An Alternative backup Route with a Loop Free Routing Table Short Distance in Ad hoc Wireless

Radwan S. Abujassar[1], Mohammed Ghanbari[2], Fellow IEEE
School of Computer Science and Electronic Engineering,
University of Essex,
Colchester, UK Wivenhoe
Park, CO4 3SQ
{rabuja,ghan}@essex.ac.uk

***Abstract*—In a mobile ad hoc wireless networks (MANET), nodes communicate with each other via packet radios. However, in a MANET network, the mobility of the nodes and power limitations lead to an increased loss of connectivity between the mobile nodes when changing the route between nodes, which requires the routing table to be frequently updated. Hence, routing protocol plays an important role in updating the routing table when the nodes move from one area to another. By pre-computing a backup routing table in advance, the source can pass traffic through it without needing to wait for the routing table to be updated from where the failure occurs. This paper proposes a new algorithm, ARTSD (Alternative Routing Table Short Distance), to improve the existing DSDV (Destination Sequenced Distance Vector) protocol by creating a mesh network with different numbers of nodes to provide multiple alternative routes. The ARTSD algorithm determines the adjacent nodes for each node on the primary path and then selects one of these adjacent nodes as the backup next hop according to the least numbers of hops to the destination. The results show that loss of data packets, throughput and end-to-end delay times between source and destination have been improved by avoiding loops on the network. The results show that the new protocol does not degrade network performance by sending extra messages to construct the new backup routing table. A simulation (using an NS2 simulator) was undertaken to demonstrate the difference between using a DSDV protocol with or without the proposed schema.**

Index Terms; Network Protocols ;(DSDV) Destination Sequenced Distance Vector ; ARTSD (Alternative Routing Table Short Distance);Wireless Network; Mobile Ad Hoc Network

I. Introduction

A MANET is a compilation of wireless mobile nodes that form a network for sending and receiving data without the need to establish an infrastructure or a centralized administrator. Each mobile node will send a packet to its destination by demanding aid from other intermediate ones with regard to the radio propagation range for each node. In a MANET, there is no central infrastructure between nodes as there is in a wired network. This causes nodes to move frequently and instantaneously without notification, which results in various problems such as loss of connectivity, and an increased holding time to update routing tables when computing the new shortest path between source and destination. This problem leads to a rise loops in the network that degrades its performance and reduces throughput. When a loss in connectivity occurs, the routing protocol starts to flood extra data packets to inform other nodes on the topology to update the routing table and compute a new shortest path. Ad hoc wireless networks are frequently affected by failures when nodes move out and into radio propagation range. It is, therefore, highly desirable to develop a recovery mechanism to improve the quality of service (QoS) of the network. When mobile nodes move or fail, two case scenarios can arise. First, if a node on the primary path moves out of range, then other nodes on it lose their connectivity. Second, the routing protocol begins to re-establish a new path between source and destination. In the meantime, loss of data packets and end-to-end delays will increase. Many different types of routing protocols have been used to solve this routing problem, including DSDV, Dynamic Source Routing (DSR) and Optimized Link State Routing (OLSR) protocols In [1], [2], [3]. In a wired network, the routing protocol generally uses distance vectors or link state routing algorithms. Both are proactive mechanisms as they send extra messages to keep the nodes up-to-date in case any information on the network changes, such as if a node joins the network or fails. When failure occurs, these protocols inform all the nodes and they start to re-compute a new routing table. More holding time is then required in order to re-send the traffic along the new route. ARTSD re-computes an alternative routing table from its original one when nodes fail, or change their position and move out of range. The DSDV and LS (Link State) protocols demand that a routing advertisement be broadcast between the nodes on the network. In DSDV, when the nodes receive these advertisements, each one knows the route from its neighbor and its distance to all the other nodes on the network. On the other hand, OLSR protocols compute the shortest path based on the complete picture for each adjacent node on the network. The ARTSD mechanism aims to recover the network from failure in a shorter period 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 improve throughput in the network when nodes fail or move out of range. The pre-computed alternative path can be used immediately without waiting for the routing protocol to re-compute a new one. In this paper, we concentrate on the distance and propagation radio range between nodes to create a primary routing table which is computed by an ARTSD algorithm. The main contribution of this paper is the development of an alternative and fully disjointed pathway computed using a routing table, which is based upon the distance between nodes and their range.

