Link-and Node-Disjoint Evaluation of the Ad Hoc on Demand Multi-path Distance Vector (AOMDV) Routing Protocol in Wireless Sensor Networks

Koffka Khan
Department of Computing and Information Technology The University of the West Indies, Trinidad and Tobago, W.I
Email: koffka.khan@gmail.com
Wayne Goodridge
Department of Computing and Information Technology The University of the West Indies, Trinidad and Tobago, W.I
Email: wayne.goodridge@sta.uwi.edu.com

ABSTRACT

This work illustrates the AOMDV routing protocol. Its ancestor, the AODV routing protocol is also described. This tutorial demonstrates how forward and reverse paths are created by the AOMDV routing protocol. Loop free paths formulation is described, together with node and link disjoint paths. Finally, the performance of the AOMDV routing protocol is investigated along link and node disjoint paths. The WSN with the AOMDV routing protocol using link disjoint paths is better than the WSN with the AOMDV routing protocol using node disjoint paths for energy consumption.

Keywords - **AOMDV, AODV, paths, loop-free, node disjoint, link disjoint, multi-criteria, AOMDV.**

Date of Submission: May 20, 2018 Date of Acceptance: June 23, 2018

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are densely packed networks with motes very close to each other. This ad hoc arrangement of motes means that there exist many pathways for communication to take place between the sensing source motes and the target destination sink mote. It is the job of the routing protocol to find these paths. Typical on-demand routing protocols, for example, (Ad hoc On-demand Distance Vector) AODV and Dynamic Source Routing (DSR) proposed for use in WSNs consume too much energy and there are no fault tolerance guarantees. One attempt at reducing energy costs and providing these fault tolerance guarantees is the use of multipath routing protocols, for example, Ad hoc On-demand Multipath Distance Vector (AOMDV) and Multi-Path Dynamic Source Routing (MP-DSR). However, high energy communication costs still remain a problem and there is still no evidence that Alternate Path Routing (APR is one type of multipath routing) provides fault tolerance guarantees to WSNs.

In WSNs the routing protocol usually finds a path on which to send packets to the sink. "Usually at the network layer, routing algorithms have the job of selecting paths to a desired destination that consumes minimal energy" [15]. There are numerous routing protocols available which aim to find these paths. In full multipath routing protocols data is sent along all paths so there is no dilemma in path selection. However, the work in this paper is on APR which does involve the selection of one path on which to send data. Usually path selection is based on a single criterion or metric which is the number of hops to reach the required destination. Generally, the shortest path is chosen when selecting a path to send data to the sink mote. The idea of selecting among different criteria has been explored using WSNs, but the focus of this paper in the context of a multiple metric path selection algorithm for WSNs using the AOMDV routing protocol along node or link disjoint paths has never been explored.

The network layer [19] hosts a WSN mote's routing protocols. The routing protocol aids network communication by setting up paths within the network. AOMDV's ancestor is the Ad hoc On-demand Distance Vector (AODV) routing protocol. These protocols set up paths to a destination on request by the source node. Hence, they are on-demand routing protocols. The paths are kept in a mote's routing tables and updated whenever a path fails. For the majority of routing protocols, the shortest path metric to the destination mote is used to determine 'best' paths to a given sink. However, this method does not always select optimal paths. This is since in context, specific requirements are accentuated in some networks.

Single path on-demand routing protocols are too costly in terms of adding towards WSN energy usage [1] [16] [6]. "Multipath routing protocols saves more energy than their single path counterparts because there are less route discovery calls, which can save as much as 30% more energy during normal network conditions" [2]. "Multipath routing protocols can be node-disjoint or link-disjoint, if a node or a link cannot participate in more than one path

Volume: 10 Issue: 01 Pages: 3674-3686 (2018) ISSN: 0975-0290

between two end motes” [9]. There exist many algorithms in the literature which attempt to derive disjoint paths in networks [3] [5] [18] [17]. Some produces only node or link disjoint paths, while others produce both.

Important questions arise when sending data along either node or link disjoint paths with some hybrid schemes sending data along both simultaneously. For example, do bottlenecks occur on shared link disjoint portions? In this case it may be preferable to have longer paths if the resulting bottlenecks are not shared. How to select the best paths to use is complex? Further, research [10] has shown that node or link disjoint path selection may be more important in path selection than path length.

The rest of paper is organized as follows. We summarize the previous work in Section 2. Section 3 contains system model description and assumptions. In Sec. 4, the proposed data collection algorithm is introduced. In Sec. 5, we evaluate performance of the proposed algorithm by simulation. The paper concludes in Section 6.

II. AN AD HOC ON-DEMAND DISTANCE VECTOR (AODV) ROUTING PROTOCOL

AODV [13] is a pure on-demand routing protocol where creation of routes happens only when desired by the source node. A unicast route is a route from a source node to a destination node. AODV has two phases, (1) Route establishment or discovery and (2) Route maintenance. A node does not perform route discovery or maintenance until it needs a route to another node or it offers its services as an intermediate node. The Route Discovery Process is completed when a route is found, and all possible routes have been examined. The AODV routing protocol uses a broadcast route discovery mechanism with hop-by-hop routing from each network mote to the next [12] [14] [7]. Sequence numbers are assigned to routes and routing table entries to supersede stale cached routing entries. Every node maintains two counters, the node sequence number and broadcast ID.

Route Request (RREQ) messages are generated when node S wants to send a message to node D (see Figure 1). It shows the RREQ packet is on its’ last hop transmission to reach the sink or destination mote. It is important to note that a reverse path is formed on the transmission of each RREQ along a hop. This is shown by the dotted arrow from mote T to mote S and from mote V to mote T. Eventually when the RREQ packet reaches the sink mote a complete path would be known by the sink mote to the source mote. Therefore, after the final hop the complete forward path will be known by the sink mote, D.
S searches its route table for a route to D. If there is no route, S initiates a RREQ message with the following components (1) IP addresses of S and D, (2) current sequence number of S and the last known sequence number of D, (3) broadcast ID from S (broadcast ID is incremented each time S sends a RREQ message). The <broadcast ID, IP address> pair of the source S forms a unique identifier for the RREQ. Suppose a node P receives the RREQ from S. P first checks whether it has received this RREQ before. Each node stores the <broadcast ID, IPaddress> pairs for all the recent RREQs it has received.
If P has seen this RREQ from S already, P discards the RREQ. Otherwise, P processes the RREQ where (1) P sets up a reverse route entry in its route table for the source S and (2) this entry contains the IP address, current sequence number of S, number of hops to S and the address of the neighbor from whom P got the RREQ.

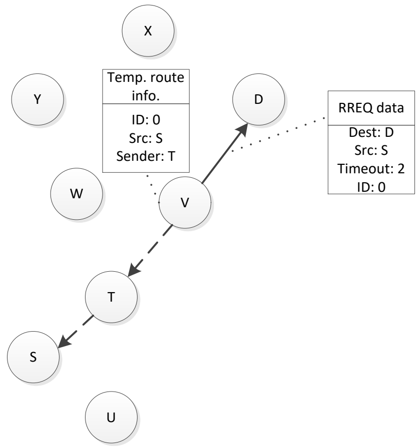

Fig. 1. A RREQ Packet: Formation of a Forward Path

Route Reply (RREP) messages are generated by the destination node in response to the first RREQ (see Figure 2). At this stage in the route discovery process a single forward path is formed, which is indicated by the dashed arrow from mote V to D and T to V. On the final hop the complete forward path will be known by the source mote, S. The destination then unicasts this RREP to the source node with the <source, destination> pairs reversed. It is now the source node.

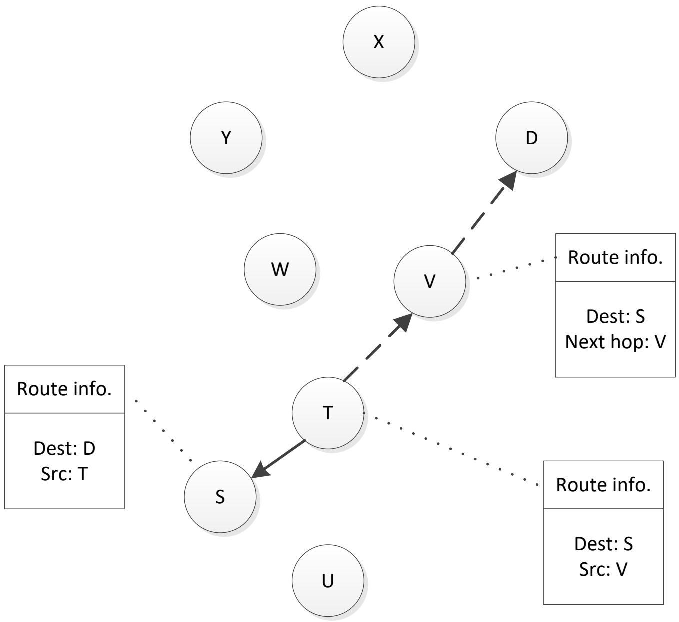

Fig. 2. A RREP Packet: Formation of a Backward Path.

