IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received March 31, 2019, accepted May 2, 2019, date of publication May 27, 2019, date of current version June 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2918990

Congestion Avoidance for Smart Devices by Caching Information in MANETS and IoT

NOUSHEEN AKHTAR[1], MUAZZAM A. KHAN[2], (Senior Member, IEEE), ATA ULLAH[3], (Member, IEEE), AND MUHAMMAD YOUNUS JAVED[4], (Senior Member, IEEE)

[1]Department of Computer Engineering, College of EME, National University of Sciences and Technology, Rawalpindi 46000, Pakistan
[2]Department of Computing, SEECS, National University of Sciences and Technology, Islamabad 44000, Pakistan
[3]Department of Computer Science, National University of Modern Languages, Islamabad 44000, Pakistan
[4]Department of Computer Science and Engineering, Heavy Industries Taxila Education City University, Taxila 47070, Pakistan

Corresponding author: Ata Ullah (aullah@numl.edu.pk)

ABSTRACT Mobile ad-hoc networks (MANETs) comprise a large number of mobile wireless nodes that can move in a random fashion with the capability to join or leave the network anytime. Due to the rapid growth of devices on the Internet of Things (IoT), a large number of messages are transmitted during information exchange in dense areas. It can cause congestion that results in increasing transmission delay and packet loss. This problem is more severe in larger networks with more network traffic and high mobility that enforces dynamic topology. To resolve these issues, we present a bandwidth aware routing scheme (BARS) that can avoid congestion by monitoring residual bandwidth capacity in network paths and available space in queues to cache the information. The amount of available and consumed bandwidth along with residual cache must be worked out before transmitting messages. The BARS utilizes the feedback mechanism to intimate the traffic source for adjusting the data rate according to the availability of bandwidth and queue in the routing path. We have performed extensive simulations using NS 2.35 on Ubuntu where TCL is used for node configuration, deployment, mobility and message initiation, and C language is used for modifying the functionality of AODV. The results are extracted from trace files using Perl scripts to prove the dominance of the BARS over preliminaries in terms of packet delivery ratio, throughput and end-to-end delay, and the probability of congested node for static and dynamic topologies.

INDEX TERMS Congestion, data rate, link capacity, MANETs, IoT.

I. INTRODUCTION

Mobile Ad-hoc Networks (MANETs) comprises of independent mobile nodes that are randomly deployed and can leave or join the network on the move. These nodes communicate with each other via wireless links to exchange information. Ad hoc allows new devices to be added quickly. Each device in the network can freely move in any direction that results in dynamic topology. Designing a network is very difficult task as there are lots of challenges and issues in its design. Each node can acts as router which forwards packet from source to destination. These nodes can be any personal device such as laptops, mobile phones etc. MANETs and IoT applications span from small networks to very large dynamic networks. Examples include military applications, low level like class room conference room, automated battle filed, rescue operations, emergency operations and also being used in VANET (Vehicular Ad hoc network) [1]. In these networks, nodes communicate with each other in multi hop fashion. When sender transmits data packet to destination node, it uses some intermediate node for communication. So each node in the networks plays equally important role. To meet the demands of design constraints of IoT enabled MANETs, an efficient routing scheme is required.

Routing provides the ways for appropriate path selection within network. While the routing protocol provides communication between routers and process the data packet from source to destination by selecting appropriate path between sender and receiver. Designing a routing protocol is very challenging task. Many routing protocols have been proposed so far. These protocols can be broadly categorized as reactive proactive and hybrid protocols [3], [4] that are also applicable in IoT for mobile devices [5] as follows; i) In this kind of protocols like DSDV (destination sequence distance vector),

The associate editor coordinating the review of this manuscript and approving it for publication was Jun Wu.

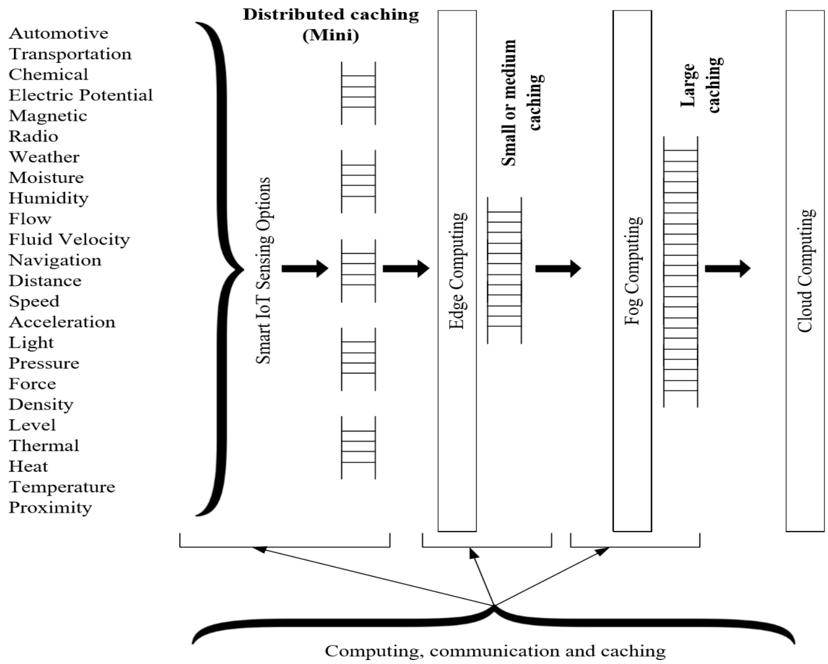


FIGURE 1. Smart sensing scenario for IoT with multiple caching, computation and communication support to manage sensing bottleneck.

mobile nodes update their routing tables by periodically exchanging routing information between them. Due to this exchange of information proactive routing protocols generate a lot of control messages and increase the network overhead. Hence these kind of routing protocols are not suitable for MANETs and IoT; ii) To overcome the limitations of proactive protocol, reactive routing protocols like AODV (Ad hoc on demand distance vector) and DSR (dynamic source routing) have been designed for MANETs. Reactive routing includes route discovery and route maintenance; iii) In this case, each node behaves proactively when it is outside the region and reactively when it enters into the region close to its destination. Network performance depends on selection of routing mechanism that ensures timely and successfully transmission of data packets with improved packet delivery rates [6]–[8]. A good routing protocol avoids congestion in the network by increasing network throughput and decreasing its overhead. In literature, many protocols have been proposed to overcome network congestion [9]–[12].

Computing, communication and caching are used in combination where caching at different network levels improves the overall latency in transmitting the sensed data towards the cloud whenever mandatory. It is focused to temporarily store the sensed data locally at cache and then either completely consume the data locally without transmission or process it locally to transmit an aggregated and compressed information to cloud. This strategy can greatly benefit to reduce congestion in MANETs and IoT scenarios by managing the sensed data locally to mitigate the effect of sensing bottleneck due to massive data transmission for variety of sensing capabilities as illustrated in figure 1 [13].

Congestion is a condition in the networks when there are too many data packets are present in the subnet. Congestion occur when network carries more load (i.e. number of packets sent to the network) then its capacity (number of packets handed by the network). Congestion leads to packet loss and bandwidth degradation. In case of MANETs and IoT, congestion does not overload mobile nodes but it effects overall coverage area. If the selected routing protocol is unable to handle congestion, following issues can arise within the network [14]–[16].

i) Increase in delay: It detects the occurrence of congestion by estimating the expected time to deliver. If there is long delay, then network congestion might be one of the reason. In such kind of situations, it is better to select some alternate path but again selection of new path and searching process depends on routing protocol selected.
ii) High overhead: in case of multipath routing more processing is required. For the selection of alternate path in case of congestion, it requires more retransmission attempts that increases network overhead.
iii) Increase in packet loss: congestion control techniques try to minimize network load by either reducing its sending rate or drops packets at intermediate node. This process increases the number of packet drop ratio that ultimately decreases network throughput [17]. Figure 2 illustrates congestion scenario among multiple senders and receivers.

This paper presents a bandwidth aware routing scheme that cache the information in queue to adjust data rates and hence congestion. Our main contributions are as follows;

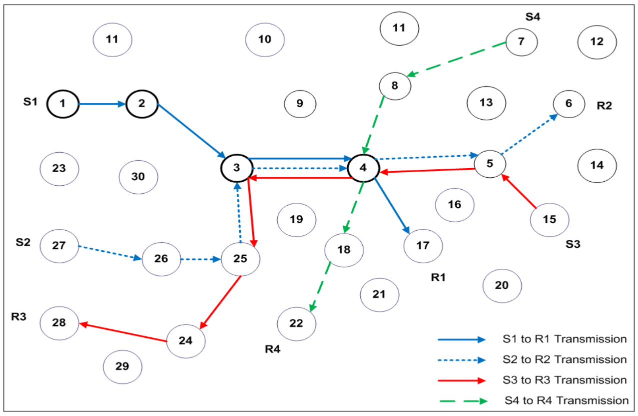


FIGURE 2. Congestion scenario with multiple senders and receivers.

