Wireless Netw (2017) 23:819–829
DOI 10.1007/s11276-015-1137-3

Adaptive reliable and congestion control routing protocol for MANET

R. Vadivel[1] · **V. Murali Bhaskaran**[2]

Published online: 16 January 2016

Abstract In mobile ad hoc networks (MANETs), the packet loss can be caused either by link failure or by node failure. Moreover, the techniques for selecting the bypass route and avoiding congestion in the bypass route are rarely handled. To overcome these, in this paper, we propose an adaptive reliable and congestion control routing protocol to resolve congestion and route errors using bypass route selection in MANETs. The multiple paths are constructed. Among which, the shortest paths are found for efficient data transmission. The congestion is detected on the basis of utilization and capacity of link and paths. When a source node detects congestion on a link along the path, it distributes traffic over alternative paths by considering the path availability threshold and using a traffic splitting function. If a node cannot resolve the congestion, it signals its neighbors using the congestion indication bit. By using simulation, we show that that the proposed protocol is reliable and achieves more throughput with reduced packet drops and overhead.

Keywords Mobile ad hoc networks (MANETs) · Quality of service (QoS) · Routing protocol · Congestion control

✉ R. Vadivel
rvadivelphd@gmail.com

[1] Department of IT, School of CSE, Bharathiar University, Coimbatore 641 046, India

[2] Paavai College of Engineering, NH-7, Pachal, Namakkal, Tamil Nadu 637 018, India

1 Introduction

The performance level of a service provided by the network to the user is the quality of service (QoS). Most of the multimedia applications have strict QoS requirements which have to be satisfied. Achieving more deterministic network behavior is the main aim of QoS provisioning, where the information from the network can be delivered in a better way and the resources of the network can be efficiently utilized. But, providing QoS solutions and maintaining end-to-end QoS with end user mobility are the major challenges in MANET [2, 11, 22, 37].

QoS routing requires a route which satisfies the end-to-end QoS requirement in terms of bandwidth or delay, but not only to find a route from source to destination. Calculating paths which are suitable for different type of traffic is generated by various applications while maximizing the utilizations of network resources. The following are the major objectives of QoS routing [3, 12, 29, 38]:

- To find a path from source to destination that satisfies users requirements.
- To optimize the usage of the network.
- To reduce the network load when unwanted things like congestion and path breaks appear in the network.

Maximizing the data packet delivery in the fast changing network topology devoid of incurring a large routing overhead is the major issue in MANET. The packet delivery ratio is reduced when broken links are encountered, which cause the data packets to be forwarded to stale or invalid paths.

The main reason for packet loss in ad hoc networks is the link failure or node failure. MAC failure is caused if there is more than one packet using this link after the physical layer failure. Although the routing protocol takes

off such packets from the queue after a failure, new packets still keep on coming into the queue. If the new incoming packets are forwarded for the failed link, they will block all other packets, thereby resulting in a network wide low throughput and long delays [5, 7, 16, 35]. Detection of such failures in an ad hoc network becomes challenging due to the lack of a centralized monitoring and management point. This causes the faulty nodes to remain for a long time in the network, which affects the performance of routing in the ad hoc network. For instance, if a defective node participating in the routing process drops data packets, subsequently a large number of packets will be lost [6, 13, 30, 39].

Congestion is a state occurring in some part of a network when the message traffic is so heavy, which slows down network response time. In wired networks, routing is handled by routers in the backbone network. The routers buffer incoming packets before sending them on their paths towards their destinations. Hence, congestion may occur in such networks when the buffers of the multi-port routers start filling up and may end up dropping packets. This problem arises when a router can receive traffic on multiple input ports at a higher rate than that it can forward [4, 28, 40].

In wireless networks, nodes utilize the CSMA/CA channel access technique, and a wireless device acting as a router in a MANET can receive traffic from one neighboring device at a time to reduce the load on the routing process on the device. A device can receive multiple packets before it can forward one packet due to the busy medium, but this is not expected to create a buffering problem at the device [10].

Network congestion is the major problem in the mobile wireless ad hoc networks. Due to limited availability of resources and the nature of the wireless network, congestion is the common issue in MANET. In such networks, due to shared wireless channel and dynamic topology, packet transmissions suffer from interference and fading. When bandwidth is exhausted, large amount of real time traffic is likely to build up and lead to congestion. Congestion causes packet loss, bandwidth degradation, and time and energy wastage. Even if the influence of congestion is reduced, it is not possible to completely overcome the congestion problem. By using some suitable procedures and rules for traffic flow, the influence of congestion can be reduced [23, 34].

TCP congestion control works very well on the Internet. The congestion control mechanisms are affected greatly by some of the unique properties of MANETs. For standard TCP, the widely differing environment in a MANET is highly problematic. Generally, congestion is concentrated on a single router when it occurs in the Internet. On the other hand, due to the shared medium, congestion in MANET affects the whole area [25].

Due to congestion, no single sender is able to collapse the network deliberately due to the comparatively low bandwidth of MANETs. The potentially severe unfairness between traffic flows is due to the effect of a single traffic flow on the network condition. When compared with the conventional wired networks such as Internet, wireless multihop networks are prone to overload-related problems. Hence, it is essential to have an appropriate congestion control protocol for the sake of network stability and acceptable performance [14, 15].

Explicit link failure notification (ELFN) is a cross-layer proposal in which TCP interacts with the routing protocol to detect route failure. ELFN messages are sent back by the routing protocol to the TCP sender from the node detecting the failure. EFLN messages have sender and receiver addresses and ports along with TCP sequence number. Hence, this TCP distinguishes the losses caused by congestion from the ones due to mobility. The fixed retransmission timeout protocol has routing error recovery in addition to the routing algorithm in which route failure is indicated whenever two successive retransmissions due to timeout occur. Hence, the TCP sender retransmits at regular intervals. Adhoc TCP (ATCP) relies on the ICMP protocol and on the explicit congestion notification (ECN) scheme to detect network partition and congestion. When three duplicate ACKs are detected, ATCP puts TCP in "persistent mode" and retransmits the lost packet from buffer. Hence, TCP congestion control is not aroused when it is not really needed. When network congestion is detected by the receipt of an ECN message, ATCP does nothing but forwards the packet to TCP so that it can invoke its normal congestion control mechanism [8, 32, 33].

