Smart Bandwidth Friendly Buffer: Handling Overflow in Wireless Mesh Networks

Abdul Razaque[1], Ahmed Haroon[2], Manoj Kumar[3], Gullapalli Amulya[4], Tariq Rahim Soomro[5]

[1]*Department of Computer Science, New York Institute of Technology, Nanjing Campus, China.*
[2,3,4] *Department of Computer Science and Electrical Engineering, Cleveland State University, OH, USA.*
[5]*Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Dubai, UAE.*

[1]arazaque@nyit.edu
[2]a.fnu18@vikes.csuohio.edu
[3]m.peddarapu@vikes.csuohio.edu
[4]a.gullapalli@vikes.csuohio.edu
[5]tariq@szabist.ac.ae

Abstract—With breakthrough of technological advancement, the significance of data transmission has been in highly demanding. On the other hand, limited buffering capacity has been great challenge that limits the Quality of Service (QoS) and degrade the performance of the network particularly in Wireless Mesh Networks (WMNs). Thus, it is important to provide an efficient utilization bandwidth and buffer management. Furthermore, the QoS in WMNs depends immensely on intelligent buffer management to avoid unexpected congestion and data loss. Some algorithms have been introduced to improve the buffer capacity and management. However, these suffer from high latency, even the loss of data because of congestion in the buffers, eventually resulting in low throughput. To address these issues we introduce the scheme "Smart Bandwidth Friendly Buffers (SBFB) for Wireless Mesh Networks. The SBFB is inspired by the features of existing schemes like the EZ Flow, WRED (Weighted Random Early Detection, and Back-off mechanism algorithms. The SBFB scheme contributes to prioritize the packets, allowing the down link nodes to perceive the buffer capacity prior to transmission and it also hunts for alternative routes in case of buffer buildup. Our proposed algorithm is validated using Network simulator-3 (NS3). Based on the experimental results, we have determined that SBFB has lesser congestion and packet loss probability when compared to other known contemporary schemes.

Keywords—Wireless Mesh Networks (WMNs); Smart Buffers; Intelligent Buffers; EZ flow; WRED; Back-off; Buffer Overflow.

I. INTRODUCTION

Wireless mesh networks have emerged to fulfill the increasing demand for better network services which have attracted more and more attention. WMNs are providing the high-speed internet services to customers at low cost [1]. They are also used in extreme emergency situations in danger areas, battlefield surveillance and real-time racing-car telemetry. WMNs essentially can also be used in VoIP, in which the QoS scheme may support local telephone calls to be routed through the mesh. U.S. military forces use WMNs for communication in field operations. Residences now use electric meters to record their readings and transfer to the billing office without any human meter readers or wired connections [2]. Also, satellites today use WMNs to transfer calls between satellite phones without having to use the earth station [3]. To guarantee a scalable network, the WMNs have to be perceptive high-speed networks providing and maintaining exceptionally high throughput, with minimum latency, for at least 12 or more hops.

This inevitably increases possibilities to deploy very large-scale WMNs in many other prospect areas. In this context, the packet scheduling techniques, routing protocols installed in the wireless nodes and buffer prescience for loss avoidance by adapting the transmission rate in case of high traffic and precluding congestion before happening are critical points that impact the effective performance of the network. There are also other factors like the buffer overflow which happens when the buffer gets full and drops the incoming packets, resulting in retransmission and increased delay. This increased delay, termed as buffer bloat, is observed even when the buffer capacity is increased. These are the main factors that affect buffers and hamper the widespread adoption of WMNs, proving a challenge for providing the smooth and efficient traffic overflow over the backhaul network [4].

When it comes to larger networks with more than single digit hops then the stability in Wireless Mesh Networks suffers a lot, as is the case even with networks with 3 to 4

hops [5]. Because of this the IEEE 802.11 networks today use very small number of hops. Also, large networks may have several source and destination nodes sending different types of traffic, which has to be prioritized, in order to let the network, have a smooth flow of traffic.

To address these factors, SBFB has used the Weighted Random Early Detection (WRED) Mechanism to prioritize packets in terms of IP precedence, the Back-off mechanism to send exponential back-off signals to transmitting relay nodes when the buffers have any chance of congestion, and a re-routing mechanism which signals the desired up-link nodes to shift to an alternate route on observing precursors of congestion.

The remainder of the paper is organized as follows, Section II discusses problem identification and significance. Section III gives and overview of existing approaches. Section IV presents the Smart Bandwidth Friendly Buffer, Section V gives the simulation setup and experimental results, and the entire paper is concluded in Section VI.

