Enhance Recovery Schema in Proactive Protocol Over MANET by Computing a Backup Path

1Albara Awajan, 2Firas A. Albalas, 2Radwan S. Abujassar

1School of Computer Science & Engineering, Al-Balqa Applid University
2Jadara University, Irbid, Salt, Jordan

Abstract: In mobile ad hoc networks (MANET), mobile nodes can form an autonomous network without the need to use an existing network infrastructure. Hence, the nodes are free to move from one area to another without any notification. The network can have high or low density of mobile nodes and this is related to the freedom. An increase/decrease in the number of mobile nodes requires the routing protocol to re-update the routing table frequently, which leads to an increase overhead and a lack of stability in the network. In this paper, performance diversities with and without computing the backup route are analysed using a dynamically changing network containing different numbers of mobile nodes that they are moving from low to high speed. The backup route considers node density, variation of speed, movements and backup route length with longer lifetimes. The backup route achieves significant improvements by reducing the packet loss and improves the continuity index for live video streaming, while the routing protocol maintains the new routing table. This backup route helps nodes to reroute the traffic to the destination whilst the new discovery service is completed. Furthermore, ABRT shows better scalability compared to DSDV routing demanding environments with high mobility and heavy traffic loads. A simulation (using an NS2 simulator) is performed to evaluate different scenarios containing a high/low densities to measure and analyse the effects of varying movement (from low to high speed) for each node in the network. We present the results of the simulations for networks of (100,50,25) mobile nodes with variation of movement (1,10,20,40 m/sec). The results show that lost packets, end-to-end delay and the continuity has achieved high performance with backup path in low speed node movements comparing with high movements.

Key words: Network Protocols ;(DSDV) Destination Sequenced Distance Vector ; ABRT (Alternative Backup Route Table );Wireless Network;Mobile Ad Hoc Network.

INTRODUCTION

In the past few years, mobile ad hoc networks have become increasingly used due to the wide availability and rapid deployment of wireless transceivers in a variety of computing devices, such as, PDAs, laptops and desktop computers. Wireless communication has brought essential changes to telecommunications and data networking. The ether is used as the transmission medium, allowing greater flexibility and networks can be deployed quickly where cabling is difficult. Good performance and low prices encourage users to choose these new kinds of networks. In such situations, a collection of mobile nodes with a wireless network interface can form a transitory network. Recently, mobile ad hoc networks have received extensive attention in both industrial and military applications because of the ability to create a network whilst moving from one place to another and the fact that they do not require any pre-designed infrastructure. Mobile nodes in MANET networks make up sets of nodes that form a network for sending and receiving data without the need for establishing an infrastructure or a centralised administrator (Perkins, C.E. and P. Hhagwat, 1994; Mohanchur Sarkar, 2010; Sung ju Lee and Mario Gerla, 2001). Each mobile node will send a packet to its destination by demanding assistance from other intermediate nodes within the range. The flexibility of node movement from one area to another, result in problems primarily due to loss of connectivity within the network. When loss of connectivity occurs, the routing protocol starts to broadcast packets to inform other nodes within the topology to update the routing table and compute a new shortest path between source and destination. Ad hoc wireless networks are frequently affected by failures; e.g. when nodes move radio propagation range. It is, therefore, highly desirable to develop a recovery mechanism to improve the quality of service (QoS) for networks. When mobile nodes move or fail, one of two things can happen. Firstly, if a node on the primary path moves out of range, then other nodes lose their connectivity; Secondly, the routing protocol begins to re-establish a new path between source and destination. As a result, loss of data packets and end-to-end delays will increase. Our goal is to analyse the network performance with and without a backup routing path. The flexibility of mobile node movement provides many challenges for routing protocols in ad hoc network. Movement in topology causes overheads and increases delay. The overhead will be higher in high density topologies according to the amount of traffic being carried by different sources and the broadcast of control packets. The ABRT has been used with a high/low
density network with varying movements to evaluate the performance of the backup route quality when loss of connectivity occurs. The ABRT can find many alternative recovery paths for high density network. The ABRT computes an alternative routing table to its original by selecting one of the adjacent nodes which has less number of hops to the destination. The ABRT will reroute the traffic once the loss of connectivity occurs on the primary path. The ABRT mechanism aims to recover from failure in less time by pre-computing a backup routing table which has alternative paths along which to pass the traffic when failure occurs. The mechanism aims to reduce delays and packet loss for a variety of applications. In this paper, we use the range between nodes to create a primary routing table which is computed by an ABRT algorithm. The main contribution of this paper is the intelligent evaluation of the backup route. Different scenarios are explained consisting of high and low density mobile nodes with various speeds.

