nh-1) \qquad (5)$$

9. Used buffer space for node 2 of Ne(j) can be extended by
   $Buf_U(2) = Buf_U(2) + ebs(2)$
10. Similarly for the remaining nodes of Ne(j), the used buffer space can be extended.
11. This process is repeated for all the nodes in along the multicast route from source to receivers.

When the sender receives this notification, it either stops sending data or reduces the transmission rate so as to handle the situation. When the data accumulated in the buffer is processed and de queued, the buffer space becomes available. Then the sender node can again start transmitting the data to its destination. Thus, by following this procedure the data packets can be avoided from being discarded.

3.5. Streaming and Buffer Management

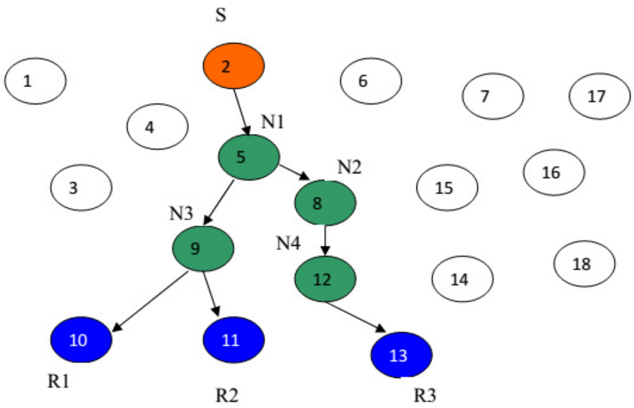

Figure 4 Streaming and buffer Management

Algorithm-3

In figure 4, S is the multicast source and R_1,R_2 and R_3 are the 3 multicast receivers. Let {Ni}, i=1,2…. m be the nodes along the path from S to R_1, R_2 and R_3. Let G be the total size of the requested data.

1. Let R_1 and R_2 join the multicast tree at time T_1 and request for RT data V_k from the source.
2. Source S starts transmitting the data V_k towards the receivers through the established multicast tree.
3. Buffer space is allocated to the nodes $\{N_i\}$ as described in Algorithm-2.
4. The traffic type is determined and the buffer size is adjusted as described in Algorithm-1
5. Let g amount of data is buffered in $\{N_i\}$ during the time T_2.
6. Let R_3 join the multicast tree at time T_3 and request for the same data V_k.
7. Then the buffered data of size g is transmitted from any $\{N_i\}$ to R3 at time T4.
8. The remaining data (G-g) is then transmitted from S to R_1, R_2 and R_3 after time T_4+x
9. If the buffer space is filled at all $\{N_i\}$, then a notification is sent to S.
10. S will temporarily stop the transmission for y seconds and resumes it once the buffer space is available at any $\{N_i\}$.

4. Simulation Results

4.1. Simulation Parameters

We use NS-2 [7] to simulate our proposed Efficient Buffer Management Protocol (EBMP). We use the IEEE 802.11 for MANET as the MAC layer protocol. It has the functionality to notify the network layer about link breakage. In our simulation, the packet size is varied as 250,500,750 and 1000. The area size is 1500 meter x 300 meter square region for 50 seconds simulation time. The simulated traffic is Video traffic.

Our simulation settings and parameters are summarized in table 1

Table 1: Simulation Parameters

No. of Nodes	50
Area	1500 X 300
MAC	802.11
Simulation Time	50 sec
Traffic Source	Video
Rate	150Kb
Propagation	TwoRayGround
Antenna	OmniAntenna
Initial Energy	10.1J
Transmission Power	0.660
Receiving Power	0.395
Packet Size	250,500,750 and 1000bytes
Receivers	10,20,30,40 and 50

4.2. Performance Metrics

We evaluate performance of the new protocol mainly according to the following parameters. We compare the Ordered Acknowledgement (OACK) [3] scheme with our proposed EBMP protocol.

Average Packet Delivery Ratio: It is the ratio of the number of packets received successfully and the total number of packets transmitted.

Average end-to-end delay: The end-to-end-delay is averaged over all surviving data packets from the sources to the destinations.