II. PROBLEM AND SIGNIFICANCE

Wireless mesh networks presently face the problem of low Quality of Service due to Buffer Mismanagement [6-8]. Buffer overflow could result the loss of important information. It could lead to extended delay due to repeated retransmission, packet collisions in channels, ineffective and bandwidth utilization [9]. It is also observed that increasing the size of the buffer to store more packets causes buffer bloat, thus causing extra delay. Avoiding data loss with reducing the latency remains a challenge, due to which it is difficult to achieve sufficient standards in data transmission and speed of data transmission decreases. To address this problem of Quality of service in WMNs, WRED algorithm classifies the incoming packets based on priorities and sets thresholds for each priority. The lowest priority packets start dropping before even the buffer is even half full, and the highest priority packet's drop only when the buffer is fully occupied. This makes sure that less important packets do not jam the buffer and more priority is given to more crucial data. In this case the dropped packets are resent, utilizing the bandwidth of network with no assurance that the packets will not be dropped by the buffer again. To address this problem, back off mechanism uses the concept of Jamming signal to pause the data transmission on encountering congestion.

In case the buffer stays congested, the Jamming (Back off) time is increased exponentially and a lot of time is wasted by the transmitter in waiting. Fortunately, EZ flow estimates the successor nodes' buffer occupancy and sends data only when the buffer is free for relaying packets, but it doesn't focus on multiple class of data, and there is no mechanism to reroute the data once the buffer starts to get congested. Congestion in EZ flow can occur in case of a large network. For congestion control, a node is not always supposed to know which successor (i.e., which neighbor relay) gets its packets. In fact, it is enough to store a minimum value of the total number of packets that are in line to be relayed at all of its successor nodes.

III. RELATED WORK

In this section, salient features of existing approaches are discussed. In [1], Aziz, *et al*. introduced new mechanism EZ-flow for data transmission in WMNs. It aims to change the minimum congestion window size according to estimated buffer occupancy of successive node. It uses broadcast signals that help detect the buffer occupancies. The approach improved the throughput performance, but focused on a line topology and few neighbor nodes. Furthermore, different data classes such as voice, video, background and Best effort are all queued without considering any priority, unlike the WRED algorithm.

In [10], Olesinski, *et al*. discussed the analysis of WRED while considering the network with transport oriented protocols. Authors used WRED congestion control mechanism for transport oriented protocols to choose the packets according to their priorities and assigned discard thresholds to the traffic. The mechanism limited the UDP/TCP traffic to a predefined percentage of the buffer's available bandwidth.

Since the transport oriented protocol traffic experiences the congestion and UDP traffic is not responsive to congestion. The scheme helps solve the problem of monopolization of the buffer bandwidth through UDP traffic. It drops the UDP packets after the discard threshold is reached. However, there is still problem of packet loss due to buffer overflow that is not handled by this scheme.

In [11], Kesselman, *et al*. concentrated on QoS and buffer overflow by creating bounded-delay through greedy algorithm for First-In First-Out (FIFO). In this technique, the packets are transmitted in the same order in which order they have arrived. The greedy algorithm drops the earliest packets from the low-value packets. The algorithm showed approximately1.5%s better performance than the tail drop algorithm. The main advantage of the algorithm is to assign an intrinsic value to each packets that helps in deciding which packets to be dropped first. But scheme failed to achieve satisfactory bandwidth utilization and throughput performance.

Kadhim in [12], have proposed a back off algorithm to improve the QoS Provisioning. The algorithm focused on the

congestion window to avoid unnecessary idle time that leads to redundant delay in the network. When the buffer of node is full, then it sends a back-off signal to the downlink node to stop the data transmission for a predefined period. Later it can resume the transmission again. This helps reduce the repeated loss of packets and collision inside the channel. The channel has a probability to remain an idle for longer time for exponential back-off.

In [13], shin *et al*., proposed an algorithm to achieve the stability. The node uses its own buffer for scheduling decision, but it requires large size of buffer. Large buffer size has two disadvantages: extended end-to-end delay and hardware compatibility.

In [14], Katti *et al*. proposed relay-node-listening procedure. In this approach, the relay nodes listen to the packet-arrival and drop that are not destined for them. This approach helps the increase the channel capacity. However, the nodes spend additional power consumption.

In [15], Biswas *et al*. present a routing mechanism named ExOR that takes an advantage of the broadcast nature to achieve cooperative diversity. As a result, it increases the throughput performance. To handle the congestion control, a node does not always precisely need to know which successor (i.e., which next-hop relay) gets its packets. It just requires to use the packet with lower values from the total number of packets that are waiting to be forwarded to its successors. A similar extension of a congestion-control from unicast to multicast is discussed by Scheuermann *et al*. in [16]. All existing approaches focus on improving the buffer forwarding capability, but our proposed approach reduces the congestion and improves the Quality of Service (QoS) e.g. throughput, bandwidth consumption and latency.

IV. SMART BANDWIDTH FRIENDLY BUFFER

The Smart Bandwidth Friendly Buffer (SBFB) scheme utilizes the information of the successor node's buffer vacancies to start packet transmission. The nodes capacity is determined with the help of message passing. When the transmission starts, the packets are classified on the basis of IP precedence into High priority, Medium priority and Low priority packets. The buffers have a set maximum threshold, for the respective priorities. The packet transmission is interrupted when the respective threshold is reached. This systematic behavior of SBFB to eradicate the problem of packet dropping and congestion is explained in subsections.