An intermediate node P may receive more than one RREP for the same RREQ. P forwards the first RREP it receives and forwards a second RREP later only if (1) the later RREP contains a greater sequence number for the destination, or (2) the hop-count to the destination is smaller in the later RREP. Otherwise, it does not forward the later RREPs. This reduces the number of RREPs propagating towards the source. RREP eventually makes it to the source (see Figure 2), which can use the neighbor sending the RREP as its next hop for sending to the destination.

If a link breaks down the intermediate node tries to perform a local repair to the needed destinations. Also, a Route Error message is sent to upstream neighbors, which lists all the destinations which are now unreachable and a "DestCount" field is used to indicate the number of unreachable destinations. When a node receives a RERR it has several possible actions to perform, (1) it checks whether the sender is its next hop towards the destination, (2) deletes or invalidates the route towards the destination if needed, (3) forwards the RERR upstream if needed or (4) rediscovers route if still needed.

Route changes can be detected by (1) failure of periodic HELLO packets, (2) failure or disconnect indication from the link level or (3) failure of transmission of a packet to the next hop (can detect by listening for the retransmission if it is not the final destination). The upstream (toward the source) node detecting a failure propagates a route error (RERR) packet to the source node with a new destination sequence number and a hop count of infinity (unreachable).

The source (or another node on the path) can rebuild a path by sending a RREQ packet. Maintenance of a Route is needed only until the destination becomes inaccessible along every path from the source or until the route is no longer desired. Nodes that are not on active paths do not maintain routing information and do not participate in routing table exchanges. Routes are based on dynamic table entries maintained at intermediate nodes. Local HELLO messages are used to determine local connectivity, which can reduce response time to routing requests and trigger updates when necessary. Once a unicast route has been established between two nodes S and D, it is maintained if S (source node) needs the route. If S moves during an active session, it can reinitiate route discovery to establish a new route to D. When D or an intermediate node moves, a route error (RERR) message is sent to S.

Advantages of AODV protocol include no central administrative system to handle routing, high scalability, need for broadcast is minimized, reduced control messages, quickly reacts to changes in the network, quick response to link breakage in active routes, loop free routes, prevents network flooding during discovery and repairs breaks in active routes locally instead of notifying source. However, one of the major disadvantages of AODV is the high latency due to the route discovery only being reactive.

III. An Ad Hoc On-Demand Multipath Distance Vector (AOMDV) Routing Protocol

Ad Hoc On-Demand Multipath Distance Vector (AOMDV) routing protocol [8] is an extension of the AODV routing protocol with the addition of enabling multiple paths to be found between a given source and destination node. Two of the major goals of AOMDV were to improve on the AODV protocol by answering the following questions: (1) In the AODV framework, how to compute multiple paths between source and destination during route discovery? and (2) How to do this with minimal additional overhead to the AODV framework? Like AODV, AOMDV has an on-demand flood-based route discovery mechanism, uses a distance vector routing algorithm and hop-by-hop routing (routing list is sorted based on hop count). There are route discovery and maintenance phases like AODV, but there are multiple paths per route discovery. The protocol ensures loop free paths similar to that of AODV, but an additional feature is the assurance that all paths found are disjoint. In the route maintenance phase AOMDV uses alternate routes on a route failure. New route discovery is only needed when all routes fail. This will result in a fewer number of 'overall' route discoveries and an advantage of having a reduction in delay and routing overhead for a given time segment. Major uses of AOMDV would be in MANETs (Mobile Ad hoc NETworks), similarly to AODV and more recently in VANETs (Vehicular Ad hoc NETworks). One drawback with AOMDV may be the fact that we are not sure if alternate path works.

Compared to other on demand multipath protocols AOMDV is unique as there is no high inter-nodal coordination overhead like Temporally-Ordered Routing Algorithm, TORA [11], alternative paths are disjoint, there is no use of source routing and minimum overhead is used to get alternative paths when compared to AODV with reuse of alternate path routing information. Loop freedom is enforced by the use of sequence numbers. Every node maintains a monotonically increasing sequence number for itself and separately maintains the highest known sequence numbers for each destination in the routing table (called "destination sequence numbers"). Destination sequence numbers are tagged on all routing messages, thus providing a mechanism to determine the relative freshness of two pieces of routing information generated by two different nodes for the same destination. The AOMDV protocol maintains an invariant, similarly to AODV that destination sequence numbers monotonically increase along a valid route, thus preventing routing loops. During route discovery a node can receive a routing update via a neighboring mote. The routing table structure of AODV (see Figure 3) contains the following fields: (destination, sequence number, hop count, next hop, timeout), while the AOMDV routing protocol contains a route list for each destination, which contains an additional last hop field:

Volume: 10 Issue: 01 Pages: 3674-3686 (2018) ISSN: 0975-0290

AOMDV (destination, sequence number, advertised hop count, route list(next hopm, …, next hopn, last hopm, …, last hopn, hop countm, …, hop countn, timeoutm, …, timeoutn).

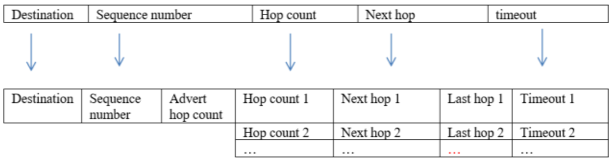

Fig. 3. Routing Table Structure: AODV and AOMDV Routing Protocols.

Route Request (RREQ) or Route Reply (RREP) packet either forms or updates a reverse or forward path. In AOMDV these routing discovery messages via a RREQ or RREP are referred to as "route advertisements." During route discovery the source mote that requires a path to a sink mote broadcasts a RREQ packet. This RREQ packet may reach the destination mote by traversing a sequence of multi-hops over one or more motes. At each wireless hop towards the destination the AOMDV routing protocol forms a series of backward pointers or paths which will form a reverse path (see Figure 5) to the source if the destination mote is reached. Indeed, the main result is the formation of reverse paths from the sink mote to the source mote. This reverse path formation is like the path left using bread crumbs in the Hansel and Gretel Fairytale [4]. If there are no intermediate motes with a path to the sink mote then the packet will eventually reach the sink mote. The sink mote in response to this RREQ packet will send a new RREP packet with the destination mote being the source mote. At each hop along the path back towards the source mote the forward path to the destination is recorded. The flowchart of the broadcast of the RREP packet is illustrated in Figure 6. Indeed, the main result is the formation of forward paths from the source mote to the sink mote.

AOMDV replaces the hop count variable with advertised hop count and for each route to a particular destination AOMDV stores a route list. The next hop variable is kept but for each route there may be multiple next hops or neighbors. Similarly, as with AODV hop count and timeout variables are kept for each route. The added field in AOMDV which may be repeated multiple times with different values is the last hop to the destination node. As was stated this is used in the computation of multiple alternate disjoint routes to the destination.

The following gives an example of how the protocol works:

Imagine node S wants to talk with node D. The network configuration is shown below (see Figure 4).

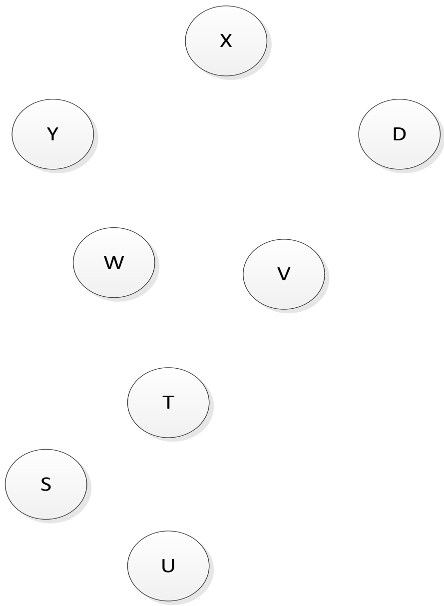

Fig. 4. Wireless Sensor Motes.

