Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2007, Article ID 19375, 14 pages
doi:10.1155/2007/19375

Research Article

Effects of Unstable Links on AODV Performance in Real Testbeds

Eleonora Borgia and Franca Delmastro

Pervasive Computing and Networking Laboratory (PerLab), Inistitute for Informatics and Telematics (IIT), National Research Council (CNR), Via G. Moruzzi, 56124 Pisa, Italy

Received 14 July 2006; Revised 24 October 2006; Accepted 30 January 2007

Recommended by Marco Conti

A link between a pair of nodes is defined *unstable* if it is characterized by a packet loss which is not negligible in one or both directions. The presence of unstable links in multihop ad hoc networks is very likely and it depends on several factors (e.g., different transmission capabilities of the devices, interferences caused by additional wireless devices). Their management by the routing protocols is of paramount importance since they negatively affect applications performance. In our previous experimental studies, we found that AODV is characterized by very low performance in some specific situations and, in this work, we demonstrate that it mainly depends on the wrong management of unstable links as valid routes. We present some policies that have been proposed in literature to avoid this problem, and we validate two of them through experimental results, exploiting also a direct comparison with the proactive routing protocol OLSR. Our results show that AODV is not able to avoid the use of unstable links, even when an alternative stable route exists. In the same conditions, OLSR outperforms AODV by correctly managing unstable links. In fact, it is able to guarantee a higher packet delivery ratio to the application by using the most stable path to reach the destination.

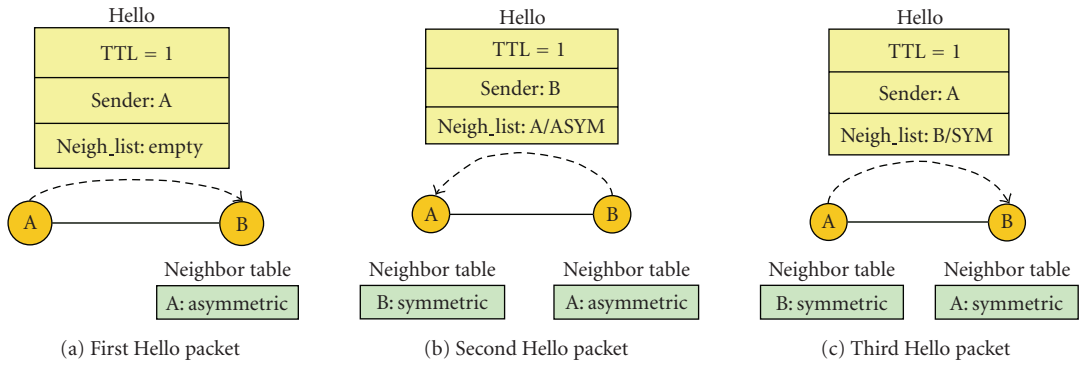
1. INTRODUCTION

Routing protocols represent the main concern of multihop ad hoc networks and for this reason they represent one of the most active research areas within the MANET domain. Specifically, the development and validation of optimized routing protocols, able to support reliable and efficient nodes communications, are of paramount importance to achieve efficient services and high applications performance.

Proactive and *reactive* protocols are the main categories of MANET routing protocols. Proactive protocols seek to maintain a constantly updated view of the network topology relying on periodic exchange of routing information between nodes. On the opposite, reactive protocols discover a route to a specific destination only when it is requested from the upper-layer applications, that is, *on demand*. These protocols maintain only routes involved in active communications until the destination becomes unreachable or it is not used for a specific amount of time.

Currently the research community mainly focuses its studies on two specific protocols: AODV (reactive) and OLSR (proactive). These protocols are the most mature from the implementation standpoint, and highlight advantages and drawbacks of the two solutions. In previous work [1–4], we selected two specific implementations of these protocols and we extensively evaluated them in small- and medium-scale testbeds. In this paper, we summarize the main results obtained by our experiments highlighting the advantages of using OLSR in terms of network topology management, applications performance, and reliability of nodes communications. In addition, we found that low performance of AODV mainly depends on the use of unstable links as part of possible valid routes.

Unstable links are generally defined as links affected by a not negligible packet loss in one or both directions. Generally, the link status varies over time depending on several factors, for example, nodes mobility, physical distances, interferences produced by additional devices in the environment. Thus, a stable link can become unstable due to the change of some conditions in a specific period of time. In the experiments presented in this paper, we define a link as unstable if it measures a valuable packet loss for the entire duration



(a) First Hello packet (b) Second Hello packet (c) Third Hello packet

Figure 1: Example of Hello messages exchange to identify bidirectional links.

of the experiment. Thus, in order to maintain the same link conditions for several experiments, we decided to use only static topologies.

Actually, unstable links can be divided into two categories: *unidirectional* and *asymmetric* links. A link between a pair of nodes is defined as unidirectional when only one of the two nodes can directly communicate with the other one. This phenomenon is generally caused by different transmission capabilities of the devices that also cause different transmission ranges. Instead, an asymmetric link is caused by the difference in interference conditions at the ends of the link that produces different link qualities in the two directions.

The main distinction between these two categories is the capability of nodes to receive data. In fact, let us consider a generic pair of nodes A, B. If the link A → B is unidirectional, B is able to receive data from A, but A cannot receive any data from B. Instead, in case of asymmetric link, the interference causes a high packet loss in one direction, that limits the reception of data packets, but it does not necessarily eliminate it at all (i.e., A can receive some packets from B).

To validate the assumption that unstable links are the main cause of AODV low performance, in this work we evaluate its performance in several scenarios affected by either unidirectional or asymmetric links, comparing AODV results with those obtained by running OLSR. From the performance evaluation study, we conclude that routing protocols need a policy to control the use of unstable links to guarantee reliable communications to upper-layer services. Proactive protocols originally provide a policy to maintain only bidirectional links as valid routes, and hence their use in multihop ad hoc networks generally improves the system performance. The same policy could also be adopted in reactive protocols even though it increases the traffic load. Thus, further techniques have been proposed to solve this problem as we explain in this paper. First of all, we give an overview of OLSR and AODV (see Sections 2 and 3, resp.) to better support the explanation of experimental results presented in Section 4. Then, in order to verify AODV low performance, we explain how the original protocol definition manages these situations, detailing then the additional policies proposed in literature to solve this problem (see Section 5). Finally, we analyze AODV behavior in presence of unidirectional and asymmetric links through real experiments, comparing its performance with that obtained by OLSR (Section 6). A final discussion is thus presented in Section 7.

2. OPTIMIZED LINK STATE ROUTING PROTOCOL (OLSR)

OLSR [5] derives from the family of link state routing protocols. It inherits from this family the proactive flooding of topology information, but it highly reduces the overall traffic load achieving a trade-off between resource constraints of wireless networks and the maintenance of a complete and updated network topology. First of all, it implements a 1-hop neighbors discovery procedure based on the exchange of Hello messages. Each node periodically broadcasts a Hello message containing the list of its 1-hop neighbors and the related link status. OLSR defines a link as symmetric if it has been verified to be bidirectional, that is, it is possible to exchange packets in both directions. Otherwise, the link is defined asymmetric. Figure 1 shows an example of the 1-hop neighbors discovery. Let us consider the pair of nodes A, B. Assuming that node A is the first one to send a Hello packet and it does not know any neighbor, it inserts an empty neighbors' list in the packet (see Figure 1(a)). Thus, when B receives the Hello packet, it checks whether it has been already recognized by A as a neighbor but, since the list is empty, it stores in its neighbor's table the link to node A as asymmetric. Then, when B sends its Hello packet (see Figure 1(b)) it inserts node A and the related link status in its neighbors' list. Thus, when node A receives the packet, it stores the link to node B as symmetric since it recognizes itself as a B's neighbor. At this point, when A sends the subsequent Hello packet (see Figure 1(c)), it adds node B to its neighbors' list, and eventually B stores the link to A as symmetric. Only when a link is recognized to be symmetric is considered as a valid route and consequently added to the routing table.

