



Cairo University  
**Egyptian Informatics Journal**

[www.elsevier.com/locate/eij](http://www.elsevier.com/locate/eij)  
[www.sciencedirect.com](http://www.sciencedirect.com)



ORIGINAL ARTICLE

# Early congestion detection and adaptive routing in MANET

T. Senthil Kumaran \*, V. Sankaranarayanan

*BSA Crescent Engineering College, Vandalur, Chennai, Tamilnadu, India*

Received 11 May 2011; revised 11 August 2011; accepted 7 September 2011  
Available online 6 October 2011

## KEYWORDS

Congestion;  
EDAPR;  
NHN;  
EDCSAODV;  
EDAODV

**Abstract** Ad hoc mobile networks are composed of mobile nodes communicating through wireless medium, without any fixed backbone infrastructure. In these networks, congestion occurs in any intermediate node when data packets travel from source to destination and they incur high packet loss and long delay, which cause the performance degradations of a network. This paper proposes an early congestion detection and adaptive routing in MANET called as EDAPR. Initially EDAPR constructs a NHN (non-congested neighbors) neighbors list and finds a route to a destination through an NHN node. All the primary path nodes periodically calculate its queue status at node level. While using early congestion detection technique, node detects congestion that is likely to happen and sends warning message to NHN nodes. The ancestor NHN node is aware of this situation and finds an alternate path to a destination immediately by applying adaptive path mechanism. Thus, EDAPR improves performance in terms of reducing delay, routing overhead and increases packet delivery ratio without incurring any significant additional cost. The performance of EDAPR was compared with EDAODV and EDCSAODV using the Ns-2 simulator. The result reveals significant improvement over EDAODV and EDCSAODV routing schemes.

© 2011 Faculty of Computers and Information, Cairo University.  
Production and hosting by Elsevier B.V. All rights reserved.

\* Corresponding author.

E-mail addresses: [senthilkumaran@bsauniv.ac.in](mailto:senthilkumaran@bsauniv.ac.in) (T. Senthil Kumaran), [sankarammu@bsauniv.ac.in](mailto:sankarammu@bsauniv.ac.in) (V. Sankaranarayanan).

1110-8665 © 2011 Faculty of Computers and Information, Cairo University. Production and hosting by Elsevier B.V. All rights reserved.

Peer review under responsibility of Faculty of Computers and Information, Cairo University.  
doi:10.1016/j.eij.2011.09.001



Production and hosting by Elsevier

## 1. Introduction

Wireless is a new technology that allows users to access information and services regardless of the geographic position. People can utilize and surf the Internet with computers (e.g., laptop, palmtop, smart phone and PDA) whenever and wherever possible. In general, wireless network can be classified into two types: infrastructure network and ad hoc network. Mobile ad hoc network (MANET) is an autonomous group of mobile users who communicate through relatively bandwidth constrained wireless links. Since the hosts are mobile, the network topology may change rapidly and unpredictably over time [1]. One of the fundamental tasks that an ad hoc network should perform is congestion control. The main objective of congestion control is to limit the delay and buffer overflow caused

by network congestion and provide better performance of the network [2]. In wire line networks, congestion control is implemented at the transport layer and is often designed separately from functions of other layers. Since wired links have fixed capacities and are independent, this methodology is well justified and has been extensively studied [2]. However, these results do not apply directly to ad hoc networks because the ad hoc networks result in large amount of packet loss, high delay, unfair scenarios and low throughputs. In ad hoc networks each mobile node has limited transmission capacity and buffer and they mostly intercommunicate by multi-hop relay [2].

In ad hoc networks, the routing protocols for MANETs fall into three categories based on the routing information update mechanism [3,4]: proactive, reactive (or on-demand) and hybrid. Proactive routing protocols, such as OLSR [1] and DSDV [4] [5], attempt to maintain consistent and up-to-date routing information from each node to every other node in the network. In the on-demand routing protocols, such as AODV [2,6] and DSR [8,10,11] routes are discovered only when they are needed. The hybrid routing protocols [2,10] combines the features of both proactive and on-demand protocols. In the case of hybrid routing protocols, each node maintains routing information about its zone using the proactive approach. It uses on-demand routing approach outside the zone. The periodic routing information updates due broken links that can lead to a large routing control overhead, when the network is scaled up such that the mobility of nodes is high.

There is another dimension for categorizing routing protocols: congestion-adaptive routing versus congestion-un-adaptive routing. We note that the existing routing protocols are congestion-un-adaptive [12–14]. When establishing a new route, it remains the same until mobility or failure results in disconnection. During packet transfer between source and destination congestion may happen, this is not handled by existing routing protocol. It may also lead to the following problems: (i) long delay, (ii) many packet losses and (iii) low throughput. The above problems become more visible in large-scale transmission of traffic intensive data such as multimedia data probable and the negative impact of packet loss on the service quality is of more significance [2]. Unlike well-established networks such as the Internet, in a dynamic network like a MANET, it is expensive, in terms of time, overhead, to recover from congestion [2].

Our motivation is to clear that congestion which is a dominant cause for packet loss in MANETs. Typically, reducing packet loss involves congestion control running on top of a mobility and failure adaptive routing protocol at the network layer [2,9]. A new perspective of this problem might be to realize congestion control in the MAC or network layer. After all, it might make sense to tackle the problem from where it emerges. An exceedingly high network load is a problem closely associated with medium access and packet forwarding [2,9].

Congestion is a dominant reason for packet drops in ad hoc networks [15]. Lu et al. [15] found that AODV is ineffective under stressful network traffic situations. They proposed a modified version of AODV (called CADV) which favors nodes with short queuing delays in adding into the route to the destination. While this modification may improve the route quality, the issues of long delay and high overhead when a new route needs to be discovered remain unsolved. Furthermore, CADV is not congestion adaptive. It offers no remedy when an existing route becomes heavily congested. A dynamic

load-aware routing protocol (DLAR) was proposed in [16]. DLAR is similar to CADV, the difference being that a node with low routing load is favored to be included in the routing path during the route discovery phase.

In our previous work, EDAODV (early detection congestion and control routing) [17] techniques have been proposed to detect the congestion well in advance and find a non-congested alternate path bi-directionally. A technique for self curing the congestion was proposed in [18] and is called the EDCSCAODV (early detection congestion and self cure routing). To utilize the concepts of EDAODV and EDCSCAODV, we propose the EDAPR routing protocol for mobile ad hoc networks. EDAPR detects congestion by using early congestion detection techniques (Section 2) and it can easily chose a non-congested alternate node from their two hop lists and establish as a route to destination immediately. The new protocol can reduce broadcast packets and find a non-congested path. Our NS2 simulation results confirms (Section 5) that our proposed protocol has a higher packet delivery rate, lower control packets and reduced end to end delay. Section 3 discusses the related works. Section 4 presents a method for an Adaptive Routing to find an alternate path to control the congestion.

