

# Resilient Packet Transmission (RPT) for the Buffer Based Routing (BBR) Protocol

Geetanjali Rathee\* and Nitin Rakesh\*\*

## Abstract

To provide effective communication in the wireless mesh network (WMN), several algorithms have been proposed. Since the possibilities of numerous failures always exist during communication, resiliency has been proven to be an important aspect for WMN to recover from these failures. In general, resiliency is the diligence of the reliability and availability in network. Several types of resiliency based routing algorithms have been proposed (i.e., Resilient Multicast, ROMER, etc.). Resilient Multicast establishes a two-node disjoint path and ROMER uses a credit-based approach to provide resiliency in the network. However, these proposed approaches have some disadvantages in terms of network throughput and network congestion. Previously, the buffer based routing (BBR) approach has been proposed to overcome these disadvantages. We proved earlier that BBR is more efficient in regards to w.r.t throughput, network performance, and reliability. In this paper, we consider the node/link failure issues and analogous performance of BBR. For these items we have proposed a resilient packet transmission (RPT) algorithm as a remedy for BBR during these types of failures. We also share the comparative performance analysis of previous approaches as compared to our proposed approach. Network throughput, network congestion, and resiliency against node/link failure are particular performance metrics that are examined over different sized WMNs.

## Keywords

BBR, Resiliency, Routing, RPT, WMN

## 1. Introduction

Today, network services (like email, the worldwide web, etc.) [1,2] have become a basic need in day-to-day communication. For providing these network services more effectively, wireless mesh network (WMN) [3-6] has turned into a popular topology that builds high performance infrastructures. Moreover, with the growth of network services, several types of network communication threats are coming into existence. To defend against these communication threats, resiliency [7-13] is a significant approach for WMN. Basically, resiliency is the capability to provide services in the face of failure. Resilient multicast routing [14] and ROMER [15] are the popular resilient routing algorithms for WMN. Both of these approaches have the problems of restricted network throughput, network congestion, and successful packet delivery in the face of node/link failures. We will now introduce both

※ This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Manuscript received August 26, 2013; first revision December 11, 2013; accepted January 17, 2014; onlinefirst October 20, 2014.

Corresponding Author: Nitin Rakesh ([nitin.rakesh@gmail.com](mailto:nitin.rakesh@gmail.com))

\* Dept. of Computer Science Engineering & Information Communication Technology, Jaypee University, Waknaghat 173215, India ([geetanjali.rathee123@gmail.com](mailto:geetanjali.rathee123@gmail.com))

\*\* Dept. of Computer Science and Engineering, Amity School of Engineering and Technology, Amity University Campus, Noida Uttar Pradesh 201313, India ([nitin.rakesh@gmail.com](mailto:nitin.rakesh@gmail.com))

of these approaches to classify these problems.

The Resilient Multicast routing protocol [14] establishes a two-node disjoint path to communicate between each  $[source, destination]$  pair. In the case of a node/link failure, traffic on unaffected path reaches the destination. This will increase network communication costs (reduces throughput). Another drawback of this approach is increased network congestion, due to a node/link failure, but multiple failures will further restrict resiliency in the network.

In ROMER [15] the source node forwards the packet by taking maximum credit cost. At each node the credit cost  $R$  and threshold value  $T$  are calculated to forward the packets from the source to the destination. The major drawback of ROMER, during the node or link failure the possibility of successful packet delivery to the destination nodes reduces based on the prescribed credit cost [16]. ROMER broadcasts the packets in the network and this results in increased network communication costs (reduces throughput) and increased network congestion, due to multiple packet delivery in the network.

In our previous work, we have proposed the buffer based routing (BBR) [16] approach, which provides an efficient solution (i.e., to route the data from source to destination) over the drawbacks of Resilient Multicast and ROMER. In BBR we allocated buffers at alternate nodes using the buffer allocation algorithm (BAA) with the aim to enhance resiliency and to increase the throughput and fast packet delivery in the network.

This paper is divided in four sections. In Section 1, we introduced the disadvantages of the Resilient Multicast and ROMER approaches. We also introduced how the BBR approach is more advantageous over these approaches. In Section 2, the BAA approach is analyzed for buffer allocation and further on in this section, the calculation of the minimum and maximum buffer size during communication from the source to the destination is analyzed. Furthermore, we have proposed a resilient packet transmission (RPT) algorithm for resilient packet transmission. The issues of time complexities during BAA and RPT are also discussed in this section. Performance evaluation and simulation results are shown in Section 3. Finally, Section 4 concludes the paper and the future scope of the work is discussed.

## 2. The Buffer Allocation Mechanism Using the BAA

### 2.1 The Buffer Allocation Mechanism Using the BAA

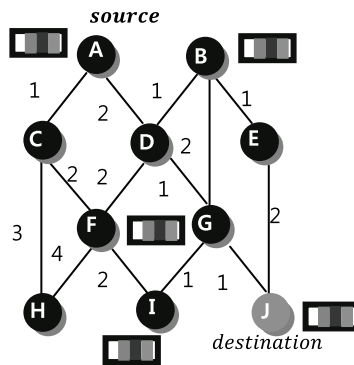
In this paper, by using an example of a network (shown in Fig. 1), we introduce an algorithm for RPT, which clarifies how packets are transmitted to the destination node, and we present its time complexities. The BBR [16] approach adopts a routing technique based on buffer allocation (i.e., we provide buffering at each node instead of maintaining the routing table). The BBR approach consists of three steps: 1) buffer allocation to the network nodes using BAA; 2) the selection of the optimum path for routing; and 3) RPT. The buffer allocation process in the network is discussed below.