2 Related work

Peng et al. [18] have presented a multi-rate multicast congestion control scheme for MANETs in order to achieve high fairness with TCP, robustness against misbehaving receivers, and traffic stability, without changing the queuing, scheduling, or forwarding policies of existing networks. This scheme only introduces very limited control traffic overhead by the on-the-spot information collection and rate control.

Karunakaran et al. [9] have presented a cluster based congestion control protocol to support congestion control in ad hoc networks with scalable and distributed cluster-based mechanisms. Their approach is based on the self-organization of the network into clusters. The congestion can be autonomously and proactively monitored by clusters

within their localized scope. Adjustment of node rates and cooperation between cluster nodes can be obtained by exchanging small amount of control packets. They have used clustering process to discover the communications between the flows. Their approach can improve the responsiveness of the system than end-to-end techniques.

Rahman et al. [20] have proposed an explicit rate-based congestion control mechanism. In their approach, throughput can be increased rapidly by allowing the routers to give explicit feedback. Using the explicit information in the feedback packets from the routers, sender's flow can be controlled. Their protocol has better performance than the conservative behavior of the TCP and TCP like protocols for multimedia streaming over the MANETs.

Nishimura et al. [17] have discussed a routing protocol in which they have used multi-agents to control network congestion for MANET. They have engaged two kinds of agents in routing. One of their agents is a routing agent (RA) which gathers information regarding network congestion and link failure. Another agent is a message agent (MA) which uses the information gathered by RA to reach the destination nodes. MAs with respect to the data packets can discover the direction using an evaluation function.

Akyol et al. [1] have proposed a wireless greedy primal dual (wGPD) algorithm for combined congestion control and scheduling in MANET. In wGPD, the scheduling decisions are taken in two phases: intra-node scheduling and inter-node scheduling. In intra-node scheduling, each node decides which packet is to be transmitted whenever it is next allowed to make a transmission. In inter-node scheduling, transmissions that interfere compete among themselves to determine who should transmit the next packet. They have defined two types of congestion control: an unreliable version and a reliable version. The first version is apt for the standard UDP protocol and used for flows tolerating loss. The second version is apt for the TCP protocol and ensured that all data are uniformly delivered to the destination.

From the above analysis of existing works, we can conclude that the works on resolving congestion in an alternate (or) by-pass route is rarely handled.

3 Problem identification and solution

In [27], we have developed an adaptive reliable routing protocol using combined link stability estimation for MANET. The main objective of this protocol is to determine a QoS path for prolonging the network life time and reducing the packet loss. We calculate a combined weight value for a path based on the parameters link expiration time, node remaining energy, node velocity, and received signal strength to predict the link stability or lifetime. When a link is likely to be broken, the previous node will cache the subsequent packets in its data buffer. When a link failure actually occurs, the upstream node with the cached data in its buffer can retransmit it through the next reliable link by using a bypass route.

It can be noticed that when the link failure occurs, the upstream node with the cached data in its buffer can retransmit it through the next reliable link by using a bypass route and a fault tolerance technique. However, the technique of choosing the bypass route and the way to avoid congestion in the bypass route are not handled in [19].

In this paper, we propose a technique for selecting the bypass route to resolve the congestion. When a source node detects congestion on a link along the path, it distributes traffic over alternative paths. The congestion is detected according to the utilization and capacity of link and paths. The distribution of traffic considers the path availability threshold and utilizes a traffic splitting function. If a node cannot resolve the congestion, it signals its neighbors using the congestion indication bit, which is not used in existing congestion control schemes.

4 Routing protocol using combined link stability estimation

4.1 Measuring the signal strength

The received signal strength in cross layer design can be calculated at the physical layer, and it is accessed at the top layers. The procedures at the physical layers have to be modified in order to reassign the measured value of received signal strength to the MAC layer along with the signal [21]. In addition, this value is stored in the routing/neighbour tables and is used in the decision-making process. The received signal strength is passed to the top layers as an interlayer interaction parameter. The received signal strength is used to improve the performance of the network by adjusting the medium access and routing protocols as per the required cross layer design.

The IEEE 802.11 is reliable MAC protocol. Since the received signal strength must reach every exposed node, it assumes a fixed maximum transmission power. When a sending node transmits RTS (Ready to Send) packet, it attaches its transmissions power level. The receiving node measures the signal strength received for free-space propagation model while receiving the RTS packet [21].

$$P_R = P_T(\lambda/4\pi d)^2 G_T G_R \quad (1)$$

where λ is the wavelength of carrier, d is distance between sender and receiver, P_R and P_T are the receiving power and transmitting power, respectively. G_T and G_R are unity gain

of transmitting and receiving omni-directional antennas, respectively.

4.2 Route expiration time (RET)

The RET is the minimum time selected from a set of link expiration times (LETs) designed for the feasible path. LET [19] is the period of link connectivity between two nodes. So, the minimum value of LET is attained in each path and the maximum number of RET, which represents the more reliable routing path, is selected.

$$RET_i = \text{Min}(LETs_i) \tag{2}$$

Thus, the RET is the minimum value among LETs of the feasible path.

The principle of LET is to estimate future disconnection time with the help of two neighbors in motion. This can be achieved by using the following method. A global positioning system (GPS) can determine the motion parameters of two neighboring nodes. The following assumptions are made; a free space propagation model whose signal strength solely depends on the distance to the transmitter, and all nodes have their clocks synchronized using the GPS clock. The duration of time can be calculated for the two nodes which remained connected by having knowledge of their motion parameters. These parameters which are obtained from the GPS include speed, direction, and radio range.

Given a prediction T_p on the continuously available time for an active link between two nodes at time t_0, the availability of this link, L (T_p), is defined as

$$\text{L}(T_p) = \text{P}\{\text{to stay valid to } t_0 + T_p | \text{Available at } t_0\}$$

indicating the probability that the link will be continuously available from time t_0 to $t_0 + T_p$.

