ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

Efficient Broadcasting in Mobile Ad hoc Networks Using AODV with TORA and Restricted Flooding

(Wireless/ Mobile Communication)

Suresh Kumar P.G.V[1], Shaheda Akthar[2]

Research Scholar, Department of CSE, Acharya Nagarjuna University, Andhra Pradesh, India[1]

Lecturer, Department of CSE, Govt. College for Women, Guntur, India[2]

ABSTRACT*:* *Broadcasting* is a process of sending message from one node to all other nodes in the network. In most of the existing protocols high memory consumption, route discovery and route maintenance is to be difficult. In this paper we propose the concept of AODV (Ad-hoc On-Demand Distance Vector protocol) with TORA (Temporally-Ordered Routing Algorithm routing protocol) and restricted flooding technique. The main objective of our approach for efficient broadcasting, to reduce memory requirements, discover the route easily and maintenance, quickly re-establish when the link fails. Restricted flooding is avoiding the unwanted nodes traversal when on-demand route creation.

KEYWORDS: Broadcasting, TORA, AODV, Mobile Ad hoc Networks, Restricted flooding.

I. INTRODUCTION

Broadcasting is process of sending a packet from a source node in the network to all other nodes in the network. Ad hoc networks are wireless, mobile networks that can be set up anywhere and anytime outside the internet or any another preexisting network infrastructure. A mobile ad hoc network (MANET) is a combination of mobile hosts that can communicate with each other.

In many existing protocols like ABR [1], ZRP [1], and LAR [1]-[2] it provides medium communication overhead. In our proposed protocol route recovery process through reverse link in TORA and AODV route recovery process through notify source and local repair. AODV (Ad-hoc On-Demand Distance Vector protocol) creates new routes when a demand occurs. In TORA (Temporally-Ordered Routing Algorithm routing protocol) feature mainly control packets are localized to area of topology change. TORA provides next available node information in the network. Restricted flooding is a technique to control unwanted nodes traversal when on-demand route creation by the AODV.

Accordingly AODV routing protocol is the best one for general mobile ad hoc networks compared with DSDV (Destination Sequenced Distance Vector) routing protocol [3].

In this paper we use Temporally Ordered Routing Algorithm (TORA) and AODV (Ad-hoc On-Demand Distance Vector protocol) for efficient broadcasting. This two algorithm and the routing techniques are consume less bandwidth and lower overhead compared with DSDV.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

The AODV protocol contains a route table to store the next-hop routing information for destination nodes in the networks. In each routing table can be used for a period of time. If a route is not requested within that period, it expires and a new route needs to be found when demanded. Each time a route is used, its lifetime is updated.When a source node has a packet to be sent to a given destination, it looks for a route in its route table.

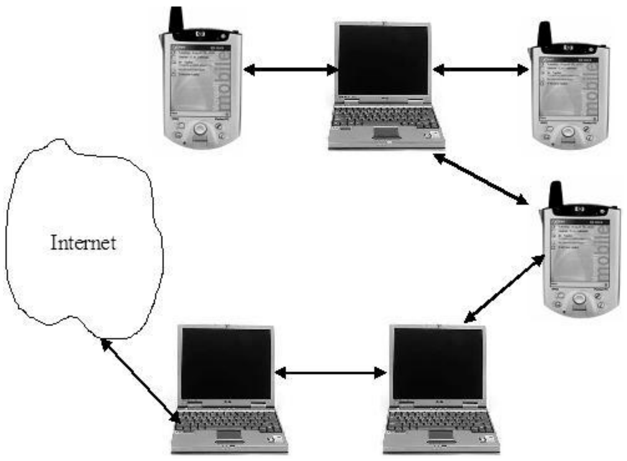

Fig. 1. Sample of a MANET

As illustrated in Fig 1. Example of a MANET, Mobile host is linked with internet through wireless devices.

In case there is one, it uses to transmit the packet. Otherwise, it initiates a route discovery procedure to find a route by broadcasting a route request (RREQ) message to its neighbors. Upon receiving a RREQ message, a node performs the following actions

- Checks for duplicate messages and discards the duplicate ones.
- Creates a reverse route to the source node (the node from which it received the RREQ is the next hop to the source node).
- Checks whether it has an unexpired and more recent route to the destination (compared to the one at the source node).

