A NOVEL TECHNIQUE FOR CONGESTION CONTROL USING RECURSIVE DCDR

Abhaya Pratap Singh[1], Sandeep Sharma[2]

[1]Lal Bahadur Singh Smarak Mahavidyalaya,Gohawar,Bijnor,UP, (India)

[2]School of ICT, Gautam Buddha University, Greater Noida, (India)

ABSTRACT

Congestion is main problem in any network, because it increases the packet loss, increase the delay in delivery of packets. The best solution to this problem is routing with congestion control mechanism. This paper proposes a method for dynamic detection of congestion and its control for the ad hoc networks using the average queue length. Firstly we check the average queue length of a node to find out the congestion status of that node and send the status message to its neighbors, which will help to find the congestion free path to the desired destination. RDCDR provides three paths from source to destination which provides reliable communication in MANETs. We simulate our proposed work using NS-2 version 2.35 and find that RDCDR provides better performance than the AODV, EDOCR, DCDR, EDCSCAODV, EDAODV routing protocols in the heavy traffic and complex network.

Keywords: RDCDR, DCDR, AODV, MANET, CFS, EDOCR, Congestion.

I. INTRODUCTION

In recent time, there are mainly two types of the networks, wired and wireless network. Generally wireless networks are classified into two categories: wireless network with base station and other one is without base station (Ad hoc network). In without base station network, there is no need of infrastructure, this network is called ad hoc network. In an ad hoc network communication created in multi-hop fashion with wireless links between nodes. The ad hoc networks have many advantages over infrastructure networks like low cost, topology [7,9].Congestion is the big problem these days in any network, it may occur in the network if the number of packets transmit through the network is greater than the carrying capacity of the network [10]. Congestion will lead to high packet loss, high end to end delay wastage of resources and throughput of the network.

The main objective of congestion control is to maximize the throughput of the network, and to reduce the packet loss and provide the reliable communication which will improve the performance of the network [2]. In wired network, congestion control techniques are implemented at the transport layer [7]. These congestion control techniques do not apply directly to the ad hoc network because ad hoc network has special challenges like bandwidth, power, node mobility, buffer size [10,13]. One of the solution is routing which will provide the best route for packet delivery [6,7].

There is classification of routing protocols: congestion control routing and congestion non-control routing. Congestion non-control routing are those techniques which only find the shortest path from source to destination like AODV [12], DSDV. Congestion control routing are those techniques which provide the best path from source to destination which will be congestion free like DCDR [8,11]. The aim of this paper is to propose a technique which measures queue length for congestion status and the find three congestion free routes for packet from source to destination node. This technique reduces packet loss, end to end delay and provides surety of delivery and improves the network performance in large and complex network. There are some existing techniques of congestion control routing which we will discuss in next section. EDAODV technique proposes a non congested path bi-directionally [5]. Another technique is self curing the congestion which is called EDCSCAODV [3]. EDOCR divides the network into sparse and dense region on the basis of neighbors which finds the congestion free route in the network [4]. DCDR is another congestion control routing technique which find the congestion free route by using non-congested two-hop neighbors [1].

Our proposed RDCDR uses three paths to transmit the packet to the destination. It uses non-congested path discovery mechanism to prevent network congestion and improve throughput and provide more surety of delivery of the packet. The paper is organized in the following way: section 2 explains the concept of RDCDR. Section 3 will provide the performance of the network when it uses RDCDR and compare it with other routing algorithms. After that we conclude our work in section 4.

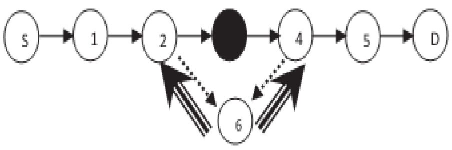
II. RELATED WORK

2.1 Congestion Detection Technique

To detect the congestion status of a node, we compute average queue size and instantaneous queue size. Now we calculate the queue status, which is difference of the instantaneous queue size and average queue size. If the queue status is less than minimum threshold value then queue will be in safe zone. If Instantaneous queue size is greater than minimum threshold but less than maximum threshold then queue is likely to be congested. But if Instantaneous queue size is greater than maximum threshold value then the queue will be congested [5].

2.1.1 Bi-directional Route Discovery

Bi-directional route discovery provides a path when a node is congested. An example of this technique is shown in the fig.1. Here the primary route from S to D node is S.1.2.3.4.5.D. Suppose that node 3 is congested, that time it sends warning to its predecessor and successor node. Then node 2 and 4 identify an alternative route which bypass node 3. And new path will be S.1.2.6.4.5.D [5].