This paper is organised as follows: Section II discusses the characterise of MANET network. Section III briefly illustrates our approach of computing the backup route and the basic concept of the ABRT algorithm. Section IV discusses related work. Section V shows the performance evaluation via simulation and Section VI concludes how ABRT can be improved and future work.

II. Characteristics of Manet:
A mobile ad hoc network (MANET) is an independent system of mobile nodes linked by wireless connections. These nodes are thus free to move arbitrarily; therefore, the topology of wireless networks can be changed swiftly and in an unpredictable manner. Generally, direct communication in MANETs is possible only between adjacent nodes. Thus, communication between distant nodes is established via multiple-hop. The locations of these nodes may change dynamically; therefore the interconnections between the adjacent nodes may change continually. Each mobile node work as a host and a router, relaying data packets from one node to another. MANETs have therefore many characteristics that make them distinct from other wireless and wired networks.

A. The Vulnerabilities and Challenges of Manet:
The key challenges in a MANET network result from the decentralised nature and lack of central administration like a base station, access point or server. The opportunity for node movement is thus very high. In addition, all communications are carried out through the wireless medium. These unique characteristics present appreciable challenges for MANETs (Van Oorschot, P., 1997; Li, W. and A. Joshi, 2008; Xing, F. and W. Wang, 2006) such as:

• Restricted Power Supply:
  In comparison with a wired network where the nodes can get an electrical supply directly from the mains. In mobile ad hoc networks (MANETs) nodes are generally operated by small batteries with a limited lifetime, which can cause problems such as loss of connectivity.

• Unreliability:
  Due to the limited battery supply and mobility, MANET cannot rely on being able to communicate with their neighbours. Further, some nodes may behave in a selfish manner when they find that there is only limited power supply.

• Lack of Centralised Management Facility:
  MANET is highly dynamic and large scale therefore they cannot be easily monitored. Benign failures in the network fairly common such as collision and packet dropping.

• Scalability:
  In MANETs nodes entering and leaving the network can change the network topology; the network may consist of hundreds or even thousands of nodes, therefore the routing protocol configurations and key management services should be adjusted to match these new conditions.

  In the case of path construction, the routing protocol uses a series of messages to compute paths along the topology between source and destination. The routing protocol can be divided into proactive (table driven) and re-active (on-demand).

B. Characterisation of Node Stability:
As a result of the variable speed of mobility approach, we propose a new algorithm that can select an adjacent node to reroute the traffic when node on the primary path failed. The frequently changing of the topology by joining/ or departure the existing node that will lead to increase the overhead in the network.
III. Our Model:

Our approach uses the Random Waypoint Model that allows the nodes to move randomly within the predefined area. Each node will identify its destination to start sending data packets. Hence, the nodes will start travelling with various of speeds from low to high which cause discontinuity changes of the topology and routing table. In the case of movement, the number of neighbours may change. Figure 1 shows the direction each node can move within the boundary range of an area. Each node will start to move by selecting a new direction with a different speed.

