Volume 4, Issue 1, January 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

DSR Protocol Optimization in MANET through Real Life Analogy

Jaideep Atri*
DCSA,Kurukshetra University , INDIA

Swati Atri
DCSA,Maharishi Dayanand University , INDIA

Abstract— A mobile ad-hoc network (MANET) is formed by a cluster of mobile hosts, each installed with a wireless transceiver, without the assistance of base stations. Several routing protocols, such as DSR, AODV, and ZRP, have been proposed for a MANET with a dynamically changing topology and each protocol in some way or other are analogous to real life situation. But in MANET, a route may suddenly become stale due to the mobility of hosts and even if one host moves away, a route may become nonexistent. The DSR protocol is an on demand and source routing based protocol. Energy, congestion and delay like in real life have always been among the important constraints to the protocols that should be kept in mind to provide efficient routing .A new optimization in DSR protocol is proposed in this paper considering an analogy to real life situation example.

Keywords— MANET, Routing, On demand, Source routing, Congestion, Delay and Optimization.

I. INTRODUCTION

The key distinguishing feature of DSR [3, 7] is the use of source routing. That is, the sender knows complete route to the destination. These routes are maintained in the route cache. The data packets carry the source route in the packet header. This situation can be explained to real life experience of a visit to unknown place. For example when we plan to visit an unknown place first of all we try to find out all the midway milestones so that we may have a route in our mind and may not get lost. When a node in the ad hoc network attempts to send a data packet to a destination for which it does not already know the route, it uses its *on demand* behaviour to dynamically determine such a route by *Route discovery* which works by flooding the network with *route request* (RREQ) packets. The Dynamic Source Routing protocol (DSR) provides excellent performance [1, 2, 5] for routing in multi-hop wireless ad hoc networks. Each node receiving an RREQ rebroadcasts it, unless it is the destination or it has a route to the destination in its route cache. Such a node replies to the RREQ with a *route reply* (RREP) packet that is routed back to the original source. RREQ and RREP packets are also source routed. The RREQ builds up the path traversed across the network. The RREP routes itself back to the source by traversing this path backward .The route carried back by the RREP packet is cached at the source for future use. If any link on a source route is broken, the source node is notified using a *route error* (RERR) packet. The source removes any route using this link from its cache. A new route discovery process must be initiated by the source if this route is still needed. DSR makes very aggressive use of source routing and route caching. No special mechanism to detect routing loops is needed. Also, any forwarding node caches the source route in a packet it forwards for possible future use. Several optimizations like gratuitous route repair, promiscuous listening [4] have been proposed and have been evaluated to be very effective by the authors of the protocol. In this paper a new optimization to the DSR protocol has been proposed taking example from the real life.

II. BASIC OPERATIONS

A. *Basic DSR Route Discovery*

The Route Discovery mechanism is carried out in two steps:
Firstly a *Non Propagating Route* request is sent with a hop limit set to zero. This situation is similar to the case when in order to look for someone to deliver him a message we first of try to locate him in our neighbourhood but if he is not found there than starts finding him elsewhere. The second option is carried out using the *Propagating Route* request .In case of *Propagating Route* request when some node **S** originates a new packet destined to some other node **D**, it places in the header of the packet a *source route* giving the sequence of hops that the packet should follow on its way to **D**. Normally, **S** will obtain a suitable source route by searching its *Route Cache* of routes previously learned, but if no route is found in its cache, it will initiate the *Route Discovery* protocol to dynamically find a new route to **D**. In this case, we call **S** the *initiator* and **D** the *target* of the Route Discovery.
For example, Figure 1 illustrates an example of *Route Discovery*, in which a node **A** is attempting to discover a route to node **E**. To initiate the Route Discovery, **A** transmits a RREQ message as a single local broadcast packet, which is received by (approximately) all nodes currently within wireless transmission range of **A**. Each RREQ message identifies the initiator and target of the Route Discovery, and also contains a unique *request id*, determined by the initiator of the RREQ. Each RREQ also contains a record listing the address of each intermediate node through which this particular copy of the RREQ message has been forwarded. This route record is initialized to an empty list by the initiator of the Route Discovery.

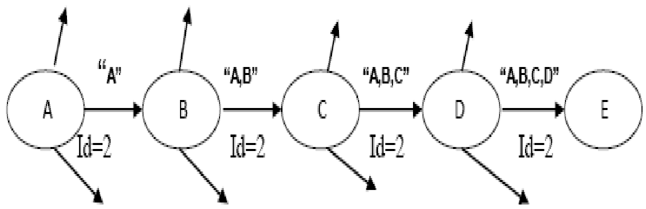

Figure 1: Route Discovery example: Node **A** is the initiator, and node **E** is the target.