This paper is organized as follows: Section II discusses related work, Section III illustrates the originality and the basic concept for the ARTSD algorithm in detail, Section IV shows the performance evaluation via simulation and Section V concludes how ARTSD can be improved and future work.

II. 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 [4]. Different routing protocols have been published for different environments to improve the network performance when connectivity is lost by nodes moving or failing [5]. In [6] 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 [7], [8]. 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 [9], [10]. 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 [11]. This is because they both use a similar algorithm.

Multipath routing has been introduced and considered in wired networks for improving their performance [12]. The proactive protocols are designed to send out extra messages to create a disjointed path from source to destination. [13], [14] presents a new algorithm called the Alternative Routing Table (ART) algorithm, which can send extra data packets to enquire if neighboring 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 [15], [16], [17], 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 [18], [19], 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. 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 [20]. Internet Protocol recovery emphasizes two cases. First, the time required to detect failure. Second, the time to compute the shortest path. In [21], 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 [22].

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 optimize along the network to invest in the provisioning of resources in cases of increased traffic or node failures [24], [23].

III. PROPOSITION

The basic principle of the ARTSD protocol is how it finds a backup path from the original routing table by degrading all nodes on the primary path, which leads to reduced calculation time to find an alternative path between source and destination. The ARTSD 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 pre-computed in advance in order to re-route data packets in case of failure. In addition, when a node on the primary path goes down or 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 ARTSD protocol, via the backup routing table, will pass traffic to destination D without waiting to re-compute a new routing table. The ARTSD 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 ARTSD protocol involves choosing one of them as a backup node to re-route a packet through it when a failure occurs. Hence, the ARTSD 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. They will reply by an acknowledgement, which will contain the answer ”Yes, I have a different route and I can be a backup node in case of failure” or ”No, I cannot be a backup adjacent to the destination”. When the source node checks the answer and it is ”Yes”, it will add this node to the new routing table as a backup. If the answer is ”No”, the node will check answers from other adjacent nodes. Hereafter, if the answer from all adjacent nodes is ”No”, then the ARTSD algorithm will send another packet to ask the adjacent node if it has a neighbor node which is unconnected to the primary path with a disjointed route to the destination (not via the primary path). If the answer is ”Yes”, then the adjacent node will send an acknowledgement that contains ”Yes, I can be a backup node via my adjacent node”. When the source node receives this ”Yes”, it will add 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 ARTSD 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 performed different mesh topologies, which act as a good examples, making each node have at least two neighbors that can re-route data packets through either of them when failure occurs.

A. Algorithms Overview

In network communications, using pre-computed second paths provides a solution for when failures occur. ARTSD evaluates the backup routing table from the primary one to ensure that the alternative path will be disjointed from the primary one. Figure 1 shows the flowchart for an ARTSD algorithm and the way it computes the backup routing table from the passing traffic. Algorithm 1 starts to compute the distance between the nodes to discover all adjacent nodes within the radio propagation range (assumed to be 250m) and then selects a route to the destination based upon the number of hops. If the distance between any two nodes is less than 250m, then the ARTSD algorithm will select this node’s adjacent node. When there is more than one adjacent node, then ARTSD will start to check which adjacent node has a disjointed path to the destination (to construct a new primary path). In addition, the ARTSD algorithm evaluates an appropriate backup route using the primary routing table from source to destination. The nodes connected to the primary path will be excluded from the next hop in the backup one. Each node takes a position (X, Y) randomly under the radio propagation range for the area 1000 by 800. In algorithm 2, the nodes on the topology will start broadcasting a small message to enquire from adjacent nodes if they have a disjointed path from the primary one with regard to the primary routing table to the destination , all nodes will check the acknowledgement answers from adjacent nodes to see if there is any available route to the destination, and if so it will then insert this node to the backup routing table as a first hop. If no adjacent node has a disjointed path, then each adjacent node asks their neighboring node if they have a disjointed route to the destination. Hence, the adjacent node sends an acknowledgement that informs the source whether a disjointed path is available. Hereafter, the source node uses the node in the backup routing as a first next hop to re-route the traffic through the backup routing table should a node on the primary path fail.

