AAODV/AAOMDV Routing Protocols: Single and Multipath Routing in WMNs

Horia Ştefănescu
POLITEHNICA University of Bucharest
Orange Romania
Bucharest, Romania
horia.stefanescu@orange-ftgroup.com

Mariusz Skrocki, Sławomir Kukliński
Orange Labs, Telekomunikacja Polska
Warsaw University of Technology
Warsaw, Poland
{mariusz.skrocki, slawomir.kuklinski}@telekomunikacja.pl

Abstract—**This paper consists of two parts. In the first part, we propose a new routing protocol, named Adaptive Ad hoc On-Demand Distance Vector (AAODV). It is able to establish routes using any per link calculated routing metric, due to its ability to separate the monitoring of the quality of the paths from the routing mechanisms. By using AAODV, the following widely used metrics: hop count, delay, jitter and Expected Transmission Count (ETX) are compared using ns-3 simulations performed in eight randomly generated topologies with different traffic patterns. The results have shown that in the case of random topologies none of the routing metrics used provides significantly better results than the other one. In the second part of our work, the AAODV functionality is enhanced by adding multipath routing and end-to-end Real-Time Monitoring (RTM) of the paths. The new improved protocol is named Adaptive Ad hoc On-Demand Multipath Distance Vector (AAOMDV). AAOMDV provides multiple paths between source and destination nodes, therefore it is mandatory to implement an algorithm that selects the preferred path and switches the traffic on it when this is expected. Our simulations performed in ns-3 provide an inside look into AAOMDV functionality and prove that AAOMDV is able to enhance network performance when the network load increases.**

Keywords-AODV; AAODV; AAOMDV; multipath; routing metrics; Wireless Mesh Networks

I. INTRODUCTION

The scope of this paper is to analyze and to propose new solutions to overcome some of the routing challenges that appear in Wireless Mesh Networks (WMNs). Typically, the WMNs are based on single path, single metric and single radio. In this paper we will focus on the influence of the new concepts, such as: real-time WMN monitoring, multi-parametric metrics and adaptive path selection in multipath routing.

Our work is composed of two parts. In the first part, we propose a new routing protocol, called AAODV [1], which is based on the well-known Ad hoc On-Demand Distance Vector (AODV) protocol [2]. The main innovation of AAODV is that it can be implemented with any kind of per link calculated routing metric. The routing metrics are used during the routing table building phase in order to select the best path in the network. The impact of AAODV routing metric type on the network performance has been verified by extensive simulations. In order to obtain generic results, eight randomly generated topologies with random background traffic were used in the simulations. In each simulated case, we monitored Packet Loss Ratio (PLR), delay and jitter of the source traffic, and the results were averaged accordingly.

In the second part of this paper, the Adaptive Ad hoc On-Demand Multipath Distance Vector (AAOMDV) is presented. AAOMDV is the AAODV protocol extended by the multipath routing and real-time path monitoring that is used for the selection of data forwarding path. The end-to-end path monitoring functions are realized by the specially designed component of the routing architecture, named Real-Time Monitoring (RTM). The RTM simultaneously monitors PLR, delay and jitter of the path. To monitor the active path, the RTM uses traffic packets. For all other paths, called inactive paths, probe messages, of size similar to the traffic packets, are sent to evaluate their quality. This way, RTM provides the real values of PLR, delay and jitter of the active paths and just an estimate of delay and jitter values for the inactive paths. Note that for the inactive paths, RTM does not evaluate the PLR value.

The behavior of AAOMDV has been verified by simulations. The results have shown that the AAOMDV protocol combined with the algorithms for path selection and switching increases the network performance in terms of PLR, delay and jitter; thus providing a better user experience. In the simulation scenarios, we used random topologies and considered different traffic patterns. The simulations were performed using ns-3 [3]. The AODV protocol was used as a benchmark.

The paper is structured as follows. This section describes the research motivation and introduces the proposed concept. Section II presents the related work. In Section III, the AAODV protocol is described, whereas Section IV shows the results of the AAODV simulation. Section V describes AAOMDV, the algorithm for discovering multiple link disjoint paths and the RTM implementation. Section VI presents in detail the algorithm applied for the best path selection in the multipath case. In Section VII, an approach used for active path switching is described. In Section VIII AAOMDV simulation results are presented. We conclude this paper in Section IX.

II. Related Work

It has been shown that the practical performance of a WMN differs from the simulated one [4]. This is the reason for which some modifications of the original protocols have been proposed in the WMN implementations. In [5], a test-WMN (ReMESH) based on Optimized Link State Routing (OLSR) protocol combined with a modified ETX metric (in fact the Minimum Packet Loss Ratio parameter) is proposed. The authors have shown that in comparison to the original OLSR, the performance of the mesh network was improved, leading to more stable routes, lower packet loss rates, smaller delays, and in many cases a small increase in the network throughput. In [6], another test-WMN, called RoofNET, is described. RoofNET is based on a routing protocol named SrcRR that tries to maximize the throughput of the paths. The results presented in both previously cited papers were obtained in a real environment.

In WMNs, the routing metrics greatly impact the network performance. As it has been proved, they should also consider physical layer phenomena, like SINR, interflow interferences or the so-called flow self-interferences, introduced by the hidden and exposed terminal problems [7]. Examples of the most popular WMN routing metrics include: hop count, delay, jitter and ETX. There are approaches, which take into account physical layer processes, e.g., traffic aware metrics like PPTT [8]. However, these metrics are mainly probabilistic ones and they impose quite complex cross-layer operations. It is worth mentioning that the hop count metric does not establish the path according to its actual quality. The other metrics select the routes taking into account parameters of the component links. Therefore the routing protocol adapts to the network state. Routing protocols that use more than one metric can be found in literature. An example of such a protocol is Sharp Hybrid Adaptive Routing Protocol (SHARP) [9]. In the existing approaches, there is no decoupling between the monitoring and routing. This is the reason why it is very difficult to find a comparison of the same routing protocol with different routing metrics used. Nevertheless, some comparisons exist, e.g., in [10], in which the authors compared ETX metric with hop count metric using a grid topology only. Unfortunately, the grid topology is a particular case and it cannot yield relevant results for a wide variety of WMNs. However, a lot of theoretical comparisons exist and the most complete are [11] and [12].

The common approach for routing in both wired and wireless communication systems is the single path approach. However, it has been observed that the reliability and the performance of the network may be improved when more than one path between source and destination nodes is used. There are few scenarios, in which the multipath feature is useful. The simplest one is to discover the additional paths and to use them as a backup when the main route fails (AODV-BR [13], AOMDV [14] and AODVM [15]). This way, it is not necessary to perform the route discovery procedure every time the path breaks, because another path is already available in the routing table. This is the major advantage, especially in the networks with mobile nodes. When the multiple paths are used simultaneously [16], the traffic may be split among them on per packet or on per flow basis, enabling the load balancing. Another possibility is to replicate the data on each of the discovered paths, thus ensuring enhanced reliability. It has been shown that the improvement of multipath may be achieved only when a limited number of additional paths are kept in the routing table. According to [14] they should be limited to two or three paths, in order to avoid the existence of stale paths in the routing table. In [14] also the disjointness of the paths in the network is considered. Two paths may be either link or node disjoint. In the first case, it is acceptable for two paths to share common nodes. However, if the mutual node fails, both paths will become useless. The second option is much stricter, but at the same time it improves the reliability of the communication. The problem appears in small or sparse networks, because it may be impossible to establish node disjoint routes there.

III. AAODV

There are two main challenges for a routing protocol, i.e., finding the best path and loops avoiding. The “best path” can be defined as the path from the source to the destination that minimizes the end-to-end PLR, delay and jitter. One of the most popular protocols designed for ad-hoc networks, AODV, accomplishes only the second property by using sequence numbers in order to find loop free paths. AODV is relatively simple (see [2]). Every time a node wants to send a packet and does not know a route to the destination, it broadcasts a Route Request (RREQ) message. When any node, including the destination, receives this RREQ, it checks whether it has received a duplicate RREQ within a fixed interval of time. If such RREQ has been received, the node silently discards the newly received RREQ. During the RREQ broadcasting period, the reverse path (from the destination to the source) is established. When the destination receives a new RREQ it responds with a unicast message to the source – Route Reply (RREP). During the transmission of the RREP, the path from source to destination (the forward path) is established. Also, note that in the meantime when RREQ and RREP are sent, the intermediate nodes set their paths to the source and respectively to the destination. AODV will not always find the best path in the network, because for path selection it uses the sequence number as the first criterion and the hop count as the second one. More than one RREQ message can be sent to find a path to a given destination, what has an impact on the AODV control traffic overhead.

