INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN 2278 – 0211 (Online)

Performance Comparison of AODV and B-AODV Routing Protocols

S.I.S. Jaffarvalli
M.Tech Student, AITS, Rajampet

Abstract:
Manets are self configuring networks. Why manets are self configuring networks? It's because they possess they dynamic topology. it is difficult to predict the topology of manets .It presents an optimized protocol: B-AODV, based on the shortage of routing finding and routing repair of AODV. In B-AODV, first through reverse request by sending BRREQ replace of RREP, it reduces the time of routing finding. Second, two hops IP record in control messages and route table can improve the rate of routing repair and reduces the times of routing findings. And it improves the function of Ad Hoc network. This simulation experiment is based on NS2 and compared the performance of AODV and B-AODV. And it compared the differences of control packets, ratio of packets, end to end delay for AODV and BAODV. It shows that B-AODV is better than AODV.

Key words*: Ad Hoc Network, on-demand Routing, AODV*

1. AODV protocol

1.1. Introduction of AODV protocol
AODV (Ad hoc on-demand distance vector routing) is a Source drive type[3] routing protocol. When a source node sent message to a target node without the routing, it sent RREQ first. When the adjacent node received RREQ with the addresses of source node and target node, it judged if it was same with the target node's address. If it was, sent RREP to source node, otherwise, checking the routings in the rout table that could reach the target node, then send RREP to source node, or continue to flooding send RREQ. AODV protocol can maintain routing nodes through broadcasting hello message regularly. If one link break, it sent ERROR message to nodes, meanwhile deleted broken records or repaired the routing.

1.2. The shortage of AODV [4]
AODV protocol created the routing between two nodes in the network based on route discovery and route maintenance.. In the route require process, it broadcasted RREQ message to target node in the form of flooding; In the route response process, target node preferentially chose the first-arrived RREP and sent RREP message. As the node strenuous exercise during finding routings, the route was easy to break, which would lead to the loss of RREP packet. In the routing maintaining process, the broken node would discard error packet and notice the source node to resend request message when broadcasting RERR packets, which would lead time delay of subsequent packet; in order to eliminate loopback caused by new touring, local repair may cause that the RREP sent by downstream node could be discarded, which reduced the probability of routing recovery. In addition, despite with routing maintenance mechanism, the node is mobile, the route may not be repaired timely or failed to repair, which reduced the utilization of network control information, and could increase the routing delay, thus affecting the performance of the network..

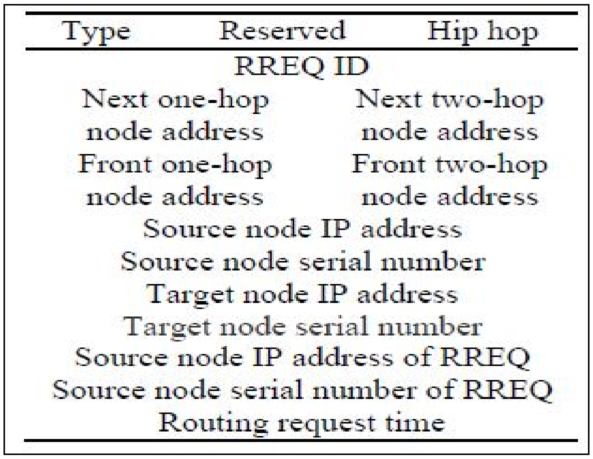
2. Design of B-AODV protocol
For the shortage of AOFV protocol, a better protocol is needed to avoid route breaking, reduce message lose rate and decrease network delay. An improved approach B-AODV based on AODV was designed, assuming that the network link was bidirectional, that was the source node and the destination node could reach each other through one route .

2.1. Routing construct
For AODV, if a node moved fast, in the reverse routing it may cause the loss of packets when RREP sent reply message through unicast, and RREP unicast back to the source node, so in the B-AODV, a control message B-RREQ similar to RREQ was devised when the reverse route was built. The source node also undertook the task of routing discovery as the target node, and after receiving RREQ it reversely sent flooding B-RREQ message to find the source node,. When the source node got first B-RREQ, it started to

transmit data, and recorded the information of both front and next two-hop nodes in RREQ and B-RREQ for rapidly rebuilding routings.