1) The scheme allows source to adjust its sending rate whenever network is near to congestion. We modify existing AODV as per available bandwidth in the path and residual queue sizes of each node in path.
2) The proposed routing mechanism modifies the RREQ and RREP messages of AODV by embedding path bandwidth and queue size in it. Moreover, RERR message is also modified to handle path break.
3) In order to provide quality of service to the routing we have used bandwidth and queue size as a metric for route selection.
4) To test the performance, we implemented our proposed technique in NS2 simulator. Results shows that proposed routing mechanism outperforms in comparison with most recent technique named, mitigation of packet loss using data rate adaptation scheme.

The rest of paper is organized as follows; Section 2 include literature review about congestion avoidance schemes, and Section 3 describes our proposed BARS scheme along with proposed algorithms. In Section 4, we have presented the results and analysis along with simulation environment. At the end conclusion and future work are presented in Section 5.

II. LITERATURE REVIEW

Many congestion avoidance routing schemes have been proposed so for. In literature routing is divided into two main categories including congestion aware routing and congestion adaptive routing. The former is further divided into cross layer MAC protocols and Rate control protocols.

A. CONGESTION AWARE ROUTING PROTOCOL

These protocols consider congestion during route establishment phase. Selected route doesn't change unless intermediate nodes move to other location or link is broken. It include cross layer MAC and rate control protocols.

1) CONGESTION AWARENESS BASED CROSS LAYER MAC PROTOCOLS

Hung-Yun *et al.* has proposed a new MAC protocol that has a per-flow notion of fairness for channel access and achieves improved end-to-end throughput fairness. Also, a new load balanced routing algorithm was proposed that improves fairness even when the underlying MAC is fair with respect to flows [18]. Lei Chen et al. has proposed a QoS-aware routing protocol that incorporates an admission control scheme and a feedback scheme to meet the QoS requirements of real-time applications. QoS-aware routing protocol uses approximate bandwidth estimation and takes the operation to network traffic [19]. K. R. Vinod *et al.* has proposed a congestion control algorithm (SPCC). This algorithm uses PEER approach for selection of shortest path from source to destination. Two parameters are used to calculate link cost during path establishment. These parameters are named as transmission power and receiving power. Congestion is controlled using path cost. Suburah *et al.* present cross layer based QoS routing (CBQR) for congestion control and route stability. It includes bandwidth aware, congestion aware and QoS based cross layer architecture of network. The protocol works on physical, MAC and network layer when data is transmitted from source to destination, the source node chooses the path which satisfies load and link capacity. By using link information, congestion is avoided [20].

2) RATE CONTROL ORIENTED PROTOCOLS

In this type of protocols, congestion is avoided based on the transmission rate. For this purpose, network status is shared with sending node so that if there is any congestion on intermediate node or if there any bottleneck link is created somewhere in middle then sending node reduces its sending rate so that congestion is avoided. Many techniques have been proposed in literature for rate control mechanism, some of them are listed below; Soundararajan and Bhuvaneswaran [21], has proposed a mechanism called multipath load balanced and rate based congestion control (MLBRCC). In this technique, the destination node transmits network information to the application which then adjusts its sending rate according to network conditions. There are so many other techniques in literature in which sender adjusts its data rate according to network condition. These schemes include linear message rate integrated control avoids congestion by taking benefit of precision control which are by default available in wireless. Rate effective network utility maximization (RENUM) works by decreasing data rate of link from source to destination. The framework attaches network utility with destination node instead of sender [22].

Tuan Anh Le *et al.* have proposed a multipath protocol based on energy aware and congestion control. This protocol avoids all congested and high energy consuming paths and selects lighter paths. The property of multipath connection is that it can transmit multiple flows simultaneously. The proposed algorithm measures the energy for both data transmission and data reception between two end hosts. Sender side calculates end to end energy consumption while at receiver side data and ACK cost is calculated [23]. Energy aware congestion control for multipath TCP (ecMTCP), transfers traffic from most congested paths to lesser loaded paths. It also transfers from higher energy cost to lower energy cost paths. This technique gets load balancing and energy saving paths. Sheeja *et al.* has proposed an effective congestion avoidance scheme for mobile ad hoc networks [24]. The scheme includes three steps as follows; i) Network monitoring to obtain congestion status; ii) Congestion detection based on queue length, channel contention and overall congestion status by observing number of packets drop; iii) Avoid all those paths which have congestions and develop a congestion free route from source to destination. The scheme improves packet delivery ratio and network throughput by minimizing delay. According to author the probability of packets in queue is computed using (1) where L_{offk} represents the offered load at the queue of node k.

$$P(Q) = \left(1 - L_{offk}\right) L_{offk}^{1} \tag{1}$$

Packet loss rate is given in (2) where t_1 and t_2 are starting and ending time respectively. Packet drop ratio is calculated as given in (3) where P_{dn} represents number of packets dropped, P_{mn} is number of packet misrouted and P_{tn} represents total number of transmitted packets [24].

$$P_{LR}(t_1, t_2) = \frac{\int_{t1}^{t2} 1\{G(t) - D_t\}^{dF(t)}}{\int_{t1}^{t2} dF(t)} \tag{2}$$

$$PDR = \frac{P_{dn} * P_{mn}}{P_{tn}} * 100. \tag{3}$$

Li Xia *et al.* has improved existing AODV by adding mechanism for congestion control and route repair in RREQ packets. Queue size is maintained on each node. According to buffer size intermediate node can judge the congestion by measuring busy degree on a node. If the node is idle, then RREP is sent immediately through that node. Otherwise, it discards RREP packet [25]. Sankaranarayanan has introduced early detection congestion and control routing protocol (EDAODV). This algorithm aims to provide alternate path when congestion occurs in a bidirectional manner i.e. both forward and reverse direction of congested node. There are three phases of this algorithm (i) route discovery (ii) congestion detection at early stages (bidirectional path discovery). Each node maintains two routing tables one is primary routing table (PRT) which is maintained during primary path establishment phase for different destinations while other is alternate routing table. (ART) which maintains alternate paths by corresponding an entry to the PRT [26].

Dynamic congestion detection and control routing (DCDR) [27] is a mechanism which reduces congestion by setting congestion free paths at initial phase of route establishment phase. This algorithm configures all congestion free paths by using CFS which is at one or two hop neighbor. In [28], a prediction based control mechanism is presented that takes the intelligent decisions based on existing knowledge and set of parameters. IRED [29] is an enhanced version of RED algorithm [30] which was developed by Elloumi *et al.* to improve MANET efficiency. RED is based on active queue management scheme in which network informs destination node about congestion level. On the contrary, IRED uses priority queue based on active queue management. Packet drop in this scheme is due to two factors including incoming data rate and length of queue. Number of packet drops are reduced and congestion is reduced. Existing techniques for route discovery rebroadcast route request packets until the desired path is established to destination node. But these scheme results in broadcast storm problem when data is transmitted from source to destination. It causes congestion at intermediate nodes. Early detection of congestion and self-cure AODV routing protocol (EDCSCAODV) is an enhancement of traditional AODV on the basis of active queue management where routes are computed on individual node. This scheme is able to detect congestion on early stages and transmits an alerts message to all neighbor nodes. On receiving network information neighbor nods detect a congestion free path is selected. This scheme reduces network delay also improves packet delivery ratio [31]. Sharma *et al.* [32] propose a hybrid mechanism of rate control and resource control to resolve congestion in network. During the process of route finding every node forwards RREQ packet to its neighboring node. If a congested node is found in the path to destination all neighbor nodes are informed by sending control message. In this process all requesting nodes are informed by setting a flag value in order to choose alternate path. In Data Rate Adaptive (DRA) [33] scheme, the queue lengths of the nodes are analyzed for path selection and adapting the data rate to avoid congestion. The main issue is that the queue length of each node is exchanged with neighboring nodes in a periodic manner that causes communication overhead and may result in complicating the congestion scenarios.

III. BANDWIDTH-AWARE ROUTING STRATEGY

In this section, we have presented an efficient **B**andwidth **A**ware **R**outing **S**trategy (BARS) for identifying path between sender and receiver. We have analyzed the available bandwidth and residual queue size to decide about recommending a suitable bandwidth value for data exchange. We have worked for following new features in the AODV.

