ISSN:2231-0711
www.ijcset.net

Performance of Routing Protocol for Ad Hoc Network Using Path Survivability Based on Load and Bandwidth Management under Back Pressure Technique

A K Daniel [*] ,R Singh [**] , J P Saini [***]

[*] *Computer Sc & Engg Department M M M Engineering College GORAKHPUR (U P) India*
[**] *Department of CS & I TM J P Rohilkhand University BAREILLY (U P) India*
[***] *M M M Engineering College GORAKHPUR (U P) India*

Abstract— **Mobile wireless ad hoc network has emerged as the self organized wireless interconnection for the various applications in random topology. However achieving reliable transmission in mobile wireless network is crucial due to change in the network topology caused by node mobility. Congested links are also an expected characteristic of such a network as wireless links inherently have significantly lower capacity than hardwired links and are therefore more prone to congestion. The protocol adapts quickly to routing changes when host movement is frequent, yet requires little or no overhead during periods in which hosts move less frequently. This protocol finds alternate mechanism that will find out optimal and reliable paths in terms of congestion and number of Hop counts, Bandwidth, Traffic load. The route maintenance process is different from of DSR because concept of delay time and alert packets is used.**

Keywords— packet, delay time, stream array, survivability factor, Hop counts, Bandwidth, Traffic load

1.INTRODUCTION

Mobile ad hoc network are communication network built up of collection of mobile devices which can communicate through wireless links. Routing is the task of directing packets from source to destination. This is particularly hard in mobile ad hoc networks due to the mobility of the network elements and lack central control. Source routing is a routing technique in which the sender of a packet determines the complete sequence of nodes through which to forward the packet; the sender explicitly lists this route in the packet's header, identifying each forwarding "hop" by the address of the next node to which to transmit the packet on its way to the destination host. Source routing has been used in a number of contexts for routing in wired networks, using either statically defined / dynamically constructed source routes. The protocol presented here is explicitly designed for use in the wireless environment of an ad hoc network. When a host needs a route to another host, it dynamically determines one based on cached information and on the results of a *path discovery* protocol. Dynamic source routing protocol offers a number of potential advantages over conventional routing protocols such as distance vector in an ad hoc network. First, unlike conventional routing protocols, our protocol uses no periodic routing advertisement messages, thereby reducing network bandwidth It uses *dynamic source routing* to route packets in an ad hoc network. Source routing is a technique in which the source node determines the entire sequence of nodes a packet has to pass through, when it is being transmitted from source to destination. The source node puts the list of addresses of all nodes in the header of the packet, so that the packet is forwarded to the destination through those specified nodes. However source routing can be done statically or dynamically. Here it does dynamically. This is done using a procedure called *path discovery*. Whenever a node has packet to send to some other node, the first node initiates the path discovery. Each node maintains a cache called *path cache* to store the routes it has gathered to different destinations. As the nodes in an ad hoc network move from place to another, some of the existing links break and the routes in the path caches of the nodes must be modified. This is done using a procedure called *path maintenance.* **Path discovery** and **path maintenance** are two main mechanisms used by proposed protocol. Section 2 Related works, section 3 the proposed protocol , finally section 4 simulation result 5 conclusions.

2. RELATED WORKS

The routing concept basically involves, two activities firstly, determining optimal routing paths and secondly, transferring the information packets through network
Routing protocols use several metrics to calculate the best path for routing the packets to its destination. Routing is mainly classified into static routing and dynamic routing. Static routing refers to the routing strategy being stated manually or statically, in the router. Static routing maintains a routing table usually written by a networks administrator. Dynamic routing refers to the routing strategy that is being learnt by an interior or exterior routing protocol. This routing mainly depends on the state of the network. The routing table is affected by the activeness of the destination. Dynamic routing protocols are classified depending on the routers tell each other and how the information kept in to their routing tables. Most of the protocols available in the networks fit into one of the two categories such as Distance vector protocols and Link state protocols. The distance vector routing protocols routers sends two pieces of information first, how far the destination is and secondly, tells in what direction (vector) to get the destination. In link state protocols, a router

doesn't provide the information about the destination instead it provides the information about the topology of the network.
Protocols can be classified into three categories Proactive, Reactive and Hybrid. Proactive routing protocols are table driven and there limitations are amount of data for maintenance and slow reaction on restructuring and failures. Reactive routing protocols find a route on demand by flooding the network with Route Request packets and there limitations are High latency time in route finding. Excessive flooding can lead to network clogging .Hybrid routing protocols are combination of proactive and reactive routing. The routing is initially established with some proactively prospected routes and then serves as the Reactive routing protocols