A. Schematic Representation of SBFB

SBFB has only two buffers per node, named Primary Buffer and Backup Buffer. The primary buffer processes all the priorities by default. However, when the primary buffer's upper threshold is reached the data is processed by the Backup buffer. Eventually, if the Backup buffer has chances of congestion i.e. if the upper threshold is reached, the node sends a Back-off signal to the sender and the rerouting procedure is also initiated by the rerouting signal. Table 1 contains the list of notations used.

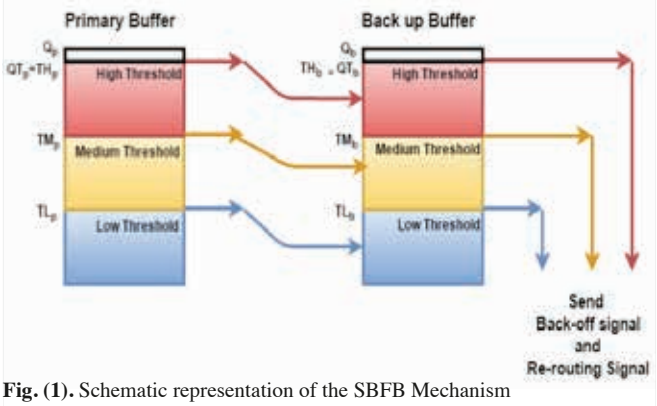

Fig. (1). Schematic representation of the SBFB Mechanism

The parameters used in Figures 1, 2 and 3 and Algorithms 1, 2 and 3 are shown in the table 1. The working process of Smart Bandwidth Friendly Buffer is depicted in Figure 1. The back-off signal is used to pause data transmission for a period of time and the rerouting signal signals the sender node to choose an alternate path for data transmission.

When an alternate path is available for packet transmission, the immediate successor node is checked for vacancy first and then the data transmission is started.

The classification of packets into priorities, estimating the number of packets, sending the back-off and the rerouting signal is done by the receiver. On the other hand, the time for pause on reception of the back off signal and the selection of an alternate route is decided by the transmitter. Thus the scheme can be divided into two parts discussed in section B and C.

Table 1. Notations and Description

Notations	Description and Definition
Notations	Description and Definition
N_p	Total packets in Primary Buffer
N_b	Total packets in Backup Buffer
H_p, H_b	High Priority Packets in Primary and Backup Buffer
L_p, L_b	Low Priority Packets in Primary and Backup Buffer
M_p, M_b	Medium Priority Packets in Primary and Backup Buffer

TP	Total Packets in Primary and Backup Buffer
TH_p	Threshold of High Priority packets in Primary Buffer
TH_n	Threshold of High Priority Packets in Backup Buffer
TM_p	Threshold of Medium Priority Packets in Primary Buffer
TM_b	Threshold of Medium Priority packets in Backup Buffer
TL_p	Threshold of Low Priority packets in Primary Buffer
TL_b	Threshold of Low Priority packets in Backup Buffer
JS	Jamming signal
RRS	Rerouting Signal
QT_p	Queue threshold of Primary Buffer
QTb	Queue threshold of Backup Buffer

(i) The mechanism at the source or relay-transmitter side (Transmitter side Mechanism) and (ii) The mechanism at the relay-receiver or receiver Side (Receiver side Mechanism).

B. The Receiver Side Mechanism of SBFB

In the receiver side mechanism, the state of the node is signaled by the receiver to the sender. If the node state is signaled as free, the data transmission starts. When the incoming packets are received, which are classified into priorities. Figure 2 below depicts the entire mechanism at the receiver side. Additionally, the priority packets have different set thresholds for each class.

The data transmission is interrupted when this threshold is reached. For example, the high priority packets have the highest thresholds, such that they are interrupted only when there is almost no possibility to store any more packets in the buffer. But the lower priority packets are interrupted far earlier. This gives a higher chance for processing higher priority packets. This system of assigning thresholds to different priority packets.

Algorithm 1: Classification of packets on the basis of IP Precedence into High, Medium and Low Priorities

1) Assign
   $N_p = H_p + M_p + L_p$
   $N_b = H_b + Mb + L_b$
   $T.P = N_p + N_s$
2) Calculate N_p, N_b, T.P
3) If $TP < QT_p + QT_s$ then
4) Send signal "node is free"
5) Else Send signal "node is busy" and check condition again
6) End If
7) End else
8) If IP Precedence = 0, 1, 2 then
9) Set Priority ← High priority
10) if IP Precedence = 3, 4, 5 then
11) Set Priority ← Medium priority
12) Else if Set IP Precedence = 6, 7 then
13) Set Priority ← Low priority
14) End if
15) End if
16) End else

1) Node State Signaling and Packet Prioritization

This section discusses the buffer capacity information requested by the sender node from the receiver. Upon obtaining the request, any relay or receiving node, broadcasts the status of its buffer occupancy to the sender by keeping track of the number of packets in the buffers. Furthermore, the incoming packets received by the node are automatically classified into high, medium and low priorities inside the buffer. This classification is done on the basis of IP Precedence.