The calculation of L (T_p) can be divided into two parts: the link availability when the velocities of the two nodes stayed unchanged between t_0 and $t_0 + T_p$, L_1 (T_p), and the one for the other cases, L_2 (T_p). That is,

$$L(T_p) = L_1(T_p) + L_2(T_p) \tag{3}$$

It is easy to calculate L_1 (T_p), which is equal to the probability that the epochs from t_0 onwards for the two nodes are longer than T_p, because T_p is an accurate prediction if the movements of the two nodes remain unchanged. Since nodes' movements are independent of each other, and the exponential distribution is 'memory less,' L_1 (T_p) is given by

$$L_1(T_p) = \left[1 - E(T_p)\right]^2 = e^{-2\lambda T_p} \tag{4}$$

However, it is difficult to give an accurate calculation for L_2 (T_p) because of the difficulties in learning about the changes in link status that are caused by changes in a node's movement.

4.3 Node's remaining energy

It is assumed that all nodes are equipped with a residual power detection device and know their physical node position. The transmitting energy for a packet can be computed as

$$Energy_{tx} = \frac{P_{size} \times Power_{tx}}{LBW} \tag{5}$$

where P_{size} is the data packet size, $Power_{tx}$ is the packet transmitting power, and LBW is the wireless link bandwidth. When a mobile node performs power control during packet transmission, the transmitting energy for one packet relative to the node distance is given as

$$Energy_{tx} = kd^{\alpha} \tag{6}$$

where k is the proportionality constant, d is the distance between the two neighboring nodes, and α is a parameter that depends on the physical environment (generally between 2 and 4).

For calculating the shorter distance between the transmitter and the receiver, the smaller amount of energy required. At each node, the total required energy is given by

$$Energy_{tot} = p \times \left(Energy_{tx} + Energy_{pro}\right) \tag{7}$$

where p is the number of packets. The energy required for packet processing ($Energy_{proc}$) is much smaller than that required for packet transmitting.

The node remaining energy or the residual energy is the energy left after the packet transmission (i.e.) residual energy $Energy_{res}$ is given by

$$Energy_{res} = Energy_{initial} - Energy_{tot} \tag{8}$$

4.4 Node velocity

Consider a node N_1 which can always communicate with another node N_2 until it reaches position p_3, which is at a distance R from N_1, where R is the transmission range of each node.

We also define $T_p = Tp_1p_3$, where Tp_1p_3 is the time taken by node N_1 to move from p_1 to p_3.

Knowing T_p, the probability that a link is available during the time period of T_p can be calculated. Assume that random walk-based mobility model defines the movement of a node. Every node in the network moves at a constant speed in a constant direction in a time duration referred as mobility epoch. Thus, the epoch length of every node is exponentially disseminated with mean λ^{-1}. The route is evenly distributed over [0, 2π] and the speed is also uniformly distributed in a known range. It is assumed that speed, direction, epoch length, and mobility of nodes are uncorrelated and the links

fail autonomously. A conservative prediction of the link being available in the time period of T_p is given based on the assumptions above as in [26]

$$L_2(T_p) = \frac{1 - e^{-2\lambda T_p}}{2\lambda T_p} + \frac{\lambda T_p e^{-2\lambda T_p}}{2} \quad (9)$$

A metric is needed to reflect this aspect and whose value should lie in the range [0, 1] for accounting the reliability of a link while selecting routes. It is impossible to combine the L (T_p) of each link along a path for estimating the path availability, because each link has a different T_p.

The term $T_p \times$ L (T_p) is used in the place of L (T_p) which is an evaluation of average available time of a link. We can restrict our interest in the estimation within T_r, by assuming that estimation will be carried out regularly with period T_r. The ratio of $T_p \times$ L (T_p) to T_r is very much concerned. A new routing metric is developed based on the above reasoning which is referred to as normalized link availability (LA_N). It can be defined as follows [26]:

$$LA_N = \min\left[\frac{T_p x L_2(T_p)}{T_r}, 1\right] \quad (10)$$

4.5 Combined metric

Finally, the combined metric for finding the link lifetime is given by

$$CRM = P_R + L_1(T_P) + Energy_{res} + LA_N \quad (11)$$

The node status is adaptively determined based on the value of CRM as given below.

Node status is Green, if $CRM_{\min} < CRM < CRM_{\max}$
Node status is yellow, if $CRM = CRM_{\min}$
Node status is red, if $CRM < CRM_{\min}$

where CRM_{min} and CRM_{max} are the maximum and minimum combined routing metric values.

Then, Bypass Route Discovery is performed as described in our previous work [27]. When a link is likely to be broken, the previous node will cache the subsequent packets in its data buffer, and then when a link failure occurs, the upstream node with the cached data in its buffer can retransmit it through the next reliable link by using a bypass route.

5 Proposed bypass route technique to resolve congestion

5.1 Multipath construction

Consider the network as a directed graph $G = (V, E)$, where V denotes the set of nodes and E represents the set of links or edges. In our technique, we propose to find multiple paths in terms of shortest and disjoint paths.

5.1.1 Shortest paths

The shortest path discovery algorithm discovers exclusive shortest paths from source S to destination D. Let sP_{sd} be the shortest paths between nodes S and D. Let N_S be the neighbor set of node S. Then, the alternative shortest paths can be estimated by using the formula,

$$sP_{sd}^{s-1} = \{S\} \oplus sP_{s-1,d} \quad (12)$$

Where, $s - 1$ is the neighbor node of S and $\oplus$ stands for concatenation of two finite sub paths. The shortest path $sP_{s-1,d}$ starts at node S and connects destination D through neighbor node $s - 1$. However, this technique of finding alternate paths through neighboring nodes may bring in the problem of routing loop.

To ensure the constructed shortest routes are loop-free, sP_{sd} must guarantee that it does not contain S as an intermediate node. Thus, any shortest path must prove the condition given in Algorithm-1.