In order to make a metric based route selection in AODV, it is necessary that the source node receives more RREPs with the same destination sequence number as the destination sequence number already stored in its routing table. Such an approach is possible if the destination node does not change its sequence number and sends more than one RREP. This simply implies that the source should send more than one RREQ to discover the route to every destination or that more copies of the same RREQ should reach the destination via different paths. In AODV, as long as the source does not have a route to the desired destination, it broadcasts a RREQ with a new identifier, even if the packets should be routed to the same destination. Of course, the maximum number of RREQs that can be sent per second is limited. The main drawback of AODV is that it calculates the paths considering only the hop count metric, which is more appropriate for wired networks than for wireless ones, in which many factors should be taken into account when finding a path. They include path self-interferences, interferences between the paths (flows), the quality of links, etc. [11]. Considering this main drawback of AODV, we propose a new routing algorithm – Adaptive AODV (AAODV) that is an improvement of the AODV protocol.

AAODV is able to calculate and simultaneously use multiple routing metrics. This process is supported by a Monitoring Layer (ML) that is independent on the routing protocol. The ML is responsible for the measurements and the calculations of metrics in a per link manner. The Monitoring Layer consists of a Metric Container (MC). The ML components are implemented in every node. Also, at each node, the metrics for every neighbor, i.e., delay, jitter and ETX are stored in its MC. The hop count metric is implicitly implemented. A new kind of HELLO messages (see Figure 1) is used for ML data dissemination.

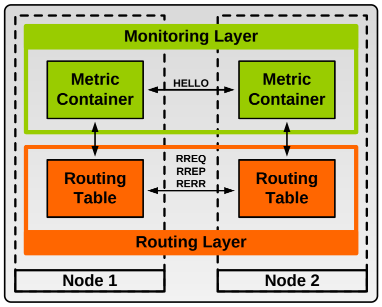


Figure 1 AAODV: Routing and Monitoring Layers

Each node sends the information about metrics for its neighbors (from the MC) in the HELLO messages, together with its node ID and timestamp. The timestamp determines the time, at which the HELLO was sent. Therefore, when a neighboring node receives the HELLO, it can calculate delay and jitter metrics. We assume that the nodes are synchronized. If the MC is empty (e.g., at the initialization of the network) the HELLO message will contain only the timestamp and an IP header. When a node receives a HELLO, it scans the message for any information addressed to it. If it finds this information it updates its MC. Afterwards, the node calculates the metrics to the source of the HELLO and updates the MC once again.

In AAODV the delay and jitter metrics are calculated according to the RFCs [17] and [18] using the HELLOs as probe messages. In our approach, the following exponential smoothing function (1) has been used for delay and jitter metrics estimation

$$d_{new} = \alpha * d_{old} + (1-\alpha) * d_{sample}, \quad \alpha \in [0;1] \quad (1)$$

where d_{old} – the old value of delay/jitter;
d_{sample} – the new sample of delay/jitter.

The ETX metric has been implemented using PLR, in the same way as in [5]. In this approach, the value of ETX per link represents in fact the probability of successful transmission of a packet, considering both the forward and reverse link delivery ratios. The following formula (2) is used for calculating per link ETX

$$ETX_{link} = P_{link} = d_f * d_r, \quad (2)$$

where d_f – forward link delivery ratio;
d_r– reverse link delivery ratio.

By default, the HELLO messages are generated every 2 seconds and the d_f and d_r are calculated by counting the successfully received HELLO messages at a node in the analyzed time window (20 s). The successful delivery per path must take into account the successful delivery on every link, therefore the path ETX is calculated as a product of link ETXs (3)

$$ETX_{path} = \prod_{i}^{n} ETX_{link_i}, \quad (3)$$

where n – number of links that constitute the path.

The path chosen from a source to a given destination in a network is the one with the highest ETX. Of course, the maximum value of the ETX is 1.

Note that the AAODV protocol is not limited to these metrics and can be implemented with any other per link calculated metric. As it was mentioned before, AAODV is based on AODV and it implements the same algorithm, which uses the sequence numbers in order to obtain loop-free paths. The main differences from AODV are as follows:

- AAODV nodes do not flood the network with RREQs when they are searching for a new route;
- AAODV nodes do not discard all the duplicate RREQs – this idea is also presented in [14].

In AAODV every time a node receives a packet, for which it does not have a route to the destination, it queues the packet and sends RREQ. If a new packet is received and needs to be routed to the same destination, the node checks two additional conditions in comparison to AODV, before it sends a new RREQ. First it checks whether another packet to the same destination exists in the queue. Then it checks if the RREQ for the packet already existing in the queue has expired or not. In the case that another packet to the same destination already exists in the queue and if the timer for the RREQ has not expired, the node does not send a new RREQ. This way, the RREQ flooding, evident in AODV, is inhibited and the overall overhead is decreased.

Moreover, as it was stated before, when an AAODV node receives a duplicate RREQ, it does not discard it immediately. The node verifies the sequence number and then checks if the metric for the path advertised in the RREQ is better than the one already existing in its routing table. If this condition is met, the node will update its routing table with the path from the RREQ, otherwise it will discard this RREQ.

Every time an AAODV node receives a RREQ, it takes the actions from the flowchart presented in Figure 2. Note that the RT entry refers to the path (stored in the routing table) to the RREQ's originator. RREQ entry refers to the path (indicated in the RREQ) to the RREQ's originator.

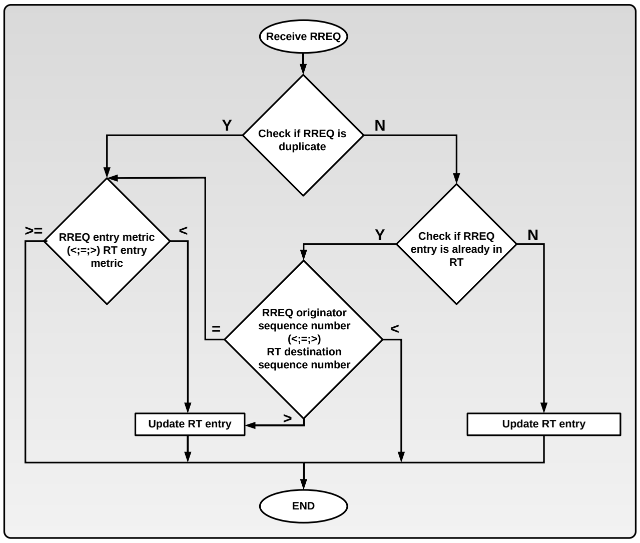

Receive RREQ
Y
Check if RREQ is duplicate
N
>=
RREQ entry metric (<;=;>) RT entry metric
<
Y
Check if RREQ entry is already in RT
N
=
RREQ originator sequence number (<;=;>) RT destination sequence number
<
Update RT entry
>
Update RT entry
END

Figure 2 Flowchart for implementing RREQ packet analysis

The algorithm depicted above proves the easy extendibility of AAODV to multipath, if instead of replacing the entry in the routing table, we will add it. This way, we will obtain more than one route to the source and RREPs will be sent on each of these paths. Multipath routing becomes effective only if the paths towards the destination are either node or link disjoint. In order to achieve this, several additional conditions should be considered.

IV. AAODV Simulations

In this section we compare the impact of the common routing metrics: hop count, delay, jitter and ETX on the network performance using ns-3 simulator.

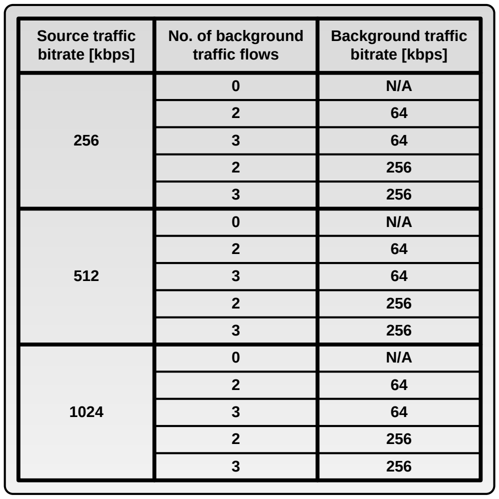
The simulated nodes were equipped with 802.11b Wi-Fi cards. The nodes used adaptive link rate that varies link bitrate from 1 Mbps to 11 Mbps. The topology was discovered by the nodes using HELLO messages. HELLO messages were broadcasted every 2 seconds, at the basic rate of 1 Mbps.