#### 2.1.1. Buffer allocation to the network nodes

According to the BBR least cost path selection, buffers are placed at alternate positions in the

network. Buffer allocation is achieved in following steps:

- 1) Select a node  $N_i$  randomly from the network, as shown in Fig. 1, where  $i = 1, 2, 3, \dots, n$  number of nodes in this network. Assign a buffer to this node and mark it as a visited node.
- 2) Choose the least cost path from node  $N_i$  to its connected neighbouring node and move to this node, convert this current node as node  $N_i$ . As the buffer allocation process is assigned for alternate nodes. As such, skip buffer assignment to this node and just mark as visited.
- 3) The, choose the least cost path again from node  $N_i$  to its connected neighbouring node. Move to this node and make that node as  $N_i$ . Assign a buffer to this node and mark it as a visited node.
- 4) This step determines whether the next node is visited or not. If a node is visited then it rolls back to its previous node and another node must be searched for again.
- 5) Repeat steps  $a - d$  until total buffer placement in network is performed. The buffering process will stop when the total buffered nodes are  $\leq n/2$  where  $n$  is the number of nodes in the network.



**Fig. 1.** Buffer allocation in the network using buffer allocation algorithm (steps  $(a - e)$ ).

Buffers are provided to maintain the resiliency in the network. If we have a network with 10 nodes (Fig. 1), then the total number of buffers allocated in the network are 5. Similarly, in a network of 15 nodes, the number of buffers allocated in network are 7. If  $(n = \text{number of nodes in the network})$ , then the total buffer placement in the network is  $\leq n/2$ . The minimum and maximum size of the buffer is 5 units of packets (i.e., only 5 packets at a time are stored inside the buffer). To reduce the traffic congestion and increase the speed of data transfer, the minimum and maximum size of the buffer is set to 5 units. To understand this more clearly, let us use the following example: Fig. 1 shows a 10 node network in which source node A and intermediate node F are buffered nodes, let exist only one packet to transmit from source A to destination node J.

Step 1: Node A sends the first packet to node C, assuming that the transmission of the first packet from node A to C is 1 second.

Step 2: Node C sends an acknowledgement (ACK) of packet 1 to node A, assuming that the ACK transmission time is also 1 second.

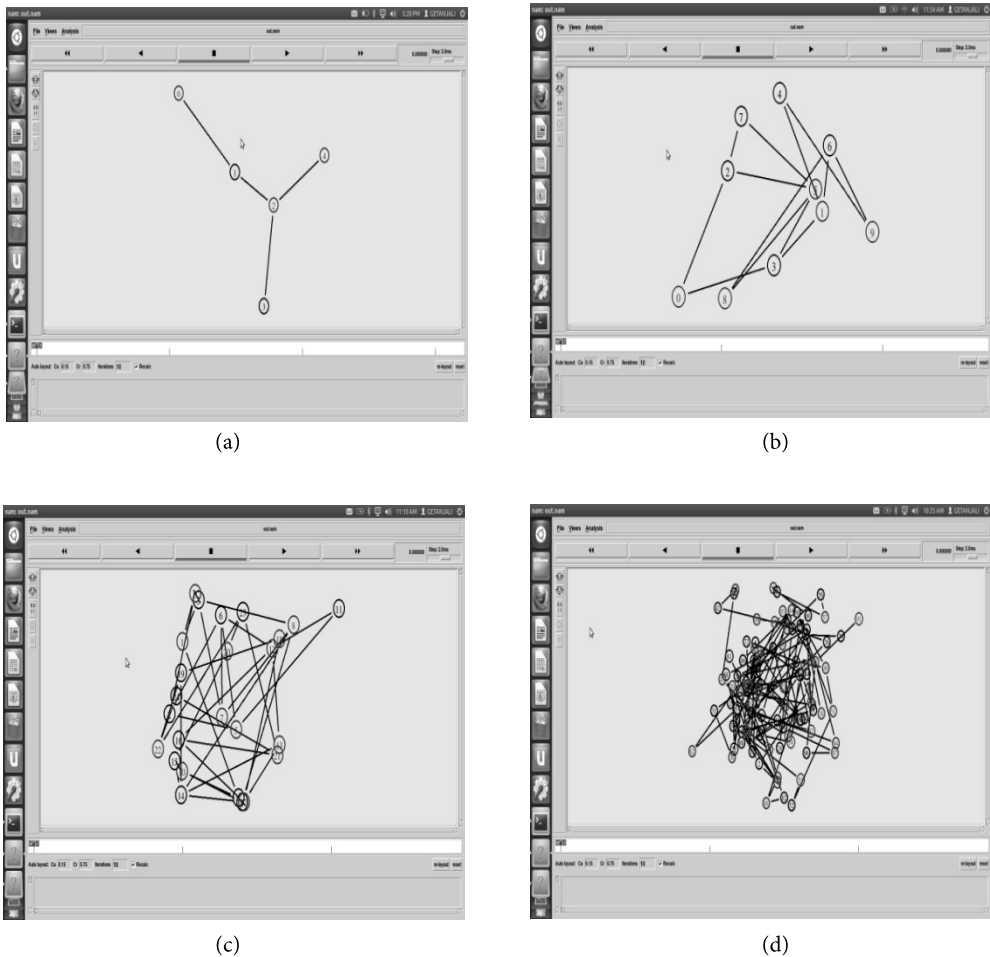
Step 3: Then, node C transmits the same packet 1 to buffered node F within the same time period (i.e., 1 second).

Step 4: Buffered node F sends an ACK to node C (in 1 second) and then node C forwards the buffered node F ACK to node A (again in 1 second).

So, the total amount of time to transmit the first packet from node A to node F is 5  $(1+1+1+1+1)$

seconds. After 5 seconds, the first packet will be removed from the A buffer node and the next packet will arrive. To increase the speed of processing and to decrease delay and congestion in the network we used a buffer size of 5 units.