## 2. Early congestion detection technique

Congestion in a network may occur at any interval, when the number of packets coming to a node exceeds its buffer capacity, the node becomes congested and starts losing packets. We can use a variety of metrics at a node to monitor congestion status. For instance, we can be based on the percentage of all packets discarded for lack of buffer space and the average queue length. We use an early congestion detection technique at a node to detect the congestion well in advance. An early congestion detection technique is a queue management algorithm with an optimization of random early detection (RED) model that makes use of direct measurement congestion status well in advance in a network [10].

The expression (1) and (2) are used to set the Minimum threshold and Maximum threshold values for the queue length.

$$\text{Minth} = 25\% \text{buffer\_size} \quad (1)$$

$$\text{Maxth} = 3 * \text{Minth} \quad (2)$$

If the queue length is less than the Minth, then the node can be classified to be in Zone I (safe zone), greater than Minth but less than Maxth is classified as Zone-II (likely to be congested zone) and if it is greater than Maxth is classified as Zone-III (congested zone) .

To detect the congestion well in advance, compute the average queue size as

$$\text{Avgque} = (1 - w_q) * \text{Avgque} + \text{Inst\_Que} * w_q \quad (3)$$

where  $w_q$ , the queue weight is a constant parameter ( $w_q = 0.002$  from RED queue experimental result [7]) and  $\text{Inst\_Que}$  is an instantaneous queue size.

In our early congestion detection model, we introduce Queue\_status over average queue size given by Eq. (4), which reflects the heaviness of the incoming traffic. Based on the Queue\_status, the mobile node can get useful information about the incoming traffic. If the Queue\_status value is large, the incoming traffic becomes bursty traffic. The continuous

growth of the Queue\_status indicates that the incoming heavy traffic is beyond the mobile node's buffer capacity and buffer overflow is imminent.

$$\text{Queue\_status} = \text{Inst\_que} - \text{Avgque} \quad (4)$$

If  $\text{Queue\_status} < \text{minimum threshold}$ , the incoming traffic is low and queue is in safe zone. If  $\text{Queue\_status} > \text{minimum threshold}$  and  $\text{Inst\_Que} < \text{maximum threshold}$ , the incoming is normal and queue is in likely to be in congested zone. If  $\text{Inst\_Que} > \text{maximum threshold}$ , the incoming traffic is heavy and queue is in congested zone as shown in Fig. 1.

### 3. Related work

#### 3.1. Bidirectional path discovery

The source discovers the route to the destination; it broadcasts an RREQ packet toward the destination, the destination responds to the first arrived RREQ and sends back an RREP packet. The RREP will travel back in the path the RREQ previously travelled and adds a new entry in its routing table. This

path becomes the primary route between the source and the destination. Each node has two routing tables, primary table (denoted as  $P_{RT}$ ) and alternate path routing table (denoted as  $A_{RT}$ ).  $P_{RT}$  is used to direct packets on the primary route, while  $A_{RT}$  directs packets on alternate path routes. If  $A_{RT} = 0$  for a node that does not appear on an alternate path route of any connection [17,19]. A primary path of a node predicts its congestion status and periodically broadcasts a congestion status packet (CSP) to its neighbors with  $\text{TTL} = 1$ . The CSP packet contains the node's congestion status and a set of parameters (source  $S$ , destination  $D$ , previous ZoneI node  $P\_ZoneI$ , Previous ZoneI hop count  $P\_Zhop$ , Next ZoneI node  $N\_ZoneI$ , Next ZoneI hop count  $N\_Zhop$ ), each for a destination appearing in the routing table [17,19].

A simplified example is illustrated in Fig. 2. A route  $S \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow D$  is initially found for the source  $S$  to the destination  $D$ . This route is called the primary route from  $S$  to  $D$ . Every packet follows the primary route. Sometime later, node 3 detects that congestion is likely to occur and sends a warning to its neighborhood, its predecessor node and the successor node. They are aware of this situation as shown in

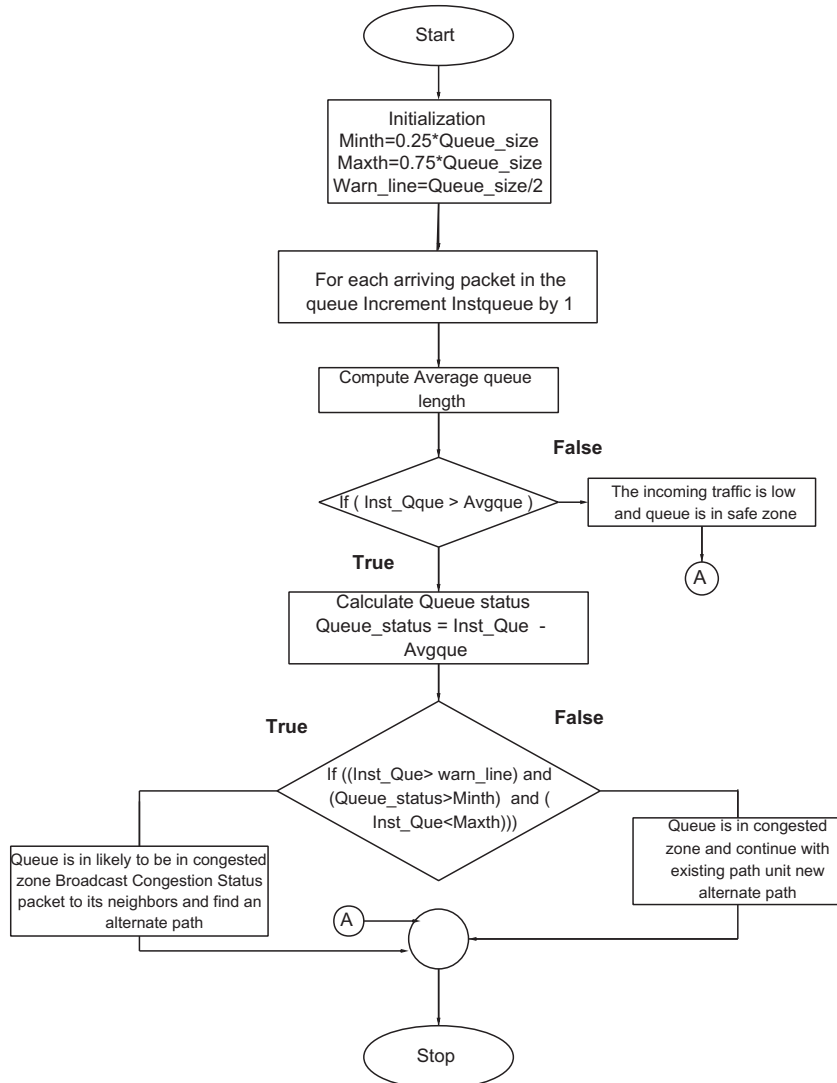
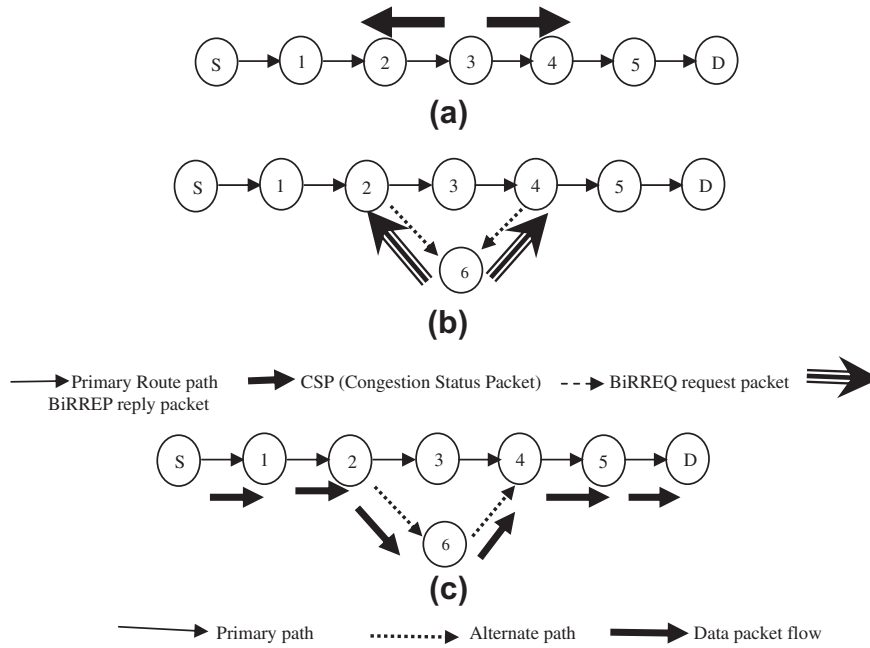