The simulation scenarios were based on eight randomly generated topologies. The nodes were distributed randomly in a square area. In order to be sure that the random topologies do not consist of isolated nodes, the possibility to communicate between any pair of nodes in the network was verified.

We considered two network sizes:

- Case 1: 16 nodes distributed uniformly in a square of 250 m x 250 m. The nodes are numbered from 1 to 16.
- Case 2: 25 nodes distributed uniformly in a square of 300 m x 300 m. The nodes are numbered from 1 to 25.

The application, which we used to measure the network performance, generated the Constant Bit Rate (CBR) traffic flow between the first node (node 1) and the last but one node (node 15 for Case 1 or node 24 for Case 2). The background traffic, Variable Bit Rate (VBR) flows, was generated between any two randomly chosen nodes that were different from the application source or the destination. The start time of the source traffic was fixed at 70 s and the start time for the background traffic was randomly chosen in the interval 40-50 s. The inter-packet interval deviation of the background traffic was equal to 1 μs. In Table 1 we summarized all the traffic simulation scenarios.



Source traffic bitrate [kbps]	No. of background traffic flows	Background traffic bitrate [kbps]
256	0	N/A
	2	64
	3	64
	2	256
	3	256
512	0	N/A
	2	64
	3	64
	2	256
	3	256
1024	0	N/A
	2	64
	3	64
	2	256
	3	256

Table 1 Traffic patterns

For each of the eight random topologies, we monitored PLR, delay and jitter of the source traffic using the traffic patterns from Table 1. In total we run over 1200 different simulations.

In Figure 3, we depicted the PLR, average delay and average jitter for the first five traffic patterns presented in Table 1 (source bitrate set to 256 kbps). We limited the delay and the jitter scales to 250 ms and 50 ms respectively. Note that the legend presented in Figure 3 is common for all the figures that follow. The averaged results from eight topologies show that AODV is outperformed by all variants of AAODV. Evaluating the network topology impact on performance, we observed that PLR, delay and jitter of source traffic can increase 5 to 6 times from one topology to another. In some cases the paths found by ETX are 2 or 3 times longer than the ones found by hop count. Despite that fact, we obtained similar results of averaged PLR, delay and jitter, regardless of the metric used. The paths discovered with AAODV and ETX have fewer retransmissions, but the length of the path affects the throughput. This leads to the conclusion that a combination between ETX and hop count should yield better results.

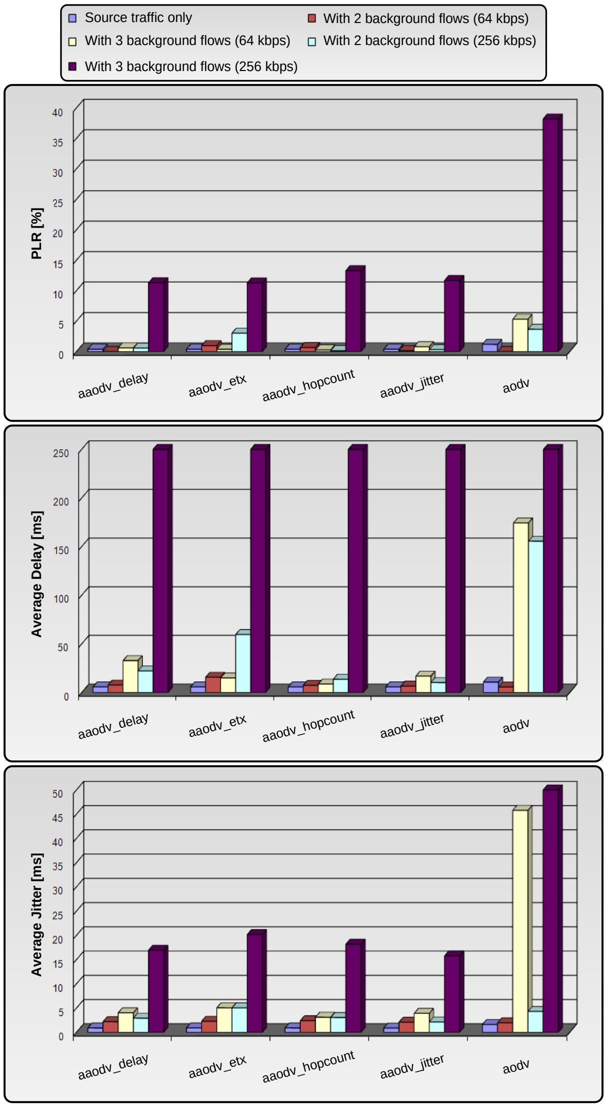


Figure 3 PLR, average delay, average jitter for the source traffic (16 nodes, 250 m x 250 m, 256 kbps)

In the next step we increased the source traffic bitrate to 512 kbps (topologies were the same). In Figure 4, we depicted the PLR, which slightly increased.

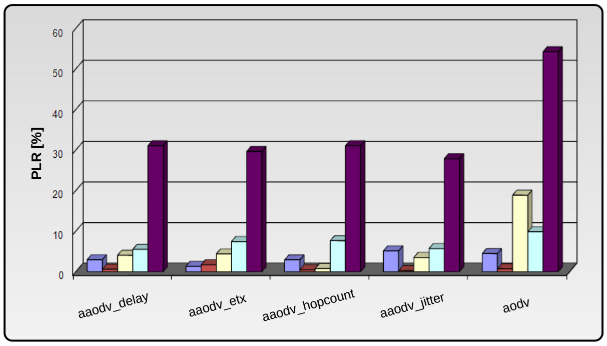


Figure 4 PLR of the source traffic (16 nodes, 250 m x 250 m, 512 kbps)

We observed that in this case the delay has dramatically increased, especially when we generated the background traffic with a bitrate of 256 kbps. The detailed analysis has shown that the source traffic delay increased very much in all cases in which the background traffic shares the paths. Note that the packets, which we used for metric calculation, were much smaller than the ones generated by the source.

In the last case, in which the source traffic was set to 1 Mbps, PLR, delay and jitter values increased more than in the previous cases. In Figure 5 we present the PLR graphic for this case. For all metrics, the delay exceeds our imposed limit of 200 ms.

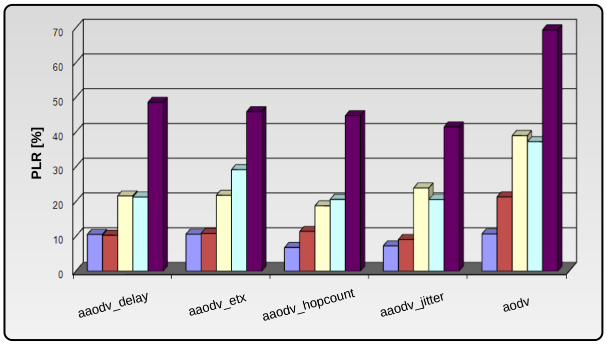


Figure 5 PLR of the source traffic (16 nodes, 250 m x 250 m, 1024 kbps)

In the second topology, in which we used 25 nodes distributed uniformly in a 300 m x 300 m square, the interferences have increased and more drastically affected the performance of the network.

In Figure 6, we depicted PLR, average delay and average jitter for the first five traffic patterns presented in Table 1, i.e., when a 256 kbps source bitrate was set. As before, we also limited the scale for the delay and the jitter to 250 ms and 50 ms respectively. Further, if we increased the source bitrate, PLR, delay and jitter have also increased until the network became unusable (according to our criteria).

The AAODV results have shown that the metrics calculation method and its usage (i.e., to determine the cost of the path) can be wrong. Firstly, the metric is used during path setup phase only, and it is calculated on a per link basis. Additionally, the metrics were calculated using HELLO messages transmitted at 1 Mbps. A direct consequence of this is that the metrics are not aware of, and do not consider the real throughputs available per links. This drawback is resolved in [19] where the HELLO messages are sent using adaptive bitrates. It also appears that routing metric estimation has to take into account the physical layer phenomena, and the path quality estimation cannot be limited to per link operations only, but should take into account the whole interference area of nodes, which constitute the path. A special care should also be given to the choice of measurements repetition frequency.