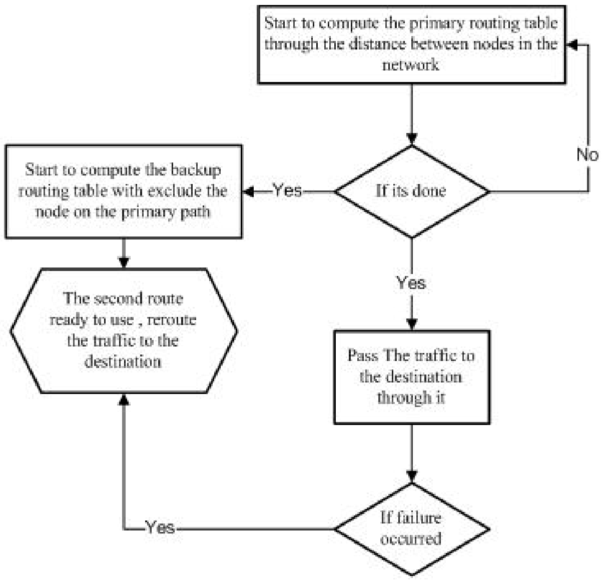

Fig. 1. The flow chart to compute primary and backup routing table

	0	1	2	D(3)
S(0)	–	1	➔ 2	·····▶ 1
1	0	–	0	·····▶ 3
2	0	0	–	➔3
3	1	1	2	–

TABLE I
Primary Routing Table

Algorithm 1 *ComputePrimaryRoutingTable* returns a set of shortest distance path for every node on the network

Algorithm 1 executes the following steps to build the routing table:

- Set small Distance = ∞
- Compute Propagation Radio Range as follow:

$$P_r = P_t G_t G_r \left(\frac{h_t^2 h_r^2}{d^4}\right) \quad (1)$$

Where *Gt* is the transmitter antenna gain; *Gr* is the receiver antenna gain; d is the distance between the antennas in meters; *ht* is the height of the transmitter antenna; *hr* is the height of the receiver antenna. Configure transmitting power Pt is 0.28 with range distance 250m.

- Compute the distance between nodes.
- X, Y the coordinate position for the nodes in net- work model which are allocated randomly.

$$d = \sqrt{(x2 - x1)^2 + (y2 - y1)^2} \quad (2)$$

- if (Dist < 250) insert node in the adjacent nodes to the list of adjacent for each node on the topology.
- Compare the adjacent nodes which one has the smallest distance to be the first next hop in the primary routing table.
- Compute the path P with the minimum hops to the destination (applying the ARTSD algorithm).

In algorithm 2, if more than one alternative path exists, then the node will select the best one based upon the number of hops. The ARTSD illustrates this in algorithm {1,2}. In addition, Table I illustrates how the adjacent node has an alternative node to the destination with a disjointed path from the primary path. The dash arrow shows the primary path from source to the destination as follows:
S(0)->1->D(3). When node (0) enquires from an adjacent node (2) if there is an alternative path to the destination. The solid arrow shows node (2) has an alternative path as follows: S(0)->2->D(3). Hence, node (2) will reply with an acknowledgement to node (0) that it can be a backup next hop in the routing table, then node(0) will insert this route as a backup path to pass the traffic in case node (1) fails or moves.

B. Example

The ARTSD algorithm is illustrated in Figure 2. Each node sends a small packet after the main routing table has been completed. {S, Y, G, D} start to create a new routing table. Node {S} sends data packets to the adjacent nodes {Y and M}, excluding node {N} because it is the first hop on the primary path to the destination {D}. Node {Y} will check from the routing table if there is any available disjointed path to the destination node {D}not via{S,N,Q}. Hence, {Y} node checks the route from its neighbors to the destination {D}. Node {Y} will send an acknowledgement to inform {S} that it can act as a backup in case node { N } moves or goes down.