1) Ability to estimate the residual bandwidth. All nodes along the path are capable to know their available resources in terms of bandwidth
2) Informs source node about current network conditions in terms of residual bandwidth so that source node can adjust its transmission rate accordingly.
3) The route recovery process immediately performs route recovery whenever there is a broken route in network.

To accommodate the above mentioned features, the packet format is changed. For example, to implement quality of service, some new fields are added into packet format. These fields are added to RREQ and RREP packets in order to carry out the bandwidth information. The major difference between the proposed methodology and other mechanisms based on AODV is the implementation of adaptive feedback method. Because of this, the source node easily finds out the current network state, links capacity and adjusts its data rate accordingly. To implement this, all nodes along the path must know their available bandwidth on the links. We have divided our proposed work in two phases as illustrated in figure 3 and a list of notations is presented in table 1.

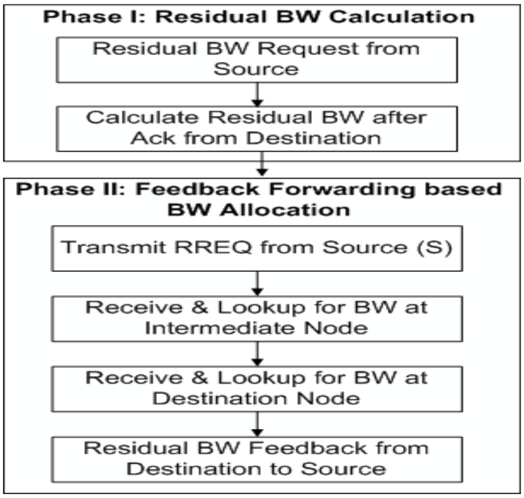


FIGURE 3. Main phases for BARS with residual bandwidth calculation and then feedback forwarding based bandwidth allocation.

TABLE 1. List of notations for BARS.

Notation	Description
N_i	Sender Node
N_j	Neighboring Node
Ts, Tr	Sending and Receiving Time
Ack	Acknowledgement
P_{size}	Total Packet Size
T	Time difference of *Tr* and *Ts*
$\widehat{BW}_{Res}$	Residual Bandwidth
$BWcon$	Bandwidth Consumed
t_i	Time Period for Bandwidth
sid	Session Identity
α	Weightage of Residual Bandwidth
Dest	Destination Node
Sn_id	Sensing Node ID
seq_num	Sequence Number
Last_BW	Last Bandwidth used at Source
BW_Rec	Bandwidth Calculated on Ack Receive
hop_{count}	Hop count in the path

A. PHASE I - RESIDUAL BANDWIDTH CALCULATION

During first phase, residual bandwidth is calculated on each node which is based on two steps; In first step, any node in the network N_i locally calculates residual bandwidth by sending residual bandwidth request to each neighbor node Nj and saves the value of transmission time and cost of transmission messages. Whereas in the second step weightage average of previous residual bandwidth is calculated in order to inform the source node regarding the latest residual bandwidth available in network at each link. Using this information source node can adjust its sending data rate in order to avoid congestion.

1) RESIDUAL BANDWIDTH REQUEST FROM SOURCE

Before starting the process of bandwidth calculation, each node finds its one hop neighbor by sending Hello message. When a node receives a hello message, it creates entry in its routing table. To maintain connectivity with neighbor nodes, every node transmits control messages to its neighbors. If neighboring node does not send back any control packet within the specific time interval, a hello message is broadcasted locally to its neighbors. If a node does not receive any hello message within the specified time interval, it means that the neighbor is no more in its transmission range and connection to its neighbor has been lost. Hello message uses two variables named HELLO INTERVAL and ALLOWED HELLO LOSS to determine the connectivity between node and its neighbor nodes. HELLO INTERVAL specifies the maximum time interval between transmissions of hello messages. ALLOWED HELLO LOSS determines the number of intervals in which a node waits for receiving hello messages without breaking connectivity to neighbor node. Recommended values for HELLO INTERVAL is one second and for ALLOWED HELLO LOSS is two seconds. It means that if a node is unable to receive a hello message from the neighbor node within two seconds of last hello message, then connection to that node is lost. These hello messages are also used to calculate residual bandwidth between a node and its one hop neighbor. To calculate the residual bandwidth on each node, it records the sending time (Ts) hello message. Each node attaches value of Ts in packet header and transmits it to its directly connected neighbor.

When the hello message reaches at the directly connected neighbor, it extracts header information from packet. If it is desired destination for which the hello packet was sent, it updates its session cache, otherwise it drops the packet. In order to notify the sender node that its packet has successfully reached at desired node, the neighbor node transmits an acknowledgement packet back to the sender. It also attaches the value of time to send Ts in the packet header. After that, residual bandwidth Ack is transmitted from destination.

2) RESIDUAL BANDWIDTH ACK AT SOURCE

Residual bandwidth estimation is performed on the sender node. It is assumed that the residual bandwidth is equal to maximum throughput between two directly connected nodes. When sender node receives HELLOACK packet, it records its packet receiving time (Tr). Now the sender calculates residual bandwidth on the link between itself and its one hop neighbor node as $BWRes = P_{size}/T$ where P_{size} is the total size of HELLO packet. It also includes the size of other MAC messages which are transmitted to the directly connected node as given in (4). In the equation, P_H and P_{H-Ack} represent sizes of HELLO and HELLOACK packets along with RTS and CTS requests generated individually for these packets.

$$P_{size} = 2(P_{RTS} + P_{CTS}) + P_H + P_{H-Ack} \tag{4}$$

In this scenario, T is the total round trip time in which the Hello message is sent and the HELLOACK is received. It can

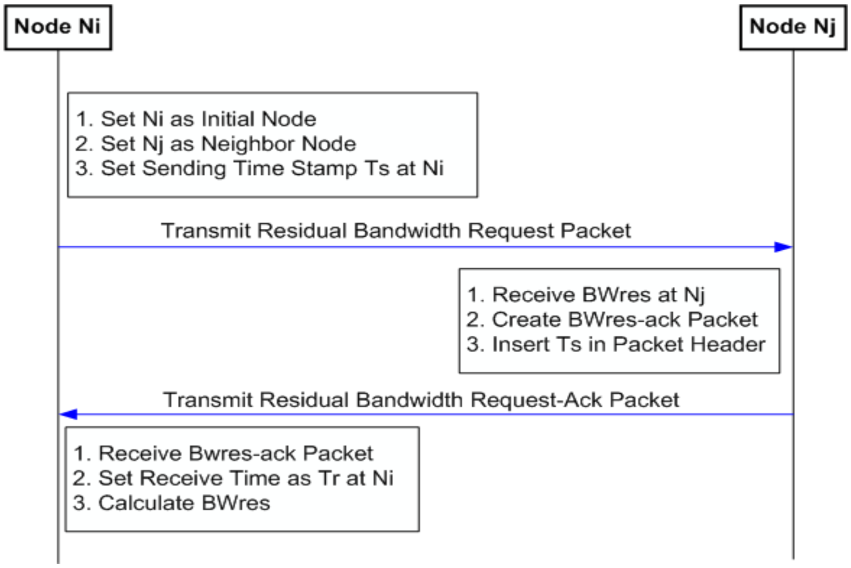


FIGURE 4. Estimation of residual bandwidth between neighboring nodes Ni and Nj.

be calculated as $T = Tr - Ts$ where Tr is the time when HELLOACK is received at sender node and Ts is the time when HELLO packet is sent by the sender. If the node is not initiator, its packet is dropped. Otherwise the packet is released after consulting or updating the routing table of neighbors.

It also calculates the delay between the send hello and receive *Ack* at initiator and finding the residual available bandwidth as illustrated in figure 3. Moreover, protocol decides about suitable route by measuring queue size and bandwidth availability of neighboring nodes as presented in algorithm 1. Residual queue size is extracted to verify the residual queue on each node in the path. It helps to add an extra amount to data that may reside in the queues and improves bandwidth utilization.

Algorithm 1 Residual Bandwidth Calculation at Node Ni

1. N_i: Set source id (sid) and destination id (destid) {N_i is intermediate node}
2. N_i: if the immediate cache update takes place then set communication type to Unicast else to Broadcast
3. N_i: set values in reply header
4. $N_i \rightarrow N_j$: transmit residual bandwidth request and save values of transmission time and cast of number of message retransmissions
5. N_j: receive message & extract data
6. $N_j \rightarrow N_i$: transmit residual bandwidth response
7. N_i: receive response message and calculate delay between N_i & N_j
8. N_i: calculate Queue utilization based Residual Bandwidth
9. N_i: set Bandwidth between N_i and N_j

B. PHASE II: FEEDBACK FORWARDING BASED BANDWIDTH ALLOCATION

In this phase, it is assumed that the route request (RREQ) packet contains the minimum required bandwidth which is sent from source node. BRREQ is retransmitted back from destination to the source node in order to create the reverse route. An intermediate node is also important to know whether it is part of route to the destination node or not. It only transmits data packet if the requested bandwidth is less than the residual bandwidth on the link. In this way appropriate route is created to transmit the packets based upon the available bandwidth so that congestion is avoided.