Through the exchange of Hello messages, every node directly knows its 2-hop neighbors since every node announces itself and the list of its 1-hop links. Then, exchanging information about the 2-hop knowledge of the network, nodes are able to recover the entire network topology. To minimize the

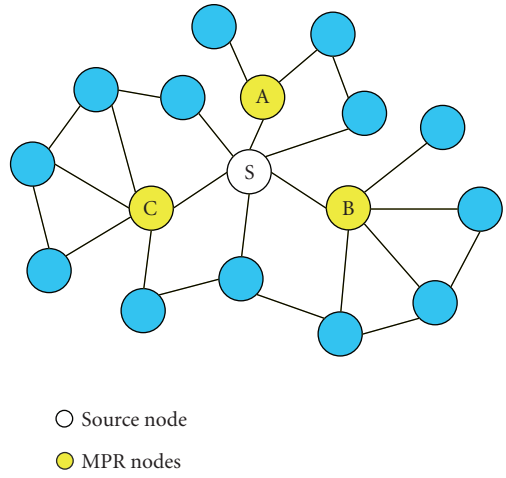


Figure 2: OLSR Multipoint Relays selection.

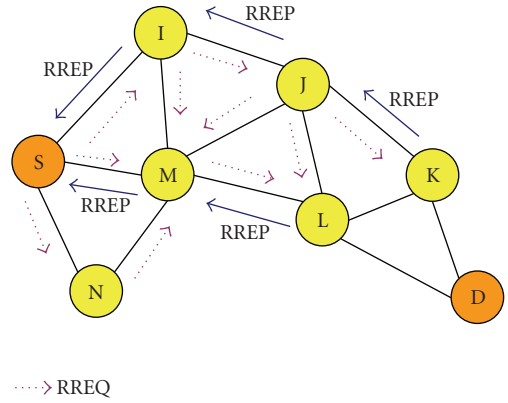


Figure 3: AODV Route Discovery procedure.

network overhead, this information is broadcasted by a selected set of 1-hop neighbors of each node through topology-control packets. These special nodes are called *multipoint relays* (MPRs). Every node identifies the MPRs among its symmetric neighbors so that it can reach all its 2-hop neighbors through them. In Figure 2, node S elects its MPR set, that is, nodes A, B, and C. Only these nodes forward routing packets received by node S, while all the other nodes, not in the MPR set of S, receive and process those packets without retransmitting them. Each node maintains also information about which nodes have elected itself as MPR, collecting their addresses in the *MPR selector set*. As a consequence, each node must retransmit only packets coming from nodes stored in its MPR selector set. This strategy limits the number of retransmissions in the network, and it is further optimized reducing the amount of information travelling in the network. In fact, instead of declaring the complete list of neighbors in the topology control packets, each node announces only a subset of them and, more precisely, the MPR selector set, that is enough to build and manage the routing tables. Thus, OLSR not only gives a complete knowledge of the network topology to every node, but also guarantees the establishment of routes that exploit only bidirectional links.

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

Reactive routing protocols discover a route only when it is required. Specifically, AODV minimizes the number of broadcast messages by creating routes on-demand via a route discovery procedure that works as follows. Whenever a traffic source S needs a route to a destination D (see, e.g., Figure 3), it initiates a route discovery by flooding a route request packet (RREQ) for the selected destination in the network, and then it waits for a route reply packet (RREP). When an intermediate node receives the first copy of a RREQ packet, if it directly knows the destination (e.g., nodes L and K in Figure 3), it sets up a reverse path to the source using the previous hop of the RREQ as the next hop of the reverse path, and it unicasts a RREP back to the source through the same path. Otherwise, it rebroadcasts the RREQ packet. Duplicate copies of the RREQ are immediately discarded upon reception at every node. If the RREQ reaches the destination (i.e., no intermediate node directly knows it), it unicasts the RREP back to the source along the reverse path. In addition, while the RREP is moving towards the source crossing the intermediate nodes, a forward path to the destination is established at each hop.

Furthermore, to have at least a partial view of the network topology even in absence of application traffic, AODV allows nodes to learn about their 1-hop neighbors by exchanging Hello-like RREP messages. AODV uses Hello-like RREP as beacons, just to announce the presence of the local node in the network. It does not define a specific packet for this message, but it directly exploits a RREP packet with TTL equal to 1. Every node periodically broadcasts these messages on the network unless it has already sent a RREQ in the last period. Thus, even though there is no request to establish a specific route, nodes are aware of their 1-hop neighbors. However, no check on the link status between pairs of nodes is implemented by the protocol, and this may cause the use of unidirectional links as valid routes.

Considering the example shown in Figure 3, due to the exchange of Hello-like RREP messages, every node knows its 1-hop neighbors. Thus, when the source node S generates its RREQ to reach the destination D, it is broadcasted on the network (dashed arrows), and the nodes that know D as 1-hop neighbor send to S the RREP on the related reverse path. Each intermediate node forwards only the first copy of every RREQ and, when it receives the related RREP, it stores the forward path to D before retransmitting the packet. In this example, nodes L and K are the 1-hop neighbors of D, and when they receive the RREQ they directly send the RREP to S. At this point, S stores in its routing table the first available path obtained by the first received RREP and, if necessary, it subsequently updates it with the shortest one (i.e., S-M-L-D).

In addition, to guarantee the validity of each discovered path, AODV defines a *route maintainance* procedure. Each node maintains a *predecessors' list* for each RREQ received.

The list contains the set of nodes from which the local node has received a copy of the RREQ. Thus, once a node observes that an active link towards a node is lost, it sends a Route Error (RERR) message to all its neighbors specified in the predecessors' list used to reach that specific destination, and it invalidates all the active routes that use the broken link. Then, every node receiving the RERR updates its data structures and forwards the message to its "predecessors" nodes, so that all the active sources become aware of the broken link. After receiving the RERR, the source node removes the route that uses the unavailable link, and starts a new route discovery to the same destination.

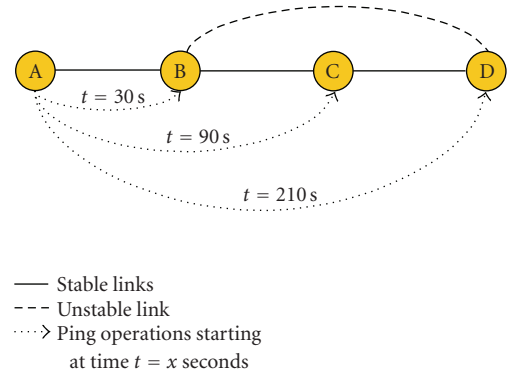


Figure 4: Ping operations on a string topology.

4. ROUTING PERFORMANCE IN SMALL- AND MEDIUM-SCALE AD HOC NETWORKS

In literature, there are many studies on routing protocols performance. Most of them are based on simulative results [6, 7], but experimental evaluations are currently increasing [8, 9]. The research community has realized that even though simulators allow the performance evaluation of protocols in different scenarios varying several parameters, they introduce symplifying assumptions that may mask real characteristics of the network [10, 11]. Thus, to obtain more realistic results, it is necessary to complement simulations studies with real experiments.

In this section, we present a summary of experimental results that we obtained by investigating small- and medium-scale multihop ad hoc networks (see [1–4] for details). These results highlight that generally OLSR outperforms AODV in terms of delays, packet loss, and scalability with the network size, introducing only a slight increase in the traffic load. In addition, in several cases, AODV becomes almost unusable. We found that AODV management of unstable links is the main reason of its low performance, as we deeply explain in Section 5.

