Journal of Mobile Multimedia, Vol. 12, No.1&2 (2016) 031-051

CoreASM-BASED EVALUATION OF THE N-AODV PROTOCOL FOR MOBILE AD-HOC NETWORKS

ALESSANDRO BIANCHI, SEBASTIANO PIZZUTILO, GENNARO VESSIO

Department of Informatics, University of Bari, Bari, Italy
{*alessandro.bianchi, sebastiano.pizzutilo, gennaro.vessio*}*@uniba.it*

Network topology awareness (NTA) is a fundamental issue in the context of computer networks: the lack of control over the topology can negatively impact performance, security, resilience, and so on. However, in Mobile Ad-hoc NETworks (MANETs), NTA is difficult to achieve because of their dynamicity. Different reactive protocols for MANETs, in which routes are established only when needed, provide different NTA to each host, depending on their algorithmic features. NACK-based AODV (N-AODV) is a variant of the well-known Ad-hoc On-demand Distance Vector (AODV) reactive protocol for MANETs which we proposed with the aim of improving the NTA of the original protocol. In this paper, an experiment aimed at comparing N-AODV to AODV is reported; it moves from an exploratory case study we conducted preliminarily. The results obtained show that a MANET adopting N-AODV exploits higher NTA than a MANET adopting AODV. Moreover, the improved awareness positively impacts effectiveness.

Key words: MANETs, network topology awareness, N-AODV, Abstract State Machines, empirical studies

1 Introduction

Network topology awareness (NTA) is a fundamental issue in the context of computer networks. Optimizing the resource allocation, minimizing the traffic overhead, reconfiguring the system after failures are only a few examples of how networks can benefit from an increased NTA. Conversely, the lack of control over the current topology can negatively impact performance, security, resilience, and so on. Much research in recent years has focused on improving NTA in several domains, e.g. peer-to-peer networks [39], Cloud systems [17] and high-performance clusters [23]. However, few studies have addressed this issue in the specific Mobile Ad-hoc NETwork (MANET) context, e.g. [40] and [42]. The present paper is focused on experimentally evaluating the improvement of NTA in MANETs adopting a routing protocol we recently proposed.

A MANET is a collection of nomadic hosts which communicate in a wireless fashion without the need of a fixed physical infrastructure: communication sessions between initiator and destination pairs are established and maintained by the cooperation of the hosts in the network [2]. The use of these networks spans in a wide range of application domains: rescue operations in case of disasters, vehicular services, data tracking of environmental conditions, and so on. MANET hosts are intended as autonomous agents, that is they can arrange themselves without conforming to a predefined topology. Moreover, during their lifetime, they can enter or leave the network at will and continuously change their relative position, so routes connecting hosts can rapidly change. In such a context, NTA encompasses the knowledge each host has about the existence of other hosts and their reachability through a route, which can traverse several intermediate hosts.

The main difference between MANETs and networks with infrastructure is that the latter can rely on centralized administrations, infrequent resizes of the population and rare topology changes. Instead, in MANETs each node has only a local view of the topology, which is limited to its neighborhood, and reaching a correct and at the same time persistent global view of the entire system is difficult due to dynamicity. Therefore, NTA is generally lower in MANETs than in networks having an infrastructure. Despite this, several MANET applications require NTA to be as high as possible: leader election algorithms, in which all hosts must agree upon the current leader, e.g. [45]; protection against attacks from malicious hosts, that should be identified [22]; high-mobility scenarios, e.g. Vehicular Ad-hoc NETworks (VANETs) [1], in which the topology changes very quickly, so information should be as up-to-date as possible; and so on.

The MANET community proposes several routing protocols to properly manage the lack of a fixed infrastructure and different protocols, depending on their algorithmic features, provide hosts with different NTA. In *proactive* protocols, such as Destination-Sequenced Distance-Vector (DSDV) [35], routes to all hosts in the network are discovered in advance and this information is constantly updated. Therefore, the NTA of each host is rather high, even if the required traffic overhead strongly impacts the overall performance. Conversely, in *reactive* protocols, such as Ad-hoc On-demand Distance Vector (AODV) [34], routing activities are performed only on-demand and this information is updated only when needed. When these protocols are used, the NTA of each host is variable and strictly depends on the information the adopted protocol produces. This information depends in turn on the content of the control packets the protocol disseminates through the network. In fact, every time a host receives a control packet, both directed to it or to be forwarded to another recipient, it updates its knowledge about the current topology on the basis of the content of the received packet.

Taking into account these considerations, we proposed a variant of AODV aimed at providing hosts with higher NTA. We called the variant NACK-based AODV (N-AODV) because the improvement of NTA is obtained through the introduction of a new unicast packet: a Not-ACKnowledgement (NACK) packet. In [6] we formally specified the protocol by means of an Abstract State Machine (ASM)-based model [19] and we proved its correctness. Moreover, in [8] we reported an exploratory case study aimed at comparing N-AODV to AODV with respect to effectiveness, efficiency and NTA. Since the empirical study and the measurement of NTA is a novelty in this research field, the exploratory case study was motivated by the need to clearly specify the design of the experiment here conducted. The provisional findings of [8] showed that N-AODV indeed improves the effectiveness and the NTA of the original protocol. However, we recognized that the results obtained could be affected by general threats to validity [47] and specific “pitfalls” of MANET simulation studies [26], [3].

The present paper reports an experiment aimed at increasing the confidence in the results obtained from the exploratory case study and improving its validity. As in our previous work, the experiment here reported has been executed within a simulation environment. However, the present study varies several issues with respect to [8], mainly concerning the environment in which the experiment is run and the set of dependent variables under investigation. With respect to the experimental environment, we have used the ASM-based development tool CoreASM [15] for implementing the simulator, instead of C++. Moreover, we have altered some features of the simulation to address some threats to validity which our previous study suffered. Thirdly, we have considered different MANET scenarios. Concerning the variables, we have analyzed one more metric to measure NTA in order to investigate this issue also from the point of view of the freshness of the information known by each host.

The rest of this paper is organized as follows. Section 2 is about related work. Section 3 describes both AODV and the variant we proposed. Section 4 deals with the experimental comparison of the two protocols, which is then discussed in Section 5. Finally, Section 6 concludes the paper.

2 Related Work

Many studies have been devoted to the improvement of NTA. The need to address this issue has mainly arisen in three domains: peer-to-peer networks, Cloud systems, and cluster and Grid systems. In peer-to-peer networks, peers randomly join and leave the network at will: this causes topology mismatches between the logical overlay network and the underlying physical topology [29]. Typically, overlay networks are not aware of the underlying physical behavior, so redundant traffic, load imbalance and extra delay in message delivery can occur [39], [27], [32]. This issue is also crucial in Cloud systems: the assignment of resources to users is typically unaware of the underlying network layout, so the network usage is often penalized [17]. Analogously, in cluster and Grid systems, the resource manager of the applications deployed for these networks is usually independent of the network topology. This causes an inefficient use of the resources, which in turn causes substantial performance degradation [23]. However, implementing the solutions above in MANETs is not feasible, due to the many fundamental differences that distinguish them. In fact, in contrast to networks having an infrastructure, MANETs suffers from: bandwidth limitation, due to the use of wireless channels; frequent topology changes, due to mobility; limited energy, due to the nature of the physical devices that they are usually composed of.

