Implementation of ETX Metric within the AODV Protocol in the NS-3 Simulator

Nenad J. Jevtic, *Member, IEEE,* and Marija Z. Malnar

***Abstract* — Modern life cannot be imagined without wireless ad hoc networks (WANETs). People use, in a daily manner, smart phones, laptops or tablet computers. With an increasing number of users of WANETs, the need for a fast throughput and small delay is increasing as well. One efficient way to provide users with such demands is to find an optimal route between source and destination node. Therefore, many routing protocols and metrics for WANETs are proposed in last decades. It is very important to test performances of the proposed protocols in various network scenarios. The computer simulation is a very efficient way to test routing protocols and to evaluate their significance and practical value. One of the most recent but very frequently used discrete-event network simulators is Network Simulator 3 (NS-3). In order to contribute to the NS-3 simulator development, an implementation of one of the most commonly used metric in WANET protocols the expected transmission count (ETX) is proposed in this paper. ETX metric is implemented within an ad-hoc on demand distance vector (AODV) protocol. The source code of this implementation is publicly available.**

***Keywords* — wireless ad hoc networks, NS-3, ETX metric, AODV protocol.**

I. Introduction

Wireless ad hoc networks (WANETs) have demonstrated their potentials in a variety of applications, from community networks to public safety and crisis management [1], [2]. There are many subsets of WANETs: wireless mesh networks (WMNs) with mostly static nodes, mobile ad hoc networks (MANETs) and vehicle ad hoc networks (VANETs), both with mobile nodes, etc. While VANET nodes (vehicles) tend to move in an organized fashion, according to roadside, traffic light, highway, etc, in MANETs nodes are free to move randomly in any direction. Having in mind all of this, it could be said that WANETs have a dynamic topology, variable link capacity, energy constraints and limited physical security. For that reason, managing the topology creation, configuration and maintenance of the WANETs is very important.

Routing protocols and metrics are responsible for maintaining network topology and determination of routes in WANETs [3]. In literature, a number of WANET routing protocols and metrics which aim to optimize network parameters such as throughput, packet loss probability, end to end delay, network gain are proposed.

In order to test routing protocols and metrics, it is possible to implement them into experimental WANET nodes, but this approach is costly and time consuming. In recent studies, a suitable network simulator which abstracts implementation details, while providing a good grade of realism is used as an alternative solution. There are many open source discreet-event simulators available online, such as Omnet++ [4], Glomosim [5], Network Simulator 2 (NS-2) [6], Network Simulator 3 (NS-3) [7], etc.

Most of the available simulators, such as Omnet++, Glomosim and NS-2, are realized as dual language simulators. One language, usually C++, is used for models, while some other language is used for network configuration. If network simulator is used only for running simulations with available models, using two languages can be advantage. However, for model developers using two languages adds an unnecessary increase in complexity. Having in mind that NS-3 is a single language simulator, written in C++, there is no surprise that it has increasing attention among wireless network researchers.

Many models of network protocols and algorithms are already implemented in NS-3, but there is no available implementation of one of the most commonly used metric expected transmission count (ETX), proposed by De Couto et al in [8]. Despite the fact that ETX has been proposed more than ten years ago, many recent studies still use ETX metric and its modifications [9 - 13]. Power-based routing metrics power ETX, power WCETT and power MIC based on ETX are introduced in [9]. Fast ETX (F-ETX) metric [10] modifies ETX in order to be used in dynamic VANET networks. Authors in [11] proposed ETX to power (ETP) algorithm which uses ETX metric to dynamically compute the transmission power and the carrier sensitivity threshold. A contact-aware ETX (CA-ETX) routing metric used to estimate the packet transmission delay caused by both packet retransmissions and intermittent connectivity is introduced in [12]. Predicted remaining deliveries (PRD) metric [13] combines various parameters, the residual energy, link quality (through ETX), end-to-end delay, and distance together to achieve a better network performance.