Algorithm-1

Assumption: S and D as the source and destination nodes, respectively

sP_{sd} be the set of shortest paths between nodes S and D

1. If ($S \notin$ nodes ($sP_{s-1,d}$)) then
2. The alternate path sP_{sd}^{s-1} will be loop free
3. Else
4. The alternate path sP_{sd}^{s-1} will be routing loop path
5. End if

The computed alternate paths from node S and D are kept in the set $A_{s,d}$. Paths that are included in set $A_{s,d}$ can be represented as,

$$A_{s,d} = \left\{sP_{sd}^{s-1} : s - 1 \in N_s,\ S \notin sP_{sd}^{s-1}\right\} \quad (13)$$

Thus, the set $A_{s,d}$ contains multiple alternate paths that connect S and D.

5.1.2 Disjoint paths

To make multipath routing efficient and consistent, it must satisfy the criteria of disjoint paths. In $A_{s,d}$, any two paths can be disjoint if and only if it solves the following conditions.

Algorithm-2

Assumption: Let S and D denotes the source and destination nodes respectively

Let sP_{sd}^{s-1} and sP_{sd}^{s-2} be two paths in the alternate path set $A_{s,d}$

1. Find common nodes in sP_{sd}^{s-1} and sP_{sd}^{s-2}
2. If the only common nodes are S and D then
3. The paths will be disjoint paths
4. Else if it has common nodes other than S and D then
5. The paths will be interconnected paths
6. End if

Thus, disjoint paths between S and D can be given as,

$$D_{s,d} = \{sP_{s,d}\} \cup A_{s,d} \tag{14}$$

5.2 Conjoint-c paths

In this approach, multiple paths are generated by defining a set $cP_{s,d}$. It includes the paths that contain more conjoint (common) links. Let P_1 and P_2 be two paths, these paths can be added to the set $cP_{s,d}$ if it satisfies the following condition,

$$|l(P_1) \cap l(P_2)| \leq c \tag{15}$$

Where l stands for links and c represents common links. We can define the set $cP_{s,d}$ at various levels,

At Conjoint level-0, the conjoint set $cP_{s,d}$ contains disjoint paths that connect S and D
At Conjoint level-1, $cP_{s,d}$ encompass paths that has at least a common link
At Conjoint level-c, $cP_{s,d}$ set contain paths with c common links

Finally, the conjoint property can be defined as,

$$cP_{s,d}^{c=0} \subseteq cP_{s,d}^{c=1} \subseteq \ldots \subseteq cP_{s,d}^{c=c} \subseteq cP_{s,d} \tag{16}$$

The algorithm for generating $cP_{s,d}$ set is given below in Algorithm-3

Algorithm-3

1. Initialize $cP_{s,d}$ to $sP_{s,d}$
2. Paths in $A_{s,d}$ are arranged in ascending order of their path length
3. Further paths for the set $cP_{s,d}$ is selected from the set $A_{s,d}$
4. While ($A_{s,d} \neq 0$) do
5. Find out shortest path sp in $A_{s,d}$
6. Remove sp from $A_{s,d}$
7. Break
8. Check path sp against all paths in $A_{s,d}$
9. Check whether it satisfies conjoint condition (equation-15)
10. If sp satisfies condition (equation-15) for all paths in $cP_{s,d}$ then
11. Add sp to the set $cP_{s,d}$
12. End

5.3 Load balancing

By using Algorithm 1, we develop a method to resolve the network congestion by distributing the traffic over group-G multipath routes. Node N calculates the multipath routes to destination D when it detects congestion on a local outgoing link L_n for which the path p_{ND} contains link L_n. Some part of the traffic to node D is then transferred to alternative paths $p \in P^G$.

Presume that every node (say node N) in the network measures utilization of its local outgoing link L_n. The link utilization is denoted as L_{Ut} and path utilization is symbolized as P_{Ut}. The utilization of path p_1 is computed as, [24]

$$P_{Ut}(p_1) = \min\{L_{Ut}(L_n) : L_n \in P^G\} \tag{17}$$

Both path and link utilizations are calculated by each node considering network or routing information forwarded by its neighboring nodes as,

$$P_{Ut}(p_{s,d}) = \min\{L_{Ut}(L_{s,i}), P_{Ut}(p_{j,d})\} \tag{18}$$

If the utilization of a local link exceeds a local congestion threshold T_H then the congestion is detected in the local link. Here, T_H is a predefined threshold value. The main goal is, by shifting a portion of the traffic to the alternative paths the utilization has to be reduced adequately and this part of traffic as bypass traffic.

A node calculates a set of alternative paths and distributes the bypass traffic over these paths whenever it detects local link congestion or receives an Absolute Congestion Index (ACI) bit from a neighbor. A node produces signals to its neighbors using the ACI bit when a node is not able to solve the congestion in the former method. The following is the procedure for congestion-triggered multipath traffic distribution to resolve congestion at the local link.

1. When node N detects local congestion on outgoing link (l, $P_{UT}(L_n) > T_H$), or receives an $ACI = 1$ bit from a neighbor node on link l, it tries to move bypass traffic onto alternative paths that avoid this link. The alternative paths are group-G paths determined using Algorithm 1.
2. Let SP = {$p \in P^G$: $P_{UT}(p) < \mu$}, where μ is called the path availability threshold.
3. Distribute traffic over the path set SP.

For its active path P_a, the node N load balances its current traffic on the entire available path. Let the current load of node N be 100 kbit/s. Let P_1, P_2, P_3, and P_4 be the available paths in SP. Node N will use the multiple paths P_1, P_2, P_3 and P_4 in addition to its primary path P_a on reception of Explicit Congestion Indication (ECI) and will

send on each of these paths with equal amount of data rate (i.e.) 100/4 = 25 kbit/s of data.

6 Simulation results

6.1 Simulation model and parameters

We use NS2 [31] to simulate our proposed technique. In our simulation, the channel capacity of mobile hosts is set to the same value: 2 Mbps. We use the distributed coordination function (DCF) of IEEE 802.11 for wireless LANs as the MAC layer protocol. It has the functionality to notify the network layer about link breakage.