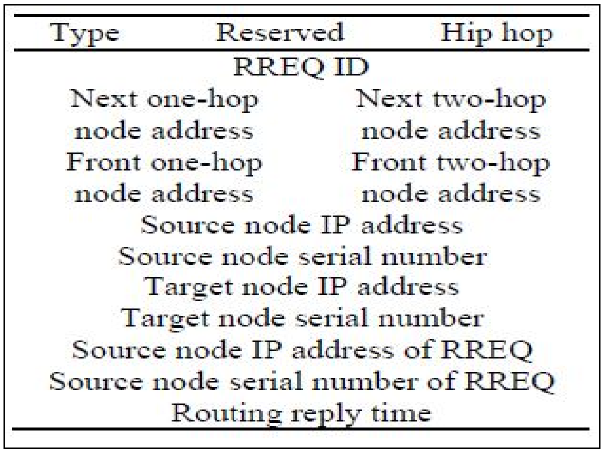
2.1.1. The route discovery process was detailed as following

When the source node had no routing to the destination node and had to send data to the destination node, it broadcast RREQ message to neighbors. The RREQ message’s format was shown in table 1.This format added front and next two-hop nodes addresses, that showed RREQ recorded not only the addresses of the current node's neighbours but also that of two-hop nodes. When one node broadcast RREQ control message, the counter of RREQ added 1 automatically. The source node IP address and RREQ ID could uniquely identify a RREQ control message. A node checked ID when receiving same RREQ control message, if that were the same, one of that was discard. This process was completely same with AODV. After getting RREQ, the intermediate node compared with its own routing information, and updated the related information in the routing table, thus relevant bytes were added: front and next two-hop nodes' addresses. The format of routing table is shown in table 2. If the intermediate node had no routing to target node, it continued to sending RREQ and constructed reverse routing. When RREQ packet reached the target node, the target node created a packet B-RREQ similar to RREQ. Table 3 showed the B-RREQ message format.



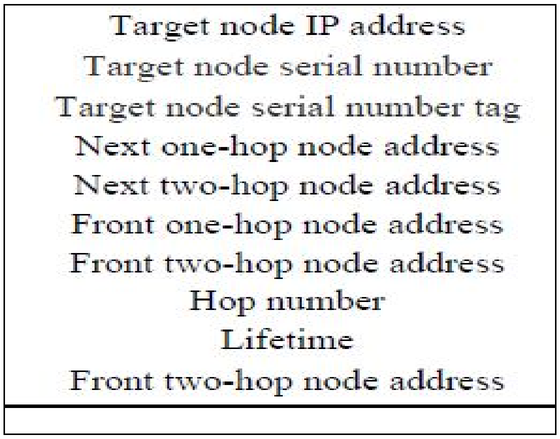
Type	Reserved	Hip hop
RREQ ID		
Next one-hop node address		Next two-hop node address
Front one-hop node address		Front two-hop node address
Source node IP address		
Source node serial number		
Target node IP address		
Target node serial number		
Source node IP address of RREQ		
Source node serial number of RREQ		
Routing request time		

Table 1: message format of RREQ



Type	Reserved	Hip hop
RREQ ID		
Next one-hop node address		Next two-hop node address
Front one-hop node address		Front two-hop node address
Source node IP address		
Source node serial number		
Target node IP address		
Target node serial number		
Source node IP address of RREQ		
Source node serial number of RREQ		
Routing reply time		

Table 2: new routing table



Target node IP address
Target node serial number
Target node serial number tag
Next one-hop node address
Next two-hop node address
Front one-hop node address
Front two-hop node address
Hop number
Lifetime
Front two-hop node address

Table 3: B-REP message format

Having created B-RREQ, the target node broadcast it to neighbor nodes immediately, same with the process that the source broadcast RREQ, which is to build reverse routing. When the intermediate node received the packet, it would check there was the same packet or not, if it was yes, then the same packet would be discarded. Otherwise it would be sent to neighbor nodes. The neighbor node received the packet, recorded the addresses of next and front two-hop nodes, and set the routing lifetime until it got the target node. The source node transmitted data immediately when receiving first B-RREQ. The routing reply process was shown in figure 1.