Figure 1 Early congestion detection.



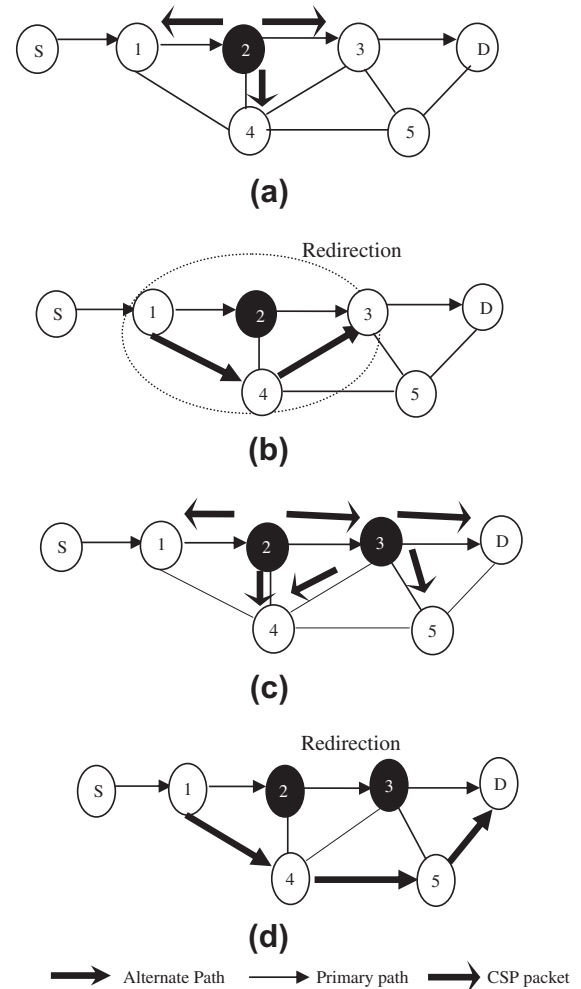
**Figure 2** Finding alternate path for reduce congestion.

Fig. 2a. In response, node 2 and 4 find a bidirectional alternate route bypassing 3 as shown in Fig. 2b. Finally 2 find an alternate path destined for  $D$  as shown in Fig. 2c. Traffic coming to 2 will bypass through alternate route  $2 \rightarrow 6 \rightarrow 4$  respectively. It is possible that no alternate path is found, in this case, it continues using the primary route  $S \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow D$  [17,19].

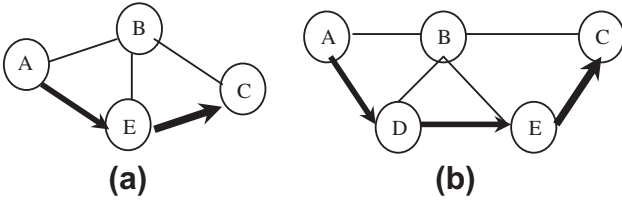
### 3.2. Self cure routing

The source discovers the route to the destination; it broadcasts an RREQ packet toward the destination, the destination responds to the first arrived RREQ and sends back an RREP packet. The RREP will travel back in the path the RREQ previously travelled and adds a new entry in its routing table. This path becomes the primary route between the source and the destination. Each node has two routing tables, primary table (denoted as  $P_{RT}$ ) and neighbors table (denoted as  $N_{RT}$ ).  $P_{RT}$  is used to direct packets on the primary route [18,20]. A primary path of a node predicts its congestion status and periodically broadcasts a congestion status packet (CSP) to its neighbors with  $TTL = 1$ . The CSP packet  $P$  contains the node's congestion status and a set of parameters (Source  $S$ , Destination  $D$ , Hop Count  $hop$ , Sequence Number  $Seq$ , Congestion Status  $Cong$  and Neighbors information  $N\_list$ ), each for a destination appearing in the routing table [18,20].

A simplified example is illustrated in Fig. 3. A route  $S \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow D$  is initially found for the source  $S$  to the destination  $D$ . This route is called the primary route from  $S$  to  $D$ . Every packet follows the primary route. Thus, node 2 and node 3 will be continuously used in forwarding the traffic, leaving the other nodes free from the traffic load [18]. As a result, node 2 detects that congestion is likely to occur and sends congestion status packet (CSP) to its neighbors. They are aware of this situation as shown in Fig. 3a. In response, node 4 is a nearby node which is a common neighbor of



**Figure 3** Example of successive local route redirection operations.



**Figure 4** Basic cases of alternative sub-paths.

primary path node 1, node 2 and node 3. Here node 4 applies case 1 self cure routing scheme as shown in Fig. 4a, to divert the traffic to other nodes based on route redirection as shown in Fig. 3b. With successive local redirection operations, the route will gradually converge to an alternative node 4 make disjoint path as shown in Fig. 3b. Finally the self cure routing successfully cure congestion and finds a non-congested alternate path  $S \rightarrow 1 \rightarrow 4 \rightarrow 3 \rightarrow D$  as shown in Fig. 3b [18].

