IOSR Journal of Electrical and Electronics Engineering (IOSRJEEE)
ISSN: 2278-1676 Volume 1, Issue 5 (July-Aug. 2012), PP 46-51
www.iosrjournals.org

Improve Performance Of Aodv Protocol In Wireless Ad-Hoc Network By Network Coding

Mr.Tushar A. Gawande[1,] Prof.N.N Mhala [2]
[1, 2](Electronics Engineering, Bapurao Deshmukh College of Engineering, Sewagram, India)

Abstract : *This papers presents network coding ideas to improve the throughput of AODV routing protocol, and further proposes a new routing protocol NCAODV. Internationally used simulation tool NS2 will be used to simulate the NCAODV routing protocol. The simulation results show that the proposed NCAODV routing protocol has an improved throughput than AODV routing protocol.*

Keywords*- Ad-hoc On-Demand Distance Vector (AODV), Devnagiri script.*

I. INTRODUCTION

Wireless communication technology has rapidly increased in the last few decades. People wish to use their network terminals (laptops, PDAs, etc.) anywhere and anytime. Wireless connectivity has given users the freedom to move where they desire. There exist numerous different wireless networks varying in the way the nodes interconnect. They can be broadly classified in two categories Networks with fixed infrastructure and Ad hoc wireless networks. Typical for networks with fixed infrastructure is using of access points. An access point (AP) can act as a router in the network, or as a bridge. Examples for this type of networks are GSM and UMTS cellular networks. In ad-hoc wireless networks the nodes themselves use each other as routers, so these nodes should be more intelligent than the nodes in an centralized networks with APs [1].

In communication networks, such as Ethernet or packet radio, throughput or network throughput is the average rate of successful message delivery over a communication channel. This data may be delivered over a physical or logical link, or pass through a certain network node. The throughput is usually measured in bits per second (bit/s or bps), and sometimes in data packets per second or data packets per time slot. Thus the term throughput is essentially synonymous to digital bandwidth consumption [1]. The throughput of a communication system will be limited by a huge number of factors such as:

1. Analog limitations - The maximum achievable throughput (the channel capacity) is affected by the bandwidth in hertz and signal-to-noise ratio of the analog physical medium. Wireless Channel Effects, Skin effect etc fall into this category affect the throughput of the system adversely.

2. IC hardware considerations - Computational systems have finite processing power, and can drive finite current. Limited current drive capability can limit the effective signal to noise ratio for high capacitance links.

3. Multi-user considerations - Ensuring that multiple users can harmoniously share a single communications link requires some kind of equitable sharing of the link. If a bottle neck communication link offering data rate R is shared by "N" active users (with at least one data packet in queue), every user typically achieves a throughput of approximately R/N, if fair queuing best-effort communication is assumed. Packet loss due to Network congestion, bit errors etc are crucial here [1].

II. AD-HOC ON-DEMAND VECTOR

The Ad-Hoc On-demand Distance Vector (AODV) routing protocol [2] is one of several published routing protocols for mobile ad-hoc networking. Wireless ad-hoc routing protocols such as AODV are currently an area of much research among the networking community.AODV is a relative of the Bellmann-Ford distant vector algorithm, but is adapted to work in a mobile environment. Each AODV router is essentially a state machine that processes incoming requests from the network entity. AODV determines a route to a destination only when a node wants to send a packet to that destination. Routes are maintained as long as they are needed by the source. Sequence numbers ensure the freshness of routes and guarantee the loop-free routing.

Whenever an AODV router receives a request to send a message, it checks its routing table to see if a route exists. Each routing table entry consists of the following fields:

1. Destination address
2. Next hop address
3. Destination sequence number
4. Hop count

If a route exists, the router simply forwards the message to the next hop. Otherwise, it saves the message in a message queue, and then it initiates a route request to determine a route. The following flow chart illustrates this process:

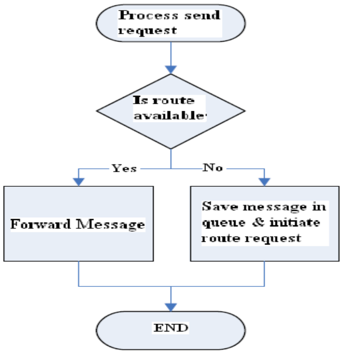


figure 1.AODV routing process

AODV nodes use four types of messages to communicate among each other. Route Request (RREQ) and Route Reply (RREP) messages are used for route discovery. Route Error (RERR) messages and HELLO messages are used for route maintenance.