All our testbeds were built using IBM ThinkPad R40/R50 laptops running Linux OS and equipped with IEEE 802.11-integrated wireless cards. We set the driver of the wireless cards to work in ad hoc mode using the 802.11b standard at 11 Mbps data rate. We selected two available implementations of the routing protocols: Unik-OLSR v.0.4.8 [12] developed by the University of Oslo, and AODV-UU (versions 0.8.1 and 0.9.1) [13] developed by Uppsala University. We mainly focused on static topologies from 4 up to 23 nodes. All the experiments were conducted in the CNR campus in Pisa exploiting both indoor and outdoor spaces.

In most of the experiments, we compared AODV and OLSR performance using the ping utility as application traffic generator. During each ping operation among pairs of nodes, we mainly analyzed the *packet loss* measured at the application level and the *delays* introduced during data transfer. Referring to delays, we analyzed the latency required to complete an *ICMP handshake* between a couple of nodes, that is, the time interval needed by the sender to receive the ICMP reply related to its ICMP request, namely, Round Trip Time (RTT). To highlight the influence of route discovery procedures on application delays, we distinguished between (i) the delay to complete the first successful ICMP handshake between a selected pair of nodes, including also all the lost ICMP packets until the first successfull handshake, (ii) the average RTT measured on the entire ping operation. In the reactive protocol the delay at point (i) includes the time needed for the route discovery procedure, while in both protocols the delay at point (ii) includes also network reconfigurations, if any. All the experiments have been repeated several times and we present average values of the performance indices. In the following subsections, we present performance results of both routing protocols in small- and medium-scale testbeds.

4.1. Small-scale testbed: string topology in indoor and outdoor environments

In these experiments, we set up a string topology of 4 nodes (see Figure 4). The main purpose was to locate nodes such that only the adjacent ones can directly communicate. Initially, we did not realize that there was an unstable link between nodes B and D, but it has been verified during the analysis of the experiments. We analyzed routing performance by executing ping operations from node A to every other destination (i.e., B, C, and D). The duration of each ping operation depends on the distance between the sender and the destination.

In all the experiments, all the nodes start running the routing protocol for 30 seconds to fill up their routing table with 1-hop neighbors in case of AODV, and with all the available routes in case of OLSR. Then, node A pings node B (i.e., the node at 1-hop distance) for 1 minute (from $t = 30$ seconds to $t = 90$ seconds). Subsequently, it pings node C for 2 minutes, and finally node D for 3 minutes (see Figure 4).

From these results, we noticed that OLSR is able to deliver almost all the generated packets towards all the nodes in the network. On the contrary, AODV works properly with nodes at most 2-hop away, while only 50% of the generated packets are successfully delivered to node D, due to frequent route reconfigurations involving the unstable link B-D. These values point out a first effect of the presence of an unstable link in the network topology. The analysis of the delays experienced

Table 1: Indoor string topology: experimental results.

	Performance indices	Ping operations		
		A → B	A → C	A → D
AODV	Packet loss	14%	9%	50%
	1st ICMP handshake delay (ms)	17.85	85.65	2132
	Average RTT (ms)	4.45	27.367	79.09
OLSR	Packet loss	0.1%	0.2%	0.1%
	1st ICMP handshake delay (ms)	15.15	55.35	50.5
	Average RTT (ms)	3.7	25.5	54.148

by the routing protocols in the same set of ping operations further highlight this issue. A summary of the results is presented in Table 1.

Considering the time required to successfully complete the first ICMP handshake, running AODV we measured an average delay of 17.85 milliseconds towards node B, 85.65 milliseconds towards node C, and 2.132 seconds to node D. These values include the time needed by the reactive protocol to discover the route to the designated destination. This procedure usually requires two or three attemps before establishing the valid route. Note that in the last ping operation (i.e., from node A to node D) several attempts are needed due to the presence of the unstable link, that influences the entire operation with a high number of route changes and the consequently increase of the packet loss.

Instead, in case of OLSR, the unstable link B-D is never considered as a valid route, and the protocol introduces a delay of about 50 milliseconds to complete the first ICMP handshake at 2 and 3 hops distance.

Thus, in this set of experiments, OLSR outperforms AODV both in terms of packet loss and delays avoiding the use of the unstable link in the valid routes.

We repeated the same experiments in outdoor environment and we noticed that performance worsens both in terms of packet loss and delays. This can be due to the fact that, in outdoor environments, where adjacent nodes were physically distant about 70 meters, the carrier sensing range did not include all the nodes of the string, in contrast with the previous indoor experiments [14]. In this case, the probability of having hidden terminal problems, causing a higher number of MAC collisions, contributes to reduce the performance of both protocols. In addition, by increasing the physical distance between pairs of nodes, the probability of packet loss increases too, as well as the wireless links instability. AODV suffers more than OLSR even in this case. In fact, it introduces up to 100% packet loss on two or three hops connections, while OLSR experiences at most a 50% packet loss at 3-hop distance.

4.2. Medium-scale testbed

To analyze the influence of the network size on routing performance, we set up a medium-scale testbed with 23 nodes mixing indoor and outdoor connections. To obtain a redundant topology with realistic wireless links in a small geographic area, physical characteristics of the buildings and heterogeneous wireless cards were exploited. Figure 5 shows the network topology graph in which there are also two unstable links at the center of the network. Our analysis pointed out that these unstable links caused several network reconfigurations during the experiments decreasing the overall system performance.

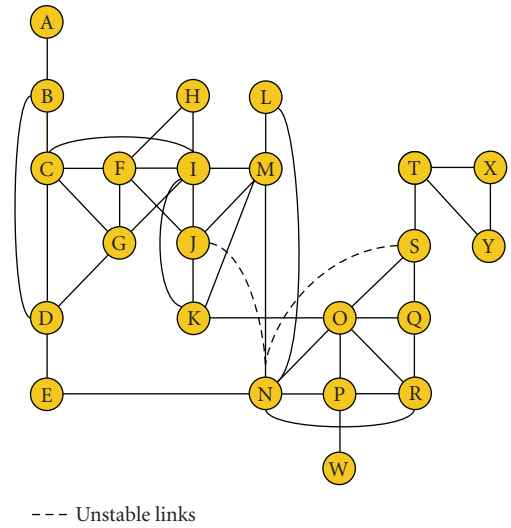


Figure 5: Medium-scale topology graph.

To evaluate routing protocols behavior in this medium-scale network, we ran concurrent ping operations between every pair of nodes and we analyzed the packet loss depending on the hop-distance between the end nodes (see Table 2). As expected, the performance of both protocols worsens by increasing the hop-distance. In fact, in case of OLSR, the packet loss increases from 15% to 45% by increasing the distance from 2 to 5 hops, and it becomes higher than 50% with connections of 6 and 7 hops. AODV performance worsens even more rapidly than OLSR since it introduces a 50% packet loss at a distance of 2 or 3 hops, and it drastically increases beyond 5 hops reaching a maximum value of 89% at 7 hops. In addition, analyzing the delay to successfully complete the first ICMP handshake, OLSR always experiences delays of about 5 seconds for nodes [4, 6] hops away, and up to 10 seconds for 7-hop connections. Instead, running AODV the delay is about 10 seconds by increasing the distance from 2 to 5 hops, and more than 15 seconds

Table 2: Medium-scale testbed: average packet loss for different numbers of hops.

	Number of hops						
	1	2	3	4	5	6	7
AODV	20%	51%	51%	61%	67%	86%	89%
OLSR	5%	15%	28%	35%	45%	52%	67%

beyond 6-hop distances. Referring to the average RTT measured on every ping operation, we measured delays lower than 200 milliseconds in case of OLSR even for 7-hop connections, while in case of AODV we experienced 700 msec delays for nodes 6-hop away, and 1 second for 7-hop connections.

In conclusion, both routing protocols performance worsen in the medium-scale testbed where the network topology is more unstable, and several network reconfigurations are necessary. These characteristics consequently affect the application performance.