In our simulation, we keep the number of mobile nodes as 150. The mobile nodes move in a 1000 m^2 × 1000 m^2 region for 50 s simulation time. We assume each node moves independently with the same average speed. All nodes have the same transmission range of 250 m. In our simulation, the speed is 10 m/s. The simulated traffic is Constant Bit Rate (CBR). The CBR flows are varied as 2, 4, 6 and 8.

Our simulation settings and parameters are summarized in Table 1.

6.2 Performance metrics

We evaluate mainly the performance according to the following metrics.

- *Average end-to-end delay* The end-to-end-delay is averaged over all surviving data packets from the sources to the destinations.
- *Average packet delivery ratio* It is the ratio of the number of packets received successfully and the total number of packets transmitted.
- *Drop* It is the average number of packets dropped.
- *Throughput* Throughput is defined as the total number of routing control packets received successfully during the transmission.

Table 1 Simulation parameters

No. of nodes	50,75,100,125 and 150
No. of flows	2, 4, 6 and 8
Area size	1000 × 1000 m
Mac	802.11
Radio range	250 m
Simulation time	50 s
Traffic source	CBR
Packet size	512
Mobility model	Random way point
Speed	10 m/s
Rate	100,150,200 and 250 Kb

- *Overhead* It is the number of control packets exchanged during the entire transmission of data packets.

The simulation results are presented in the next section. For comparison, the Hop-by-Hop [36] technique is implemented in NS-2 using the same simulation settings described above. Then, we compare the proposed adaptive reliable congestion control routing protocol (ARCCRP) with Hop-by-Hop technique and the standard TORA protocol.

6.3 Results

6.3.1 Based on nodes

In this experiment, we vary the network size by increasing the number of nodes as 50, 75, 100, 125 and 150. We keep the number of flows as 8 and data rate as 250 kb.

From Fig. 1 we can see that when we increase the number of nodes, the delay is increased linearly. It can be observed that TORA has the highest delay followed by Hop-by-Hop. ARCCRP has significantly less delay than the Hop-by-Hop and TORA.

When we increase the number of nodes, the number of packets dropped increases and hence the packet delivery ratio decreases slightly. From Fig. 2, we can see that ARCCRP has high delivery ratio followed by Hop-by-Hop, while TORA has the least delivery ratio. Figure 3 shows that the packet drop is less in ARCCRP followed by Hop-by-Hop, whereas TORA has the highest packet drop.

When the number of nodes is increased, the control packets exchanged will increase resulting in the increased overhead. But, from Fig. 4, we can see that, ARCCRP has slightly less overhead than the existing Hop-by-Hop protocol.

6.3.2 Based on flows

In this experiment, the number of CBR connections or traffic flows is varied from 2 to 8 for 150 nodes network to increase the amount traffic in the network. The traffic rate is kept as 250 kb.

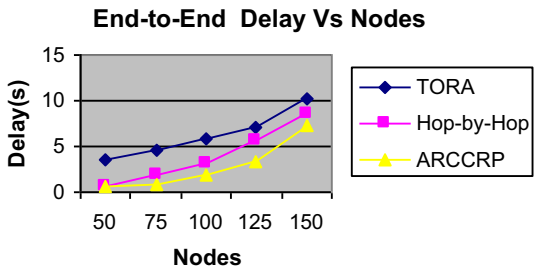

Fig. 1 Delay versus nodes

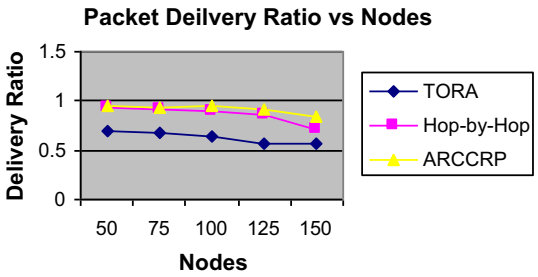

Fig. 2 Delivery ratio versus nodes

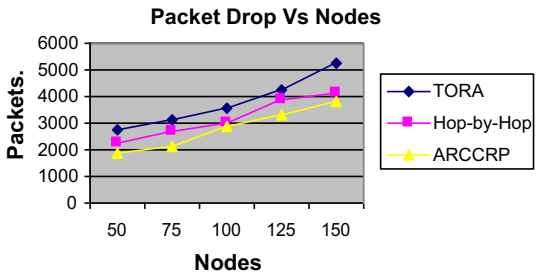

Fig. 3 Packet drop versus nodes

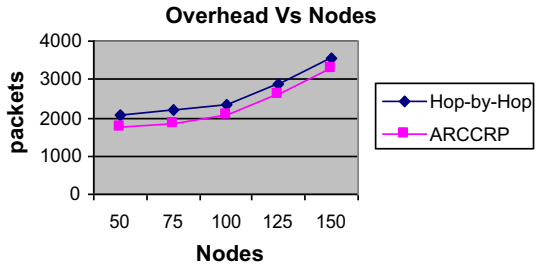

Fig. 4 Overhead versus nodes

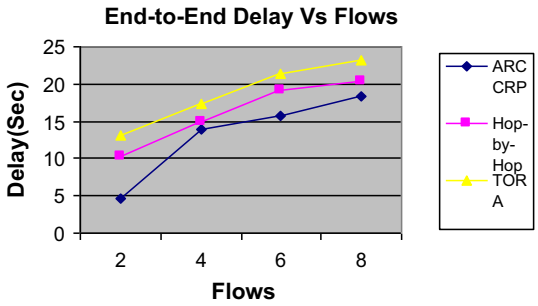

Fig. 5 Delay versus flows

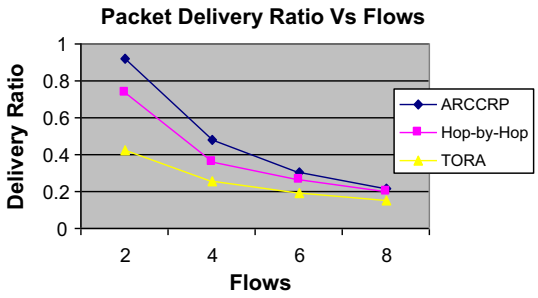

Fig. 6 Delivery ratio versus flows

When we increase the number of flows, it results in congestion. Hence, the delay and packet drop will increase, whereas the throughput and packet delivery ratio will decrease.