AODV Route Discovery:

When a node needs to determine a route to a destination node, it floods the network with a Route Request (RREQ) message. The originating node broadcasts a RREQ message to its neighboring nodes, which broadcast the message to their neighbors, and so on. To prevent cycles, each node remembers recently forwarded route requests in a route request buffer (see next section). As these requests spread through the network, intermediate nodes store reverse routes back to the originating node. Since an intermediate node could have many reverse routes, it always picks the route with the smallest hop count.

sequence numbers), or is itself the destination, the node generates a Route Reply (RREP) message, and sends this message along the reverse path back towards the originating node. As the RREP message passes through intermediate nodes, these nodes update their routing tables, so that in the future, messages can be routed though these nodes to the destination.

Notice that it is possible for the RREQ originator to receive a RREP message from more than one node. In this case, the RREQ originator will update its routing table with the most "recent" routing information; that is, it uses the route with the greatest destination sequence number. (See section on sequence numbers).

The vowels (modifiers) can be placed at the left, right (or both), top or bottom of the consonant. The vowels above the header line are called ascenders or upper modifiers and vowels below the consonants are called descenders or lower modifiers. Two consecutive lines touch or overlap each other due to these modifiers. This makes the segmentation of handwritten Hindi text very complex.

The Route Request Buffer:

In the flooding protocol described above, when a node originates or forwards a route request message to its neighbors, the node will likely receive the same route request message back from its neighbors. To prevent nodes from resending the same RREQs (causing infinite cycles), each node maintains a route request buffer, which contains a list of recently broadcasted route requests. Before forwarding a RREQ message, a node always checks the buffer to make sure it has not already forwarded the request. RREQ messages are also stored in the buffer by a node that originates a RREP message. The purpose for this is so a node does not send multiple RREPs for duplicate RREQs that may have arrived from different paths. The exception is if the node receives a RREQ with a better route (i.e. smaller hop count), in which case a new RREP will be sent.

Each entry in the route request buffer consists of a pair of values: the address of the node that originated the request, and a route request identification number (RREQ id). The RREQ id uniquely identifies a request originated by a given node. Therefore, the pair uniquely identifies a request across all nodes in the network.

To prevent the route request buffers from growing indefinitely, each entry expires after a certain period of time, and then is removed. Furthermore, each node's buffer has a maximum size. If nodes are to be added beyond this maximum, then the oldest entries will be removed to make room.

Expanding Ring Search:

The flooding protocol described above has a scalability problem, because whenever a node requests a route, it sends a message that passes through potentially every node in the network. When the network is small, this is not a major concern. However, when the network is large, this can be extremely wasteful, especially if the destination node is relatively close to the RREQ originator. Preferably, we would like to set the TTL value on the RREQ message to be just large enough so that the message reaches the destination, but no larger. However, it is difficult for a node to determine this optimal TTL without prior global knowledge of the network.

To solve this problem, I have implemented an expanding ring search algorithm, which works as follows. When a node initiates a route request, it first broadcasts the RREQ message with a small TTL value (say, 1). If the originating node does not receive a RREP message within a certain period of time, it rebroadcasts the RREQ message with a larger TTL value (and also a new RREQ identifier to distinguish the new request from the old ones). The node continues to broadcast messages with increasing TTL and RREQ ID values until it receives a route reply.

If the TTL values in the route request have reached a certain threshold, and still no RREP messages have been received, then the destination is assumed to be unreachable, and the messages queued for this destination are thrown out.