5. INFLUENCE OF UNIDIRECTIONAL AND ASYMMETRIC LINKS IN AODV

Experimental results presented in the previous section showed that AODV performance is highly variable due to some unstable links that are exploited by the protocol as valid routes to forward packets. To better understand the protocol behavior in these cases, in this section we focus on the influence of unstable links on routing and applications performance, analyzing possible mechanisms to control their use. As previously said, unstable links are mainly divided into two categories: unidirectional and asymmetric links. In real experiments, it is much more likely to find asymmetric links than unidirectional links, especially in indoor environments where the structural characteristics of the buildings and the presence of additional devices, like access points, can introduce interference on the wireless channel. However, as a first step to analyze AODV's behavior in presence of unstable links, it is important to explain how the protocol addresses this issue. Note that the protocol specification refers only to unidirectional links since they represent the extreme condition of unstable links. Thus, in this section we evaluate AODV management of unidirectional links and the possible policies to avoid their use. Then, in Section 6 we analyze AODV performance in real experiments characterized by either unidirectional or asymmetric links.

Note that in literature [15, 16] it was originally claimed that using unidirectional links in addition to using only bidirectional links had two specific advantages: to improve network connectivity, and to provide shorter paths. However, to be effective for routing, unidirectional links should exist long enough to allow the routing protocol to compute routes through them and to use such routes to forward data packets. Actually, unidirectional links caused by variation in interference levels may have not a very long life. In [17], it is demonstrated with simulations results that unidirectional links do not significantly improve network connectivity, and ignoring them only marginally increases the shortest path length.

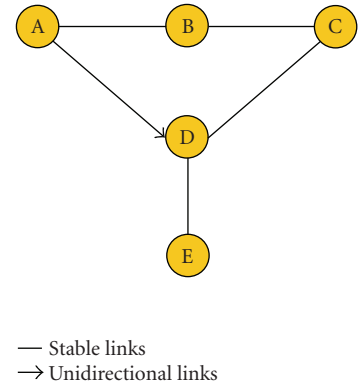


Figure 6: Network topology with unidirectional links (scenario 1).

Starting from these considerations, some techniques to avoid the use of unidirectional links in AODV have been proposed in [17]. However, before explaining these techniques, it is worth pointing out how AODV originally establishes valid routes exploiting unidirectional links.

Let us consider the network topology shown in Figure 6. Suppose that node A sends a RREQ for node E and node D receives the RREQ through the unidirectional link A → D before those carried from the alternative path A-B-C-D. In this case, not only node D discards all the subsequent RREQs received by node C, but it also tries to send RREP messages to A through the unidirectional link. Thus, all the RREP transmissions fail, and node A experiences repeated route discovery failures.

This is a direct consequence of the 1-hop neighbors discovery procedure implemented by AODV. In fact, as previously explained in Section 3, every node periodically sends a Hello-like RREP message (TTL equal to 1) to announce its presence in the network. This message does not contain information about the neighbors known by the sender. Hence, the receiving node is not able to check whether the link is bidirectional, and it adds the source node to its routing table as a valid 1-hop route. In the previous example, node D receives Hello-like RREPs from A through the unidirectional link, and it considers A as its 1-hop neighbor, but node A cannot receive any message from D. This general case is the basis of the generation of unidirectional routes in AODV.

To better understand how it is possible to handle this phenomenon and correctly analyze AODV performance in our real testbeds, in the following subsections we briefly explain the most important techniques proposed in [17] (Blacklisting, Hello packets, and ReversePathSearch). Note that Blacklisting technique is the only one included in the latest AODV specification [18], while the others are not currently implemented, thus they cannot be experimentally evaluated in this work.

5.1. Blacklisting

This technique reactively eliminates unidirectional links detecting RREP transmission failures during route discovery procedures. To this end, each node sending a RREP packet (except those used as Hello-like messages) waits for an explicit acknowledgment (RREP-ACK) from the related destination. Thus, if the node does not receive the RREP-ACK before the related timeout expiration, it stores the destination of the RREP in a "blacklist" set. Since the blacklisted nodes are identified as sources of unidirectional links, the local node discards all the subsequent RREQs received from them to avoid the creation of unidirectional routes. Nodes are removed from the blacklist set after a timeout.[1]

Actually, this solution is not completely effective. In fact, it considers only the originators of RREQ packets as possible sources of unidirectional links, but the same nodes periodically broadcast Hello-like RREP messages. Thus, even though a node is blacklisted, its Hellos-like are not discarded at the receivers, that continue to consider it as a valid 1-hop neighbor. Referring to the previous example shown in Figure 6, when node D receives a RREQ from A to discover node E, it sends a RREP to A and it is able to detect the unidirectional link since it cannot receive the RREP-ACK from node A. In this case, node D inserts node A in the blacklist, and discards the following RREQs received from it. However, D continues to receive Hello-like RREPs from A and considers it as 1-hop neighbor in the routing table. Hence, in case node E sends a RREQ to D to discover A, D replies with the available link A-D not realizing that it is unidirectional. Thus, this technique cannot be considered completely effective in case of unidirectional links. To confirm this assertion we give an exhaustive explanation in Section 6 analyzing several scenarios.

5.2. Hello packets

Local Hello messages can be used not only to announce the presence of each node in the network, but also to broadcast their local connectivity. A node can determine its 1-hop neighbors listening for their Hello packets, and it can forward this information including the list of neighbors in its Hello packets, as implemented by OLSR and other proactive protocols. If a node does not find itself in the Hello packet of its neighbor, it marks that link as unidirectional. Thus, everytime a node receives a RREQ packet, first of all it must check whether the originator node is marked as a source of unidirectional link, and in this case it discards the packet. Otherwise it correctly manages the RREQ. In respect of the Blacklisting technique, this policy proactively eliminates unidirectional links, checking the bidirectional knowledge of the 1-hop neighborhood.

[1] In the protocol specification this timeout is generally set to the maximum time required by the node to perform the allowed number of RREQ retries.

5.3. ReversePathSearch

The authors of [17] propose also an alternative policy to the previous ones. This technique does not explicitly remove unidirectional links, but since those links are considered as faults in the network connectivity, multiple paths between pair of nodes are discovered to implement a fault-tolerant routing protocol. The ReversePathSearch technique exploits the RREQ flooding to discover multiple reverse paths to the source. For this reason, all RREQ copies are examined at intermediate nodes and at the final destination. For each received RREQ a node stores in its routing table the next hop to be used for the related RREP and the hopcount (to avoid possible routing loops[2]). Thus, when an intermediate node receives a RREQ and it has a valid path to the destination, first of all it checks whether a RREP has been already sent back for the same route discovery. If not, it sends back a RREP along the reverse path, storing the next hop used for the RREP. Otherwise it stores the possible reverse next hop and discards the packet. In case it has no valid path to the destination, it rebroadcasts the RREQ. Then, if the final destination receives one or more RREQs, it sends back a RREP for each reverse path allowing the exploration of multiple paths concurrently.

In addition, every intermediate node executes the same check also before forwarding RREP packets. Specifically, when an intermediate node receives a RREP, if it has one or more valid paths to the source, it checks whether it has already sent back a RREP on one of these paths. If not, it chooses one of the available paths and forwards the RREP storing the used next hop, otherwise the RREP is discarded. In this way, a single path between source and destination is established, even though every node maintains possible reverse paths in its routing tables. Actually, the possibility to explore alternative reverse paths is exploited in case of RREP failures (generally due to the presence of unidirectional or asymmetric links). In this case, when a RREP fails at a node, the corresponding reverse path is erased and the node tries another alternative reverse path. If no alternative path is available, the node sends a Backtrack Route REPly (BRREP) to inform its neighbors (in the direction of the source of the RREP) to try other reverse paths. Considering the example shown in Figure 7, there are two alternative routes between nodes S and D, one of them characterized by a unidirectional link (A → E). As a first step, S begins a route discovery procedure broadcasting RREQ messages. Suppose that node E receives firstly a RREQ from A. Since E directly knows D, it sends a RREP to A (step 2) and stores the reverse path in its routing table. Then, when E receives the RREQ from C, it stores C as next hop for an alternative reverse path, and discards the packet. However, the RREP forwarded by A to S fails due to the undirectional link, and A sends back a BRREP to E to notify the failure (step 3). At this point, E erases the reverse path through A from its routing table, and it forwards the RREP to C, that eventually reaches

[2] Rules to establish and maintain loop-free routes are explained in [19].