Packet Drop: It is the number of packets dropped during the data transmission

4.3. Results & Analysis

The simulation results are presented in the next section.

A. Based on Packet Size

In order to analyse the effect of buffer space adjustment, packet size of the video traffic is increased from 250 to 1000 bytes.

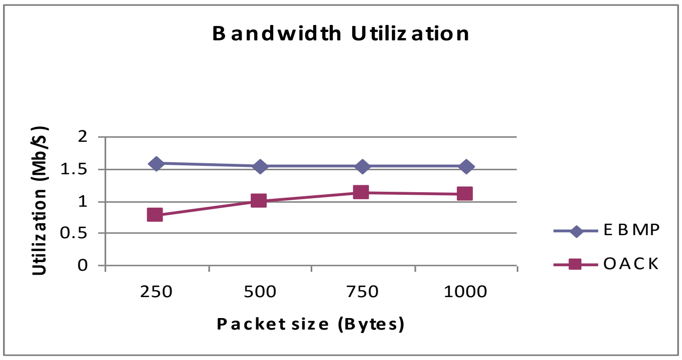

Fig 5: Psize Vs Bandwidth Utilization

Figure 5 shows the results of bandwidth utilization by varying the packet size from 250 to 1000 bytes for EBMP and OACK. Since the number of packets will be reduced when the packet size is increased, the bandwidth utilization slightly increases. Since EBMP efficiently allocates the bandwidths for prioritized traffic, it attains 35% increased bandwidth utilization when compared to OACK.

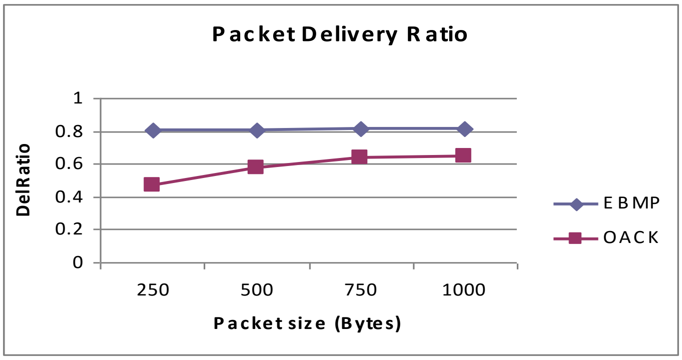

Fig 6: Psize VsDelivery Ratio

Figure 6 shows the results of packet delivery ratio by varying the packet size from 250 to 1000 bytes for EBMP and OACK. Since the number of packets will be reduced when the packet size is increased, the delivery ratio slightly increases. Since EBMP avoids buffer overflow by sending warning messages, it has 28% increased delivery ratio when compared to OACK.

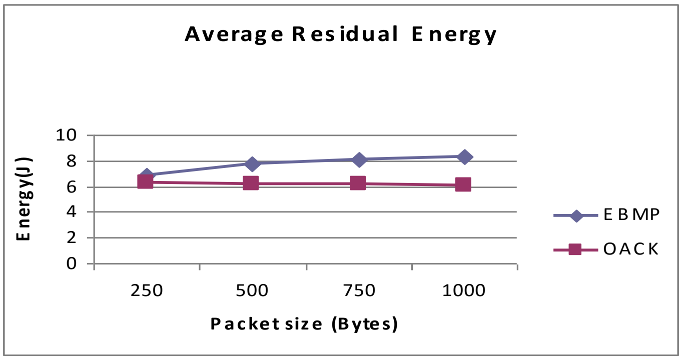

Fig 7: Psize Vs Residual Energy

Figure 7 shows the results of average residual energy by varying the packet size from 250 to 1000 bytes for EBMP and OACK. Since the number of packets will be reduced when the packet size is increased, the residual energy slightly increases. Since EBMP consists of energy efficient multicast tree, the residual energy is 19% higher than OACK.