Sequence Numbers:

Each destination (node) maintains a monotonically increasing sequence number, which serves as a logical time at that node. Also, every route entry includes a destination sequence number, which indicates the "time" at the destination node when the route was created. The protocol uses sequence numbers to ensure that nodes only update routes with "newer" ones. Doing so, we also ensure loop- freedom for all routes to a destination. All RREQ messages include the originator's sequence number, and its (latest known) destination sequence number. Nodes receiving the RREQ add/update routes to the originator with the originator sequence number, assuming this new number is greater than that of any existing entry. If the node receives an identical RREQ message via another path, the originator sequence numbers would be the same, so in this case, the node would pick the route with the smaller hop count.

If a node receiving the RREQ message has a route to the desired destination, then we use sequence numbers to determine whether this route is "fresh enough" to use as a reply to the route request. To do this, we check if this node's destination sequence number is at least as great as the maximum destination sequence number of all nodes through which the RREQ message has passed. If this is the case, then we can roughly guess that this route is not terribly out-of-date, and we send a RREP back to the originator.

As with RREQ messages, RREP messages also include destination sequence numbers. This is so nodes along the route path can update their routing table entries with the latest destination sequence number.

Link Monitoring & Route Maintenance:

Each node keeps track of a precursor list, and an outgoing list. A precursor list is a set of nodes that route through the given node. The outgoing list is the set of next-hops that this node routes through. In networks where all routes are bi-directional, these lists are essentially the same.

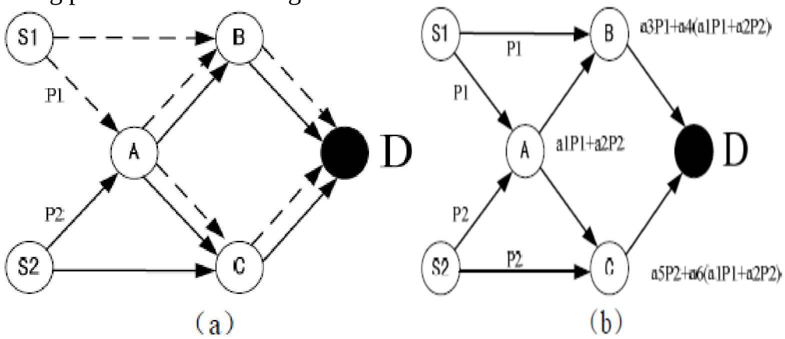
Each node periodically sends HELLO messages to its precursors. A node decides to send a HELLO message to a given precursor only if no message has been sent to that precursor recently. Correspondingly, each node expects to periodically receive messages (not limited to HELLO messages) from each of its outgoing nodes. If a node has received no messages from some outgoing node for an extended period of time, then that node is presumed to be no longer reachable.

Whenever a node determines one of its next- hops to be unreachable, it removes all affected route entries, and generates a Route Error (RERR) message. This RERR message contains a list of all destinations that have become unreachable as a result of the broken link. The node sends the RERR to each of its precursors. These precursors update their routing tables, and in turn forward the RERR to their precursors, and so on. To prevent RERR message loops, a node only forwards a RERR message if at least one route has been removed.

III. USING NETWORK CODING TO IMPROVE AODV ROUTING PROTOCOL

Network coding may be used in a peer-to-peer network to reduce the amount of routing information required by peers to achieve near optimal throughput. In large networks this can be a significant advantage, since otherwise the amount of routing overhead would scale with the size of the network. Unlike scenarios such as the butterfly network example above, network coding does not increase the maximum achievable throughput of a peer-to-peer network. Wireless networks have been designed using the wired network as the blueprint. The design abstracts the wireless channel as a point-to-point link, and grafts wired network protocols onto the wireless environments. However wireless medium is fundamentally different. While wired networks have reliable and predictable links, wireless links have high bit error rate, and their characteristics could vary over short time-scales. Further, wired links are unicast links, but the majority of wireless links are broadcast links. The characteristics of wireless networks might all seem disadvantageous at first glance, but most of them can be used to our advantage with a fresh design. This can be achieved through Network coding. Network coding is a technique where, instead of simply relaying the packets of information they receive, the nodes of a network will take several packets and combine them together for transmission. This can be used to attain the maximum possible information flow in a network. Network coding is a field of information theory and coding theory.