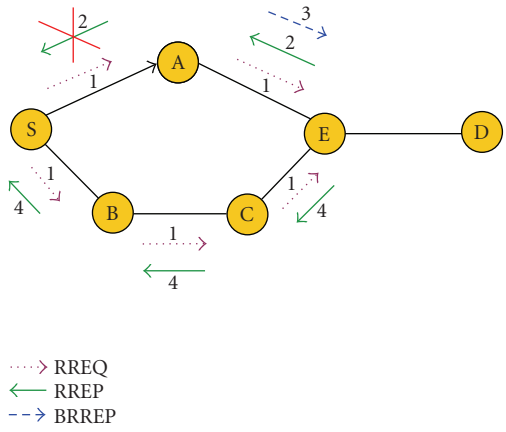


Figure 7: ReversePathSearch example.

node S. Technical details of the algorithm can be found in [17, 19].

The authors of this last solution extensively evaluated all the three techniques through simulative studies, and showed that the ReversePathSearch performs better than the others because of its ability to explore multiple paths. However, even in this case all the RREQs coming from a unidirectional link are discarded, but there is no action on Hello-like RREP messages. Thus, even this technique cannot avoid the use of unidirectional links when the node that receives a RREQ directly knows the destination through its Hello-like RREPs. In this case there is no RREP failure, but repeated route discovery failures.

Hence, from the analysis of these techniques it is clear that the most effective technique is that based on Hello packets that guarantees the use of only bidirectional links. To support this assumption in the next section, we report experimental results obtained by running the routing protocol on different small-scale network topology characterized by unidirectional and asymmetric links.

6. EXPERIMENTAL RESULTS

In this section, we report experimental results of AODV performance in presence of unidirectional and asymmetric links, with the additional support of the Blacklisting technique. Then, in order to highlight application performance improvements when using only bidirectional links, we compare AODV with OLSR, which implements the Hello packets technique.[3]

Since it is very difficult to establish perfect unidirectional links in real experiments, we divide the experimental analysis into two parts. Firstly, we conducted several experiments with different network topologies and application scenarios using iptables firewall to emulate multihop connections and to force the establishment of unidirectional links. Then, we repeated some experiments in real multihop configurations, replacing iptables unidirectional links with asymmetric links. These asymmetric links have been established by varying the transmission power of the wireless cards and exploiting the structural characteristics of the buildings.

Note that, in case no application traffic is generated on top of the routing protocol, AODV only generates periodic Hello-like RREP messages to announce the presence of the local node to its 1-hop neighbors. Thus, to execute route discovery procedures and analyze AODV performance in presence of unidirectional and asymmetric links, we used the ping utility.

6.1. *Unidirectional links experiments (iptables configurations)*

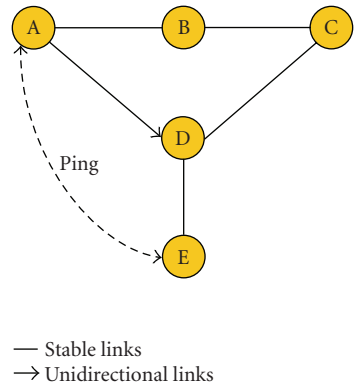
We consider three experimental scenarios characterized by a small ad hoc network of five nodes, see Figures 8, 9, and 10. In every scenario, we defined two different sets of experiments swapping the end nodes of the ping operation to highlight different failures of the route discovery procedure in the same scenario. The experiments have been conducted running both AODV and OLSR, and the performance evaluation mainly focuses on the same indices used in the previous experiments: packet loss, end-to-end delay to successfully complete the first ICMP handshake (including also all the lost ICMP packets until the first successfull handshake), and the average RTT measured on the entire ping operation. The distinction between the last two indices highlights the cost of the reactive protocol every time a new route has to be discovered. All the results are averaged over three consecutive trials.

6.1.1. *Scenario 1*

In the scenario shown in Figure 8, we used nodes A and E as end nodes for the ping operation. As discussed in Section 5, this scenario represents the worst case for AODV management of unidirectional links. In fact, executing the ping operation in both directions, AODV route discovery completely fails. We ran two sets of experiments. In both cases we used the first 30 seconds to stabilize the network topology, running only the routing protocol. In case of AODV, only Hello-like RREP messages are exchanged in this period, discovering 1-hop neighbors. Instead, in case of OLSR, routing tables are filled up with all the network nodes. In the first set of experiments node A pings node E for 120 seconds, while in the second set of experiments, node E pings node A for the same amount of time. Packet loss results are summarized in Figure 8.

In the first case, when node A pings node E, AODV experiences a 100% packet loss. In fact, for each route discovery procedure generated by node A, node D receives the RREQ directly from A through the unidirectional link A → D, and it discards all the subsequent RREQs forwarded by B and C, thus losing the possibility to discover the alternative path to

[3] The hardware and software components are the same used in the previous testbeds presented in Section 4.



Scenario 1	AODV packet loss	OLSR packet loss
(a) Ping: A → E	100%	0%
(b) Ping: E → A	98.3%	0.33%

Figure 8: Unidirectional links: scenario 1 (network topology and experimental results).

E.[4] In addition, all the RREP messages sent by node D to A are lost, and node A continues to send RREQs to discover the route. Once the maximum number of RREQ retransmissions is reached, the ping application reports a "Destination Host Unreachable" message at the source node since no ICMP handshake has been completed. In this case, the Blacklisting technique has no effect on the management of the unidirectional link A → D. In fact, node D requires a RREP-ACK for each RREP message sent to A, but it never receives them. Thus, it adds node A to its blacklist and it has to discard all the subsequent RREQs generated. Since node A is the originator of all the requests, node D discards also the RREQs forwarded by nodes B and C, losing the possibility to discover the alternative route.

On the contrary, OLSR experiences no packet loss in this set of experiments since it considers only the bidirectional path A-B-C-D-E as a valid route to the destination. In this case, we measured an average end-to-end delay for the first ICMP handshake equal to 16.9 milliseconds, while the average RTT on the ping operation is 7.72 milliseconds.

In the second set of experiments, reversing the ping operation from node E to A, AODV experiences an average packet loss of 98.3%, maintaining the route E-D-C-B-A only for few packets. In fact, when node E begins its route discovery procedure, node D receives the RREQ, and since it directly knows node A as its 1-hop neighbor, it replies to node E allowing the creation of the route E-D-A. At this point node E starts sending ICMP requests to D that forwards them to A, which cannot receive them due to the unidirectional link. In this set of experiments, the measured packet loss is not 100% because, due to possible MAC collisions, node D loses some Hello-like RREPs from A. In those cases, D generates a RERR to E to announce the unreachable destination A, and E has to repeat the route discovery procedure. Thus, when D receives the RREQ from E, it has no route to the destination A and it has to forward the RREQ, and it obtains the alternative route by node C. However, when D receives the subsequent Hello-like RREP from A, it updates its routing table with the original route, causing the failure of the next ICMP packets. Also in this case, the Blacklisting technique has no effect on the route discovery procedure. In fact, node D requires the RREP-ACK to E and it always receives them, but the same procedure is not applied to Hello-like RREP messages received by A. As a consequence, node D never learns about the presence of the unidirectional link A → D.

