International Journal of Engineering Works
ISSN-p: 2521-2419
ISSN-e: 2409-2770
Vol. 6, Issue 12, PP. 547-555, December 2019
https://www.ijew.io/
https://doi.org/10.34259/ijew.19.612547555

Utilization of Unidirectional Links in AD-HOC Networks

Naeem khan[1], Dr. Majid Ashraf[2]
[1,2]UET Peshawar
eee103096@gmail.com[1]
Received: 16 December, Revised: 26 December, Accepted: 28 December

Abstract**—** In ad hoc network the unidirectional links and hidden node appear very frequently. There are few techniques that are used to avoid both unidirectional links and hidden nodes. Request to send/Clear to send (RTS/CTS) technique is used to avoid hidden link scenario and hello, blacklisting and reverse path search are used to avoid unidirectional links. In our research we opted for Dynamic source routing (DSR) which basically considers every route to be bidirectional, but as nodes moves frequently in ad hoc network these two problems occur. In the first part of the paper, we have implemented (RTS/CTS) and blacklisting techniques to avoid hidden links and unidirectional links to look into the improvement in the Dynamic source routing (DSR) by calculating certain parameters such as, Throughput (packet delivery at sink), End-to-End Delay, Network load and Packet delivery ratio. Furthermore our thesis also look into the link failure recovery, as nodes are continuously moving while data transferring as well so the node can move away from each other in these cases so the link broke down between the source and destination nodes so to avoid this scenario We implemented a mechanism of route recovery to efficiently tackle this problem. The result shows that the improved Dynamic source routing (Improved DSR) has shown more stability and performs very good overall in every performance parameter.

Keywords**—** Unidirectional links, hidden links, RTS/CTS, hello, blacklisting, reverse path search, Dynamic source routing (DSR).

I. INTRODUCTION

A network that does not have any base stations or centralized established infrastructure, a few nodes combines together to Form a network which is temporary pass information to each other for a specific purpose and they may be inter connected to each other through Wi-Fi links is called adhoc network. The links between the nodes are by default considered as bidirectional but due to certain reasons like nodes transceiver power difference, hurdles in signal propagation and sometime noise in the environment exists and these phenomena's creates unidirectional links between these nodes as shown in "fig.1".

These links can create problems in communication between the nodes if it is not tackled. Asymmetric (unidirectional) links are the main area of our research. Different researchers have different approaches to handle the unidirectional link [1] [2] [3] [4] [5] [6] [7] [8]. Using the unidirectional links is not so good, as it gives a high-overhead in the network [1]. So the question arises here to use unidirectional links or not. This is questionable and can vary from researcher to researcher. There is another problem to handle in ad hoc networks i.e. hidden link between nodes. The figure 2 shows hidden link scenario, due to hidden links the nodes cannot hear each other properly and this entirely jeopardizes the network. In the later part of our work we will look into the link failures while data transferring. As we know these all nodes are mobile in nature they frequently changes there position so it can move away from each other so that they lost their connection and the link in broken. This issue can also be detected due to interference in the environment.

Our research work is to efficiently look into the unidirectional links; avoid Hidden node cases and looking into link failure issues, thus ensuring the transfer of data packets over the network to avoid networks jeopardizing in DSR protocol.

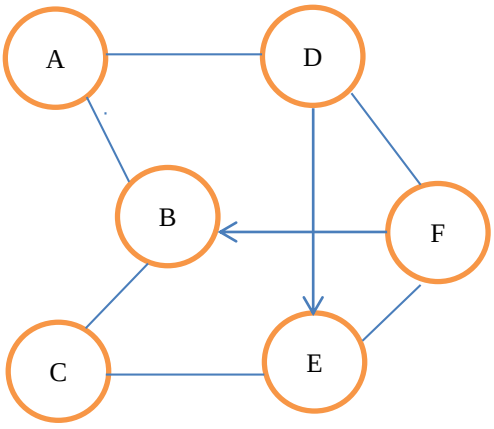

Figure 1. Bi-directional Adhoc network

II. DSR PROTOCOL

The Dynamic source Routing (DSR) is a special protocol that uses it resources when the route is needed for communication. A node is DSR sustain route in its caches memory or routing table of the other nodes. The cache tables of these nodes are updated when a node gets information about new routes of its neighbor. The protocol has two phase's i.e.

- Discovery of its routes
- Maintenance of the routes it has already found

When there is a desire of a single node that he wants to send some information to a node, it will look into its routing table that if he had its address or not, if he had an address of a route to this specific node it will connect to this node by send RREQ directly through that route and when he gets RREP from that node it will start sending information. But if there he found no route in its memory it will start a mechanism by flooding the RREQ to its neighbors and if there is no destination node in the neighbor, they will send it to their neighbors and this process is continued until the destination node is found. When a node want to send data, first it sends RREQ packet in which it adds the address of the node and its own address for RREP and it also adds a very unique number through which all the nodes can identify that this packet has been sent by source which source node. When any of the nodes receives this packet it looks into its own table that if he knows the path, if he do not he floods the RREQ packet [9][12]. RREQ is received by a node; the address it holds of a destination is its own or of other node but it knows its route so this node will generate a RREP packet. The node will append its own information and the information from the RREQ packet into this and will send it through a route on which it has received the RREQ packet as shown in "fig. 4" DSR protocol uses two methods to maintain its route i.e.