In general network coding technology can effectively reduce data packet transmission times, reduce the usage of wireless network bandwidth, save wireless resources [5]. In wireless networks, there are multiple nodes may establish route to the same destination node at the same time. Now consider two source nodes S1, S2 are looking for a route to reach the node D, it sends a route request packet RREQ packet, the original route request process of AODV routing protocol shown in Fig. 2.



As shown in Fig. 2 & Fig. 3 P1, P2, respectively expresses a RREQ packet for S1, S2 sends to the destination node D, in the Fig. 2, node A, B, C do a linear combination for input message then output, The remaining nodes simply forward the received message. You can see, originally, it should transmit two RREQ packets at the A, B, C nodes, after encoding, these nodes transmission times reduced once. That is, the RREQ packet transmission times reduce total of 3 times in this process. Finally, it decodes at the destination node D to receive RREQ packets P1, P2. NCAODV protocol uses such multi-path to encode the routing requests broadcast packets [7, 8, 9].

NCAODV protocol packet processing

The packet processing of NCAODV routing protocol is as follows:

1) The source node S generates NC-RREQ request packet and initializes it in accordance with NCAODV routing protocol. Because at this time it has not encoded, the global encoding vector is empty. The source node S broadcasts NC-RREQ message to the network.
2) The intermediate M receives the NC-RREQ packet P, first establishes reverse route in accordance with the destination address and last hop address. Then, check the buffer, if there are request packets from different source nodes but reach the same destination address, then uses random network coding to generate P, and do linear combination with these packets in buffer. The partial coding system is randomly selected from finite field GF (2s). The node M will broadcast the generated encoding packets.

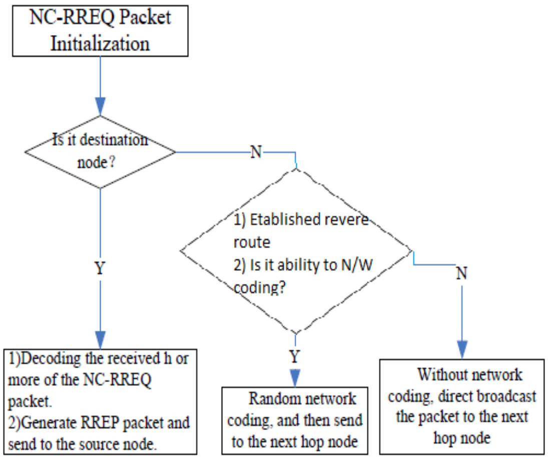

NC-RREQ Packet Initialization
Is it destination node?
N
Y
1) Etablished revere route
2) Is it ability to N/W coding?
N
Y
1)Decoding the received h or more of the NC-RREQ packet.
2)Generate RREP packet and send to the source node.
Random network coding, and then send to the next hop node
Without network coding, direct broadcast the packet to the next hop node

figure 4.NCAODV routing protocol flowchart

3) If the destination node received h or more NCRREQ packets, you can decode. You can use Gaussian elimination method to decode, and need to use the global encoding vector. As long as there are h linear independent equations in linear equations, you can successfully decode. As long as the finite field GF (2s) is large enough, the probability of successful decoding can close to l. After decoding, the destination node first increases record in the routing table, then generates RREP packet, and sends to the source nodes along the reverse route that established in NC-RREQ transmission process.
4) In the process of sending RREP packet to the source node, relay nodes set up forward route to reach the destination node.
5) After the source node S receives the RREP packet, it chooses the best route. At this point, the route establishment process ends. The packet processing of NCAODV routing protocol NC-RREQ is expressed as follow with the flow chart in Fig. 4.

IV. PERFORMANCE EVALUATION

To obtain performance analysis about NCAODV, we use NS2 simulation tool [10], which is current world's most popular network simulation tool.AODV routing protocol has been presenting in the NS2simulation tool. Firstly, we save an AODV source codes for comparing with the improved performance, then according to Fig. 4 protocol flowchart to add the network coding ideas to AODV routing protocol, and achieve NCAODV routing protocols.

