International Journal of Wireless Information Networks (2019) 26:48–60
https://doi.org/10.1007/s10776-018-0418-z

Black Hole Attack Prevention in Wireless Peer-to-Peer Networks: A New Strategy

Peter Ndajah[1] · **Abdoul Ousmane Matine**[2] · **Mahouton Norbert Hounkonnou**[3]

Received: 31 August 2017 / Accepted: 14 November 2018 / Published online: 6 December 2018

Abstract
Mobile peer-to-peer networking (MP2P) is a relatively new paradigm compared to other wireless network technologies. In the last 10–15 years, it has gained tremendous popularity because of its usefulness in applications such as file sharing over the Internet in a decentralized manner. Security of mobile P2P networks represents an open research topic and a main challenge regarding the vulnerability of these networks and their convenience to different security attacks such as black hole, Sybil, etc. In this work, we analyze the black hole attack in wireless P2P networks using the AODV as the routing protocol. In a black hole attack, a malicious node assumes the identity of a legitimate node, by creating forged answers with a higher sequence number, and thus forces the victim node to prioritize it as a relay node. We propose a solution based on a modification of the AODV routing protocol, taking into account the behavior of each node participating in the network. The performances of our proposal are evaluated by simulation.

Keywords Wireless network · P2P · Peer-to-peer · Black hole · ns-2

1 Introduction

In recent years, many standards of connectivity and wireless technologies have emerged. These technologies allow users to easily connect a wide range of computing and telecommunications devices without the need to purchase, transport or connect cables. There are a lot of technologies and wireless standards. The most notable of these are Bluetooth, Infrared Data Association (IrDA), HomeRF, and the IEEE 802.11 standards also called WiFi. The advantage of forming a peer-to-peer (P2P) network is to provide wireless communication between heterogeneous devices anytime and anywhere without any pre-established communication infrastructure [1]. However, the biggest problem with these networks is security [2]. The ability to listen, denial of service, and identity theft attacks are easier to achieve in this type of wireless and infrastructureless networks. The routing problem in these networks is complicated. This is mainly due to the nature of Mobile P2P networks which are characterized by a double absence: that of a fixed infrastructure and that of any centralized administration [2, 3].

Peer-to-peer networks generally come in three types. The three variants are unstructured networks, structured networks and hybrid networks. See [4–6]. In these networks, transmitted packets must traverse nodes (routers) before reaching their destination. Initially, the networks were all very small in size and it was therefore possible to manually configure the route for each destination. But soon, the size of the networks increased. It therefore became necessary to automate this task using routing algorithms. However, even though routing algorithms have reached some kind of maturity, the security aspect has not received and equal amount of emphasis. Thus, the question is no longer the search for the optimal route but the search for the most secure path [7]. Some routing protocols concerned with the concept of security are Ad Hoc On Demand Distance Vector (AODV), Dynamic Source Routing (DSR) and Optimized Link State Routing (OLSR) but the protocol that will be the main focus of our study is the AODV protocol.

✉ Peter Ndajah
headofschoolsmathit@ucc.edu.jm; ndajah@gmail.com

[1] University of the Commonwealth Caribbean, 17 Worthington Avenue, Kingston, Jamaica

[2] Institut de Formation et de Recherche en Informatique, Université d'Abomey Calavi, Cotonou, Republic of Benin

[3] International Chair in Mathematical Physiques and Applications, Université d'Abomey Calavi, Cotonou, Republic of Benin

The main purpose of any routing strategy is to implement a robust and efficient routing management. In general, any routing protocol is based on methods and mechanisms that can be grouped into three main classes: proactive routing protocols, reactive routing protocols, and hybrid routing protocols. In view of the difficulty of routing in wireless networks, existing strategies use a variety of techniques to solve this problem [8]. According to these techniques, several classifications have appeared. The classifications described in Table 1 are from the IETF MANET working group [8].

The first criterion used to classify routing protocols concerns the type of vision they have on the network and the roles they assign to the different nodes.

The flat routing protocols assume that all nodes are equal (Fig. 1). Hierarchical routing protocols work by assigning different roles to mobile devices that vary from one another. Some nodes are chosen and assume special functions that lead to a multi-level view of the network topology. An illustration is given in Fig. 2. A classification of routing techniques in common use are source routing and hop-by-hop routing. In source routing, the packet header contains the list of the different relay nodes to the destination. The best known protocol of this class is the DSR protocol. Hop-by-hop routing consists in giving only one packet the address of the next node to the destination. Most routing protocols use this technique e.g. AODV and OLSR [9]. Another classification, inherited from the wired world is the link state and distance vector routing. Both methods require periodic updating of the routing data broadcast by the various network routing nodes. Protocols based on link state are based on information gathered about the state of the links in the network. This information is disseminated periodically in the network. The main routing protocols in ad hoc networks that belong to this class are Temporally-Ordered Routing Algorithm (TORA), OLSR and Topology Dissemination Based On Reverse Path Forwarding (TBRPF). The protocols based on the distance vector are based on an exchange between neighbors of the distance information of the known destinations. The most widely known routing protocols based on the distance vector are: DSR, Destination-Sequenced Distance Vector (DSDV) and AODV [9].

Table 1 The main MANET routing protocols

Protocol	DSR	AODV	OLSR	TBRF
Category	Reactive	Reactive	Proactive	Proactive
Architecture	Flat	Flat	Flat	Hierarchical
Algorithm	Source Routing	Distance Vector	Link Status	Link Status