1) TRANSMISSION OF RREQ PACKET FROM SOURCE

In BARS, route is selected based on source requirement. Source indicates the minimum requested bandwidth that must be guaranteed, in RREQ packet. This packet with requested bandwidth titled as Bandwidth-oriented Route Request (BRREQ). The new packet with bandwidth extension includes session id (*sid*) with source address. Session id uniquely identifies each flow and is incremented by timer each time when new RREQ is generated. When a node receives a BRREQ packet, reverse route is created with *sid* and BRREQ is rebroadcasted. The same procedure continuous until BRREQ reaches at destination.

2) RECEIVE AND LOOKUP FOR BANDWIDTH REQUEST AT INTERMEDIATE NODE

Each node which receives BRREQ packet, it first extracts headers from the received packet. When BRREQ packet is received by an intermediate node, it checks either it has route to destination or not. If receiving node has route to destination, it then compares the values of requested bandwidth with residual bandwidth of each node. If requested bandwidth is less than node residual bandwidth, intermediate node forwards data packet. This can be done by sending immediate cache update message to the last node. This message includes bandwidth value of next link which is greater than previous link. If there are two paths to destination with same distance and one of the outgoing link is being used by multiple senders, then there are chances of packet drop on that link. In this situation, node transmits packet on link with higher bandwidth to avoid congestion.

3) RECEIVE AND LOOKUP FOR BANDWIDTH REQUEST AT DESTINATION NODE

After receiving a BRREQ packet, the node extracts headers from the packet. In this case, if receiving node is destination then it performs two calculations as follows. Firstly, end-to-end residual bandwidth $\widehat{BW}_{Res}(t_i)$ is calculated along the path as given in (5) where $I\widehat{BW}_{Res}(t_i)$ represents weighed average of residual bandwidth on the path. Actual value of residual bandwidth at time period t_i is represented as $BW_{Res}(t_i)$. Moreover, the weighted average of last residual bandwidth values is represented as $I\widehat{BW}_{Res}(t_i - 1)$. In this case, α is used to set the weightage of current residual bandwidth which is set to 0.8 or 80% in our case. On the contrary, the value of $(1 - \alpha)$ is 0.2 which is used for taking weightage of previous average of values for residual bandwidth. It explores that higher priority is assigned to the current residual bandwidth to take the impact of current bandwidth value. Equation (6) explores to use $BW_{Res}(t_i)$ at start when $t_i = 0$ and calculate $I\widehat{BW}_{Res}(t_i)$ for $t_i > 0$.

$$I\widehat{BW}_{Res}(t_i) = \alpha BW_{Res}(t_i) + (1-\alpha) \times I\widehat{BW}_{Res}(t_i - 1) \tag{5}$$

$$\widehat{BW}_{Res}(t_i) = \begin{cases} IBW_{Res}(t_i) & t_i > 0 \\ BW_{Res}(t_i) & t_i = 0 \end{cases} \tag{6}$$

Secondly, the destination node also estimates the suitable value of bandwidth to be consumed ($BWcon$) by the source node. It is calculated in order to check if each node on the path can support the requested bandwidth from source node. To calculate consumed bandwidth, it is important to take into account the intra flow interference which is also called mutual interference. The parameter used in BARS to calculate intra flow interference is hop count (HC). HC is determined by measuring the distance of each node from source to destination along the path. The highest value of HC ($HCmax$) along the path is considered for calculation of consumed bandwidth. After finding the value of $HCmax$, the consumed bandwidth is calculated at destination node as $BWcon = HCmax \times BWreq$ where $BWreq$ is requested bandwidth by the source, the destination node compares the value of consumed bandwidth with end to end residual bandwidth. In case when the consumed bandwidth is greater than residual bandwidth, then it informs source node to reduce data rate as *BWavl/HCmax*.

4) BANDWIDTH FEEDBACK FROM DESTINATION TO SOURCE NODE

After performing all calculations of bandwidth, the destination node transmits bandwidth based route reply (BRREP) packet to source node. While passing through each intermediate node, all nodes verify current residual bandwidth with the bandwidth contained in the packet header. If the residual bandwidth is less than bandwidth given in packet header, then node replaces its value with the residual bandwidth. Each node forwards BRREP packet on reverse route back to the source node. The process of packet forwarding on reverse path is same as used in packet path from source to destination. Each node checks reverse path entry in the routing table. The node forwards the packet on desired path if it has a reverse path entry in the routing table. BRREP packet including feedback is checked at each node if it is the desired recipient (source node). If yes, then extract all headers from the packet. Now the source node compares the value of requested data rate with feedback sent by the destination node in terms of consumed bandwidth and adjusts its data rate. In single flow, data rate sent by the source node remains the same. This process greatly helps in avoiding the congested network paths. As a result, the communication of messages takes place in a faster way. End-to-end delay is reduced and throughput is increased. The scheme also decreases the packet loss and enhances the packet delivery ratio. For the end-to-end bandwidth estimation, we have utilized the residual bandwidth and residual queue sizes at each intermediary on the path as illustrated in algorithm 2.

IV. IMPACT OF CACHING, COMPUTATION AND COMMUNICATION

In this section, we explore the impact of Caching, Computation and Communication (CCC) over a number of performance metrics. In the similar vein, network bandwidth is a precious resource for IoT data transmission, saving the bandwidth and optimizing the transmission of data is a one of the key success for adopting IoT in all domains. Caching can be a viable solution for saving communication cost by reusing the sensed data and the information stored in the cache, as long as the information is not outdated and valid. Packet delivery ratio (PDR) is an evaluation metric used to estimate the ratio of packets received to the total send packets. Reducing the communication cost by using computing to consume data locally while also eliminate sending duplicate data by implementing smart caching strategies will reduce the amount of bandwidth consumed, decrease the packets drop, and increase the PDR. For example, if 3000 packets were sent by sensing devices in a region and due to sensing bottleneck 1200 packets are dropped then PDR $= (1800/3000)*100 = 60\%$. Due to caching, if data is sustained at sensing devices and unique values are aggregated before transmission, then number of packet drops can be reduced. In some cases, only 150 packets were dropped that results in PDR $= (2850/3000)*100 = 95\%$.

Network throughput is an important indicator of the performance and the quality of the network connection. Throughput indicates the ratio of successful packets delivery on the network. Dropping packets and network congestion lower the throughput and the quality of the network. CCC can maximize the throughput by minimizing the number of the chunk data sent over the network as the duplicated data is eliminated

Algorithm 2 End-to-End Bandwidth Calculation

```
S: Assign requested BW and related parametric values
S: if sn_id AND Last bandwidth are NULL then
    if sn_id & Dest Not in RT via Lookup then
        Set requested BW and last_BW to 0
    else Discard the Request endif
    else
    Lookup for Data Rate for the Destination in RT
    Set requested BW and last_BW to Data Rate
endif
S → N_i: Transmit RREQ message towards Dest
N_i : Lookup route between currNode and Dest
N_i : if Dest is not neighbor AND currNode != Dest then
        Find neighbor where Residual BW > Requested BW
    else if Dest is not neighbor AND currNode not Dest
        Set Bandwidth of Node by Look up in neighbor List
    else if Dest is not neighbor
        Set bandwidth of node by Lookup in Neighbor List
    end if
N_i: If CurrNode is Not Dest then
        If sn-id and Dest found via Lookup then
            Save Dest, source, sn_id last_BW in repository
            Update last_BW by Lookup for source
            Set src_adrs to CurrNode,
            Set hop count, seq_num
        Else
            Update sn_id and set src_adrs to CurrNode
            Update hop count and seq num.
        End if
    Else if CurrNode is Dest
        If sn_id and Dest ID found via Lookup then
            Update seq_num and last_BW
            Save Dest and src ID, sn_index and last_BW
            Send response to source where BW=lastBW
        Else
            BW_Rec=residualBW/hop_count
            BW_Rec = Evaluate, Adjust residual Queue size
            If BW_Rec is less than Requested BW then
                Update seq_num, Send to source
                Set BW to BW_Rec
        Else
        Update seq_num, save Dest and src ID, sn_index
        Send response to source where BW=BW_rec
        End if
    End if
S: lookup for route to Dest and transmit packet
```

before the data aggregation operation, also it minimizes the size of the data packet by utilizing smart computing functions. Throughput = Total_Bytes_Sent / unit_time. In case of PDR, 1800 packets were received that shows 1800*256 = 460, 800 bytes received in 6.73 seconds where packet size is 256 bytes. In this case throughput = 460, 800/6.73 = 68.46 KB per second. Due to caching, 729600 bytes is received in 6.73 seconds as we have 2850 received packet sized 256 bytes each. The throughput = 729600/6.73 = 108.41 KB per second.