Simulation Environment :

The whole simulation uses kinds of simulation scenarios, respectively simulates end to end delay, throughput, and the successful delivery rate of packets. The nodes randomly distributed in the 400*400m2 rectangular areas.

The first scene, in order to get the throughput and successful delivery rate of packets, the simulation used total 40 nodes. In order to test performance under different scenarios, the pause time varies from 5ms gradually increasing to 100 ms during the simulation

The second scene, in order to get end to end delay performance of improved NCAODV, a total of 40 nodes, with varies pause time from 5ms gradually increasing to 100ms during the simulation.

Simulation result:

The ns2 simulator simulate both AODV & NCAODV,we get following result in Fig 5 & Fig 6.

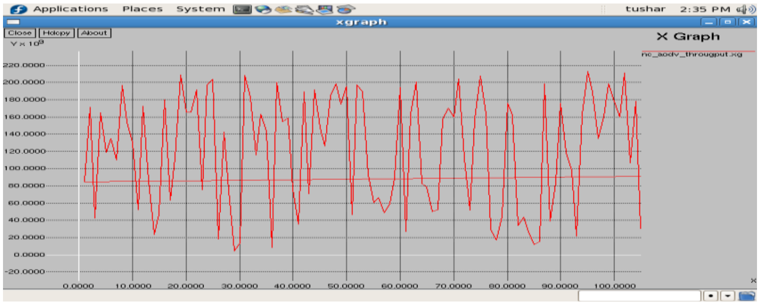


figure 5. Throughput for AODV

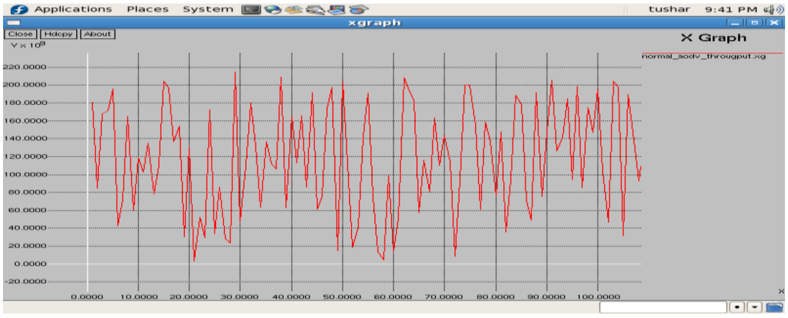


figure 6. Throughput for NCAODV

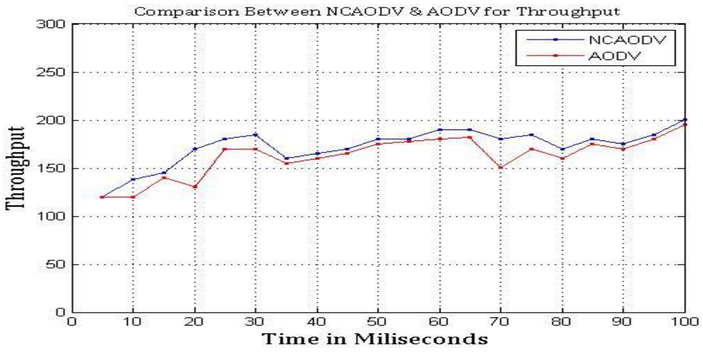


figure 7.comparison between NCAODV & AODV for throughput

Fig.7 shows the comparison between throughput under different number of NCAODV and AODV for time. Comparing AODV & NCAODV we get following comparison shows NCAODV gives better throughput than AODV.