2.1. Problem Statement: Routing in Ad Hoc Networks
The basic routing problem is that of finding an ordered series of intermediate nodes that can transport a packet across a network from its source to its destination by forwarding the packet along the series of intermediate nodes. In traditional hop-by-hop solutions to the routing problem, each node in the network maintains a routing table. For each known destination, the routing table lists the next node to which a packet for the destination should be sent. The routing table at each node can be thought of as part of a distributed data structure that, taken together, represents the topology of the network. The goal of the routing protocol is to ensure that the overall data structure contains a consistent and correct view of the actual network topology. If the routing tables at some nodes were to become inconsistent, then packets can loop in the network. If the routing tables were to contain incorrect information, then packets can be dropped. The problem of maintaining a consistent and correct view becomes harder as there is an increase in the number of nodes whose information must be consistent, and as the rate of change in the true topology increases. The challenge in creating a routing protocol for ad hoc networks is to design a single protocol that can adapt to the wide variety of conditions present inside ad hoc networks. The alternative to a periodic routing protocol is one that operates in an on-demand fashion. On-demand protocols are based on the premise that if a problem or inconsistent state can be detected before it causes permanent harm, then all work to correct a problem or maintain consistent state can be delayed until it is proven to be needed.
The DSR protocol have the following state of action under link failure occurs

1. When a link failure is reported in DSR the current node propagating a message to its previous node and sends it back to its previous node until it reaches the sender. This process of sending messages hop by hop makes delay or wastage of time. Shown in figure 3.2
2. When a message reporting link failure is being propagated through intermediate nodes, all the encountered nodes have to check whether it is using the broken link in any of its route. The nodes which has not using broken link also check and do unnecessary extra processing.
3. The routes stored in DSR route cache are not arranged in order so any random route may selected based on order in which they stored in route cache. Thus might end up with route which has inferior path then leave behind a better one. as stored.

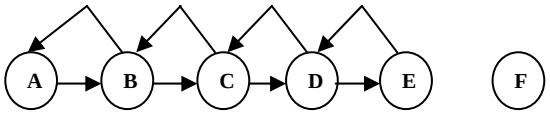


Figure1. Operation of DSR when a link failure is reported to the source node

Each mobile host participating in the ad hoc network maintains a *route cache* in which it caches intermediate source routes. When host sends a packet to another host, the sender first checks its route cache for a source route to the destination. If a route is found, the sender uses this route to transmit the packet. If no route is found, the sender may attempt to discover by using the *route discovery* protocol.
Route Maintenance is the mechanism by which a node sending a packet along a specified route to destination detects that route has broken, as the two nodes have moved too far apart. Each node along the route forwards the packet to the next hop indicated in the packet's header, and attempts to confirm that the packet was received by that next node; a node may confirm this by means of acknowledgment, passive acknowledgment. After a limited number of local retransmissions of the packet, a node in the route is unable to make this confirmation, it returns a Route Error to the source of the packet, identifying the link is fail / broken. The sender then removes this broken link from its Route Cache; for subsequent packets to this destination, the sender may use any other route to destination from its Cache, it may attempt a new Route Discovery for that target if necessary. Unlike routing protocols using distance vector or link state algorithms, DSR uses dynamic source routing which adapts quickly to routing changes when host movement is frequent.

3. PROPOSED PROTOCOL

The Proposed protocol based on DSR algorithm for selecting optimal routing path only. The proposed protocol uses an **alert packet** and **UPD packet**.