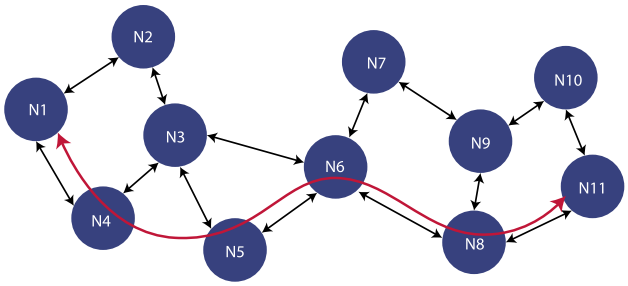

Fig. 1 Flat routing

1.1 How the AODV Protocol Works

The AODV protocol is a reactive routing protocol. This means that routes are built on demand. If a source node wants to send data packets to a destination node, it must establish and maintain a route to that destination node for the time it is using it. The AODV protocol is based on the use of two mechanisms: Route Discovery and Route Maintenance, in addition to node-to-node routing, sequence number principle and exchange. It uses three types of routing packets: Route REQuest (RREQ), Route REPly (RREP) and Route ERRor (RERR). With the AODV protocol, each node must maintain a list of its active neighbors. This list is obtained by a periodic exchange of the HELLO messages of each node with its immediate neighbors. When a source node S wants to send data to a destination D and no route to that destination is stored in the source routing table, node S initiates a route discovery procedure. The source node S sends to its neighbors a route request Route REQest (RREQ) which contains the address of S, the identifier of the request, a sequence counter, the address of D and the number of hop counters with an initial value of zero. The source will wait for a RREP-WAIT-TIMEOUT period, if a response is received then the route discovery operation is terminated, otherwise it rebroadcasts the RREQ and waits for a longer period if no response is received, it will continue the RREQ for a maximum number of RREQ-RRTRIES attempts (03

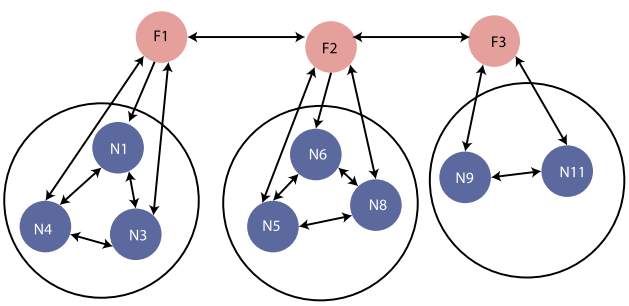

Fig. 2 Hierarchical routing

attempts). If after RREQ-RETRIES route attempts there is no response then the process is aborted and an error message is reported to the application. After a certain waiting period (10 s), the application re-requests the route and consequently the route discovery operation is initiated [10]. Each node that receives the RREQ message looks in its local routing table if there is a route to node D, otherwise the node that handles the RREQ query increments the number of hops and redistributes it again. When the request reaches destination D or a node that knows a route to the destination, a Route REPly (RREP) response is broadcast on the same reverse path (RREQ) route. Figure 3 shows an example of a route search in a network with AODV. Once the source S receives the message, it starts sending data to D [11]. The destination node then generates a RREP packet which makes the inverse path and informs the source node of the path to be taken (Fig. 4).

1.2 The Black Hole Attack

According to [7, 12], malicious behaviors in AODV can be grouped into two categories, namely atomic and compound. The first is the result of handling a single routing message. The second one is defined as a collection of atomic actions. The manipulations performed on a routing message include deleting a packet, modifying one or more fields of the packet before retransmitting it, making up a response to the receipt of a RREQ route request and actively build routing packets without even receiving routing messages. See [9, 13]. Compound Attacks are manifested in the form of regularly repeated elemental attacks and the creation of routing loops (illustrated in Fig. 5). The black hole attack occurs when a node refuses to cooperate in the routing process. Selfish nodes refuse to relay control messages from other nodes that are intended to be broadcast throughout the entire network [14]. In the black hole attack, the intruder systematically blocks all received messages. This type of attack can cause connectivity breakdown across multiple nodes. The black hole attack, according to Maha [15], is a serious problem that needs to be solved in P2P wireless networks. It can occur when a source node initiates a route discovery process

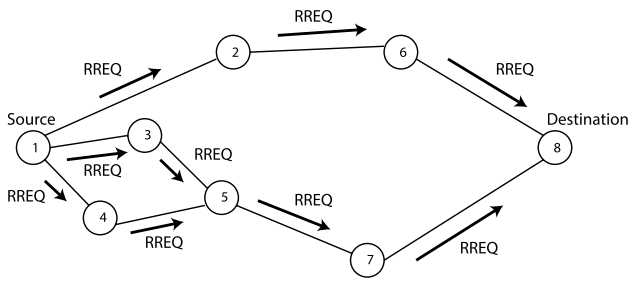

Fig. 3 Route request

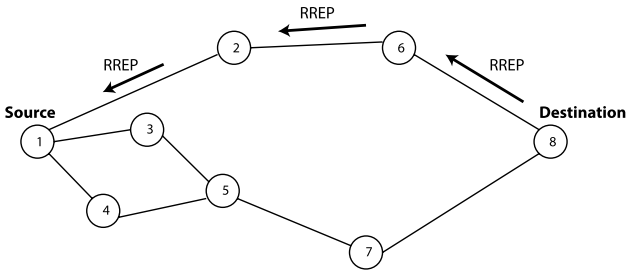

Fig. 4 Route response

by issuing a RREQ packet. The corrupt node, on receiving it will respond with a RREP packet with a sequence number not only erroneous but also high in order to increase its chances of being part of the route. Indeed, if its RREP