The BAA node generation results are shown in Fig. 2. The terminologies used in RPT (Table 3) and BBR (Table 4) are shown in Table 2.



**Fig. 2.** Simulation results of buffer allocation algorithm for (a)  $n=5$ , (b)  $n=10$ , (c)  $n=25$ , and (d)  $n=100$  network sizes.

### 2.1.2. Selection of the optimum path for routing

In the BBR approach, the routing table consist of seven parts—i.e., 1) node, 2) node address, 3) next hop, 4) next hop buffered, 5) next hop address, 6) cost, and 7) buffered node. Initially, the buffer space will not be allocated to the next hop and the next hop buffered field. These fields are updated when the BAA is executed. When the BAA is terminated, the last visited node takes the updated value, which it broadcasts to all of the network nodes. Thus, the routing table of each node of the network is updated. Table 1 shows the RT of buffered node J, which is the last step of the BAA algorithm.

**Table 1.** Routing table of buffered node 'J'

Node ID	Node address	Next hop ID	Next hop buffered?	Next hop address	Cost	Buffered node?
A	...	C	No	...	1	Yes
A	...	D	No	...	2	Yes
B	...	D	No	...	1	Yes
B	...	G	No	...	2	Yes
B	...	E	No	...	1	Yes
C	...	F	Yes	...	2	No
C	...	H	No	...	3	No
D	...	A	Yes	...	2	No
D	...	B	Yes	...	1	No
D	...	F	Yes	...	2	No
D	...	G	No	...	1	No
E	...	B	Yes	...	1	No
E	...	J	Yes	...	2	No
F	...	C	No	...	2	Yes
F	...	D	No	...	2	Yes
F	...	H	No	...	4	Yes
F	...	I	No	...	2	Yes
G	...	D	No	...	1	No
G	...	B	Yes	...	2	No
G	...	I	No	...	1	No
G	...	J	Yes	...	1	No
H	...	C	No	...	3	No
H	...	F	Yes	...	4	No
I	...	F	Yes	...	2	No
I	...	G	No	...	1	No
J	...	G	No	...	1	Yes

### 2.1.3. Resilient packet transmission

The aim of RPT is to successfully transfer information from the source to the destination node of a network, even during failures. For this purpose, the routing path from the source to the destination in RPT must have following characteristics [16]:

- a) The route must contain a minimum number of buffered nodes.
- b) If more than one path has the same number of buffered nodes, then it will select the lowest cost.

Initially, the BBR approach allocates buffers to the entire network (as described in Part (i)) and after that the packet transmission starts. In the network the less than or equal to  $n/2$  nodes are buffered. During packet transmission, the non-buffered node forwards packet to next node and send acknowledgement (ACK) to its preceding node. The preceding buffered node will store the packet until an ACK is received from the next buffered node. When the ACK is received from the next buffered node, the preceding buffered node deletes the packet from the buffer. The detailed explanation of packet transmission and failure cases has been already discussed in our previous paper [16]. Table 3 shows the proposed RPT algorithm.

## 2.2 Complexity Analysis

This section shows the time complexity of the proposed BAA [16] and RPT approach. The BBR approach is previously allocating buffer till the destination but now, we have updated the BBR approach to work for the un-traversed link beyond destination (until all the nodes are traversed). The time complexity of both BAA and RPT is executed as  $O(N \log_2(N))$  using the Brute-force method. Tables 4 and 5 show the time complexity of the BAA and RPT approaches.

**Table 2.** Terminologies used in BAA and RPT Algorithms

Terms used in algorithms (Table 3-5)	Description
<b>Buffered node <math>N_i</math></b>	Signifies the node that is a buffered node (i.e., already assigned a buffer)
<b>Next[<math>N_i</math>]</b>	It is used to visit the next node in the network during the buffer allocation process
<b>(Node→Node<sub>next</sub>)</b>	During the allocation of buffers in the network, we maintained a list of nodes that are visited and assigned buffers. So, Node→Node <sub>next</sub> , which defines if the next node (i.e., node→node <sub>next</sub> ) is present in the visited list and then rolls back to its parent node (i.e., previous→ $N_i$ )
<b>Buffer[<math>N_i</math>]</b>	Signifies the array that contains the node id of the buffered node
<b>visited list[ ]</b>	This is an array that contains the node id and stores the information of the node that was previously visited in the network
<b><math>N_i \leftarrow \min\_cost\_next [N_i]</math></b>	It is used to select the node that has the minimum link cost from the current node and then the node that has the less minimum cost will become the next current node
<b><math>N_i \neq \text{visited\_list}[ ]</math></b>	In our algorithm for each next node $N_i$ we check whether $N_i$ is present in visited list. If $N_i$ is present in the visited list array, then we go for the next node, which has the “next_min_cost( $N_i$ ).” Otherwise, we continue our process from the current node
<b>Buffered array[ ]</b>	This is an array that contains the node id of the nodes, which has already been buffered
<b>visited list[ ] + 1 = <math>N_i</math></b>	This maintains the visited current node $N_i$ into the visited_list[]
<b>assign_buffer(<math>N_i</math>)</b>	It is a function used to assign the buffer to the node $N_i$
<b>rollback(previous→<math>N_i</math>)</b>	If node $N_i$ is present in the visited list and its next node is also present in the visited list and then we rollback to the parent node of $N_i$
<b>array_buffer[ ]←<math>N_i</math></b>	This is an array that contains the information about the buffered node, if the node $N_i$ is found to be eligible to assign and then it is added to this array
<b>update_routing_table()</b>	This is a function that is used to update the routing table