3.1 Alert Packet

Alert packet is a type of acknowledgement packet. It is small size packet of fixed length (i.e. its size will not increase as it visits node). It is sent from destination to source at regular intervals of time (**delay time**) on the same path that has been selected for sending data packets (in reverse order). Its purpose is to tell source that the current working path is still valid.[4]

3.1.1 Delay Time

The Delay time will depend upon Traffic and Distance between two farthest pair. The average time required by the alert packet to traverse the most distantly separated nodes in the network (i.e. diameter of the tree) at the condition of average congestion level i.e. to the number of

hops necessary for a packet to reach from any host located at one extreme edge of the network to another host located at the opposite extreme, as the *diameter* of the network. In case of congestion free conditions the time taken by a message to travel to and fro between two nodes is at least 2 times the packet to travel in one direction. So, in ideal conditions a source is supposed to get acknowledgement after 2 times the time the packet travel in one direction. So, we have assumed that the source gets the acknowledgement after approximately 2.5 times the time taken by the packet to travel in one direction.[5][6][7]

3.1.2 Traffic:

Traffic is the average congestion that can exist in longest path of the network for timely delivery of alert packet in worst case (i.e. highly congested network).

3.1.3 Distance between Two Farthest Pair

It is quite obvious that the average time taken by the alert packet to reach from destination to source will always be less than or equal to the average time taken by the alert packet to traverse the longest distance possible in the same network i.e. delay time for the diameter of the network will be maximum.

3.2 Proposed Algorithm

The algorithm uses initial phase of DSR algorithm for route construction and for deciding all possible paths between given source and destination. Every node has a *path cache*. *Path Cache* is a kind of buffer that holds information about paths emanating from that node as a source. When a node has a packet to send to some destination and does not currently have a route to that destination in its *Path Cache*, the node initiates **Path Discovery** to find a route; this node is known as the *source* of the path discovery, and the destination of the packet is known as the *destination*. The source transmits a **Route Request Packet (RRP)** as a local broadcast, specifying the target and a unique identifier from the source. Each node possesses a **stream array (S)** of **Size** equal to number of adjacent neighbors to that node. For determining the size of the stream array, let a node ***i*** broadcasts a small packet to its adjacent neighbours. Now, all nodes which receive this packet respond by sending another small packet to node ***i***. The number of respond packet the node ***i*** receives is equal to the size of the stream array. This process is done at regular intervals of time . Initially all elements of this array are assigned to ***null***. Each element can have only two values either 0 or 1. A **upstream** link from neighbour node to given node is denoted by " 0" and a **downstream** link from given node to neighbour node is denoted by " 1".The downstream link from ***i*** to ***j*** (or upstream link from ***j*** to ***i***) doesn't mean that the link ***i→j*** will be used for transferring packet from ***i*** to ***j***. It only means that the path from the source to destination will contain a link ***i→j*** i.e. data packet will be transferred from ***i*** to ***j*** and the acknowledgement will follow link ***j*** to ***i***.[8][9]

3.3 Path Discovery

Whenever a node has a packet to send to another node it checks its *path cache* first. If there is no existing route for the required destination node in the path cache, the source node broadcasts a *route request* with a *sequence number* and *destination address*. The *sequence number* identifies a particular route request. Each node maintains a table called *request table*, which maintains the information about all the route requests of the node has received before. The source node keep a note of the route request in the request table.

Let ***i*** be the node that broadcasts RRP and ***j*** be one of the adjacent neighbours of ***i***. Whenever node ***i*** broadcasts a RRP it updates only those elements of its stream array that have ***null*** value present in it by inserting 1 in $\mathbf{S_{ij}}$ where ***j*** belongs to one of the neighbours of ***i*** . Correspondingly, neighbours that will receive RRP will update their stream array by inserting 0 in $\mathbf{S_{ji}}$ where ***j*** receives packet from ***i***. Now, ***j*** will also broadcast this received RRP to all its adjacent neighbours and updates only those elements of its stream array which have ***null*** value present in it. After receiving RRP, neighbours of ***j*** will respond in similar until destination node is found. When a node receives the route request, it checks its stream array. And if the node is the destination of the route request, it sends a route reply back to the source node by simply copying the path from the **RRP**. If the node is an intermediate node, it checks its path cache. If it has a route to the destination in its cache, the node sends a route reply to the source node by copying the path from its cache and the path from the **RRP**. If the intermediate node does not have any available route to the destination in its path cache, it rebroadcasts the **RRP**.After receiving the first Route Reply Packet RRP, by source sends all the data packets to the destination using the path retrieved from the route reply.Additional route reply that source node gets will act as Backup path or an **alternate** to the current working path. [10][11][12][13]