The main goal of this paper is to give a detailed implementation of ETX metric within one of the frequently

Paper received April 20, 2018; revised June 2, 2018; accepted June 10, 2018. Date of publication July 31, 2018. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Aleksandar Nešković.

This paper is a revised and expanded version of the paper presented at the 25th Telecommunications Forum TELFOR 2017 [16].

This research is supported by the Serbian Ministry of Science and Technological Development, projects number TR320025 and TR36047.

Corresponding author Nenad J. Jevtic is with Faculty of Transport and Traffic Engineering, University of Belgrade, Vojvode Stepe 305, 11000 Belgrade, Serbia (phone: 381-11-3091377; e-mail: n.jevtic@sf.bg.ac.rs).

Marija Z. Malnar is with the Faculty of Transport and Traffic Engineering, University of Belgrade, Vojvode Stepe 305, Belgrade, Serbia (phone: 381-11-3091322, e-mail: m.malnar@sf.bg.ac.rs).

used WANET protocols, Ad-hoc on demand distance vector (AODV) [14] in NS-3 simulator. The source code of this implementation is free to use and can be found at [15].

This paper is organized as follows. In section 2 a short overview of ETX metric and AODV protocol is given. In section 3 the implementation of ETX metric in AODV protocol, called AODV-ETX is provided. The fourth section gives the evaluation of the proposed implementation in NS-3 simulator. Concluding remarks are given in section 5.

II. ETX METRIC WITHIN AODV PROTOCOL

ETX metric is one of the most commonly used metrics in various routing protocols. In this paper AODV protocol is chosen since this is one of the most frequently used protocols in MANETs.

A. ETX metric

ETX metric of a link is represented by the predicted number of data transmissions required to send a packet over that link, including retransmissions.

If p_f (forward) represents the probability of successful packet transmission, and p_r (reverse) the probability of successfully received ACK packet, then the probability that packet is successfully sent and acknowledged will be $p_f \cdot p_r$. The ETX for a link l, is given by [8]:

$$ETX_l = 1/(p_f \cdot p_r). \quad (1)$$

The probabilities p_f and p_r are measured using dedicated link probe packets (LPPs). Each node broadcasts LPP of a fixed size, at an average period τ. To avoid accidental synchronization, period τ is jittered by up to ±10%. Because the probes are broadcast, nodes do not acknowledge or retransmit them, so the calculation of probabilities p_f and p_r requires a special technique. Every node remembers the number of received LPPs during the last w seconds allowing it to calculate the probability p_r at any time t as [8]:

$$p_r = count(t-w, t)/(w/\tau). \quad (2)$$

Count $(t-w, t)$ is the number of LPPs received during the window w, and w/τ is the number of LPPs that should have been received. In the case of the link X→Y, this technique allows X to easily measure p_r by counting successfully received LPPs from Y. But due to the lack of acknowledgements, node X cannot determine the probability p_f. Since the calculation of ETX metric for link X→Y requires both p_f and p_r, each LPP sent by node Y contains the number of LPPs received from X during the last w seconds. This allows node X to calculate the p_f.

An empirical method is used for choosing the w [8]. Experiments found that a value of $w=10\tau$ performs well. Usually the value of τ =1s is used. The metric of the route r, is the sum of the ETX values for each link l in the route [8]:

$$ETX_r = \sum_{l \in r} ETX_l \quad (3)$$

B. AODV protocol

AODV [14] is a reactive routing protocol, which means that a route is requested only when a node has data to send. AODV uses three basic control packets Route Request (RREQ), Route Replay (RREP) and Route Error (REER).

In order to give a short overview of AODV protocol route discovery mechanism, the simple network shown in Fig. 1 is analyzed. Low loss links are represented with dashed lines (links S-A, A-B, B-C, C-D), and high loss link with full line (link A-C). The assumption is that a source node, S, wants to send a packet to a destination node, D. Based on metric used in the original AODV protocol, the best route is the one with the smallest hop count. Therefore route S-A-C-D, which includes high loss link A-C, is chosen.