In case those two conditions are hold, the node replies to the source node with a RREP message containing the last known route to the destination. Otherwise, it retransmits the RREQ message [1].

A reactive protocol tends to be more efficient than proactive protocols in terms of control overhead and power consumption because routes are created only when it’s required. On the other hand, proactive protocols need periodic route updates to keep information updated and valid. In addition, many available routes might never be needed, which increases the routing overhead [1].

TORA is adaptive and scalable routing algorithm based on the concept of link reversal. It finds multiple routes from source to destination in a highly dynamic mobile networking environment. An important design concept of TORA is that control messages are localized to a small set of nodes nearby a topological change. Nodes maintain routing information about their immediate one-hop neighbors. The protocol has three basic functions are route creation, route maintenance, and route erasure.

The main strength of the protocol is the way it handles the link failures. TORA's reaction to link failures is optimistic that it will reverse the links to re-position the DAG for searching an alternate path [1].

ADOV and TORA routing protocol should provide efficient broadcasting service to the network through route creation when demand and maintenance the route. AODV provide on-demand routing service to the node when the necessary.

The rest of this paper is organized as follows. In Section II, we review related work. We define some terms and describe the concept broadcasting algorithms and its categories. The TORA technique and its operations are explained in section III. ADOV working methods are described in section IV. The performance analysis described in section V. We conclude the work in VI.

II. RELATED WORK

Generally, TORA and AODV are reactive protocols. Reactive protocols are more specific in Ad Hoc networks. Once the final destination has been reached by these broadcasts, an answer is built and forwarded back to the source. This source can then transmit the data on the newly discovered route. Each device used for forwarding the routing packets has learned the route at the same time. The disadvantage of this design is the amount of routing traffic exchanged between devices. In the case of a large topology, the traffic will be spread on each link until the end node is found. It also can result in a high latency.

Reactive protocols are the most advanced design proposed for routing on Ad Hoc networks. They define and maintain routes depending on needs. There are different approaches for that, but most are using a backward learning mechanism or a source routing mechanism.

Reactive protocols consist of two major phases:

1) Route Setup Phase:

In this phase, a route between the source and destination is setup on demand. The basic mechanisms (and their parameters) used in this phase are:

a) Flooding: It is responsible for distributing the source's route request in the network. Its parameter is the range of flooding, which is specified by the Time To Live (TTL) field in the IP header.

b) Caching: Caching is an optimization to reduce the overhead of flooding. If a node has a cached route to the destination, it will reply to the source's route request. Its parameter is whether aggressive caching should be used. i.e. should the nodes use all the overheard route replies and should they cache multiple routes to the destination.

2) Route Maintenance Phase:

This phase is responsible for maintaining the path between the source and the destination. The basic mechanisms used in this phase are Error Detection, Error Notification and Error Recovery.

A. Basic idea of Restricted Flooding

Restricted flooding is a technique to avoid the unwanted nodes traversal in the geographic region. Flooding having more traffic load, high cost for node traversal, collision in the traversal from source node to the all the nodes in that region, whereas restricted flooding not permitted this type collision, contention.

Distance from the source node to the destination is used to determine nodes participation in the route discovery process. In restricted flooding all nodes participating in a per-application overlay receive the data; however, the nodes receive the information once. The data does not necessarily traverse every link in the overlay, it must reach every node.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

III. THE TORA TECHNIQUE

A . Basic idea of TORA

TORA uses a metric referred to as the height of the node to assign a direction to links for forwarding packets to a given destination.

The Temporally Ordered Routing Algorithm (TORA) is a highly adaptive, efficient and scalable routing algorithm. It is a source-initiated on-demand protocol and it finds multiple routes between the source and the destination. TORA is a fairly complicated protocol but its main feature is that when a link fails the control messages are only propagates around the point of failure. While other protocols need to re-initiate a route discovery when a link fails, TORA would be able to patch itself up around the point of failure. This feature allows TORA to scale up to larger networks but has higher overhead for smaller networks.

B. Functions of TORA

There are three functions: creating routes, maintaining routes and erasing routes.