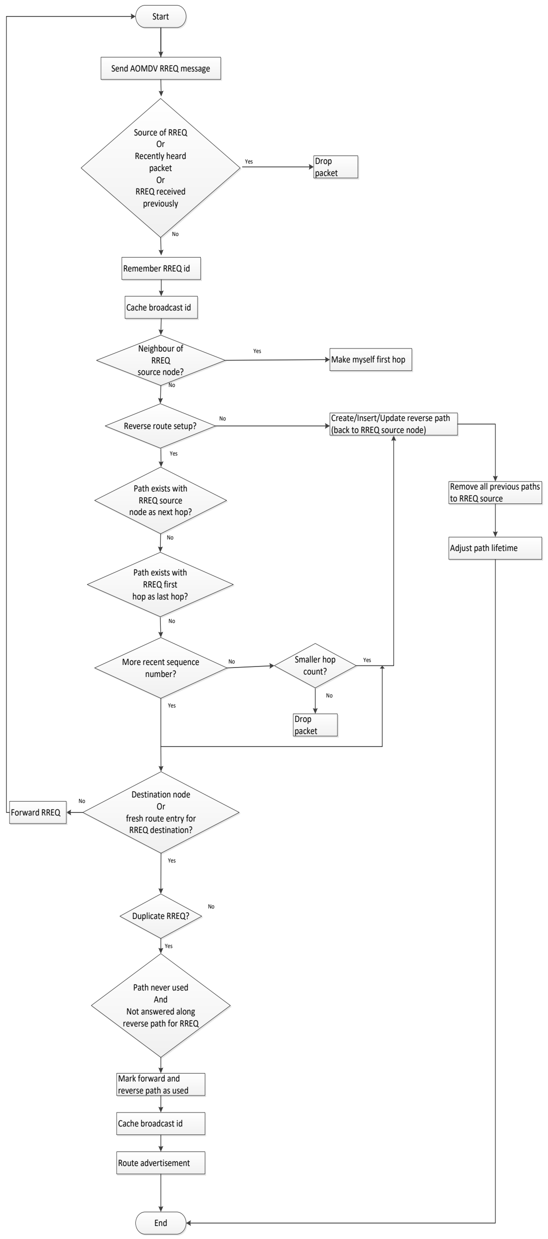

Fig. 5. AOMDV Forward Path Formation.

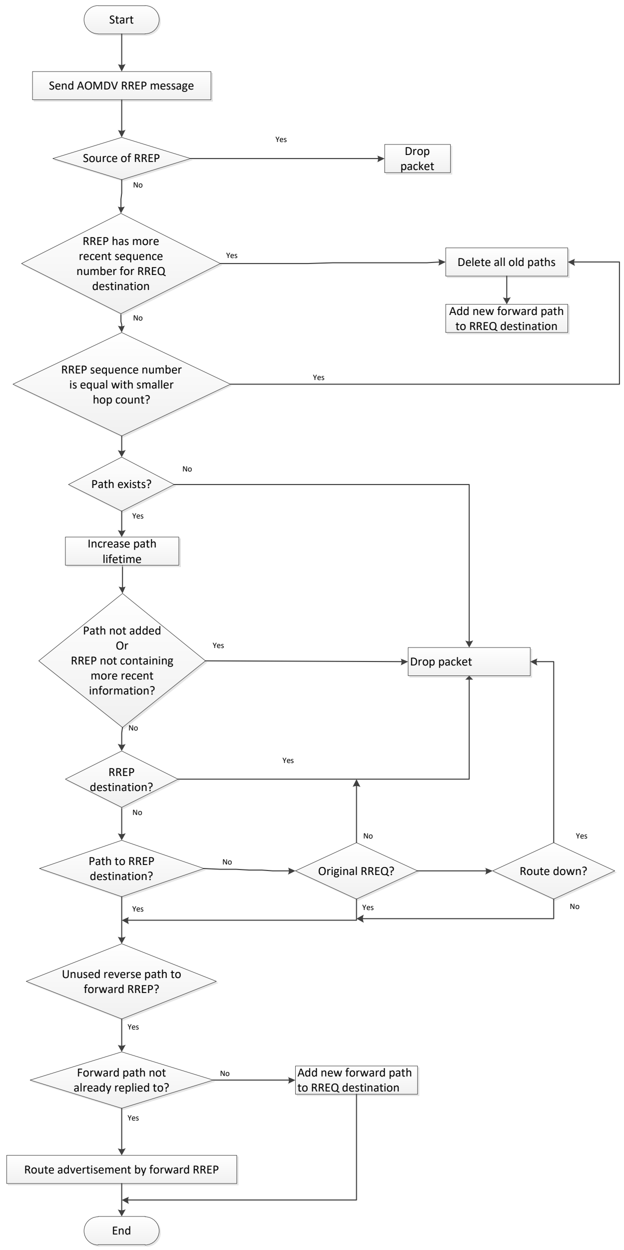

Fig. 6. AOMDV Reverse Path Formation.

Node S starts sending a Route Request Messages (RREQ) in broadcast (see Figure 7). These messages have an ID of the route query, the source and destination, and the maximum lifespan of the request.

Volume: 10 Issue: 01 Pages: 3674-3686 (2018) ISSN: 0975-0290

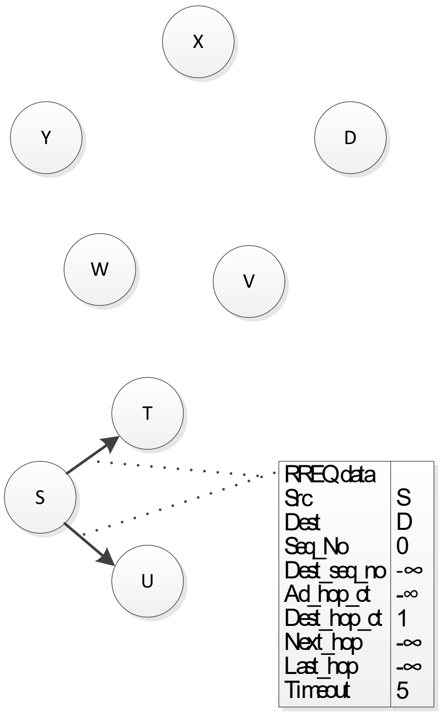

Fig. 7. Route Discovery: Forward Pass-Step 1.

When node V and W receive the RREQ message, they check if they have already received a RREQ query with that same Source and ID. As it is not the case of node V or W, they rebroadcast the RREQ (see Figure 8).

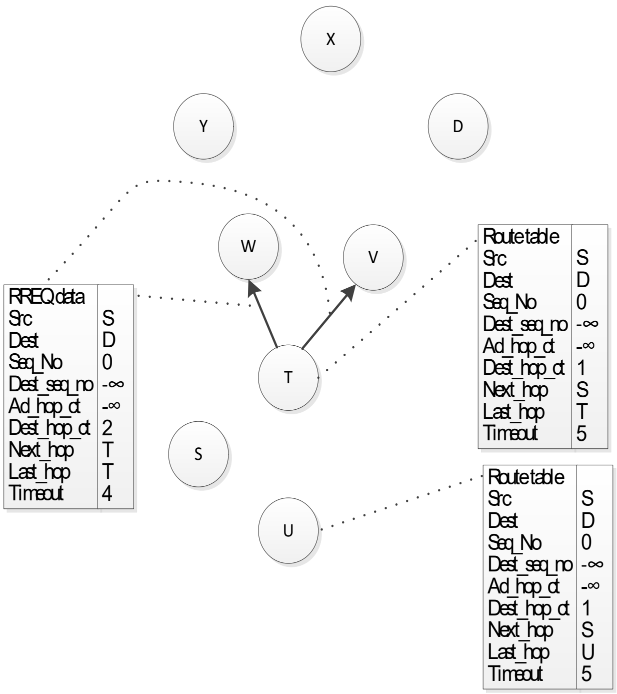

Fig. 8. Route Discovery: Forward Pass-Step 2.