In network communications, the use of pre-computed second paths provides a solution for when failure or loss of connectivity occurs. The pre-computed path performs well for high density. In ad-hoc wireless, mobile nodes in the same range can exchange messages with their neighbours. Through these messages, the routing protocol will start to compute the routing table between source and destination. The ABRT algorithm computes the backup routing table from the primary one to ensure that the alternative path will be disjointed from the primary one. Our motivation is to find an appropriate backup path between source and destination, and reroute the traffic via this backup once failure occurred on the network. In the ABRT algorithm, it computes the backup route without degrading the network performance. When the routing protocol has computed the routing table for the topology then the ABRT algorithm will start to check the adjacent nodes, for each node on the primary path (from the routing table) it will check each adjacency to find available route to the destination not connected to the primary path. The ABRT algorithm will select the best adjacent node based on the distance between it and each node on the primary path. When the node starts to move it is time find a new adjacent node that is still in the range as a backup in case the other node moves out of range. However, each node on the primary path will select an adjacent node has a disjoint path to the destination with guarantee local loop freedom in the network. Hence, a loop will be avoided and the packets will not return back to any nodes through which the traffic.

Fig. 1: Simulated Movement in the Random Mobility Model.

A. Proposition:

The basic principle of the ABRT protocol is how to find a backup path from the original routing table by excluding all nodes on the primary path. This leads to a reduced calculation time by excluding the number of hops connected to the primary path. The ABRT protocol selects a backup adjacent node with respect to the original routing table. This gives each node the ability to anticipate the second shortest path between source and destination. However, the second shortest path is precomputed in advance in order to re-route data packets in the case connectivity loss. In addition, when a node on the primary path moves from the area, the adjacent nodes will detect a link break by receiving a link layer feedback signal and from the HELLO data packets which confirm the failure by not receiving any acknowledgement during the interval time. The ABRT protocol, via the backup routing table, will pass traffic to destination without waiting to re-compute a new routing table. The ABRT protocol is a pre-active mechanism, because it is able to compute a new routing table (including a backup path) for all nodes on the topology in advance. However, we do assume here that the source node has at least one adjacent node not connected to the primary path. The ABRT protocol involves choosing one of them as a backup node to re-route a packet through when a failure occurs. Hence, the ABRT protocol will exclude the first next hop for each node with regard to the routing table. Hereafter, when all nodes receive the data packets, they will start to check if there is an alternative route to each destination. In the case of high density of mobile nodes, the ABRT algorithm will select the adjacent node based on the range distance from each node on the primary path. When the nodes start to move, the ABRT will reroute the traffic via the adjacent node which has already been selected. Each adjacent node will reply by an acknowledgement, which will contain a 0 or 1. When the acknowledgement packets contain 1 that means the adjacent node has a different route and it can be a backup
node in the case of failure. If the acknowledgement packet contains "0", then this node cannot be a backup adjacent to the destination. When the source node checks the packets and check if it includes 1, it will add this node to the new routing table as a backup. If it contains a "0", the node will check answers from other adjacent nodes. Hereafter, if all acknowledgement headers from all adjacent nodes contain "0", then the ABRT algorithm will send another packet to ask the adjacent node if it has a neighbour which is unconnected to the primary path with a disjointed route to the destination (not via the primary path). If any acknowledgement contains "1", then the adjacent node will send an acknowledgement that contains "1" then it can be a backup node via my adjacent node. When the source node receives this acknowledgement, it adds the adjacent node as a first hop backup and the node that is adjacent to the first hop will add it as a second next hop. The ABRT algorithm works until the backup routing table has been completed for all nodes in the network. Each node on the primary path connects with more than one adjacent node. Each node on the primary path is able to re-route traffic through an appropriate adjacent node. We examine high and low density topologies, making each node have at least two neighbours that can re-route data packets when failure occurs. The ABRT is described in algorithm 1.