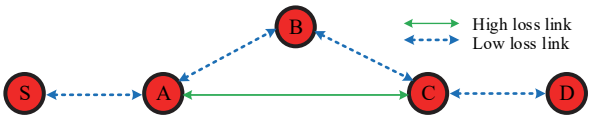

Fig. 1. Simple network.

In order to show a route discovery procedure the timeline of RREQ and RREP packet exchange, is shown in Fig. 2.

Source node S first broadcasts RREQ (with hop count set to 0). Since node S has only one neighbor, node A, it is the only one node that receives RREQ (Step 1).

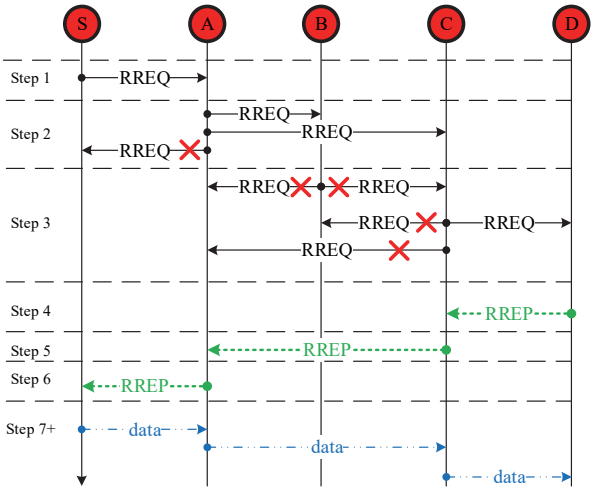

Fig. 2. RREQ and RREP packet exchange within AODV protocol.

After receiving RREQ, node A creates (or updates) its own routing table entry to the source node. The value of hop count to the source node is the value from the received RREQ packet (where hop count was 0) increased by one. Node A checks if D flag (destination only flag) in the received RREQ is set to 1. In this case, node A updates received RREQ packet by incrementing the hop count and rebroadcast updated RREQ (Step 2). Three RREQs from node A will be received, since it has 3 neighbors: S, B and C. In the AODV protocol if a received RREQ packet from the same source node and with the same ID is already seen, a node discards that RREQ. Therefore, RREQ sent back to node S is discarded.

Both nodes B and C receive RREQ from node A, update their routing tables with a new route to the source and a new hop count (now hop count is set to 2), and rebroadcast RREQ (Step 3). Since node B has two neighbors, node A and C, and the RREQ from node A is already received by node C, both RREQs sent from node B will be discarded. The same will happen with RREQs sent from node C and received by nodes A and B. Only the RREQ sent from node C and received by node D will not be discarded.

Finally, when destination node D receives RREQ from node C, it first creates (or updates) its routing table with a route to the source node. The hop count for that routing table entry is set to the value from received RREQ increased by one (now hop count is set to 3). The destination node than generates unicast RREP packet with hop count set to 0, and sends it back to the source node (Steps 4-6) using the newly established route. When any intermediate node receives RREP packet, it increments hop count of RREP, creates (or updates) its routing table entry to the destination node, and forwards updated RREP to the source node.

When source node receives RREP it now has the route to the destination and it can send data packets (Steps 7 and later).

In AODV protocol, there is also a possibility to allow intermediate nodes (such as A, B and C in Fig.1) to answer to RREQ by sending RREP packet. This feature is achieved by setting the D flag in RREQ packet to 0. In that case, if some intermediate node has a route to the destination, it compares DSNs (destination sequence numbers) from received RREQ and from its own routing table. If DSN of RREQ packet is greater than DSN from its routing table, intermediate node has outdated route, and acts as D flag is set to 1. But, if DSN from routing table is greater or equal, the intermediate node generates RREP packet. In this situation, an intermediate node sends a unicast RREP packet back to the source node with hop count field set to the number of hops to the destination node (taken from its routing table).

If some node detects a link failure, it deletes the routes that contain this link in its routing table and notify its neighbors of the link failure by generating REER packet.

III. Implementation of AODV-ETX protocol in NS-3

To use the ETX metric, the AODV protocol is modified in a few details.