Considering the example shown in Fig. 3c, nodes 2 and 3 both fall likely to be congested zone and send congestion status packet (CSP) to its neighbors. They are aware of this situation as shown in Fig. 3c. In response, Node 4 is a nearby node which is a common neighbor of primary path node 1, node 2 and node 3. Here node 4 applies case 2 self cure routing scheme as shown in Fig. 4b. After first redirection of case 2, node 2 is bypassed. Similarly node 5 is a common neighbor of primary path node 3 and node D. After second redirection of case 2 as shown in Fig. 4b, node 3 is bypassed as shown in Fig. 3d. Finally the self cure routing successfully cure congestion and finds a non-congested alternate path  $S \rightarrow 1 \rightarrow 4 \rightarrow 5 \rightarrow D$  as shown in Fig. 3d [18].

#### 4. Adaptive routing

Early congestion detection and adaptive routing in MANET called EDAPR is a uni-cast routing protocol for MANET. It reduces network congestion, by minimizing the flooding of traffic, find a non-congested path between source and destination. We present this as complete design with an in-depth evaluation using EDAPR routing protocol. When the source host wants to transmit a data packet to a destination, EDAPR protocol first constructs a NHN set (non-congested neighbors) which connects both one hop and two hop neighbors and initiates the route discovery procedure by using NHN set to find a non-congested path to a destination [1,16]. After the route discovery, data packet is transmitted to the destination. Hence, the EDAPR can reduce the overhead and automatically finds

the non-congested path and decreases the flooding packets. EDAPR consists of the following components

1. NHN set construction.
2. Route discovery.
3. Adaptive routing.

##### 4.1. NHN set construction

Each mobile host selects its NHN set from among its non-congested one-hop neighbors. This set is selected in such a way that it covers all two-hop nodes. The NHN set of source host  $S$ , denoted by  $NHN(S)$ , is then an arbitrary subset of the non-congested one-hop neighborhood of  $S$  which satisfies the following condition: every node in the strict two-hop neighborhood of  $S$  must have a link towards  $NHN(S)$  and it should not fall in congested zone. The NHN setup is an initialization procedure, each mobile host periodically calculates its congestion status by using early congestion detection algorithm. Every mobile host broadcasts its congestion status by using a CSP (congestion status packet) packet to its one hop neighbors on the network. Now, each mobile host learns its one-hop non-congested neighbor nodes and records the information into its non-congested one-hop list. After that, each mobile host exchanges its one-hop non-congested neighbor information thereby learning its two-hop non-congested neighbor nodes. At this point, each mobile host constructs its NHN-set by selecting a subset of its one-hop non-congested neighbor nodes so that the mobile hosts in the subset can forward its broadcast traffic to the two-hop neighbor nodes to minimize the flooding traffic. Each mobile host updates all the information in its routing table.

The format of each entry in the routing table is  $\langle \text{src\_addr}, \text{dst\_addr}, \text{hop\_cnt}, \text{NHN\_Node}, \text{NHNSET}, \text{congest\_status} \rangle$ , where  $\text{src\_addr}$  is the source address,  $\text{dst\_addr}$  is the destination address,  $\text{hop\_cnt}$  is the hop count,  $\text{NHN}$  is the non-congested  $\text{NHN\_node}$  address,  $\text{NHNSET}$  is the non-congested neighbors List and  $\text{congest\_status}$  is the neighbor's congestion status. Table 1 shows non-congested neighbors' information and Fig. 5 shows a NHN set selection.

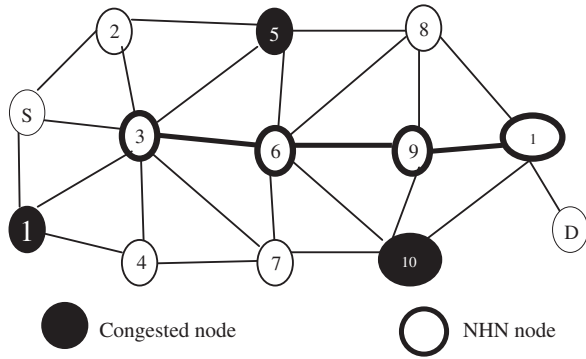
##### 4.2. Route discovery

When the source host wants to transmit a datagram to a destination, the source generates the RREQ packet for communication using NHN-set nodes towards to the destina-

**Table 1** One hop and two hop neighbors information.

Node	One hop non-congested	One hop congested	Two hops non-congested	Two hops congested
$S$	2,3	1	4,6,7	5
2	$S,3$	5	4,6,7	1
3	$S,2,4,6,7$	1,5	8,9	10
4	3,7	1	$S,2,6$	5, 10
6	3,7,8,9	5,10	$S,2,4,11$	1
7	3,4,6	10	$S,2,8,9$	1, 5
8	6,9,11	5	3,7, $D$	10
9	6,8,11	10	3,7, $D$	5
11	8,9, $D$	10	6	5, 10
$D$	11	—	8,9	10



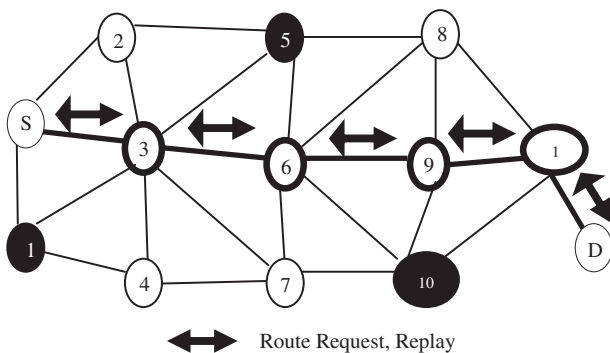


**Figure 5** NHN set (non-congested two hops neighbors) selection.

tion. The source host first checks its two-hop list. If the destination host is in its two-hop list, then the datagram is transmitted by following the routing table's path. If the destination host is not in its two-hop list, the source host broadcasts the RREQ to the NHN-set on the network. When the NHN-set receives this RREQ packet, the NHN-set also checks their two-hop list. If the destination host is in their two-hop list, then the NHN-set forwards directly the RREQ to the destination host. The destination responds to the first arrived RREQ and sends back an RREP packet. The RREP will travel back in the path the RREQ previously travelled and adds a new entry in its routing table. This path becomes the primary route between the source and the destination. In case if the destination host is not in their two-hop list, then it modifies the sequence-number and hop-count, and rebroadcasts this RREQ to the network.