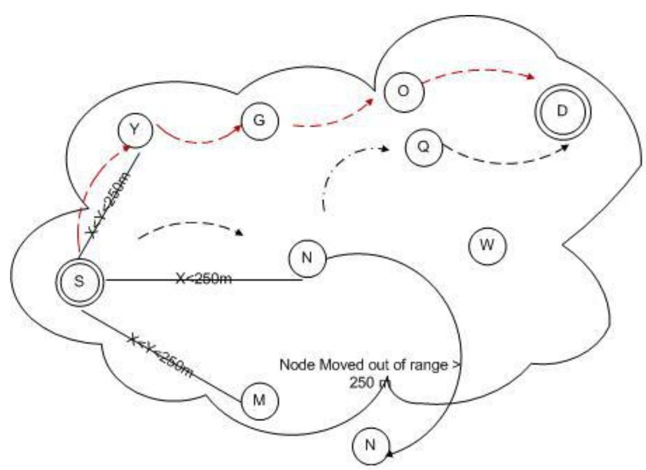

Fig. 2. Reroute the traffic when primary path fail

In Figure 2, node {M} cannot be a backup because there is a common node in its path to the destination, which is node {N}. Therefore, the source node checks answers from all other adjacent nodes. Node {M} will send an acknowledgement to inform the source node that there is no disjointed path to the destination {D}.

On the other hand, when the source node receives an answer from node {Y}, which has a disjointed path, then it will add itself into the backup routing table as a first next hop in case of failure. The ARTSD algorithm works on all nodes in the network to find all possible disjointed paths in the network topology. In addition, the ARTSD algorithm deals with all nodes on the topology as being both source and destination nodes.

Algorithm 2 *AlternativePath* returns a set of alternative paths for every possible path in the routing table.

G(V, E) is an oriented graph with two sets, a set of vertices V and a set of edges E, where an edge e = (v, u), e ∈ E, v, u ∈ V is a connection from vertex v to vertex u. A path P is a set if edges $e_1, e_2, ..., e_n$, such that if v, u, x ∈ V, then $e_i = (v, x)$, $e_{i+1} = (x, u)$, for all $1 \leq i \leq n-1$.

procedure *AlternativePath*(T_r)
T_r: The routing table
V: The vertex set in graph $G(V, E)$
$\Gamma(v)$: The set of adjacent vertices to a vertex v
$P_r(T_r, s, d)$: The path connecting the vertex s to d as in T_r
$P_a(s, d)$: An alternative path such that $P_r(T_r, s, d) \cap P_a(s, d) = \varnothing$
S_P: The set of all generated alternative paths
q_{sub}: A path
Q: A queue of couple (path, vertex)
Enqueue: Insert an element in a queue
Dequeue: Removes an element from a queue
Front: The element at the front of a queue
$S_P \leftarrow \varnothing$
for all $s \in V$ **do**
  for all $d \in V$ **do**
    if $s \neq d$ **then**
      $q_{sub} \leftarrow \varnothing$
      $Q \leftarrow \varnothing$
      $Enqueue(Q, (q_{sub}, s))$
      while $Q \neq \varnothing$ **and** $p_a(s, d) = \varnothing$ **do**
        $(q_{sub}, x) \leftarrow Front(Q)$
        for all $k \in \Gamma(x)$ **do**
          $e \leftarrow (x, k)$
          if $(q_{sub} \cup e) \cap P_r(T_r, s, d) = \varnothing$ **then**
            if $P_r(T_r, k, d) \cap P_r(T_r, s, d) = \varnothing$ **then**
              $p_a(s, d) \leftarrow q_{sub} \cup e \cup P_r(T_r, k, d)$
              $S_P \leftarrow S_P \cup p_a(s, d)$
              break
            else
              $Enqueue(Q, (q_{sub} \cup e, k))$
            end if
          end if
        end for
        $Dequeue(Q)$
      end while
      $Q \leftarrow \varnothing$
    end if
  end for
end for