- Route error packet
- Acknowledgement

When node finds error in its transmission it generates the Route error packet (RERR) and sends it to other nodes. When the other nodes receive this specific packet it initiates a process and removes the route of the node from its route cache. The acknowledgment packet is used in DSR to find that the Packet has been successfully received at the specific node or neighbor. There is also a passive acknowledgment by which the node knows that the Packet has been forwarded to other nodes.

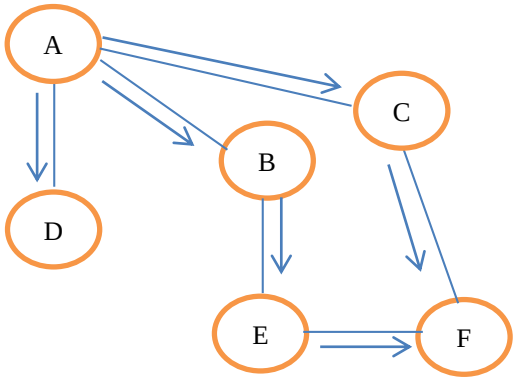

Figure 2: RREQ packet sent from A to F

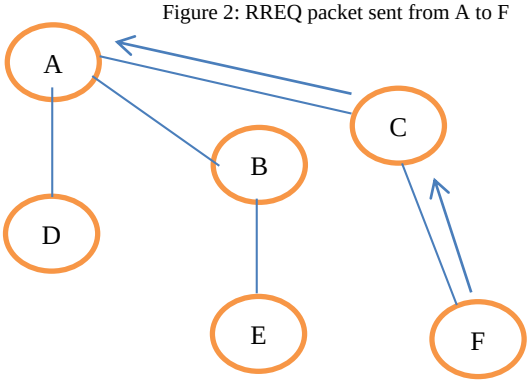

Figure 3: RREP packet sent from F to C to A

III. TECHNIQUE FOR HANDLING UNIDIRECTIONAL LINK, HIDDEN LINK AND LINK FAILURE

The node in discussion has to send data to some other node called destination, so it will send RREQ to all neighbor nodes, when the RREQ is received at neighbor nodes it will reply with Acknowledge Packet (ACK) back to source node, if the source node does not receive Acknowledgement (ACK) from any of the neighbor node it will have to check two scenarios under which the concern node did not reply.