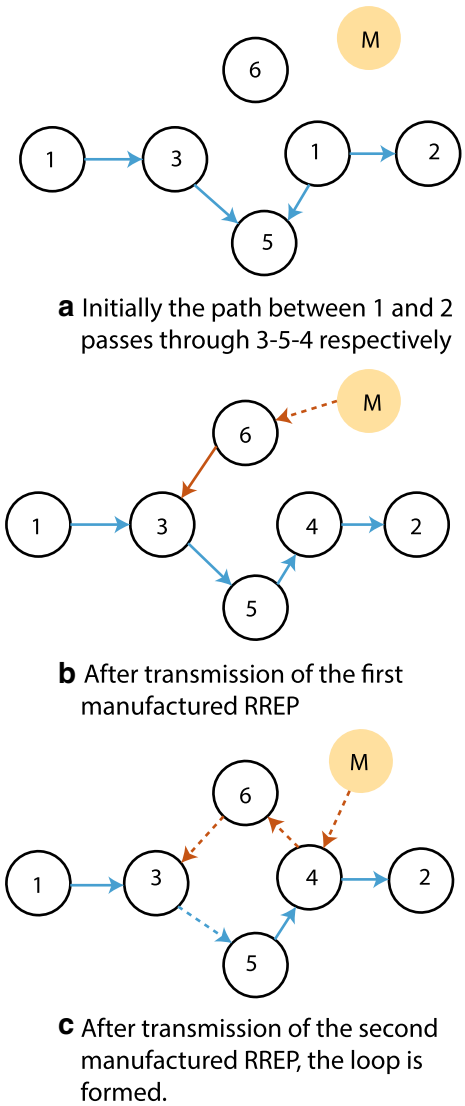

Fig. 5 Creation of a routing loop

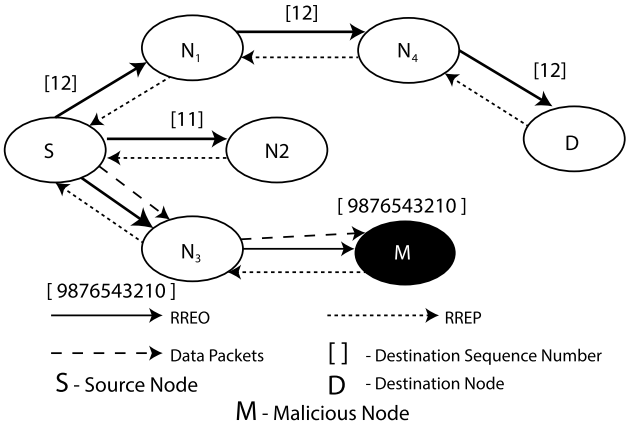

Fig. 6 Black hole attack

packet reaches the source first compared to the responses of legitimate nodes, it can integrate into the route to intercept and control some or all of the traffic exchanged within the network. This attack is severe because the legitimate nodes will update their routing tables with false information and the malicious node does not need to intervene a second time. Figure 6 illustrates this attack where the source S wants to transmit data to the destination D [16].

1.3 Existing Methods Developed Against the Black Hole Attack

Several solutions have been proposed to overcome the security problems in wireless networks. [17] proposed a solution against the black hole attack by modifying the AODV protocol. In this method, each intermediate node must include the next hop information when sending a RREP packet. Once the source has received the RREP packet and before sending the data packets, it retrieves the address of the next hop and sends it a new request for a route (Further Request) in order to verify that this next hop has a route to the intermediate node that sent the response message, and that it also has a route to the destination node. The proposed mechanism is effective in detecting the black hole attack. However, sending a FRREQ packet from the source node to the next hop and waiting for the next-hop FRREP packet increases the load of the overhead routing between the source and the next hop. Al-Shurman et al. [18] proposed two solutions. The first solution proposed is to find more than one route to the destination (at least three different routes). The source sends a RREQ packet to the destination node using these three routes. The destination, the malicious node and the intermediate nodes will respond to this packet. The sending node puts its data packets in a buffer until it receives more than one RREP response. When the source receives RREPs and if the destination routes have shared nodes, the source can recognize a safe path to the destination, and the packets will be transmitted. This solution can guarantee to find a secure route to the destination, but the main disadvantage is the waiting time. The second proposed solution exploits the sequence number included in the header of each packet. The node in this situation needs to have two additional tables: the first table includes the sequence numbers of the last packet sent to each node in the network. And the second table contains the sequence number received from each sender. During the route response phase, the intermediate node or destination must include the sequence number of the last received packet from the source that triggered the route request. Once the source receives this RREP, it will retrieve the last sequence number and then compare it with the value stored in its table. If it matches, the transmission will take place. If it does not, the node is a malicious node. However, both solutions have the end-to-end delay as a disadvantage. According to the solution proposed by [19], the source must wait for other answers with details about the next hop before sending the data packets to the destination. Once a node or the source has received the first RREQ, it sets a timer in the Timer Expired Table to collect new requests from the different nodes and store them in sequential order. The disadvantage of this solution is the end-to-end delay, since the source node must wait for other route responses before sending the data. Himral et al. [20] proposed a method for finding secure routes and preventing malicious nodes by checking for a significant difference between the source node's sequence number and the intermediate node that sent the first RREP. Typically, the first response will then be from the malicious node with a very high destination sequence number. The response will be stored as the first entry in the route reply table (RR-table). Next, it compares the first received destination sequence number with the source node's sequence number, and if there is a large difference between them, it is certain that this response comes from a malicious node, the entry must be removed immediately from the table. This method offers the following advantages: (1) The malicious node is identified in the initial phase and is removed immediately. (2) No changes are made in the other operations of the AODV protocol. (3) Better performance with slight modification. However, the method cannot find multiple malicious nodes. The solution proposed in [21] modifies the behavior of AODV to include a mechanism for verifying the received RREP sequence number. When the source node receives a RREP packet it compares the sequence number of the received RREP with a threshold value. The responding node is suspected to be a black hole if its sequence number is greater than the threshold value. The source node adds the suspected node to its black list, and propagates a control message called an alarm to make the blacklist known to its neighbors. The threshold is the calculated average of the difference between the destination sequence number in the routing table and the destination sequence number in the RREP in a period of time. The main advantage of this protocol is that the source node announces the black hole to its neighbors

so that the black hole is ignored and deleted. An algorithm presented in [22] to detect the black hole attack is based on a preprocessing called Pre-Process-RREP. It is simple and does not change the operation of any of the intermediate or destination nodes. It does not even have to change the normal functioning of the AODV protocol. The process continues to accept RREP packets and calls a function called Compare-Pkts (p1 packet, p2 packet) that compares the destination sequence number of the two packets and selects the packet with a higher destination number if the difference between the two numbers is not substantially high. The packet containing an exceptionally high destination sequence number is suspected of being from a malicious node.

2 Tools and Methodology

To implement our solution, we made use of a DELL brand laptop with the following characteristics:

1. Operating system: Linux Ubuntu version 13.4
2. Hard Drive: 320 GB
3. CPU: Celeron (R) Dual-Core CPU T3000 @ 1.80 GHz
4. Memory: 2 GB of RAM

2.1 Simulation Environment

In order to implement and review the performance of our protocol, we chose the ns-2 simulator. Network Simulator (ns-2) is a discrete object-oriented event simulator written in C++ with an interface that uses the Object Tool Command Language (OTcl). It provides a fairly detailed environment allowing, among other things, simulations of IP, TCP, routing and multicast protocols on both wireline and wireless links. It is mainly built on object-oriented paradigms, code reuse, modularity, and is useful in studying the dynamic nature of communication networks. Through C++ and OTcl, it is possible to model any type of network and to describe the conditions of simulation: the network topology, the type of traffic that circulates, the protocols used, the communications that take place, etc. The C++ language is used to describe the internal functioning of the simulation components while the OTcl language provides a flexible and powerful means of simulation control such as triggering events, configuring the network, collecting statistics, and so on. ns-2 was originally designed to run on both Unix and Linux operating systems, but can also be installed on a Windows system. This is done through the Cygwin emulator which offers a Linux environment under Windows. The ns-2 simulator is supplied as a package that contains all the files necessary for its installation. It is found under the name: `ns-allinone-version.tar.gz`. In our simulations, we used `ns-allinone-2.35.tar.gz`. ns-2 is available for free, includes a large number of models, simulates wireless networks quite well and is an excellent choice for peer-to-peer network simulation [23]. A comparative study of wireless network simulators is presented in Appendix C.