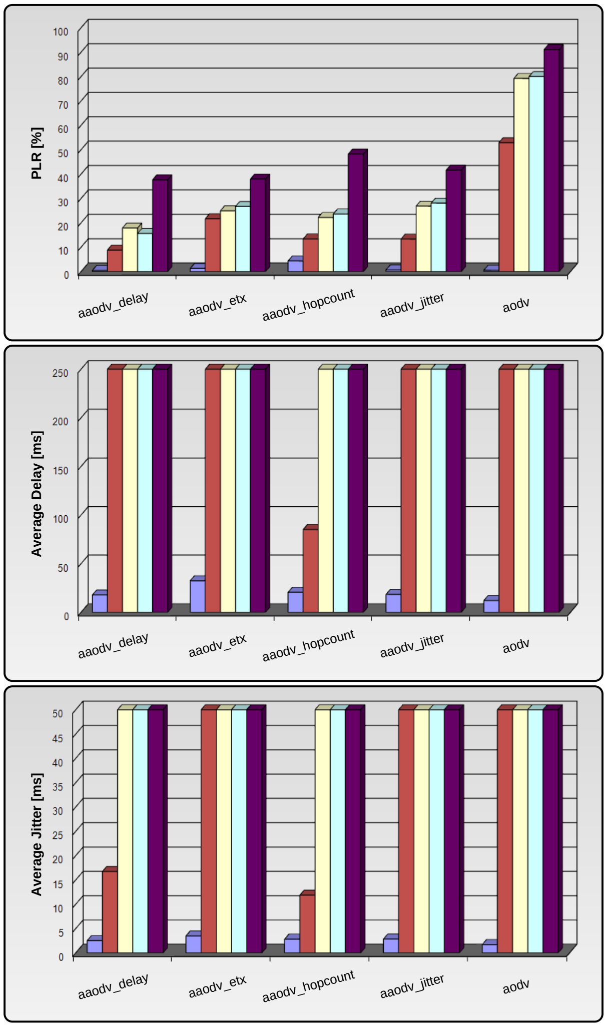


Figure 6 PLR, average delay, average jitter of the source traffic (25 nodes, 300 m x 300 m, 256 kbps)

V. AAOMDV

The previous section has shown that none of the metric behaves better than the other ones in case of single path routing. Considering this result, we propose a new routing protocol, named AAOMDV. It enables discovering of multiple paths between a source and a destination. By using RTM, AAOMDV is able to detect the degradation of the quality of the data forwarding path, and to trigger a path change based on algorithms presented later in this paper. Moreover, the functional separation of routing and monitoring mechanisms in AAOMDV make it more scalable and flexible.

A. *Paths disjointness*

As stated above, AAOMDV is able to find multiple link disjoint paths in the network. Please note, that the disjointness property refers strictly to the set of routes established between the same source and destination pair. The routes between different source and destination nodes can share the same links. In order to enhance the network performance, it is desirable for the traffic flows to follow paths that do not have many common links. In AAOMDV, this is accomplished by the usage of active path selection and switching algorithms described in Sections VI and VII.

In order to be link disjoint, the paths should fulfill two conditions indicated in [14]:

- For every created path the next hop must be different;
- The last hop towards the destination must differ from path to path.

B. *Paths discovery algorithm*

The paths discovery algorithm in AAOMDV is based on the Route Request – Route Reply (RREQ/RREP) messages exchange, present also in AAODV. The only difference is that in AAOMDV these messages contain also the information about the last hop on the path in order to be able to achieve the disjointness property.

Every time a node wants to send a packet and it does not have any available path to the destination in its routing table, it sends a RREQ message via the broadcast channel. Note that, like in AAODV and in contrast to AODV, the node sends only one RREQ. If no RREP has been received after a determined period of time, another RREQ is broadcasted. During the RREQ propagation, the reverse paths from the destination and the intermediate nodes are set up to the originator of this RREQ. The flowchart depicted in Figure 7 shows how the multiple paths are established during the RREQ broadcasting.

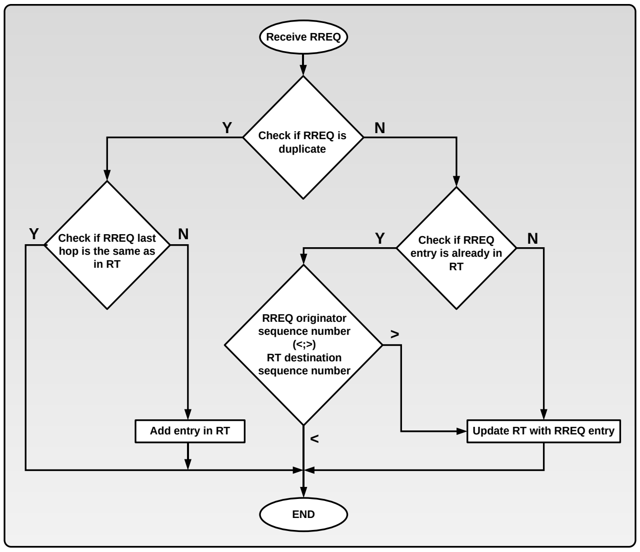

Receive RREQ
Check if RREQ is duplicate
Y
N
Check if RREQ last hop is the same as in RT
Y
N
Check if RREQ entry is already in RT
Y
N
RREQ originator sequence number (<;>) RT destination sequence number
>
<
Add entry in RT
Update RT with RREQ entry
END

Figure 7 AAOMDV paths discovery algorithm – RREQ broadcasting

The information about the last hop on the path allows accepting or rejecting the newly obtained path. This information refers to the penultimate node on the path towards the RREQ originator. Every established path has to have a unique last hop and next hop addresses, and this way the link disjoint paths towards the originator of the RREQ are established. Moreover, this approach also helps in loop avoidance.

AAOMDV can be easily adapted to support the node disjoint mode. The node disjointness property would be enabled through the rejection of the duplicated RREQ messages in the intermediate nodes. However, in a small WMN the probability of establishing multiple node disjoint paths between the same source and destination pair is quite low. For this reason, the adaptive selection of the mode can be a desired solution, i.e., if the network density is big enough and the connectivity between nodes is relatively high, then only the node disjoint paths should be allowed. This way the communication reliability would be improved. On the contrary, if each node has only few neighbors, then only the link disjoint path should be allowed.

As in AAODV, the sequence number mechanism is used to prevent the nodes from keeping the obsolete information in their routing tables. The loops can be caused by the acceptance of all the duplicated RREQs and by keeping the stale paths in the routing table. AAOMDV avoids both these situations. All the paths to the particular destination must have the same sequence number and if the one with a higher number would be found, all the previously established paths have to be deleted. When the node receives the RREQ with the same sequence number, it verifies the quality of this path and compares it with the others in order to keep the best available ones. The number of additional paths is limited. In Figure 8 an example of setting up multiple paths is presented.

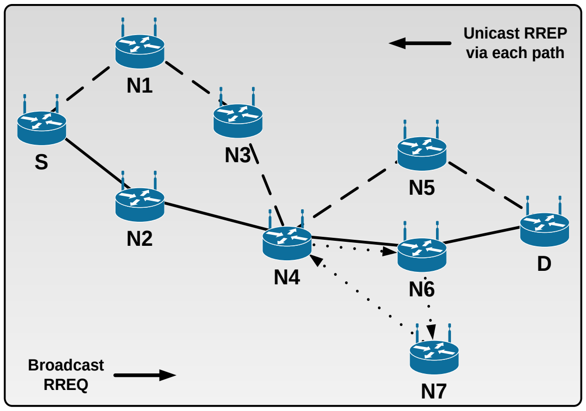
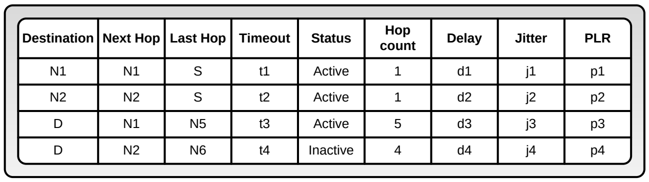


Figure 8 Setting up multiple paths

In order to establish multiple paths from the source to the destination and at the same time from the intermediate nodes to the same destination, unicast RREPs are sent on each of the paths established using the RREQ broadcasting. As we consider link disjoint paths and not node disjoint paths, an intermediate node can have more than one path to the originator of the RREQ. Hence, when an intermediate node receives the RREP, it should know onto which path this RREP message is to be forwarded. The last hop information helps to distinguish between all the paths. Please note that even if AAOMDV is implemented as a link disjoint multipath routing protocol, it is possible that it discovers only node disjoint paths. This drawback is explained in Section C.