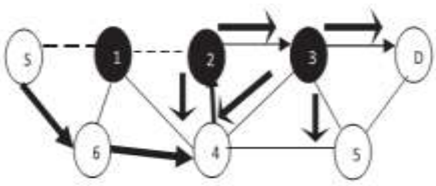


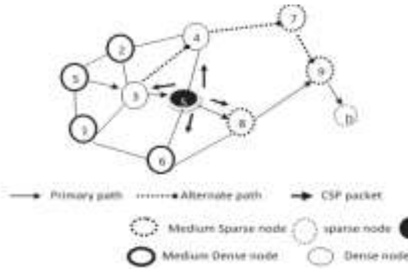
Fig. 1: Bi-directional route discovery [5] **Fig. 2: Self-cure routing [3]**

2.1.2 Self-Cure Routing

Self-cure routing works when there is more than one node are congested. Example of this technique is shown in the fig.2. Here the primary route from S to D node is S.1.2.3.D. But nodes 1, 2 and 3 are congested, at that time these nodes send the CSP packet to their neighboring nodes. Here we apply self-cure routing which bypasses the nodes 1, 2 and 3 in the redirections. Now the new route will be S.6.4.5.D [3].

2.1.3 Optimal Route Discovery

If a node in the primary route is congested that time it sends warning to its neighbors and updates their routing table and find the new route, this is called optimal control routing. As in the fig.3, the primary route is S.3.5.8.9.D, but node 5 is congested, it sends warning to its neighbors and they update their routing tables. Node 3 initiates the alternate route discovery process and find the new route S.3.4.7.9.D [4].



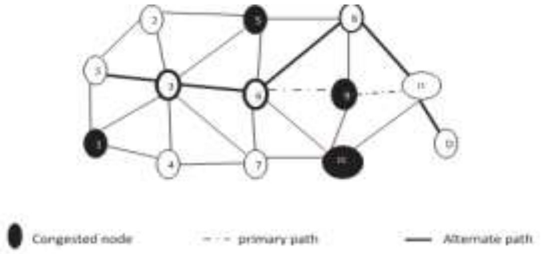


Fig. 3: Optimal route discovery [4] **Fig. 4: Congestion-free route discovery [1]**

2.1.4 Congestion-free Route Discovery

This technique (DCDR) creates the routing table which has the one-hop neighbors as well as two-hop neighbors with their congestion status. Then it finds a primary path which will be non-congested. As in example, fig.4, primary route is S.3.6.9.11.D, but node 9 is being congested. This time node 9 sends the warning to node 6 and 11, which updates routing table and node 6 chooses another non-congested route as S.3.6.8.11.D [1].

III. RECURSIVE DCDR

Recursive DCDR (RDCDR) is a congestion control routing technique for ad hoc networks which provides three non-congested paths from source to destination. It reduces the network congestion and provides guaranteed delivery of the packet to the destination. If the network is already congested that time RDCDR unable to find the route. RDCDR constructs congestion-free set (CFS) to connect both one-hop and two-hop neighbors. Then source finds the congestion-free primary route. With the help of this primary path, RDCDR divides the whole network into three groups and find the non-congested path in each group. This technique helps in the large and complex networks. The RDCDR protocol consists of these components which are discussed here:

3.1 Dynamic Congestion Detection

This algorithm uses three parameters, Mnth, Mxth and Wq, where first two are queue thresholds which are prefixed and last one is queue weight parameter to compute average queue size.

Mnth = 35% Queue_size (1)

Mxth = 2* Mnth (2)

Now to find the average queue length, there is an expression:

$$\text{Avgnew} = (1 - w_q) * \text{Avgold} + \text{Inst\_que} * w_q \quad (3)$$

The weight parameter, w_q, regulates network congestion and work as a time constant of low-pass filter. Initially, w_q, is set to 0.002 [14].

The algorithm for dynamic congestion detection is shown in Algorithm I [1].