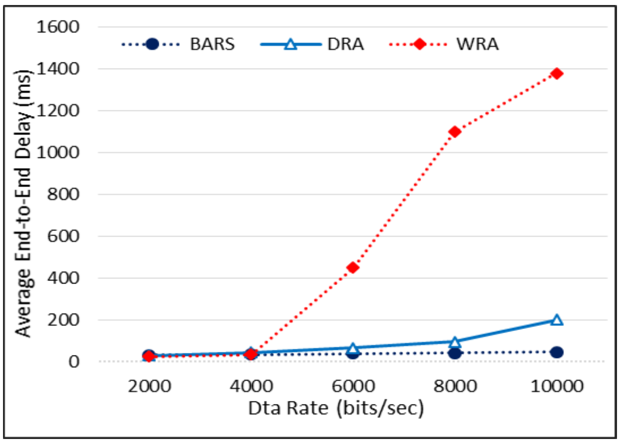
Energy consumption is improved through CCC during data exchange and congestion avoidance. Due to CCC, a number of packets will be cached for a threshold value of time to avoid sensing bottleneck at physical world and congestion on the network entities or servers in the cyber world. Our CCC model considers caching at first level and then data is exchanged in the aggregated and de-duplicated format. It reduces communication overhead by sharing less number of messages towards servers that reduces energy consumption as well. The cost of sending one bit on wireless network is 50 Nano joules. If 3000 packets were sent by a sensor in an ideal environment, the total consumed energy is (1800*8)*50 = 0.0012 joule. In the case without caching, we noticed 1200 dropped packets that consumed (1200*8)*50 = 0.000048 Joule which is about 40% of consumed energy. Hence, 150 dropped packets after integrating caching costs (150*8)*50 = 0.00006 J which is 5% of total energy consumed.

V. FORMAL MODELING AND ANALYSIS OF BARS

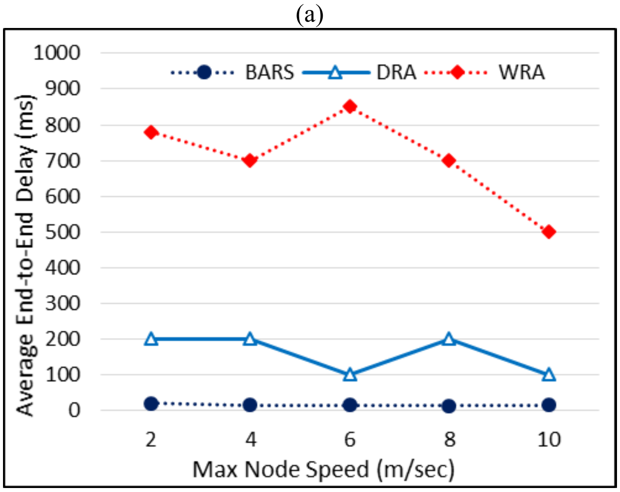
We have formally verified estimation of residual bandwidth between the neighboring nodes N_i and N_j of our proposed BARS using Rubin Logic [34]. It ensures the modeling which is considered to be near to the real implementation using some coding language like C in our case. During formal modeling, a number of sets are maintained for storing the related parameters like ID_{N_i}, $BW_{N_{ij}}$ and QF_{N_j} in POSS() called possession sub set. Sender N_i contains and receiver N_j maintains ID_{N_j}, $BW_{N_{ji}}$, QF_{N_i} in POSS(). Next is the BEL() which is belief sub set to store parameters required during algorithmic or other related calculations. A list of notations is presented in table 2. In Rubin Logic, a global set comprises of entities, roles of entities, and global variables. Next, a local set is maintained that contains the POSS(), BEL() and behavior list BL() subsets as illustrated in table 3. We have analyzed the operations performed between N_j and N_j as per sequence of execution in implementation in C language. During modeling phase, a number of functions are utilized that include send, receive, generate sequence number, set life time, set packet type, set destination address and port. For the analysis, we have compared storage requirements for POSS() and BEL() subsets. During modeling, local level parameters are considered in POSS() set. After processing, the scope of

TABLE 2. List of notations for local set in BARS.

Sr.	Notation	Description
1.	$BW_{N_{ij}}$	Bandwidth Available between N_i and N_j
2.	QF_{N_j}	Residual Queue Factor for neighbor N_j
3.	s	Initial Value for Sequence Number
4.	SQ_{N_i}	Sequence Number at Source N_i
5.	LT_{N_i}	Life Time of Packet set by Source N_i
6.	ΔT	Allowed Interval for a Packet to Expire
7.	Ψ	Packet Loss Factor
8.	C_T	Current Time
9.	D_A	Destination Address
10.	TS_{N_i}	Time Stamp
11.	P_D	Destination Port
12.	PA_{ktNi}	Packet Sent by Source Node N_i
13.	M_1	Message sent from N_i to N_j
14.	CT_{N_j}	Current Time at N_j



(a)



(b)

FIGURE 5. N average end-to-end delay for variation in (a) data rates and (b) maximum node speed.

parameters is expired using forget() operations. It includes the parameters like SQ_{N_i}, LT_{N_i}, PT_{N_i}, D_A, TS_{N_i} and P_D that are expired after message sharing and processing.

TABLE 3. Local set for BARS at sender Ni and receiver Nj.

1. Sender (N_i)

POSS(N_i) = $\{ID_{N_i}, BW_{N_{ij}}, QF_{N_j}\}$
BEL(N_i) = { #(ID_{N_i}), #($BW_{N_{ij}}$), #(QF_{N_j}) }
BL(N_i) =
GenSeQNum (s) → SQ_{N_i}
SetLifeTime (Δt, ψ) → LT_{N_i}
SetPType() → PT_{N_i}
SetDestAdrs() → D_A
TimeStamp(C_T) → TS_{N_i}
SetDestPort() → P_D
Send ($PA_{kt_{Ni}}$) → M_1
Update (M_1)

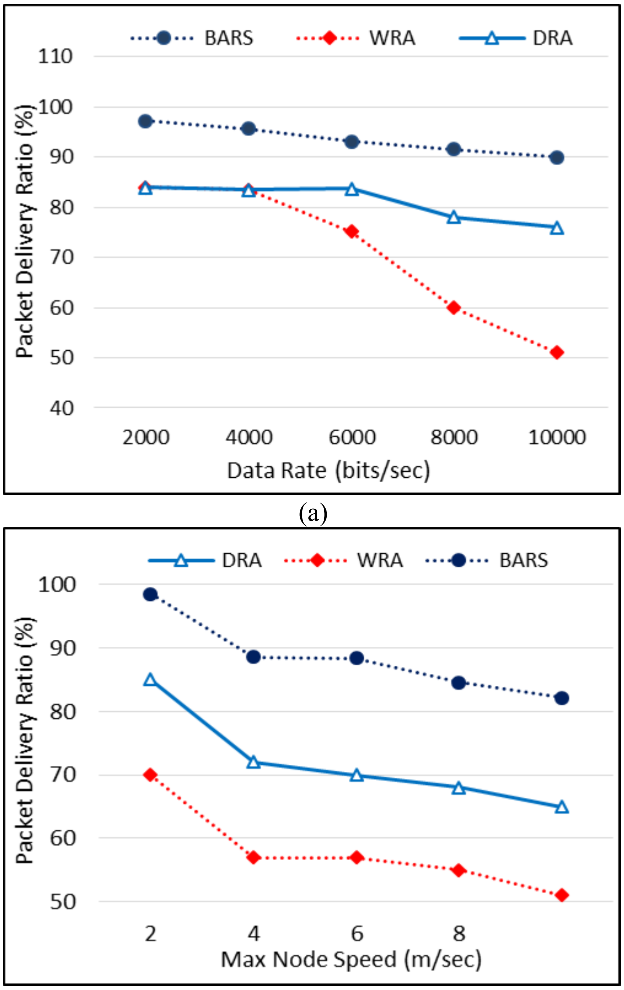
Receive (M_2) to get [SQ_{N_j}, TS_{N_j}, QF_{N_j}]
VerifyLifeTime $\left(\left(CT_{N_i} - LT_{N_j}\right) \geq \Delta T\right) \rightarrow Abort$
SetBW(TS_{N_j}, QF_{N_j})