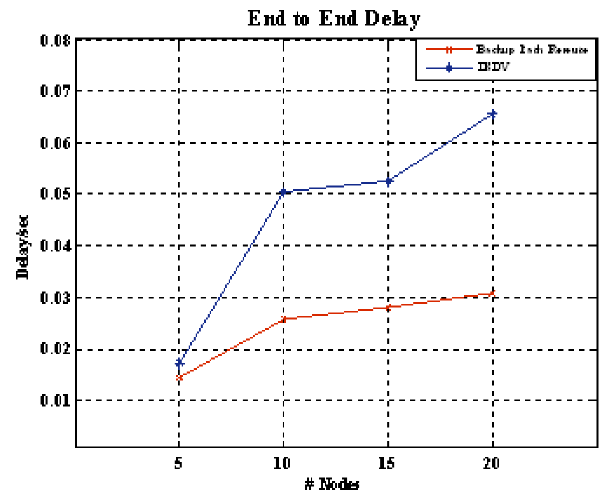

Fig. 3. End to End Delay

hand, when the source node receives an answer from node {Y}, which has a disjointed path, then it will add itself into the backup routing table as a first next hop in case of failure. The ARTSD algorithm works on all nodes in the network to find all possible disjointed paths in the network topology. In addition, the ARTSD algorithm deals with all nodes on the topology as being both source and destination nodes.

IV. Simulation Experiment

A. Simulation Environment

Network simulation (NS2) was performed to evaluate the proposed algorithm. 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. A mobile ad hoc network performs as follows: different network scenarios were created, each having a different number of mobile nodes to demonstrate the effect of node movement or failure during simulation time. We configured an area of 1000 m by 800 m. Failure could occur randomly by any node on the primary path. A radio propagation range with transmission power of 0.28 watt was used, allowing each node to send or receive a packet to or from its neighbors for a distance of up to 250 m. For each scenario, the simulation time ran for 250 seconds. In the free space model, the signal power became weak, as $1/d^2$ where d is the distance between radios. The movement type was of a Two Way Ground Model with a channel capacity of 2Mb/s. The packet size was 512 bytes. We compared ARTSD with DSDV because they are both proactive protocols. We used an IEEE 802.11 Distributed Coordination Function (DCF), as the wireless channel can share in an ad hoc configuration.