The conditions for which the buffer is considered free or busy and the conditions for Packet prioritization are discussed in Algorithm 1.

In step 1, values are assigned. The number of packets in the primary buffer and the secondary buffer are calculated in step 2. From steps 3-4, total number of packets in primary buffer and backup are checked and signal of free or busy is sent. If buffer has a capacity, then free signal is sent otherwise busy. Furthermore, the total number of packets in the node relative to the queue thresholds are determined. QT_p and QT_b are set such that packets have less of a chance to experience packet drop due to no buffer vacancy. This idea is further discussed in algorithm 2.

In addition, the signaling is done when the sender node requests for the status of the node. If the node is free, then the transmission is initiated, but if the node is busy then another path is selected. If other paths are unavailable, the status is checked periodically and another route is determined simultaneously.

The classification of IP packets in the order of their precedence is explained in steps 08 through 13. If the packets are of precedence 0, 1 or 2 then they are classified as high priority and if the packets are of precedence 3, 4 or 5 then they are classified as medium priority and the 6 and 7 precedence packets are classified as low priority packets.

2) Distinct Threshold Assignment to distinct packet class

In this segment the threshold assignment procedure is discussed in detail. The received packets by the node use the entire buffer when the buffer has only one priority of packets. But when the buffer processes different types of packets at the same time, the threshold assignment is different. For example, the threshold of high priority packets is set to QT, where QT is the queue threshold of the buffer.

This means that the high priority packets experience interruption only when the entire buffer is full. The threshold of medium priority packets is set to 2Q/3 which allows these packets precedence over the low priority packets whose threshold is set to Q/3 typically.

Algorithm 2 illustrated below provides a detailed explanation on how the low priority packets are handled by the buffer.

Algorithm 2: Threshold calculation and Buffer Activation for Low priority packets

```
1)  Read    *\ Packet is Low priority
2)  If   Hp +Mp = 0 then
3)  Assign TLp = QTp;
4)  If   Hp +MP > 0 then
5)  Assign     TLb = QTb/3
6)  End if
7)  End if
8)  If   Hb +Mb = 0 then
9)  Set   TLb = QTb
10) if    Hb +Mb > 0 then
11) Set   TLp = QTp/3
12) End if
13) End if
14) Read    *\Active buffer is Primary Buffer
15) If   QTp > Np  && TLp > Lp  then
16) Send packet to Active Buffer
17) Else If   QTb > Nb && TLb > Lb   then
18) Send packet to inactive buffer
19) Active Buffer ← Back-up buffer
20) End If
21) End else
```

In step 3, if the buffer is empty then the threshold of low priority packets is set to Q, otherwise the threshold is set to Q/3 which is the default threshold of the low priority packets. In step 7, the concept of buffer activation is discussed. If the primary buffer is empty then it is the Active buffer for the low priority packets, but if it is full then the condition for backup buffer is checked after step 9.

In step 4, if the backup buffer is free then the backup buffer is considered as the Active buffer. The idea behind this approach of activating the buffers is to make a default buffer for the respective priorities. Now all the low priority packets are directed to the Active buffer for the low priority packets and the high and medium packets are directed to their respective active buffer likewise.

The above Algorithm 2 describes the case only for low priority packets. A similar approach is taken for other priority packets also. The only difference is in the setting of thresholds. For example, high priority packets have a threshold of Q irrespective of the buffers vacancy or occupancy. For medium priority packets the threshold is set to Q if the buffer contains only low priority packets or only medium priority packets. Otherwise the threshold is set to 2Q/3.

C. The Transmitter Side Mechanism of SBFB

The transmitter node can be a source node or a relay node that is transmitting data to its downlink, which is addressed as the transmitter node. After the calculation of a path from source to the destination, each sender node deploys the same procedure.

The transmitter node first sends a signal to the receiver demanding its buffer occupancy. If the receiver acknowledges that the node is free for relaying data, the transmitter starts packet transmission. Otherwise, the sender waits for a back off time (Tb) to check the status of the receiver again, and concurrently searches for an alternate path. The Pictorial representation of the operation on the sender side is depicted in Figure 3.

Once the transmission has started, if a Jamming signal (JS) is detected from the receiver, the node waits for a back-off time [6] each time before checking the status of the node again. If the node is free, the transmission is resumed. Or else, the back off time is increased exponentially. As it is noticeable the sender has to wait for Tb at two instances:

(i) When the sender finds the receiver busy and
(ii) When JS is detected. (During transmission)

The jamming signal is detected only during transmission. So, it is then only that the value of 'k' is incremented. Otherwise, for the case (i), the value of 'k' remains constant, i.e. 1 so the transmitter waits for constant 'Tb¬ ' and requests the node state periodically. During these cases (i) and (ii), if a different route is discovered, the state of the node is requested and data transfer is initiated. To better understand the process, Algorithm 3 illustrates the Back off mechanism in SBFB.