Algorithm I: Dynamic congestion detection

```
//initialization
Avgnew = 0
Avgold = 0;
Inst_que = 0
Mnth = 0.35 * queue_size
Mxth = 2*Mnth
Queue_util[] = {0.85,0.875,0.9}
Wq = 0.002;
Warn_line = queue_size / 2
//For each arriving packet in the queue
Inst_que++
//Calculate average queue size
If the queue is not empty then
Avgnew = (1-wq) Avgold+ Inst_que * wq
If (Avgnew < Mnth && Inst_que < Warn_line) then
Begin
Queue_status = ''Safe'';
Else if (Avgnew > Mnth && Avgnew < Mxth) then
Begin
Queue_status = ''Likely to be congested'';
// Initiate Alternate Route Discovery Process
If (Inst_que > Mxth && alter_path = FALSE) then
Maxth = Queue_util[i++]*buff_size;
Else
Queue_status = ''Congested'';
Avgold = Avg;
Wq = Wq*N*P
End
End
For each departing packet in the queue
Inst_que - -
```

Algorithm II: Primary path discovery

```
When the source wants to find a route to a destination
Begin
Construct CFS set for all mobile hosts
/* CFS – congestion free CFS –set nodes of the network */
For each node pair (S, D)i. where i=1 to (N-1) /*D = 2, 3,
   4...N*/
Hops = 0; Routei = Null;
/* Src: source node; Dst: destination node; Route: output
   path set
generated for node pair (S, D), set to be Null */
If (Dst is in two hop list of Si) Then
Path generated for pair (Si,Di)
Set Routei = TRUE
Hops = 2
Else
CFS = Si;
Call Procedure PATH (input:CFS, Di; output: Routei)
End
Procedure PATH (input:CFS, Di; output: Routei)
Begin
If (Dst is in CFS) Then
Path generated for pair (Si,Di)
Set Routei = TRUE
Increment Hops by 1
Return
Else If (CFS-SET is not in Routei) and (CFS-SET's two-
   hop list does not contain Di) Then
/* Hops: number of hops */
Begin
Increment Hops by 1
Add CFS-SET to Routei
For each neighboring node Neib of node CFS-SET Do
/* Neib: the neighbor CFS-set node of CFS-set */
PATH (Neib,Di, Routei)
End
End
```

If the average queue length is less than Mnth, then node is in safe zone. While average queue length is greater than Mnth and less than Mxth, then node is likely to be in congestion. Finally, if the average queue length is greater than Mxth, node will be in congested zone. This component is same as DCDR.

3.2 CFS Construction

The CFS of source host S, denoted by CFS(S), is an arbitrary subset of the non-congested one-hop neighborhood of S must have a link toward CFS(S), and it should not fall in the congested zone [1,7]. CFS initiates a procedure in which each mobile host calculates its congestion status and broadcast its status to its one-hop neighbors. After that each node creates its non-congested one-hop list. Now each node shares its list to know about the non-congested two-hop neighbor nodes and record them. Fig.5 shows the non-congested neighbor information and Table.1 shows process of CFS selection.

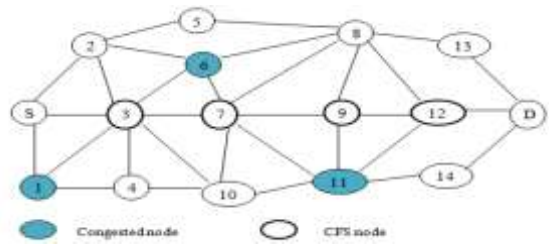


Fig. 5: Neighbor information

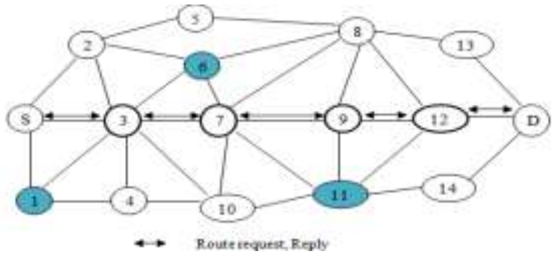


Fig.6: Primary path discovery

3.3 Primary Path Discovery

When we send a packet, the source node creates the route request (RREQ) packet for all of the CFS nodes. Source firstly checks in its two-hop list, if destination found then data packet is transmitted through that path. If not, source broadcasts the RREQ packet to all of the CFS nodes. When CFS node receives a RREQ packet, it checks in its two-hop list, and sends the RREQ packet to destination. This procedure continues till destination not found. Destination node responds to the first RREQ packet, and sends back route reply (RREP) packet. The RREP packet follows the same path to the source and adds the entry to the routing table. This is the primary path between source and destination node.