3.4 Desired Characteristic For Selection Of Path

The route reply packet contains three parametric factor values for path determination

3.4.1 Number Of Hops Travelled (N)

The route reply packet travels back from destination to source node it counts number of intermediate nodes traversed. as Hops count

3.4.2 Traffic Load (T) At Any Node

The outgoing and incoming traffic at any node may be calculated by the size of output and input buffer of that node. Note that input buffer may act as output buffer and vice versa.

Traffic Load (T) = (Outgoing traffic at that node) - (Incoming traffic at that node).

The incoming traffic is more than outgoing traffic or traffic at the node is more define as negative traffic load and denoted as **0**

The outgoing traffic is more than incoming traffic or traffic at the node is less define as positive traffic load and denoted as **1**

As route reply packet travels from destination to source it extracts value from every travelled node. Finally, when it reaches the source node it has the traffic load value of all the nodes of corresponding paths. The more number of zeros in any path, more the traffic in the path and less the number of zeros in the path, less the traffic.

3.4.3 Bandwidth (B)

The route reply RRP travels from destination to source it visits all nodes and links joining those nodes that come in path. So, as it travels it stores the available bandwidth of the link. The minimum value among all traversed links is the available Bandwidth **B** for data transmition .

3.4.4 Survivability Factor (SF)

The Survivability factor of every path will be calculated on the basis of Bandwidth, Traffic Load, Hops parameters.

SF=function(B,T,N)

SF α 1/N
SF α 1/T
SF α B
SF α B /N*T
SF α B /N*T
SF = K B /N*T

K is system constant of network K $\geq$ 1 and
SF$\rightarrow$0 mean N and T are very large
SF$\rightarrow\infty$ mean N and T are small
More the number of hops (N) less survivable is the path. As the number of hops increase number of links will also increase so chances of link failure is more. More the traffic load (T) in the path less survivable is the path. More the bandwidth (B) more survivable is the path.On the basis of SF we shall priorities the paths .The more the SF more the priority of survivability of the path.

3.4.5 Logical Path Construction

The directed tree will be constructed for a given network which contain all the possible paths between source and destination. The tree structure will be formed, and all the subsequent route replies nodes and links are created that are not present in current tree, if a node is already present in the tree then a threaded link will be formed.The source will be the root of the tree and destination will always be one of the leaf nodes. This tree will contain all the possible paths from the source to the destination. This tree will be stored in the root cache of the source node.[14][15]16]

3.5 Proposed Model

Mobile Ad-hoc Network (MANET) has with available bandwidth B and no of nodes be n and distance between nodes is D and load at each node be T
The following figure-1 shows the wireless network of nodes as Base Station,
B= Total Available Bandwidth
n_i=Nodes Name
T_i=Load at node n_i

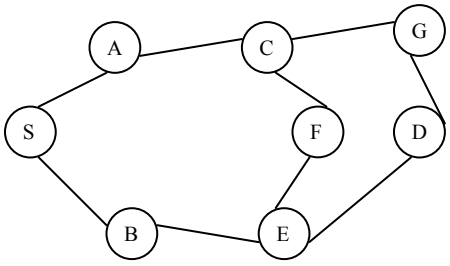
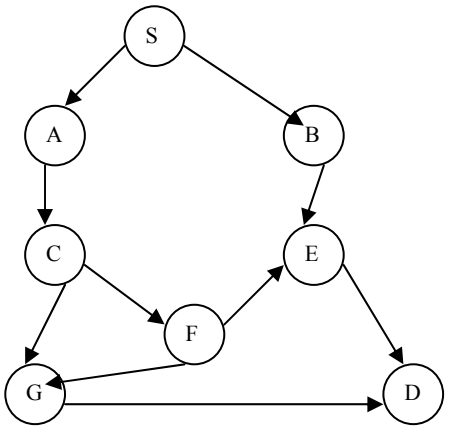