- **Creating routes:** Creating routes is performed on demand using a query/replay process.
- **Maintaining routes:** When a node loses its last downstream link the algorithm reorients the directed acyclic graph such that all downstream paths lead to the destination.
- **Re-optimization of routes:** TORA does not compute the shortest path: paths may be suboptimal. It starts close to optimal and tends to *loosen*, as it reacts to topological changes. A secondary mechanism, not tied to the rate of topological change, is used to re-optimize routes.
- **Partition detection and erasing routes:** Partitions are detected when a *reversal* reaches a node with no downstream links and all of its neighbors have the same *reflected reference level*, which it previously defined. A node that detects a partition initiates the process of erasing the invalid routes.

C. Operations in TORA

The TORA attempts to achieve a high degree of scalability using a flat, non-hierarchical routing algorithm. In its operation the algorithm attempts to suppress, to the greatest extent possible, the generation of far-reaching control message propagation. In order to achieve this, the TORA does not use a shortest path solution, an approach which is unusual for routing algorithms of this type.

TORA builds and maintains a Directed Acyclic Graph rooted at a destination. No three nodes may have the same height. Information may flow from nodes with higher heights to nodes with lower heights. Information can therefore be thought of as a fluid that may only flow downhill. By maintaining a set of totally-ordered heights at all times, TORA achieves loop-free multipath routing, as information cannot flow uphill and so cross back on itself.

The main enhancement comes from the introduction of a time value which stores the time since a link failure. TORA also maintains a DAG by means of an ordered quintuple with the following information: t time of a link failure, oid originator id, r reflection bit indicates 0=original level 1=reflected level, d integer to order nodes relative to reference level, i the nodes id. The triplet (t,oid,r) is called the reference level and the tuple (d,i) is said to be an offset within that reference level. The heights of the nodes for a given destination to each other determine the direction of the edges of the directed acyclic graph.

The DAG is destination oriented (routed at the destination) when the quintuples which represent the heights are maintained in lexicographical order, the destination having the smallest height, traffic always flowing downstream. Heights are however not needed for route discovery; instead a mechanism as in LMR is used. Nodes which do not currently need to maintain a route for themselves or for others won't change a height value.

Each node has a Route- required flag for that purpose, additionally the time since the UPD (update-) packet was sent is recorded. Each node maintains a neighbour table containing the height of the neighbour nodes. Initially the height of all the

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

nodes is NULL. (This is not zero "0" but NULL "-") so their quintuple is *(-,-,- ,-,i)*. The height of a destination neighbour is *(0,0,0,0,dest)*.

In the principle of TORA any data packet transmission is always routed from a higher to a lower neighborhood. Since the destination is the node with the lowest height the packets are flowing down to the destination [6].

D. Route maintenance in TORA

Route maintenance in TORA has five different cases according to the flowchart below:

Case 1) Generate: The node has lost its last downstream link due to a failure. The node defines a new *reference level*, so it sets oid (originator id) to its node id and t to the time of the failure. This is done only if the node has upstream neighbours. If not it sets its height to NULL.

Case 2) Propagate: The node has no more downstream link due to a link reversal following the receipt of an update packet and the reference levels (t,oid,r) of its neighbours are not equal. The node then propagates the references level of its highest neighbour and sets the offset to a value which is lower (-1) than the offset of all its neighbours with the maximum level.

Case 3) Reflect: The node has lost its downstream links due to a link reversal following the receipt of an update packet and the reference heights of the neighbours of the node are equal with the reflection bit not set. The node then reflects back the refence height by setting the reflection bit.

Case 4) Detect: The node has lost its downstream links due to a link reversal following the receipt of an update packet and the reference heights of the neighbours of the node are equal with the reflection bit set. This means that the node has detected a partition and begins the route erasure procedure. The height values are set to NULL.

Case 5) Generate: The node has lost its last downstream link due to a link reversal following the receipt of an update packet and the reference heights of all the neighbours are equal with the reflection bit set and the oid of the neighbours heights isn't the node's id. The node then sets t to the time of the link failure and sets oid to its own id. The d value is set to 0. This means that the link failure required no reaction. The node experienced a link failure between the time it propagated a higher reference (from someone else) and the time this level got reflected from a place further away in the network. Because the node didn't define the new reference level itself this is not necessarily an indication of a partitioning of the network. So the node simply defines a new higher reference level with the time of the link failure.