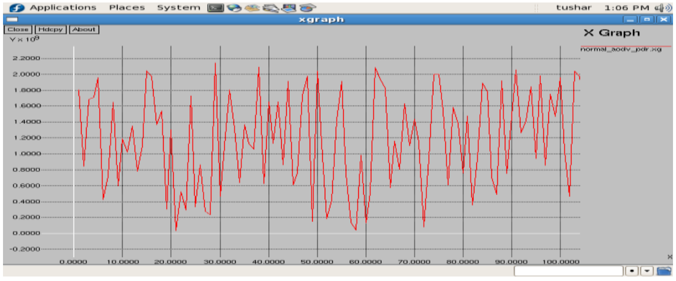


figure 8. Packet delivery ratio for AODV

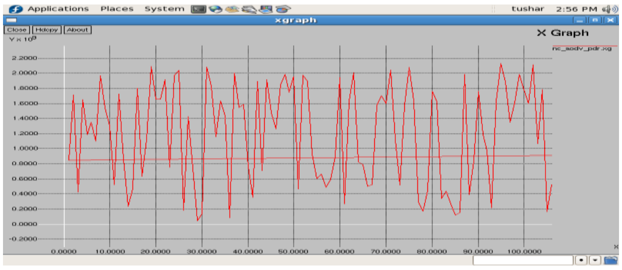


figure 9. Packet delivery ratio for NCAODV

The packet successful delivery rate is the ratio of the number of packets received by the destination node and number of packet send by CBR source node. The ns2 simulator simulate both AODV and NCAODV, we get above result in Fig .8 & Fig .9,

The NCAODV gives better packet delivery ration than AODV, shown by their comparison in Fig.10,when the time is increases, NCAODV has a higher packet delivery rate than AODV.

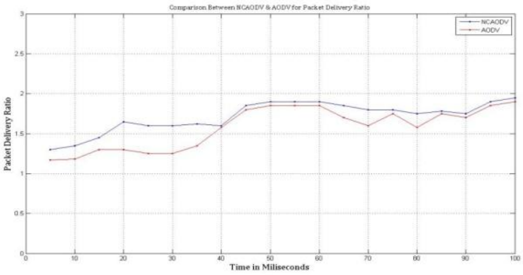


figure 10.comparison between NCAODV &AODV for packet delivery ratio

The NCAODV improved the performance which is based on AODV routing protocol.

V. CONCLUSION

The use of network coding can over the throughput problem of standard AODV protocol. The proposed method of NCAODV algorithm has a certain coding gain than the traditional method which did not use network coding, when the network data flow increases, it can effectively improve the throughput of the whole network.

REFERENCES

[1] Schiller J. Mobile Communications. Addison Wesley, 2nd edition, 2003.

[2] Royer E.M. Perkins C.E. Ad-hoc on-demand distance vector routing. Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, p.90, 1999.

[3] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, “Network Information Flow,” IEEE Transactions on Information Theory, vol. 46, no. 4, July 2000, pp. 1204-1216.

[4] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, ”Architecture and Evaluation of An Unplanned 802.11b Mesh Network”, In: Proc of ACM MobiCom 2005, Cologne Germany, Sep 2005, pp. 31-42.

[5] Yang Tao, Yong Li, Jian Li, and Hong-cheng Huang, "Improve the Throughput of AODV Based on Network Coding Ideas", WASE International Conference on Information Engineering 2010.

[6] S. Li, R. Yeung, and N. Cai, “Linear Network Coding”(PDF (http:/ / pdos. lcs. mit. edu/ decouto/ papers/ li03. pdf)), in IEEE Transactions on Information Theory, Vol 49, No. 2, pp. 371-381, 2003.

[7] P. Gupta, and P. R. Kumar, “The Capacity of Wireless Networks”, IEEE Transactions on Information Theory, vol. 34, no. 5, pp. 910-917, 2000.

[8] R.Draves, J.Padhye, and B.Zill. "Comparison of Routing Metrics for Multi-Hop Wireless Networks". In Proceedings of ACM SIGCOMM, 2004.

[9] Z.Li and B.Li. "Network coding: The case for multiple unicast sessions". In Allerton Conference on Communications, 2004.

[10] R.Draves, J.Padhye, and B.Zill. "Comparison of Routing Metrics for Multi-Hop Wireless Networks". In Proceedings of ACMSIGCOMM,2004