**Table 3.** Resilient packet transmission algorithm

- 
1. The source (buffered) node starts transmission through the least cost path by considering its routing table (in accordance to Section 2.1.3 (a) and (b))
  2. Wait for two ACK's (buffered node/non buffered node)
- 
3. **If** (next[N<sub>i</sub>]!=buffered node N<sub>i</sub>)
  4.       Receive and forward the packets to its downstream node N<sub>i</sub> and starts RTT (round trip time).
  5.       Send ACK to its previous node N<sub>i</sub>.
  6.       Go To Step(3)
  7.       **End: if**
  8.       **Else If** (next[N<sub>i</sub>]=buffered [N<sub>i</sub>])
  9.       Store and forward the packet to next N<sub>i</sub> and starts RTT
  10.       Send ACK to its previous node N<sub>i</sub>
  11.       Wait for two ACKs.
  12.       **If** (two ACKs received within RTT)
  13.       Go To Step(8)
  14.       **Else**
  15.       Go To Step(22)
  16.       **End: else if**
  17.       **Else**
  18.       **If** (Packet receive within its RTT)
  19.       Repeat Step 3
  20.       **End: if**
  21.       **Else**
  22.       Failure (N<sub>i</sub>)
  23.       **End: else**
  24.       **End: elseif**
  25.       **Else** (destination node)
  26.       Receive and store packet
  27.       Send ACK to its previous node.
  28.       **End: else**
- 

**Failure (N<sub>i</sub>)**

1. **If** (failure N<sub>i</sub>=exist)
  2. Preceding the buffered node and select another least cost path to the destination node.
  3. **End if**
-

**Table 4.** Time complexity of buffer allocation algorithm

1.	Select random node $N_i$ from the network	$O(1)$
2.	Assign buffer ( $N_i$ )	$O(1)$
3.	Visited list [] = $N_i$	$O(\log_2(N))$
4.	<b>If</b> (next [ $N_i$ ] != null)	$O(1)$
5.	$N_i \leftarrow \text{min\_cost\_next } [N_i]$	$O(N)$
6.	<b>If</b> ( $N_i = \text{visited list } []$ )	$O(\log_2(N))$
7.	Select next_min_cost ( $N_i$ )	$O(1)$
8.	Go To Step (4)	$O(N \log_2(N))$
9.	<b>End if</b>	
10.	<b>Else</b>	
11.	<b>If</b> (previous [ $N_i$ ] = buffered array [])	$O(\log_2(N))$
12.	Skip ( $N_i$ )	$O(1)$
13.	Visited list [] + 1 = $N_i$	$O(\log_2(N))$
14.	Go To Step (4)	$O(\log_2(N))$
15.	<b>End if</b>	
16.	<b>Else</b>	
17.	Allocation( $N_i$ )	$O(N \log_2(N))$
18.	Go to Step(4)	$O(N \log_2(N))$
19.	<b>End else</b>	
20.	<b>End else</b>	
21.	<b>Else</b>	
22.	Rollback ( $N_i$ );	$O(1)$
23.	<b>End else</b>	
24.	Repeat Step 2 until two rollbacks occur at the same node;	$O(1) \times N \log_2(N)$
25.	Return;	
<hr/>		
<b>Allocation (<math>N_i</math>)</b>		
<b>Begin</b>		
1.	<b>If</b> ( $N_i \neq \text{visited\_list} []$ )	$O(\log_2(N))$
2.	assign_buffer( $N_i$ )	$O(\log_2(N))$
3.	visited_list [] + 1 = $N_i$	$O(1)$
4.	<b>End if</b>	
5.	<b>Else</b>	
6.	rollback (previous $\rightarrow N_i$ )	$O(1)$
7.	<b>End else</b>	
8.	<b>End</b>	
<hr/>		
<b>assign_buffer (<math>N_i</math>)</b>		
<b>Begin</b>		
1.	assign buffer to $N_i$	$O(1)$
2.	array_buffer [] $\leftarrow N_i$	$O(1)$
3.	update_routing_table()	$O(\log_2(N))$
4.	<b>End</b>	
<b>Total complexity of Buffer allocation algorithm:</b>		$O(N \log_2(N))$



**Table 5.** Complexity analysis of RPT

1.	Select random node $N_i$ from the new Source (buffered) node starts transmission through least cost path by considering its routing table	$O(\log_2(N))$
2.	Wait for two ACKs (buffered node/non buffered node)	$O(1)$
3.	<b>If</b> (next[ $N_i$ ]!=buffered node $N_i$ )	$O(1)$
4.	Receive and forward the packets to its downstream node $N_i$ and starts RTT(round trip time)	$O(1)$
5.	Send ACK to its previous node $N_i$	$O(1)$
6.	GoTo Step(3)	$O(N)$
7.	<b>End: if</b>	
8.	<b>Else If</b> (next[ $N_i$ ]=buffered [ $N_i$ ])	$O(1)$
9.	Store and forward the packet to next $N_i$ and starts RTT	$O(\log(N))$
10.	Send ACK to its previous node $N_i$	$O(1)$
11.	Wait for two ACKs	$O(1)$
12.	<b>If</b> (packet received within RTT)	
13.	Go To Step(8)	$O(N)$
14.	<b>Else</b>	
15.	Go To Step(22)	$O(N)$
16.	<b>End: elseif</b>	
17.	<b>Else</b>	
18.	<b>If</b> (Packet receive within its RTT)	$O(1)$
19.	Repeat Step 3	$O(N)$
20.	<b>End: if</b>	
21.	<b>Else</b>	
22.	Failure( $N_i$ )	
23.	<b>End: else</b>	$O(\log(N))$
24.	<b>End: else if</b>	
25.	<b>Else</b> (destination node)	$O(1)$
26.	Receive and store packet	$O(1)$
27.	Send the ACK to its previous node	$O(1)$
28.	<b>End: else</b>	
<b>Failure (<math>N_i</math>)</b>		
1.	<b>If</b> (failure $N_i$ =exist)	$O(1)$
2.	Preceding buffered node select another least cost path to destination node.	$O(\log_2(N))$
3.	<b>End if</b>	
<b>Total complexity of RPT algorithm:</b>		$O(N\log_2(N))$

The time complexities of the Buffer Allocation and RPT algorithms are  $O(N\log_2(N))$ .