In our schema, the pre-computed second paths provide a good solution when failures occur. ABRT evaluates the backup routing table from the primary one to ensure that any alternative path will be disjointed from the primary. Figure 2 shows the primary and backup path. When the routing protocol determines the primary path the ABRT algorithm will start to check each node to determine how many adjacent nodes are within range and not connected to the primary path. The ABRT computes the backup routing table from passing traffic. In fig 2 (a), the primary path from source to destination is computed by a routing protocol. By using the radio propagation range the ABRT can discover any adjacent nodes by from to compute the backup path to the destination, which is based upon the number of hops. When there is more than one adjacent node, the ABRT will start to check which node has a disjointed path to the destination (i.e. to construct a backup path). In addition, the ABRT algorithm evaluates an appropriate backup route using the primary routing table. Any nodes connected to the primary path will be excluded from the next hop in the backup route. Each node randomly takes a position (X, Y) in the radio propagation range for an area 1000m by 1000m. In ABRT, the nodes will start to enquire from their adjacent nodes if they have a disjointed path from the primary (with regard to the primary routing table) to the destination. Because ABRT considers all nodes as a source and destination then each node will check the acknowledgements from adjacent nodes to see if there is any available route to the destination. If one of the adjacent nodes become the intended node it will then insert it into the backup routing table as a first hop. If more than one alternative path exists, then the node will select the best one (based upon the number of hops) that has a greater history for reliability.

```
Algorithm 1 AlternativePath ABRT

1. procedure AlternativePath(T_g, x, d, edges_to_avoid)
2. T_g: The routing table
3. V: The vertex set in graph (G(V,E))
4. T_i(v): The set of adjacent vertices to a vertex v
5. x: The source vertex
6. d: The destination vertex
7. edges_to_avoid: A list of edges to exclude while implementing the alternative path
8. p(x, d): An alternative path such that edges_to_avoid ∩ P(x, d) = ∅
9. e: A path
10. Q: A queue of couple (path,vertex)
11. Enqueue: Insert an element in a queue
12. Dequeue: Remove an element from a queue
13. Front: The element at the front of a queue
14. p(x, d) ← ∅
15. if s ≠ d then
16. e_0 ← ∅
17. Q ← ∅
18. Enqueue(Q, (e_0, s))
19. while Q ≠ ∅ and p(s, d) = ∅ do
20. (e_0, s) ← Front(Q)
21. for all k ∈ T_i(s) do
22. c ← (s, k)
23. if (e_0 ∪ c) ∩ edges_to_avoid = ∅ then
24. if p(k, d) ∩ edges_to_avoid = ∅ then
25. p(s, d) ← e_0 ∪ P(k, d)
26. break
27. else
28. Enqueue(Q, (e_0 ∪ c, k))
29. end if
30. end for
31. Dequeue(Q)
32. end while
33. Q ← ∅
34. return p(x, d)
35. end procedure
```
IV. Related Work:

In MANET mobility, all nodes can move from one domain to another without notification. Hence, the task for the routing protocol becomes very complex because it cannot predict how many nodes will move in or out of range (Perkins, C., 2003). Different routing protocols have been published for different environments to improve the network performance when connectivity is lost by nodes moving or failing (Saeed Shalbuz, 2006). In (Kim, D., 2003) indicates that MANET’s features and environments, such as Bandwidth (BW), resources and limited energy, result in high QoS, security and reliability. There are two categories of routing protocols for MANETs proactive and reactive (Poonam, K. Garg and M. Misra, 2010; Rashidi, R., 2009). A proactive routing protocol updates pairs of nodes by flooding via a periodic broadcast. This brings routing tables up-to-date for each node in the network. However, a reactive routing protocol detects a new route only when it is required. Some proactive routing protocols (such as DSDV, OLSR, CGSR and WRP) trigger messages that can detect links when they fail (Park, V.D. and M.S. Corson, 1997; Moy, J., 1995). Based on these messages, the routing protocol can construct and maintain routes to the destination. Reactive protocols such as DSR, AODV (On-demand Distance Vector and TORA, overheads will be reduced because new paths between two nodes will only be created when a failure occurs. The DSDV routing protocol is based on the Bellman-Ford algorithm. It is similar to a DSR protocol which is an AODV protocol (Subir Kumar Sarkar, 2007). This is because they both use a similar algorithm. Multipath routing has been introduced and considered in wired networks for improving their performance (Lee, S.J. and M. Gerla, 2000). The proactive protocols are designed to send out extra messages to create a disjointed path from source to destination. (Radwan Abujassar and Mohammed Ghanbari, 2011; Marti, S., 2000) presents a new algorithm called the Alternative Routing Table (ART) algorithm, which can send extra data packets to enquire if neighbouring nodes have an alternative and disjointed pathway from the primary one to its destination in a wired network. The number of data packets sent to each node will depend on the number of adjacent nodes that are not connected to the primary path. An efficient routing protocol algorithm has been constructed in terms of achieving robustness and fast convergence in case a node goes down. As such, we have enhanced the ART algorithm to make it work on a wireless ad hoc network by constructing a primary routing table based on the propagation range with the number of hops, and a backup routing table based upon the number of adjacent nodes to select the best path. In (Prayag Narula, 2008; Juan Antonio Cordero Emmanuel Baccelli, 2010; Dhote, C.A., 2010), the OLSR routing protocol is shown to be proactive in ad hoc networks. The OLSR has Multi Point Relay (MPR) nodes which are used to send link state messages to construct a routing table. In OLSR, two kinds of broadcasts are sent HELLO and Topology Control (TC) messages. Each node will send HELLO message to its neighbours to check if connectivity is up or down every two seconds, as a waiting time of six seconds is considered too long. The TC message is thus based on the information collected by the HELLO messages. The interval time is five seconds. The holding time is fifteen seconds to detect failure. In (Yong, H., 2007; David B. Johnson and David A. Maltz, 1996), the source node knows the complete route hop-by-hop to the destination. The route from source to destination is stored in a route cache. Hence, the ARTSD algorithm is considered a proactive mechanism that can create a new backup routing
table with the shortest path between source and destination. We have enhanced the ART algorithm by constructing a new algorithm, which finds adjacent nodes for each node on the topology based on the propagation radio range (Abujassar, R.S. and M. Ghanbari, 2011). The OLSR routing protocol will re-route data packets only after re-computing a new routing table and updating the information to all nodes on the topology. Depending upon the HELLO message interval times, the re-routed traffic will take longer and this will lead to an increased loss of data packets and reduced throughput (Broch, J., 1998). Internet Protocol recovery emphasises two cases. First, the time required to detect failure. Second, the time to compute the shortest path. In (Perkins, C.E. and E.M. Royer, 1999), the author mentions how recovery of the network can be achieved within a short time when failure occurs. The aim of IP recovery is to offer a loop free protection mechanism in the network. Loops in a network are one of the main problems with some existing techniques. The constraint-based routing protocols use the metrics instead of the shortest path between nodes to find a suitable route. In QoS routing schema, the Core Extraction Distributed Ad Hoc Routing (CEDAR) algorithm has been introduced for a medium size ad hoc network. CEDAR is an on demand routing protocol (Veres, A., et al., 2001). It advertises to all core nodes with high link bandwidths to compute the path. The idea behind the multiple paths QoS routing schema is to try to find a number of paths between source and destination based on high capacity bandwidth requirements. Service providers guarantee a network performance via a set of measurements such as delay, jitter and loss of data packets. These parameters are a part of QoS that can optimise along the network to invest in the provisioning of resources in cases of increased traffic or node failures (Liao, W.H., et al., 2001; Royer, E.M. and C.K. Toh, 2001). In (Abujassar, R.S. and M. Ghanbari, 2011), the author proposes a new algorithm, DBRT (Driven Backup Routing Table), to improve the existing proactive protocols such as DSDV (Destination Sequenced Distance Vector) protocol by creating a backup routing table to provide multiple alternative routes. The DBRT algorithm identifies adjacent nodes for each node in the same range and then selects one of these as a backup next hop according to the available path to the destination. In (Ververidis, C.N. and G.C. Polyzos, 2005), the author discuss how to enhance service discovery in terms of service discovery and energy consumption. In addition, the paper is mentioned about the affections of node speed and density on the duration of discovered services. In (Wang, J. and S. Medidi, 2008), the author present the problems that will be arise in the MANET routing protocol design such as broadcast storm, stale route and faulty node or frequent of nodes movement from area to another. The author introduce a new routing protocol call Densityfirst ad hoc routing protocol. This protocol consider the node density, route length to choose the routes with longer life times and better throughput. In our work we evaluate the performance in different density of nodes when there is backup route has computed in advance to improve the performance in the network.