A. Implementation of LPP packet end ETX calculation

In order to implement ETX metric, additional LPP packets are used to measure ETX, as previously described. The ETX implementation measures link loss ratios with LPP packets that are broadcasted over the network [8]. Each node broadcasts one LPP per second, and remembers LPPs received from its neighbors over the last ten seconds. Using relatively small LPPs saves bandwidth. The structure of LPP packet is shown in Fig.3.

The LPP packet consists of seven fields:

- *Type* (8 bit field that indicates type of AODV packet),
- *LPP ID* (8 bit field for identification of LPP packet),
- *Originator IP Address* (32 bit IP address of node that generates LPP packet),
- *Originator Sequence Number* (32 bit sequence number of node that generates LPP),
- *Number of Neighbors* (8 bit filed that indicates number of neighbors of the node that generates LPP packet),
- *Neighbor IP Address* (32 bit IP address of a neighbor from which at least one LPP packet is received in last 10 seconds),
- *ForwardLppCount* (8 bit field that indicates number of received LPP packets from the neighbor, with IP address given in previous filed, in last 10 seconds).

Last two fields in LPP packet are repeated for each neighbor. A maximal number of neighbors is indicated with *n*, in Fig. 1, and is written in field *Number of Neighbors*.

Type	8b
LPP ID	8b
Originator IP Address	32b
Originator Sequence Number	32b
Number of Neighbors (n)	8b
Neighbor IP Address 1 (32b)	n * 40b
Forward LPP Count (8b)	

Fig. 3. Structure of LPP packet.

In order to calculate ETX metric, in each node a new neighbor table is introduced. Each table entry corresponds to one neighbor. Table entry has three fields:

- *NeighborIpAddress*, which contains IP address of neighbor,
- *ReverseLppCount*, which tracks number of LPP packets received from the neighbor, with IP address given in previous filed, in last 10 seconds, and
- *ForwardLppCount*, which tracks number of LPP packets that the same neighbor has received from this node in last 10 seconds.

ReverseLppCount is obtained by counting received LPPs from appropriate neighbor. *ForwardLppCount* is obtained from received LPP packets, as previously described in section II. Having these two values for each neighbor, node can calculate ETX metric for all neighbors.

B. Modification of routing table, RREQ and RREP packets

In order to include ETX metrics in AODV-ETX protocol, the field which indicates ETX metric needs to be added in several ways.

First, each routing table entry is expanded with the ETX filed. In AODV protocol routing table entries are classified by destination. If there is more than one route to a destination, routes are classified by hop count, where the best route is the one with the smallest hop count. With AODV-ETX protocol, routes should be classified regarding the ETX value of the route, where the best route is the one with the smallest ETX value. If more than one route with the same ETX value exists, the best is the one with the smallest number of hops.

Modifications of RREQ and RREP packets are simple, the additional field, which represents ETX value is added in both packets. When source node, S, creates RREQ it sets the initial value of ETX to zero. This field is handled in a similar manner as hop count. As RREQ is forwarded to the destination node, ETX field is updated in each node according to (3). The same principle is used in sending the RREP packets.

Since ETX generally is not an integer value and NS-3 simulator implements serialization and deserialization of packets, appropriate bit representation of this field should be provided. For the simplicity of (de)serialization function a following procedure is implemented. ETX field has a length of 32 bits. The value of ETX calculated using (1) is

multiplied by 10^4 and rounded to the nearest integer value. Using integer values as ETX simplifies implementation in NS-3 simulator and multiplication provides resolution of 4 decimal digits.

In the AODV-ETX protocol, the ETX values are cumulative, calculated as the sum of the ETX of the links on the chosen route, equation (3). Therefore, this implementation is limited to 4294 nodes in the route, which is more than enough for simulations.

C. *Handling RREQ and RREP packets*

As in basic AODV protocol, RREQ can be received by both intermediate and destination nodes. In order to explain differences between handling RREQ and RREP packets within AODV and AODV-ETX protocol, the same simple network shown in Fig. 1 is analyzed. Again, node S wants to send packets to the node D. The timeline of RREQ and RREP exchange within AODV-ETX protocol is shown in Fig. 4.