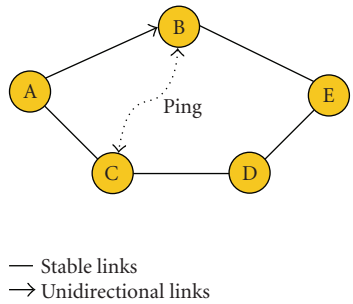
In the same experiment OLSR measures an average packet loss of 0.33% exploiting the route E-D-C-B-A. The average delay to complete the first successful ICMP handshake is equal to 14.36 milliseconds, and the average RTT on the entire ping operation is 7.65 milliseconds. Thus, we can claim that using Hello packets containing 1-hop neighborhood information completely avoid the use of unidirectional links, while the Blacklisting technique completely fails in this scenario since it is not able to detect the unidirectional link during the route discovery procedure.

Note that AODV and OLSR adopt different policies to generate Hello messages. In fact, the frequency with which these messages are sent on the network and their validity time in the protocols data structures are different. Specifically, AODV Hello-like RREP messages are sent every 1 second and they are considered valid for only 2 seconds. Instead in OLSR, Hellos are broadcasted with a period of 2 seconds and their validity time is set to 6 seconds. This means that in AODV it is sufficient to lose 2 Hello messages from a neighbor to invalidate a 1-hop route, while OLSR needs 3 Hello failures to discard the route. Thus, AODV suffers the loss of Hello messages more than OLSR, increasing the probability of route changes.

6.1.2. Scenario 2

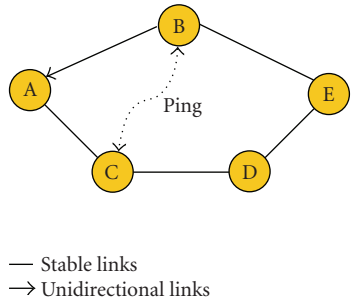
This scenario, whose network topology is shown in Figure 9, is characterized by the unidirectional link A → B, and the experiments consist of ping operations from B to C and vice versa. This scenario differs from the previous one since the two possible routes to reach the destination do not share any

[4] AODV identifies a duplicate RREQ packet from the packet identifier and the Originator IP address that are maintained also in the forwarded packets.



Scenario 2	AODV packet loss	OLSR packet loss
(a) Ping: C → B	98%	0%
(b) Ping: B → C	77.3%	0.33%

Figure 9: Unidirectional links: scenario 2 (network topology and experimental results).



Scenario 3	AODV packet loss	OLSR packet loss
(a) Ping: C → B	100%	0%
(b) Ping: B → C	88.6%	0%

Figure 10: Unidirectional links: scenario 3 (network topology and experimental results).

intermediate nodes. In the first experiment (after the initial phase of 30 seconds during which the network topology is stabilized) node C pings node B for 120 seconds. When node A receives the first RREQ packet from C, it does not know B as 1-hop neighbor due to the unidirectional link, and it has to forward the RREQ. When B receives the RREQ from A, it realizes to have a 2-hop route to C and sends a RREP message to A, which is lost. At the same time, when node E receives the same RREQ forwarded by node D, it sends back to C a RREP announcing the route C-D-E-B. At this point the source node C stores in its routing table the correct route, and starts sending ICMP requests to B. However, when B receives the ICMP requests, it sends ICMP-reply packets on the route B-A-C, not realizing that it is an erroneous path. In this way, two different paths are used by ICMP requests and replies, since nodes C and B have an asymmetric view of the network topology. In addition, B has the possibility to discover the alternative route only in case it temporarily loses Hello-like RREPs from A. In this case it has to send a RREQ to find a route towards C to reply to ICMP packets, and it can exploit the bidirectional path to successfully deliver application packets. However, this route is maintained only for few packets, since B updates its routing table every time it receives a Hello-like RREP from A.[5] In this experiment, we measured an average packet loss of 98% due to rare route reconfigurations. Even in this case the Blacklisting does not avoid the use of the unidirectional link since node B does not discard Hello-like RREPs received from A. Instead, OLSR successfully delivers all the application packets using the correct path C-D-E-B, experiencing an average delay of 10.83 milliseconds to complete the first ICMP handshake, and 6.64 milliseconds as average RTT for the entire ping operation.

Considering then the reverse ping operation from B to C, AODV experiences a 77.3% average packet loss because, when B starts the route discovery procedure, its RREQ is not received by node A due to the unidirectional link, but it reaches node D that directly knows the final destination. Hence, B discovers the right path for ICMP request (B-E-D-C). When the first ICMP request reaches node C, this one does not have a valid route to the source node B, and it has to execute a new route discovery. In this case, also node A receives the RREQ and forwards it to node B (because A does not know B as 1-hop neighbor). At this point, B realizes that exists a 2-hop route to reach the final destination through A, and it updates its internal routing table. At the same time, node C recovers the route back to B from node D and it successfully complete the first ICMP handshake. However, C does not receive the following ICMP requests since B sends them through node A causing their failure. Thus, the first ICMP handshake is successfully completed with an average delay of 17.76 milliseconds, and the correct path is re-established during the experiment due to the loss of some Hello-like RREPs from A, reducing the packet loss and correctly completing

[5] Actually, AODV-UU updates the kernel routing table with a 1-hop route only after receiving three consecutive Hello-like RREPs from the same node, while in the protocol specification the reception of 1 Hello-like RREP is sufficient to update the routing table. This difference slightly increases the time interval in which the protocol is able to maintain the alternative (stable) path, after the lost of a Hello-like message.

the remaining ICMP handshakes with an average RTT of 37.15 milliseconds.

OLSR also in this case experiences no packet loss directly discovering the bidirectional path B-E-D-C. It introduces an average delay of 9.653 milliseconds to complete the first ICMP handshake, and 4.44 milliseconds as average RTT on the entire ping operation. Note that, even though AODV correctly completes the first ICMP handshake, it introduces a higher delay than OLSR, and the average RTT is affected by the possible route reconfigurations due to the loss of Hello-like messages.

6.1.3. Scenario 3

As final set of experiments, we examine the scenario shown in Figure 10 where the same ping operations between nodes B and C are executed. The only difference with the previous scenario is the direction of the unidirectional link (B → A). In this case, when node C starts pinging node B, node A receives the RREQ generated by C and it replies with the 2-hop route C-A-B since it receives Hello-like messages from B. For this reason even in this experiment C measures a 100% packet loss. The alternative route can be discovered only if A loses some Hello-like RREPs from B, because in this case it has to forward the RREQ to B that is lost. As a consequence, B has to accept the reverse path B-E-D-C as a valid route. Instead, as in all the previous experiments, OLSR experiences no packet loss, measuring average delays of the same order of the previous results.

Finally, when node B pings node C, we point out the Blacklisting failure. Specifically, when B executes the first route discovery, both its 1-hop neighbors (A and E) receive the RREQ message. Node A directly sends the RREP to B, but the packet is lost due to the unidirectional link. Thus, A adds B to its blacklist. At the same time, E forwards the RREQ originated by B, and receives the RREP from D announcing the 2-hop route C-D-E, that is then forwarded to the source node B. Thus, the first ICMP packet reaches the destination C, but at this point C has to execute a route discovery to B since it has no available route to that destination. Hence, C's RREQ is received by both A and D. Even though node A "blacklists" B as source of a unidirectional link, it continues to consider it as a valid 1-hop neighbor, since the Blacklisting technique is not applied to the transmission of Hello-like RREP messages.[6] Thus, A sends a RREP to C announcing the route C-A-B, and C tries to send the ICMP replies through that route. In this way, all the ICMP replies fail, and only in case C loses some Hello-like RREPs from A it is able to discover and use the correct path. For this reason, in this experiment we experienced a 88.6% average packet loss since only few packets use the route B-E-D-C with an average RTT of 15.7 msec. Even in this case OLSR measures no packet loss and an average RTT of 4.32 msec.