According to Figure 8, source S broadcasts RREQ every time it searches for paths to destination D. During the RREQ propagation phase, the intermediate nodes complete their routing tables with paths towards S. These paths have different last hops (N1 and N2). When D receives the RREQ, it has to respond with a corresponding RREP. As it can be seen from Figure 8, N4 has two paths to S and when the RREP reaches N4 through the path drawn by the dashed line, the RREP should be forwarded to S on the same path. This is accomplished using last hop information to S, which is represented by N1. The dotted arrows in Figure 8 show a possible loop due to acceptance of a duplicate RREQ. N4 receives the RREQ, accepts it, completes its routing table with a path to S (drawn by a solid line) and rebroadcasts the RREQ. Node N6 receives it and performs the same operations as node N4. At the end, the RREQ will reach N4 again from node N7. Node N4 will reject this duplicate RREQ as the last hop (N2) is the same. In result, the loops are avoided. Note that multiple paths can also be set in intermediate nodes.

In Figure 9 we depicted the routing table available at node S after the route discovery procedure. As it can be seen, for every path we have more than one metric associated. The metrics are used to determine the quality of the paths as described in Section VI. Every path is associated with its own timeout that is updated when the path is used. Note that this timer will not be updated in the intermediate nodes if the path is not used.



Destination	Next Hop	Last Hop	Timeout	Status	Hop count	Delay	Jitter	PLR
N1	N1	S	t1	Active	1	d1	j1	p1
N2	N2	S	t2	Active	1	d2	j2	p2
D	N1	N5	t3	Active	5	d3	j3	p3
D	N2	N6	t4	Inactive	4	d4	j4	p4

Figure 9 The routing table structure at node S

In fact, all the paths will have the timer updated in all the nodes including the intermediate ones, because the probe messages will be sent via the inactive paths, as it is described later.

C. Link disjointness problem

The topology shown in Figure 10 is considered.

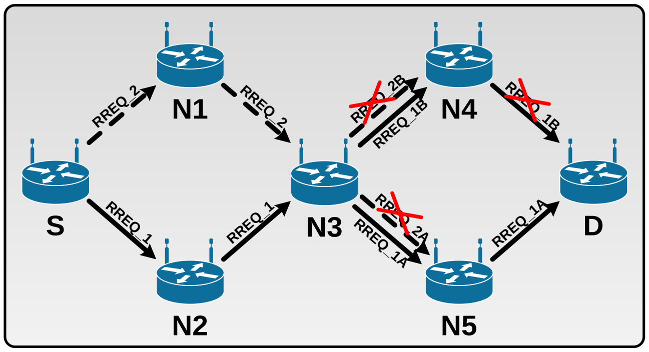


Figure 10 Link disjointness problem

Considering Figure 10, it should be possible to find two link disjoint paths from source S to destination D, namely [S, N1, N3, N4, D] and [S, N2, N3, N5, D]. However, this is not possible in some situations. At the beginning, node S broadcasts a RREQ message. The packet processing time at every node may be different and because of that the RREQ copies arrive at different times in the intermediate nodes. The first copy (RREQ_1) goes through N2 and in N3 it is forked. RREQ_1A reaches the destination D through N5, the first path [S, N2, N3, N5, D] is discovered and the RREP is sent on it. RREQ_1B gets to D via N4, but it must be discarded, as the last hop N2 is common for the already added path. In the meantime RREQ_2 arrives to N3 through N1. N3 adds the second path to S in its routing table and rebroadcasts RREQ_2. It is received in N4 and N5 nodes, but both of them must reject it, while in both cases it does not create a new link disjoint path. This happens, because RREQ_1 is still buffered and the first hop to S, i.e., N3 is the same for both RREQ copies. The result is that only one path may be established.

The described problem shows that in some cases it is hard to find more link disjoint paths. This is a drawback in a small network, as it becomes very difficult to find multiple paths between source and destination. One solution for this problem may be the reduction of the time during which the RREQ is buffered in nodes. We also expect that in more dynamic scenarios it would be possible to achieve link disjoint paths, e.g., when RREQ_1B copy will be lost between N3 and N4.

D. Routes Limit Validation

The wireless medium is highly unstable and the transmission quality changes in time. Due to the mobility of nodes, frequent topology changes may occur. These are the reasons to store a limited number of backup paths in the routing table. This way, the content of the routing table is more recent. After a new path is added to the routing table, AAOMDV verifies whether the number of paths does not exceed the maximum number permitted, defined by the configurable parameter – *routes_limit*. As mentioned before, it has been proven in [14] that the gain of a multipath is achievable with two/three paths for one destination. If the limit is exceeded, then the quality of all the paths is evaluated using the algorithm described in Section VI and the worst one is deleted from the routing table.

E. Enhanced Monitoring Layer – Real Time Monitoring

The AAOMDV nodes can have more than one path to a destination, but only one is used for data forwarding – in the routing table it has the active flag set (see Figure 9). During the route discovery phase, the nodes activate the first path that they obtain towards the destination. Note that at this step the nodes do not consider any kind of metric – the nodes start routing the data packets from the queue as soon as they obtain the first path to the destination. After a determined period, starting from the first RREP, the source sends a message named Route Activation (RACTV, see Figure 11) to activate the path to the destination that has been chosen as the best one using the path selection algorithm from Section VI (INITIAL_MODE). The mentioned period is named MULTIPATH_DISCOVERY_TIME and has a default value of 2 seconds.

The Enhanced Monitoring Layer (EML) of AAOMDV incorporates all the functionality of the AAODV ML and has some new features. The most important feature of the EML is the capability to monitor in real-time regime the multiple end-to-end paths available between the source and the destination. Using the traffic packets sent from the source to the destination, it is possible to monitor PLR, delay and jitter of the active path. Probe messages are sent in order to evaluate delay and jitter of the inactive paths.

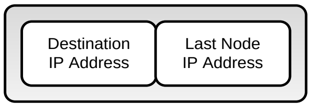


Figure 11 RACTV header

The ETX can be used to substitute the PLR metric on the inactive paths. Note that the evaluation of PLR, delay and jitter on the active path is realistic, as it considers the real traffic. The delay and jitter evaluation on the inactive paths is done by using active probing and it does not reflect the real delay and jitter, which would be achieved if we routed the data traffic on them. The probe messages, called Route Probes (RPRBs), sent on the inactive paths, have the same payload size as the averaged payload size of the data packets transmitted during the last 5 seconds between the source and the destination through the active path. Once again, it should be noted that although the size of the probes is appropriately matched, the transmission of only two packets cannot emulate the real flow of packets. Therefore, the results for the delay and jitter are estimative. The RPRB message header is depicted in Figure 12. The delay and jitter are calculated according to [17] and [18].

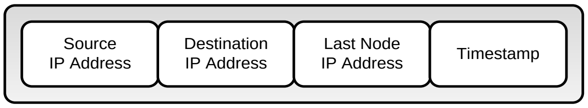


Figure 12 RPRB header

The information about all the paths between the source and the destination is evaluated at the destination and sent back to the source using a Route Report message (RRPRT), shown in Figure 13. The size of this message is variable since the number of paths obtained between the source and the destination, although limited, is not fixed.

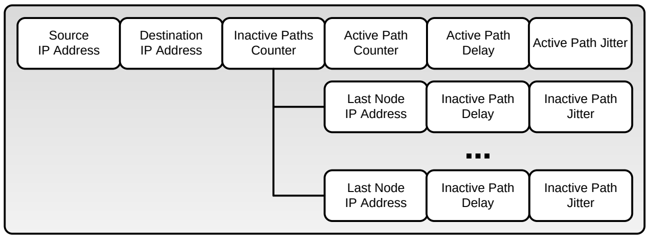


Figure 13 RRPRT header

The Active Path Counter represents the number of packets successfully received by the destination during the last 5 seconds interval. When it receives the RRPRT message, the source can calculate the PLR for this period as it knows how many packets it has sent. In fact, all the information known at the source about the end-to-end paths is 5 s old. The delay and jitter are evaluated at the destination and sent back in the RRPRT messages to the source. The sequence of the messages exchange implemented in the EML for the RTM is depicted in Figure 14.

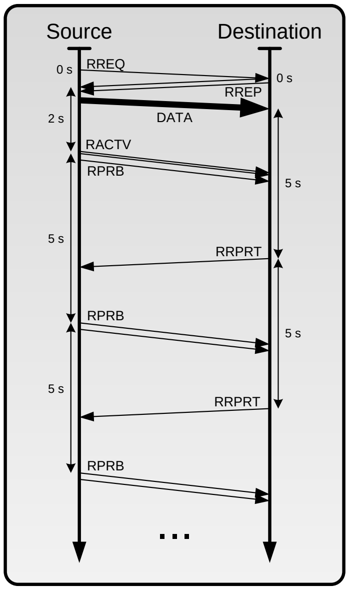


Figure 14 The exchange of messages in EML

In order to take full advantage of the multipath feature and of the RTM, it is necessary to determine the best path and to send the traffic using this path. The path selection and switching algorithms are protocol independent, and may be applied to any multipath routing protocol that does not use all the known paths simultaneously for data forwarding. Both algorithms will be described in the following sections.