V. Simulation Experiment:
A. Simulation Environment:
A Network simulator (NS2) was used to evaluate the proposed algorithm with high/low density nodes and freedom of movement for each node in the network. We compared the simulation results of the DSDV protocol with and without our schema. NS2 offers good support for node mobility in ad hoc networks. We created a network consisting (25,50,100 nodes), and we simulated each scenario with each nodes speed in (m/sec) distributed between 1,10,20 and 40m/s. A radio propagation range with a transmission power of 0.28 watts was used, allowing each node to send or receive a packet to or from its neighbours over a distance of up to 250m. At the physical and data-link layer we used the IEEE 802.11b protocol. We used a Random Waypoint Mobility model (Van Oorschot, P., 1997) for the node movement. We simulated each scenario with different speeds. During the simulation, each node’s speed changes from low to high (in meters/second). The first speed is 1m/s (low mobility), then the nodes start to move from low to medium speed (10 m/sec) , and then to 20,40m/s (high mobility speed). The simulation duration was 250 seconds in each experiment. The packet size was 512 bytes and the bit rate was set to 2Mb/sec. We used an IEEE 802.11 Distributed Coordination Function (DCF), as the wireless channel, which can share in an ad hoc configuration. In relation to density, we simulated two scenarios: the first included 50 nodes moving on a terrain of 1000 by 1000 meters, following the random waypoint model with variations in speed from (1-40)m/s (minimum speed was still 1m/s), the second scenario (high density) was identical to the first but included 100, 50 nodes. Both scenarios had a duration of 250 seconds each.

Based on the above, we addressed three different cases: (i) 25 nodes moving with from low to high speed set as 1, 10, 20, 40m/sec; (ii) 50 nodes moving randomly in different ways, changing speed from low to high set as 1, 10, 20, 40m/sec; (iii) 100 nodes moving randomly while changing speed from low to high set as 1, 10, 20, 40m/sec. Traffic rates was generated from the source node to destination during the simulation time with 200Kb packets. We repeated each experiment 10 times in order to get stable results and to allow averaging.

B. Performance and Analyses Evaluation:
We examine the low density longer live nodes and compare them with high/medium densities. This is because of the difficulty in discovering and maintaining new paths along with increased overhead associated with high density. The high speed movement has an increased probability of paths breaking compared to nodes.
move at a slower speed. We measured the packet delay for high/low density with high/low speed movements. We observed that when nodes start to move from a low to a high speed, the loss of packets increases.

Figures 3, 4, and 5 show the number of packets received for high and low speed with high and low density. In the case of ABRT protocol, in some cases there is a high delay time according to the numbers of mobility nodes and the different movements. The delay time becomes higher when the density of the mobile nodes is high in relation to the time to compute the new routing table. In addition, the DSDV routing protocol uses a primary path to send the data packets to the destination. When the primary path goes down, DSDV will start to re-compute the new one which will be the shortest new primary path with respect to the computational time that is required. In the case of ABRT, a number of packets will receive with a delay it can be high or low. This is because when the primary path 'goes down' then ABRT will pass the traffic via the backup route which can contain a large number of hops to the destination. When ABRT reroutes the packets through an alternative path each node on the backup path will try to get rid from of these packets by sending them to the next hop before the loss of connectivity occurs. By changing the speed and based on the direction movement the delay becomes smaller where the destination and source are moved closer to each other. When the number of nodes increases the time for updating the new routing table is also increased.