Fig.6 shows the primary path selection after CFS selection. The source host S has a non-congested one –hop list {2,3} and non-congested two-hop list {4,5,7,10}, and found that destination is not in its two-hop list then source node chooses node 3 and adds it to CFS list.

The procedure repeats and node 3 forwards the RREQ packet to node 7and node 7 forwards to node 9, and finally node 9 finds the destination in its two-hop list, so it forwards the packet to the destination D through node 12. Destination D receives RREQ packet and sends the RREP packet to source which follows the reverse path of RREQ to the source node.

A route S.3.7.9.12.D is found from source S to destination D. This is a non-congested path, called primary path. The primary path discovery algorithm is given in Algorithm II [1].

3.4 Recursive Technique

Now we have our primary path from source to destination node. After it RDCDR divide the whole network into three parts, first part will include the nodes in the primary path, second part will contain the nodes which are on the left side and last part contain the nodes which are on the right side of the primary path including source and destination nodes.

Call the primary path discovery process to find the paths in each set. And we get three paths from source to destination node. If any group has no non-congested path, return no path. At that time we send our packet to those paths and we receive that packet at destination node.

This technique provides the guarantee of receiving the packet at destination node. When a packet receives by the destination node, rest of two packets from other two paths will be discarded by the destination node.

As shown in fig. 6, when the network will divide, first group contains nodes in primary path Gp{S,3,7,9,12} and remaining two groups will be Gl{2,5,6,8,13}, Gr{1,4,10,11,14}. In the last two groups, add the source and destination nodes, and then they forms as Gl{S,2,5,6,8,13,D}, Gr{S,1,4,10,11,14,D}. Now initiate the primary path procedure in each group. But there is no route in Gr because node 1and 11 are congested, but Gl provide a non-congested route S.2.5.8.13.D.

Algorithm III: Recursive technique
Add all nodes to an array 'Ga' where each element of this array is a node of the network.
/* Ga is the set of all nodes in the network. */
For each node in Routei
Add node to Gp
/* Gp is the set of the nodes in the primary path */
Ga = Ga – Gp
For 'i'th node in Ga
left=0;
right=0;
For 'j'th node in Gp
If node 'i' is left to node 'j', then
left++;
If node 'i' is right to 'j', then
right++;
End of the loop of Gp
If left>right, then
Add node 'i' to Gl
Else
Add node 'i' to Gr
End of loop of Ga
Gl = Gl + Src + Dst;
Gr = Gr + Src + Dst;
/* Src : source node, Dst : Destination node*/
Return (Gp, Gl, Gr)
For Gl and Gr
Call primary path discovery
Return two paths Pl and Pr from source node to destination node
/* path Pl for Gl and path Pr for Gr */
Return primary path, Pl, Pr.
End

Node	*One-hop non-congested*	*One-hop congested*	*Two-hop congested*	*Two-hop congested*
S	2,3	1	4,5,7,10	6
2	S,3,5	6	4,7,8,10	1
3	S,2,4,7,10	1,6	5,8,9	11
4	3,10	1	S,2,7	6,11
5	2,8	-	S,3,7,9,12,13	6
7	3,8,9,10	11	S,2,4,5,12,13	1
8	5,7,9,12,13	6	2,3,10,D	11
9	7,8,12	11	3,5,10,13,D	6
12	8,9,D	11	5,7,13	6
13	8,D	-	5,7,9,12	6
14	D	11	12,13	-
D	12,13,14	-	8,9	11

Fig.7: Transmission of Data Packet

Data packet goes to all paths which are provided by the groups. It provides more surety for delivery of packet to the destination. Fig. 7 shows the transmission of data packet after RDCDR and recursive technique is given in Algorithm III. In RDCDR all paths provided by the groups are non-congested.

IV. PERFORMANCE STUDY

To evaluate the performance of RDCDR, we compare it with other routing techniques, DCDR, EDOCR, EDCSCAODV, EDAODV and AODV in ad hoc networks using Network Simulator (NS-2). We consider these metrics for comparison, are end to end delay and packet delivery ratio when we vary the no. of connections and vary CBR rate.

4.1 Simulation Configuration

Table 2 shows various simulation parameters. Fig. 8 shows various simulation results. Part (a) shows end to end delay when number of connections was varied from 10 to 100, part (b) shows packet delivery ratio with variation in no. of connections and CBR rate was 10packets per second, while part (c) shows end to end delay when CBR rate was varied from 10 to 100 packets per second and no. of connections are set at 25 connections.