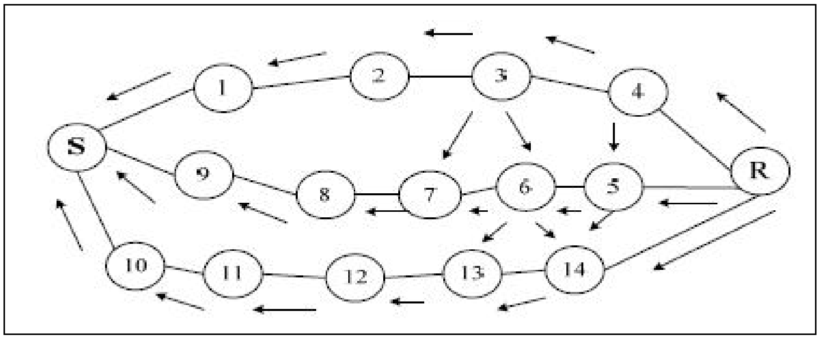


Figure 1: send B-REQ

Unlike that RREP returned to the source node with one-hop and unicast, as R->4->3->2->1->S, in BAODV, B-RREQ was broadcast with many routings to choose,
R->5->6->7->8->9->S,R->14->13->12->11->10->S and so on. The source node S could transmit data in a Routing with most energy. The problem that RREP often lost as the intermediate nodes moved in AODV can be solved.

2.2. Control routing message computation

In B-AODV, it seems that routing overhead was added for sending B-RREQ, however, it was reduced indeed by proving. A network environment was set here: There were N nodes in the Ad Hoc network; In the AODV protocol, the number of transmitting control message during routing discovery was AONV(N); In the B-AODV protocol, the number of transmitting control message during routing discovery was BAONV(N);Assuming that there were m nodes transmitting message during routing discovery, as the AODV protocol says, if the first routing discovery was right, the routing node number of transmitting control message is:

AODV(m)=m-1+t (1)

Where, t is the number of nodes transmitting routing reply message.

If there was more than 1 time during routing discovery in AODV, then it should be

AODV(m)=c(m-1+t) (2)

Where c meant the number of routing discovery process.

If data transmitted following B-AODV, there was at least 1 stable routing to be found to transmit data during routing discovery, so only 2m-2 nodes were deeded to transmit data:

B-AODV(m)=o(2m-2) (3)

From this we can draw the conclusion: when c>1, AODV protocol produces much more Routing over headin the routing building process.

2.3. Routing maintenance

Shown in figure 3, in the AODV algorithm, if the link between nodes 2 and 3 had been broken, node 2 need to repair local routing, send RREQ, and find routing to the target node R. RREQ was forwarded to node4 through node F, until reaching node R, then R replied RREP back to node 2 along the reverse routing. Thus a new effective routing was built. If routing repair failed within certain repair time, it sent RRER message to source node, and the source node resent RREQ to find routing. In the new algorithm, each routing table maintained the addresses of adjacent two-hop nodes, when the next node 3 in the routing table that it would arrive to target node lost effectiveness, meaning that the routing was invalid and it need local repairing. Different from the AODV, node 2 didn't wait for the RREQ message of target node R, as next two-hop nodes' addresses were initially recorded in the routing table, node 2 looked for the next two-hop node 4, then 2 sent RREQ message to node 4. When RREQ reached node 4 through F, mean while node 4 owned active path to target node, so node 4 could send RREP directly to node F, then node2. Thus a valid routing from node 2 to node R was built, as shown in figure 2.