The process is repeated until it finds the destination host. Finally the source found a non-congested path to destination. After the route discovery, data packet is transmitted to the destination. Hence, the EDAPR can reduce the overhead and automatically finds the non-congested path. Hence, this new protocol can decrease the flooding packets. Fig. 6 shows the route discovery after the NHN-set selection. The source host S has a non-congested one-hop list consisting of mobile hosts {2, 3, 4}; a non-congested two-hop list consisting of mobile hosts {4, 6, 7}; the source has chosen node 3 as a NHN and add to NHN list. The first mobile host S checks its two-hop list to see if it includes the destination host D.

If destination host D is not in this list, the sources host S forwards the RREQ packet to the next NHN node 3. Then,



**Figure 6** Route discovery process through NHN set.

node 3 would check the two-hop list. If the destination is not inside, NHN node 3 forwards RREQ to next NHN node 6, NHN host 6 would check the two-hop list. If the destination is not inside, then NHN node 6 forwards RREQ to next NHN node 9, now node 9 finds the destination node D is in the two-hop list, then node 9 forwards this packet through NHN node 11 to the destination node D. Destination node D receives the RREQ packet and then returns the RREP packet to the source. The RREP follows the reverse path of RREQ to the source host. A route  $S \rightarrow 3 \rightarrow 6 \rightarrow 9 \rightarrow 11 \rightarrow D$  is found for the source S to the destination D. this path is a non-congested path between source and destination. After the route discovery, datagram is transmitted to the destination. This route is called the primary route from S to D. Every packet follows the primary route. The route discovery algorithm I is as shown given in Fig. 7. This algorithm initially takes 2% extra delay for constructing NHN set than other approach, but while data transmission, the end to end delay is reduced around (10–15% than other approaches. So that the impact of NHN set construction delay is nil because it is one at a time when regular packet forwarding is in progress.

#### 4.3. Alternate path routing

A primary path of a NHN node predicts its congestion status and periodically broadcasts a congestion status packet (CSP) to its neighbor NHN nodes. The CSP packet contains the node's congestion status and a set of parameters (source Src, Destination Dst), each for a destination appearing in the routing table. When the Neighbors of NHN node receives a CSP packet from its primary path congested NHN node regarding destination D, they are aware of the congestion status of NHN node and updated accordingly. After receiving CSP packet from a neighbor NHN node, the ancestor NHN node will be identified new NHN node from its neighbor list and construct a new NHN set from current node to destination. After constructing new NHN set, it will be exchanged new NHN set to its neighbors.

When the NHN node receives new NHN set, it first compares the (Src, Dst) pair information in its routing table. If the entries do not match add new entry of NHN set in its routing table otherwise it updates itself in its routing table. Finally the processor NHN node calls route discovery process to find a route to destination. The alternate path finding algorithm does not incur any significant overhead because for every NHN set information, only one extra broadcast message is necessary to inform one of the neighboring nodes to update its routing table and also route finding process does not incur any cost to find an alternate path. The alternate path finding is shown in Fig. 8.

In Fig. 9a shows that NHN node 9 detects that congestion is likely to occur and sends a warning to its neighbors of NHN node 6 and 11, they are aware of this situation and update their non-congested neighbors list in their routing table. In response, processor NHN node 6 chooses a new NHN node 8 from its non-congested neighbors list because node 8 is a common node for node 6 and node 8 and it finds route to destination using NHN node 8 as shown in Fig. 9b. Traffic coming to 6 will be routed through new route  $S \rightarrow 3 \rightarrow 6 \rightarrow 8 \rightarrow 11 \rightarrow D$  respectively. It is possible that if no NHN nodes are found, in this case, it continues using the primary route  $S \rightarrow 3 \rightarrow 6 \rightarrow 9 \rightarrow 11 \rightarrow D$ . The new path is non-congested path but not necessarily the shortest path.

**Algorithm I (Route Discovery process).****Input:** when the NHN sets receive RREQ packet from Source**Output:** Efficient route between source and Destination

```

Begin
1. Construct NHN set for all mobile hosts
   /* NHN – non congested NHN –set nodes of the network */
2. For each node pair (S, D)i, where i=1 to (N-1) /*D=2, 3, 4 . . . N*/
3.   Hops=0; Routei=Null;
   /* Src: source node; Dst: destination node; Route: output path set generated for node pair (S, D), initially set to be Null */
4.   IF(Dst is in two hop list of Si) Then
5.     Path generated for pair (Si, Di)
6.     Set Routei = TRUE
7.     Hops =2;
8.   Else
9.     NHN=Si;
10.    Call Procedure PATH (input:NHN, Di; output: Routei)
11.  End If
12. End Loop
End

Procedure PATH (input: NHN, Di; output: Routei)
Begin
1. IF(Dst is inNHN) Then
2.   Path generated for pair (Si, Di)
3.   Set Routei = TRUE
4.   Increment Hops by 1
5.   Return
6. Else
7.   IF NHN-SET is not in Routei and (NHN-SET's two-hop list does not contain Di) Then
8.     /* Hops: number of hops */
9.     Begin
10.    Increment Hops by 1
11.    Add NHN-SET to Routei
12.    For each neighboring node Neib of node NHN-SET Do
13.      /* Neib: the neighbor NHN-set node of NHN-set */
14.      PATH (Neib,Di, Routei)
15.    End Loop
16.  End If
End

```

**Figure 7** Route discovery algorithm.**5. Performance study***5.1. Performance metrics*

EDAPR, EDCSCAODV and EDAODV are implemented using the Network Simulator (Ns2.34) [21]. A comparison of EDAPR's performance to EDAODV and EDCSCAODV routing protocols in MANET is made. Thus the observation is presented below as:

We considered the following important metrics for the evaluation:

1. *Packet Delivery Ratio (PDR)*: The ratio between the number of packets received by the destination and the number of packets sent by the source.
2. *End-to-End Delay*: The delay a packet suffers from leaving the sender to arriving at the receiver.
3. *Routing overhead*: The total number of Control packets transmitted during the simulation time. For packets sent over multiple hops, each transmission over one hop is counted as one transmission.