As with AODV protocol, node S broadcasts RREQ, but now with both hop count and ETX values set to 0. When an intermediate node receives RREQ it first calculates ETX value to the neighbor from which RREQ packet was received, based on neighbor's IP address and previously obtained *ForwardLppCount* and *ReverseLppCount* values. Then, node updates its own routing table to the source node with new ETX value which is the sum of calculated ETX value to the neighbor and ETX value from received RREQ.

In the example shown in Fig. 4, in Step 1, node S will broadcast RREQ. Since node A is the only neighbor of node S, it receives RREQ, calculates ETX metric for the link S-A, and updates its routing table with route to S. Then, node A checks the D flag in the received RREQ. If it is set to 1, node A updates RREQ with calculated new ETX value and incremented hop count, and rebroadcasts it (Step 2). If the flag D is set to 0, the procedure is the same as with the basic AODV protocol, with a difference that, besides hop count value, a current value of ETX metric from routing table will be sent as well.

In the basic AODV protocol, any intermediate (as well as destination) node answers only the first received RREQ and ignores any succeeding RREQ from the same source node and with the same ID. In AODV-ETX, if a node receives RREQ from already seen source node, it rebroadcasts RREQ (or answers with RREP packet) if the value of ETX from received RREQ has a smaller ETX value than the appropriate value from the routing table, or if ETX values are equal, but the new hop count is smaller.

In the Step 3, RREQs from node B are received in nodes A and C. In the basic AODV protocol both RREQs were discarded since they have already been seen. But, with AODV-ETX protocol, when node C receives RREQ packet from node B (circled RREQ in Fig. 4) it will not be automatically discarded. Node C first checks if the ETX metric in this packet is smaller than the ETX metric that was previously received from node A and stored in routing table of node C. If this is the case, node C would not discard RREQ from node B. Based on ETX metric, the path S-A-B-C has smaller ETX metric than the path S-A-C. This is possible since the link A-C (Fig. 1) is a link with a high loss, and A-B and B-C are low loss links. Therefore, node C updates ETX and hop count fields and rebroadcasts RREQ received from B.

When destination node D receives RREQ, it first creates (or updates) its routing table with a new route to the source node. Value of ETX for that routing table entry is a sum of ETX metric from received RREQ and ETX of link D-C. Node D then generates RREP packet (Step 4) with ETX and hop count values set to 0, and sends it back to the node C.

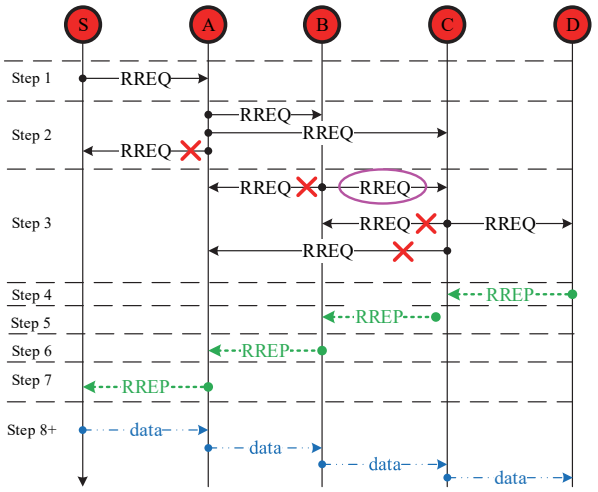

Fig. 4. RREQ and RREP packet exchange within AODV-ETX protocol.

When RREP packet is received in node C, it increments hop count of RREP, updates ETX value by adding ETX of the link D-C, creates (or updates) its routing table entry to the destination node, and forwards updated RREP (Step 5) to the source node over the best route (in this case over node B). All other intermediate nodes on the route to the source node follow the same procedure (Steps 6 and 7). Finally, as with the basic AODV protocol, when RREP is received by the source node, it will start sending data packets (Steps 8 and later).