When another node receives a RREQ, if it is the target of the Route Discovery, it returns a RREP message to the initiator of the Route Discovery, giving a copy of the accumulated route record from the RREQ; when the initiator receives this RREP, it caches this route in its Route Cache for use in sending subsequent packets to this destination. Otherwise, if this node receiving the RREQ has recently seen another RREQ message from this initiator bearing this same request id, or if it finds that its own address is already listed in the route record in the RREQ message, it discards the request. Otherwise, this node appends its own address to the route record in the RREQ message and propagates it by transmitting it as a local broadcast packet (with the same request id). In returning the RREP to the initiator of the Route Discovery, such as node **E** replying back to **A** in Figure 1, node **E** will typically examine its own Route Cache for a route back to **A**, and if found, will use it for the source route for delivery of the packet containing the RREP. Otherwise, **E** may perform its own Route Discovery for target node **A**, but to avoid possible infinite recursion of Route Discoveries, it must piggyback this RREP on its own RREQ message for **A**. Node **E** could also simply reverse the sequence of hops in the route record that it trying to send in the RREP, and use this as the source route on the packet carrying the RREP itself. When initiating a Route Discovery, the sending node saves a copy of the original packet in a local buffer called the *Send Buffer*.

B. *Basic DSR Route Maintenance*

Most important operation after route discovery is the *route maintenance* [3, 7]. When originating or forwarding a packet using a source route ,each node transmitting the packet is responsible for confirming that the packet has been received by the next hop along the source route; the packet is retransmitted (up to a maximum number of attempts) until this confirmation

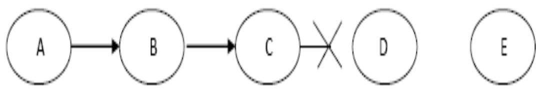

Figure 2: Route Maintenance: Node **C** is unable to forward a packet from **A** to **E** over its link to next hop **D[7]**

of receipt is received. For example, in Figure 2, node **A** has originated a packet for node **E** using a source route through intermediate nodes **B**, **C**, and node **D**. In this case, node **A** is responsible for receiving packet at **B**, node **B** is responsible for receiving packet at **C**, node **C** is responsible for receiving packet at **D**, and node **D** is responsible for receiving packet finally at the destination **E**. If the packet is retransmitted by some hop the maximum number of times and no confirmation message is received, this node returns a RERR message to the original sender of the packet, identifying the link over which the packet could not be forwarded. For example, in Figure 2 shown above , if **C** finds itself unable to deliver the packet to the node **D**, then **C** returns a RERR to **A**, stating that the link from **C** to **D** is currently "broken". Node **A** then removes this broken link from its cache; any retransmission of the original packet is a function for upper layer protocols such as TCP. For sending such a retransmission or other packets to this same destination **E**, if **A** has in its Route Cache another route to **E** (for example, from additional route replies from its earlier Route Discovery, or from having overheard sufficient routing information from other packets), it can send the packet using the new route immediately. Otherwise, it may perform a new Route Discovery for this target.

This situation can be compared to the real life example of failure in the delivery of registered post. For example when any sort of fault is detected in the postal address the post is declared undelivered and the post is sent back to the sender address in order to look for some other alternative.

C. *Replying to ROUTE REQUESTs using Cached Routes*

A node receiving a RREQ for which it is not the target, searches its own *Route Cache* [7] for a route to the target of the RREQ .If found, the node generally returns a RREP to the initiator itself rather than forwarding the RREQ. In the RREP, it sets the route record to list the sequence of hops over which this copy of the RREQ was forwarded to it, concatenated with its own idea of the route from itself to the target from its Route Cache. This situation is analogous to real life situation when we succeed in locating a person home address and keeps it in our memories in order to locate him again if needed.

III. LIMITATION OBSERVED AND PROPOSED SOLUTION

All routing protocols that use on-demand Route Discovery must include some kind of route caching system, since the originator of a packet cannot afford the cost of doing a Route Discovery operation for every packet it wishes to send.

Once the originator discovers a route through the network, it must remember the route in some kind of cache for use in sending future packets. DSR, in particular, makes even greater use of the route cache, using it not only to cache routes for the purpose of originating packets, but also for the purpose of allowing nodes to answer route requests targeted at other nodes. Now, if the route saved in the route cache gives a route error message when the data packet is sent onto it then this situation can arise due to following two situations:

i. Either due to movement of some intermediate node

ii. or due to the movement of destination node.