When node V and W rebroadcast the RREQ, node S, V and W receive that message (see Figure 9). Node S, as it already knows that id/source simply ignores it. Node V and W are in the situation where it is the first time they receive a message with that source and id. So, they rebroadcast the RREQ. Nodes that have already received that RREQ, once again ignore it. Node D, the destination, receives the RREQ. It is now time to make the forward path using Route Reply Messages (RREP). This is done by looking at the temporary routing information tables that show the various hops performed to reaching the destination.

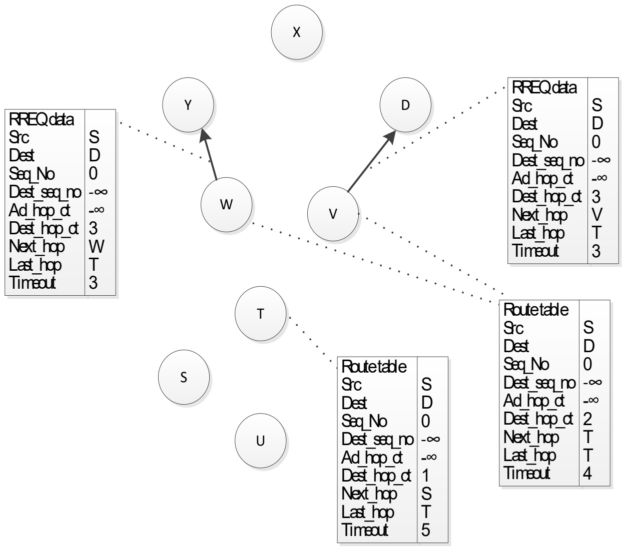

Fig. 9. Route Discovery: Forward Pass-Step 3.

Although the destination mote receives a RREQ message from one path (T-V-D), there is another path (T-W-Y-X) which is still transmitting messages (assuming a time-based system with no internal or external network disruptions). This is illustrated on Figure 10 and Figure 11.

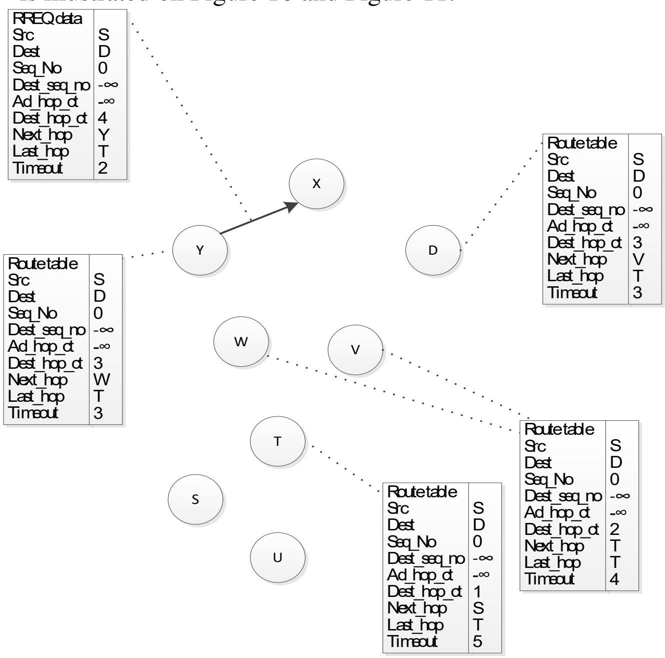

Fig. 10. Route Discovery: Forward Pass-Step 4.

Volume: 10 Issue: 01 Pages: 3674-3686 (2018) ISSN: 0975-0290

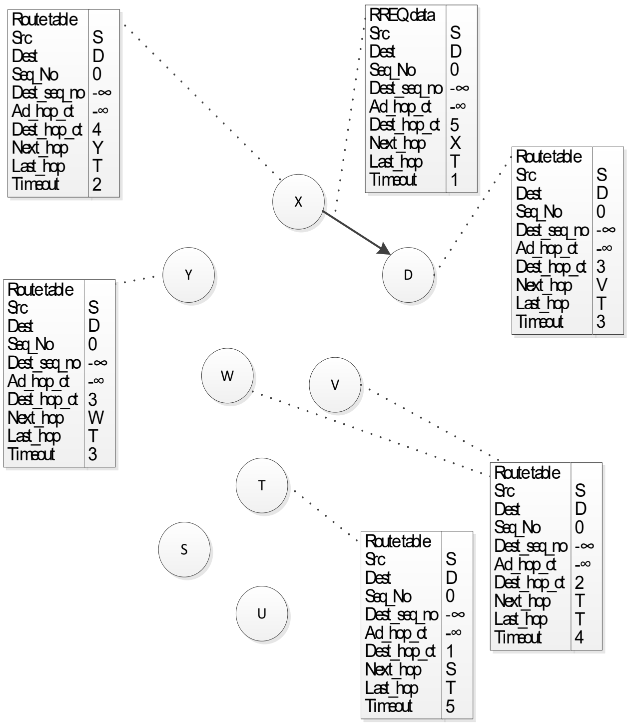

Fig. 11. Route Discovery: Forward Pass-Step 5.

The forward pass is now complete (see Figure 12) and all nodes have backward pointers to their previous hops.

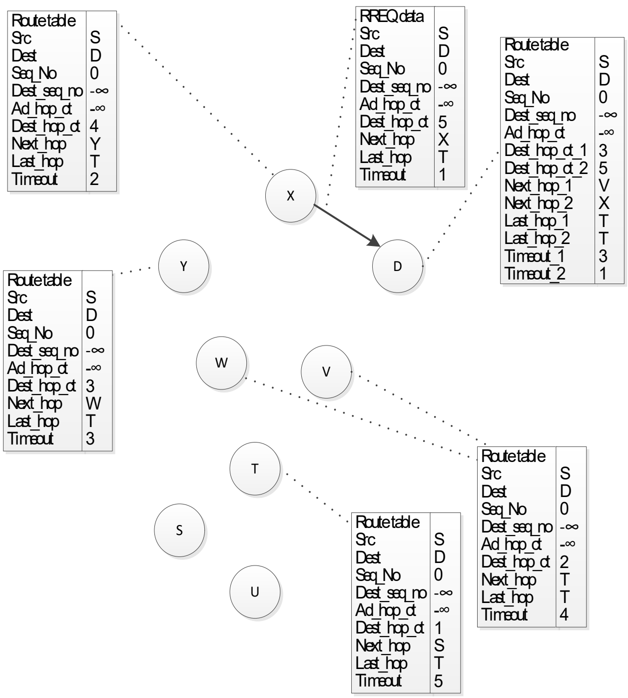

Fig. 12. Route Discovery: Forward Pass-Step 6.

The destination node now issues a RREP to the source (see Figure 13). To show the forward path setup a time-based system is used. The shorter path is thus first (D-V-T-S) is illustrated first.

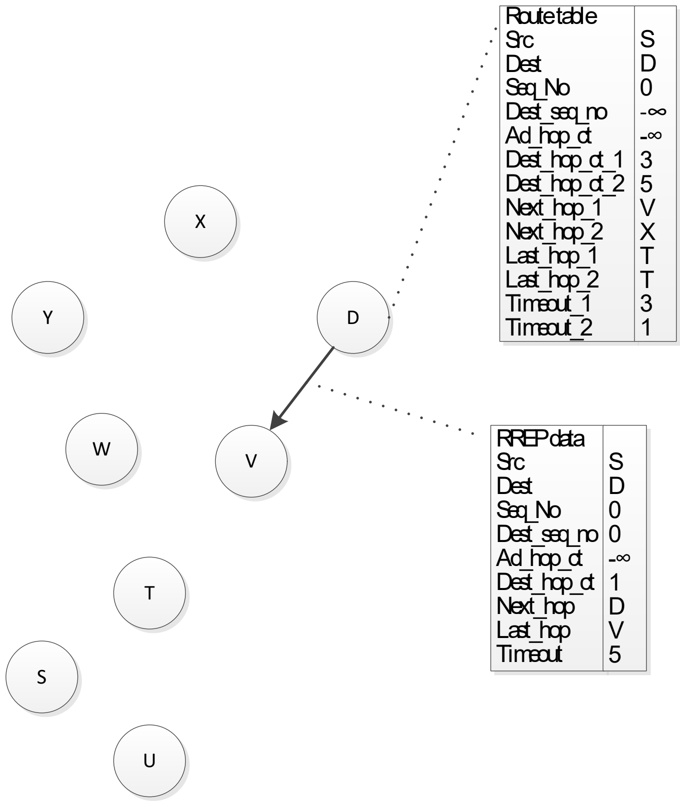

Fig. 13. Route Discovery: Backward Pass-Step 1.