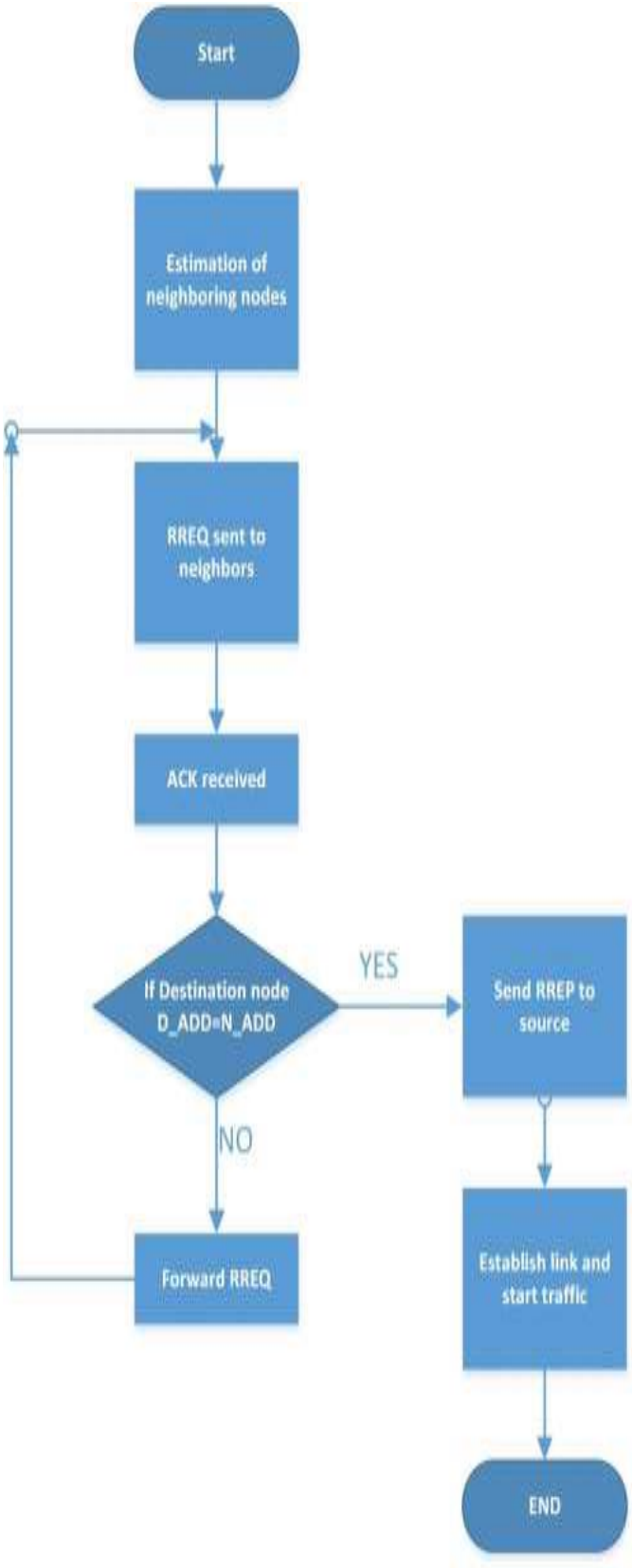

Figure 4. DSR protocol Process

A. Hidden node scenario

Another problem researcher's encounter while deploying adhoc network in a certain area is hidden nodes. This is a problem that occurs when a node A want to communicate with other node AP (access point) and at the same time some other node B is also trying to communicate with the same node (access point) as seen in "fig .5". If one of the nodes is trying to send a large number of data packets so we can encounter a lot of packet drops as neither of the packet will pass as both A and B are sending data packets. To handle this problem we can use different solutions.

1. RTS/CTS (Request to send/Clear to send). Node will send RTS to AP and when AP responds with CTS it will send its data.
2. By enhancing antenna range so the node will now not be a hidden node as we are using CSMA/CA, in which every node will wait for its due turn to send data.
3. We can also encounter this problem due to some obstacle so first we have to remove the obstacle is some cases or in other cases where we cannot move obstacle we will move the node location.

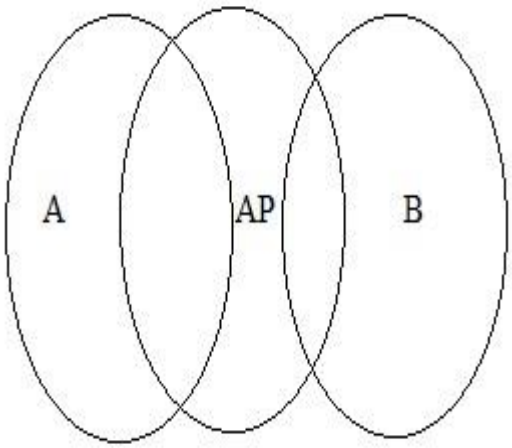

Figure 5. Hidden node

B. Unidirectional link scenario

In the second place if the node is not hidden node then the source node will check for the unidirectional link, for that, it will initiate a counter and will wait for Maximum Upper threshold each time it sends the RREQ to the specific node, when the counter reaches the count three and the node did reply with RREP packet so the node will go into its natural process. If the node did not reply with RREP packet the node will perform three processes there as shown in figure 4.

a) Blacklist the node
b) Update its routing table
c) Inform the concern nodes that this link has been blacklisted

If the Hello Packet (RREQ) is Acknowledge by the neighbor and it will compare its Address with the Destination Address send by source node, if the node Address matches with Destination Address it will send RREP packet and the source node will establish the link and will start traffic between them. If the neighbor node is not the destination node so it will send the RREQ Packet to its neighbor to find out the path to the destination node and the same process will be performed again and again till the RREQ Packet has reached the Destination node as shown in "fig. 6"[11][12]. Once the link is established between the source node and destination node there we can enconter link failure.

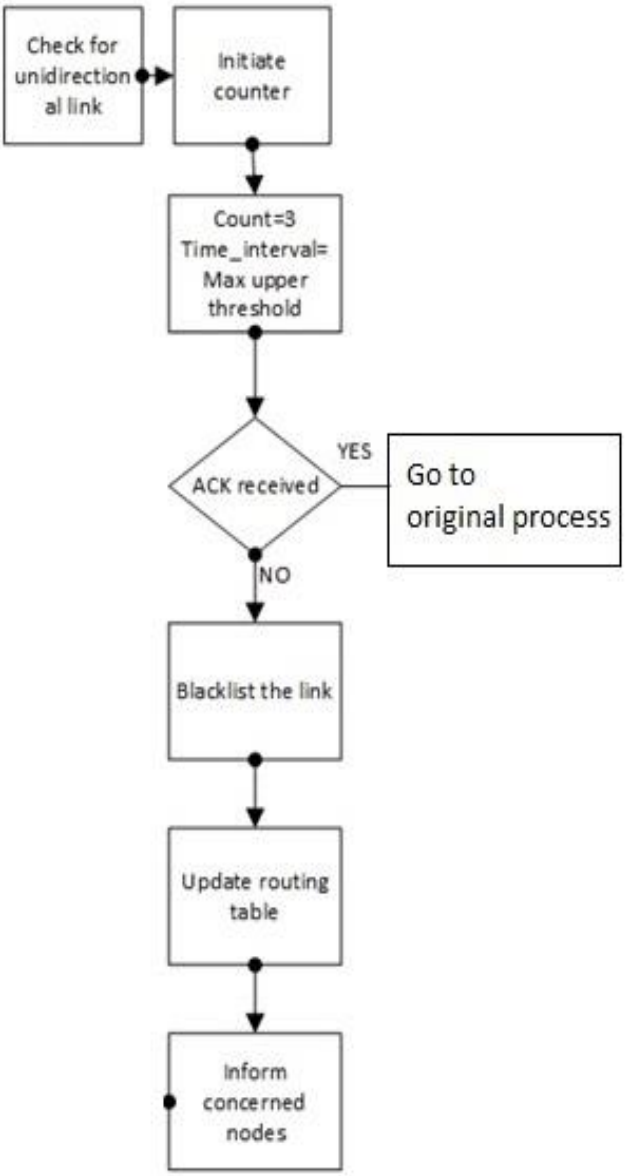

Figure 6. Black listing process

C. Link failure

When the source node and destination node transfer data there may be a link failure it can be due to many reasons.