Now this situation can be best understood from the real life situation as below. For example let a person A have to reach a hospital and he has some clues with him regarding the path to be followed .Like he knows that in order to reach to hospital he first need to approach B who can then guide him to the way to hospital and B stays near college which is on the way to hospital. In case of normal situation he can make it easily to the hospital as shown above by just going to B and then moving to hospital by getting the route from B. But if after reaching to the college if A get to know that B

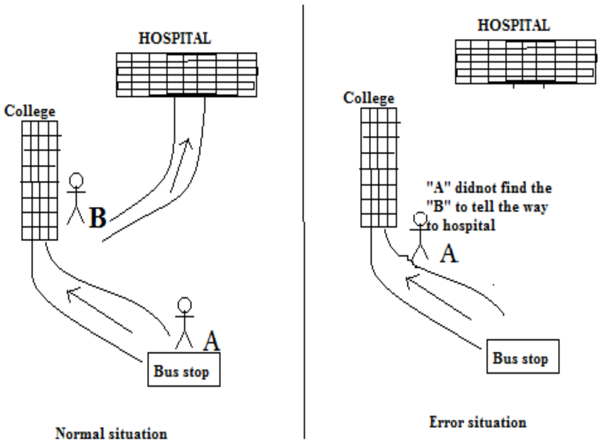

Figure 3: Real life example

is not here then in this situation in spite of going back to bus stop and then start looking for the hospital A will try to find hospital from college itself. This situation is similar when compared to DSR routing shown below in which when B discovered a route error to destination Z it informed this error back to S which is the source node. And then S will start a route discovery again for fresh route to Z by broadcasting the route request messages over whole network.

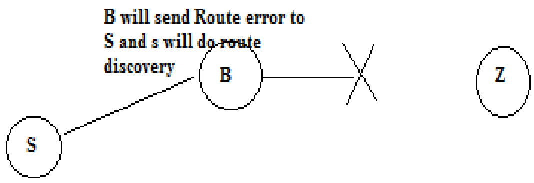

Figure 4: Route maintenance in case of unavailability of route

When the broadcasting of route request packets is done over whole network from the source node it leads to congestion over the network and also leads to wastage of energy which is utilised by the different forwarding nodes over the network. So in spite of route discovery to be carried out by S the route discovery for Z should start from B considering one more point in view that nodes always moves within a limited scope. It does look simple and of little help in the situation above which is showing just one intermediate node but can prove to be utter most help in the situation as below where the number of intermediate nodes are significant since in case of error if node Y in spite of informing the s will itself start to discover the route to Z which should lie close to it rather than S. This can prevent a lot of wastage of energy as well as the time. The node Y after locating Z through route discovery can send the route to the source node S which can transmit the packet over this link.

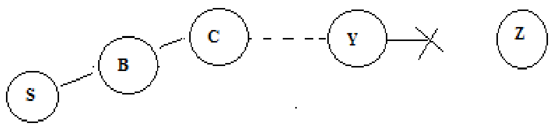

Figure 5: Situation showing Route error close to destination on a longer route.

Now one more optimization that can be implied here from real life is that as in the case above when A reached the college and didn't find B .one of the possible reason of B for not being there at the college can be that probably B could have moved to some another place and since the probability of any object for moving within certain period of time is limited to a particular area. So in spite of trying to find the hospital directly which can exist far away from the college why not we should try to firstly locate B within certain area and if it is not found there then we may move on with fresh route discovery from college this case can be explained diagrammatically as in Figure 6. The first case in the figure shows the possible area in which B can move which in case of DSR protocol can be some limited number of node from the intermediate node. However while choosing the possible area of movement the factors which should be considered are

I. It should not be too large that the source route would have become stale.

2 it should not be too small that the missing node would have moved out of it.

Thus a trade off between above two factors can be decided and we may try to look for missing node upto some limited number of nodes from the node where the error actually occurred now as shown in the other cases of figure if the B is found within this possible area then we may carry on further from B but if it is not found then we may carry on with route discovery for the destination node from the intermediate node.