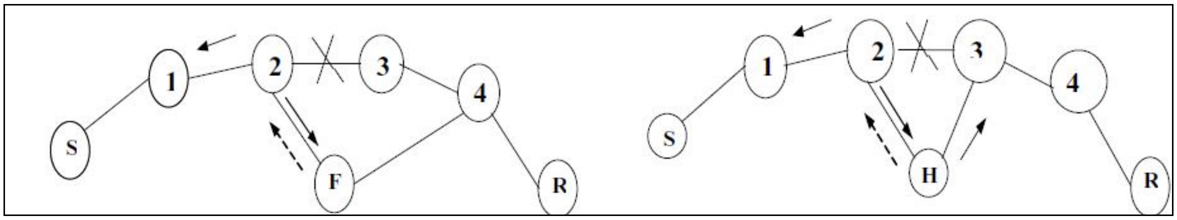


Figure 2: routing repair :find 2 and 4, build new routing
Figure 3: routing repair :build 2 and 3

If there was no common neighbor node between node 2 and 3, shown in figure 4, node 3 could not escape next-hop range of node 2. So node 3 was likely to be one of node 2's next two-hop nodes. In this case, after receiving RREQ from node 2, node H found that there was no node 4 but node 3 in its neighbors, then node H sent RREP to node 2, aiming to notice that it was the common neighbor node with node 3, thus node 2 would rebuild a new routing through node H and 3 to the target node. If the node 3 and the node 4 lost efficiency, shown in figure 5, node 2 sent RREQ finding the target node R,

to find new routing of broken path; if routing discovery failed, node 2 sent RRER to neighbor nodes untilback to source node S, S resent RREQ to start a new round of routing discovery, which was same withAODV protocol. Due to the movement speed of nodes was limited, there will be less in reality. This reduced the cost of routing maintenance as well as routing delays. If the case was special, node 4 was the target node, completely same with original AODV algorithm, it sent RREQ to find the next two-hop nodes, i.e. the target node. When node 3 being the target node, send RREQ do find the target node directly. Above that, the downstream neighbor nodes of the fracture was more closer than the target node to the upstream nodes of the fracture, so the finding speed would be faster, that was to say, the finding time was shorter than AODV routing algorithm, so the new protocol is more optimized.

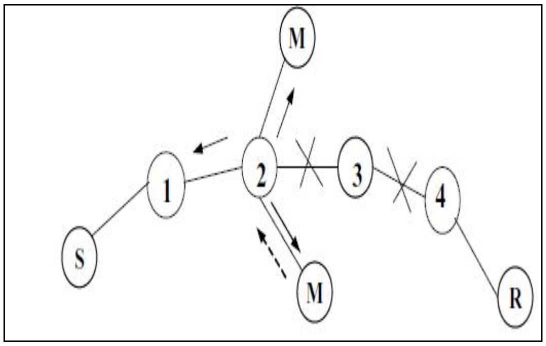


Figure 4: routing repair failure the next two jump node is invalid

3. Experimental Analysis

The performance of the routing protocol depended on[6]: the Routing overhead, the ability to send packets, the delay from node to node and the packet posted success rate, etc. Moreover different scenes had a big impact to the routing protocol. Same routing protocols in different simulated scenarios, its routing performance also had a great deal of difference. The main factors of the ad hoc network scenes: the number of network nodes, the changing speed and the movement speed of the network. Therefore, when comparing and analyzing different routing protocols, the network scene must be taken into account .The simulink environment here was: windows xp+fedor core4+NS2. 29.

3.1. Performance indicators of routing protocol

Performance indicators of routing protocol were as follows [7] [8[9]:

(1)Successful packets posted rate

Successful packets posted rate =

(Number of sent packets – number of discarded packets) / number of sent packets (8)

(2)Average delay from node to node

$$\boxed{\frac{1}{N}\sum_{i=1}^{N} rt_i - st_i}$$

Where N represented the number of successfully-transmit packets, rt was the time of packets reaching the target node, was the time of packets being sent.