Figure -2

Logical Path Construction



Values In The Stream Array Of Each Node

Figure -3

For Node ***S***, $S_{sa}=S_{sb}=1$

For Node ***A***, $S_{ac}=1$ And $S_{as}=0$

For Node ***C***, $S_{cg}=S_{cf}=1$ And $S_{ca}=0$

For Node ***B***, $S_{be}=1$ And $S_{bs}=0$

For Node ***E***, $S_{ed}=1$ And $S_{eb}=S_{ef}=0$

For Node ***F***, $S_{fe}=S_{fg}=1$ And $S_{fc}=0$

For Node ***G***, $S_{gd}=1$ And $S_{gc}=S_{gf}=0$

For Node ***D***, $S_{dg}=S_{de}=0$

Paths Created

So in order to select path from S to D It select path from the following

1. ***S→A→C→G→D***
2. ***S→A→C→F→G→D***
3. ***S→A→C→F→E→D***
4. ***S→B→E→D***

3.5.1 Selection Of Path

1. Distance is based in the number of hop counts. Distance of selected path is minimum or optimum.
2. Traffic Load in selected path is minimum

Hence ***S→B→E→D*** is the selected path

3.5.2 Algorithm For Path Discovery

```
{ Get the next node based on the current node value}
Begin
Step 1
i   sends RRP to all adjacent neighbours
If  Stream array  Sij = null
      Sij ← 1
Node j received   RRP   and    Sij ← 0
If j is destination node
Then
```

```
Send RRP
Else
  If j has any path to D in route catch
Then send RRP
Else i=j go to step 1
Else Node j reject the RRP as

Sij ≠ null
End.
```

3.6 Path Maintenance

Whenever a node ***i*** sends a data packet to the next hop, it waits for the acknowledgement from the receiving node ***j***. Node ***i*** may not receive the acknowledgement packet for some reason. The reason may be due to the failure of link present between ***i*** and ***j*** or due to high congestion present in that link. If the node ***i*** detects that the link to the next hop (i.e. ***j***) is in error; it sends the data packet to previous node in that path (i.e. ***k***) and checks for the availability of any path from that node to destination in its path cache.

There are following two cases

(A) If it finds any path it changes the route map of the data packet and also sends an UPD packet to the source ***s*** informing about the particular link failure and amended path. After receiving the UPD packet the source deletes all paths that contain failed link and also updates newly chosen path as the current working path.

(B) If path is not found it simply sends the UPD packet to the source informing about the particular link failure. Now source does the processing according to the following conditions

(i) If UPD packet reaches the source before the expiry of the delay time, source deletes all those paths that contain the failed link. Now the source chooses the alternate path having the highest priority from its path cache.

(ii) If delay time expires before the arrival of UPD packet source ***S*** chooses alternate path that have highest priority from its path cache and sends the data packet through it. If source doesn't have any alternate path then it again starts the path discovery process.

Let the link between F and E fails The Dashed line between nodes F and E shows the the link ***F→E*** is broken. Shown in Figure-4

Where F is ***ith*** node, ***E*** is ***jth*** node, and node C ***kth***.

When link ***F→E*** failure path cache of C is checked for presence of any path to D.If found, then data packet is sent to D through that path and a UPD packet is sent to S for informing about the new path and also of particular link failure. The source S will delete all the paths containing the failed link. If not found then only UPD packet is sent to S informing it about the link failure. In this case again, if the UPD packet arrives at S before the delay time expires, the source will delete all the paths containing the failed link. Also, the path with maximum value of survivability factor will be chosen from the remaining ones. But if the delay time expires before the UPD packet arrives at S, the source will choose new path having highest value of survivability factor and send data packet through it. Now, when UPD packet arrives source S checks whether the new chosen path contains failed link or not. If not S continues to send packets through the selected path and also deletes all paths having the failed link present in it from its path cache. If yes then it cancels the current transaction and also deletes all paths that contain failed link. Then, it chooses the new path having the highest value of survivability factor.