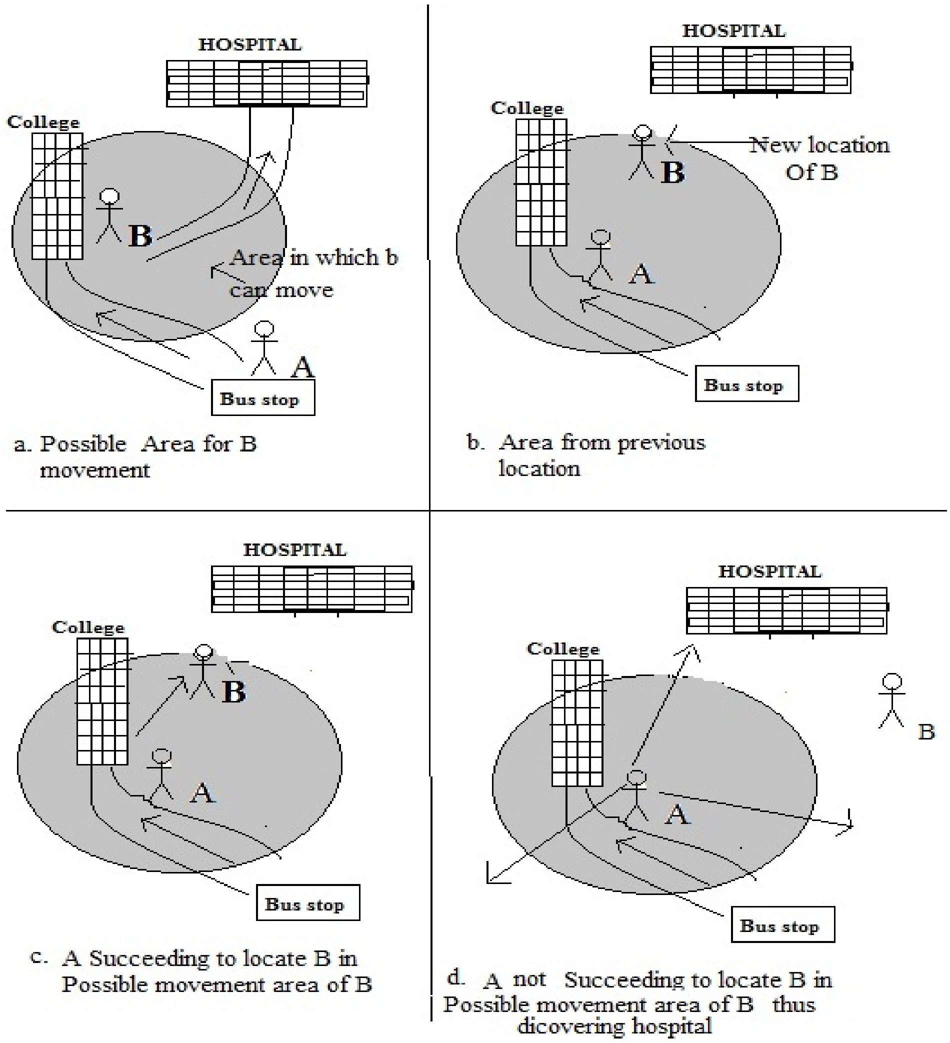

Figure 6: Optimization by guessing possible movement of intermediate node.

IV. CONCLUSION

The Dynamic Source Routing protocol (DSR) provides an excellent performance for routing in multi-hop wireless ad hoc networks. A key reason for this good performance is the fact that DSR operates *entirely* on demand. In this paper, the principle mechanisms of *Route Discovery* and *Route Maintenance* used by DSR are presented and a new optimization is proposed in the DSR protocol after considering the real life analogy to DSR protocol.

REFERENCES

[1] A. Tuteja, R. Gujral and S. Thalia, "Comparative Performance Analysis of DSDV, AODV and DSR Routing Protocols in MANET using NS2," IEEE/ACE , june 2010.

[2] 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 Feb. 2001.

[3] D. Johnson and D. Maltz , "Dynamic Source Routing in Ad Hoc Wireless Networks," T. Imielinski and H. Korth, Eds. Mobile Computing, Ch. 5, Kluwer, 1996.

[4] D. Maltz et al., "The Effects of On-demand Behavior in Routing Protocols for Multihop Wireless Ad Hoc Networks", IEEE JSAC, vol. 17, no. 8, Aug. 1999.

[5] H. Ehsan and Z. A. Uzmi, "Performance Comparison Of Ad Hoc Wireless Network Routing Protocols," IEEE INMIC, 2004.

[6] G. Lakshmikanth, M.A.Gaiwak and P.D. Vyavahare, "Simulation Based Comparative Performance Analysis of Adhoc Routing Protocols ," TENCON 2008 - 2008 IEEE Region 10 Conference ,Nov. 2008.

[7] D. Johnson, J.Broch, and D. Maltz., "DSR:The Dynamic Source Routing Protocol for Multi-Hop Wireless Adhoc Networks," http://www. Monarch.cs.rice.edu/monarch-papers/dsr-chapter00.pdf .

[8] K.U.R. Khan, R. U. Zaman and K.A Reddy, "Performance comparison of on demand and Table Driven Ad hoc Routing Protocols using NCTUns," IEEE Tenth international conference on computer modelling and simulation, April 2008.

[9] Shih-Lin Wu, Sze-Yao Ni, Yu-Chee Tseng , Jang-Ping Sheu, "Route Maintenance in a Wireless Mobile Ad hoc Network," IEEE HICSS, Jan. 2000.

[10] Zhaohua Long and Zheng He, "Optimization and Implementation of DSR Route Protocol based on Ad hoc network," IEEE WiCom ,Sept. 2007.