### 3. Performance Evaluation

In this section we analyze the performance of the BBR approach with previously defined approaches (i.e., Resilient Multicast [14] and ROMER [15]). We evaluate the network performance using some parameters (i.e., throughput [defined in terms of cost], network congestion [in terms of packet transmission], and resiliency against node/link failures [in terms of fault tolerance]). We considered the cost as the total amount of delays that occur during the transmission of packets from the source to the destination. While packet transmission is the total number of packets transmitted at a time in the network, the fault tolerance [17-19] is the possible numbers of paths that exist after a failure. We evaluated the performance of Resilient Multicast, ROMER, and BBR over five different network sizes (i.e., 5, 10, 15, 20, 25, 100). Let us first consider, a network of five nodes (see Fig. 3(a)) where  $A$  is the source node,  $E$  is the destination node, and compare these approaches.

1) **Network Throughput:** throughput is defined in terms of cost (i.e., the total amount of delay that occurs during the transmission of packets from the source to the destination). We also evaluated the network throughput of these approaches for network size 5 (Fig. 3(a)).

- The Network Throughput of the Resilient Multicast [14]: this approach selects at least two node disjoint paths to send data packets. To analyze the throughput, we calculated the total amount of costs of selected disjoint paths to reach from  $S - D$  (see Fig. 3(a)). Source  $A$  selects two disjoint paths (i.e.,  $A - B - D - E$ ), which consumes  $1 + 2 + 2 = 5$  units (through  $A - B$ ,  $B - D$ ,  $D - E$ ) and  $A - C - E$ , which consumes  $2 + 1 = 3$  units of cost (through  $A - C$ ,  $C - E$ ). As such, the packet is sent to its destination node by consuming (cost of path 1 + cost of path 2)  $5 + 3 = 8$  units.
- The Network Throughput of ROMER [15]: This approach forwards the packet by taking the maximum credit cost (which is assumed at the source node). ROMER states that each node has some cost and the packets will be forwarded by every node after calculating the value of credit cost  $R$  and threshold value  $T$ . This is done in such a way that:

**If** ( $R > T$ )

Then the node forwards the packet;

**Else**

Discard the packet.

The detailed steps of this approach for network size 5 are as follows:

- a) Initially  $A$  broadcasts data to its downstream nodes (i.e.,  $B$  and  $C$ ), which consumes a total of 3 units of cost to send the packets ( $A - C = 2$  unit) and ( $A - B = 1$  unit).
- b) Assuming that the cost of node  $B = 55$  and node  $C = 50$ , then the packet will be forwarded by downstream nodes after calculating the value of  $R$  and  $T$ . At node  $B$  if  $R < T$  then it discards the packet. At node  $C$ , assuming that  $R > T$ , then it forwards the packet to its downstream nodes  $D$  and  $E$ . So the total cost of sending the packets from  $C - D$  and  $C - E$  is  $1 + 1 = 2$  units.
- c) The further value of  $R$  and  $T$  are calculated at node  $D$  and then it forwards the packet to destination  $E$  after consuming 2 units of cost.
- d) Now, the cost to send the packets from  $A - E$  is (cost in (i) + cost in (ii) + cost in (iii)) (i.e.,  $3 + 2 + 2 = 7$  units of cost).