From the fig.8a, we analyze that when connections are 10, that time all technique reacts as same, but when no. of connections increases, RDCDR demonstrates around 13% reduction in delay over the DCDR, 19% over EDOCR, 27% over EDCSCAODV, 30% over EDAODV and 34% reduction over AODV. The reason is that AODV incurred congestion due to increasing traffic but RDCDR is unaffected because it has its non-congested paths and alternative routes.

MAC	802.11
Bandwidth	2 Mbps
Area	1400m, 1400m
Nodes	100
Antenna	2 Ray ground
Node Placement	Uniform
Queue size	50 packets
Queue type	Random Early Detection
Data traffic	CBR
No. of connection	10-100 connections
Packet rate	10-100 pkts/sec
Pause time	30 seconds
Simulation time	900 seconds

Tab 2: Simulation parameters

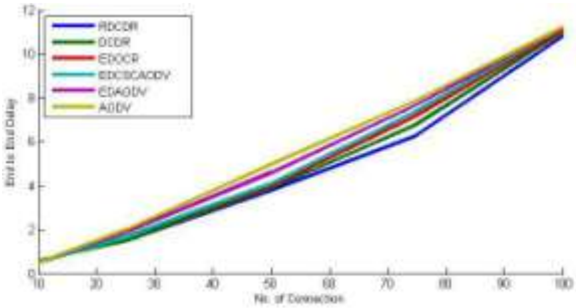


Fig.8a : End to End delay vs No. of connections

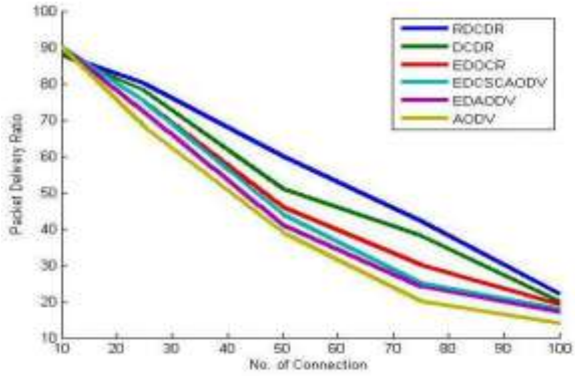


Fig.8b : Packet delivery ratio vs No. of connections

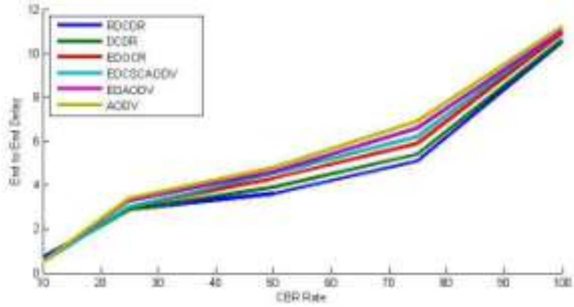


Fig. 8c : End to End delay vs CBR rate

From the fig.8b, we conclude that packet delivery ratio is same when no. of connections is less and as they increases packet delivery ratio decreased because traffic increased and congestion occurs in the network. But RDCDR provide the highest packet delivery ratio than other techniques.

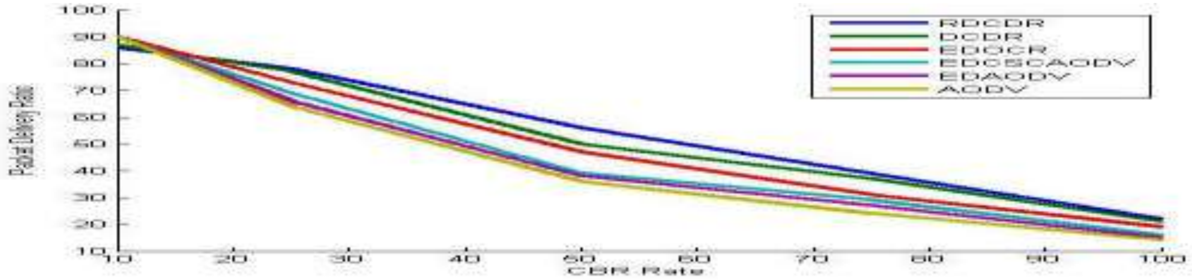


Fig. 8d : Packet Delivery ratio vs CBR rate

Fig. 8c shows that when we increase the CBR rate with fixed number of connections at 25, at lower CBR rate end to end delay for all protocols is similar because there all protocols work greatly on low traffic. As we increase the CBR rate, delay also increases and RDCDR shows better results over other protocols. RDCDR reduced delay 11% over DCDR, 15% over EDOCR, 21.5% over EDCSCAODV, 26% over EDAODV and 29% over AODV. When the traffic was too high, that time all protocols has similar results because network incurred heavy traffic.