Mote V updates its routing table with a pointer to the destination mote D, which is the source mote of the RREP packet (see Figure 14). A forward path pointer is formed.

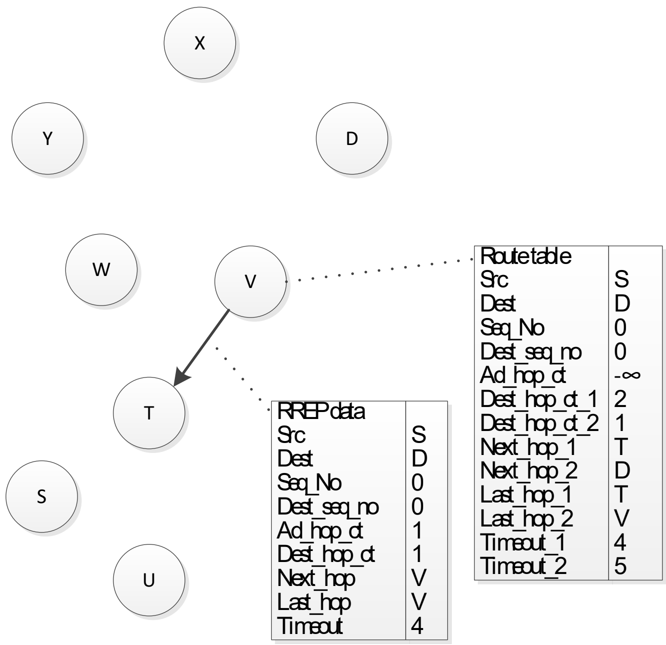

Fig. 14. Route Discovery: Backward Pass-Step 2.

Similarly, when mote T receives the packet it records its previous hop and forwards the packet (see Figure 15).

Volume: 10 Issue: 01 Pages: 3674-3686 (2018) ISSN: 0975-0290

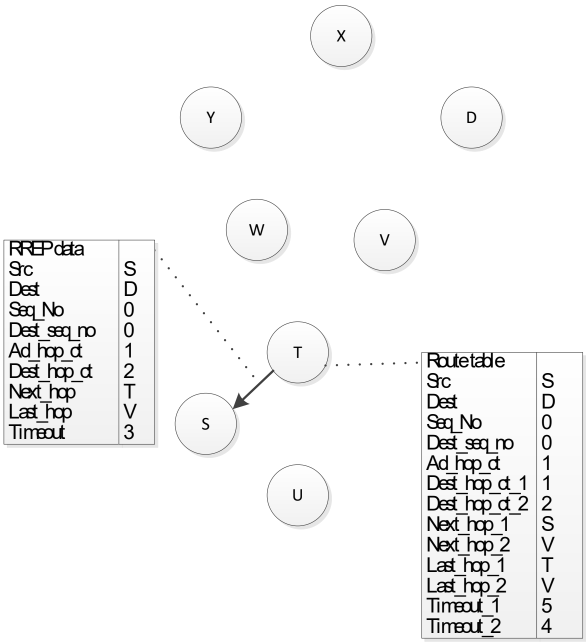

Fig. 15. Route Discovery: Backward Pass-Step 3

Node S upon receiving the RREP creates a new route entry, where he indicates that to messages with destination node S (see Figure 16). As he is the destination the process ends here.

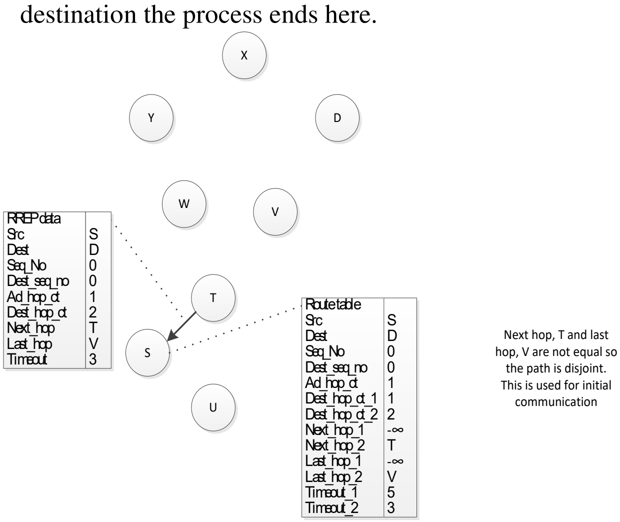

Fig. 16. Route Discovery: Backward Pass-Step 4.

The illustration now continues with the longer path now used. But at the same time (maybe this node got the message the same time as mote V) (see Figure 17):

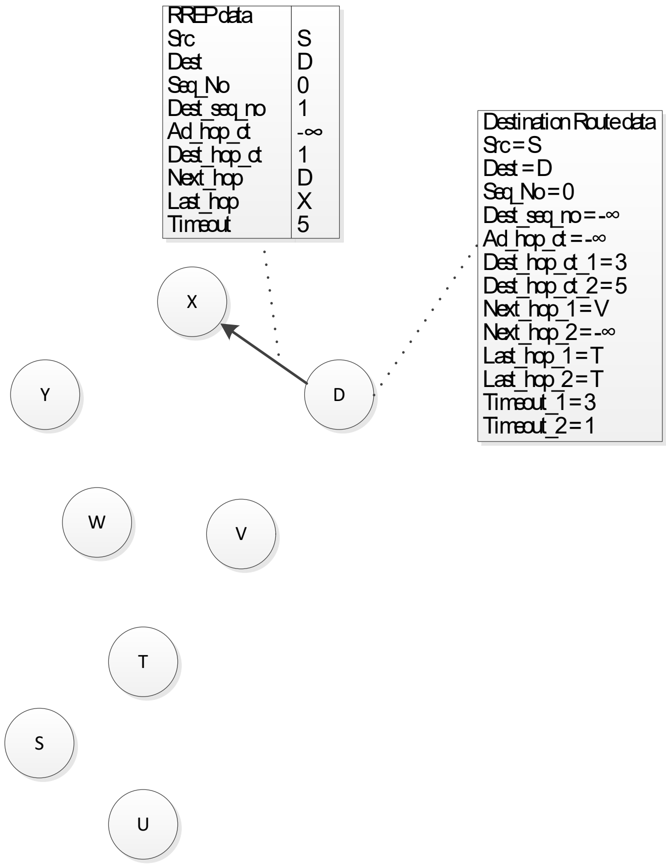

Fig. 17. Route Discovery: Backward Pass-Step 5.

Node X upon receiving the RREP creates a new route entry, where he indicates that to messages with destination node S, the next node he sends message is node D (because it is in his range) (see Figure 18). He then sends the RREP to the previous node present in his temporary routing information, node Y. This happens throughout the process and it is how the reverse paths are formed in the AOMDV routing protocol by the forward pass.

Volume: 10 Issue: 01 Pages: 3674-3686 (2018) ISSN: 0975-0290

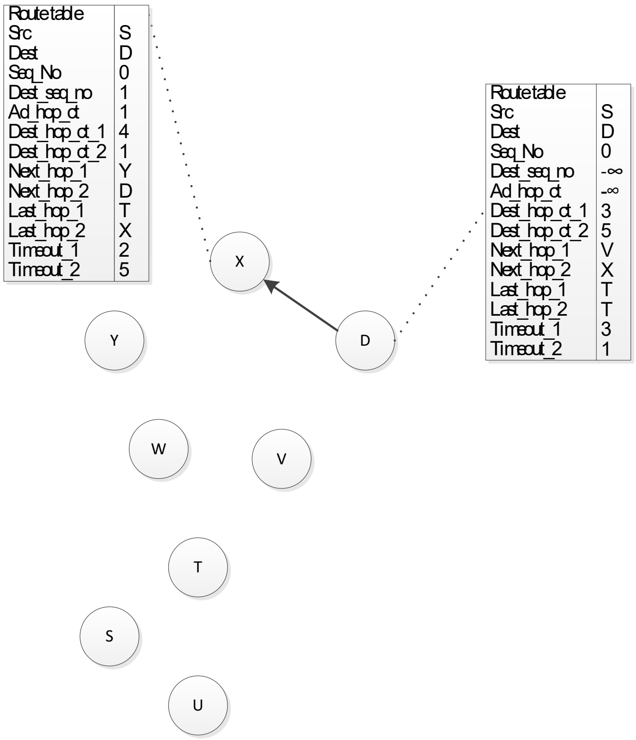

Fig. 18. Route Discovery: Backward Pass-Step 6

Mote X sends the RREP packet to mote Y (see Figure 19).