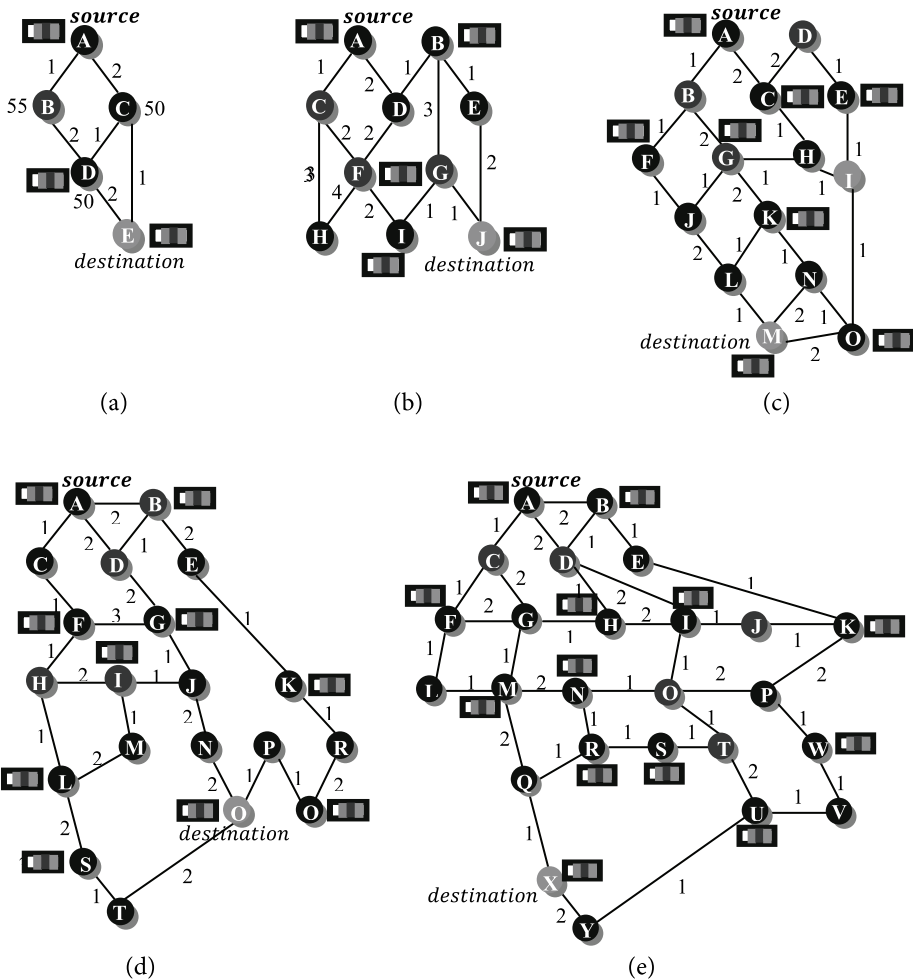
- The Network Throughput of BBR [16]: according to our proposed approach, the packets are sent through a route, which has the least number of buffered nodes. In Fig. 3(a), the path  $A - C - E$  contains the minimum amount of buffered nodes (i.e., 2). So it sends the packet using the  $A - C - E$  path by consuming  $2 + 1 = 3$  units (cost of  $A - C + C - E$ ). Hence, the BBR approach costs the least to send data packets from  $S - D$ .
- 2) **Network Congestion:** this is defined in terms of the rate of packet transmissions (i.e., the total number of packets transmitted at a unit of time in a network). Let us evaluate network congestion on these approaches.
- The network congestion on Resilient Multicast: Fig. 3(a) shows that source  $A$  has 20 packets to transmit to the destination node. [14] sends the packets through two-node disjoint paths (i.e.,  $A - B - D - E$  and  $A - C - E$ ). Twenty packets will be sent through either of these paths (i.e.,  $A - C - E$ ). The Resilient Multicast uses a redundant copy of these packets to send through to other path (i.e.,  $A - B - D - E$ ). If the source node has 20 packets to transmit, [14] will send 40 packets in the network.
  - Network congestion on ROMER [15]: to provide the successful delivery of packets to the destination node, it delivers the redundant copy of packets in the network (see Fig. 3(a)). Let's assume that source node  $A$  has 20 packets to transmit; to calculate the cost consider the following steps:
    - a)  $A$  forwards traffic to both nodes (i.e.,  $B$  and  $C$ ), so the total number of packets are  $20 + 20 = 40$ .
    - b)  $B$  discards the packet in case ( $R < T$ ) and node  $C$  forwards the redundant copy of packets to  $D$  and  $E$ . Now, the number of packets to transmit are  $20 + 20 = 40$  packets.
    - c)  $D$  forwards the packets through a single path (i.e.  $D - E$ ), which transmits 20 packets.
    - d) The total number of packets inside the network is  $40 + 40 + 20 = 100$  packets (traffic size of ((i) + (ii) + (iii))).
  - The Network Congestion on BBR [16]: this forwards the packets according to its buffer capacity, as shown in Fig. 3(a). Buffers are placed at alternate positions. Each buffer is 4 units in size. If source node  $A$  has 20 units to transmit, then only 8 packets will be forwarded inside the network at a time.
- 3) **Network Resiliency:** resiliency against node/link failure is measured in terms of fault tolerance. Fault tolerance is defined as the possible number of paths that exist after failure. Let's assume that node  $B$  has failed.
- Resiliency in the Resilient Multicast [14]: resiliency is achieved in [14] through a two-node disjoint path (See Fig. 3(a)). Source  $A$  selects two paths  $A - B - D - E$  and  $A - C - E$ . If node  $B$  fails, then resiliency is achieved by using a second path (i.e.,  $A - C - E$ ). If nodes  $B$  and  $C$  fail, then there is no possible way to reach to destination.
  - Resiliency in ROMER [15]: resiliency is achieved in [15] by forwarding the redundant copy of packets in the network. Initially, [15] forwards the packets through 3 possible paths (i.e.,  $A - B - D - E$ ,  $A - C - D - E$ ,  $A - C - E$ ). If node  $B$  fails, then there exist 2 other possible paths (i.e.,  $A - C - D - E$  and  $A - C - E$ ).
  - Resiliency in BBR [16]: resiliency is achieved in BBR by storing data packets in buffers that are placed at alternate positions in the network. When a failure occurs in the network during the

transmission of data packets, the previously buffered node selects another efficient path for packet transmission (see Fig. 3(a)). If node *B* fails, then there is no effect inside the network because we sent the data packets through the *A – C – E* path.

To evaluate the accuracy of the performance evaluation of Resilient Multicast, ROMER, and BBR we analyzed the network parameters (throughput, network congestion, resiliency) on 10, 15, 20, and 25 network sizes (see Table 6).