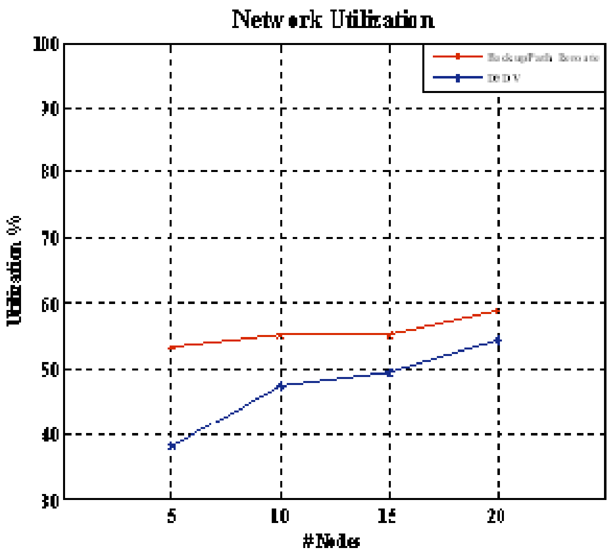

Fig. 4. Utilization

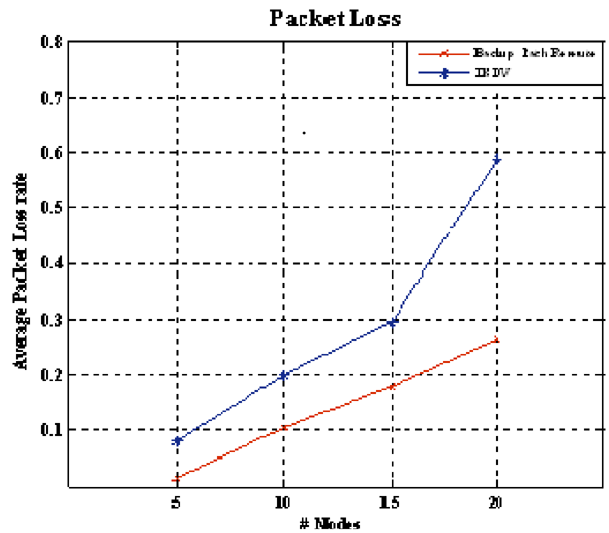

Fig. 5. Loss Of packets

B. Results and Analysis

Figure 3 shows the results for the end-to-end delay measured against a different number of mobility nodes. The end-to-end delay in ARTSD protocol was less than the DSDV protocol, because the latter has to retain all its routes in a static routing table. When any node on the primary path moved out of range or failed, the DSDV routing protocol needed to re-compute a new routing table. In ARTSD, 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. Figure 4 shows that the amount of the network resources that is utilized by DSDV is less compared to ARTSD, which is because ARTSD needs to generate extra data packets so that it can construct the new backup routing table. These packets show that increased utilization of the network does not degrade network performance.

Figure 5 shows the average loss of data packets for DSDV and ARTSD for different number of nodes in different scenarios. Traffic is re-routed along an alternate path, which is computed by the ARTSD protocol. The DSDV protocol shows that 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 that offer the greatest probability for collision to occur in the network.

Figure 6 shows the throughput for both ARTSD and DSDV protocols. The data packets are sent from their source to their destination hop-by-hop. The ARTSD protocol achieved a higher throughput compared to the DSDV. When a collision occurred, through- put was decreased in both. The IEEE 802.11 sending RTS packets can reduce collisions in the network.

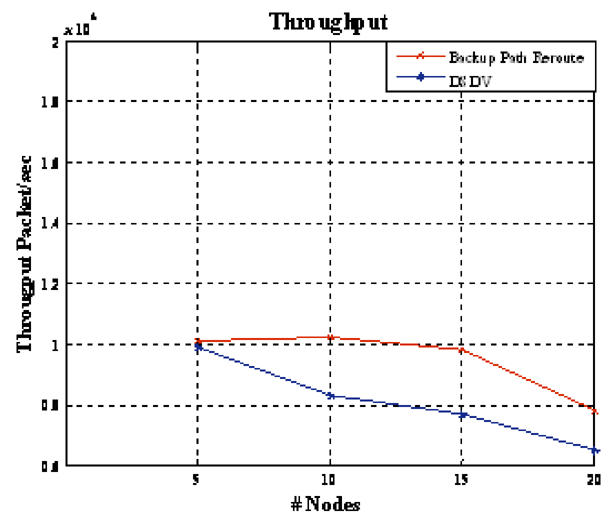

Fig. 6. Throughput

However, the DSDV protocol needs a period of time to re-route the traffic. The length of this period remains undesirable. It takes two seconds for each node to re-compute a new routing table and a medium topology will take fifteen seconds. However, ARTSD 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, ARTSD 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 an increased throughput between the source and destination.

V. Conclusions

This paper has presented a new protocol for computing an alternative backup routing table which finds alternative path for each node on the network. The ARTSD algorithm computes a backup routing table based on the distance between nodes and number of hops to the destination. The backup routing table includes alternative and disjointed next hop to destination for all nodes on the topology. The ARTSD protocol provides a recovery path that gives the shortest distance between source and destination. We have shown that the backup paths contain fewer numbers of hops compared with those produced by the DSDV protocol. For real traffic, the results show that the ARTSD protocol reduces the loss of data packets and delay between source and destination nodes. In future work, the ARTSD protocol will be designed to deliver a backup routing table that contains more than one backup path between the source and destination, in order to improve the QoS when more than one node moves or fails.

References

[1] C.E Perkins and P.Hhagwat. Highly dynamic destination sequence vector routing (dsdv) for mobile computers. pages 234–244. Computer Communication, 1994.

[2] Mohanchur Sarkar, K.K.Shukla, and K.S.Dasgupta. Article:performance analysis of proactive and reactive transport protocols using a proactive congestion avoidance model. *International Journal of Computer Applications*, 6(5):10–17, September 2010. Published By Foundation of Computer Science.