IV. WORKING METHODS IN AODV

AODV is main aim to reduce the number of broadcast messages sent to the nodes throughout the network by discovering routes on-demand instead of keeping complete up-to-date route information. AODV is a very simple, efficient, and effective routing protocol for Mobile Ad-hoc Networks which do not have fixed topology.

A. Algorithm for AODV Routing Protocol

```
// S is the source node; D is the destination node
// RT = Routing Table
S wants to communicate with D
 if RT of S contains a route to D
S establishes communication with D else
   S creates a RREQ packet and broadcasts it to its neighbors
   // RREQ contains the destination Address (DestAddr),
 // Sequence Number (Seq) and Broadcast ID (BID) for all nodes N receiving RREQ
    if (RREQ was previously processed) discard duplicate RREQ
    end if
    if (N is D)
       send back a RREP packet to the node sending the RREQ
    else if (N has a route to D with SeqId >= RREQ.Seq)
       send back a RREP packet
    else
       record the node from which RREQ was received broadcast RREQ
    end if
 end for
 while (node N receives RREP) and (N != S)
    forward RREP on the reverse path
    store information about the node sending RREP in the RT end for
  S receives RREP
  S updates its RT based on the node sending the RREP S establishes communication with D
end if
```

B. Working of AODV

Each mobile host in the network acts as a specialized router. The routes are obtained as when needed, thus making the network self-starting. Each node in the network maintains a routing table with the routing information entries to its neighbouring nodes, and two separate counters are a node sequence number and a broadcast-id. When a node (say as, source node ‘S’) has to communicate with another node (say as, destination node ‘D’), it increments its broadcast-id and initiates path discovery by broadcasting a route request packet RREQ to its neighbors. The RREQ contains the following fields [9]:

- Source-addr
- Source-sequence - to maintain freshness info about the route to the source.
- Dest-addr
- Dest-sequence - specifies how fresh a route to the destination must be before it is accepted by the source.
- Hop-cnt

The pair (source-addr, broadcast-id) is used to identify the RREQ uniquely. Then the dynamic route table entry establishment begins at all the nodes in the network that are on the path from S to D.

RREQ travels from node to node, it automatically sets up the reverse path from all these nodes back to the source. Each node that receives this packet records the address of the node from which it was received. This is called as reverse path setup.

The nodes maintain this information for enough time for the RREQ to traverse the network and produce a reply to the sender and time depends on network size.

If an intermediate node has a route entry for the desired destination in its routing table, it compares the destination sequence number in its routing table with that in the RREQ. If the destination sequence number in its routing table is less than that in the RREQ, it rebroadcasts the RREQ to its neighbors. Otherwise, it unicasts a route reply packet to its neighbor from which it was received the RREQ if the same request was not processed previously (this is identified using the broadcast-id and source-addr).

Once the RREP is generated, it travels back to the source, based on the reverse path that it has set in it until travel to this node. As the RREP travels back to source, each node along this path sets a forward pointer to the node from where it is

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

receiving the RREP and records the latest destination sequence number to the request destination. This is called forward path setup.

If an intermediate node receives another RREP after propagating the first RREP towards source it checks for destination sequence number of new RREP. The intermediate node updates routing information and propagates new RREP only,

– If the Destination sequence number is greater, or

– If the new sequence number is same and hop count is small, or Otherwise, it just skips the new RREP. This ensures that algorithm is loop-free and only the most effective route is used [3].