*5.2. Simulation configuration*

The network consisted of 100 nodes in a 1400, 1400 m terrain size. The radio range is 250 m with 2 Mbps bandwidth. The

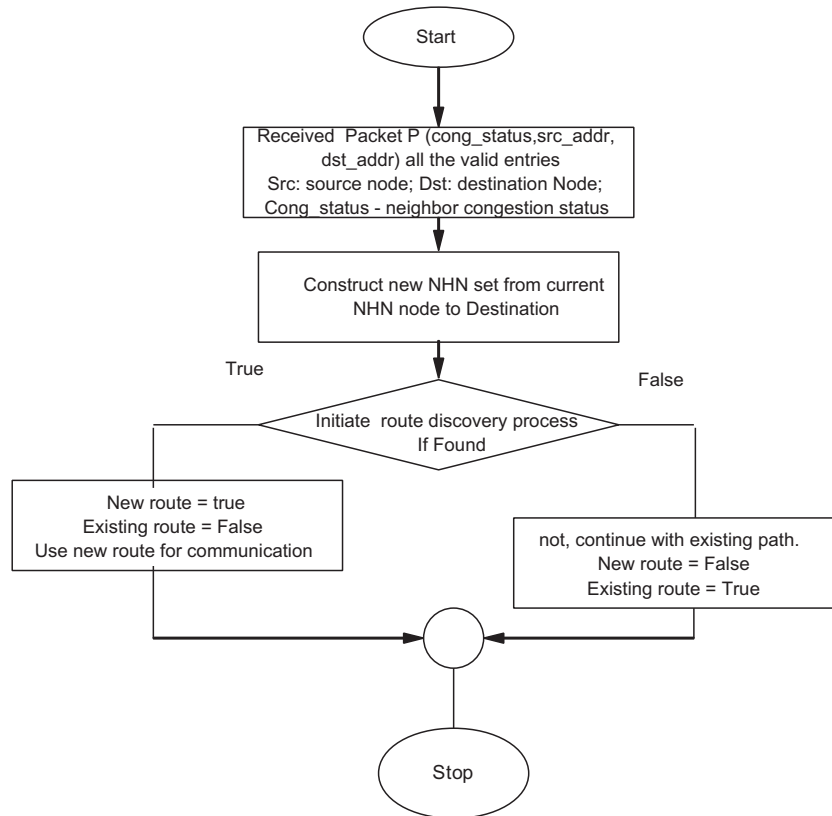
MAC layer was based on IEEE 802.11 DCF (distributed coordination function). The channel propagation model we used 2-ray ground reflection models. An interface queue at the MAC layer could hold 50 packets before they were sent out to the physical link. Link breakage was detected from MAC layer feedbacks. A routing buffer at the network layer could store up to 64 data packets. This buffer keeps data packets waiting for a route, such as packets for which route discovery had started but no reply arrived yet. The routing protocols we used EDAPR, EDAODV, and EDCSCAODV. The data flow used CBR which varies from 4 packets to 16 packets and flows varies from 10 to 50 flows. The Maximum speed of the node is 10 m/s and simulation time is 900 s.

*5.3. Varying number of connections*

In this simulation, the number of connection (source and destination) is varied from 10 to 50, CBR sending rate 4 packets/s, maximum node speed 10 m/s and pause time 30 s.

Fig. 10a, b and c show the end-to-end delay, packet delivery ratio and Routing overhead for EDAPR, EDCSCAODV and EDAODV respectively.

The results in Fig. 10a shows that the delays incurred by the Three protocols (EDAPR, EDCSCAODV and EDAODV) are similar when it is set to 10 flows. This is because at low offered load, the network congestion level falls in safe zone. As an outcome of the end-to end delay is minimum by incurring data



**Figure 8** Alternate path finding process.

packets on route to their destinations. In the case of high offered load (e.g. 20 flows) and the network congestion level falls in likely to congestion zone, the end to end delay incurred by the three protocols increases almost linearly with an increased offered load. However, at 30 flows, the EDAPR routing protocols demonstrate a 4% reduction of the delay over the EDCSCAODV and 10% reduction over EDAODV. The reasons are as follows: when the network falls likely to congested zone (Zone II level), immediately all three protocols try to find a non-congested alternate path. While finding an alternate path, in EDAODV, the non-congested primary path predecessor and successor node tried to find an alternate path explicitly. In EDCSCAODV the intermediate nodes are implicitly trying to find an alternate path. Whereas EDAPR is concerned, it has a two hop NHN set, so that it can easily chose a non-congested alternate node from their two hop lists and establish as a route to destination immediately. This is due to the fact that the numbers of forwarding NHN nodes are minimal; it leads decreased network congestion. When EDAPR is compared with EDCSCAODV and EDAODV, at high offered loads (between 40 and 50 flows), the delay is reduced by around 12% over EDCSCAODV and reduced by 20% over EDAODV.

Fig. 10b shows the achieved packet delivery ratio of the three protocols which is similar when the offered load is below 20 flows. This is because when the number of flows is less, the number of nodes initiating route discovery operation is also less. When the no of flows increases from 30 to 50, as an outcome, more RREQ packets are generated and transmitted which lead to a high consumption of the node's buffer which

causes network congestion. This in fact leads to a fewer number of data packets delivered at the destinations, by there degrading the network's performance. But, it can be noticed from Fig. 10b that initially the EDAPR which is constructed two hop NHN set. It knows all non-congested neighbors both one hop and two hops neighbors so that it takes minimum number control packets to find an alternate path than EDCSCAODV and EDAODV. At offered load of 30–50 flows, the packet delivery ratio is increased from 6% to 13% when compared against the EDCSEAODV. Whereas compares with EDAODV, it increases from 10% to 21%. The difference in the achieved packet delivery ratio is due to the reduction of the number of nodes involved in the broadcasting of RREQ packets in congested networks, leading to a reduction of node's buffer occupancy. As a result more communication bandwidth is available that for data transmission.

In regard to routing overhead Fig. 10c shows, when the offer load is low (e.g. 20 flow) EDAPR did not seem better performance than EDCSCAODV and EDAODV. This is because at low offered load, the network congestion level falls in safe zone. When the offer load is increased from 30 to 50 flows, EDAODV incurred the heavy routing overhead and consumed the heaviest control packets to find an alternate path. The EDCSCAODV consumed more control packets to find an alternate path, whereas EDAPR required the least control packers from 16% to 9% as much as the overhead of EDCSCAODV and 33% to 16% over EDAODV. EDAPR seemed unaffected by increasing traffic because, EDAPR resolves congestion by using NHN set which if implicitly distributes over the alternate paths. This was the reason for