To summarize, all the presented experiments highlight that AODV fails with high probability in presence of unidirectional links, while OLSR implementing Hello packets technique to discover only bidirectional 1-hop neighbors completely avoids route failures. This technique could also improve AODV performance both in small- and medium-scale networks, even though it increases the traffic load. However, in the reality it is not common to find perfect unidirectional links while it is highly probable to have asymmetric links, that is, links affected by a not negligible packet loss in one or both directions. To point out routing protocols performance in such conditions, we report in the next section some experimental results related to different scenarios. Some of them try to reproduce scenarios 2 and 3 analyzed in this section, while the others consider an extreme scenario in which even OLSR performance is poor.

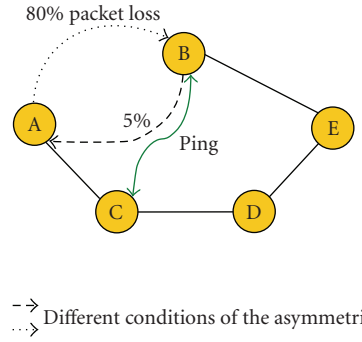
6.2. Asymmetric links experiments (real multihop configurations)

In order to compare performance results in presence of unidirectional and asymmetric links in the same network topology, we firstly set up a real testbed that reproduces scenario 3. As previously mentioned, it is very difficult to configure the network topology to exactly reproduce the same link instability conditions for all the experiments as they depend on the interference conditions that in real environments are highly variable.

To present average values of the performance indices, we had to repeat the same experiment several times, even in different days, trying to establish everytime the same conditions of the asymmetric link on each scenario. Nevertheless, we were not always able to obtain the same configuration. For this reason, the link conditions are slightly different when using the two routing protocols as they are shown in Figures 11 and 12 running AODV and OLSR, respectively.[7] In both cases, we measured different packet losses on the link A-B, especially in the direction from node A to node B. However, in both cases the asymmetric link tends to be a unidirectional link, even though some packets can flow in both directions causing the generation of several route changes during the same ping operation. In fact, in these experiments we define a new performance index as the number of route changes experienced by both protocols during every ping operation involving nodes B and C (averaged over the number of trials). The analysis of the experiments highlights that OLSR always outperforms AODV executing the ping operation from B to C and vice versa (see tables in Figures 11 and 12). Specifically, in the worst case, that is, executing the ping operation from B to C, AODV experiences a 30.33% packet loss and 3 route changes on average, while OLSR introduces a 10% packet loss using always the stable path B-E-D-C to reach the destination. Referring to the delays experienced by both protocols, it

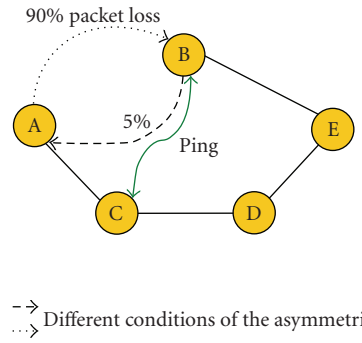
[6] Note that node A has sufficient information to avoid the use of the unidirectional link B → A, but it is not correctly managed. In fact, it would be sufficient that A checks whether B is blacklisted everytime it receives a Hello-like RREP from it before considering it as a valid 1-hop neighbor. The Blacklisting specification does not take into account this feature.

[7] To evaluate the link conditions in terms of packet loss at the beginning of each experiment, we used an asymptotic traffic generated by NetPerf [20], while we continued to use the ping utility to evaluate the performance of the routing protocol in the multihop configuration.



Scenario 3 asymmetric link	AODV (average values)			
	Packet loss	No. of route changes	Delay 1st ICMP handshake	Average RTT
(a) Ping: C → B	19%	1	1007 ms	61.66 ms
(b) Ping: B → C	30.33%	3	486.8 ms	35.32 ms

Figure 11: Asymmetric links: AODV performance on scenario 3.



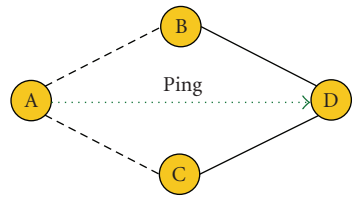
Scenario 3 asymmetric link	OLSR (average values)			
	Packet loss	No. of route changes	Delay 1st ICMP handshake	Average RTT
(a) Ping: C → B	2%	0	9.81 ms	4.928 ms
(b) Ping: B → C	10%	0	8.42 ms	11.07 ms

Figure 12: Asymmetric links: OLSR performance on scenario 3.

is worth pointing out that OLSR introduces less than 10 msec delay to complete the first successfull ICMP handshake in both directions, while AODV experiences a maximum delay of 1 second in the ping operation from C to B, since it initially tries to send the ICMP reply on the unstable path B-A-C.

These results highlight the difference between the case of unidirectional and asymmetric links. Unidirectional links represent the extreme case in which packets transmission is allowed in only one direction, and the reactive routing protocol maintains constantly an asymmetric view of the network topology at the end nodes of the related links. Instead, in the reality, the characteristics of an asymmetric link vary over time, and the sporadic transmissions and receptions of routing packets cause several route changes. In both cases, our analysis pointed out the advantages of using a proactive protocol that guarantees to deliver a higher number of packets using always the most stable path. In addition, referring only to the OLSR behaviour in presence of asymmetric or unidirectional links, we observed that it obviously performs better when the link is perfectly unidirectional. In fact, in this case, OLSR always uses the bidirectional path experiencing no packet loss. Instead, when asymmetric links are present, OLSR tends to alternative stable and unstable paths decreasing its performance.

Varying the interference conditions on the asymmetric link (i.e., measuring valuable packet loss in both directions), we noticed that both protocols are affected by a not negligible packet loss caused by several route changes. Specifically, even



Ping: A → D	Packet loss	No. of route failures
AODV	29.6%	3.6
OLSR	40%	6

Figure 13: Network topology with two unstable links and related experimental results.

though the packet loss measured by both protocols is of the same order of magnitude, OLSR experiences a lower number of route changes than AODV. In fact, due to the characteristics of the asymmetric link, it is possible that sometimes nodes A and B are able to exchange Hello packets. However, running OLSR the link A-B is considered as a valid 1-hop route only after the bidirectionality check (see Section 2). Instead, in case of AODV the single transmission of a Hello-like RREP is sufficient to consider a node as a valid 1-hop neighbor. Thus, AODV suffers more route changes than OLSR.

Note that all the experiments summarized in the three scenarios presented in these sections are characterized by two possible paths connecting source and destination nodes, and one of them is affected by an unstable link while the other is stable. Performance results highlight the disadvantages of using unstable links as valid routes both in terms of packet loss and delays, causing frequent and repeated route discovery failures (mainly in the reactive protocol). However, when a unique path, characterized by an unstable link, connects source and destination, the behavior of the two routing protocols differs. In fact, in this condition, the reactive protocol is able to use the unstable path to deliver at least few packets. On the contrary, no packets are successfully delivered when the proactive protocol is used. The same behavior can be observed when there are multiple paths between source and destination, all affected by unstable links. In this case, both protocols alternative the available paths to deliver applications packets further increasing the number of route changes. As an example we analyzed a real multihop scenario in which all the routes to a specific destination are characterized by an unstable link affected by a 50% packet loss in both directions (see Figure 13). In this case, nodes A and D are used as the end nodes of the ping operation.

The experiment outline is the same of the previous ones: all the nodes start running only the routing protocol for 30 seconds, then node A pings node D for 120 seconds. The same experiment has been repeated three times to present an average value of the performance results. Just from the analysis of the packet loss, it is clear that in this case AODV performs better than OLSR. In fact, with OLSR we observed (on average) a 40% packet loss while only 29.6% running AODV. This is also due to the fact that in this situation the ping operation is chacterized not only by route changes but also route failures, that is, sometimes both protocols lose all the routes to the destination, consequently losing the application packets. Specifically, OLSR experiences (on average) 6 route failures during a ping operation that cause every time (on average) the loss of 7 ICMP packets. Instead, AODV experiences (on average) 3.6 route failures characterized by the loss of 3.13 ICMP packets.