Step by step explanation of Fig. 2 is as follows:

1. Source ’S’ has to send data to destination.
2. S sends RREQ to its neighbors A, B, C.
3. B finds the path in its routing table (with destination seq-number s1 and hop count c1) and sends RREP to S.
4. C sets up reverse path.
5. C forwards RREQ to its neighbors D and E.
6. E sets up reverse path.
7. E forwards RREQ to its neighbors F and G.
8. E deletes the reverse path after a time out period as it does not receive any RREP from F and G.
9. D finds the path (with destination seq-number s2 which is greater than s1 and hop count c1) in its routing table and sends RREP to C.
10. C receives RREP from D and sets up forward path and forwards RREP to S.
11. A sets reverse path; forwards RREQ to its neighbors; receives RREP (with path of hop count c2 which is greater than c1); sets forward path; and forwards this RREP to S.
12. S receives a path info from C (with destination seq-number s2 and hop count c1), another path info from B (with destination seq-number s1 and hop count c1), and another path info from A (with destination seq-number x which is less than s1 and s2 and hop count c2 which is less than c1).
13. S chooses path info from C (which was originated from D), giving first priority to the path with greatest destination sequence number and then second priority to the path with smallest hop count. Though path given by A is of smallest hop count, it is ignored because the destination sequence number is greater than the path from C.

C. Route Table Management

Each mobile node in the network maintains a route table entry for each destination of interest in its route table. Each entry contains the following information:

– Destination
– Next hop
– Number of hops
– Destination sequence number
– Active neighbors for this route
– Expiration time for the route table entry

The other useful information contained in the entries along with source and destination sequence numbers is called soft-state information associated to the route entry. The information about the active neighbors for this route is maintained so that all active source nodes can be notified when a link along a path to the destination breaks. And the purpose of route request time expiration timer is to purge the reverse path routing entries from all the nodes that do not lie on the active route [3].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 2, February 2016

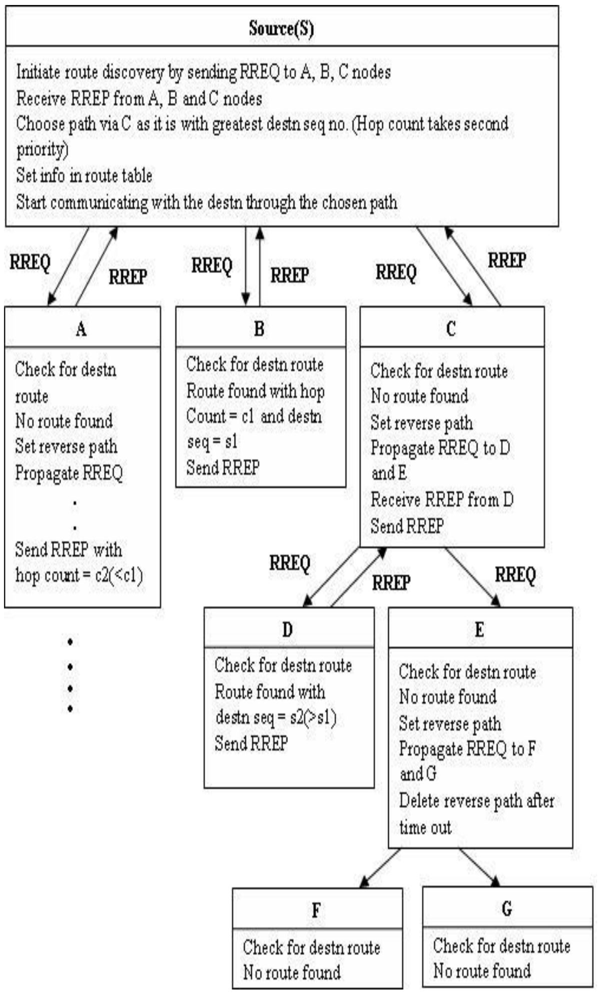
Source(S)
Initiate route discovery by sending RREQ to A, B, C nodes
Receive RREP from A, B and C nodes
Choose path via C as it is with greatest destn seq no. (Hop count takes second priority)
Set info in route table
Start communicating with the destn through the chosen path
RREQ
RREP
A
Check for destn route
No route found
Set reverse path
Propagate RREQ
Send RREP with hop count = c2(<c1)
B
Check for destn route
Route found with hop Count = c1 and destn seq = s1
Send RREP
C
Check for destn route
No route found
Set reverse path
Propagate RREQ to D and E
Receive RREP from D
Send RREP
D
Check for destn route
Route found with destn seq = s2(>s1)
Send RREP
E
Check for destn route
No route found
Set reverse path
Propagate RREQ to F and G
Delete reverse path after time out
F
Check for destn route
No route found
G
Check for destn route
No route found

Fig.2. Route finding process in ADOV Routing protocol.