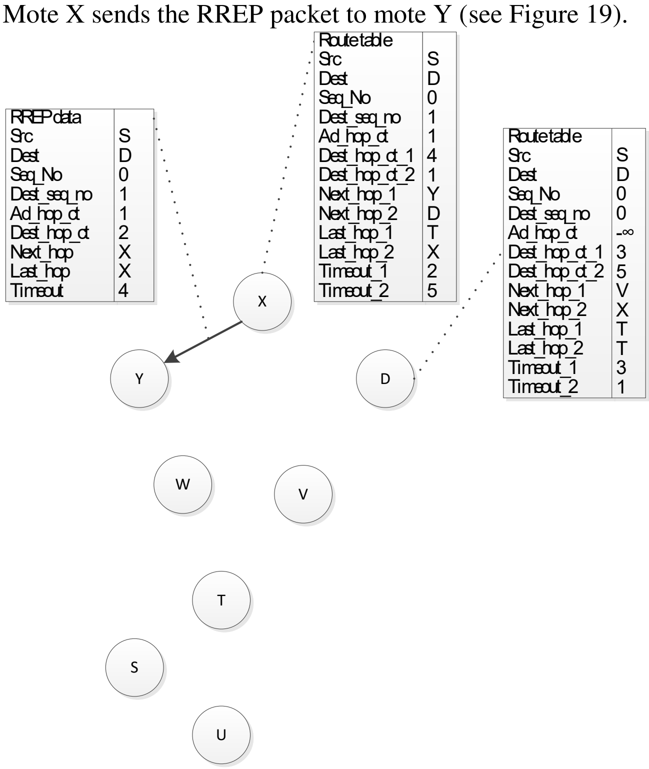

Fig. 19. Route Discovery: Backward Pass-Step 7

For brevity assume that mote Y updated its routing table, sent the RREP packet to mote W, which also updated its routing table and sent the RREP to mote T (see Figure 20).

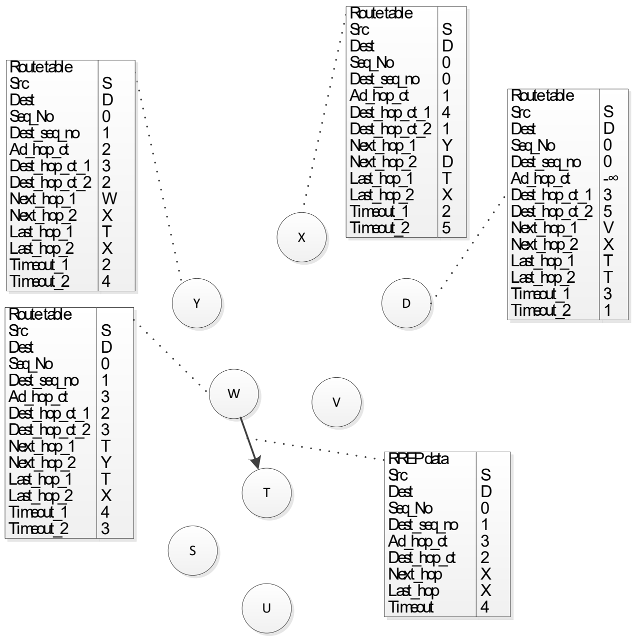

Fig. 20. Route Discovery: Backward Pass-Step 8.

Mote T updates its routing table and sends the RREP packet to mote S (see Figure 21).

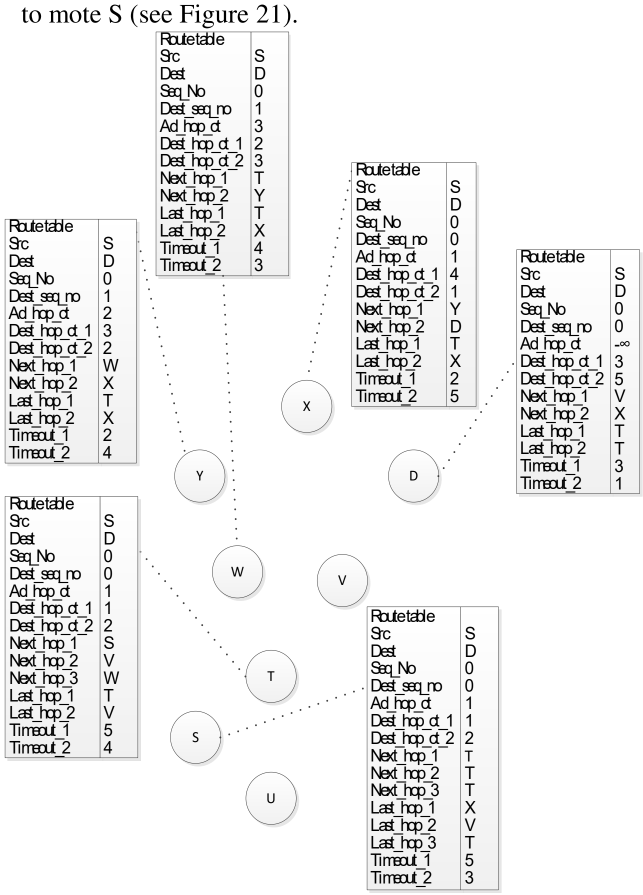

Fig. 21. Route Discovery: Backward Pass-Step 9.

S is the destination of the RREP packet, so it updates its routing table and drops the packet. The source will record the update of the receipt of the RREQ message after it is received. The backward pass of the route discovery

Volume: 10 Issue: 01 Pages: 3674-3686 (2018) ISSN: 0975-0290

process is now complete with each mote containing a forward path pointer from the source mote, S to the destination mote D. Note that Extra RREPs and RERRs for multipath discovery and maintenance and extra fields in routing control packets (RREQs, RREPs, RERRs) constitute added overhead in AOMDV compared to AODV.

4. Loop free paths

Modification of route update rules for each node in AOMDV to have more than one path to one particular destination. This should be done so that loop freedom is not compromised. Computing multiple loop-free paths prohibits a node from advertising random paths, (because this can result loops as different nodes have different destination hop counts) and from accepting all advertised paths (can result in loop formation).

On Figure 22, Node 7 is the destination and node 0 is the source.

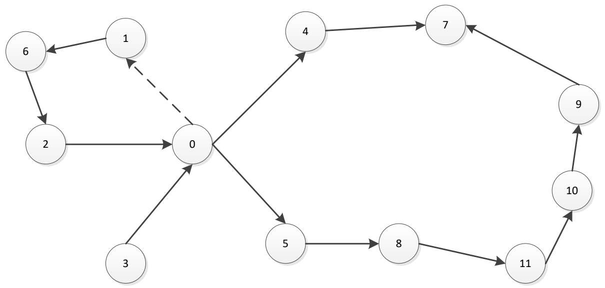

Fig. 22. Multiple Loop Free Paths- 1.

Node 0 has two paths to node 7:

1. A six-hop count path via node 5 (0→5→8→11→10→9→7)
2. A two-hop count path via node 4 (0→4→7)

Let node 0 advertise the six-hop path to node 3 and two hop path to node 2. Nodes 2 and 3 know how to get to node 7 but by using different paths with different destination hop counts. After some time, node 1 advertises a 5-hop count path to node 7. Node 0 does not know if node 1 is upstream or downstream as it only gets the destination hop count in the route advertisement. Node 0 then forms a path via node 1 forming a loop. This scenario occurs as node 0 is using two destinations hop counts for route advertisement so when it compares new route advertisements to them it will find a better one if the new advertisement is shorter than its longest advertisement. Here the path with hop count 5, even though greater than 2 is less than 7, so it is updated causing the loop formation.

In this example node 10 is the destination and node 0 is the source. Nodes 5 and 1 advertise paths with hop counts of value 5 to node 0. Suppose node 0 accepts node's 5 offers and obtains a 6 destination hop count to node 10 (see Figure 23).