2. Neighbor (N_j)

POSS(N_j) = $\{ID_{N_j}, BW_{N_{ji}}, QF_{N_i}\}$
BEL(N_j) = { #(ID_{N_j}), #($BW_{N_{ji}}$), #(QF_{N_i}) }
BL(N_j) =
Receive (M_1) to get [SQ_{N_i}, LT_{N_i}, PT_{N_i}, D_A, TS_{N_i}, P_D]
VerifySeQNum ()
VerifyLifeTime $\left(\left(CT_{N_j} - LT_{N_i}\right) \geq \Delta T\right) \rightarrow Abort$
IncrSeQNum () → SQ_{N_j}
TimeStamp(C_T) → TS_{N_j}
SetQFactor() → QF_{N_j}
Send ($PA_{kt_{Nj}}$) → M_2
Update (M_2)

VI. RESULTS AND ANALYSIS

In this section, we discuss simulation setup, working of proposed algorithm and simulation results. Network simulator NS2.35 has been used to test the performance of proposed scheme. The radio propagation model is Two Ray Ground while Queue type is Drop tail with maximum length of 50 packets. Transmission range is set to 250m and interference range is set as 550m. Type of traffic flow is constant bit ratio (CBR) which streams over UDP with packet size of 1024bits. Bandwidth aware routing is achieved by modifying the existing implementation of AODV in C. Initially, the queue size is monitored to test the queue utilization during congestion. We have also measured the residual bandwidth and estimated bandwidth and then shared to the above layers. Simulation scenarios can be categorized in two scenarios; First scenario consists of random deployment of static nodes where number of nodes are varied from 30 to 50 with



(a)

(b)

FIGURE 6. Packet delivery ratio for variation in (a) data rate and (b) max node speed.

10 sender receiver pairs. In second scenario, mobility is considered with 10 senders to transmit a regular traffic of CBR using UDP and hence 10 UDP-receivers. We have tested the performance of proposed scheme by changing nodes speed from 2m/sec to 50 m/sec. Parameters of proposed BARS are shown in Table 4. By varying data rate and nodes speed, we have compared our work with DRA [33] and without rate adaption (WRA) strategies.

A. NETWORK MODEL OF PROPOSED BARS

Simulation scenario consists of 40 to 50 nodes placed in random manner. There are 10 send and receive pairs initially and then we have increased them up to 50 percent of total nodes. We have performed a number of simulations on both static and mobile network topologies. We have evaluated our algorithm using two simulation scenarios. The first scenario includes random deployment of nodes varying from 30 to 50 nodes with 10 sender receiver pairs. While in second scenario we add mobility in nodes. We have compared our work with DRA [33] and WRA by varying data rate and nodes speed.

TABLE 4. Simulation parameters for BARS.

Parameter	Value
Simulation Time	1000 s
Number of nodes	50
Network Region	1000m × 1000m
Transmission Range	250 m
Packet size	1024 b
Mobility model	Random way point
Traffic type	Constant bit rate (CBR)
MAC	IEEE802.11
Source-destination pairs	10 – 50%
Data rate	2048 – 10,240 bits/sec
Max speed	1 – 50 m/s

B. AVERAGE END TO END DELAY

Figure 5(a) illustrates the impact of data rate on end-to-end delay in case of static deployment of nodes. Delay increases by increase in packet loss. So increase in packet loss increases number of re-transmissions which results in increase in end –to-end delay. Packet loss in DRA scheme is high so it resulted in high packet loss. While in proposed scheme we have reduced number of packet drop by controlling data rate on basis of available bandwidth. This data rate controlling method avoids congestion which ultimately reduces delay in network. It elucidates that for a data rate of 8000 bits/sec, average end to end delay is 1100ms, 100ms and 14ms in static topology of WRA, DRA and BARS respectively. Figure 5(b) elucidates the end-to-end delay for mobile nodes where number of packet drop increases due to dis-connectivity between wireless nodes. Our proposed scheme provides solution of path break rapidly and reconnects nodes. That is why number of packet drop reduces which reduces delay in network. In case of dynamic topology, for a node speed of 8m/sec, end to end delay is 700ms, 200ms and 18ms in WRA, DRA and BARS respectively. In this scenario BARS dominates the counterparts in both static and dynamic scenarios.

C. PACKET DELIVERY RATIO (PDR)

Increase in data rate effects negatively on packet delivery. In DRA scheme when data rate increases packet loss become higher due to frequent overflow of queues. While in proposed scheme we have adopted data rate on the basis of available bandwidth rather queue. We have observed that number of packet drop in proposed scheme is much less than that of DRA scheme. Figure 6(a) elucidates the impact of data rate on packet delivery ratio for static nodes. In case of static topology for a data rate of 8000 bits/sec, packet delivery ratio is 60% in WRA and 78% in case of DRA while in our proposed scheme it is 90%. In case of mobility for DRA scheme, each node needs to store data packets in queue for longer time before forwarding. So chances of data drop become higher. Although this scheme provides data rate adaption but when nodes moves with higher speed this scheme fails. On the contrary, proposed scheme provides rapid path re-establishment right after dis-connectivity between mobile nodes. Figure 6(b) illustrates dynamic environment for packet delivery ratio with respect to maximum node speed. It indicates that for a node speed of 8m/sec, packet delivery ratio is 55% and 69% for WRA and DRA respectively whereas in case of our proposed scheme it is 80%. Results show that our proposed scheme dominates the counterparts in case of both static topology and dynamic topology.

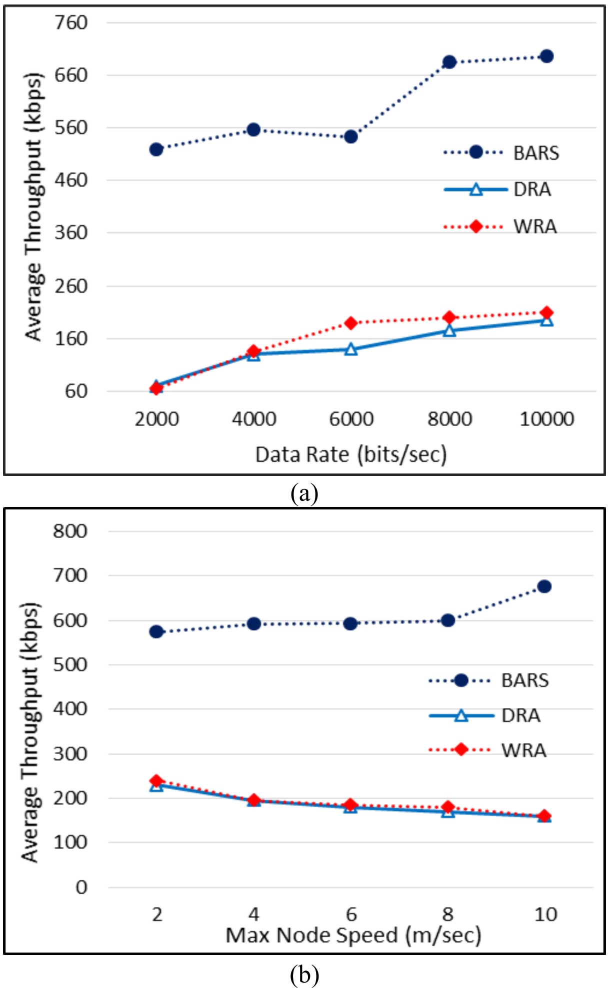


FIGURE 7. **Average throughput for different (a) data rates and (b) max node speed.**

D. THROUGHPUT

Figure 7(a) elucidates the throughput in static scenario where data rate is varied from 2000 to 1000 bits per second. Results show that by increase in data rate throughput also increases. Throughput is also inversely proportional to congestion in network. In proposed scheme, we have handled congestion successfully that's why throughput of our proposed scheme is much greater than DRA. Results depict that throughput is 210 kbps, 160 kbps and 670 kbps in WRA, DRA and BARS respectively for a data rate of 8000 bits/sec. In case of mobility, nodes change their position frequently. Sending nodes become unable to find destination node which resulted in less throughput. Figure 7(b) illustrates that for a node speed of 8m/sec, the average throughput is 200 kbps in case of WRA and 190 kbps for DRA scheme while it is 600 kbps in case of BARS respectively. Results proves the dominance of BARS in case of static and mobile nodes respectively.