**Table 6.** Parameter growth of 3 approaches in a network with 5 nodes

Parameter metric	Resilient multicast	ROMER	BBR
Throughput	8	7	3
Network congestion	40	100	8
Resiliency	1	2	1



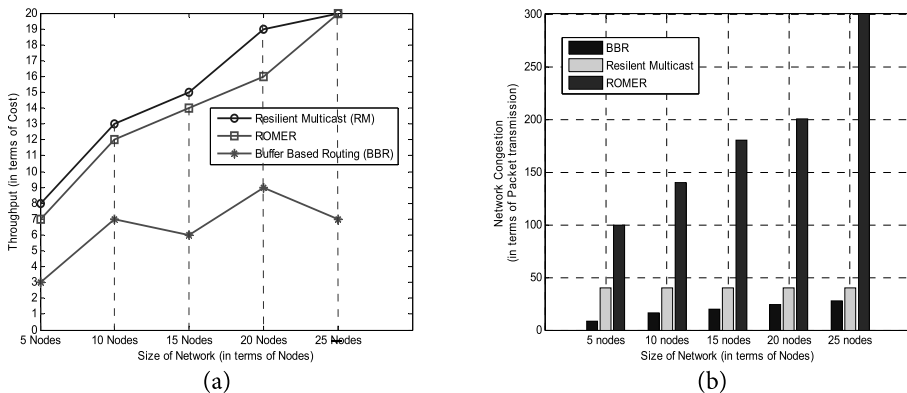
**Fig. 3.** Network with different sizes ((a), (b), (c), (d) and (e) represents 5, 10, 15, 20 and 25 network sizes, respectively).

For network size 10, we analyzed the network parameters (throughput, network congestion and resiliency against node/link failure) on 3 approaches (i.e., Resilient Multicast [14], ROMER [15] and BBR [16]) using a network of 10 nodes.

See Fig. 3(b) for the throughput measures in terms of cost. [14] selects two-disjoint paths (i.e.,  $A - C - F - I - G - J$ ), which consumes 7 units of cost and  $A - D - B - E - J$ , which consumes 6 units of cost. So, the total amount of cost consumed by Resilient Multicast is 13 units. In [15], let nodes  $C, F, G$  discard the packet. As such, it consumes 12 units of cost ( $A - C, A - D, D - F, D - B, B - G, B - E, E - J$ ) and it sends the packets through the  $A - D - B - E - J$  path while BBR [16] takes 7 units of cost through the  $A - D - B - E - J$  path. Network congestion is measured in terms of packet transmission. Suppose source node  $A$  has 20 packets to transmit, [14] transmits 40 packets using the  $A - C - F - I - G - J$  and  $A - D - B - E - J$  paths inside the network. [15] transmits 140 packets through the  $A - D - B - E - J$  path (assume that nodes  $C, F, G$  discard the packet), while BBR [16] only has 4 packets to transmit through the  $A - D - B - E - J$  path. While resiliency is measured in terms of possible path exist after failure. In [14], if node  $D$  fails, then the packet will be forwarded through  $A - C - F - I - G - J$ . In [15] there exist 2 possible paths (i.e.,  $A - C - H - F - I - G - J$  and  $A - C - F - I - G - J$ ). In BBR, the packets will be forwarded through  $A - C - F - I - G - J$  (an explanation has been described in 5 nodes of the network). Table 7 shows the parameter growth of 3 approaches in a network of 10 nodes.

**Table 7.** Parameter growth of 3 approaches in a network with 10 nodes

Parameter metric	Resilient multicast	ROMER	BBR
Throughput	13	12	7
Network congestion	40	140	16
Resiliency	1	2	1



**Fig. 4.** Comparative analysis of network throughput and congestion in Resilient Multicast, ROMER, and BBR for different network sizes.

For network size 15 (see Fig. 3(c)), the throughput of Resilient Multicast, ROMER, and BBR is 15, 14 and 6; network congestion is 40, 180 and 20; and the resilient paths are 1, 8, and 1. For network size 20 (see Fig. 3(d)), the throughput is 19, 16, 9; network congestion is 40, 200, and 24, and the resilient paths are 1, 5, and 1. For network size 25 (see Fig. 3(e)), the throughput is 20, 20, and 7; network

congestion is 40, 300, and 28; and the resilient paths are 1, 8, and 1, as shown in Table 8. Fig. 4(a) shows the comparative analysis of network throughput for the Resilient Multicast, ROMER, and BBR approaches, while Fig. 4(b) shows the robustness of the BBR approach during network congestion. This allows us to provide a simulation of a 10, 100 node network.

**Table 8.** Comparison of network parameters on 3 approaches [14-16]

Parameter metric	Approaches	5-node	10-node	15-node	20-node	25-node	100-node
<b>Throughput</b> (in terms of cost)	Resilient Multicast	8	13	15	19	20	89
	ROMER	7	12	14	16	20	76
	BBR	3	7	6	9	7	66
<b>Network congestion</b> (in terms of Packet transmission)	Resilient Multicast	40	40	40	40	40	40
	ROMER	100	140	180	200	300	650
	BBR	8	16	20	24	28	88
<b>Resiliency</b> (against node/link failure)	Resilient Multicast	1	1	1	1	1	1
	ROMER	2	2	8	5	8	38
	BBR	1	1	1	1	1	1

## 4. Conclusion and Future Work

In this work we have considered node/link failure issues to provide an analogous performance with the BBR approach. We have proposed the RPT algorithm as an antidote during such failures. We have evaluated the time complexity of the RPT approach as  $O(N \log_2(N))$ . Furthermore, we have proven that the performance of BBR is comparatively improved over previously proposed approaches (i.e., Resilient Multicasting and ROMER). We have done so by using network throughput, resiliency against node/link failures, and network congestion as the parameters for the 5, 10, 15, 20, and 25 nodes network and show the simulation results of the 10 and 100 node networks (in Fig. 1). In the future, to achieve more accurate results with the BBR approach, we will apply it in the physical environment and will study the results for any other possible measures of routing parameters.