Algorithm 3: The SBFB transmission pause at the sender node

```
1)  Initialize (Tb: Back off time, K: Back off constant = 1)
2)  Read                    *\ Jamming Signal
```

3) Assign $T_b = 2^k$
4) Calculate T_b
5) Delay Transmission *\ Delay time = Tb
6) If K < 15 then
7) K ← K+1
8) Else if K ≥ 15 then
9) Set K ← K
10) End If
11) End else
12) If "Node is free" then
13) Assign K ← 1
14) End if

In step 1, the parameters are initialized. The Back off constant is set to 1 initially. The sender node detects the jamming signal in step 2. The transmission is then paused for a calculated time (T_b microseconds) and K is then incremented. If the value of 'K' reaches 15, it remains constant thereafter. These points are discussed in steps 3 through 9.

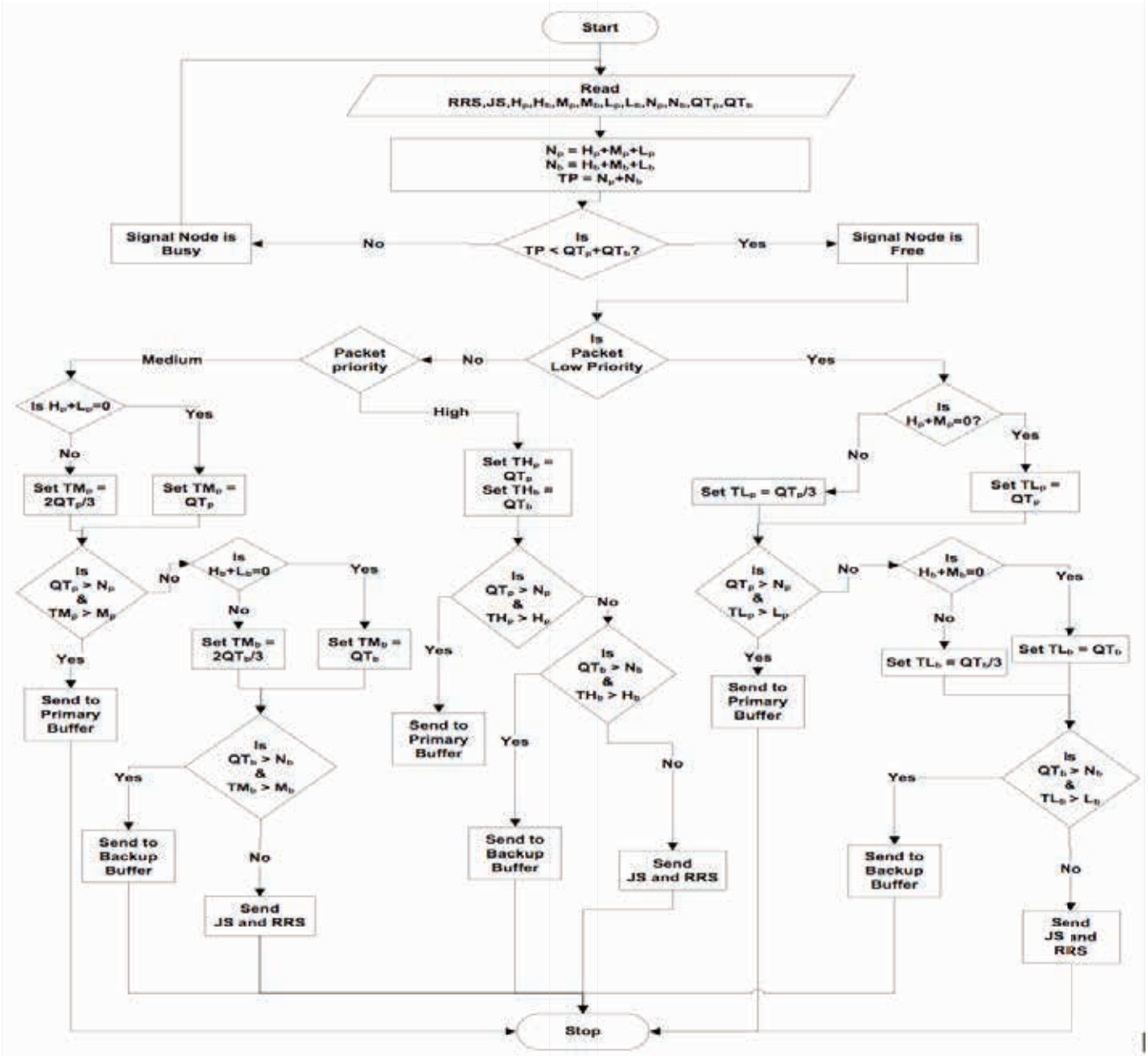

Fig. (2). Flowchart for SBFB Mechanism on the Receiver Side

Fig. (3). SBFB mechanism at the transmitter side

If the data transmission is resumed during this process, the value of K reaches 1 again. This is shown in step 13.