I. Destination or Source node moves away from the range of each other.
II. Due to any interference between the Source and Destination nodes.
III. Due to frequencies variation of Source and Destination nodes.

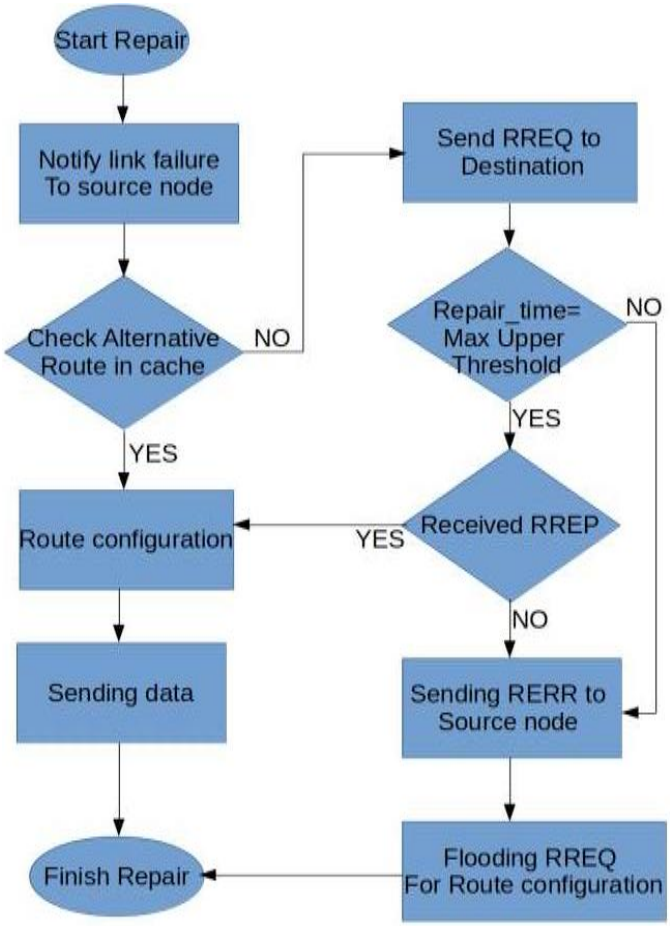

Figure 7. Link failure process

First the Source node will check the alternate route in its routing table to reach destination, if the route is found then the Source node will configure the route and after establishing the route it will start the traffic once again. But if the node don't have any route in its routing table, it will resend a RREQ packet towards the Destination node, it will wait till the Maximum upper threshold time, if the RREP is received so it will configure the route and establish the link and will start traffic but if the RREP is not received the at source the RERR packet will be sent to Source node and it will again start the main process by sending a RREQ packet to the neighbors as shown in "fig. 7".