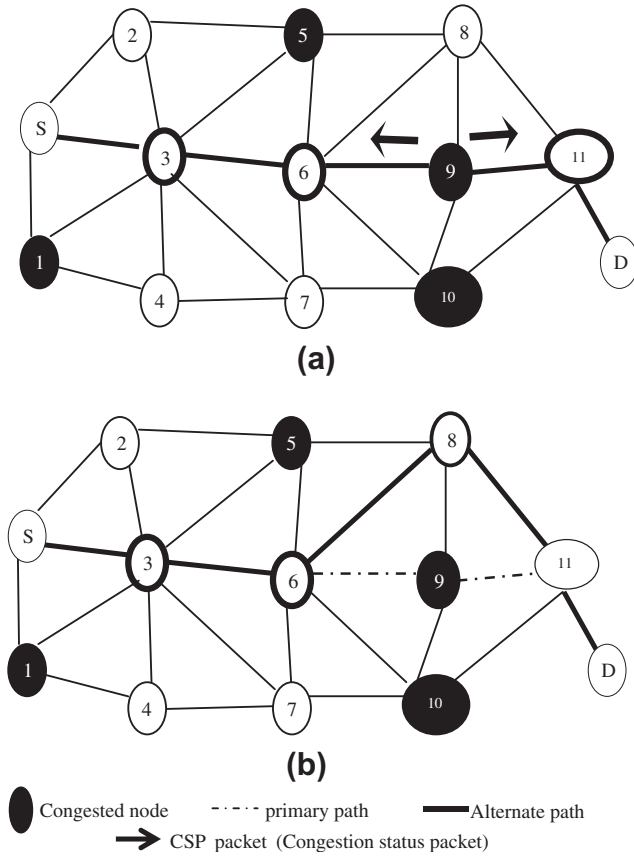


Figure 9 Alternate path finding process.

routing overhead of EDAPR to be less than EDCSCAODV and EDAODV.

#### 5.4. Varying CBR load

In this simulation, the number of connection (different source and different destination) is kept at 20. The CBR sources send data packets to the destinations at different rates, varying from 4 packets/s to 16 packets/s.

Fig. 11a, b and c show the End-to-End delay, packet delivery ratio and Routing overhead for EDAPR, EDCSCAODV and EDAODV respectively.

One observes that the End-to-End delay in EDAPR, EDCSCAODV and EDAODV as shown in Fig. 11a. When the data packet-sending rate is low (less than 8 packets/s), the delay incurred by three protocols increases almost linearly with increase offered load and the delay variation between EDAPR, EDCSCAODV and EDAODV seem unchanged. When the packet rate is high (more than 8 packets/s), the network is in zone II level congestion, EDAODV used bidirectional path discovery mechanism to find alternate path whereas, EDCSCAODV uses self cure mechanism to find an alternate path implicitly but EDAPR uses NHN set, it takes all two hops non-congested node so that it can find an alternate path with minimum cost. The EDAPR demonstrate a reduction in the delay over the EDCSCAODV and EDAODV. This is because the number of forwarding nodes are reduced leading to unnecessary broadcast and network congestion. Compared with both EDCSCAODV and EDAODV at high packet rate (10–16 packets/s), the delay is reduced from 14% to 12% over

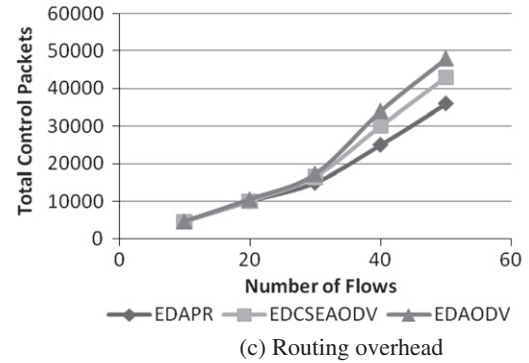
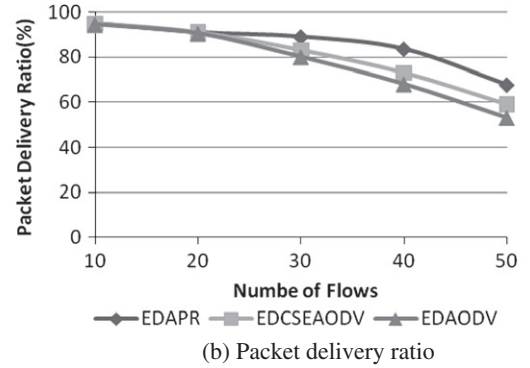
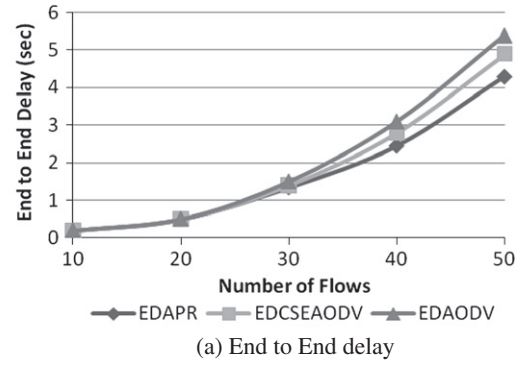


Figure 10 Performance when number connections (source and destination) change.

EDCSCAODV from 20% to 15% is reduced over EDAODV respectively.

In regard of packet delivery ratio Fig. 11b, When the packet rate was small (less than 8 packets/s), EDAPR, EDAODV and EDCSCAODV delivered similar loads of packets. This was because the network traffic was not yet heavy. But, When the packet rate is high (10–16 packets/s), the network comes in zone II level congestion, EDAODV uses bidirectional path discovery mechanism to find alternate path whereas, EDCSCAODV uses self cure mechanism to find an alternate path implicitly but EDAPR uses a NHN set so that it finds an alternate path immediately, EDAPR seems an improved at least 13–15% packet delivery ratio than EDCSCAODV which shows an improvement 19–22% than EDAODV.

Fig. 11c shows the routing overhead between EDAPR, EDCSCAODV and EDAODV. When the traffic load was small (4–8 packets/s), the routing overhead between EDAPR, EDCSCAODV and EDAODV was similar. More impressively, when the traffic was heavier (10–16 packets/s), the routing overhead of EDAPR was reduced from 15% to 13% than