To the best of our knowledge, few works have addressed NTA in the specific MANET domain. For example, because of the similarities existing between P2P overlay networks and MANETs, such as self-organization and the equal nature of nodes, [10] and [42] propose an approach based on the construction of an overlay network over the physical topology. In contrast to these solutions, N-AODV does not add a new layer on the physical one, but modifies the existing route discovery mechanism. In [49] the knowledge about the topology (*link stability awareness* in the paper) is improved by injecting redundant packets into the network. Conversely, N-AODV does not rely on the redundancy of existing packets, but make use of a new control packet.

In [40] a mobile multi-agent based framework is developed to make each node aware about the position of the other nodes. Mobile agents asynchronously collect topology information, such as the node velocity and the node location, through GPS; then distribute them to all the other nodes. This local awareness gives each node a proactively refreshed perception of the network topology. By contrast, our approach reactively produces information only when needed. Moreover, in order to accomplish this, GPS support is not required. Finally, in [38] the authors propose the so-called PRioritized EPidemic (PREP) routing. Each node executes a neighboring discovery algorithm to create and maintain a set of bi-directional links with neighbors. When a node notices a change of its neighborhood set, this information is spread through the network in an epidemic fashion. The awareness of the topology then serves to assign priorities to the exchanged messages in order to increase performance. Also in this case, the main difference between PREP and N-AODV is that the former adopts a proactive approach to exchange topological information, whereas the latter does not alter the reactive nature of the original AODV protocol.

In its original definition, AODV is one of the most widely used routing protocols for MANETs. Indeed, it is one of the protocols currently standardized as RFC (Request For Comments) by the IETF MANET working group [34]. Since its appearance, much study has focused on improving the protocol with respect to several issues. Some variants have been proposed in order to reduce communication failures due to topology changes. Both Reverse-AODV (R-AODV) [24] and Modified-AODV (M-AODV) [41] overcome this problem by building all possible routes between initiator and destination. Other variants deal with energy consumption; for example Modified Reverse-AODV (MR-AODV) [25] selects the best routes on the basis of the energy of the hosts; whereas, Optimized-AODV (O-AODV) [4] prevents hosts from forwarding packets if their remaining energy is under a certain threshold. Finally, several improvements of AODV concern security issues; they deal with: the use of cryptography for securing data packets during their transmission, e.g. Secure-AODV (S-AODV) [51], or the adoption of the so-called trust-methods, in which nodes are taken into account during communications only if they are considered trusted, e.g. Trusted-AODV (T-AODV) [50]. Differently from the variants above, our proposal has been conceived with the purpose of improving the NTA of the original protocol.

The present paper specifically deals with the experimental comparison between AODV and N-AODV. Typically, in order to evaluate protocol performance and compare different solutions, the principal way adopted in the MANET community is the use of tools simulating the network behavior. Several tools are available and widely adopted, for example Network Simulator ns-3 [30] and its variants, like Monarch [28] and OLSR [31]. Other studies (for example [13], [14] and [48]) are conducted using tools developed ad-hoc. In this paper, we followed the latter approach using the Abstract State Machine-based [19] development tool CoreASM [15] for implementing the simulator we have used. Contrary to traditional simulators, several formalisms have been successfully applied for simulating the MANET behavior, for example process calculi [43] and Petri nets [5]. In this work, we chose CoreASM mainly for two reasons. Firstly, the translation of the ASM-based specification of N-AODV we provided in [6] into a CoreASM-based executable model has represented a natural prosecution of our work. Secondly, as we have shown in [7], the ASM framework provides a versatile formalism for capturing the simultaneous execution of the hosts in a MANET and their own computation. Moreover, it is worth noting that ASMs are Turing-equivalent [19], so our CoreASM-based simulator can be considered equivalent, in terms of expressiveness, to the simulation environments traditionally adopted in the literature. To our best knowledge, this is the first time an executable ASM-based model has been used for automatically simulating MANET hosts.

Analogously to other empirical studies on MANETs, e.g. [46] and [20], our simulator requires usual parameters as input: the network size, which allows us to consider different MANET populations; the node mobility, capturing the speed at which nodes change their position; and the number of runs, which expresses the communication attempts between couples of nodes. Another simulation issue is related to the adoption of a mobility model in order to express the movement of the network. According to [16], we adopt a mobility model that abstracts the physical behavior of the nodes and represents the position of each node by its relative position with respect to its neighbors. Concerning the output produced by the simulator, we take into account usual metrics for measuring the effectiveness and the efficiency of both protocols. As well as, for example, in [36] and [37], we consider the rate of success of the protocol executions and the overhead due to the dissemination of control packets through the network, respectively. Instead, since NTA has not been experimentally investigated in the MANET literature before, we propose some metrics for measuring this issue tailored to our purposes.

The experimental simulation has been conducted in accordance with the general framework for empirical studies described for example in [47]. More precisely, the experimental design, the statistical analysis of the outcome and its interpretation are reported in the rest of the present paper in order to ensure the repeatability of the experiment and its statistical soundness. Finally, it is worth noting that the framework provided in [47] also discusses general threats to the validity of the experimental findings. [26] and [3], instead, consider more specific threats in the domain of MANET studies. Both issues are taken into account in this paper.

3 The Protocols

3.1 Ad-Hoc On-demand Distance Vector

Ad-hoc On-demand Distance Vector (AODV) [34] is a reactive protocol: routes are built only on-demand. The protocol combines two mechanisms, namely *route discovery* and *route maintenance,* in order to maintain knowledge of routes stored into routing tables. The routing table associated with each node lists all the discovered (still valid) routes towards other nodes in the network and information on them. In particular, an entry of the routing table of the host *i* concerning a node *j* includes: the *address* of *j*; the last known *sequence number* of *j*; the *hop count* field expressing the distance between *i* and *j*; and the *next hop* field identifying the next node in the route to reach *j*. Note that the sequence number is a monotonically increasing number maintained by each node: it helps other nodes to express the freshness of the information about the nodes for which entries are maintained.

When an initiator node wants to start a communication session to a destination node, it first checks if destination is in its neighborhood (so that it is directly reachable), or a route to destination is currently stored in its routing table. If so, the protocol ends and the communication session simply starts. Otherwise, initiator starts route discovery, in order to discover the desired route, by broadcasting route request (RREQ) packets to all its neighbors. Among the others, an RREQ packet includes: *initiator address* and *broadcast id*; *destination address*; *destination sequence number*, which expresses the latest available information about destination; and *hop count*, initially set to 0, and increased by each intermediate node. Note that the pair *initiator address* and *broadcast id* uniquely identifies the packet; in this way, duplications of RREQs that nodes have handled before can be ignored.

When an intermediate node *n* receives an RREQ, it updates the routing table entry for initiator, concerning both the sequence number of initiator and its next hop field; if an entry for initiator does not exist, it is created. Then the process is reiterated. More precisely, *n* checks if destination is one of its neighbors, or if it knows a route to destination with corresponding sequence number greater than or equal to the one contained into the RREQ (this means that its knowledge about the route is more recent). In both cases, *n* unicasts a route reply (RREP) packet back to initiator. Otherwise, *n* updates the hop count field and rebroadcasts the RREQ to all its neighbors. An RREP packet contains: *initiator* and *destination address*; *destination sequence number* and *hop count*. The process successfully ends when a route to destination is found. While the RREP travels towards initiator, routes are set up inside the routing tables of the traversed hosts by creating an entry for destination when needed. Once initiator receives the RREP, the communication session can start.