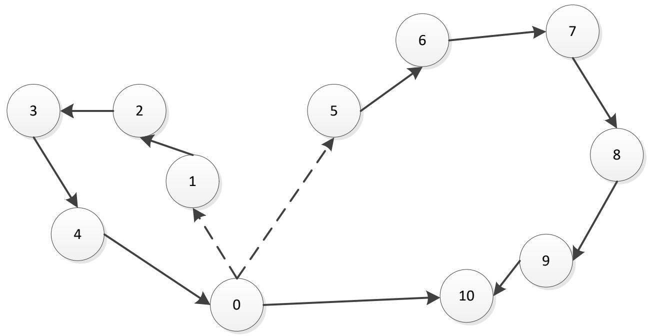

Fig. 23. Multiple Loop Free Path- 2.

It cannot determine whether node 1 is upstream or downstream as it offered the same hop count. The routing loop 0→1→2→3→4→0→10 may be formed if node 0 later accepts this route.

Multiple paths with loop freedom require a certain set of constraint satisfaction rules to be adhered too. Constraint satisfaction rules:

1. Maintain routes only for highest destination sequence numbers.
2. Destination sequence number:
   - Always advertise longer routes to the ones currently advertised. If the node is advertising a route of length x and a longer one is found that is of length x+1 then this new length would be advertised.
   - If you are advertising a route, never accept a longer one. If the node is advertising a route of length x and a longer one is found that is of length x+1 then the route of length x+1 would be rejected (not accepted).

To sum up the rules break down into: Advertise the maximum hop count in routing table for neighboring nodes but accept the lowest route.

AOMDV uses advertised hop count to have an identical sequence number for multiple paths with the same destination. During first advertisement a particular destination sequence number that denotes the length of the longest available path thus far is the advertised hop count value. This value is only changed when the destination sequence number changes. A greater number of alternate paths are maintained by nodes advertising the longest path lengths.

4.1 Alternate Path Selection

If a node k between nodes i and j on a path to the destination node d fails during data transmission and an alternate path is chosen which also has node k on its path, then this path will not facilitate data transfer. In multipath routing, because of the high computational costs to obtain additional paths their selection must be done intelligently. This means that in the case of node failure the selection of an alternate path should be such that the damaged node is

Volume: 10 Issue: 01 Pages: 3674-3686 (2018) ISSN: 0975-0290

not on the alternate path. This would ensure that data would be transmitted improving the resilience of the network. The same argument applies for link failures within the network.

4.2 Node Disjoint Paths

In AOMDV disjointness is between one pair of nodes, the source node, s and destination node, d. With AOMDV all paths from s to d are disjoint, where disjoint paths can be either node disjoint or link disjoint.

The number and quality of discovered disjoint paths are determined by the dynamics of the route discovery process. One can place a limit on the number and length of alternate paths maintained at each node. However, the cardinality and length of alternate paths are not optimized. This is justified by the diminutive lifetime of paths in mobile ad hoc networks and probably higher overheads of distributed computation of alternate (greatest number of paths and shortest path lengths) disjoint paths.

To check for link disjointness added information is required for each path. Two possible ways are used:

1. Each node maintains complete path information for all paths (disjointness check is easy)
2. M aintain last hop information for every path in addition to next hop, where last hop is from source node, S to destination D.

A simple case which illustrates link disjoint paths is given in Figure 24.

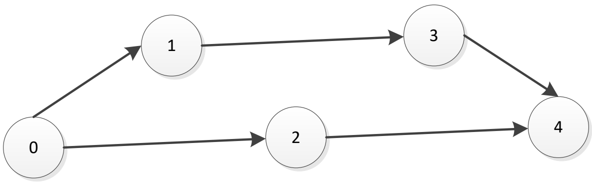

Fig. 24. Link Disjoint Paths- 1.

Single and last hop for one (with zero as the source mote on Figure 34: (node 3 has one next hop which is node 4 on a path to node 10) and two (with 0 as the source mote on Figure 25: all motes have two links/mote hops) hop paths.

Rule 1: if two paths from a node S to a destination D are link disjoint, then they must have unique next hops and unique last hops. The converse is not always true.
Rule 2: if every node on a path ensures that all paths to the destination from that node differ in their next and last hops then two paths from a node S to a destination D are link disjoint. Thus, using rule 2 will result in having two unique downstream neighbors being link disjoint when they have unique last hops.

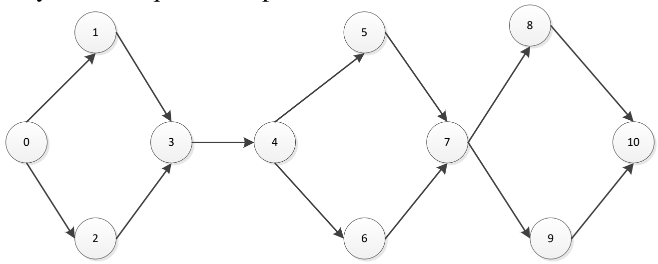

Fig. 25. Link Disjoint Paths- 2.

Path from node 0 to node 10 via node 2 (0→2→3→4→6→7→9→10) and node 1 (0→1→3→4→5→7→8→10) are not link disjoint even though they satisfy the conditions of Rule 1having different next and last hops.
In Figure 26 Rule 2 is satisfied having every node on the path from the source node 0 to destination node 12 differing in their next and last hop.

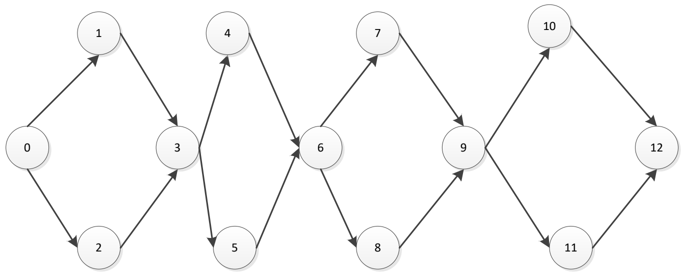

Fig. 26. Link Disjoint Paths- 3

The resulting paths via node 1 (0→1→3→4→6→7→9→10→12) and node 2 (0→2→3→5→6→8→9→11→12) are link disjoint. Link disjointness is ensured since any node on the path cannot have two paths with the same next hop. So here, each node can have two paths as they have available two separate next hops. For example, the path from node 7 to node 9 (7→9) and from node 8 to node 9 (8→9) are trivially link disjoint, because node 7 and node 8 are distinct neighbors of node 9. Take also for instance, the fact that each path advertisement will include the last hop of the path. Therefore, node 7 and node 8 will get the last hops when advertised by node 9, respectfully for the two different paths via node 10 and 11. The resulting paths from node 7 and node 9 will be link disjoint as they maintain distinct last hops from node 10 and 11. Similarly the same will occur for nodes 4 and 5, eventually resulting in the source node 0 having two distinct link disjoint paths to node 12. Finally, it must be noted that the paths via node 1 and node 2 are not node disjoint.

IV. PERFORMANCE OF THE AOMDV ROUTING PROTOCOL

To evaluate WSNs with the M-AOMDV routing protocol, the ns-2 simulator [21] is used. The ns-2 simulator is used extensively in evaluating the performance of ad hoc network routing protocols. The averages and/or variance found when running the simulation several times would give a statistical analysis of a simulation. However, the ns2 simulator is deterministic. This means that for every run the simulator would produce the same result. This would give a zero variance for any number of results. For the averages and variances to be meaningful the statistical analysis would require randomness [20]. To do this in ns2 a random seed is required to be set. In the simulation TCL script the seed was set to zero, so NS2 will change the seed according to clock and counter. This meant that the simulation results will be different for every run. A total of 30 runs were done and the average was taken from all the runs. This average result is used in the analysis sections of this thesis. These simulations model radio propagation using the realistic two-ray ground reflection model and account for physical phenomena such as signal strength, propagation delay, capture effect, and interference [22]. The Medium Access Control (MAC) protocol used is the IEEE 802.11 Distributed Coordination Function (DCF) (Bianchi 2000). This protocol has been used as a MAC layer protocol in WSNs for energy efficiency with commendable results. [24]

The energy model used in ns2 allowed the starting energy value of the mote to be set. All motes received the same initial energy value: 200 Joules. The mote residual energy is obtained from the ns2 energy model and this is the mote energy value at any specific point in time during the simulation. A 1000-byte packet size is used. However, the data frame length used in the experiment is 50 bytes. This means that at the MAC layer a 1000-byte packet will be broken down into twenty, fifty-byte data frames which will be sent across the network. This 50-byte data transfer size is still comparable to that of the TinyOS packet used specifically for WSNs. However, there will still be a lot of energy use at the MAC layer in order to break the 1000-byte packet into chunks. Thus, by setting the packet size to 1000 bytes the energy usage of the network is tested under the prevailing condition of high energy use at the mote. This is a necessary condition for high energy use which is critical to the simulation. Simulations are done under static (motes do not move) wireless sensor network environments.