From Fig. 5, we can see that when we increase the number of flows, ARCCRP has less delay than Hop-by-Hop and TORA protocols.

From Fig. 6, we can see that when we increase the number of flows, ARCCRP has high delivery ratio when compared to Hop-by-Hop and TORA protocols. Figure 7 shows that ARCCRP has lower drop than Hop-by-Hop and TORA. In Fig. 8, we can see that ARCCRP has the highest throughput followed by Hop-by-Hop and TORA.

6.3.3 Based on rate

In this third experiment, the CBR traffic rate is varied from 100 to 250 kb to increase the amount traffic in the network, for 8 flows.

When we increase the transmission rate, it results in congestion. Hence, the delay and packet drop will increase, whereas the throughput and packet delivery ratio will decrease.

From Fig. 9 we can see that when we increase the rate, ARCCRP has less delay than Hop-by-Hop and TORA protocols.

From Fig. 10 we can see that when we increase the rate, ARCCRP has high delivery ratio when compared to Hop-by-Hop and TORA protocols. Figure 11 shows that ARCCRP has lower drop than Hop-by-Hop and TORA. In Fig. 12, we can see that ARCCRP has the highest throughput followed by Hop-by-Hop and TORA.

When the rate is increased, the control packets exchanged will increase resulting in the increased overhead. From Fig. 13, we can see that ARCCRP has slightly less overhead than the existing Hop-by-Hop protocol.

6.3.4 Based on pause time

In this experiment, we vary the pause time as 5, 10, 15, 20, and 25 with number of nodes as 50, rate as 250 kb, and number of flows as 8.

When we increase the pause time, mobile disconnections will be lower, thereby reducing route failures. So, increased throughput and packet delivery ratio with reduced packet drops can be achieved. The delay is also decreasing since retransmissions are reduced.

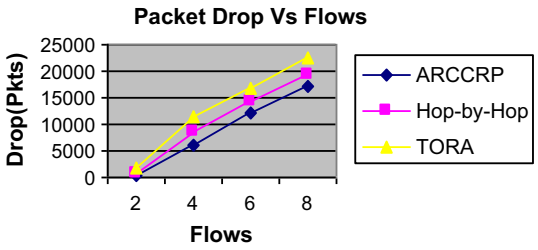

Fig. 7 Packet drop versus flows

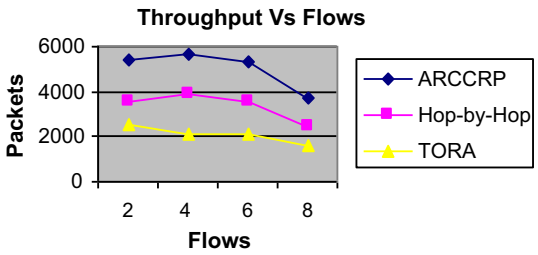

Fig. 8 Throughput versus flows

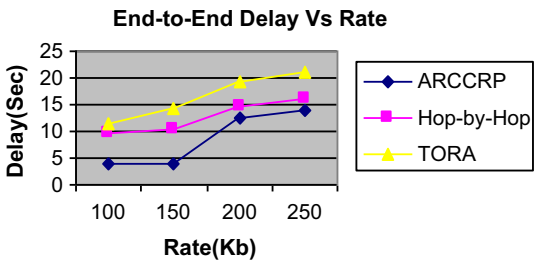

Fig. 9 Delay versus rate

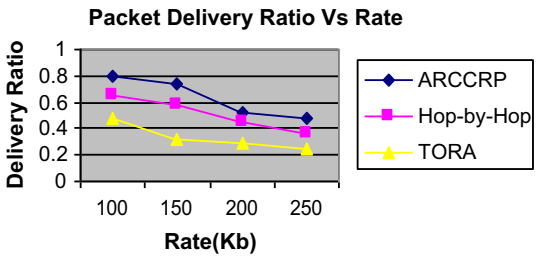

Fig. 10 Delivery ratio versus rate

From Fig. 14, we can see that than ARCCRP has less delay than Hop-by-Hop and TORA protocols. From Fig. 15, we can see that when we increase the pause time, ARCCRP has high delivery ratio when compared to Hop-by-Hop and TORA protocols. Figure 16 shows that ARCCRP has lower drop than Hop-by-Hop and TORA. In Fig. 17, we can see that ARCCRP has the highest throughput followed by Hop-by-Hop and TORA.