**Fig. 3:** Delay packets 25 Nodes.

**Fig. 4:** Delay packets 50 Nodes.

**Fig. 5:** Delay packets 100 Nodes.
Based on figs. 3, 4 and 5, we have shown in figure 6 the average end-to-end delay for the three cases. In ABRT, the backup path is pre-computed in case a node on the primary path moves or fails. In this case, the node that is connected to the newly failed or moved node will re-route the traffic according to the backup routing table. In some cases, the ABRT and DSDV average delay are similar with a higher performance for the ABRT. This is because when loss of connectivity occurs, the ABRT reroutes the packets to destination via an alternative path which can be longer than the primary path. In high speed case, the end to end delay is less, because when the nodes move in a random direction it increases the probability of the source node being directly connected with the destination. As a result the packets arrive at the destination in a shorter time. In addition, when the destination starts to move away from the source node, the ABRT algorithm has an alternative path over which to pass the traffic to the destination. In the high density (case low speed), the alternative path may contain a large number of hops when the destination node is distant from the source. In addition, the delay for each packet will increase in relation to the time taken to update the routing table. The low density results show the packet delay is reduced for a number of reasons: Firstly, less additional messages are broadcast between the nodes. Secondly, less time is required to compute the routing table and less traffic is sent between the nodes in the network. The delay in the ABRT protocol, which is obvious from figure 6, was less than that for the DSDV protocol because the latter has to retain all its routes in a static routing table. When any node on the primary path moves out of range or fails, the DSDV routing protocol needs to re-compute a new table when new nodes join the network or depart. The performance for ABRT shows the effect of computing an alternative path. The backup path passes the traffic to the destination from the reliable adjacent node which can continue sending packets to the destination before losing the connection.

Fig. 6: End to End Delay.

We evaluate the packet loss with various speeds of 1, 10, 20, 40 m/sec. Figure 8 shows low density with low/high speed for each node in the topology. In low density, it is shown that the average packet loss decreases compared with the DSDV protocol. When the speed increases, the packet loss will increase when the nodes start to move from low to high speeds. Hence, the loss of packets will increase in relation to the rates of changes in the routing table. The nodes need to rebroadcast messages to start an update the routing table. In the case of ABRT, the second reroute reduces packet loss when the node moves from low to high speeds. The alternative path can alleviate packet loss when connectivity is lost compared with DSDV protocol.

The results shown in figure 7, 9, illustrate that by increasing node density by one half, the service duration distributions follow the same pattern. As a result of only a modest increase in packet update size good scaling is observed when density increases the length of routing messages plays a significant role in the high density cases where congestion is present. Hence, when messages increase according to node density, there is a corresponding increase in packet loss due to congestion.

Figure 7 shows the result for 50 nodes with various speeds. In some cases packet loss is less than that shown for 100 and 25 nodes seen in figs 9, 8. The variation of speed from low to high will result in an increase in message correspondingly broadcasting between the nodes, which will increase the packet loss. The way node moves is an important factor affecting delay packet loss. This is because the destination nodes can move far from the source, thus requires a large number of hops for the packets to arrive at their destination.

Figure 9 shows a high density case with packet loss increasing for DSDV and reducing for ABRT. The DSDV protocol takes up to 15 seconds to compute the new routing table for medium density. The ABRT algorithm computes the backup path based on of adjacent node which has a disconnected path to destination. In the case of variations of speed from low to high, the packet loss increased for both DSDV and ABRT. Traffic is
rerouted along an alternate path, which is computed by the ABRT protocol. In the DSDV protocol, the loss of data packets increases based on the number of nodes and the number of hops to the destination. The DSDV protocol generates messages to maintain the routes. When a collision occurs, the packet loss is increased for both. This because will increase the probability of collision. The IEEE 802.11 sending RTS packets can reduce collisions in the network. However, the DSDV protocol needs a period of time to re-route the traffic.