V. SIMULATION SETUP AND RESULTS

We simulate Smart Bandwidth Friendly Buffer using NS3. The Hybrid Wireless Mesh Protocol (HWMP) is deployed [17-20] as the routing protocol. The network contains 360 wireless nodes that are randomly deployed over the 1850 X 1700 square meter field. We generated number of 45 flows. We let the Buffer capacity to 85 packets at each node. The nodes are dynamically distributed in the network. When the simulation begins, the mobile wireless nodes move back and forth in the network regions. Each simulation run lasts for 40 minutes. As our primary goal, the performance of our proposed SBFB is determined and then compared with WRED, Back-off, and EZ Flow mechanisms. The simulation parameters are presented in Table 2.

Table 2. Simulation Parameters with description.

Parameters	Description
Transmission Range	250 meters
Sensing Range	550 meters
Bandwidth of node	360 Kb/Sec
Simulation time	40 minutes
Number of Nodes	360
Network Size	1850 X 1700 m2
Buffering capacity	85 Packets at each node
Node Speed	0 m/sec to 15 m/sec
Data Packet size	1024 bytes
Initial pause time	20 Seconds
Number of hops in network	18 Maximum
Number of Flows	45
Pause time	30 Seconds
Routing Protocol	Hybrid Wireless Mesh Protocol

The networks transmission range is set to 120 meters. The node speed is set to a maximum of 15 m/sec and the maximum number of hops is 22. There are 120 maximum destination nodes, which participate in the event and maximum of 42 sources. We obtain several results, and use the following metrics to demonstrate the performance of the Smart Bandwidth Friendly Buffer in multi-hop Wireless Mesh Networks:

- Network Congestion Probability
- Packet loss Probability
- Average Throughput VS Number of hops

A. Network Congestion Probability

The result shows when the number of nodes increase from 40 to 360, the congestion probability of the network increases for all the schemes. For the Back-off mechanism, the congestion probability is the highest for up to 200 nodes. Then it decreases as the EZ flow reaches highest probability. The SBFB maintains a low probability throughout. A discernable difference can be seen for the maximum number of nodes where the congestion probability of EZ flow is 1.8% and for SBFB it is less than 0.8%. Hence, we observed an impressive decrease in congestion using SBFB.

The Network congestion probability in the network is determined by calculating the ratio of number of packets sent to the number of packets undergoing congestion in the network. The percentage is calculated considering the case of 100 packets. The simulation is carried out considering different number of nodes and plotting the congestion probability percentage values. Figure 4 shows the graphical representation of the network's congestion probability % with respect to different values of nodes participating in the network.

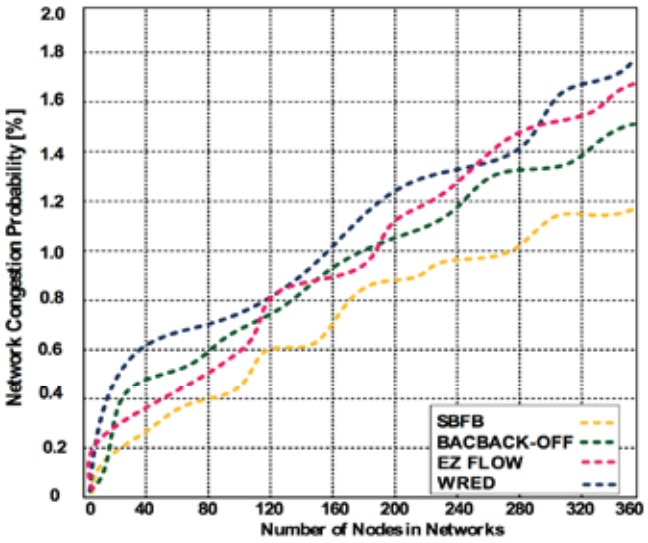

Fig. (4). Network Congestion Probability % versus Number of nodes in the Network

B. Packet Loss Probability

The second metric focusses on packet loss in the network. The total number of packets generated by the network to the number of packets lost are calculated for determining the packet loss. Packet loss can occur due to collision in the network or by buffer overflow. Through extensive simulation, the graph for packet loss probability % versus number of nodes is plotted, which is depicted in figure 5.