3.6.1 Algorithm For Path Maintenance

```
{ Get the node and link of the working path }

If      {Link failed}
        i→j
Then i  send packet to k where k is i-1 node  &
Check for Alternative path from Route Catch
If  (Alternative path found)
Then
Change the route map of packet and send UPD packet to
Source s for information and delete all path contain failed
links

Else
 Send UPD packet to source
If
   UPD packet reaches before delay time expires
Then
Choose new path
```

4. SIMULATION, RESULT

The performance of the protocol is evaluated using simulation experiments with C++, Ns-2 simulator with Mobility Framework .A flat network is assumed as clusters Networks .A Node sends a packet, to sets RTS (Request-to- Send) flags of its neighbors and the intended receiver sets CTS (Clear-to-Send) flags of its neighbors. Nodes whose RTS or CTS flag is set cannot transmit data, except the sender. When the sender finishes sending the data, RTS/CTS flags are cleared by the nodes which originally set those flags. The Node wants to send a packet sets RTS flags of its neighbors, and each intended receiver sets CTS flags of its neighbors. Therefore, in broadcast, collision may occur. However, collisions are ignored in our simulation. The simulated network area is a N x N meter square, and M mobile nodes are roaming randomly in all directions at a predefined speed in this area. Each Node has a finite buffer, and packets are lost when buffer overflow occurs. Control packets have higher priority over data packets in simulations Propagation delay is assumed to be negligible, and it is assumed that packets always arrive without any bit error. The source Node generates packets at a constant rate. Extensive simulation results obtained by varying several network parameters and workload configuration The values of the network parameters used in simulations are those specified in the IEEE 802.11. We evaluate the performance improvement in terms of throughput due to the use of a densely populated network. Specifically, we consider a network of 5 to 80 Node with an increasing number of neighbors from 5 to 20 nodes. Each Node has a

traffic flow with infinite demands towards one of its neighbors. In Figure-5 to Figure-6 shows the throughput of all traffic flows, with available Channels Bandwidth.

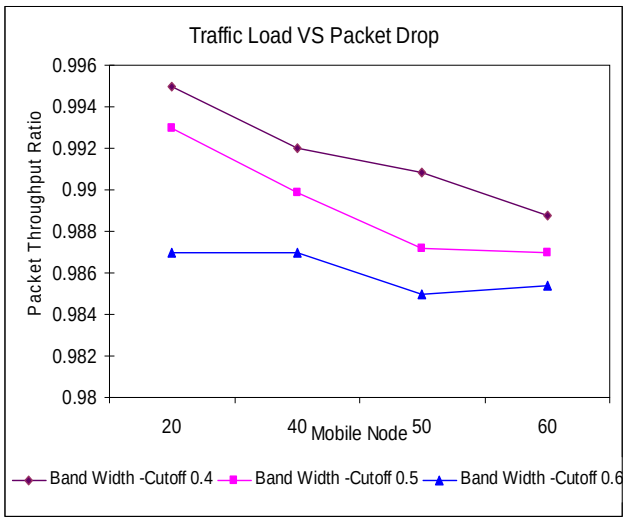


Figure- 4

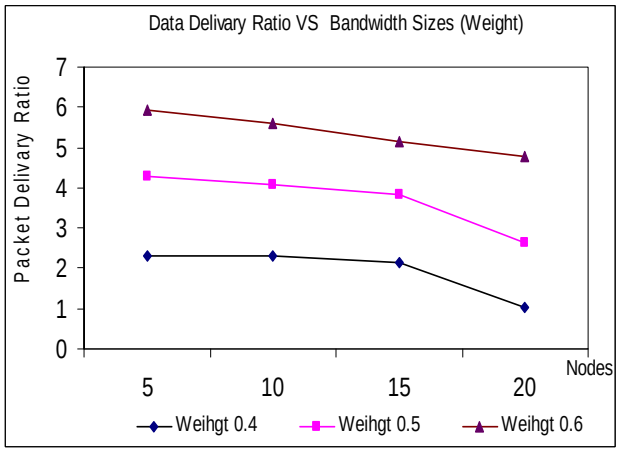


Figure- 5

5. Conclusions

The proposed protocol finds multiple paths from a source to destination. The above protocol extracts merits from DSR, exploits it by augmenting additional features of alert packets and delay time. The intended path between any given pair shall be achieved in timely and efficient manner with a frugal usage of resources. A useful data structure tree has been employed for the accomplishment of the set of objectives of the algorithm. Besides giving multiple paths it identifies the exact location of the link failure. The paths are arranged in order of decreasing survivability factor that how much survivable or reliable the path. The path with more value of survivability factor is chosen first (in case of link failure). The reliability of the chosen path will be more so, performance improved