2.2 Simulation Process in ns-2

In the ns-2 simulator, actions are associated with events rather than time. For example, each physical activity on a network is translated into an event that is queued. A processing time is then assigned to each event, which makes it possible to schedule the queue. The simulation script (extension .tcl, see Appendix A) must be entered as a file that ns-2 will use to produce an output file containing the results. The trace file (extension .tr) gives all information about the exchanges that occurred throughout the simulation, the movements, the protocols used, and so on. An excerpt of this file and its explanation are presented in Appendix B and the results are analyzed using the Perl language. The latter is used to extract the fields needed to calculate the evaluation and analysis parameters. External tools such as Excel, GNUPLOT, MATLAB or XGRAPH are then used to plot the analysis curves. In order to visualize a network simulation under ns-2, the Network Animator (NAM) graphical interface is used. NAM is a visualization tool which has two main interests: to represent the topology of a network described with ns-2, and to display temporally the results of an ns-2 execution trace. For example, it is capable of representing TCP or UDP packets, breaking a link between

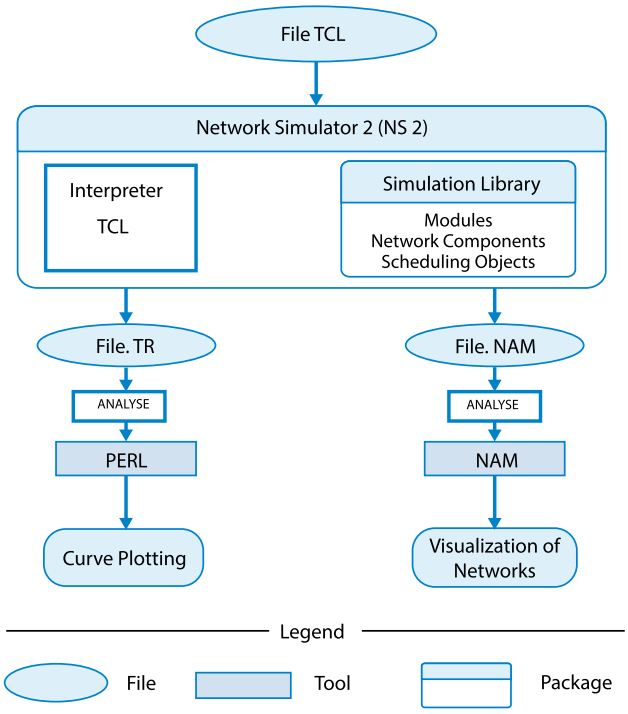

Fig. 7 Simulation process in ns-2

Table 2 Components available in ns-2

Type	Description
Application	Web, FTP, Telnet, Traffic generator (CBR, etc.)
Transport	Static, Dynamic (distance vector)and Multicast (DVMRP, PIM)
Queue management	RED, Drop Tail, Token bucket
Discipline of service	CBQ, SFQ, DRR, Fair Queuing
Transmission system	CSMA/CD, CSMA/CA, Point-to-Point CA

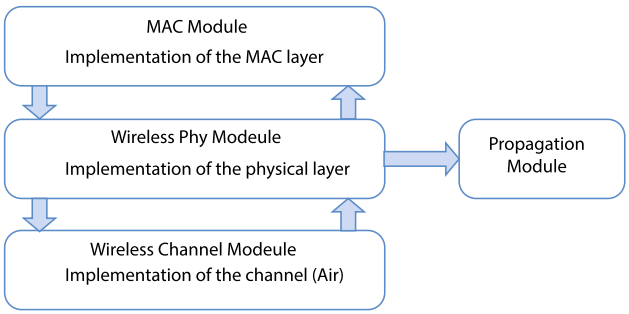

Fig. 8 Implementing IEEE 802.11 in ns-2

nodes, or representing the rejected packets of a full queue. Figure 7 summarizes these explanations.

ns-2 contains libraries for the generation of network topologies such as cable networks, wireless networks, P2P wireless networks, satellite networks, etc. It contains the functionalities necessary for the study of unicast or multicast routing algorithms, transport, session, reservation protocols, embedded services, and application protocols such as File Transfer Protocol (FTP). The components available in ns-2 by category is summarized in Table 2.

2.3 Implementing IEEE 802.11 in ns-2

Implementation of the 802.11 standard in ns-2 includes different modules as shown in Fig. 8. It shows the different exchanges established between the modules. The `WirelessChannel` module simulates the channel (air) for wireless communications. The `WirelessChannel` class inherits from the `Channel` class. It is then used to interconnect all the mobile nodes in a simulation scenario and to exchange frames between a transmitter and receivers in the communication zone. It does not handle interference or collisions. The `WirelessPhy` module simulates the physical layer and inherits the `Phy` class. Its role is to send and receive packets from the communication channel. It is the interface between the modules `WirelessChannel` and `MAC802.11`. The Propagation module is used to determine the power of each received frame that arrives at WirelessPhy. The individual power depends on each propagation model, the distance between the transmitter and the receiver. The MAC 802.11 module includes most of the functions of the IEEE 802.11 standard and inherits from the `MAC` class. It consists of channel access management, the physical header detection function, and the collision management function. The `MAC-802.11` class interacts with the `mac-timers` class, defining the different timers necessary for the proper operation of the access to the channel.

After the simulation runs, ns-2 generates a trace as a text file containing all events of the simulation. The processing of this file makes it possible to extract from it the desired information. Each event is represented in the trace file with a row containing twelve fields. Table 3 gives a view of the structure of a line of the trace file under ns-2.

3 Proposed Protocol and Implementation

Various solutions have been proposed for securing wireless networks against the black hole attack. But these solutions are deficient as we have listed in Sect. 1. The main objective of this section is to present our contribution. We therefore present a description of the proposed protocol, then we discuss the environment used to implement the solution. Finally, we analyze the performance of our protocol with various simulations.