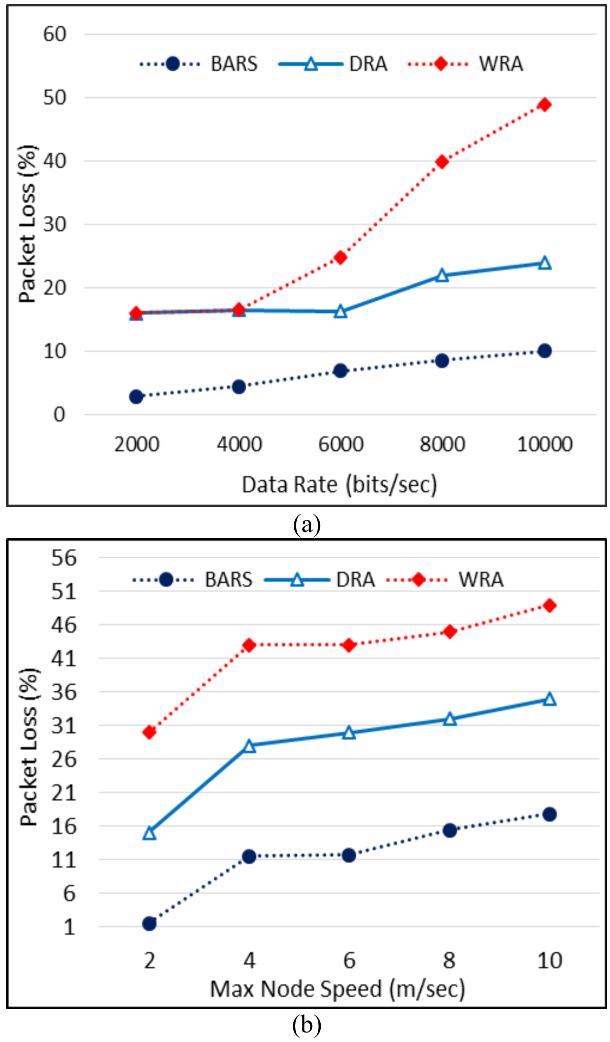


FIGURE 8. **Packet loss percentage for (a) data rate variations and (b) maximum speed of node.**

E. PACKET LOSS

Packet loss occurs when sending node becomes unable to forward packet to destination node. Congestion is one of main reasons of packet loss. Figure 8(a) illustrate the packet loss with respect to change in data rate. In DRA scheme packet loss is high because nodes store data packets in queue when data rate is increased, queues become overflow although data rate adaption mechanism is adapted but it does not provide any mechanism if congestion occurs due to link capacity.

While in proposed scheme we have taken link bandwidth as a parameter to avoid occurrence of congestion. We observed that packet loss is much less than DRA. In case of static topology for a data rate of 8000 bits/sec, packet loss ratio is 40%, 21% and 9% in WRA, DRA and BARS respectively. Figure 8(b) elucidates the results for dynamic topology in which packet loss ratio is 46%, 31% and 21% in WRA, DRA and BARS respectively for a node speed of 8m/sec. Results indicates that packet loss in proposed methodology is much lesser than its counterparts.

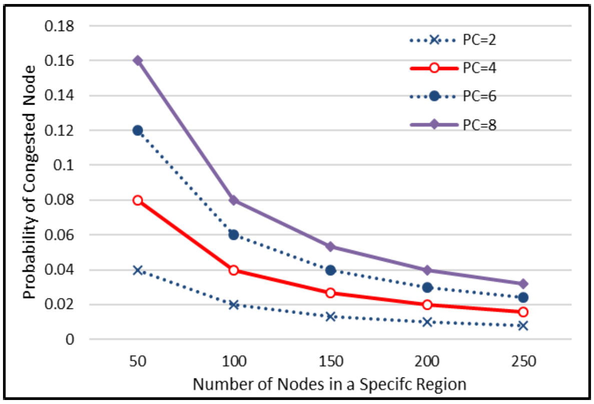


FIGURE 9. Probability that a certain node is under scongestion.

F. PROBABILITY OF CONGESTION AT NODE

Number of nodes can be affected due to congestion and more number of packets are dropped through that nodes. All the paths that contain these nodes also suffer from information loss. We have measured the probability that a particular node is under congestion Pr_C when a total of c nodes are congested. Figure 9 elucidates the probability of task failure for a number of tasks initiated. It illustrates that for 300 tasks assigned during a particular time frame, probability of task failure is 0.01333, 0.02666, 0.04 and 0.0533 when total congested nodes are 2, 4, 6 and 8 respectively.

$$Pr_C = 1 - \binom{N-1}{c-1} \Big/ \binom{N}{c} = \frac{c}{N} \tag{7}$$

VII. CONCLUSION

The proposed scheme presents the bandwidth aware routing where the current network status is evaluated in terms of residual bandwidth and residual size of interface queue to cache the information. Packet from source to destination node can suffer from mutual interference between packets in the same flow that can cause congestion. To manage the congestion, Hello messages are used to predict residual bandwidth on a link by taking average of most recent values of residual bandwidth and cached entries in queues at intermediaries by checking queue sizes. Next, the estimation of consumed bandwidth involves the maximum amount of bandwidth that each node can support. On the basis of residual bandwidth, the source node transmits data packets accordingly and manages re-establishment of broken links by adopting capabilities of caching, computation and communication. The algorithm detects route break during neighbor discovery when Hello message is not received at neighbor node. To check the impact of proposed BARS, a number of simulations have been carried out to check its performance using NS 2.35. C language has been used to implement functionality for bandwidth estimation, rate control and queue management. Perl scripts are used to get results from a number of trace files obtained for existing and proposed schemes. Results show that the proposed BARS scheme outperforms counterparts in terms of packet delivery ratio, end-to-end delay, packet loss and throughput and probability for existence of congested node for static and dynamic scenarios. In future, we shall include quality of service factors like energy aware route selection in combination with bandwidth estimation.

REFERENCES

[1] G. A. Walikara and R. C. Biradar, "A survey on hybrid routing mechanisms in mobile ad hoc networks," *J. Netw. Comput. Appl.*, vol. 77, pp. 48–63, Jan. 2017.
[2] A. Čolaković and M. Hadžialić, "Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues," *Comput. Netw.*, vol. 144, pp. 17–39, Oct. 2018.
[3] K. Y. Ajay and T. Sachin, "QMRPRNS: Design of QoS multicast routing protocol using reliable node selection scheme for MANETs," *Peer-to-Peer Netw. Appl.*, vol. 10, no. 4, pp. 897–909, Jul. 2017.
[4] K. D. Santosh and T. Sachin, "Energy efficient routing protocol for MANET based on vague set measurement technique," *Procedia Comput. Sci.*, vol. 58, pp. 348–355, Aug. 2015.
[5] I. U. Din, M. Guizani, S. Hassan, B.-S. Kim, M. K. Khan, M. Atiquzzaman, and S. H. Ahmed, "The Internet of Things: A review of enabled technologies and future challenges," *IEEE Access*, vol. 7, pp. 7606–7640, 2019.
[6] T. A. Krupa and M. S. Benakappa, "A survey: Routing protocols in MANETs," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 2, no. 7, Jul. 2014.
[7] M. Anuradha and G. S. A. Mala, "Cross-layer based congestion detection and routing protocol using fuzzy logic for MANET," in *Wireless Networks*, vol. 23, no. 5. New York, NY, USA: Springer, 2017, pp. 1373–1385.
[8] R. Vadivel and V. M. Bhaskaran, "Adaptive reliable and congestion control routing protocol for MANET," in *Wireless Networks*, vol. 23, no. 3. New York, NY, USA: Springer, 2017, pp. 819–829.
[9] M. Geetika, G. Mahesh, and K. C. Umesh, "A survey on congestion control in MANET," *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 2, pp. 101–998, Mar. 2014.
[10] G. Ali, "Real-time routing algorithm for mobile ad hoc networks using reinforcement learning and heuristic algorithms," in *Wireless Networks*, vol. 23, no. 3. New York, NY, USA: Springer-Verlag, 2017, pp. 703–714.
[11] A. A. Sarfaraz, K. T. Senthil, A. S. S. Syed, and S. Subburam, "Cross-layer design approach for power control in mobile ad hoc networks," *Egyptian Inform. J.*, vol. 16, no. 1, pp. 1–7, Mar. 2015.
[12] R. C. Debarati, R. Leena, and M. Nilesh, "Implementing and improving the performance of AODV by receive reply method and securing it from Black hole attack," *Procedia Comput. Sci.*, vol. 45, pp. 564–570, Mar. 2015.
[13] M. A. Bouras, A. Ullah, and H. Ning, "Synergy between communication, computing, and caching for smart sensing in Internet of Things," *Procedia Comput. Sci.*, vol. 147, pp. 504–511, Jan. 2019.
[14] R. Mamata, P. R. Umesh, P. Niharika, K. N. Surendra, and P. P. Sambhu, "Congestion control mechanism for real time traffic in mobile adhoc networks," in *Computer Communication, Networking and Internet Security*. Singapore: Springer, vol. 5, 2017, pp. 149–156.
[15] N. Umapathi, N. Ramaraj, D. Balasubramaniam, and M. R. Adlin, "An hybrid ant routing algorithm for reliable throughput using MANET," *Intell. Comput. Appl.*, vol. 343, pp. 127–136, Feb. 2015.
[16] M. Abinasha B. M. Manab, and S. Bobby, "A congestion control algorithm for mobility model in mobile ad-hoc networks," *Int. J. Comput. Appl.*, vol. 118, no. 23, pp. 27–31, May 2015.