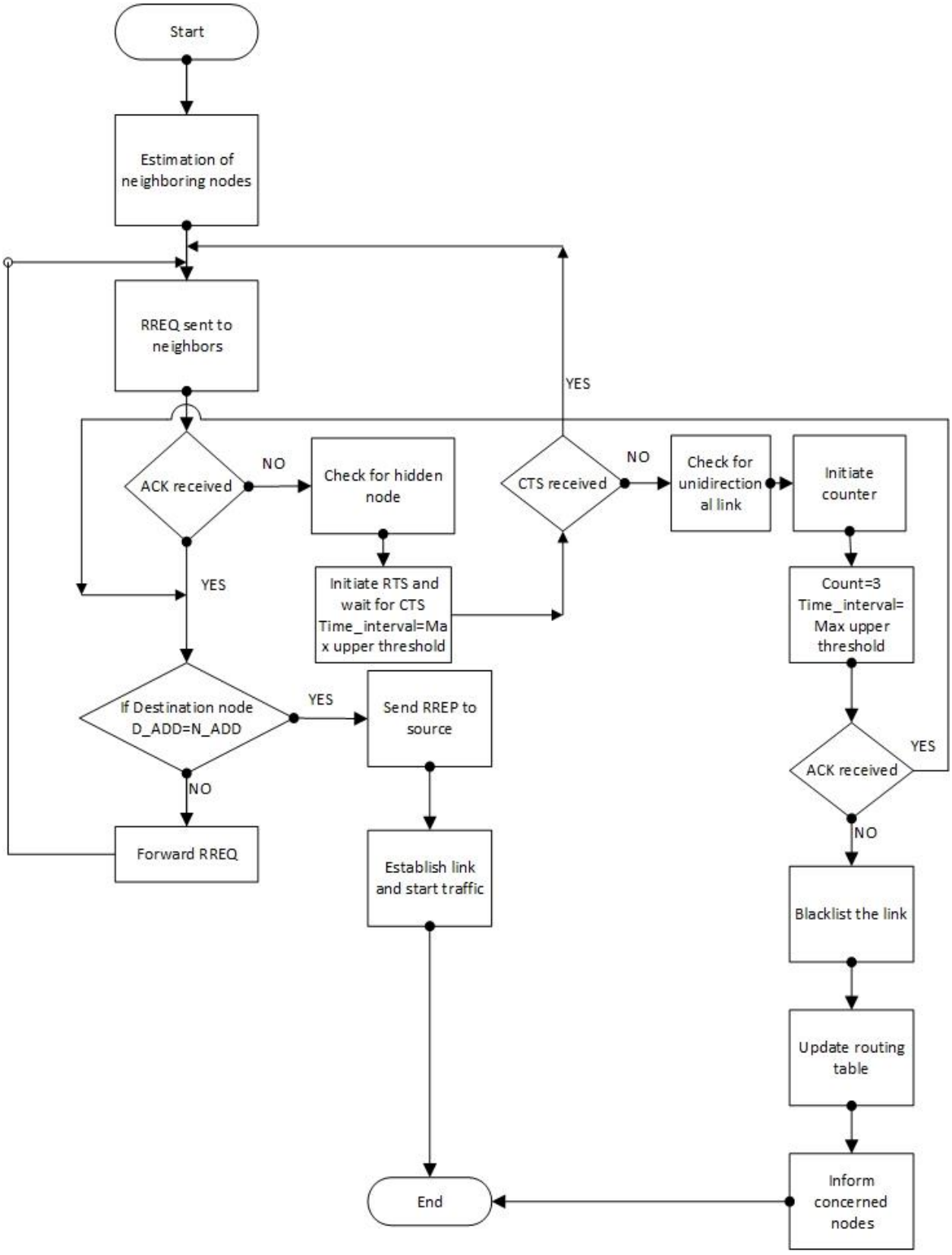

Figure 8. Flow chart of the whole process, Hidden node, unidirectional link and link failure

IV. SIMULATION RESULTS

In this chapter we will see into the simulations we had performed and evaluate the results. We had found during our research under the DSR protocol utilizing unidirectional links in adhoc networks. We are evaluating our results through four different parameters (1) Throughput (2) End-to-End delay (3) Network load (4) Packet Delivery ratio.

We will look into two sets of experiments i.e. (1) Comparison of DSR vs. DSR with unidirectional link and hidden nodes (2) Comparison of DSR with unidirectional Link and hidden nodes vs. DSR with unidirectional Link and hidden nodes including Link Failure recovery (Imp DSR with Link failure).

A. Comparison of DSR vs. DSR with unidirectional link and hidden nodes(Imp DSR)

We have used matlab as a simulation environment for our research purpose. We created a 100 nodes environment to find out the results. The result will be shown in two different scenarios mentioned above and there performance metrics. First we implemented the improved DSR protocol by including unidirectional link and hidden nodes scenarios. Then we compare their graphical response and we find out that throughput of the packets has shown improvement as shown in table. All other parameters such as End-to-End delay, Network load and Packet Delivery ratio has shown huge improvements it can be seen in the table and from the graphs.

Table 1: Throughput, End-to-End delay and Network load of DSR and Improved DSR

Rounds of data Packet sent	**Throughput of DSR**	**Throughput of Imp DSR**
1000	25%	75%
5000	3%	15%
	End-to-End Delay of DSR	**End-to-End Delay of Imp DSR**
1000	6ms	1ms
5000	9ms	3ms
	Network Load of DSR	**Network Load of Imp DSR**
1000	70%	10%
5000	90%	80%

Table 2: Packet Delivery ratio of DSR

Rounds of data Packet sent	**Packets received at destination (DSR)**	**Packets Drop (DSR)**
80000	43000 Approximately	37000 Approximately

Table 3: Packet Delivery ratio of Improved DSR

Rounds of data Packet sent	**Packets received at destination Imp DSR**	**Packets Drop Imp DSR**
80000	49000 Approximately	31000 Approximately

From the graphical response we can see that throughput has been incresed with avoidance of unidirectional links and hidden nodes adjustments. As here we are avoiding unnecessary data packets send on unidirectional links and then waiting for the response fro the specific node we are ncreasing our throughput ability and decresing the End-to-End Delay. Throughput in ths scenario has been achieved upto 12 to 10 percent more than simple DSR. The End-to-End Dealy has been minimsed upto 5 to 6 ms as shown in "fig10". We can aslo see that the load on the network has been reduced upto 10 percent from DSR protocol. The packet Delivery ratio is another parameter that has been enhanced due to our proposed solutions, improved DSR has shown a increased upto 8 percent as shown in "fig .12".