(3)Routing overhead

Routing overhead can compare adaptation capacity and efficiency of different routing protocols. The formula was:

Routing overhead = total number of routing control packets

3.2. Results analysis

Through the simulation of AODV and B-AODV, compare their performance in the number of nodes and maximum speed of node movement.

(1)Packet delivery ratio

a. Packet delivery ratio of different nodes

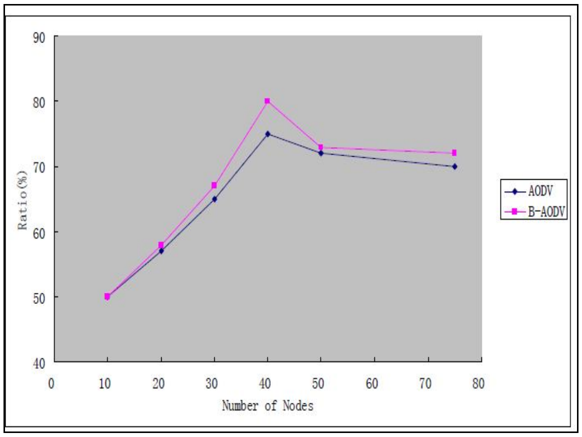


Figure 5: packet delivery ratio of different nodes

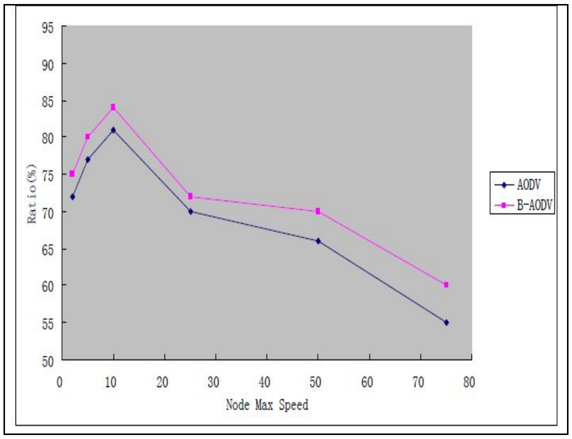
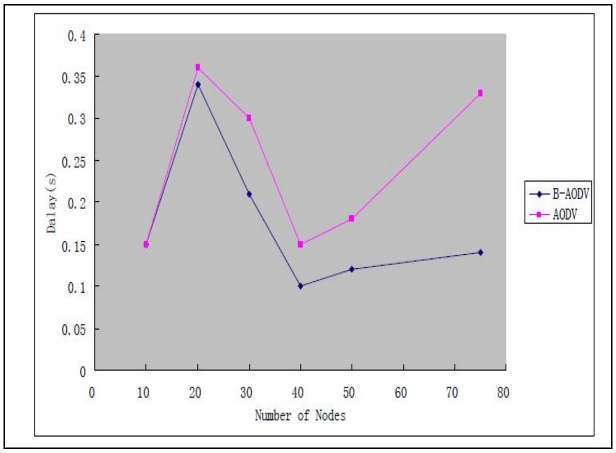


Figure 6: packet delivery ratio at different speeds

From figure 5, it can be seen that the packet delivery ratio of the improved protocol was almost same with that of the original protocol when there were less nodes, for which the probability of routing repair was low after the routing broke. When there were much more nodes, the delivery ratio cut down obviously, for which with more nodes, the nodes moved frequently thus the probability of routing breaking was higher.



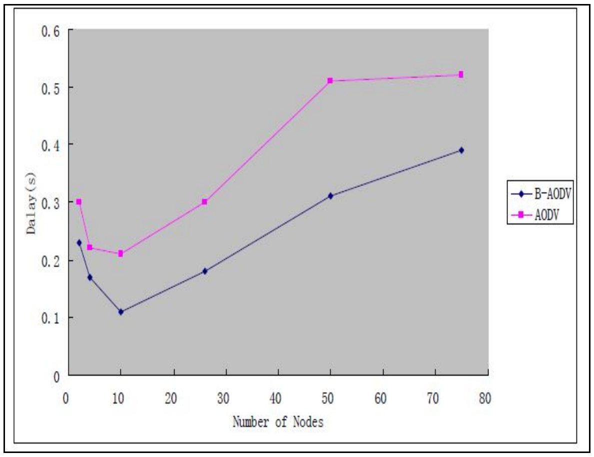


Figure 7: end to end delay of different nodes *Figure 8: end to end delay of different nodes*