VI. Active Path Selection

The active path selection algorithm should be implemented in each node in the network. Two modes of the active path selection algorithm that can be distinguished are called INITIAL_MODE and NORMAL_MODE. The first one is used to select the best path of the available ones discovered during MULTIPATH_DISCOVERY_TIME. At the beginning of the communication, the end-to-end real metrics are not available, so the per link calculated metrics, i.e., delay, jitter, ETX and hop count are considered. The second mode of the algorithm is used when a new RRPRT message is received at the source node. In this case, the real metrics, i.e., the real delay, real jitter, PLR and hop count are considered. AAOMDV deals with multi-parametric metric and it needs a special algorithm to compare the quality of two or more paths and to select the better one.

Let $M = \{m_1, m_2, m_3, ..., m_k\}$ be the set of metrics related to every path. $W = \{w_1, w_2, w_3, ..., w_k\}$ is the vector of weights, which express the importance of every particular metric.

The algorithm for paths comparison utilizes all the metrics M **simultaneously** to evaluate and confront the two paths with each other. At the initial stage of the algorithm all the metric weights are defined. These weights allow differentiating the importance of the metrics. This way it is possible, e.g., to favor the paths with a smaller packet loss level by assigning a higher value for the weight related with the PLR metric. The weights are set with respect to the application needs. Here we consider also the hop count metric. Hop count metric should be considered when building a metric as the throughput achievable in an arbitrary WMN is proportional to $\Theta(W \cdot n^{-1/d})$, where d is the dimension of the network, n the number of nodes and W is the total bandwidth. In a two dimensional network, the throughput can be as small as $\Theta(W \cdot n^{-1/2})$ [20]. The hop count metric does not cause any implementation problems, because it is already used by the AODV protocol. The next parameters that must be defined are the threshold values for all the metrics considered in the algorithm. A path that has a metric, which exceeds its threshold, is considered the worst path. If all the available paths to a destination are considered as the worst path, it is desirable to send a new RREQ to the destination and establish new paths (of course, if this is possible). Both sets of parameters, i.e., weights and thresholds, may be configured and adjusted according to operator or user requirements, e.g., in the policy based approach. After these steps, the algorithm is ready to compare the paths. For all the comparisons the Composite Metric is calculated. The paths are compared two by two. For calculating the Composite Metric formula 3 applies:

$$Composite\_Metric_{p1} = \\ = \frac{d_{p1} \cdot w_d}{d_{p1} + d_{p2}} + \frac{j_{p1} \cdot w_j}{j_{p1} + j_{p2}} + \frac{p_{p1} \cdot w_p}{p_{p1} + p_{p2}} + \frac{h_{p1} \cdot w_h}{h_{p1} + h_{p2}} \qquad (3)$$

where:

d, j, p, h – delay, jitter, PLR and hop count values;

w_d, w_j, w_p, w_h – weights correlated with metrics.

Formula 3 is applicable when the algorithm is used in NORMAL_MODE, i.e., when real metrics can be used. In INITIAL_MODE per link calculated metrics are used and the component associated with the PLR must be replaced by a corresponding component calculated for the ETX metric. The ETX indicates the packet delivery ratio and the path with a higher ETX value is considered to be better, therefore the ETX component must be subtracted from formula 3. In the NORMAL_MODE, the ETX is also used instead of the PLR in the same way for the evaluation of the Composite Metric for inactive paths. In order to improve the stability of the network and to give a priority to the currently active path, the algorithm defines one more configurable parameter, i.e., ACTIVE_PATH_MARGIN. If the evaluated path is active, then this parameter specifies how much better (in percent) the inactive path should be in order to replace the active one. The default value of this parameter is 10%.

In the case when two paths have equal Composite Metrics, the first one is indicated. This way the changes in the network configuration are avoided, because the currently active path is always preferred as it is the first one, when compared with any other route.

VII. Active Path Switching algorithm

The challenge appears when two paths are created between different pairs of source and destination nodes, but parts of these paths are common for both of them. It is also possible that the backup paths overlap. In this scenario, if both sources start to send the traffic via the common links, then the active paths performance degrades; therefore both nodes will switch to the second available path. The situation may repeat, causing the so-called flip-flop phenomena and will lead to oscillations in the network configuration. To overcome this challenge, an algorithm that controls the active paths switching from node to node is needed. The algorithm can be either centralized or distributed. In the centralized approach, the switching should be controlled by a central entity. In the second approach, the decision of path switching is distributed among nodes. In this paper we will implement a distributed algorithm for the path changing. The distributed approach is a more scalable solution. No central node is needed, so it is possible to use it regardless of the network size, which can be understood as the number of nodes, as well as the spatial extent. Nodes are able to autonomously adapt to the changes in the network and make autonomic decisions according to the results of the performance measurements. Their decisions are taken locally, but finally it should lead to the global optimization of the network configuration.

The proposed algorithm works as follows. All the source nodes that are currently sending data to their destination nodes are aware of the quality of each path to the destination stored in their routing tables. This knowledge is obtained using monitoring of both active and inactive paths. As described previously, the active path monitoring is piggybacked in the data packets and the RPRB messages are used to evaluate the inactive ones. The source nodes take their decisions based on the periodically received RRPRT messages from the destination nodes. Every time the source node receives a RRPRT message, it updates its routing table with fresh measurement results, compares all the available routes with one another and chooses the best one. If it is the same as the currently active path, nothing happens. If another path is chosen, the algorithm starts working. First, a random number (*random_number*) with uniform distribution is chosen from a range of [0; 100]. In the algorithm a configurable threshold value (*change_threshold*) is defined and if the *random_number* is smaller than *change_threshold* nothing changes and the next RRPRT message is awaited. In the opposite case, the node makes the better path active, deactivates the previously active one, applies these changes in its routing table, sends a RACTV message on the new active path and starts to send data using it. The *random_number* and the *change_threshold* values have critical impact on the algorithm behavior. Their task is to avoid the continuous changes of the active path and to stabilize the protocol functionality. It is possible that the *change_threshold* value may be inaccurate and the *random_number* may be always smaller. In this case the performance of the network will be weak, although it could be simply improved by changing the active path. For this reason a new parameter (*change_limit*) has been defined that determines the limit value of the path change cancellation. The consecutive unsuccessful attempts to switch the path are calculated. If their number exceeds the *change_limit* value, the node changes the active path regardless of the *random_number* value. The election of distribution type and threshold values has a great impact on the network performance (see Section VIII). Figure 15 shows the structure of the distributed algorithm.

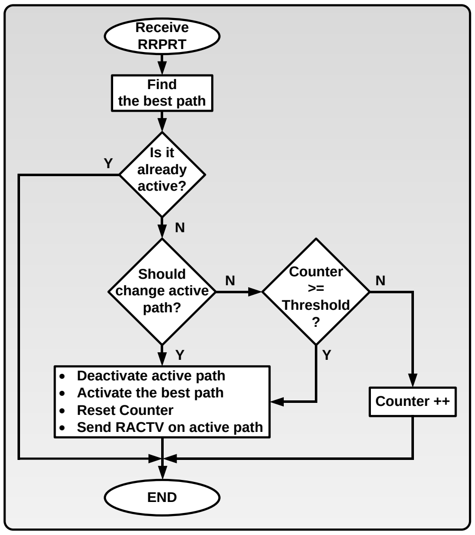


Figure 15 Distributed algorithm for active path switching

VIII. AAOMDV Simulations

In this section the performance of AAOMDV is compared to AODV. The comparison was made in a network that consisted of 16 nodes, randomly located in a square area of 300 m x 300 m. Each node was equipped with IEEE 802.11b Wi-Fi card and the communication range was ca. 175 m. We run the simulations with three different random number generator seed values, in order to get a different placement of nodes. The obtained results were averaged. In order to be sure that the random topologies do not consist of isolated nodes, the possibility to communicate between any pair of nodes in the network was verified.

The source traffic was generated between a specific pair of nodes, while the background traffic sources and destinations were chosen randomly. The maximum number of paths in the routing table for a specific destination has been set to 3.

As reference for the AAOMDV performance evaluation we used AODV. To check the behavior of both protocols with a different network load we changed the source and the background traffic throughput and the number of background traffic flows. In AAOMDV simulations we additionally checked the influence of the *change_threshold* and the *change_limit* parameters on the performance of the network. The simulation scenarios of AODV/AAOMDV are presented in Table 2.