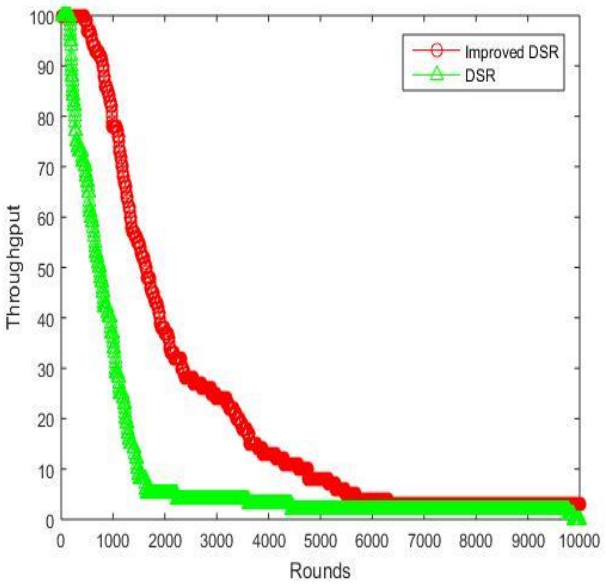

Figure 9: Throuput of DSR and Imp DSR

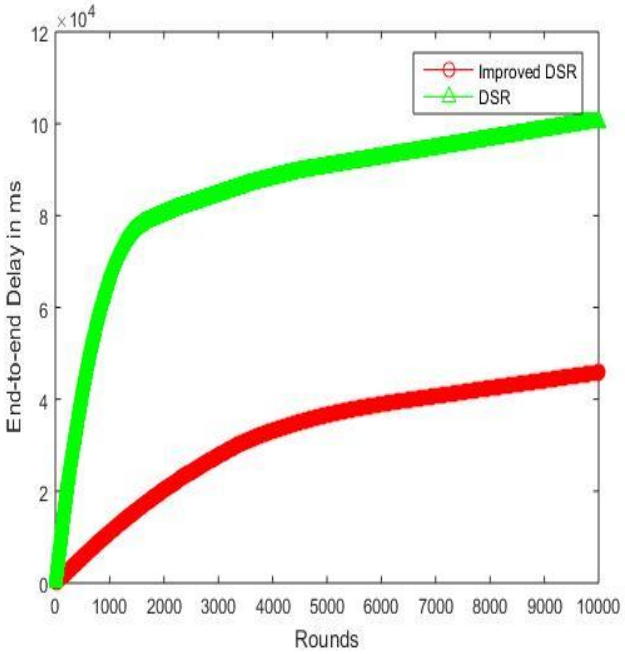

Figure 10: End-to-End Delay of DSR and Imp DSR

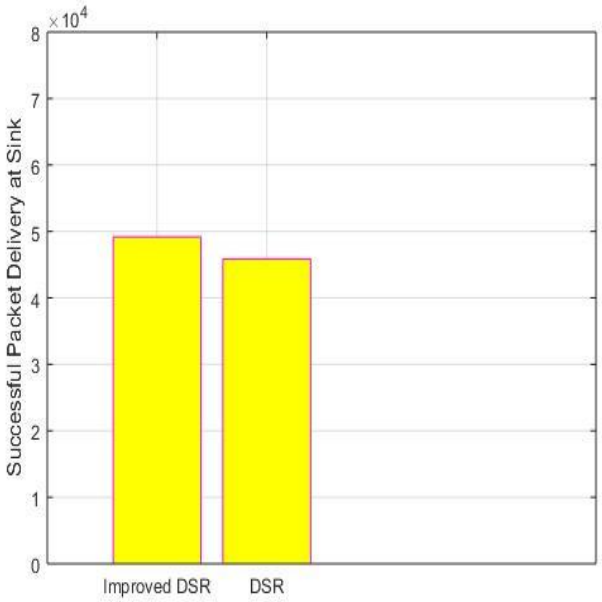

Figure 12:Packet Delivery Ratio of DSR and Imp DSR

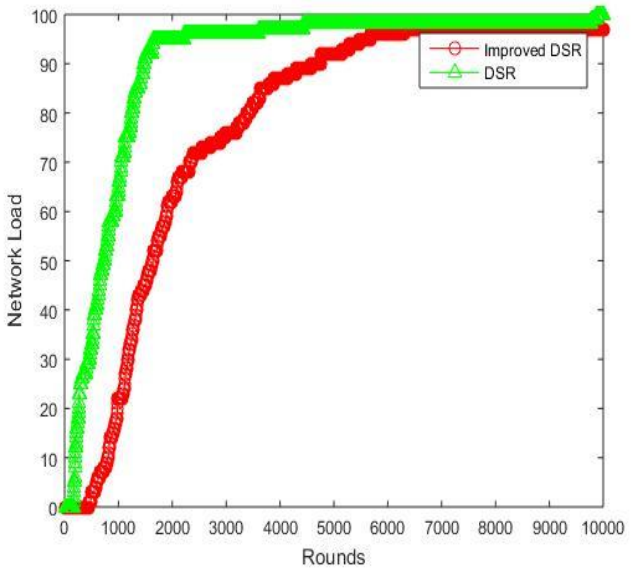

Figure 11: Network Load of DSR and Imp DSR

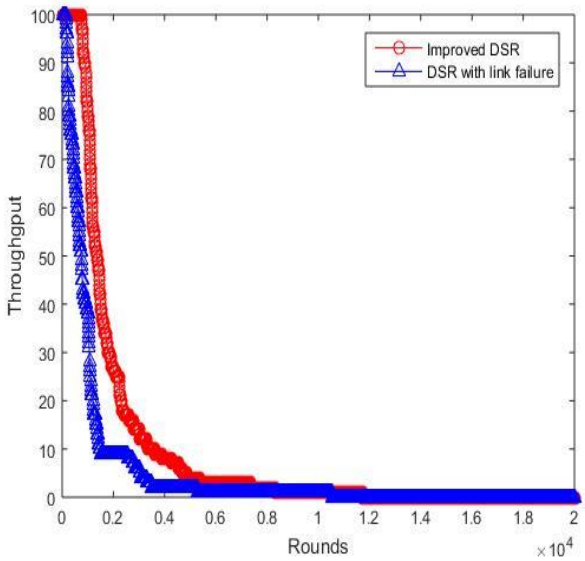

Figure 13: Throughput of Imp DSR and Imp DSR with link failure

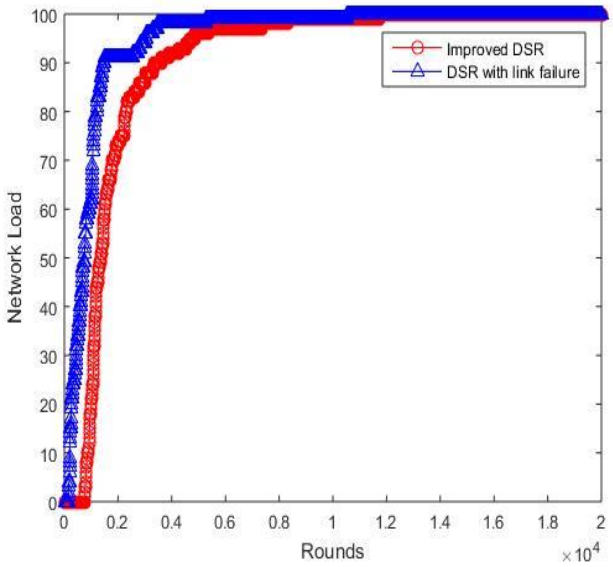

Figure 14: Network load of Imp DSR and Imp DSR with link failure

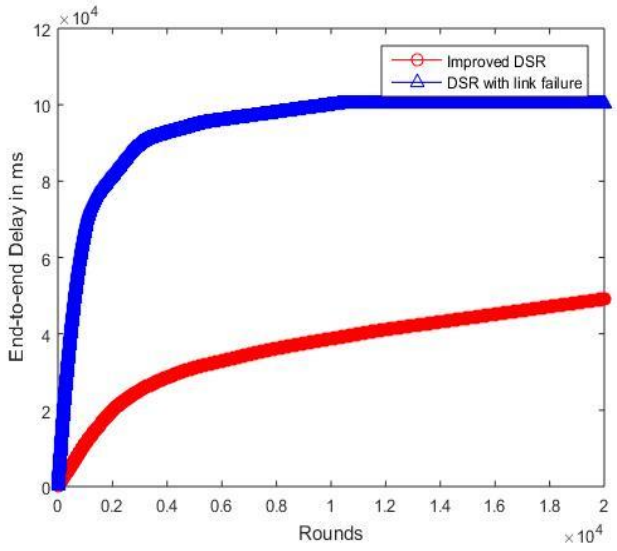

Figure 15: End-to-End Delay of Imp DSR and Imp DSR with link failure

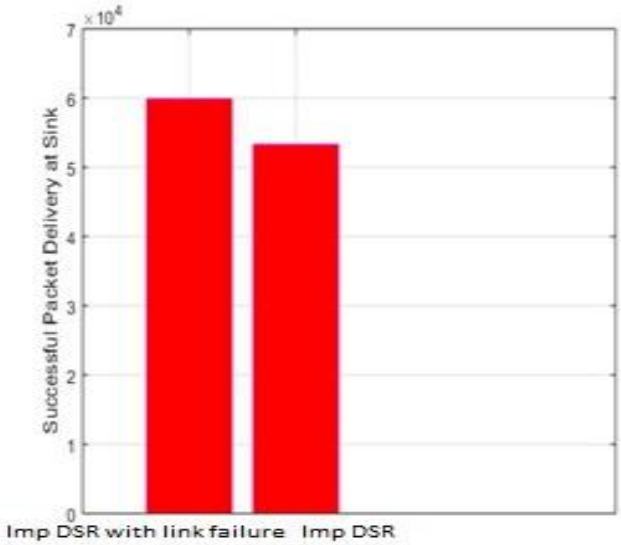

Figure 16: Packet Delivery ratio of Imp DSR and Imp DSR with link failure

Table 4: Throughput, End-to-End delay and Network load of Imp DSR and Imp DSR with link failure

Rounds of data sent	**Throughput of Imp DSR**	**Throughput of Imp DSR with link failure**
1000	75%	90%
4000	10%	15%
	End-to-End Delay of Imp DSR	**End-to-End Delay of Imp DSR with Link failure**
1000	1ms	3ms
4000	3ms	8ms
	Network Load of Imp DSR	**Network Load of Imp DSR with Link failure**
1000	10%	5%
4000	75%	70%

Table 5: Packet Delivery ratio of Imp DSR

Rounds of data sent	**Packets received at destination (Imp DSR)**	**Packets Drop (Imp DSR)**
70000	52000 Approximately	18000 Approximately

Table 6:. Packet Delivery ratio of Imp DSR With link failure

Rounds of data sent	**Packets received at destination (Imp DSR with link failure)**	**Packets Drop (Imp DSR with link failure)**
70000	61000 Approximately	8000 Approximately