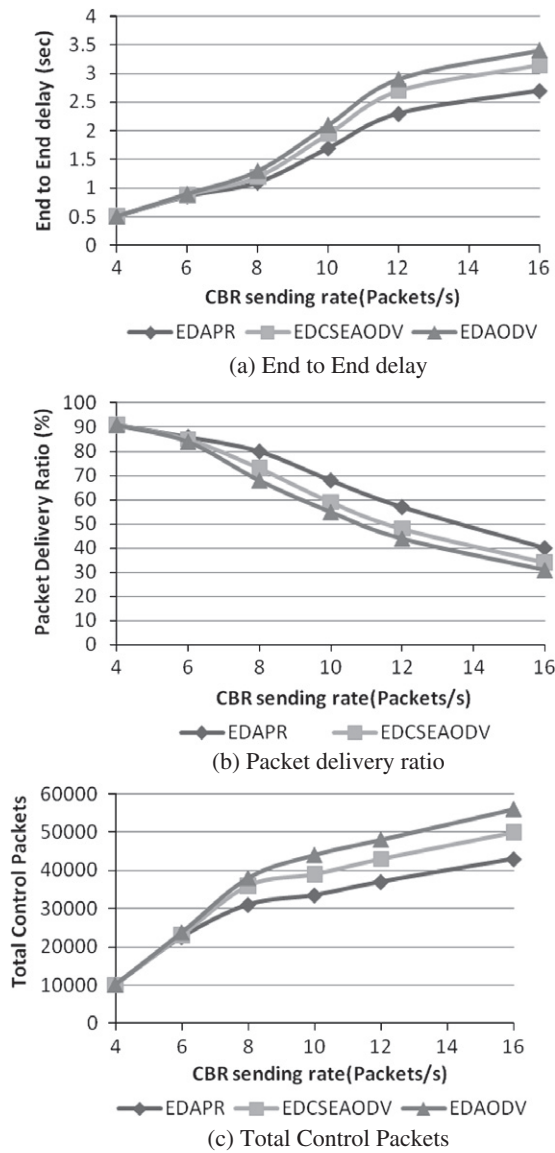


Figure 11 Performance when CBR load changes.

the routing overhead of EDCSCAODV which is reduced from 30% to 22% than EDAODV. The reason is as follows: when the traffic was heavier, while EDAODV tried to find an alternate path to the destination by broadcasting a more route request, where as EDCSCAODV applied self cure mechanism to find an alternate path implicitly as far as EDAPR is concerned initially it found a NHN set which consists of one hop and two hop neighbors so that it can easily find an alternate path than EDCSCAODV and EDAODV. Therefore, less number of route request packets was consumed in over EDCSCAODV and EDAODV. The difference between EDAPR, EDCSCAODV and EDAODV is in terms of delay, delivery ratio and routing overhead, EDAPR seems for better than both.

## 6. Conclusions

In this paper, we have proposed a novel way of accomplishing congestion control in wireless multihop networks: early congestion detection and adaptive routing in MANET

(EDAPR). EDAPR works with fewer packet losses than other techniques that are not adaptive to congestion. This is because EDAPR tries to prevent congestion from occurring in the first place, rather than dealing with it reactively. A key in EDAPR design is the NHN nodes selection. The NHN node is aware of a potential congestion ahead. It finds a non-congested route between source and destination, so that the congestion is controlled as a result. EDAPR does not incur heavy overhead to find non-congested paths because only NHN nodes forward the broadcast control packets during the flooding process. The technique substantially reduces the overhead as compared to the existing flooding mechanism. It also monitors congestion status during data transmission. If any congestion likely to happen, it adapts congestion to find an alternate route through NHN set. EDAPR also provided a short end-to-end delay compared to other techniques. Our ns-2-based simulation has confirmed the advantages of EDAPR and demonstrated reduction of End-to-End delay, routing overhead and improvement of packet delivery ratio over EDAODV and EDCSCAODV.

## References

- [1] Siva Ram Murthy C, Manoj BS. Ad hoc wireless networks – architectures and protocols. Pearson Edu 2007.
- [2] Duc A, Tran, Harish Raghavendra. Congestion adaptive routing in mobile ad hoc networks. IEEE Trans Parallel Distrib Syst 2006;17(11):16–28.
- [3] Corson S, Macker J. Mobile ad hoc networking (MANET): routing protocol performance issues and evaluation considerations. RFC 1999;2501.
- [4] Johnson, Maltz D. Ad hoc networking. Addison-Wesley; 2001.
- [5] Perkins CE. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. Proc ACM SIGCOMM 1994;234–344.
- [6] Perkins CE, Belding-Royer EM, Chakeres I. Ad hoc on demand distance vector (AODV) routing. IETF Internet Draft 2003.
- [7] Gui C, Mohapatra P. A framework for self-healing and optimizing routing techniques for mobile ad hoc networks. ACM Trans Wireless Networks 2008;14(1):29–46.
- [8] Broch J, Johnson D, Maltz D. The dynamic source routing protocol for mobile ad hoc networks. IETF Internet Drafts 1999.
- [9] Christian Lochert, Björn Scheuermann, Martin Mauve. A survey on congestion control for mobile ad-hoc networks. Wiley Wireless Commun Mobile Comput 2007;7(5):655–76.
- [10] Floyd S, Jacobson V. Random early detection gateways for congestion avoidance. IEEE/ACM Trans Networking 1993;1(4):397–413.
- [11] Feng Gang, Agarwal AK, Jayaraman A, Siew Chee Kheong. Modified RED gateways under bursty traffic. IEEE Commun Lett 2004;8(5):323–5.
- [12] Li VOK, Lu Zhenxin. Ad hoc network routing 2004. In: IEEE international conference on networking, sensing and control, vol. 1; 2004. p. 100–5.
- [13] Ramanathan R, Redi J. A brief overview of ad hoc networks: challenges and directions. Commun Mag IEEE 2002;40(5):20–2.
- [14] Royer E, Toh CK. A review of current routing protocols for ad hoc mobile wireless networks. IEEE Pers Commun 1999;6(4):46–55.
- [15] Lee SJ, Gerla M. Dynamic load-aware routing in ad hoc networks. Proc IEEE Intl Conf Commun 2001;3206–10.
- [16] Lu Y, Wang W, Zhong Y, Bhargava B. Study of distance vector routing protocols for mobile ad hoc networks. Proc IEEE Intl Conf Pervasive Comput Commun (PerCom) 2003:187–94.

- [17] Senthil Kumaran T, Sankaranarayanan V. Early detection congestion and control routing in MANET. In: Proceedings of the seventh IEEE and IFIP international conference on wireless and optical communications networks (WOCN 2010); 2010. p. 1–5.
- [18] Senthil Kumaran T, Sankaranarayanan V. Early detection congestion and self cure routing in MANET. In: Proceedings of Springer LNCS computer and information science, vol. 142 (Pt. 3); 2011. p. 562–7.
- [19] Yingqun Yu, Giannakis GB. Cross-layer congestion and contention control for wireless ad hoc networks. *IEEE Trans Wireless Commun* 2008;7(1):37–42.
- [20] Yun-Sheng Yen, Hung-Chieh Chang, Ruay-Shiung Chang, Han-Chieh Chao. Routing with adaptive path and limited flooding for mobile ad hoc networks. *Elsevier transaction on computers and electrical engineering*, vol. 36(2); 2010. p. 280–90.
- [21] NS2 Network Simulator [last accessed 03.02.2000]. <<http://www.isi.edu/nsnam/ns/>>.