In order to minimize latency, the source node starts sending data packets after receiving the first RREP. However, if an additional RREP with better metric is received later, the routing table of the source node would be updated with a better route, and a new route will be used for later data packet transmissions.

IV. Evaluation

In this section, the simulation results, comparing the AODV-ETX with AODV are shown.

A. *Simulation setup*

For the development of this additional patch originally the NS-3.26 simulator was used. Recently, the newer versions of simulator were introduced. The most recent version, NS-3.28, has included some minor changes in the original AODV module. Therefore, the AODV-ETX model has been updated for new version NS-3.28. Since basic AODV protocol differs in newest and previous versions of NS-3, this updated patch cannot work with earlier versions of simulator. Model parameters that have been used in the following experiments are summarized in Table 1.

TABLE 1: SIMULATION SETUP PARAMETERS.

Parameter	*Values*
Protocols	AODV, AODV-ETX
MAC protocol	IEEE 802.11 b
Simulation time	100 s
Number of nodes	25
Bandwidth	2 Mbps
Traffic type	CBR
Packet size	512 bytes

In order to test the AODV-ETX protocol behavior, in this paper, a grid network that consists of 25 nodes, distributed as in Fig. 5, is analyzed. The propagation conditions are set so that each node has a maximum of 4 neighbors, the middle nodes have 4, the edge nodes have 3, and the nodes in the corners have 2 neighbors. Data traffic is generated from node 24 to node 0, using traffic generator *OnOffApplication*, which is installed in node 24. Data packets are received by application *PacketSink*, which is installed in node 0.

For obtaining simulation results (throughput, end to end delay, packet loss ratio and overhead) the NS-3 tracing system is used. Although the NS-3 simulations can use *FlowMonitor* for acquiring simulation results, this module is not reliable for measurement of routing overhead. Therefore, in the original AODV, as well as in the AODV-ETX protocol, an appropriate trace source is created and used for analyzing the overhead of both protocols. This trace source is triggered whenever protocol has a routing packet to send. Connecting to this trace source from configuration script enables user to capture and analyze control packets in AODV and AODV-ETX protocols, so the information on routing overhead could be obtained.

Throughput and packet loss are easily obtained using existing trace sources *Tx* and *Rx* in *OnOffApplication* and *PacketSink*, respectively. Trace source *Tx* is triggered whenever *OnOffApplication* sends, and *Rx* whenever *PacketSink* receives a data packet. User script can be connected to these sources and analyze sending and receiving of data packets and calculate appropriate statistics.

End to end delay measurement for application packets is obtained by modifying the traffic generator *OnOffApplication*. In order to modify the traffic generator, a packet header is added, which contains the time stamp of the sending time of this data packet. On packet reception the packet header is removed and the difference between sending and receiving time of the packet is calculated.

In order to test route selection based on ETX metric, a network topology with different link losses is used. In the chosen testing setup, Fig. 5, links between nodes 0-1, 1-2, 2-3, 3-4, 4-9, 9-14, 14-19 and 19-24 are low loss links, which form the route with low ETX, while other links have high losses and therefore form routes with high values of ETX metric. It should be expected that AODV-ETX protocol always chooses the best route 24-19-14-9-4-3-2-1-0 and therefore provides better performance results.

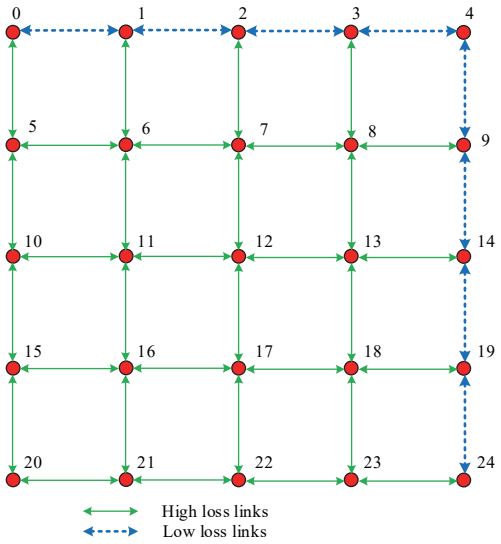

Fig. 5. Testing setup.