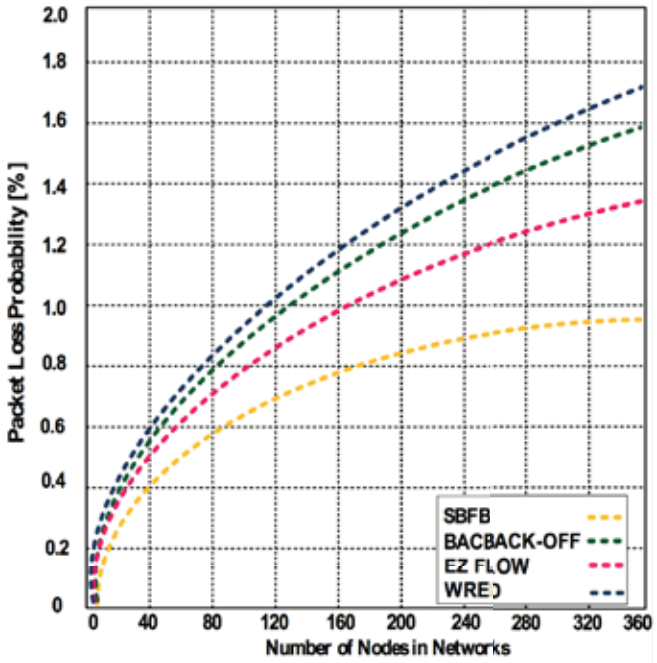

Fig. (5). Packet Loss Probability % versus Number of nodes in Network

It can be inferred from the graph that the packet loss probability of EZ Flow, Back-off mechanism and WRED starts form around 0.8% and follows a common trend, reaching around 1.8% for 360 nodes. Whereas for the SBFB mechanism a fine improvement in the packet loss avoidance can be seen for 360 nodes, where the packet loss probability is below 1.2%. This proves that SBFB mechanism has done a good job in sufficiently decreasing the packet loss probability in WMNs.

C. Average Throughput VS Number of Hops

Here, we designed multimedia supported scenario for sending and receiving the voice/video. We generated 45 flows simultaneously. When the number of hops increase, the throughput performance is degraded.

In Figure 6, an average throughput performance of depicted of SBFB and other competing bandwidth mechanism. Based on the experimental results, we observed that our proposed SBFB has hedge over other competing mechanisms. Our proposed SBFB reduces the throughput up to 4 hops and then it gives stable throughput. Whilst other mechanisms reduce the throughput up to 5 hops, but they further reduce the throughput. At the end of hops 18, our mechanism produced 1.33 Mb/sec, but other mechanisms have 1.0-1.15 Mb/sec. Based on the results, we observed that our mechanism produced 0.18-0.33 Mb/sec throughout performance.

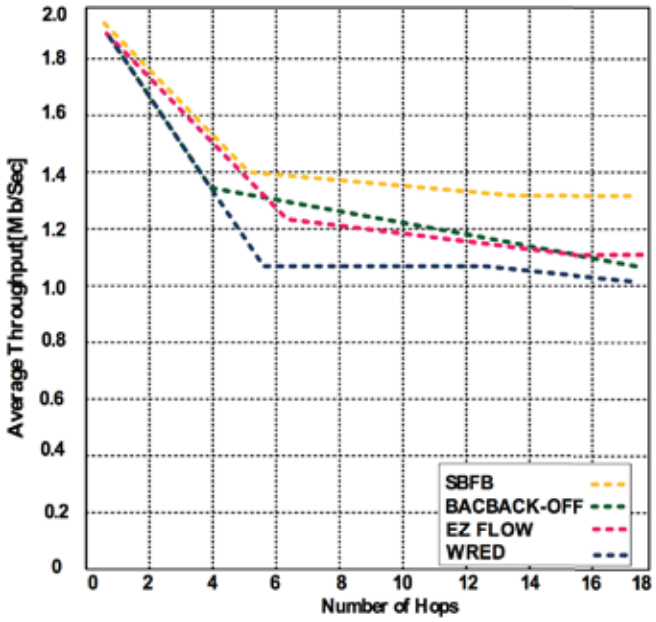

Fig. (6). Average throughput performance of SBFB, BACK-OFF, EZ FLOW and WRED on different number of hops

VI. CONCLUSION

The Smart Bandwidth Friendly Buffer is introduced for wireless mesh networks. The proposed algorithm determines the buffer vacancy of the successor node prior to packet transmission. The algorithm queues the packets based on the priority. Furthermore, it sends a rerouting signal, and ordering the packets in case of expected congestion on the network. The proposed SBFB is coded in C++ and converted the code into object tool command language (OTCL) and run on NS3. The experimental results confirm that the SBFB performs better with respect to processing different traffic loads, congestion avoidance and QoS provisioning. Furthermore, simulation results proved that SBFB outperforms other competing approaches of similar nature: EZ flow, WRED, and BACK-UP in terms of the network congestion probability, the packet loss probability and throughout performance. In future, we will explore other metrics such as latency, accuracy and bandwidth utilization in the small-to-large WMNs

REFERENCES

[1] A. Aziz, D. Starobinski, P. Thiran and A. El-Fawal. "EZ-Flow: Removing turbulence in IEEE 802.11 wireless mesh networks without message passing." In *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies,* 2009, pp: 73-84.