Conversely, the route discovery mechanism fails when: no RREQ reaches a node which is in the destination neighborhood; or no RREQ reaches a node whose routing table contains a route to destination; or a previously set timeout expires while initiator is waiting for RREPs. The first two cases depend on the non-reachability of destination; instead, the last case can be due to either the isolation of the destination, or too long distances, or changes in the topology during the packet transmission.

In the event that a link breakage occurs, for example because a previous neighboring node is no longer in the neighborhood, the protocol executes route maintenance. A node *n*, involved in a broken link with a node *m*, firstly removes from its routing table all the entries concerning the destinations reachable through *m*. Then, *n* unicasts a route error (RERR) packet towards initiator to notify the error, so that routes are invalidated. After receiving the RERR, if initiator still requires a route to the unreachable destination, it has to reinitiate the route discovery mechanism.

3.2 NACK-based AODV

AODV is a distance vector protocol, so it does not give nodes a complete view about the topology: each node directly knows only its neighbors and, for non-neighboring nodes, it only knows the next hop to reach them. This knowledge results in a low network topology awareness: we conceived N-AODV with the aim of improving this issue.

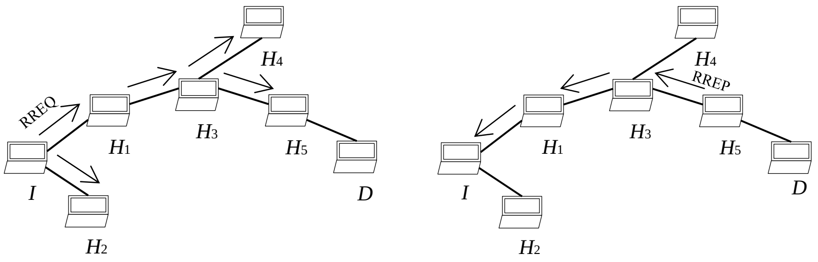
In AODV, when an intermediate node *n* receives an instance of an RREQ and does not know a proper route to reach the desired destination, it simply rebroadcasts the RREQ to all its neighbors. Instead, in N-AODV, in addition to rebroadcasting the RREQ, *n* unicasts a NACK (Not ACKnowledgement) packet back to initiator. The NACK is so used to inform all nodes between *n* and initiator that, roughly speaking, *n* “does not know anything” about destination. Each NACK packet includes the addresses and the sequence numbers of *n* and initiator and the distance, expressed in hops, between them. Note that, albeit a NACK packet expresses ignorance about a desired destination, it provides information gain: it contains information about the node sending it which is spread through the network.

It is worth remarking that our modification only affects route discovery; therefore, for what concerns route maintenance, N-AODV behaves exactly as AODV.

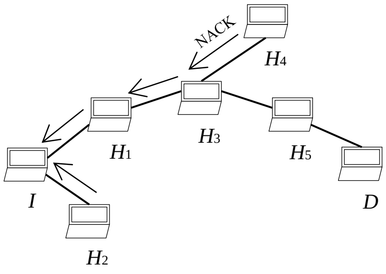
Let us illustrate how the protocol works in a typical scenario. Assume we have the scenario in Fig. 1 in which the MANET has just been initialized (i.e., routing tables are empty): an initiator node I wants to start a communication session to a destination node D. There are several intermediate nodes: H_1, …, H_5. Black lines represent physical connection links. Black arrows represent the dissemination of control packets through the network. In order to discover a route to reach D, RREQ packets are disseminated across the network starting from I (Fig. 1-a). Thanks to the route discovery mechanism, H_3, H_4 and H_5 become aware about the existence of I, because they receive an RREQ produced by I; instead, H_1 and H_2 do not need this information, because I is in their neighborhood. Since H_5 is directly connected to D, it can reply the received request by unicasting an RREP packet back to I (Fig. 1-b). Note that, in order to reach I, the RREP must travel across H_3 and H_1. In this way, in AODV, I is aware about: H_1 and H_2, because they are in its neighborhood, and D, because information about it is contained into the received RREP. Moreover, because of the forwarding activities needed to realize unicasting, also H_1 and H_3 are aware about D. Instead, in N-AODV, all hosts reached by an RREQ, but not able to reply it, unicast a NACK packet back to I (Fig. 1-c). More precisely, H_1, H_2, H_3 and H_4 unicast a NACK to I. In this way, in addition with respect to AODV, I is aware about H_3, which is an intermediate node in the route to reach D, and about H_4, which is a node reached by an RREQ but not involved in the route to reach D. Moreover, because of the NACKs forwarding, H_1 is aware about H_4.

Therefore, during route discovery, in AODV routing tables are updated every time a node receives an RREQ or an RREP. Instead, in N-AODV, routing tables are updated every time a node receives an RREQ, an RREP or a NACK. In particular, the usage of NACKs provides information gain under three points of view. Firstly, when a route to destination is found, initiator is aware not only about the next hop in the route to reach destination, but also about all the intermediate hosts lying in that route (excluded the originator of the received RREP). Note that, even if the route discovery fails, initiator is aware about all hosts reached by an RREQ until the timeout expiration. Secondly, initiator is also aware about all hosts reached by an RREQ but not involved in a route to reach destination. Finally, because of the forwarding activities due to NACKs unicasting, all hosts in the reverse route to reach initiator are aware about the senders of the various NACKs.

In accordance with the description above we expect that N-AODV improves NTA thanks to the addition of the NACK control packet. Therefore, we also expect an increased traffic overhead. In order to evaluate these issues, the empirical approach has been used.



a) Broadcasting of RREQs b) Unicasting of an RREP



c) Unicasting of NACKs

Figure 1. Control packets delivered in AODV and N-AODV.

4 Experiment

In order to compare N-AODV to AODV with respect to NTA, an experiment has been conducted through simulations. Analogously to the exploratory case study [8], the research questions posed for the experiment are:

1. Which protocol provides nodes with higher NTA?
2. Which protocol exploits better effectiveness?
3. Which protocol is more efficient?

Both protocols have been simulated in different scenarios and some metrics have been collected with the aim of answering the questions above.

4.1 The Simulation Environment in CoreASM

In the exploratory case study we were only interested in each single communication attempt between couples of nodes, so we developed an ad-hoc C++ simulator. Instead, in the present work, we are interested into studying multiple communication attempts executed in parallel. Taking advantage of the intrinsic parallelism of the ASM execution, CoreASM makes the implementation of the MANET computation easier.

CoreASM [15] is a development tool for both the design and the experimental validation of ASM models, which supports the execution of their formal specification. Briefly speaking, ASMs are finite sets of so-called *rules* of the form *if condition then updates* which transform the computational states of the machine [19]. An ASM state is an algebraic structure, i.e. a domain of objects with functions and relations defined on them. On the other hand, in a rule, *condition* is a first-order formula whose interpretation can be *true* or *false*; whereas, *updates* is a finite set of assignments whose execution consists in changing in parallel the value of the specified functions and relations to the indicated value. An ASM computational step is a transition from one state to another: in the current state, all conditions are checked, so that all updates in rules whose condition evaluate to *true* are simultaneously executed, and the result is a transition of the machine from that state to another. Several ASMs can then be composed in a so-called Distributed ASM to capture concurrent computations [9]. Each ASM allows us to model the behavior a single node executing the protocol; whereas, the entire DASM is able to represent the overall behavior of the network.