The above Fig. 2 is an example, which shows how the route to the destination is found by AODV routing protocol.

V. PERFORMANCE ANALYSIS

In order to evaluate the performance of the TORA and AODV are route discovery procedure is involved when the routes are requested by the source and special route request packets are flooded to the network starting with the immediate node neighbors. The route discovery process comes to an end when once a route is formed or multiple routes are obtained to the destination. Route maintenance. procedure maintains the information of active routes for the duration of their lifetimes.

Performance of TORA was compared with Ideal Link-State (ILS)[8].TORA outperformed ILS (in terms of bandwidth utilization and end-to-end delay) under conditions of high rates of topological change. As network size was increased, TORA outperformed ILS at lower rates of change.

TORA performs better (than ILS) as rate of topological change is increased and as network size is increased. TORA is a fairly complicated protocol but its main feature is that when a link fails the control messages are only propagates around the point of failure. While other protocols need to re-initiate a route discovery when a link fails, TORA would be able to patch itself up around the point of failure. This feature allows TORA to scale up to larger networks but has higher overhead for smaller networks.

The Ad hoc On Demand Distance Vector (AODV) routing algorithm is a routing protocol designed for ad hoc mobile networks. AODV is capable of both unicast and multicast routing. It is an on demand algorithm, meaning that it builds routes between nodes only as desired by source nodes. It maintains these routes as long as they are needed by the sources. Additionally, AODV forms trees which connect multicast group members. The trees are composed of the group members and the nodes needed to connect the members. AODV uses sequence numbers to ensure the freshness of routes. It is loop-free, self-starting, and scales to large numbers of mobile nodes.

Combining this TORA, AODV routing protocols we get minimize communication overhead, less memory consumption, route discovery and route maintenance is easy compare with other routing protocols. Reactive protocols tend to be more efficient than other protocols in terms of control overhead and power consumption because routes are only created when needed. Restricted flooding is to control the unwanted nodes traversal.

VI. CONCLUSION

Thus we have presented TORA and ADOV combining this two routing protocol for reducing time delay when the failure occurs to re-establish the route and reducing memory consumption due to the requirements. In this combination with restricted flooding increase nodes traversal is easy through reducing unnecessary nodes restricted. TORA is more scalable and reliable algorithm on mobile networks. If the network gets wide and the mobility is high, the message overhead for the link reversals having to be propagated through the network can get rather important. The AODV is to reduce the number of broadcast messages sent through the entire network by discovering routes on-demand instead of keeping complete up-to-date route information.

Future study includes are implemented in secure ADOV with TORA for route discovery process and based on 1-hop neighbor information for reducing time complexity of forwarding nodes to efficient broadcasting in mobile ad hoc networks.

REFERENCES

[1] Azzedine Boukerche, “Algorithms and Protocols for Wireless Mobile Ad.Hoc Networks”, Wiley IEEE Press Nov 2008.
[2] Young-Bae Ko and Nitin H. Vaidya, “Location-Aided Routing (LAR) in mobile ad hoc networks “, Wireless Networks 6 (2000) 307–321
[3] Krishna Gorantala, “Routing Protocols in Mobile Ad-hoc Networks”, Master’s Thesis in Computing Science, June 15, 2006.
[4] Narayanan Sadagopan, Fan Bai, Bhaskar Krishnamachari, Ahmed Helmy, “PATHS: analysis of PATH duration Statistics and their impact on reactive MANET routing protocols”.
[5] C. Perkins, “Ad Hoc on Demand Distance Vector (AODV) Routing”, IETF Internet draft, work in progress, 1997.
[6] V. Park, S. Corson, “Temporally Ordered Routing algorithm (TORA) version 1 Functional specification”, Park 1977.
[7] Qing He , Huanbei Zhou , Hui Wang , Li Zhu, “Performance Comparison of Two Routing Protocols Based On WMN” IEEE 2007.
[8] Vincent D. Park, M. Scott Corson, “A Performance Comparison of the Temporally-Ordered Routing Algorithm and Ideal Link-State Routing”
[9] C. Perkins and E. Royer. Ad hoc on-demand distance vector routing. In Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications, 1999, pp. 99–100.