Table 3 Structure of a line of the trace file

1	2	3	4	5	6
Event	Time	From node	To node	Pkt type	Pkt size
7	8	9	10	11	12
Flags	Fid	Src	Dst Addr	Seq Num	Pkt ID

3.1 The Proposed Protocol

The concept of trust has been applied in telecommunications with the notion of prior knowledge of identities. But today, the development of new communication models such as ad hoc networks and P2P wireless networks, make this vision of trust obsolete [24]. In addition, trust is not a technical problem; it is a social problem opposed to the notion of security. We need trust when security is not enough. We propose a new protocol that is able to ensure secure exchanges in P2P wireless networks, while taking into account the characteristics of these networks. In this model, each node in the network maintains an activity table. In the activity table, it stores the identifier of a node, the number of data packets, the number of route request packets (RREQ) and the number of response packets (RREPs) received from this node in order to evaluate confidence. Also, we have a directory containing the public keys of all the nodes of the network so that each packet that will be sent from say node A to node B is signed with a digital signature (used by gmail for a long time) [25]. When a legitimate node receives a packet, it checks whether the packet is signed and increases the number corresponding to the type of packet received in its activity table. If the received packet is of RREP type, it consults its activity table to check one of the equations below. According to the values stored in this table, it decides whether the node is a trusted node or not. Whenever a black hole node receives a data packet, it removes it directly. Thus, when it receives a RREQ packet, it responds by sending a false unsigned RREP without consulting its routing table and it does not rebroadcast the RREQ to the other nodes. Based on this behavior, a legitimate node will not receive any data packets or a RREQ packet from a malicious node. It receives only RREP response packets and therefore, assuming that:

```
NB-D: the number of data packets received from a
node X
NB-RREQ:  the number of RREQ packets received
from a node X NB-RREP: the number of RREP
packets received from a node X
If (NB-D + NB-RREQ > NB-RREP) then: X is a
trusted node
If ((NB-D + NB-RREQ != 0) and (NB-RREP > NB-D +
NB-RREQ) then: X is a known node
If (NB-D + NB-RREQ = 0) then: X is an unknown
node and can be a black hole node
```

In the following, we present the general idea of the protocol:

```
Step 1: The source node S begins the route
discovery phase
Step 2: Each intermediate node receives a RREQ
and stores the source sequence number(SSN)
Step 3: When an intermediate node receives a
RREP, it first checks if the node is signed and
if it exists in the blacklist, if the condition
is true, it directly suppresses it. Otherwise,
it goes to step 4
Step 4: In this step, it checks first if the
packet is signed. If so, it checks a bit added
to the format of the RREP packet, to prevent
several nodes from checking the same packet
several times.
If(the bit = 1) then:
the RREP has already been verified by a node and
the next node will no longer need to re-check
the packet (in this case the node is judged to
be trusted or known)
redirect RREP to the source otherwise  (bit = 0)
switch(status of the node)
case 1: The node is judged to be trusted. Set
the bit = 1
redirect RREP to the source
case 2: the knot is judged known set the bit = 1
redirect RREP to the source
case 3: The node is unknown (unsecured route,
and the node can be a black hole)
if (DSN >> SSN)
it does not return it to the source. Add the
node to the blacklist remove the RREP
if not
set the bit  =  1
redirect RREP to the source
end if
```

3.2 Implementation

In order to implement our protocol, we used the ns-2 simulator [http://www.isi.edu/nsnam/ns/], and we made several changes at several levels. First, we implemented the attack, and then we incorporated the proposed protocol which is a modified version of the AODV protocol. We chose the AODV protocol because it consumes less energy, reduces the routing overhead and is more adaptable to dynamic networks. Preparing the implementation environment involves installing the ns-2 network simulator under the Linux Ubuntu 13.4 operating system. We used version ns-2.35. The installation is carried out as follows:

```
1. cd
2. sudo apt-get update
3. sudo apt-get install build-essential autoconf
automake libxmu-dev git openjdk-7-jre
4. cd /usr/local/
5. sudo git clonegit://github.com/ paultsr/ns-
allinone-2.35.git
6. cd ns-allinone-2.35/
7. sudo ./install
8. close the TERMINAL
```

Type the command `./make` in the command terminal in the ns-2.35 directory in order to compile ns-2 and generate all ns-2 files necessary for its operation. To implement our proposal, we added two protocols in ns-2, namely: BHAODV and SBAODV for Black Hole AODV and Black Hole Solution AODV respectively. In the first, we implemented the black hole attack and in the second we implemented our proposal. In the following, we will discuss the addition of the SBAODV protocol. In [26], the implementation of a new routing protocol in the ns-2 is described. To implement our contribution, we have used the details explained in this paper. All routing protocols in ns-2 are installed in the `ns-2.35` directory. We duplicated the AODV protocol in this directory and changed the directory name to `SBaodv`. All files in the duplicate folder in the directory are renamed for `SBaodv` such as `SBaodv.cc`, `SBaodv.h`, `SBaodv-rqueue.cc`, `SBaodv-rqueue.h`, except for `aodv-packet.h`, in order to use the same data packets of the AODV. Thus all functions, structures and variables are modified. We also modified the `SBaodv.cc` file to introduce our proposed function. After the changes, we modified two common files that are used in ns-2 to integrate the new protocol into the simulator. These changes are explained below:

```
%SBAODV patch
SBAODV{ set ragent [$self create-SBaodv-agent
$node]}
simulator instproc create-SBaodv-agent
{ node} {
set ragent [new agent/SBAODV [$node node-addr]]
$self at 0.0 "$ragent start" # start
BEACON/HELLO Messages
$node set ragent-$ragent
return $ragent
```

We also added the lines below in the Makefile file in the root directory of `ns-2.35`.

```
Sbaodv/Sbaodv_rtable.o SBaodv/SBaodv_
rqueue.oSBaodv/SBaodv_rtable.o
/SBaodv/SBaodv_rqueue.o
```

After all these modifications, we compiled ns-2 again by executing the instructions `make clean`, `make` to create the object files. For the BHAODV, we proceeded in the same way as for the SBAODV. The implementation of the malicious behavior is done on the routing layer by the modification of the AODV protocol as well.