[3] Sung ju Lee and Mario Gerla. Split multipath routing with maximally disjoint paths in ad hoc networks, 2001.

[4] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (aodv) routing protocol. IETF Internet RFC 3561,2003.

[5] Saeed Shahbaz, Gholamreza Ghassem-Sani, Hamidreza Rabiee, and Mohammad Ghanbari. A distributed intelligent ad-hoc network. volume 4308–2006. SpringerLink, 2006.

[6] D. Kim, J. Garcia, and K. Obraczka. Routing mechanisms for mobile ad hoc networks based on the energy drain rate. volume 2, pages 161–173. IEEE Transactions on Mobile Computing, 2003.

[7] Poonam, K. Garg, and M. Misra. Trust enhanced secure multipath dsr routing. *International Journal of Computer Applications*, 2(2):63–69, May 2010. Published By Foundation of Computer Science.

[8] R. Rashidi, M.A.J. Jamali, A. Salmasi, and R. Tati. Trust routing protocol based on congestion control in manet. In *Application of Information and Communication Technologies AICT International Conference*, pages 1–5, 2009.

[9] V.D. Park and M.S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. pages 1405–1413. IEEE International Conference on Computer Communications (INFO- COM, 1997.

[10] J. Moy. Link-state routing in routing in communications networks. pages 135 – 157. M.E. Steenstrup, Prentice Halls, 1995.

[11] Subir Kumar Sarkar T.G. Basavaraju and C Puttamadappa. Impact of mac layer on the performance of routing protocols in mobile ad hoc networks. volume 3:8, pages 541–548. International Journal of Information and Communication Engineering, 2007.

[12] S.-J. Lee and M. Gerla. Aodv-br: backup routing in ad hoc networks. In *Wireless Communications and Networking Conference,2000. WCNC. 2000 IEEE*, volume 3, pages 1311 –1316 vol.3, 2000.

[13] Radwan Abujassar and Mohammed Ghanbari. An efficient algorithm to create a loop free backup routing table. pages 148–152.International Conference on Information and Computer NetworksICICN, 2011.

[14] Marti S., Giuli T. J., and Baker. Mitigating routing misbehavior in mobile ad hoc networks. pages 255–265. In Proceeding of Sixth Annual International Conference Mobile Computing and Networking (MobiCom) New York, 2000.

[15] Prayag Narula, Sanjay Kumar Dhurandher, Sudip Misra, and Isaac Woungang. Security in mobile ad-hoc networks using soft encryption and trust-based multi-path routing. *Comput. Commun.*, 31:760–769, March 2008.

[16] Juan Antonio Cordero Emmanuel Baccelli. Ospf over multi-hop ad hoc wireless communications. *IJCNC*, 2(5):37–56, September2010.

[17] C. A. Dhote, M.A.Pund, R.S. Mangrulkar, and Makarand R. Shahade. Article: Hybrid routing protocol with broadcast reply for mobile ad hoc network. *International Journal of Computer Ap- plications*, 1(10):108–113, February 2010. Published By Foundation of Computer Science.

[18] H. Yong, C. Chuanhe and Wenming. Trusted dynamic source routing protocol. pages 1632–1636. Wireless Communications, Networking and Mobile Computing WICOM, 2007.

[19] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks mobile computing. pages 153 – 181. Kluwer Academic Publishers, 1996.

[20] Broch J., Maltz D. A., Johnson D. B., Hu Y.C., and Jetcheva. A performance comparison of multihop wireless ad hoc network routing protocols. pages 85–97. proceeding of International Con- ference Mobile Computing and Networking (MobiCom) ACM Press, 1998.

[21] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. pages 90–100. Workshop Mobile Computing Systems and Applications (WMCSA), 1999.

[22] A. Veres et al. Supporting service differentiation in wireless packet networks using distributed control. IEEE, 2001.

[23] E. M. Royer and C.-K. Toh. A review of current routing protocols for ad-hoc mobile wireless networks. pages 46–55. IEEE Pers, 2001.

[24] W. H. Liao et al. A multi-path qos routing protocol in a wireless mobile ad hoc network. IEEE ICN, 2001.