Traditional simulators are effective, mainly for evaluating performance and comparing different solutions, but they are not able to formally model MANETs. In other words, these tools are just simulators: they implement the network at a low abstraction level, but they cannot support specifications at a higher level. Therefore, studying typical problems of mobile systems (e.g., concurrency, synchronization, starvation, etc.) is hard. Conversely, the theoretical foundation of ASMs allows for the representation of systems at high levels of abstraction, so providing general investigations about properties of interest. In fact, ASMs allowed us to formally prove the correctness of N-AODV in [6].

MANET hosts are characterized by several physical features: the amplitude of transmission ranges, the direction and the speed of movement, and so on. Therefore, a realistic MANET model should include some hosts moving quickly, other hosts moving slowly, or stopping; the direction and the speed of movement can be constant for a period of time, then they can change; some hosts can have higher transmission range than others; and so on. However, simulating all aspects of a MANET is very difficult and sometimes impossible. According to [11], "every system model is tailored depending on the goal of the simulation project". For the goal of our study, we adopted a mobility model that subsumes the physical behavior of the hosts, so that the direction and the speed of movement, as well as transmission ranges, are abstractly represented within the mobility model. In fact, we can assume that the topology of the network is implicitly defined by the connections among hosts, i.e. by the current position of each node together with the amplitude of its transmission range. Therefore, for the purposes of the simulation, for each host we take into account only its current neighborhood. The MANET topology is so represented by an $N \times N$ symmetric *connectivity matrix C* such that:

$$c_{i,j} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are neighbors} \\ 0 & \text{otherwise,} \end{cases}$$

where $1 \le i, j \le N$, with N indicating the network size. The hosts' mobility is then simulated through the random re-definitions of the values of the $c_{i,\,j}$ elements. As a result, the transmission ranges of the hosts are not static, but are dynamically redefined every time C changes, and they are considered symmetric, i.e. if node n can directly communicate to node m, then m can directly communicate to n. Thanks to this approach, the direction and the speed of movement become irrelevant for the purposes of the simulation, because implicitly modeled as evolution of the C matrix, representing the transitions from one set of neighbors to another for each host. Moreover, this approach makes unnecessary a precise definition of the hosts' transmission ranges, because implicitly defined by the concept of neighborhood. For the purposes of our research, the dynamic and symmetric characteristics of transmission ranges, albeit not completely realistic, are not a pitfall, as discussed later in the paper. Our approach is very similar to that followed in [16], in which the authors show that this mobility model is able to cover the main aspects of traditional mobility models, such as random walk and random way point.

Typically, MANET empirical studies, e.g. [36] and [18], take into account several communication attempts executed in different scenarios. In the present study a scenario is defined by the pair <*network size*, *node mobility*>. The values considered for *network size* are 10, 25 and 50 hosts. The choice of these values is confirmed by looking at several well-known empirical studies on MANETs, e.g. [46], [20] and [44]. On the other hand, according to the literature (e.g., [13] and [36]), the values considered for *node mobility* are distinguished in *high* and *low*. This parameter expresses the probability to change an element of the connectivity matrix so that a new physical link is established or a pre-existing physical link is broken. In order to precisely define the values for *high* and *low*, a number of preliminary trials were executed before the study. Thanks to these trials we could establish that the *high* value is 3/20, i.e. the probability to change an element in the C matrix from a 0 to a 1 or vice versa is 3/20. Analogously, the *low* value is 1/20. Moreover, note that the initial setting of the connectivity matrix is random. In the exploratory case study, the same mobility model was adopted. However, we used the expression *host connectivity*, instead of *node mobility*, to emphasize the representation of movement as randomized evolution of the connectivity matrix. Actually, they express the same concept. Furthermore, note that the values chosen for this parameter are different. In fact, the choice of these values depends on the inherent characteristics of the adopted simulator, so we could not use the values we chose in the exploratory case study.

We executed 500 runs of the simulator over the six different scenarios above for each protocol. Each run corresponds to a new communication intention by an initiator node, that is in each run an initiator-destination pair is randomly chosen. A run is successful when: the destination is found in the neighborhood of initiator; or a route to destination is recorded into the initiator's routing table; or a route to destination is discovered by the route discovery mechanism. Otherwise, a run fails when: the destination cannot be reached, i.e. because of the lack of direct or indirect links from initiator; or when the timeout expires. We established the timeout expiration after the execution of 25 ASM computational steps of the model of initiator. According to [21], thanks to this approach, the times of events are completely under the control of the experimenters. Moreover, the approach provides a more realistic model: the activity of each host is driven by its own internal behavior, instead of an unrealistic external centralized clock.

Note that in the exploratory case study we took into account scenarios with 10, 20 and 30 hosts and, for each scenario, we executed 300 runs. It is also worth noting that the timeout mechanism was not implemented in our previous study, so we were not able to take into account the effects of transmission delay. Moreover, the results were obtained by sequential runs, i.e. each run corresponded to a single communication attempt involving a couple of nodes, and the topology randomly changed only at the end of each run. Instead, in the present experiment, firstly we have introduced the timeout, so that the results can be affected by transmission delay. Secondly, multiple runs can be now simultaneously executed, i.e. at any one time several communication attempts involving different couples of nodes can be in progress. Moreover, the topology is not frozen during the run execution, then changed at the end of each run, but can randomly change at any time.

One more remark concerns the independence of runs. It has been ensured by adopting the following simulation strategy: for each protocol, the C matrix is randomly set in the first run of each scenario's simulation; then, it is randomly re-defined during each simulation execution.

Finally, note that also the network size is expressed at the beginning of the first run, but it can decrease because of the simulated movement: some hosts can go outside the reachability range of the overall network, so becoming isolated. This happens when a row in the connectivity matrix representing a host's neighborhood only includes 0s. Nevertheless, isolated hosts can re-join the network when at least one 0 in their rows becomes a 1 again.

4.2 Metrics and Research Hypotheses

In the present work, as well as in our previous study, we compare N-AODV to AODV with respect to NTA, effectiveness and efficiency. In order to evaluate NTA, the following metrics have been measured (in the following N is the network size and $|RT_i|$ is the number of the entries in the routing table of the i-th host).

Routing tables size (RTS). For each run, it is the total number of the routing tables entries in all hosts:

$$RTS = \sum_{i=1}^{N} |RT_i|.$$

Routing tables updates (RTU). For each run, it is the total number of times routing table updating activities are executed by all hosts.

Network awareness lag (NAL). For each run, it expresses the delay in storing up-to-date information in the routing table of every host. It is given by:

$$NAL = \frac{1}{N}\sum_{i=1}^{N} RT\_lag_i,$$

where RT_lag_i is the delay in storing up-to-date information in the routing table of the i-th host. It is given by:

$$RT\_lag_i = \frac{1}{|RT_i|}\sum_{j=1}^{|RT_i|} \Delta_j\,,$$

where Δ_j is the difference between the actual sequence number of host j and the value of the sequence number stored into the j-th entry of the routing table of host i. It is given by:

$$\Delta_j = act\_seq\_num_j - known\_seq\_num_j.$$

In other words, Δ_j represents the delay of the awareness about the j-th host; RT_lag_i represents the average delay about all hosts known by the i-th host; *NAL* represents the average delay of the awareness of all hosts about the entire network.

Broadcast activations (BA). For each run, it counts the total number of times RREQ packets have been broadcasted during the possible execution of a route discovery mechanism.