3.3 Network Performance

We evaluated our solution using the following metrics:

- Delivery rate: the ratio between the number of data packets received by the destination nodes and the number generated by the source nodes.
- Normalized routing overhead: This is the ratio between the number of packets of the control (RREQs, RREPs, RERRs) generated by the routing protocol and the number of data packets received.
- Bandwidth: is the amount of information transmitted per unit of time.
- The data packets received by the destinations.

3.4 Simulation Results

After the phase of implementing the solution under ns-2.35, we developed a TCL script [27] allowing us to configure and execute the various simulations. The results of each simulation are saved in a trace (.tr) file specified in the TCL script. We generated a 20-node network and created a UDP connection between the nodes. We used the CBR (Constant Bit Rate) application to generate constant packets through the UDP connection. The size of the CBR packets is chosen to be 512 bytes. Scenario duration is 100 s and CBR connections start at 1.0 s and continue until the end of the simulation, in a space of 500×500. Our simulations are performed using IEEE 802.11 for the MAC layer and Random Waypoint Model [28] as a model of node mobility. In the latter, a mobile node begins by staying in a place for a certain period of time called the pause time. Once this period is complete, the node moves to a randomly selected destination with a selected travel speed in the range [minspeed, maxspeed]. Once this destination is reached, it remains stationary during the specified pause time and then repeats the process. To do this, we used the `/setdest` utility of ns-2, which is the random generator of node motion scenarios. Thus, to generate random traffic patterns, the utility used is `/cbrgen`. To exploit and filter the desired results, we wrote AWK scripts. We then displayed the graphs by the Excel program. Table 4 summarizes the parameters of our experimental model.

Figures 9 and 10 show the simulation of the network with and without the attack after integrating the solution respectively.

Table 4 Simulation parameters

Parameter	Value
Simulator	ns-2.35
Simulation time	100 s
Number of nodes	20
Number of malicious nodes	1
Pause time	2 s
Simulation area	500*500
Traffic	CBR
MAC	802.11

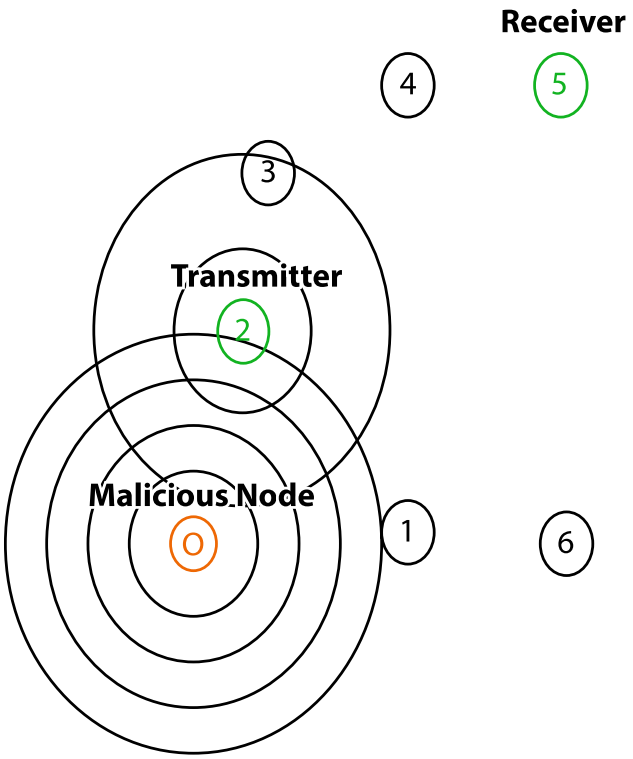

Fig. 9 Simulated black hole attack in AODV

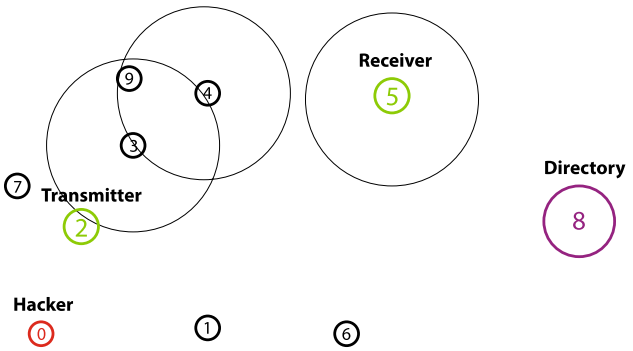

Fig. 10 Simulated black hole attack in SBAODV (malicious node neutralized)

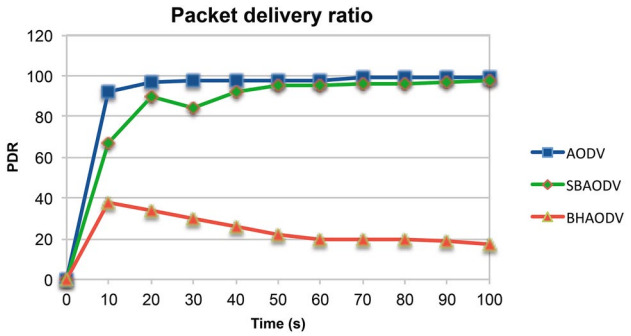

Fig. 11 Packet delivery ratio

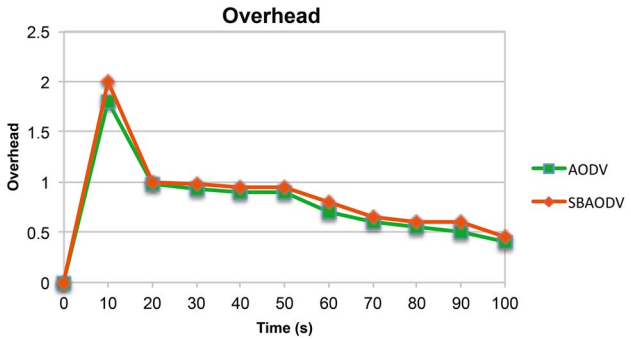

Fig. 12 Network overhead

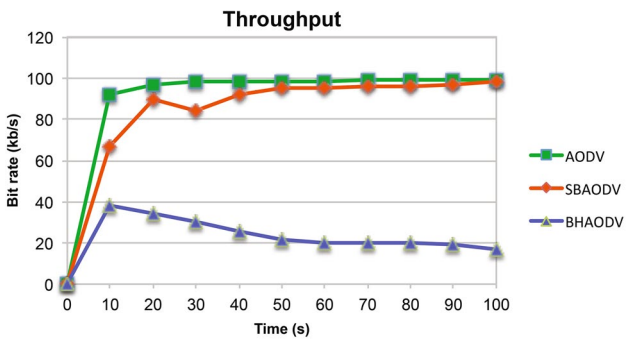

Fig. 13 Throughput against time

Figure 11 shows an increasing evolution of the rate of packet delivery as a function of time, the SBAODV PDR after the suppression of the attack is clearly higher than that of BHAODV, and over time it becomes approximately equal to that of the AODV.