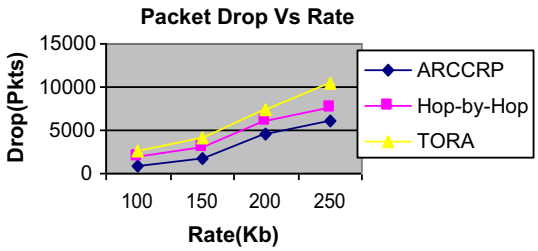

Fig. 11 Packet drop versus rate

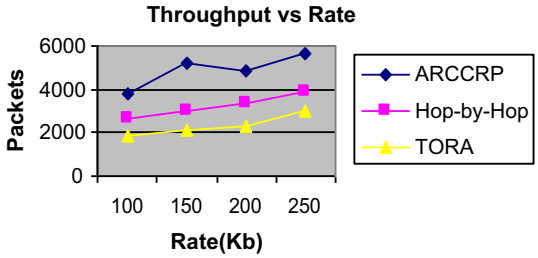

Fig. 12 Throughput versus rate

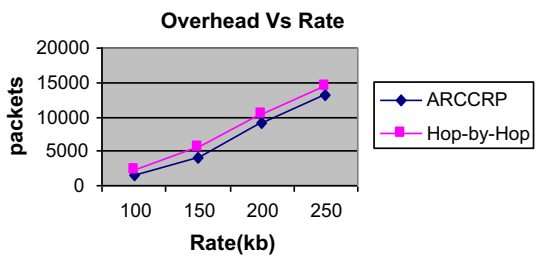

Fig. 13 Overhead versus rate

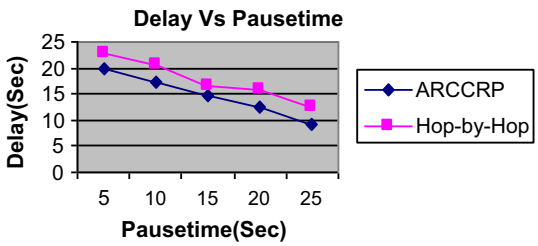

Fig. 14 Delay versus pause time

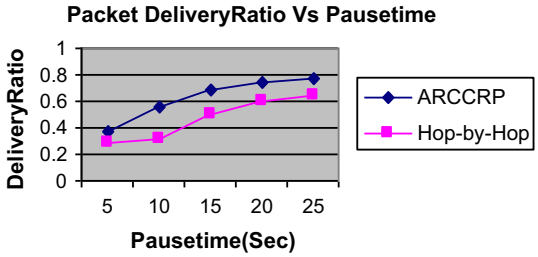

Fig. 15 Delivery ratio versus pause time

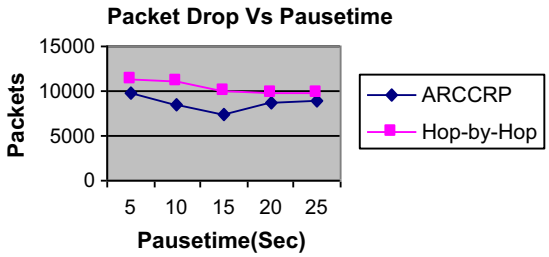

Fig. 16 Packet drop versus pause time

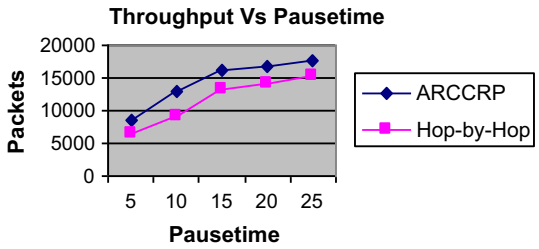

Fig. 17 Throughput versus pause time

7 Conclusion

In this paper, we have proposed an enhanced technique to resolve congestion using bypass route selection in MANETs. When a node detects congestion on a local outgoing link L, it calculates the multipath routes to destinations for which the path contains link L. Some portion of the traffic to node is then shifted to alternative paths. Congestion is detected on a local link if its utilization exceeds a local congestion threshold T_H. The objective is to minimize the utilization to a more acceptable level by shifting a portion of the traffic to the alternative paths and this part of traffic as bypass traffic. A node calculates a set of alternative paths and distributes the bypass traffic over these paths whenever it detects local link congestion or receives an Explicit Congestion Indication (ECI) bit from a neighbor. A node produces signals to its neighbors using the ECI bit. By simulation results, we have shown that the proposed technique is reliable and achieves more throughput with reduced packet drop. As a future work, we will compare the proposed technique with more existing works and present more detailed analysis.

References

1. Akyol, U., Andrews, M., Gupta, P., Honny, J., Saniee, I., & Stolyar, S. (2008). Joint scheduling and congestion control in mobile ad hoc networks. In *Proceedings of the IEEE INFOCOM*, pp. 619–627.
2. Attar, Alireza, et al. (2012). A survey of security challenges in cognitive radio networks: Solutions and future research directions. *Proceedings of the IEEE, 100*(12), 3172–3186.
3. Busch, C., et al. (2012). Approximating congestion + dilation in networks via "quality of routing" games. *IEEE Trans. Computers, 61*(9), 1270–1283.
4. Demestichas, P., et al. (2004). Service configuration and traffic distribution in composite radio environments. *IEEE Transactions on Systems, Man, and Cybernetics, Part C, 34*(1), 69–81.
32. de Morais Cordeiro, C., & Agrawal D. P. Ad hoc and sensor networks: Theory and applications.
5. Duarte, P. B. F., et al. (2012). On the partially overlapped channel assignment on wireless mesh network backbone: A game theoretic approach. *IEEE Journal on Selected Areas in Communications, 30*(1), 119–127.
6. Dvir, A., et al. (2011). Backpressure-based routing protocol for DTNs. *ACM SIGCOMM Computer Communication Review, 41*(4), 405–406.
7. Hou, X., & Tipper, D. (1990). Impact of failures on routing in mobile ad hoc networks using DSR. *Proceedings of IEEE Journal on Selected Areas in Communications, 8*(9), 1696–1708.
8. Jiang, T., et al. (2012). QoE-driven channel allocation schemes for multimedia transmission of priority-based secondary users over cognitive radio networks. *IEEE Journal on Selected Areas in Communications, 30*(7), 1215–1224.
9. Karunakaran, S., & Thangaraj, P. (2010). A cluster based congestion control protocol for mobile ad hoc networks. *International Journal of Information Technology and Knowledge Management, 2*(2), 471–474.
10. Khan, M. A., et al. (2012). Game dynamics and cost of learning in heterogeneous 4G networks. *IEEE Journal on Selected Areas in Communications, 30*(1), 198–213.
11. Li, P., et al. (2012). CodePipe: An opportunistic feeding and routing protocol for reliable multicast with pipelined network coding. INFOCOM 2012 pp. 100–108.
12. Li, P., et al. (2014). Reliable multicast with pipelined network coding using opportunistic feeding and routing. *IEEE Transactions on Parallel and Distributed Systems, 25*(12), 3264–3273.
13. Liu, J., et al. (2015). A novel energy-saving one-sided synchronous two-way ranging algorithm for vehicular positioning. *Mobile Networks and Applications, 20*(5), 661–672.
14. Liu, L., et al. (2015). Physarum optimization: A biology-inspired algorithm for the steiner tree problem in networks. *IEEE Transactions on Computers, 64*(3), 819–832.
15. Lochert, C., Scheuermann, B., & Mauve, M. (2007). A survey on congestion control for mobile ad-hoc networks. *Wiley Wireless Communications and Mobile Computing, 7*(5), 655–676.
16. Meng, T., et al. (2015). Spatial reusability-aware routing in multihop wireless networks. IEEE TMC.
17. Nishimura, K., & Takahashi, K. (2007). A multi-agent routing protocol with congestion control for MANET. European Conference on Modelling and Simulation, pp. 1–6.
18. Peng, J., & Sikdar, B. (2003). A multicast congestion control scheme for mobile ad-hoc networks. *GLOBECOM '03: Proceedings of the IEEE Global Telecommunications Conference, 5*, 2860–2864.
19. Rahebi, S., & Asadi, M. (2009). WBRR: A weight based reliable routing method in mobile ad hoc network. *Australian Journal of Basic and Applied Sciences, 3*(3), 1888–1897.
20. Rahman, K. C., & Hasan, S. F. (2010). Explicit rate-based congestion control for multimedia streaming over mobile ad hoc networks. *International Journal of Electrical and Computer Sciences IJECS-IJENS, 10*(04), 28–40.
21. Ramachandran, B., & Shanmugavel, S. (2009). Received signal strength-based cross-layer designs for mobile ad hoc networks. *IJCSNS International Journal of Computer Science and Network security, 9*(1), 192–200.
22. Santhi, G., Nachiappan, A. (2010). A survey of QoS routing protocols for mobile ad hoc networks. *International Journal of Computer Science and Information Technology (IJCSIT), 2*(4).