In order to evaluate effectiveness, we have taken into account the rate of success of each protocol.

Rate of success (RS). It is the ratio between the total number of times the protocol successfully ends and the total number of runs.

The efficiency of the protocols has been measured by the traffic overhead induced by the control packets disseminated through the network in each run; in the following simply control overhead.

Control overhead (CO). For each run, it is the sum of RREQs and RREPs in the case of AODV. It is the sum of RREQs, RREPs and NACKs in the case of N-AODV.

The choice of the metrics for evaluating effectiveness and efficiency is usual in MANET studies, so it does not need further discussion. However, some comments are needed for the NTA metrics. Routing tables express the knowledge each node in the MANET has about the existence of a route to reach other nodes. When a node *n* enters the MANET, it does not know anything about other nodes, so its routing table is empty. Then, entries are added as a result of the discovery of routes, or removed because of link breakages. Therefore, the size of the routing table of node *n* gives account of the number of the hosts in the MANET *n* knows. However, this metric only provides a partial view of NTA, because it does not consider updates of existing entries in routing tables, due to the discovery of new, up-to-date routes towards already known nodes. In order to consider this issue, the metric measuring routing tables updates is also investigated: it gives account of both adding/updating entries. NTA is also considered from the point of view of the freshness of the information stored in routing tables. In fact, by checking sequence numbers, NAL implicitly measures how the information about the discovered routes is up-do-date on the average. Remember that up-to-date information implies a greater chance that the already known routes are still valid. The metric gives account of the delay each host has in knowing the current network topology, which in turn depends on how the already discovered routes are obsolete. It is worth noting that this metric was not taken into account in our previous study. Lastly, one more view on NTA concerns the ignorance a host has about the topology. In order to measure this issue, for each run we consider the number of times all hosts need to broadcast RREQ packets. In fact, this need arises every time a node ignores a way to reach the destination.

It is worth remarking that some metrics we considered in the previous study are not taken into account in the present work. They are the rate of success due to topology, the rate of success due to initiator's awareness and the rate of success due to route discovery, regarding effectiveness; and the number of RREQs and RREPs disseminated through the network in each run, regarding efficiency. Indeed, they simply represent partial contributions to the rate of success and control overhead, respectively. Actually, since the rate of success and the control overhead are sufficient to express the effectiveness and the efficiency of the protocols, we here focus only on them.

Note that only the rate of success is normalized with respect to the total number of runs, because aimed at expressing percentages. Instead, the NTA and the efficiency metrics only consider the values observed in each run, so normalization is not necessary. In this view, in order to answer research questions (1) and (3), concerning NTA and efficiency, respectively, we state the following research hypotheses for each related metric:

- H_0 (null hypothesis): there is no statistically significant difference for that metric between AODV and N-AODV;
- H_a (alternative hypothesis): there is a statistically significant difference for that metric between AODV and N-AODV.

In order to answer research question (2), the rate of success is not studied with the same approach because it is characterized by only one value for each protocol; so, in each scenario, the values for both protocols are simply compared.

In order to collect these metrics, we have implemented an *observer agent* that supervises the MANET computation from an external point of view. This solution makes the data collection easier; moreover, it does not impact their values: the computation of the observer agent only makes the simulation slower, but the chronological time is not considered in our study.

Finally, it is worth noting that data have been collected at the end of each run. However, since multiple runs can be executed in parallel, collected data are not necessarily concerned with the communication attempt between the two nodes selected at the beginning of the last executed run. In fact, they can concern with previously chosen initiator/destination pairs whose communication attempt was still in progress.

5 Discussion

5.1 Analysis of the Results

The results of the experiment are reported in Appendix A. In order to discuss the rate of success, it is presented as percentage of successful protocol executions with respect to the total number of runs. Instead, in order to test the research hypotheses concerning the NTA and the efficiency metrics, we report the results of the Mann-Whitney test. We have adopted this non-parametric alternative to the t-test because the two groups of observations that are to be compared are independent of each other and the normality assumption is not always respected. More precisely, according to [12], the Mann-Whitney test is suitable when there are two samples from possibly different populations with the following assumptions: both samples are random samples from their respective populations; in addition to the independence within each sample, there is mutual independence between the two samples; the measurement scale is at least ordinal. The significance level of the test (p-value) evaluates as usual to 0.05, i.e. the test is statistically significant if the p-value is lower than 0.05. Therefore, if the p-value obtained executing the test is lower than or equal to 0.05, then H_0 is rejected and H_a is accepted, otherwise H_0 is accepted and H_a is rejected.

In order to evaluate NTA, we firstly consider the results concerning routing tables. In fact, the entry in the routing table of a host *i* associated with a host *j* describes the information *i* has about *j*: the existence of *j*; its sequence number; the distance in hops; the next hop in the route that the messages from *i* must follow to reach *j*. Table A.1, concerning the results of the Mann-Whitney test for routing tables size, reports that the routing tables of the hosts adopting N-AODV always include more entries than the hosts adopting AODV. Moreover, the difference is always statistically significant. Therefore, for RTS, H_a is accepted. Table A.2, concerning the results of the Mann-Whitney test for routing tables updates, shows that the mean number of times routing tables are updated in N-AODV is always greater than in AODV. However, the difference is not statistically significant in two scenarios (<10 hosts, High mobility> and <50 hosts, High mobility>). Therefore, for RTU, H_a is accepted, except for these scenarios. Table A.3, concerning the results of the Mann-Whitney test for network awareness lag, shows that the average delay in storing up-to-date information in routing tables is always lower in N-AODV then in AODV and the difference is always statistically significant. Therefore, for NAL, H_a is accepted. The three issues above mean that, in general, N-AODV allows every host to have *more* information about the topology than AODV, that this information is updated more *frequently* and that is more *recent*, in the sense that it refers to knowledge gained with lower delay. The more efficient use of routing tables has effects on the overall behavior of the protocol. In fact, thanks to the greater knowledge about the network topology, the need to activate or reiterate the route discovery mechanism is lower in N-AODV than in AODV. This is acknowledged by looking at Table A.4, concerning the results of the Mann-Whitney test for broadcast activations: the number of times RREQs are broadcasted is always lower in N-AODV and the difference is statistically significant except for the scenario <10 hosts, Low mobility>. Therefore, for BA, H_a is accepted except for this scenario. In the light of these observations, we can answer the first research question stating that N-AODV provides nodes with higher NTA.

Secondly, NTA positively impacts the effectiveness of the protocol: Table A.5, concerning the rate of success, shows that N-AODV has higher rate of success than AODV in all scenarios. We can then answer the second research question stating that N-AODV is more effective than AODV. Moreover, we can observe that, for both protocols, decreasing the mobility in MANETs with the same network size determines an improvement of the rate of success. This is easily explained considering that the lower change of the topology (due to the lower value of node mobility) determines an higher probability to find the proper route in the routing tables of the involved nodes.