Source traffic bitrate [kbps]	No. Of background flows	Background traffic bitrate [kbps]	change_threshold	change_limit
128 256 512	0	-	0 25 50 75 100	0 1 2 3 10 100
	1 2	64 128 256		

Table 2 Simulation parameters

We performed about 2000 simulations to obtain the results.

Firstly, we verified the proper behavior of AAOMDV using PyViz visualiser [21], which is a part of the ns-3 simulator. Figure 16 shows an example of PyViz output. It can be observed that between the same source and destination two different paths have been established by AAOMDV.

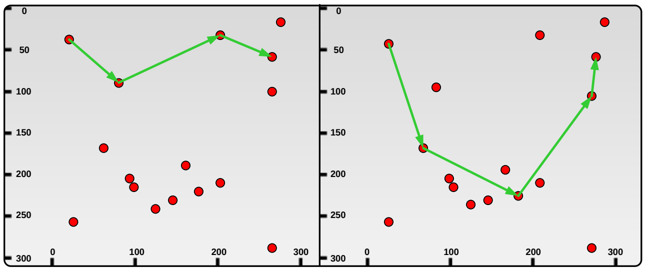


Figure 16 PyViz output – two paths found in the network

Figure 17 shows the influence of the AAOMDV *change_threshold* and *change_limit* parameters on PLR, average delay and average jitter, when only the source traffic was generated in the network. The detailed analysis of the mutual interdependencies between the above mentioned AAOMDV parameters led us to the following conclusions:

- In order to permit nodes to change their active path frequently, both parameters should be set to a low value;
- If both parameters have relatively high values then it is very hard to switch the active path;
- A node is always restricted from path changes if the *change_threshold* is set at 100% and the *change_limit* is set much higher than 0 (e.g., 100). The initially chosen path will be used until it gets lost;
- A node is always permitted to change the path if the *change_threshold* is set to 0% or the *change_limit* is set to 0. It means that no postponing of the path change is allowed.

Therefore, both parameters are dependent on each other and their values must be correlated in order to obtain the desired configuration.

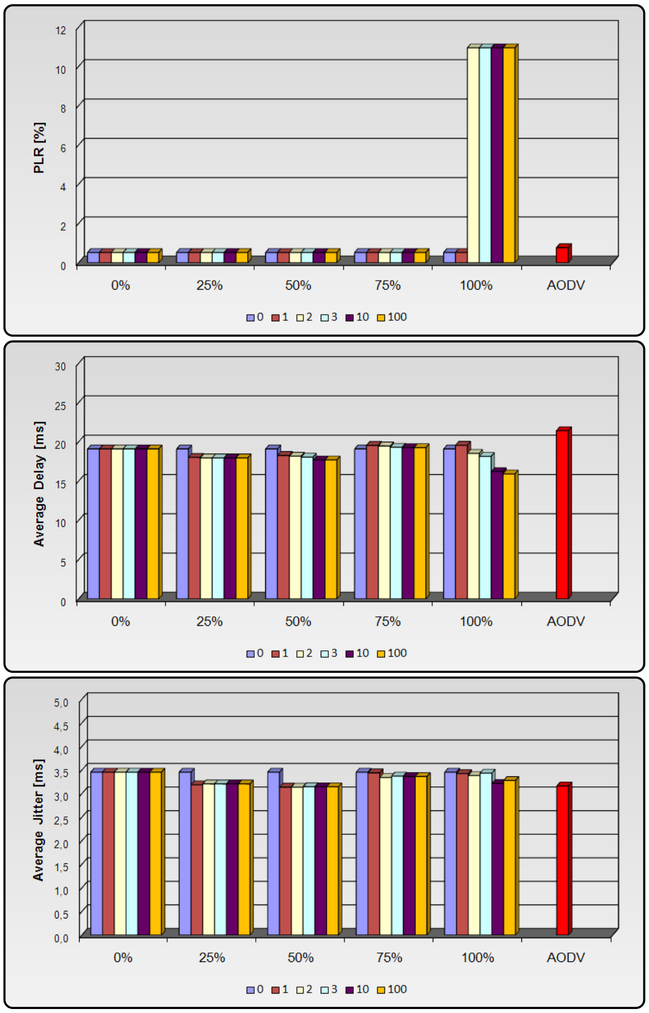


Figure 17 PLR, average delay, average jitter of the source traffic (source traffic – 128 kbps, no background traffic)

It can be observed that the frequent changes of the active path were advisable for improving the quality of transmission. If switching of the active path was not permitted, the traffic packets started to be lost. On the other hand AODV, which is a single path protocol based on the hop count metric, yielded good overall results. This means that when the network load was low, the Composite Metric did not outperform the hop count metric. When the throughput of the traffic increased and the network load grew, the nodes started to switch the paths more frequently (see Figure 18).

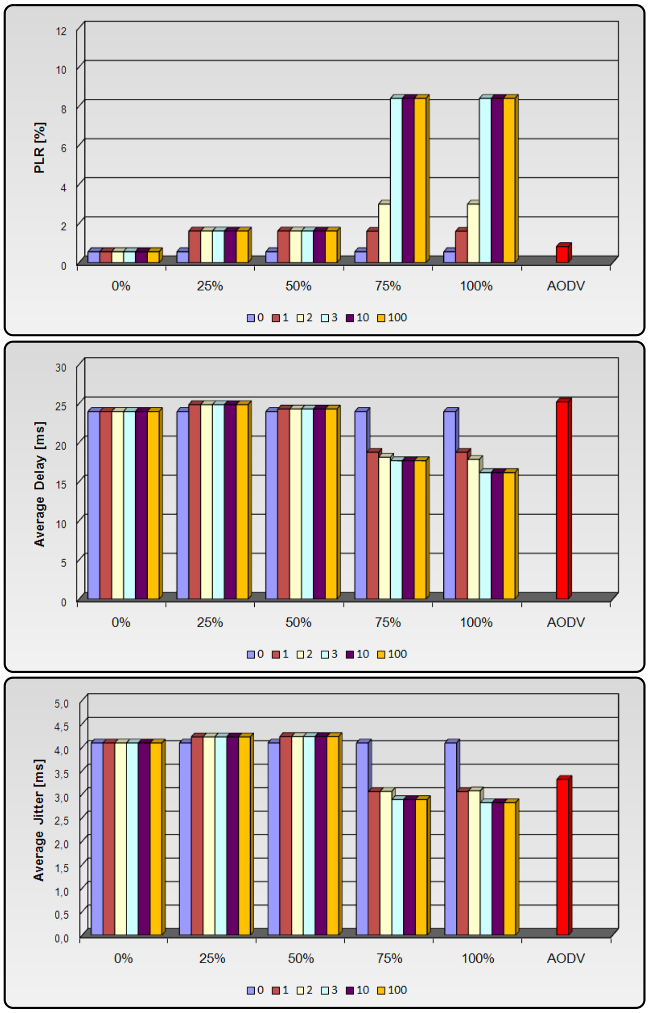


Figure 18 PLR, average delay, average jitter of the source traffic (source traffic – 256 kbps, no background traffic)

Figure 19 shows that in some cases the AAOMDV routing protocol was able to outperform the AODV significantly. In all simulated cases the AAOMDV PLR was at least three times lower than the AODV PLR. The delay and jitter had acceptable values in both cases. As we mentioned before, very high *change_threshold* and *change_limit* values cause that the first obtained path is used until it gets lost regardless of its parameters. Therefore, this configuration shows the impact of the routing metric used on the network performance. The conclusion is that when the network load increases, the Composite Metric provides better results than the hop count metric. It should be noted that when the traffic generated in the network increased, it became more viable to limit the number of active path changes. Similar results are also depicted in Figure 20. In this case, AAOMDV also outperformed AODV, although the PLR was a little bit higher. The obtained results have confirmed that the two parameters, i.e., *change_threshold* and *change_limit* have a great impact on the overall network performance, and that permanent monitoring of the paths and the dynamic paths switching provides better results than usage of AODV.