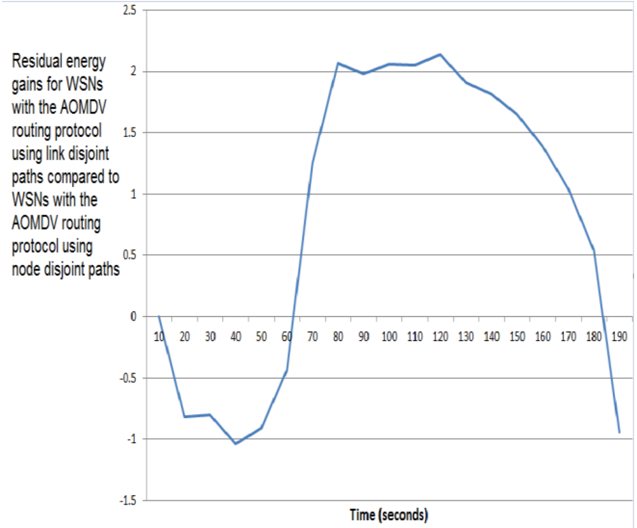

Fig. 27 Differences in Residual Energy of a WSN using the AOMDV Routing Protocol along Link Disjoint Paths and a WSN with the AOMDV Routing Protocol along Node Disjoint Paths

The graph on Figure 27 shows the residual energy gains for the WSN with the AOMDV routing protocol using link disjoint and node disjoint paths. The WSN with the AOMDV routing protocol using node disjoint paths show greater gains for the first 53 seconds of the simulation, with a peak gain of 1.05% at 35 seconds. From 57 to 74 seconds there is a rapid increase in residual energy gains for the WSN with the AOMDV routing protocol using link disjoint paths. Overall the WSN with the AOMDV routing protocol using link disjoint paths benefits from having better residual energy than the WSN with the AOMDV routing protocol using node disjoint paths, with a peak of 2.2% at 115 seconds. The WSN with the AOMDV routing protocol using link disjoint paths is better than the WSN with the AOMDV routing protocol using node disjoint paths for energy consumption.

V. CONCLUSION

This work illustrates the AOMDV routing protocol. Its ancestor, the AODV routing protocol is also described. This tutorial demonstrates how forward and reverse paths are created by the AOMDV routing protocol. Loop free paths formulation is described, together with node and link disjoint paths. Finally, the performance of the AOMDV routing protocol is investigated. The WSN with the AOMDV routing protocol using link disjoint paths is better than the WSN with the AOMDV routing protocol using node disjoint paths for energy consumption.

REFERENCES

[1] Akkaya, Kemal, and Mohamed Younis. "A survey on routing protocols for wireless sensor networks." Ad hoc networks 3, no. 3 (2005): 325-349.

[2] Al-Karaki, Jamal N., and Ahmed E. Kamal. "Routing techniques in wireless sensor networks: a survey." IEEE wireless communications 11, no. 6 (2004): 6-28.

[3] Andrews, Matthew, and Lisa Zhang. "Hardness of the undirected edge-disjoint paths problem." In Proceedings of the thirty-seventh annual ACM symposium on Theory of computing, pp. 276-283. ACM, 2005.

[4] Bettelheim, Bruno. "" Hansel and Gretel, my favorite fairy tale"." Psychotherapie, Psychosomatik, medizinische Psychologie 37, no. 1 (1987): 1-9.

[5] Frank, Howard, and Wushow Chou. "Topological optimization of computer networks." Proceedings of the IEEE 60, no. 11 (1972): 1385-1397.

[6] Guy, Chris. "Wireless sensor networks." In Sixth International Symposium on Instrumentation and Control Technology: Signal Analysis, Measurement Theory, Photo-Electronic Technology, and Artificial Intelligence, vol. 6357, p. 63571I. International Society for Optics and Photonics, 2006.

Volume: 10 Issue: 01 Pages: 3674-3686 (2018) ISSN: 0975-0290

[7] Handziski, Vlado, Andreas Koepke, Holger Karl, Christian Frank, and Witold Drytkiewicz. "Improving the energy efficiency of directed diffusion using passive clustering." In European Workshop on Wireless Sensor Networks, pp. 172-187. Springer, Berlin, Heidelberg, 2004.
[8] Marina, M., and S. Das. "On-Demand Multipath Distance Vector Routing in Ad Hoc Networks (AOMDV)." (2001).
[9] Marina, Mahesh K., and Samir R. Das. "On-demand multipath distance vector routing in ad hoc networks." In Network Protocols, 2001. Ninth International Conference on, pp. 14-23. IEEE, 2001.
[10] Meghanathan, Natarajan. "A node-disjoint multi-path routing protocol based on location prediction for mobile ad hoc networks." International Journal of Engineering, Science and Technology 2, no. 5 (2010): 66-80.
[11] Park, V., and S. Corson. Temporally-Ordered Routing Protocol (TORA). IETF Internet Draft, draft-ietf-manet-tora-sepc-00. txt, 1997.
[12] Perkins, Charles E., Elizabeth M. Royer, Samir R. Das, and Mahesh K. Marina. "Performance comparison of two on-demand routing protocols for ad hoc networks." IEEE Personal communications 8, no. 1 (2001): 16-28.
[13] Perkins, Charles, Elizabeth Belding-Royer, and Samir Das. Ad hoc on-demand distance vector (AODV) routing. No. RFC 3561. 2003.
[14] Royer, Elizabeth M., and Charles E. Perkins. "Multicast operation of the ad-hoc on-demand distance vector routing protocol." In Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, pp. 207-218. ACM, 1999.
[15] Shah, Rahul C., and Jan M. Rabaey. "Energy aware routing for low energy ad hoc sensor networks." In Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE, vol. 1, pp. 350-355. IEEE, 2002.
[16] Shah, Rahul C., and Jan M. Rabaey. "Energy aware routing for low energy ad hoc sensor networks." In Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE, vol. 1, pp. 350-355. IEEE, 2002.
[17] Shi, Elaine, and Adrian Perrig. "Designing secure sensor networks." IEEE Wireless Communications 11, no. 6 (2004): 38-43.
[18] Singh, Shio Kumar, M. P. Singh, and Dharmendra K. Singh. "Routing protocols in wireless sensor networks–A survey." International journal of computer science & engineering survey (IJCSES) Vol 1, no. 63-83 (2010): 29-31.
[19] Zimmermann, Hubert. "OSI reference model--The ISO model of architecture for open systems interconnection." IEEE Transactions on communications 28, no. 4 (1980): 425-432.
[20] Wang, Yunbo, Mehmet C. Vuran, and Steve Goddard. "Stochastic modeling of delay, energy consumption, and lifetime." In The Art of Wireless Sensor Networks, pp. 11-56. Springer, Berlin, Heidelberg, 2014.
[21] Issariyakul, Teerawat, and Ekram Hossain. Introduction to network simulator NS2. Springer Science & Business Media, 2011.
[22] Kotz, David, Calvin Newport, Robert S. Gray, Jason Liu, Yougu Yuan, and Chip Elliott. "Experimental evaluation of wireless simulation assumptions." In Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems, pp. 78-82. ACM, 2004.
[23] Bianchi, Giuseppe. "Performance analysis of the IEEE 802.11 distributed coordination function." IEEE Journal on selected areas in communications 18, no. 3 (2000): 535-547.
[24] Lee, Seungjoon, Bobby Bhattacharjee, and Suman Banerjee. "Efficient geographic routing in multihop wireless networks." In Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing, pp. 230-241. ACM, 2005.

Biographies and Photographs

Koffka Khan received the M.Sc., and M.Phil. degrees from the University of the West Indies. He is currently a PhD student and has up-to-date, published numerous papers in journals & proceedings of international repute. His research areas are computational intelligence, routing protocols, wireless communications, information security and adaptive streaming controllers.

Wayne Goodridge is a Lecturer in the Department of Computing and Information Technology, The University of the West Indies, St. Augustine. He did is PhD at Dalhousie University and his research interest includes computer communications and security.