Figure 12 illustrates the routing load as a function of time, it is slightly more than that of the AODV. Figure 13 shows the flow as a function of time, for the BHAODV protocol it is very low, after the integration of our module we notice that the flow begins to increases to match that of the AODV. In

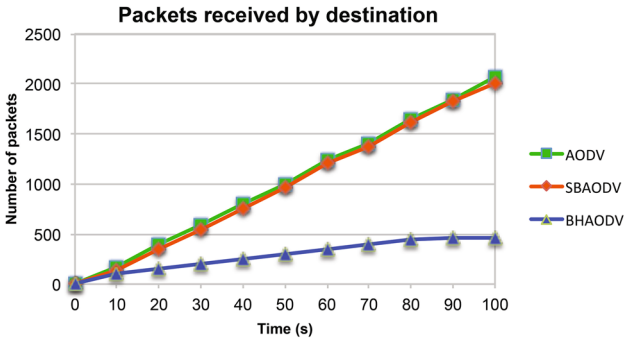

Fig. 14 Packets received by destination

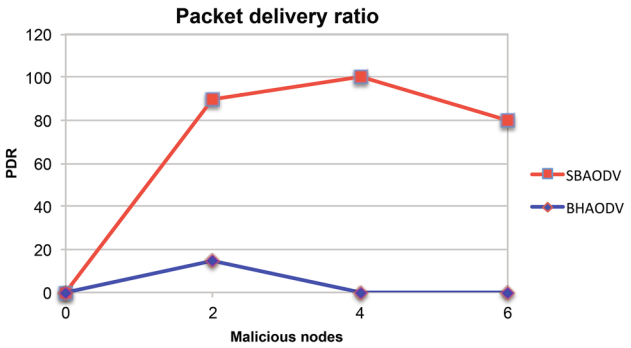

Fig. 15 PDR versus number of malicious nodes

Fig. 14, the number of data packets sent by legitimate nodes and received by their actual destinations is calculated. The curve of the AODV under attack is much lower compared to the other two. Indeed, one sees very well that the attacker succeeded in isolating the legitimate nodes and absorbing the traffic. The packets received in this case are those of the nodes that are far from the malicious node, since 20 nodes were used. If one reduces the number of nodes (to 7 for example), we noticed that the curve of BHAODV becomes 0.

As can be seen in Fig. 15, with the increase in the number of malicious nodes the packet delivery rate for BHAODV tends very rapidly to zero, the PDR for the SBAODV protocol is much higher than that of BHAODV and it begins to decrease when the number of malicious nodes is 4. Figure 16 also shows that the bit rate for the BHAODV protocol rapidly goes to zero by increasing the number of malicious nodes, whereas the SBAODV protocol begins to decrease from 4 malicious nodes. Same thing in Fig. 17, we note that the SBAODV protocol is more efficient in terms of data packets received by the destination nodes compared to BHAODV.

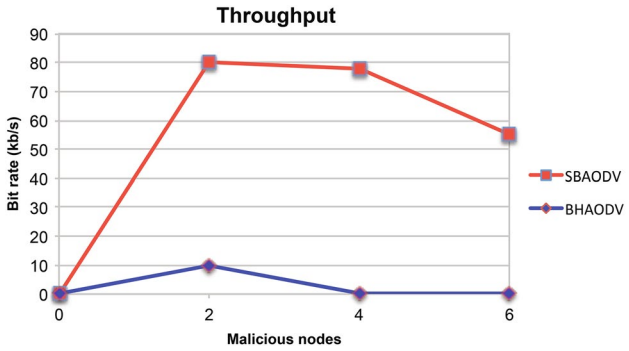

Fig. 16 Throughput versus number of malicious nodes

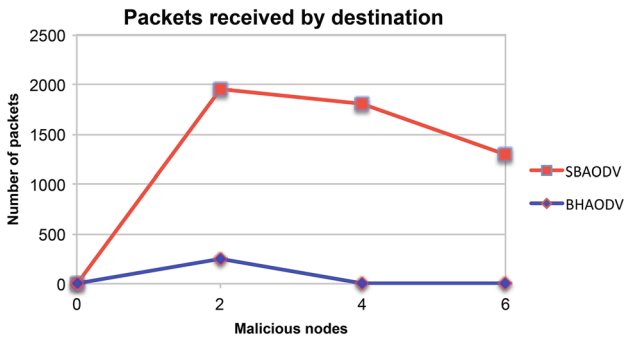

Fig. 17 Received packets versus number of malicious nodes

4 Conclusion

This work has mainly focused on the security of peer-to-peer wireless networks, which is a real challenge because of the characteristics of these networks. In this paper, we proposed a protocol to detect dishonest actions and to secure the exchange of data in these networks. We were interested in security at the routing level, specifically we based our study on the AODV protocol. The particular characteristics of P2P wireless networks make them very vulnerable to several forms of attacks. A specific example of one of these attacks is the black hole attack. This type of attack can represent a major threat to the proper functioning of the network. In this paper, we explored the black hole attack and proposed a new approach to avoid it. The solution exploits the digital signature and the activity of each node in the network to evaluate confidence. In this way, the protocol detects the suspicious behavior associated with the black hole attack. To evaluate the performance of the protocol, we implemented the solution under the ns-2 simulator, performed a set of simulations, presented and interpreted the results obtained.