B. Results

In order to test the implementation of ETX metric within AODV protocol, simulations were done with different traffic loads, as shown in Table 2.

Both basic AODV and AODV-ETX protocols have chosen an appropriate eight hop route, but while AODV randomly chooses the route, AODV-ETX always chooses the route 24-19-14-9-4-3-2-1-0 (see Fig. 6 and Fig. 7). This is expected, since this route contains links that have low loss and therefore the best ETX metric.

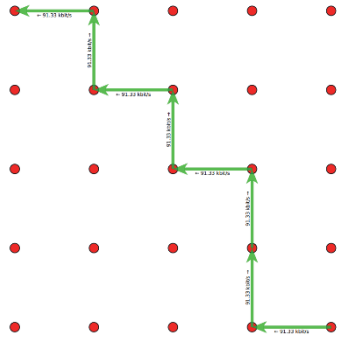

Fig. 6. Route for sending data packets chosen by AODV protocol.

Although both protocols have chosen routes with the same hop count (8), since the AODV protocol does not favor the low loss path, the simulations results show that on average it chooses route which gain in greater delay and packet loss ratio than with AODV-ETX.

As can be seen from Table 2, with increasing the data load, the overall network performance is slightly improved. Having in mind that the number of data packets is increasing at higher data loads, while route discovery procedure and routing overhead are practically the same regardless of data load, this is expected.

TABLE 2: SIMULATION RESULTS.

Data load [kbps]	*Protocol*	*Hop count*	*ETX*	*End to end delay*		*Throughput [kbps]*	*Packet Loss Ratio [%]*	*Overhead*	
				[ms]	*Relative [%]*			*[packets]*	*[kbps]*
10	AODV	8	-	54.16	-	9.99	1.05	118	164.66
10	AODV-ETX	8	8.11	45.17	16.6	9.99	0.82	121	224.83
15	AODV	8	-	50.35	-	14.79	1.82	120	167.11
15	AODV-ETX	8	8.11	39.42	21.7	14.83	1.37	121	225.16
20	AODV	8	-	46.54	-	19.66	2.07	120	166.69
20	AODV-ETX	8	11.09	36.11	22.4	19.75	1.64	121	225.32
30	AODV	8	-	42.73	-	29.46	2.85	120	166.87
30	AODV-ETX	8	8.22	32.53	23.9	29.50	1.78	123	228.31

Overhead analysis also shows expected results. The number of generated routing protocol packets (RREQ, RREP, RREP-ACK and RERR) is practically the same for both protocols. Table 2 shows that every node sends, on average, around 120 packets during the simulation time. However, Hello packets from AODV protocol are replaced with the larger LPP packets in AODV-ETX protocol, which results in greater overhead.

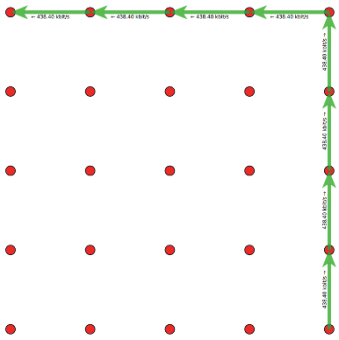

Fig. 7. Route for sending data packets chosen by AODV-ETX protocol.

Every neighbor adds 5 bytes in the LPP packet, which creates potentially large LPP packets. Also, every RREQ and RREP has an additional ETX field of 4 bytes in AODV-ETX protocol. This leads to larger overhead in AODV-ETX protocol, as can be seen from Table 2. In order to reduce overhead, implementation can be easily changed to reduce the size of ETX field from 4 bytes to 3 or even 2 bytes. However, this leads to limitation in the maximum number of hops allowed in one route or/and reduction in ETX resolution.