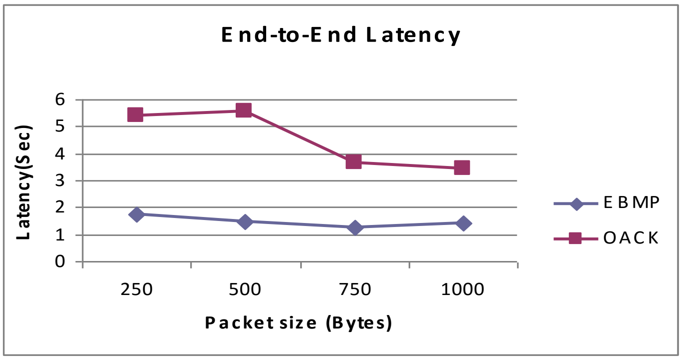

Fig 8: Psize Vs Latency

Figure 8 shows the results of latency by varying the packet size from 250 to 1000 bytes for EBMP and OACK. Since the number of packets will be reduced when the packet size is increased, the latency also slightly decreases. Since EBMP has proactive caching and streaming mechanism, it has 66% reduced latency when compared to OACK.

B. Based on Receivers

The number of multicast receivers is increased from 10 to 50 for packet size of 500 bytes.

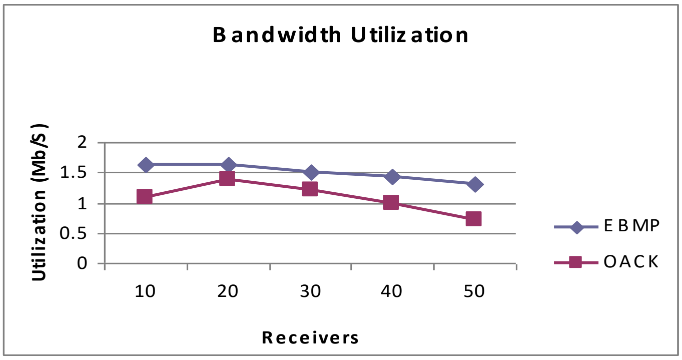

Fig 9: Receivers Vs Bandwidth Utilization

Figure 9 shows the results of bandwidth utilization by varying the number of receivers from 10 to 50 for EBMP and OACK. Since the volume of traffic will be increased when the number of receivers is increased, the bandwidth utilization slightly decreases. Since EBMP efficiently allocates the bandwidths for prioritized traffic, it attains 28% increased bandwidth utilization when compared to OACK.

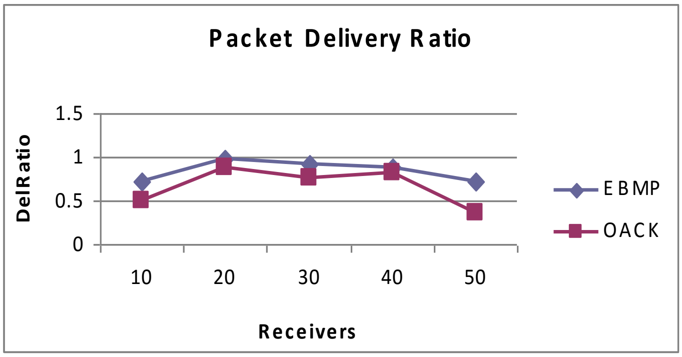

Fig 10: Receivers Vs Delivery Ratio

Figure 10 shows the results of packet delivery ratio by varying the number of receivers from 10 to 50 for EBMP and OACK. Since the volume of traffic will be increased when the packet size is increased, the delivery ratio slightly decreases. Since EBMP avoids buffer overflow by sending warning messages, it has 23% increased delivery ratio when compared to OACK.