Fig. 8d concludes that packet delivery ratio was similar when the packet rate was small. As we increase the packet rate, packet delivery decreases because traffic increases in the network and network goes into the congested network. Here RDCDR performance is better than other techniques. RDCDR improves its performance by 7% over DCDR, 15% over EDOCR, 22% over EDCSCAODV, 28% over EDAODV and 31% over AODV. At highest traffic we can't say that any protocol is best because all protocols packet delivery ratio is less than 20%.

V. CONCLUSION AND FUTURE WORK

Congestion control techniques are used in MANETs mainly, because congestion needs to be detected for a reliable communication. To solve this congestion problem, we have proposed Recursive DCDR technique which estimates the congestion status at node level, RDCDR controls the congestion using non-congested paths and provide more reliable delivery of packet using three paths which are non-congested rather than one. Our simulation results show that RDCDR perform better than DCDR, EDOCR, EDCSCAODV, EDAODV, and AODV for heavy traffic and complex network. It does not provide good performance in low traffic and small network. RDCDR has some limitations also which follows: (i) If network is already congested, that time RDCDR does not able to find the congestion-free path. (ii) This work did not include any wireless losses. One can work on the limitations with which there could be substantial improvement in the performance of RDCDR.

REFERENCES

[1]. Senthilkumaran, T.and V. Sankaranarayanan, "Dynamic congestion detection and control routing in ad hoc networks", Elsevier Transaction on Computers and Electrical Engineering 2013, 25-34.

[2]. Ihsan Ayyub, Qazi, Lachlan, L. H. Andrew, and Taieb Znati, "Congestion Control With Multipacket Feedback", IEEE/ACM Transactions on Networking, vol. 20, no. 6, December 2012, 34-46.

[3]. Senthilkumaran, T. and V. Sankaranarayanan, "Early detection congestion and self cure routing in MANET" , proceeding of springer LNCS Computer and Information Sciences, 2011, 562-567.

[4]. Senthilkumaran, T. and V. Sankaranarayanan, "Early congestion detection and optimal control routing in MANET", European Journal of Scientific Research 63, 2011, 15-31.

[5]. Senthilkumaran, T. and V. Sankaranarayanan, "Early detection congestion and control routing in MANET, In Proceedings of the Seventh IEEE and IFIP International Conference on wireless and optical communication networks 2010, 33-39.

[6]. Yen, Yun-sheng, Chang, Hung-Chieh, Chang, Ruay-Shiung, Chao,Han-Chieh, "Routing with adaptive path and limited flooding for mobile ad hoc networks", Elsevier Transaction on Computers and Electrical Engineering, 2010, 78-83.

[7]. A. Tananbaum, Computer Networks, Fourth edition, Prentice Hall, 2009.

[8]. Yingqum, Y. and G.B. Giannakis, "Cross layer congestion and contention control for wireless ad hoc networks", IEEE Transactions on wireless communication 2008, 128-136.

[9]. Murthy,S.R., and B.S Manoj, " Ad hoc wireless networks architectures and protocols", Pearson Education,Delhi, second ed.2007.

[10]. Lochert, C., Scheuermann, B.and M. Mauve, "A Survey on Congestion Control for Mobile Ad-Hoc Networks", Wiley Wireless Communications and Mobile Computing 7 (5), 2007, pp. 655–676.

[11]. Tran, D.A., Raghavendra, H., "Congestion adaptive routing in mobile ad hoc networks", IEEE Transaction on Parallel and Distributed System, 2006, 164-170.

[12]. Perkins, C.E., Belding-Royer and E.M., Charkers, “Ad hoc on demand distance vector routing”, IETF internet draft, 2003, 236-241.

[13]. Ramanathan, R., Redi, J., “A brief overview of ad hoc networks: challenges and directions”, IEEE communication magazine, 2002, 20-22.

[14]. Floyd, S., 1997. RED: Discussion of Setting Parameters. [Online]. Available from: <http://www.icir.org/floyd/REDparameters.txt>.