The better performance exploited by N-AODV are obtained by the addition of the NACK control packet. Nevertheless, this determines higher traffic over the network. In fact, Table A.6, concerning the results of the Mann-Whitney test for control overhead, shows that the mean number of control packets spread through the network in the case of N-AODV is higher in most scenarios. This is expected because N-AODV makes use of three control packets instead of the two packets used by AODV (RREQs and RREPs). However, the difference between the protocols is not statistically significant in the scenario <25 hosts, Low mobility>. Moreover, the results are surprising when looking at the scenarios with 50 hosts: the mean value for N-AODV is lower than AODV in both cases, even if the differences are not statistically significant. Therefore, for CO, H_a is accepted except for three scenarios. This means that the use of three control packets instead of two has effects on small networks, but in larger networks the behavior of N-AODV with respect to control overhead is not distinguishable from the behavior of AODV. In order to better understand this issue, recall the results regarding broadcast activations: we can argue that, in the long run, the overhead N-AODV injects in the network is counterbalanced by the information carried on by NACKs. In fact, this information results in a lower need to activate the route discovery mechanism or to reiterate it. In other words, there is less need to send control packets through the network. Therefore, we can answer the third research question stating that AODV is more efficient than N-AODV, but when the network size increases the behavior of both protocols is statistically equivalent.

In summary, the results obtained show that N-AODV, thanks to the NACK control packet, really improves NTA in MANETs. This improvement, in turn, determines higher effectiveness in find routes to the desired destinations. Moreover, in larger networks, it counterbalances the increased network overhead, so that the efficiency of N-AODV is not worse than the traditional AODV.

5.2 Analysis of the Threats

Wohlin and others in [47] emphasize that empirical studies are usually affected by threats to the validity of the findings, which can be classified into: *conclusion*, concerning the features of the experiment that can lead to incorrect results; *internal*, related to factors not controlled by the experimenters which can influence the results; *construct*, dealing with the relationship between the theory driving the experiment and what is observed during it; *external*, related to the ability to generalize the results obtained. In addition to these general threats, we also take into account specific limits of MANET simulation studies, recognized as “pitfalls” by several authors (e.g. [26] and [3]). These pitfalls mainly concern: the model validation, in fact the simulation must be validated with respect to a real-world implementation; the precision and abstraction of the simulation, because simulations inherently provide an abstract view of the system under study and are imprecise; repeatability, in the sense that details allowing researchers to easily reproduce the reported experiment must be given; statistical validity, i.e. the data analysis must be based on mathematical rigor. Note that the presence of threats to validity and pitfalls do not mean that the experiment or its findings are not reliable, but only indicate that possible factors cannot be properly considered or that generalization is not always permitted.

With respect to the classification of threats, it is worth noting that our experiment does not present threats to the conclusion validity, nor internal validity. In particular, concerning the conclusion validity, the threats suffered by our previous study [8], i.e. the sequential execution of runs, the freeze of the mobility during each run and the absence of the timeout mechanism have been overcome thanks to the adoption of ad-hoc approaches inside the simulator. Nevertheless, the present experiment suffers from the construct validity. It is threated by the specific values adopted for defining the high/low levels of node mobility: they are confounding factors, in the sense that they make impossible to tell if the results obtained strictly depend on the protocols or are influenced by these factors. In other words, a different choice of the values for node mobility could produce results with different trends. Finally, the present study is affected by threats to the external validity. In fact, the choice of different but limited scenarios can impact the generalization of the findings to all possible scenarios. This issue is an intrinsic limit of simulation studies in that the cardinality of the set of all possible scenarios is uncountable. However, we think that the scenarios we considered are tailored to our purposes.

Concerning the specific pitfalls of MANET simulation studies, our work suffers from two limits. On one hand, the behavior of the simulator is validated with respect to the original specification of each protocol ([34] for AODV and [6] for N-AODV), but is not validated with respect to a real world implementation. This is simply due to the current status of our research: N-AODV has been recently proposed and no real implementations at this moment exist. On the other hand, the precision and abstraction of the simulation is affected by a purely randomized strategy: this approach does not allow us to consider specific MANET applications, for example patterns of movement, possible presence of obstacles, and so on. Although our simulator is not validated with respect to a real world implementation, the findings of our experimental analysis are quantitatively confirmed by several studies reported in literature, e.g. [36] regarding effectiveness, [37] regarding control overhead and [33] regarding broadcast activations. Analogous considerations can be done for N-AODV and the improvement of NTA but only with respect to our previous study [8], because it was the first empirical study aimed at evaluating N-AODV. Concerning NTA and effectiveness, the results of the present work are qualitatively confirmed by what observed in the preliminary study. Moreover, for what concerns efficiency, also in the previous study we noticed that the behavior of N-AODV improves when the network size increases.

Note that in our study the adoption of symmetric transmission ranges is not a pitfall. In the route discovery mechanism the more realistic, asymmetric model causes the following risk. The transmission range of a node *n* can be greater than the transmission range of one of its neighbors *m*. Therefore, if *n* successfully sent an RREQ to *m*, then the RREP or the NACK packet expected by *n* can be not received back. However, since our purpose is to emphasize the differences between the protocols when adding the NACK packet inside the route discovery mechanism, we were not interested into studying the effects of packet loss due to this reason.

6 Conclusion

Network topology awareness is very important for MANETs; however, it is difficult to achieve. In fact, in a MANET, each node can be directly aware only about its neighbors. Instead, for what concerns non-neighboring nodes, it can possibly obtain information indirectly and this information can became obsolete, in short time, because of mobility. In contrast to proactive protocols, in which the information about the topology is constantly updated, in reactive protocols routes to other hosts are established only when required: on one hand, this greatly decreases the traffic overhead; on the other hand, this provides nodes with lower NTA. N-AODV is a variant of the well-known AODV reactive protocol for MANETs which we proposed with the aim of increasing the NTA of the original protocol. The variant is based on the principle that nodes must not necessarily reply a request for a destination only when they “know something”, but also when they “ignore something”. This has been accomplished by introducing the NACK packet.

The proof of correctness of N-AODV has been discussed in [6], so here we have experimentally validated its effectiveness. The study has been executed by simulations and in accordance with the best practices for empirical studies recommended in [47]. Moreover, since the domain of MANETs includes specific issues, their common pitfalls ([26], [3]) have also been discussed. The experiment here reported is based on an exploratory case study we conducted preliminarily in [8]. Table 1 summarizes the features of our two studies.

Table 1. Features of the two studies.

Issue	**Previous study**	**Present study**
Experimental environment	C++ based simulator	CoreASM based simulator
Simulation features	Sequential runs, topology changes only at the end of each run, absence of the timeout	Parallel runs, topology changes at any time, presence of the timeout
Parameters	Network size: 10, 20 and 30 hosts; host connectivity: high and low; number of runs: 300	Network size: 10, 25 and 50 hosts; node mobility: high and low; number of runs: 500
Metrics	Routing tables size, routing tables updates, broadcast activations; rate of success; control overhead	Routing tables size, routing tables updates, network awareness lag, broadcast activations; rate of success; control overhead

Both the exploratory case study and the experiment here reported are a novelty in the research on MANETs in that they apply for the first time the empirical approach on the study of NTA. The findings of both studies show that a MANET adopting N-AODV exploits higher NTA than a MANET adopting AODV. Moreover, this improvement positively impacts the effectiveness of the original protocol. In the long run, a MANET adopting N-AODV benefits from an increased NTA because each node has a greater chance to already know a route to the desired destination, so it has a lower need to initiate or reiterate a route discovery mechanism in order to establish this route.

This paper also contributes by providing for the first time an executable ASM-based model for evaluating the MANETs' behavior. Abstract State Machines are very suited for representing a MANET computation: their expressiveness as well as the intrinsic parallelism of their rules execution allows for the implementation of simulation environments at any desired level of abstraction. This experience, once again, acknowledges the effectiveness of formal approaches to the development and the analysis of critical and complex systems, such as MANETs.