Fig. 7: Loss Of packets for 50 Nodes.

Fig. 8: Loss Of packets for 25 Nodes.

Fig. 9: Loss Of packets for 100 Nodes.
This is delay undesirable. It takes two seconds for each node to re-compute a new routing table and a medium topology will take fifteen seconds. However, ABRT will start to compute two routing tables; first, a primary routing table based on the distance between nodes and then a backup routing table. The latter table will start to be calculated after the primary one has been computed. Therefore, ABRT will re-route the traffic directly to an adjacent node on the backup routing table if a node on the primary path moves out of range or fails. This will lead to a reduction in the packet loss and an increase in the continuity index between the source and destination.

Figure 10 measures the continuity index for each scenario. Therefore, the continuity index can be defined as follows:

\[
\text{Continuity Index} = \frac{N_p}{N_s};
\]

where \(N_p\) is the number of blocks which arrive before playback deadlines and \(N_s\) is the number of all blocks in one content. Each scenario shows the continuity index per one second for the packets arriving at high and low density. Any packets arriving after one second will be negligible. This figure indicates that the ABRT algorithm has achieved in low density a high continuity index compared with high density. This mean that the backup route has improved the continuity index in live streaming packets, as expected. However, in low density comfortable viewing performance is achieved for video traffic. In figure 11, the continuity index reduces when the nodes move from low to high speed. As mentioned above, if any packets arrive after 1 second the user needs to play out in the case of live streaming. This figure shows that broadcasting messages will increase overhead, which will affect the continuity index for live streaming. However, the ARBT algorithm shows a high continuity index compared with the DSDV protocol.

**VI. Conclusions:**

Ad hoc networking have received much attention research in recent years, as available wireless networking and mobile computing resource bases are now capable of supporting this technology. A variety of new routing protocols targeted specifically at the ad hoc networking environment have been proposed. The DSDV protocol
shows the effect of computes a backup routing table in advance for a high/low density node with a variation in the movement speed of the nodes when a loss of connectivity occurs. No detailed performance comparison between the protocols has previously been available. This paper has shown the effect computing a backup route in an ad hoc networking protocol for mobile nodes with various speeds of movement, and has analysed the effect of the backup route on real applications. Using this simulation environment, we have presented the results of a detailed comparison for an ad hoc network routing protocol with our proposed algorithm. We have simulated each protocol in networks of 25,50,100 nodes moving at (1-40)m/sec and communicating with each other. In some cases, the number of densities studied perform well, such as when there are many alternative backup paths leading to the destination. In other scenarios high density has drawbacks, such as increasing the congestion and the overhead in the network. DSDV performs quite predictably, delivering virtually all data packets when node mobility rate and movement speeds are low. But it fails converge as node mobility increases. In scenarios with 100 or 50 sources, the network was unable to handle all of the traffic generated by the routing protocol and a significant fraction of data packets were dropped. In cases of high speed, the nodes move in random ways leading to the possibility of the destination moving nearer to the source in a short time. This leads to a reduced delay. On the other hand, high speed shows an improvement in the delay time with an increase in packet loss. The high speed makes the routing protocol update the routing table frequently because of the new nodes joining and existing nodes departing. The performance of ABRT has been very performed well at all mobility rates and movement speeds. However, its use of source routing to increase the continuity index for live streaming traffic compares well with the DSDV protocol. However, the earlier one still requires the transmission of many packets to construct a backup route which can produce an extra overhead in the network. In future work, we will ensure the backup routing table contains a path that has a longer life so as to offer a better throughput for real applications.

REFERENCES


Radwan Abujassar and Mohammed Ghanbari, 2011. An efficient algorithm to create a loop free backup routing table, pages 148-152. International Conference on Information and Computer Networks ICICN.