[2] O. Häggström. *Here Be Dragons: Science, Technology and the Future of Humanity*. Oxford University Press, 2015.

[3] M. M. G. Ghorale and A. O. Bang. "Wireless Ad-Hoc Networks: Types, Applications, Security Goals," *International Journal of Advent Research in Computer and Electronics,* vol. sp. Iss. National Conference "CONVERGENCE", 2015.

[4] A. Razaque and K. Elleithy. "Proportional Study of TCP Variants over Heterogeneous Wireless Networks," *International Journal of Computer Applications,* vol. 108, no. 20, pp: 24-31, 2014.

[5] T. Ali, L. T. Jung and I. Faye. "Classification of Routing Algorithms in Volatile Environment of Underwater Wireless Sensor Networks," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 6, no. 2, 2014.

[6] A. Showail, K. Jamshaid, and B. Shihada, "Buffer Sizing in Wireless Networks: Challenges, Solutions, and Opportunities", *IEEE Communication Magazine,* vol. 54, no. 4, pp: 130-137, 2016.

[7] K. Jamshaid, B. Shihada, A. Showail and P. Levis "Deffating Link Buffers in a Wireless Mesh Network," *Journal of Wireless Ad Hoc Networks,* vol. 16, pp. 266-280, 2014.

[8] J. Ye, J-X. Wang, J-W. Huang and T-S. Li, "Hop distance fairness for wireless mesh network based on queue management," *Journal of Central South University,* vol. 19, no. 10, pp: 2832–2838, 2012. doi:10.1007/s 11771-012-1349-y.

[9] M. Almasri, K. Elleithy and A. Razaque, "Analytical Study of Pre-Congestion Notification (PCN) Techniques," *International Journal of Computer Networks & Communications (IJCNC),* vol. 4, no. 4, 2012.

[10] W. Olesinski and S. Driediger. "Fair WRED for TCP UDP traffic mix." U.S. Patent 7,616,573. Nov. 2009.

[11] A. Kesselman, Z. Lotker, Y. Mansour, B. Patt-Shamir, B. Schieber and M. Sviridenko. "Buffer overflow management in QoS switches," *SIAM Journal on Computing,* vol. 33, no. 3, pp: 563-583, 2004.

[12] D. J. Kadhim, S. H. Abdulhussain, B. M. Ridha and A. M. Abbas. "A Balanced Backoff Algorithm for IEEE 802.11 Wireless Network," *Iraqi Journal of Applied Physics,* vol. 8, no. 1, 2012.

[13] J. Shin, D. Shah, and S. Rajagopalan. "Network Adiabatic Theorem: An efficient randomized protocol for contention resolution," In *Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems,* 2009, pp: 133-144.

[14] S. Katti, H. Rahul, H. Wenjun, D. Katabi, M. Medard and J. Crowcroft. "Xors in the air: Practical wireless network coding," *IEEE/ACM Transactions on Networking,* vol. 16, no. 3, pp: 497–510, 2008.

[15] S. Biswas and R. Morris. "Exor: opportunistic multi-hop routing for wireless networks," In *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM),* 2005, pp: 133-144.

[16] B. Scheuermann, M. Transier, C. Lochert, M. Mauve and W. Effelsberg. "Backpressure multicast congestion control in mobile ad-hoc networks," In *Proceedings of 2007 ACM CoNEXT Conference,* 2007, Art. no. 23.

[17] J. Ben-Othman and Y. I. S. Benitez, "IBC-HWMP: A novel secure identity-based cryptography-based scheme for Hybrid Wireless Mesh Protocol for IEEE 802.11s," *Concurrency and Computation: Practice and Experience,* vol. 25, pp: 686–700, 2013, doi:10.1002/cpe.1813.

[18] S. O. Cheikh, M. M. Hassan and A. Geuroui, "New Metric for HWMP Protocol (NMH)," *International Journal of Computer Networks & Communications,* vol. 5, no. 2, 2013.

[19] M. S. Islam, Y. J. Yoon, M. A. Hamid and C. S. Hong, " A Secure Hybrid Wireless Mesh Protocol for 802.11s Mesh Network," in *Computational Science and Its Applications – ICCSA 2008* (Lecture Notes in Computer Science), O. Gervasi, B. Murgante, A. Laganà, D. Taniar, Y. Mun, M. L. Gavrilova, Eds, Springer Berlin Heidelberg, 2008, pp: 972 - 985.

[20] J. Ye and K. A. Hua, "Scalability Study of Wireless Mesh Networks with Dynamic Stream Merging Capability," in *Multimedia Communications, Services and Security* (Communications in Computer and Information Science), A. Dziech, A. Czyżewski, Eds, Springer Berlin Heidelberg, 2011, pp: 324-330, DOI 10.1007/978-3-642-21512-4_39.