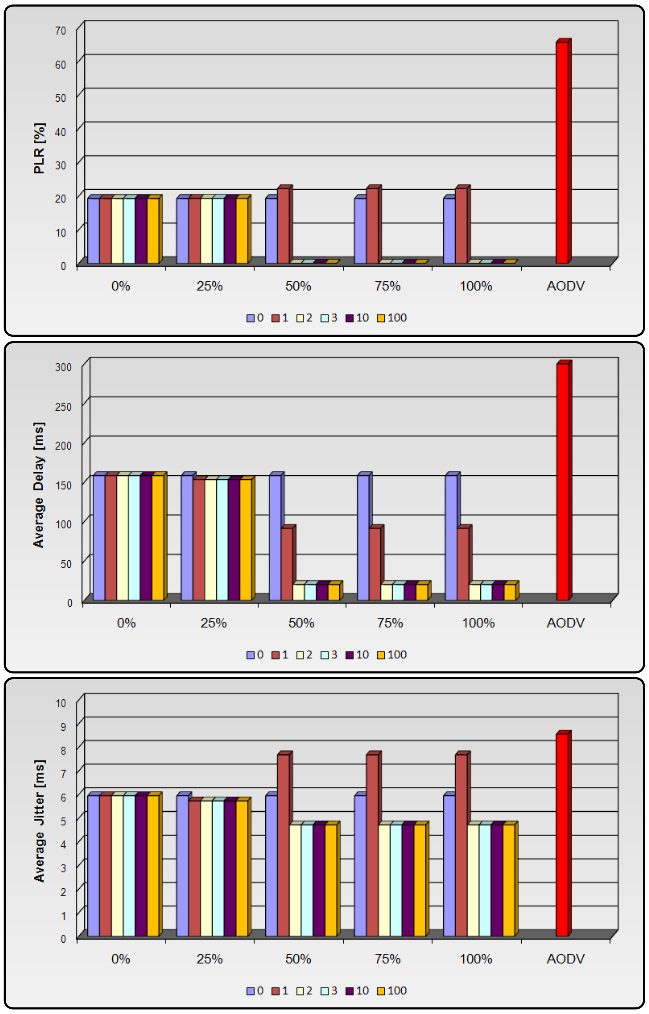


Figure 19 PLR, average delay, average jitter of the source traffic (source traffic – 512 kbps, one background traffic – 64 kbps)

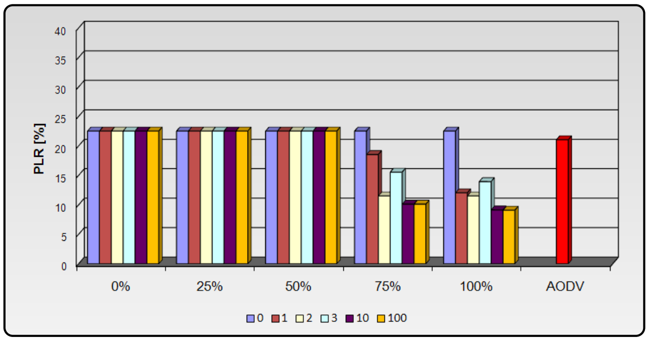


Figure 20 PLR of the source traffic (source traffic – 256 kbps, one background traffic – 128 kbps)

IX. CONCLUSIONS AND FUTURE WORK

The starting point of this paper was a comparison between the most popular routing metrics, i.e., hop count, delay, jitter and ETX, and our main goal was to determine the best one to be used in WMNs. To achieve that efficiently, we designed a new protocol AAODV, able to calculate the path cost based on more than one metric. AAODV, due to the separation of routing from paths monitoring, can use any per link calculated routing metric. The results of simulations led us to the conclusion that none of the analyzed metrics behaves significantly better than the others. This conclusion was in a sense predictable as none of the tested metrics is traffic aware or fully addresses the challenges that appear in a wireless environment, e.g., interflow and intraflow interferences, exposed and hidden terminal problems, etc. The evident problem of AAODV is that the path quality is only monitored during the path setup phase. This is why we decided to add the multipath capability to AAODV and enable continuous end-to-end monitoring of all the paths. In order to find the best path we defined the Composite Metric that takes into account PLR, delay and jitter weighted appropriately to network operator preferences. In AAOMDV a distributed path switching algorithm has been implemented and the Composite Metric is used for the active path selection.

The benchmark for the AAOMDV performance evaluation was AODV. It has been observed that when the network load was low both AAOMDV and AODV yielded good, similar results. It cannot be affirmed that one of the two routing protocols outperforms the other. However, when the network load increased, AAOMDV outperformed AODV by providing about two times smaller PLR and delay of the analyzed traffic.

A weak point of AAOMDV is the necessity of fine tuning of the *change_limit* and the *change_threshold* parameters in order to optimize the network performance. This procedure should be modified in order to have self-tuning properties and we will focus on it in our future work.

REFERENCES

[1] H. Ştefănescu, M. Skrocki, and S. Kukliński, AAODV Routing Protocol: The Impact of the Routing Metric on the Performance of Wireless Mesh Networks, The Sixth International Conference on Wireless and Mobile Communications (ICWMC 2010), pp. 331-336, September 20-25, Valencia, 2010.

[2] C. Perkins, E. Belding-Royer, and S. Das, Internet Engineering Task Force (IETF): Ad hoc On-Demand Distance Vector (AODV) Routing, Request for Comments (RFC) 3561, July 2003.

[3] http://www.nsnam.org/

[4] D. Johnson and G. Hancke, Comparison of two routing metrics in OLSR on a grid based mesh network, Ad hoc Networks, Volume 7 (2), pp. 374-387, 2009.

[5] D. Passos, D. V. Teixeira, D. C. Muchaluat-Saade, L. C. S. Magalhães, and C. V. N. Albuquerque, Mesh Network Performance Measurements, In 5th International Information and Telecommunication Technologies Symposium, 2006.

[6] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, Architecture and Evaluation of an Unplanned 802.11b Mesh Network, ACM Mobicom Conference, pp. 31-42, September 2005.

[7] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, MACAW: A Media Access Protocol for Wireless LAN's, Proceedings of the Conference on Communications Architectures, Protocols and Applications, pp. 212-225, London, United Kingdom, 1994.

[8] S. Yin, Y. Xiong, Q. Zhang, and X. Lin, Traffic-aware Routing for Real Time Communications in Wireless Multi-hop Networks, Wireless Communication and Mobile Computing, Volume 6 Issue 6, pp. 825-843, September 2006.

[9] V. Ramasubramanian, Z. J. Haas, and E. G. Sirer, SHARP: A Hybrid Adaptive Routing Protocol for Mobile Ad Hoc Networks, In Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing, pp. 303-314, Annapolis, Maryland, June 2003.

[10] X. Ni, K. Lan, and R. Malaney, On the performance of expected transmission count (ETX) for wireless mesh networks, Proceedings of the 3rd International Conference on Performance Evaluation Methodologies and Tools, Athens, Greece, 2008.

[11] R. Baumann, S. Heimlicher, M. Strasser, and A. Weibel, A Survey on Routing Metrics, TIK Report 262, ETH Zürich, February 2006.

[12] M. E. M. Campista at al., Routing Metrics and Protocols for Wireless Mesh Networks, IEEE In Network, Volume 22 Issue 1, pp. 6-12, 2008.

[13] S. J. Lee and M. Gerla, AODV-BR: Backup routing in Ad Hoc networks, Proceedings of IEEE WCNC 2000, Volume 3, pp. 1311-1316, Chicago, September 2000.

[14] M. K. Marina and S. R. Das, Ad hoc on-demand multipath distance vector routing, Wireless Communications & Mobile Computing, Volume 6 Issue 7, pp. 969-988, November 2006.

[15] Z. Ye, S. V. Krishnamurthy, and S. K. Tripathi, A framework for reliable routing in mobile ad hoc networks, IEEE INFOCOM, Volume 1, pp. 270-280, Sanfrancisco, March 2003.

[16] S. J. Lee and M. Gerla, Split Multipath Routing with Maximally Disjoint Paths in Ad hoc Networks, Proceedings of IEEE ICC 2001, pp. 3201-3205, Helsinki, Finland, June 2001.

[17] G. Almes, S. Kalidindi, and M. Zekauskas, A One-way Delay Metric for IPPM, IETF RFC 2679, September 1999.

[18] C. Demichelis and P. Chimento, IP Packet Delay Variation Metric for IP Performance Metrics (IPPM), IETF RFC 3393, November 2002.

[19] D. Aguayo, J. Bicket, and R. Morris, SrcRR: A High Throughput Routing Protocol for 802.11 Mesh Networks, http://pdos.csail.mit.edu/~rtm/srcrr-draft.pdf

[20] B. S. Manoj and R. R. Rao, Wireless Mesh Networks: Issues and Solutions, In: Y. Zhang, J. Luo, and H. Hu, Wireless Mesh Networking: Architectures, Protocols and Standards, Auerbach Publications,Ch. 1, pp. 3-48, New York, USA, 2007.

[21] http://www.nsnam.org/wiki/index.php/PyViz