[17] K. M. Saleem, S. Waris, I. Ali, M. H. Anisi, and M. I. Khan, "Mitigation of packet loss using data rate adaptation scheme in MANETs," *Mobile Netw. Appl.*, vol. 23, no. 5, pp. 1141–1150, Dec. 2018.

[18] H. Hung-Yun and R. Sivakumar, "Improving fairness and throughput in multi-hop wireless networks," in *Semantic Scholar*, Jul. 2001, pp. 569–578.

[19] L. Chen and W. B. Heinzelman, "QoS-aware routing based on bandwidth estimation for mobile ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 3, pp. 561–572, Mar. 2005.

[20] S. Sharma, D. Jindal, and R. Agarwal, "An approach for congestion control in mobile ad hoc network," *Int. J. Emerg. Trends Eng. Develop.*, vol. 3, no. 7, pp. 217–223, 2017.

[21] S. Soundararajan and R. S. Bhuvaneswaran, "Multipath load balancing & rate based congestion control for mobile ad hoc networks (MANET)," in *Proc. Int. Digital Inf. Commun. Technol. Appl. (DICTAP)*, Bangkok, Thailand, May 2012, pp. 30–35.

[22] S. Guo, X. Zhu, and Y. Yang, "Optimal and distributed resource allocation in lossy mobile ad hoc networks," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Shanghai, China, Apr. 2013, pp. 1744–1749.

[23] A. L. Tuan, S. H. Choong, S. Lee, H. Jung, and A. Razzaque, "ecMTCP: An energy-aware congestion control algorithm for multipath TCP," *IEEE Commun. Lett.*, vol. 16, no. 2, pp. 275–277, Feb. 2012.

[24] S. Sheeja and R. V. Pujeri, "Effective congestion avoidance scheme for mobile ad hoc networks," *Int. J. Comput. Netw. Inf. Secur.*, vol. 1, pp. 33–40, Jan. 2013.

[25] L. Xia, Z. Liu, Y. Chang, and P. Sun, "An improved AODV routing protocol based on the congestion control and routing repair mechanism," in *Proc. WRI Int. Conf. Commun. Mobile Comput.*, Yunnan, China, Jan. 2009, pp. 259–262.

[26] T. S. Kumaran and V. Sankaranarayanan, "Early detection congestion and control routing in MANET," in *Proc. 7th Int. Conf. Wireless Opt. Commun. Netw. (WOCN)*, Colombo, Sri Lanka, Sep. 2010, pp. 1–5,

[27] T. S. Kumaran and V. Sankaranarayanan, "Dynamic congestion detection and control routing in ad hoc networks," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 25, no. 1, pp. 25–34, Jan. 2013.

[28] S. Hariom and K. M. Pradeep, "Congestion control using predictive approach in mobile ad hoc network," *Int. J. Soft Comput. Eng. (IJSCE)*, vol. 3, no. 4, pp. 2231–2307, Sep. 2013.

[29] S. Simaiya, A. Shrivastava, and N. P. Keer, "IRED algorithm for improvement in performance of mobile ad hoc networks," in *Proc. Int. Conf. Commun. Syst. Netw. Technol.*, Bhopal, India, Apr. 2014, pp. 283–287.

[30] O. Elloumi and H. Afifi, "RED algorithm in ATM networks," in *Proc. IEEE ATM Workshop*, Lisboa, Portugal, May 1997, pp. 312–319.

[31] T. S. Kumaran and V. Sankaranarayanan, "Early congestion detection and self cure routing in MANET," in *Computer Networks and Information Technologies* (Communications in Computer and Information Science), vol. 142. Berlin, Germany: Springer, 2011, pp. 562–567.

[32] A. Sharma and K. Pawar, "CRAODV: A new approach for congestion control in mobile ad-hoc network," in *Proc. Symp. Colossal Data Anal. Netw. (CDAN)*, Indore, India, Mar. 2016, pp. 1–5,

[33] M. S. Khan, M. H. Anisi, S. Waris, and M. I. Khan, "Data rate adaptation strategy to avoid packet loss in MANETs," in *Proc. 1st Int. Conf. Future Intell. Veh. Technol.*, Porto, Portugal, Jan. 2016, pp. 57–66.

[34] A. D. Rubin and P. Honeyman, "Nonmonotonic cryptographic protocols," in *Proc. Comput. Secur. Found. Workshop (CSFW)*, Jun. 1994, pp. 100–116.

NOUSHEEN AKHTAR received the BSSE degree from Fatima Jinnah Women University, Rawalpindi, Pakistan, in 2012, the MSSE degree from the Department of Computer Engineering, College of EME, National University of Science and Technology (NUST), Rawalpindi, in 2017, where she is currently a Visiting Lecturer. She was a Visiting Lecturer with APCOMS Islamabad, Pakistan, from 2013 to 2016. Her research interests include wireless networks, mobile ad-hoc networks, routing mechanisms, and fog computing.

MUAZZAM A. KHAN received the master's degree with a major in mobile ad-hoc networks from IIUI and the Ph.D. degree in computer sciences as sandwich program from IIUI and UMKC, USA, in 2011, the Ph.D. degree from the University of Ulm, Germany, in 2013, and the Ph.D. degree from the University of Missouri, KC, USA, in 2016. He is currently a Tenured Associate Professor and the Associate Dean of the Department of Computing, SEECS, National University of Sciences and Technology, Islamabad, Pakistan. He joined Awk University Mardan as an Assistant Professor/Chair of the CS Department, in 2011. He has been with the School of Computer and Electrical Engineering, University of Missouri, USA, and the School of Computer Science, University of Ulm, Germany, as Postdoctoral Fellow. He joined the National University of Sciences and Technology Islamabad as an Assistant Professor, in 2013. He was with the Networking and Multimedia Lab, University of Missouri, Kansas, USA. His research interests include the IoT, wireless networks sensors, body area networks, image processing, image compression, image encryption, and data network security.

ATA ULLAH received the B.S. and M.S. degrees in computer science, respectively, from the COMSATS Institute of Information Technology (CIIT), Islamabad, Pakistan, in 2005 and 2007, and the Ph.D. degree in computer science from IIUI, Pakistan, in 2016, in the area of wireless network security. He is currently an Assistant Professor with the Department of Computer Science, National University of Modern Languages (NUML), Islamabad, since 2008. From 2007 to 2008, he was a Software Engineer with Streaming Networks, Islamabad. He joined the NUML, where he was a Lecturer/Head Project Committee, from 2008 to 2015, and an Assistant Professor with the Department of Computer Science, from 2015 to 2017. From 2017 to 2018, he was a Research Fellow of the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. He has rejoined NUML, since 2018. He has supervised 112 projects at under graduate level. He has secured positions in one international and 45 national level software competitions. He has received ICT funding for the development of projects. He has authored 37 papers in peer-reviewed ISI-indexed impact factor journals. He is also a reviewer and a guest editor of journals and international conference papers. He has programming expertise in C, C#, Java, PHP, and NS2. His areas of interests include WSN, the IoT, cyber physical social thinking (CPST) space, health-services, NGN, VoIP, and their security solutions.

MUHAMMAD YOUNUS JAVED received the B.E. degree from UET at Lahore, Pakistan, in 1983, and the M.Sc. and Ph.D. degrees from the University of Dundee, U.K., in 1988 and 1991, respectively. He is currently a Professor and the Dean of the Faculty of Engineering and Technology, Heavy Industries Taxila Education City, Taxila, Pakistan, since 2015. From 1991 to 2015, he was with the College of Electrical and Mechanical Engineering (C-EME), National University of Science and Technology (NUST), as the Dean Faculty of Engineering and C-EME. He has published several research papers in well-reputed ISI-indexed journals and conferences. His research interests include distributed and parallel systems, digital image processing, algorithmics, and computer networks.