On the opposite of the previous experiments, in which the constraint of link bidirectionality guarantees high performance to OLSR, in this case it represents the main cause of its high packet loss. In fact, the loss of some Hello packets in one of the two directions of the asymmetric link can compromise the validity of the entire route. Therefore, if an alternative stable route exists, OLSR uses it for all the time introducing no packet loss and supporting high application performance. Instead, if all the available routes are unstable, OLSR experiences several route failures and a consequent high packet loss, while AODV is able to deliver a higher number of application packets.

Hence, from these results we can conclude that the presence of asymmetric and unidirectional links generally penalizes AODV performance more than OLSR if at least an alternative stable path exists. Instead, whether the only available route is unstable, AODV gives to the application the opportunity to exploit the unstable link to deliver even few packets. However, in this case, the real advantage of delivering a small percentage of packets rather than nothing strictly depends on the application.

7. CONCLUSIONS

In this work, we deeply analyze the influence of both unidirectional and asymmetric links on AODV performance through real experiments. We set up a small ad hoc network testbed considering specific scenarios affected by this kind of links and we divided the performance evaluation into two parts. First, we used iptables firewall to configure different network topologies emulating multihop connections and forcing the establishment of unidirectional links. In these scenarios, we compared AODV performance, implementing the Blacklisting technique, and OLSR, that implements Hello packets technique to establish only bidirectional paths. All the experiments were characterized by two possible paths from the source to the destination, one of which involving the unidirectional link. From the performance results, we identified possible failures of Blacklisting technique in AODV and the advantages of using only stable paths. Then, since in real conditions it is much more likely to find asymmetric links than unidirectional links, we reproduced some of the previous scenarios in a real multihop configuration replacing the unidirectional link with an asymmetric one. In

both conditions we highlighted that OLSR generally outperforms AODV measuring very low packet loss and using always the most stable path. Nevertheless, these results strictly depend on the assumption that there exists at least a stable path to reach the destination. In fact, examining a specific scenario in which all the possible routes are characterized by an unstable link we found that OLSR performance is poor since it is not able to maintain a stable route to the destination, while AODV is able to exploit also the unstable link achieving a packet delivery ratio higher than OLSR. Thus, in case all the available paths are unstable, AODV gives to the application the opportunity to deliver at least few packets, but if a stable path exists the probability of route changes generally increases running AODV, consequently producing higher packet loss and delays. In addition, in the specific case of unidirectional links AODV is not able to discard the erroneous paths producing repeated route discovery failures and it prevents the application from correctly executing. Hence, AODV needs a correct policy to manage unidirectional links, and our results indicate that Hello packets technique is the most effective even though it slighthly increases the traffic load. On the other hand, OLSR, in addition to maintain a complete knowledge of the network topology, exploits only bidirectional links guaranteeing a higher packet delivery ratio using anyway the stable route, if any.

Therefore, to merge the advantages of both protocols limiting the side effects, a hybrid approach, enhanced with a cross-layer interaction with the upper-layer services could represent a good solution. In this case, the routing protocol could dynamically decide to consider or not unstable links as valid routes depending on the applications requirements, and on possible link status information derived from the MAC layer.

REFERENCES

[1] E. Borgia, M. Conti, F. Delmastro, and L. Pelusi, "Lessons from an ad-hoc network test-bed: middleware and routing issues," *Ad Hoc & Sensor Wireless Networks*, vol. 1, no. 1-2, pp. 125–157, 2005.

[2] E. Borgia, "Experimental evaluation of ad hoc routing protocols," in *Proceedings of the 1st International Workshop on Pervasive Wireless Networking, in Conjunction with the 3rd IEEE International Conference on Pervasive Computing and Communications Workshops*, pp. 232–236, Kauai Island, Hawaii, USA, March 2005.

[3] E. Borgia, M. Conti, F. Delmastro, and E. Gregori, "Experimental comparison of routing and middleware solutions for mobile ad hoc networks: legacy vs cross-layer approach," in *Proceedings of the ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis (E-WIND '05)*, pp. 82–87, Philadelphia, Pa, USA, August 2005.

[4] E. Borgia, M. Conti, F. Delmastro, E. Gregori, and A. Passarella, "MANET perspective: current and forthcoming technologies," in *Proceedings of the 15th IST Mobile & Wireless Communications Summit*, Mykonos, Greece, June 2006.

[5] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," October 2003, RFC 3626.

[6] S. R. Das, R. Castañeda, and J. Yan, "Simulation-based performance evaluation of routing protocols for mobile ad hoc networks," *Mobile Networks and Applications*, vol. 5, no. 3, pp. 179–189, 2000.

[7] S. R. Das, C. E. Perkins, and E. M. Royer, "Performance comparison of two on-demand routing protocols for ad hoc networks," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, vol. 1, pp. 3–12, Tel Aviv, Israel, March 2000.

[8] R. S. Gray, D. Kotz, C. Newport, et al., "Outdoor experimental comparison of four ad hoc routing algorithms," in *Proceedings of the 7th ACM Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '04)*, pp. 220–229, Venice, Italy, October 2004.

[9] W. Kiess and M. Mauve, "A survey on real-world implementations of mobile ad-hoc networks," *Ad Hoc Networks*, vol. 5, no. 3, pp. 324–339, 2007.

[10] G. Anastasi, E. Borgia, M. Conti, and E. Gregori, "Wi-fi in ad hoc mode: a measurement study," in *Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications (PerCom '04)*, pp. 145–154, Orlando, Fla, USA, March 2004.

[11] H. Lundgren, E. Nordstrom, and C. Tschudin, "Coping with communication gray zones in IEEE 802.11b based ad hoc networks," in *Proceedings of the 5th ACM International Workshop on Wireless Mobile Multimedia (WOWMOM '02)*, pp. 49–55, Atlanta, Ga, USA, September 2002.

[12] OLSR Implementation. Institute for Informatics, Oslo University (Norway). http://www.olsr.org/.

[13] AODV Implementation. Department of Information Technology, Uppsala University (Sweden). http://user.it.uu.se/~henrikl/aodv/.

[14] G. Anastasi, E. Borgia, M. Conti, E. Gregori, and A. Passarella, "Understanding the real behavior of mote and 802.11 ad hoc networks: an experimental approach," *Pervasive and Mobile Computing*, vol. 1, no. 2, pp. 237–256, 2005.

[15] M. Gerla, L. Kleinrock, and Y. Afek, "A distributed routing algorithm for unidirectional networks," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '83)*, San Diego, Calif, USA, December 1983.

[16] S. Nesargi and R. Prakash, "A tunneling approach to routing with unidirectional links in mobile ad-hoc networks," in *Proceedings of the 9th International Conference on Computer Communications and Networks (ICCCN '00)*, pp. 522–527, Las Vegas, Nev, USA, October 2000.

[17] M. K. Marina and S. R. Das, "Routing performance in the presence of unidirectional links in multihop wireless networks," in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MOBIHOC '02)*, pp. 12–23, Lausanne, Switzerland, June 2002.

[18] "Ad Hoc On-demand Distance Vector Routing," http://www.ietf.org/rfc/rfc3561.txt?number=3561.

[19] M. K. Marina and S. R. Das, "On-demand multipath distance vector routing in ad hoc networks," in *Proceedings of the 9th International Conference on Network Protocols (ICNP '01)*, pp. 14–23, Riverside, Calif, USA, November 2001.

[20] Netperf Traffic generator. http://www.netperf.org/netperf/NetperfPage.html.