The delay of B-AODV was lower. Because for the original protocol the RREP reply must be limited

during routing finding, on the other hand, when the routing broke, AODV must send RREQ to the target node then rebuild routing through returning RREP, however for B-AODV, it only sent RREP to the front two-hopnodes then built forward routing, so the delay would be lower.

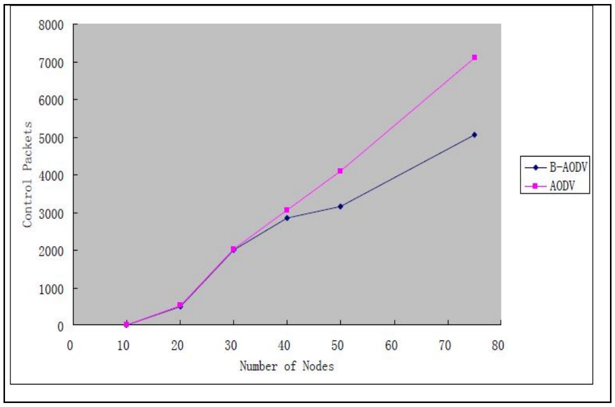


Figure 9: routing overhead

From figure 9, when there were less nodes, the numbers of two control packets were similar, but when the number of nodes added, B-AODV protocol was clearly optimized than AODV protocol.

4. Conclusion

It can be clearly observed that the key to designing Ad Hoc routing protocol is to solve the problem that fast node movement makes complex changes in network structure. On the basis of the comparing and analyzing AODV routing protocol, an improved B-AODV protocol, which ameliorated the routing finding and local repair for rebuilding new routing. The new protocol improves the routing repair capacity and packet successfully-posted ratio, reduces the delay from node to node, and decreases the routing overload. Nonetheless, the nodes in the network do not only have a two-way link, the single link protocol optimization is the goal of future research.

5. References

1. Y.Kim,J.Jung,S.Lee C.Kim,A Belt-Zone Method for Decreasing Control Messages in Ad Hoc Netowkors.ICCSA.2006:42-46.
2. Ankit Sggarwal, Bhumika Garg. Survey on secure AODV for Ad Hoc networks routing mechanism. JInternational Journal of Advanced Research in Computer Science and Software Engineering, 2012; 2(3):203-206.
3. L. Abusalah, A. Khokhar, and M. Guizani, A survey of secure mobile ad hoc routing protocols,. Commun.Surveys Tuts., 2008;10(4):78–93.
4. Marchang Ningrinla. Light-weight trust-based routing protocol for mobile ad hoc networks. J IET Information Security, 2012;6(2):77-83.
5. Kum Dong-Won. An efficient on-demand routing approach with directional flooding for wireless meshnetworks. J Journal of Communications and Networks, 2010;12(1):67-73.
6. Nakayama Hidehisa, Kurosawa, Satoshi, etc. a dynamic anomaly detection scheme for AODV-Basedmobile ad hoc networks. J IEEE Transactions on Vehicular Technology, 2009; 58(5):2471-2481.
7. Lee Breslau Deborah Estrin etc. Advances in Network Simulation. IEEE Computer,2000;5:78-85.
8. Wing Chung Hung, K L Eddie Law, A Leon-Garcia. A Dynamic Multi-Channel MAC for Ad Hoc LAN. In 21st Biennial Symposium on Communications. Kingston. Canada. 2002; 7:132-138.
9. R. Choudhary, S.Bhandhopadhyay and K. Paul. A Distributed Mechanism for topology discovery in AdHoc Wireless Networks Using Mobile Agents.2000;(5):96-101