B. *Comparison of DSR with unidirectional Link and hidden nodes (Imp DSR) vs. DSR with unidirectional Link and hidden nodes including Link Failure recovery (Imp DSR with Link failure).*

In the second part we established a link failure recovery when a link is gone down while transmitting data between the source and Destination node as shown in "fig. 8". That gives a clear idea of how we will recover from a link failure by finding a shortest alternate path for data delivery, if we didn't found any path in cache; we will send RREQ and look into if the link can established, if it did not established we will send a RERR and will start the process again. In the above tables we have shown the Comparison of DSR with unidirectional Link and hidden nodes (Imp DSR) vs. DSR with unidirectional Link and hidden nodes including Link Failure recovery (Imp DSR with Link failure).

From he graphical response the improvement can be seen up to 13% in Imp DSR with link failure. The total improvement in packet delivery ratio from DSR is 23% from the Imp DSR. In total which accumaletly gives us a value of 30 % from DSR protocol. The improvement is due to the alternate paths we find for data transfer and we don not wait for a long a time and we do not drop packets as our node is not sending any packets if the the link is broken.

C. *Trade off*

When we implemented the link failure recovery, so our End-to-End delay has been increased up to 4 to 5ms from the improved DSR that can be seen in Table 5 and "fig.15". Due to looking into route cache for new shortest routes and may be restarting the process if we did not found the possible route, so this in return will increased our delay time.

CONCLUSION

In our thesis we are trying to enhance the capabilities of DSR routing protocol by implementing two main changes into it.

(1) DSR with hidden link and unidirectional links (improved DSR).

(2) Improved DSR with link failure recovery.

We are using the blacklisting technique to avoid the unidirectional links and RTS/CTS technique to avoid hidden links. Now the second problem was of link failure and its recovery. We used a simple technique of looking into any availale shortest path for communication is a specific time interval or else restart the whole process again after Max upper threshold time.

Our research of improving DSR routing protocol by the above mentioned techniques had shown a very good improvement in every performance parameter. It gives multiple advantages such as:

(1) Immunity from unidirectional links

(2) Avoiding hidden nodes

(3) Link failure recovery.

But we also have to face some tradeoffs in imp protocol with link failure regarding End-to- End delay.

REFERENCES

[1] Samir R. Das Mahesh K. Marina, "Routing Performance in the Presence of Unidirectional Links in Multihop Wireless Networks," Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing, pp. 12-23, 2002.

[2] Zygmunt J. Haas, Benjamin P. Manvell Marc R. Pearlman, "Using Multi-Hop Acknowledgements to Discover and Reliably Communicate over Unidirectional Links in Ad Hoc Networks," in IEEE Wireless Communications and Networking Conference, 2000.

[3] RAVI PRAKASH, "A Routing Algorithm forWireless Ad Hoc Networks with Unidirectional Links," Wireless Networks , vol. 7, no. 6, pp. 617-625, 2001.

[4] Daniel Mossé Venugopalan Ramasubramanian, "BRA: A Bidirectional Routing Abstraction for Asymmetric Mobile Ad Hoc Networks," IEEE/ACM TRANSACTIONS ON NETWORKING, pp. 116-129, 2008.

[5] SHIOW-FEN HWANG, CHYI-REN DOW YI-YU SU, "An Efficient Cluster-Based Routing Algorithm in Ad Hoc Networks with Unidirectional Links," JOURNAL OF INFORMATION SCIENCE AND ENGINEERING, pp. 1409-1428, 2008.

[6] Jian-de Lu, Jia-jia Tang Zhen-zhong Wang, "Neighbor Monitoring Mechanism to Solve Unidirectional Link Problem in MANET," in Wireless and Mobile Communications, International Conference, 2007, pp. 55-59.

[7] LIU Yuan-an, LIU Kai-ming, ZHAI Lin-bo, YANG Ming ZHUANG Lin, "An adaptive algorithm for connecting mobile ad hoc network to Internet with unidirectional links supported," The Journal of China Universities of Posts and Telecommunications, pp. 44-49, 2010.

[8] Sung-Ju Lee, Jun-Beom Lee Young-Bae Ko, "Ad Hoc Routing with Early Unidirectionality Detection and Avoidance," in IFIP International Conference on Personal Wireless Communications, 2004, pp. 132-146.

[9] Y. Hu, D. Maltz D. Johnson, "Request for Comments: 4728 (The Dynamic Source Routing Protocol (DSR))," Rice University, Microsoft Research, Experimental 2007.

[10] Jorg Nolte Reinhardt Karnapke, "Unidirectional Link Counter - A Routing Protocol for Wireless Sensor Networks with Many Unidirectional Links," in 14th Annual Mediterranean Ad Hoc Networking Workshop , 2015.

[11] Juan-Antonio Cordero , Jiazi Yi , Yuichi Igarashi Thomas Clausen, "Use 'em or lose 'em: On unidirectional links in reactive routing protocols," Elsevier Science Publishers, pp. 51-64, 2018.

[12] Y.Hui Y.Fengjie, "Research on DSDV routing protocol based on wireless mesh network," in Chinese control and decision conference, 2018.