23. Shrivastava, L., Tomar, G. S., & Bhadauria, S. S. (2011). A survey on congestion adaptive routing protocols for mobile ad-hoc networks. *International Journal of Computer Theory and Engineering, 3*(2), 189–196.
24. Sohn, S., Mark, B. L., & Brassil, J. T. (2006). Congestion-triggered multipath routing based on shortest path information. In *Proceedings of the 15th International Conference on Computer Communications and Networks, (ICCCN'06)*.
25. Song, Y., et al. (2014). A biology-based algorithm to minimal exposure problem of wireless sensor networks. *IEEE Transactions on Network and Service Management, 11*(3), 417–430.
26. Su, X., Chan, S., & Chan, K. S. (2007). RLAR: Robust link availability routing protocol for mobile ad hoc networks. In *IEEE International Conference on Communications, ICC '07* (pp. 4759–4766).
27. Vadivel, R., & Bhaskaran, V. M. (2010). Adaptive reliable routing protocol using combined link stability estimation for mobile ad hoc networks. In *International Conference on Modeling, Optimization, and Computing (ICMOS 20110)* West Bengal, India, AIP Conference on Proceedings 1298, pp. 625–632.
28. Vasilakos, A., et al. (1998). Evolutionary-fuzzy prediction for strategic QoS routing in broadband networks. *The 1998 IEEE International Conference on Fuzzy Systems Proceedings, 2*(1), 1488–1493.
29. Wu, C., Zhang, F., & Yang, H. (2010). A novel QoS multipath path routing in MANET. *International Journal of Digital Content Technology and its Applications, 4*(3), 132–136.
30. Xue, Y., & Nahrstedt, K. (2003). Fault tolerant routing in mobile ad hoc networks. In *Proceedings of the IEEE WCNC* (pp. 1174–1179). IEEE: New Orleans.
31. Network Simulator, http://www.isi.edu/nsnam/ns.
33. Yang, M., et al. (2015). Software-defined and virtualized future mobile and wireless networks: A survey. *ACM/Springer Mobile Networks and Applications, 20*(1), 4–18.
34. Yao, Y., et al. (2013). EDAL: An energy-efficient, delay-aware, and lifetime-balancing data collection protocol for wireless sensor networks. *MASS*, pp. 182–190.
35. Yen, Y. S., et al. (2011). Flooding-limited and multi-constrained QoS multicast routing based on the genetic algorithm for MANETs. *Mathematical and Computer Modelling, 53*(11–12), 2238–2250.
36. Yi, Y., & Shakkottai, S. (2007). Hop-by-hop congestion control over a wireless multi-hop network. *IEEE/ACM Transactions on Networking, 15*(1), 133–144.
37. Youssef, M., et al. (2014). Routing Metrics of Cognitive Radio Networks: A Survey. *IEEE Communications Surveys and Tutorials, 16*(1), 92–109.
38. Zeng, Y., et al. (2013). Directional routing and scheduling for green vehicular delay tolerant networks. *Wireless Networks, 19*(2), 161–173.
39. Zhang, X. M., et al. (2015). Interference-based topology control algorithm for delay-constrained mobile ad hoc networks. *IEEE Transactions on Mobile Computing, 14*(4), 742–754.
40. Zhou, L., et al. (2010). Context-aware middleware for multimedia services in heterogeneous networks. *IEEE Intelligent Systems, 25*(2), 40–47.

R. Vadivel is a Assistant Professor in the Department of Information Technology, School of Computer Science and Engineering, Bharathiar University, Coimbatore, Tamil Nadu, India. He obtained his Diploma in Electronics and Communication Engineering from State Board of Technical Education in the year 1999, B.E., Degree in Computer Science and Engineering from Periyar University in the year 2002, M.E., degree in Computer Science and Engineering from Annamalai University in the year 2007 and Ph.D., degree in CSE from Manonmaniam Sundaranar University in the year 2013. He has published 20 papers in journals and 15 papers in Conferences both at National and International level. He is a life member of ISTE, ISCA, CSI and ACS, IAENG. Also he is an Associate Member of the Institution of Engineers (India) AMIE. His areas of interest include Computer Networks, Network Security, Information Security, etc.

V. Murali Bhaskaran is a Principal of Paavai College of Engineering, NH-7, Pachal, Namakkal, Taminadu, India. He obtained his B.E. Degree in Computer Science and Engineering from Bharathidasan University in the year 1989, M.E. degree in Computer Science and Engineering from Bharathiar University in the year 2000 and secured University III rank and Ph.D in Computer Science and Engineering from Bharathiar University in the year 2008. He has published 4 papers in Journals and 12 papers in Conferences both at National and International level. He is a life member of ISTE, CSI and ACS. Also he is a fellow member in IE (FIE). His areas of interest include Computer Architecture, Computer Networks, Network Security, Information Security, etc.