V. CONCLUSION

The main goal of this paper is to provide a detailed implementation of widely used ETX metric in NS-3 simulator, as well as the complete source code that can be found in [15]. In order to test this implementation, a simple simulation scenario is created and the basic AODV protocol is compared with the AODV-ETX protocol. As expected, AODV-ETX gives a better performance as compared to AODV in terms of end-end delay and packet loss ratio, whereas AODV gives a better performance in terms of routing overhead.

Since the simulation results show the expected network behavior, it can be concluded that the implementation of ETX metric in NS-3 simulator is successful, and that it can be used as a guideline for other implementations and modifications of AODV-based protocols and metrics that use ETX.

Bearing in mind that routing protocols are realized as separate models in NS-3 simulator, part of the future work will be the implementation of the ETX metric in other routing protocols, such as DSDV (Destination-Sequenced Distance-Vector), OLSR (Optimized Link State Routing), DSR (Dynamic Source Routing), LQSR (Link State Source Routing), etc.

REFERENCES

[1] I.F. Akyildiz, X.Wang, "A Survey on Wireless Mesh Networks", *IEEE Comm. Magazine,* Vol. 3, No. 9, Sep. 2005, pp. S23–S30.

[2] M.Portman, *Wireless Mesh Networks for Public Safety and Disaster Recovery Applications*, Auerbach Publications, 2006.

[3] Y.Yang, J.Wang, R.Kravets, "Designing Routing Metrics for Mesh Networks", *Proc.of the IEEE Workshop on Wireless Mesh Networks (WiMesh)*, 2005, pp. 1-9.

[4] OMNeT++ [Online]. Available: http://www.omnetpp.org.

[5] GlomoSim. [Online]. Available: http://pcl.cs.ucla.edu/projects/domains/glomosim.htm

[6] NS-2. [Online]. Available: https://www.isi.edu/nsnam/ns/.

[7] NS-3. [Online]. Available: http://www.nsnam.org/

[8] S. J. De Couto, D. Aguayo, J. Bicket, R. Morris. **"**A high–throughput path metric for multi-hop wireless routing**",** *Wireless Networks*, Vol. 11, No 4, 2005, pp. 419–434.

[9] M. Malnar, N. Nešković, A. Nešković, "Novel power-based routing metrics for multi-channel multi-interface wireless mesh networks", *Wireless Networks*, Vol. 20, No. 1, January 2014. pp. 41-51.

[10] S. Bindel, S. Chaumette, B.Hilt, "F-ETX: An Enhancement of ETX Metric for Wireless Mobile Networks", *Proc.of the Int. Workshop on Communication Technologies for Vehicles*, 2015, pp 35-46.

[11] T. Ropitault, N. Golmie, "ETP algorithm: Increasing spatial reuse in wireless LANs dense environment using ETX", *Proc. of IEEE Symp. Personal, Indoor, and Mobile Radio Communications*, 2017, DOI10.1109/PIMRC.2017.8292351

[12] S. Yang, U. Adeel, Y. Tahir, J. McCann, "Practical Opportunistic Data Collection in Wireless Sensor Networks with Mobile Sinks", *IEEE Trans. on Mobile Comp.*, Vol. 16, No. 5, May 2017, pp. 1420 - 1433.

[13] X. Lai, X. Ji, X. Zhou, L. Chen, "Energy Efficient Link-Delay Aware Routing in Wireless Sensor Networks", *IEEE Sensors Journal*, Vol.18, No. 2, 2018, pp. 837 - 848.

[14] C. E. Perkins, E.M. Belding–Royer, S.R. Das, "Ad Hoc On demand Distance Vector (AODV) routing", *RFC 3561*, 2003.

[15] AODV-ETX code [Online]. Available: https://github.com/neje/ns3-aodv-etx

[16] N. J. Jevtic and M. Z. Malnar, "The NS-3 simulator implementation of ETX metric within AODV protocol", *2017 25th Telecommunication Forum (TELFOR)*, Belgrade, 2017, pp. 1-4.