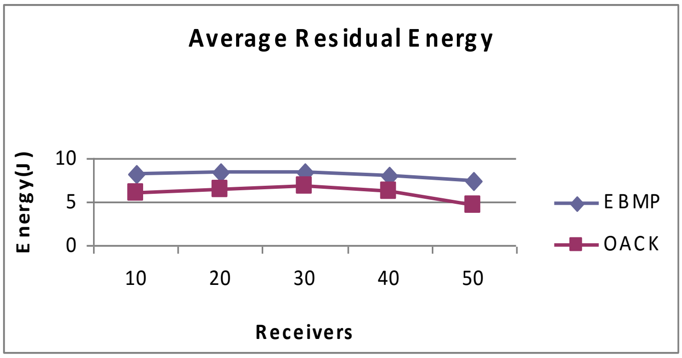

Fig 11: Receivers Vs Residual Energy

Figure 11 shows the results of average residual energy by varying the number of receivers from 10 to 50 for EBMP and OACK. Since the volume of traffic will be increased when the packet size is increased, the residual energy slightly decreases. Since EBMP consists of energy efficient multicast tree, the residual energy is 26% higher than OACK.

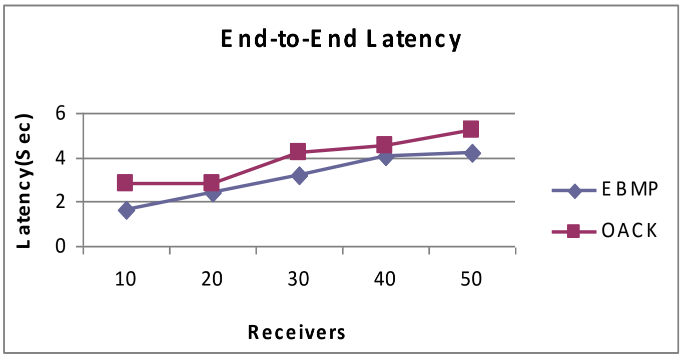

Fig 12: Receivers Vs Latency

Figure 12 shows the results of latency by varying the number of receivers from 10 to 50 for EBMP and OACK.

Since the volume of traffic will be increased when the number of receivers is increased, the latency slightly increases. Since EBMP has proactive caching and streaming mechanism, it has 22% reduced latency when compared to OACK.

5. Conclusion

In this paper, an efficient buffer management protocol is developed for streaming data in multicast groups. The frequently requested video data are buffered in intermediate nodes along the multicast tree from the source to the receivers. Cumulative weight of the packets in the real-time buffer is then estimated based on number of hops, deadline and waiting time. Based on the estimated weight value, transmission priorities are assigned. The buffer space is dynamically adjusted depending on the number of intermediate nodes along the multicast tree from the source to the receivers. The source is notified before the buffer is to be completely filled, so that it can stop sending data or lower the rate of data transmission. Simulation results show that the proposed buffer management protocol reduces the latency and energy consumption while increasing the packet delivery ratio.

References

1.Muhammad Aamir and Mustafa A. Zaidi,"A Buffer Management Scheme for Packet Queues in MANET",TSINGHUA SCIENCE AND TECHNOLOGY,ISSN ll 1007-0214 ll 01/10 ll pp543-553,Volume 18, Number 6, December 2013.

2.Nishu Garg,Ruchika Sharma and Parul Pal, "Packet buffering in Manet",IJCSNS International Journal of Computer Science and Network Security, VOL.11 No.6, June 2011.

3.Tariq Alahdal,Raed Alsaqour, Maha Abdelhaq, Rashid Saeed, and Ola Alsaqour, "Reliable Buffering Management Algorithm Support for Multicast Protocol in Mobile Ad-hoc Networks",Journal of Communications, Vol. 8, No.2, February 2013.

4.Wei Kuang Lai, Mu-Lung Weng and Yo-Ho Lin, "Improving MANET performance by a hop-aware and energy-based buffer management scheme",Wirel. Commun. Mob. Comput., 2012 .

5.Jani Lakkakorpi, Mikko Pitk anen and J org Ott, "Using Buffer Space Advertisements to Avoid Congestion in Mobile Opportunistic DTNs",WWIC'11 Proceedings of the 9th IFIP TC 6 international conference on Wired/wireless internet communications-Springer,2011.

6. Kavitha Subramaniam and Latha Tamilselvan," Predictive Energy Efficient and Reliable Multicast Routing In MANET", Research Journal of Applied Sciences, Engineering and Technology 9(9): 706-714, 2015

7. Network Simulator:http:///www.isi.edu/nsnam/ns