Future work will follow two roads: on one hand, we are interested in improving the confidence of the findings by replicating the experiment here conducted. The replications should face the other remarked threats to validity and pitfalls. For example, more complex mobility models, including patterns of movement, could be taken into account in order to realize simulations related to specific applications. On the other hand, the reactive protocol N-AODV must be compared to some proactive ones. In fact, this class of routing protocols is able to discover periodically all routes, so, ideally, exploiting higher NTA. Nevertheless, we suspect that they inject higher traffic overhead in the network. Precise comparisons of performance greatly help practitioners in choosing the most appropriate protocol.

Acknowledgements

We want to thank Gianluca Gennaro Bevilacqua, for his valuable suggestions which led to improvements of the experimental design, and Pasquale Busco, for his help in developing the simulator.

References

1. Abuelela M. and Olariu S. Taking VANET to the clouds. In: 8th International Conference on Advances in Mobile Computing and Multimedia, 2010, pp. 6-13.
2. Agrawal D.P. and Zeng Q.A. Introduction to Wireless and Mobile Systems. Thomson Brooks/Cole, 2003.
3. Andel T.R. and Yasinsac A. On the credibility of manet simulations. Computer, 39(7), 2006, pp. 48-54.
4. Bhatsangave S.P. and Chirchi V.R. OAODV Routing Algorithm for Improving Energy Efficiency in MANET. International Journal of Computer Applications, 51(21), 2012, pp. 15-22.
5. Bianchi A. and Pizzutilo S. Studying MANET through a Petri Net-Based Model. In: 2nd International Conference of Evolving Internet, 2010, pp. 220-225.
6. Bianchi A., Pizzutilo S. and Vessio G. Preliminary Description of NACK-based Ad-hoc On-demand Distance Vector Routing Protocol for MANETs. In: 9th International Conference on Software Engineering and Applications, 2014, pp. 500-505.
7. Bianchi A., Pizzutilo S. and Vessio G. Suitability of Abstract State Machines for Discussing Mobile Ad-hoc Networks. Global Journal of Advanced Software Engineering, 1, 2014, pp. 29-38.
8. Bianchi A., Pizzutilo S. and Vessio G. Comparing AODV and N-AODV Routing Protocols for Mobile Ad-hoc Networks. In: 13th International Conference on Advances in Mobile Computing and Multimedia, 2015, pp. 159-168.
9. Börger E. and Stärk R. Abstract State Machines: A Method for High-Level System Design and Analysis. Springer-Verlag, 2003.
10. Botía Blaya J.A., Demeure I., Gianrossi P., Garcia Lopez P., Martínez Navarro J.A., Meyer E.M., Pelliccione P. and Taste-Cherel F. POPEYE: providing collaborative services for ad hoc and spontaneous communities. Service Oriented Computing and Applications, 3(1), 2009, pp. 25-45.
11. Boukerche A. and Bononi L. Simulation and Modeling of Wireless, Mobile and Ad Hoc Networks. In Basagni S., Conti M., Giordano S. and Stojmenovic I., eds., Mobile Ad Hoc Networking, 2004, pp. 373-410.
12. Conover W.J. Practical Nonparametric Statistics. John Wiley and Sons, 1980.
13. Das S.R., Castaneda R. and Yan J. Simulation based performance evaluation of mobile ad hoc network routing protocols. ACM/Baltzer Mobile Networks and Applications Journal, 5(3), 2000, pp. 179-189.
14. Demir T. Simulation of Ad Hoc Networks with DSR Protocol. In: 16th International Symposium on Computer and Information Sciences, 2001, pp. 617-625.
15. Farahbod R., Gervasi V. and Glässer U. CoreASM: An extensible ASM execution engine. Fundamenta Informaticae, 77(1-2), 2007, pp. 71-103.
16. Fehnker A., Höfner P., Kamali M. and Mehta V. Topology-based Mobility Models for Wireless Networks. In: 10th International Conference on Quantitative Evaluation of Systems, 2013, pp. 389-404.

17. Georgiou S., Tsakalozos K. and Delis A. Exploiting Network-Topology Awareness for VM Placement in IaaS Clouds. In: 3rd International Conference on Cloud and Green Computing, 2013, pp. 151-158.
18. Gupta A.K., Sadawarti H. and Verma A.K. Performance analysis of AODV, DSR & TORA Routing Protocols. International Journal of Engineering and Technology, 2(2), 2010, pp. 226-231.
19. Gurevich Y. Sequential Abstract State Machines Capture Sequential Algorithms. ACM Transactions on Computational Logic, 1(1), 2000, pp. 77-111.
20. Hsu C.S., Sheu J.P. and Tung S.C. An On-Demand Bandwidth Reservation QoS Routing Protocol for Mobile Ad Hoc Networks. In: IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, 2006, pp. 198-207.
21. Jefferson D.R. Virtual time. ACM Transactions on Programming Languages and Systems, 7(3), 1985, pp. 404-425.
22. Kannhavong B., Nakayama H., Kato N., Jamalipour A. and Nemoto Y. A study of a routing attack in OLSR-based mobile ad hoc network. International Journal of Communication Systems, 20(11), 2007, pp. 1245-1261.
23. Kikuchi H., Karki B.B. and Saini S. Topology-Aware Parallel Molecular Dynamics Simulation Algorithm. In: International Conference on Parallel and Distributed Processing Techniques and Applications, 2006, pp. 1083-1088.
24. Kim C., Talipov E. and Ahn B. A Reverse AODV Routing Protocol in Ad Hoc Mobile Networks. Emerging Directions in Embedded and Ubiquitous Computing, LCNS 4097, 2006, pp. 522-531.
25. Kumar P., Kumar R., Kumar S. and Kumar R.D. Improved Modified Reverse AODV Protocol. International Journal of Computer Applications, 12(4), 2010, pp. 22-26.
26. Kurkowski S., Camp T. and Colagrosso M. MANET Simulation Studies: The Incredibles. ACM SIGMOBILE Mobile Computing and Communications Review, 9(4), 2005, pp. 50-61.
27. Mirrezaei S.I., Shahparian J. and Ghodsi M. A topology-aware load balancing algorithm for P2P systems. In: 4th International Conference on Digital Information Management, 2009, pp. 1-6.
28. Monarch Simulator. http://www.monarch.cs.cmu.edu
29. Nemati A.G. and Takizawa M. Data Transmission Procedures for a Multi-Source Streaming Model in Mobile Peer-to-Peer (P2P) Overlay Networks. J. of M. Multimedia, 5(1), 2009, pp. 45-63.
30. Network Simulator ns-3. https://www.nsnam.org
31. OLSR Web Site. http://www.olsr.org
32. Papadakis H., Roussopoulos M., Fragopoulou P. and Markatos EP. Imbuing unstructured P2P systems with non-intrusive topology awareness. In: 9th International Conference on Peer-to-Peer Computing, 2009, pp. 51-60.
33. Patil V.P. Performance Evaluation of on demand and Table driven Protocol for Wireless Ad hoc Network. International Journal of Computer Engineering Science, 2(9), 2012, pp. 1-13
34. Perkins C.E., Belding-Royer E. and Das S.R. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561, http://tools.ietf.org/html/rfc3561, 2003.
35. Perkins C.E. and Bhagwat P. Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. Computer Communication Review, 24(4), 1994, pp. 234-244.
36. Perkins C.E., Royer E.M., Das S.R. and Marina M.K. Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks. IEEE Personal Communications, 8(1), 2001, pp. 16-28.
37. Qasim N., Said F. and Aghvami H. Mobile Ad Hoc Networking Protocols' Evaluation through Simulation for Quality of Service. Int. Journal of Computer Science, 36(1), 2009, pp. 1-9.
38. Ramanathan R., Hansen R., Basu P., Rosales-Hain R. and Krishnan R. Prioritized epidemic routing for opportunistic networks. In: 1st international MobiSys workshop on Mobile opportunistic networking, 2007, pp. 62-66.