## References

- [1] O. Kwon and Y. Wen, "An empirical study of the factors affecting social network service use," *Journal of Computers in Human Behavior*, vol. 26, no. 2, pp. 254-263, 2010.
- [2] T. Hu, R. S. Poston, and W. J. Kettinger, "Nonadopters of online social network services: is it easy to have fun yet?," *Journal of Communications of the Association for Information Systems*, vol. 29, no. 1, pp. 441-458, 2011.
- [3] M. Manoufali, H. Alshaer, P. Y. Kong, and S. Jimaa, "Technologies and networks supporting maritime wireless mesh communications," in *Proceedings of the 6th Joint IFIP Wireless and Mobile Networking Conference (WMNC)*, Dubai, 2013, pp. 1-8.
- [4] W. S. Kim and S. H. Chung, "Design of optimized AODV routing protocol for multi-interface multi-channel wireless mesh networks," in *Proceedings of the IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, Barcelona, Spain, 2013, pp. 325-332.
- [5] L. Song and Z. B. Xia, "An anycast routing protocol for wireless mesh access network," in *Proceeding of the IEEE WASE International Conference on Information Engineering (ICIE)*, Taiyuan, Shanxi, 2009, pp. 82-85.

- [6] B. K. Gupta, B. M. Acharya, and M. K. Mishra, "Optimization of routing algorithm in wireless mesh networks," in *Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*, Coimbatore, India, 2009, pp. 1150-1155.
- [7] J. Dong, R. Curtmola, and C. Nita-Rotaru, "Secure high-throughput multicast routing in wireless mesh networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 5, pp. 653-668, 2011.
- [8] H. Herrman, D. E. Stewart, N. Diaz-Granados, E. L. Berger, B. Jackson, and T. Yuen, "What is resilience?" *Canadian Journal of Psychiatry*, vol. 56, no. 5, pp. 258-265, 2011.
- [9] H. Mustafa, X. Zhang, Z. Liu, W. Xu, and A. Perrig, "Jamming-Resilient Multipath Routing," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 6, pp. 852-864, 2012.
- [10] N. Rakesh and V. Tyagi, "Failure recovery in XOR'ed networks," in *Proceedings of the IEEE International Conference on Signal Processing, Computing and Control (ISPC)*, Wagnaghat, India, 2012, pp. 1-6.
- [11] N. Rakesh and V. Tyagi, V. "Failure detection using contour approach on network coded parallel networks," *Procedia Engineering*, vol. 38, pp. 763-770, 2012.
- [12] S. Lee, B. Bhattacharjee, A. Srinivasan, and S. Khuller, "Efficient and resilient backbones for multihop wireless networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 11, pp. 1349-1362, 2008.
- [13] T. Bu, M. C. Chan, and R. Ramjee, "Connectivity, performance, and resiliency of IP-based CDMA radio access networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 8, pp. 1103-1118, 2006.
- [14] X. Zhao, J. Guo, C. T. Chou, and S. Jha, "Resilient multicasting in wireless mesh networks," in *Proceedings of the 13th International Conference on Telecommunications*, Polo de Aveiro, Portugal, 2006, pp. 1-4.
- [15] Y. Yuan, H. Yang, S. H. Wong, S. Lu, and W. Arbaugh, "ROMER: resilient opportunistic mesh routing for wireless mesh networks," in *Proceedings of the IEEE Workshop on Wireless Mesh Networks (WiMesh)*, 2005, pp. 1-9.
- [16] G. Rathee, A. Mundra, N. Rakesh, and S. P. Ghreera, "Buffered based routing and resiliency approach for WMN," in *Proceedings of the International Conference on Human Computer Interactions (ICHCI)*, Chennai, India, 2013, pp. 1-7.
- [17] B. K. Gupta, A. Mundra, and N. Rakesh, "Failure detection and recovery in hierarchical network Using FTN approach," *International Journal of Computer Science*, vol. 10, no. 2, pp. 414-421, 2013.
- [18] K. Shah, G. Dua, D. Sharma, P. Mishra, and N. Rakesh, "Transmission of successful route error message (RERR) in routing aware multiple description video coding over mobile ad-hoc network," *International Journal of Multimedia & Its Applications*, vol. 3, no. 3, pp. 51-59, 2011.
- [19] N. Rakesh and V. Tyagi, "Parallel architecture coding: link failure-recovery mechanism (PAC: LF-RM)," *International Journal of System Assurance Engineering and Management*, vol. 4, no. 4, pp. 386-396, 2012.



### Geetanjali Rathee

She is a Teaching Assistant (TA) in the Department of Computer Science Engineering (CSE) & Information Communication Technology (ICT) with Jaypee University of Information Technology (JUIT), Wagnaghat, Solan-173234, Himachal Pradesh, India. She received her B.Tech. degree in Computer Science and Engineering (CSE) from Bhagwan Mahavir Institute of Engineering and Technology (BMIET), Haryana in the year 2011. Now she is undertaking the Master Degree course under supervision of Dr. Nitin Rakesh in Jaypee University of Information and Technology (JUIT), Wagnaghat, Solan-173234. Her research interests include Resiliency in Wireless Mesh Networking, Routing Protocols, and Networking.



### **Nitin Rakesh**

He is an Assistant Professor (Senior Grade) in the Department of Computer Science Engineering & Information Communication Technology with Jaypee University of Information Technology (JUIT), Wagnaghat, Solan 173234, Himachal Pradesh, India. He received his doctorate in Department of Computer Science and Engineering from JUIT, Wagnaghat in 2012. In 2007, he received his Master of Technology Degree in Computer Science and Engineering from Jaypee Institute of Information Technology, Noida, India and received Bachelor in Technology Degree in Information Technology from AEC, Agra in the year 2004. He is a member of Institute of Electrical and Electronics Engineers (IEEE-USA), Life Member Computer Society of India (LMCSI), and International Association of Engineers (IAENG). His research outlines emphasis on Network Coding, Interconnection Networks & Architecture, Fault-tolerance & Reliability, Networks-on Chip, Systems-on-Chip, Network Algorithms, Parallel Algorithms and Fraud Detection, Online Phantom Transactions.