6 Future Scope

In future there can be further evaluation of our scheme by using more realistic mobility of nodes in the simulation. We believe the advantage of providing traffic information will be significant in those environments.

References

1. C. K. Toh. A Novel Distributed Routing Protocol to Support Ad–Hoc Mobile Computing. In Conference Proceedings of The 1996 IEEE Fifteenth Annual International Phoenix Conference on Computers and Communications, pages 480–486, March 1996.
2. David B. Johnson and David A. Maltz. "Dynamic source routing in ad hoc wireless networks", Mobile Computing, Kluwer Academic Publishers, edited by Tomasz Imielinski and Hank Korth, chapter 5, pages 153–181, 1996.
3. S. R Das, C. E Parkins and E. M Royer: Performance Comparison of Two on Demand Routing Protocols for ADHOC Networks, in Proceedings of IEEE INFOCOM 2000, Tel Aviv, Israel, 2630 Mar 2000.
4. Dan Yu and Hui Li: A Model for Performance Analysis of Mobile Ad hoc Networks, Mobile Network and Applications, Vol. 6, No. 3/June 2001.
5. Chi-Chun Lo, Bin-Wen Chuang," A Novel Approach of Backup Path Reservation for Survivable High-Speed Networks", IEEE Communications Magazine • March 2003,pp 146-152.
6. Vincent D. Parka and M. Scott Corsonb," A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks", Copyright 1997 IEEE. Published in the Proceedings of INFOCOM'97, April 7-11, 1997 in Kobe, Japan
7. Thomas Stidsen, Bjorn Petersen, Kasper Bonne Rasmussen," Optimal Routing with Single Backup Path Protection", IEEE Communications Magazine, 40(1):34–41, 2002
8. ChristianLochert, Bjorn Scheuermann, Martin Mauve, "A Survey on Congestion Control for Mobile Ad-Hoc Networks", published in Wiley Wireless Communications and MobileComputing7(5),pp.655–676,June2007.
9. S.Karunakaran & P.Thangaraj," A CLUSTER BASED CONGESTION CONTROL PROTOCOL FOR MOBILE AD HOC NETWORKS", International Journal of Information Technology and Knowledge Management July-December 2010, Volume 2, No. 2, pp. 471-474
10. C.E.Perkins, E.M.Royer, S.R.Das, and M.K.Marina, "Performance comparison of two on demand routing protocols for ad hoc networks,"IEEE Personal Communications, vol.8, pp.16–28, Feb.2001.
11. David B. Johnson. Routing in Ad Hoc Networks of Mobile Hosts. In Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'94) , pages 158–163, December 1994.
12. David B. Johnson and David A. Maltz. Dynamic Source Routing in Ad HocWireless Networks. In Mobile Computing, edited by Tomasz Imielinski and Hank Korth, chapter 5, pages 153–181. Kluwer Academic Publishers, 1996.
13. David B. Johnson , David A. Maltz , Yih - Chun Hu, and Jorjeta G. Jetcheva. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. Internet-Draft, draft- ietf-manet-dsr-07.txt, February 2002.
14. Josh Broch, David B. Johnson, and David A. Maltz. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. Internet-Draft, draft-ietf-manet-dsr-03.txt, October 1999. Earlier revisions published June 1999, December1998, and March 1998.
15. David B. Johnson. Routing in Ad Hoc Networks of Mobile Hosts. In Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications, pages 158–163. IEEE Computer Society, December 1994.
16. C. K. Toh. A Novel Distributed Routing Protocol to Support Ad–Hoc Mobile Computing. In Conference Proceedings of The 1996 IEEE Fifteenth Annual International Phoenix Conference on Computers and Communications, pages 480–486, March 1996.