39. Rostami H. and Habibi J. A Mathematical Foundation for Topology Awareness of P2P Overlay Networks. Grid and Cooperative Computing, LNCS 3795, 2005, pp. 906-918.
40. RoyChoudhury R., Paul K. and Bandyopadhyay S. An agent-based protocol to support multimedia communication in ad hoc wireless networks. In: 15th International Parallel and Distributed Processing Symposium, 2000, pp. 2026-2033.
41. Sedrati M., Bilami A. and Benmohamed M. M-AODV: AODV variant to Improve Quality of Service in MANETs. International Journal of Computer Science Issues, 8(1), 2011, pp. 429-436.
42. Shiguo W. and Ji H. A topology-aware peer-to-peer protocol applicable to wireless network. In: International Conference on Network Infrastructure and Digital Content, 2009, pp. 1004-1009.
43. Singh A., Ramakrishnan C.R. and Smolka S.A. A process calculus for Mobile Ad Hoc Networks. Science of Computer Programming, 75(6), 2010, pp. 440-469.
44. Tamilselvan L. and Sankaranarayanan V. Prevention of Co-operative Black Hole Attack in MANET. Journal of Networks, 3(5), 2008, pp. 13-20.
45. Vasudevan S., Kurose J. and Towsley D. Design and analysis of a leader election algorithm for mobile ad hoc networks. In: 12th Int. Conference on Network Protocols, 2004, pp. 350-360.
46. Wang L., Zhang L., Shu Y. and Dong M. Multipath source routing in wireless ad hoc networks. In: Canadian Conference on Electrical and Computer Engineering, 2000, pp. 479-483.
47. Wohlin C., Runeson P., Höst M., Ohlsson M.C., Regnell B. and Wesslén A. Experimentation in Software Engineering. Springer, 2000.
48. Wu H.K. and Chuang P.H. Dynamic QoS Allocation for Multimedia Ad Hoc Wireless Networks. Mobile Networks and Applications, 2001, pp. 377-384.
49. Wu D., Wang R. and Zhen Y. Link stability-aware reliable packet transmitting mechanism in mobile ad hoc network. Int. Journal of Communication Systems, 25(12), 2012, pp. 1568-1584.
50. Xiaoqi L., Lyu M.R. and Jiangchuan L. A trust model based routing protocol for secure ad hoc networks. In: 2004 Aerospace Conference, vol. 2, 2004, pp. 1286-1295.
51. Zapata M.G. Secure Ad-hoc On-Demand Distance Vector Routing. ACM SIGMOBILE Mobile Computing and Communications Review, 6(3), 2002, pp. 106-107.

Appendix A: Results

The results of the Mann-Whitney test applied to the metrics concerning network topology awareness are reported in Tables A.1 to A.4. As well as Table A.6, that concerns control overhead, they show the mean values (columns 2 and 3) and the p-value obtained (column 4); beside each p-value the accepted hypothesis is reported. Instead, Table A.5 concerns the rate of success and simply reports, for each protocol, the percentage of success in each scenario.

Table A.1. Results of the Mann-Whitney test for routing tables size.

Scenario	**AODV(mean)**	**N-AODV (mean)**	**p-value**
10 hosts/High mobility	13.092	21.13	< 0.0001 (accepted H_a)
10 hosts/Low mobility	25.772	28.544	< 0.0001 (accepted H_a)
25 hosts/High mobility	129.948	164.78	< 0.0001 (accepted H_a)
25 hosts/Low mobility	182.272	225.4240	< 0.0001 (accepted H_a)
50 hosts/High mobility	575.566	809.966	< 0.0001 (accepted H_a)
50 hosts/Low mobility	1030.224	1135.414	< 0.0001 (accepted H_a)

Table A.2. Results of the Mann-Whitney test for routing tables updates.

Scenario	**AODV(mean)**	**N-AODV (mean)**	**p-value**
10 hosts/High mobility	0.912	1.236	0.5661 (accepted H_0)
10 hosts/Low mobility	0.52	0.878	0.0016 (accepted H_a)
25 hosts/High mobility	3.404	5.182	0.0106 (accepted H_a)
25 hosts/Low mobility	2.202	3.46	0.0413 (accepted H_a)
50 hosts/High mobility	9.598	13.158	0.5598 (accepted H_0)
50 hosts/Low mobility	6.556	8.056	0.0369 (accepted H_a)

Table A.3. Results of the Mann-Whitney test for network awareness lag.

Scenario	**AODV(mean)**	**N-AODV (mean)**	**p-value**
10 hosts/High mobility	6.103	2.6136	< 0.0001 (accepted H_a)
10 hosts/Low mobility	5.3769	2.8883	< 0.0001 (accepted H_a)
25 hosts/High mobility	3.8867	2.2163	< 0.0001 (accepted H_a)
25 hosts/Low mobility	3.3082	1.5847	< 0.0001 (accepted H_a)
50 hosts/High mobility	2.2764	1.047	< 0.0001 (accepted H_a)
50 hosts/Low mobility	1.5191	1.1705	< 0.0001 (accepted H_a)

Table A.4. Results of the Mann-Whitney test for broadcast activations.

Scenario	**AODV(mean)**	**N-AODV (mean)**	**p-value**
10 hosts/High mobility	1.914	1.446	0.0002 (accepted H_a)
10 hosts/Low mobility	1.104	1.03	0.5848 (accepted H_0)
25 hosts/High mobility	4.24	3.052	0.0022 (accepted H_a)
25 hosts/Low mobility	2.722	2.226	0.0088 (accepted H_a)
50 hosts/High mobility	11.376	5.496	< 0.0001 (accepted H_a)
50 hosts/Low mobility	7.972	3.414	< 0.0001 (accepted H_a)

Table A.5. Rate of success.

Scenario	**AODV (%)**	**N-AODV (%)**
10 hosts/High mobility	61.6	69
10 hosts/Low mobility	66.8	71.2
25 hosts/High mobility	71	76
25 hosts/Low mobility	71.4	82.4
50 hosts/High mobility	78.2	80.6
50 hosts/Low mobility	85.4	90

Table A.6. Results of the Mann-Whitney test for control overhead.

Scenario	**AODV(mean)**	**N-AODV (mean)**	**p-value**
10 hosts/High mobility	0.908	1.544	0.0125 (accepted H_a)
10 hosts/Low mobility	0.634	1.178	< 0.0001 (accepted H_a)
25 hosts/High mobility	3.79	5.332	0.0232 (accepted H_a)
25 hosts/Low mobility	2.61	4.132	0.1055 (accepted H_0)
50 hosts/High mobility	11.15	10.768	0.0898 (accepted H_0)
50 hosts/Low mobility	8.846	7.74	0.4286 (accepted H_0)