Appendix A: Simulation Script

```
###  File name: "Application1.tcl"
#### Define options
set val(chan) Channel/WirelessChannel;
# channel type
set val(prop) Propagation/TwoRayGround;
# radio-propagation  model
set val(netif) Phy/WirelessPhy;
# network interface type
set ns_[new Simulator]
# create the nam and trace file:
set tracefd [open Application1.tr w]
$ns_trace-all $tracefd
set namtrace [open Application1.nam  w]
$ns_namtrace-all-wireless $namtrace  $val(x)
$val(y)
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)$
puts "Starting  Simulation........"
$ns_at 25.0 "stop"
$ns_run
```

Appendix B

See Fig. 18.

```
s 0.000000000 _0_ AGT --- 0 cbr 1000 [0 0 0 0] ------- [0:0 5:0 32 0] [0] 0 0
r 0.000000000 _0_ RTR --- 0 cbr 1000 [0 0 0 0] ------- [0:0 5:0 32 0] [0] 0 0
s 0.000000000 _0_ RTR --- 0 AODV 48 [0 0 0 0] ------- [0:255 -1:255 30 0] [0X2 1 1 [5 0] [0 4]] (REQUEST)
s 0.000595000 _0_ MAC --- 0 AODV 106 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0X2 1 1 [5 0] [0 4]] (REQUEST)
r 0.001443737 _9_ MAC --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0X2 1 1 [5 0] [0 4]] (REQUEST)
r 0.001443786 _8_ MAC --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0X2 1 1 [5 0] [0 4]] (REQUEST)
r 0.001443797 _7_ MAC --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0X2 1 1 [5 0] [0 4]] (REQUEST)
r 0.001444047 _4_ MAC --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0X2 1 1 [5 0] [0 4]] (REQUEST)
r 0.001444207 _6_ MAC --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0X2 1 1 [5 0] [0 4]] (REQUEST)
r 0.001444278 _5_ MAC --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0X2 1 1 [5 0] [0 4]] (REQUEST)
r 0.001444330 _3_ MAC --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0X2 1 1 [5 0] [0 4]] (REQUEST)
r 0.001444591 _2_ MAC --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0X2 1 1 [5 0] [0 4]] (REQUEST)
r 0.001446110 _1_ MAC --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0X2 1 1 [5 0] [0 4]] (REQUEST)
r 0.001468737 _9_ RTR --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0X2 1 1 [5 0] [0 4]] (REQUEST)
r 0.001468786 _8_ RTR --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0X2 1 1 [5 0] [0 4]] (REQUEST)
r 0.001468797 _7_ RTR --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0X2 1 1 [5 0] [0 4]] (REQUEST)
r 0.001469047 _4_ RTR --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0X2 1 1 [5 0] [0 4]] (REQUEST)
r 0.001469207 _6_ RTR --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0X2 1 1 [5 0] [0 4]] (REQUEST)
r 0.001469278 _5_ RTR --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0X2 1 1 [5 0] [0 4]] (REQUEST)
r 0.001469278 _5_ RTR --- 0 AODV 44 [0 0 0 0] ------- [5:255 0:255 30 0] [0X4 1 [5 4] 10.000000] (REPLY)
r 0.001469330 _3_ RTR --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0X2 1 1 [5 0] [0 4]] (REQUEST)
r 0.001469591 _2_ RTR --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0X2 1 1 [5 0] [0 4]] (REQUEST)
r 0.001471110 _1_ RTR --- 0 AODV 48 [0 ffffffff 0 800] ------- [0:255 -1:255 30 0] [0X2 1 1 [5 0] [0 4]] (REQUEST)
s 0.001624278 _5_ MAC --- 0 ARP 86 [0 ffffffff 5 806] ------- [REQUEST 5/5 0/0]
s 0.001689494 _4_ RTR --- 0 AODV 48 [0 ffffffff 0 800] ------- [4:255 -1:255 29 0] [0X2 1 1 [5 0] [0 4]] (REQUEST)
s 0.002042524 _1_ RTR --- 0 AODV 48 [0 ffffffff 0 800] ------- [1:255 -1:255 29 0] [0X2 1 1 [5 0] [0 4]] (REQUEST)
r 0.002312742 _6_ MAC --- 0 ARP 28 [0 ffffffff 5 806] ------- [REQUEST 5/5 0/0]
r 0.002312774 _7_ MAC --- 0 ARP 28 [0 ffffffff 5 806] ------- [REQUEST 5/5 0/0]
r 0.002312806 _4_ MAC --- 0 ARP 28 [0 ffffffff 5 806] ------- [REQUEST 5/5 0/0]
r 0.002312838 _2_ MAC --- 0 ARP 28 [0 ffffffff 5 806] ------- [REQUEST 5/5 0/0]
r 0.002313033 _8_ MAC --- 0 ARP 28 [0 ffffffff 5 806] ------- [REQUEST 5/5 0/0]
r 0.002313155 _3_ MAC --- 0 ARP 28 [0 ffffffff 5 806] ------- [REQUEST 5/5 0/0]
r 0.002313303 _9_ MAC --- 0 ARP 28 [0 ffffffff 5 806] ------- [REQUEST 5/5 0/0]
```

Fig. 18 Simulation trace file

Appendix C

See Table 5.

Table 5 Wireless network simulation

Simulators	NS2	NS3	OMNET++	NCTUns	GloMoSim	J-Sim	JIST/SWANS
License	Free	Free	Free	Free	Free	Non-commercial	Non-commercial
Graphical interface	No	No	Yes	Yes	Limited	–	–
Language	C++/OTcl	C++/Python	C++/NED	C	C	Simula	Java
Available modules	Wired networks, wireless, ad hoc, sensors	Wired networks, wireless, ad hoc	Wired networks, wireless, ad hoc	Wired networks, wireless, ad hoc	Wired networks, wireless, ad hoc	Wired networks, wireless, ad hoc	Wired networks, wireless, ad hoc
Documentation and support extension	Excellent	Poor	Average	Good	Average	Good	Good
Extension	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Simulation technique	Discrete event simulator	Discrete event simulator	Discrete event simulator	Discrete event simulator	Discrete event simulator	Discrete event simulator	Discrete event simulator

References

1. Y. Challal, *Sécurité de l'Internet des Objets: Vers Une Approche Cognitive et Systémique des Réseaux et Télécommunications*. Université de Technologie de Compiègne, 2012.
2. B. Wu, et al., *A Survey on Attacks and Countermeasures in Mobile Ad Hoc Networks*, Department of Computer Science and Engineering, Florida Atlantic University, SpringerBerlin, 2006.
3. D. Alexandre and E. Samuel, *Peet-to-Peer*, Master Professionnel, Système Informatique et Réseaux, Université Claude Bernard, Lyon, 2007.
4. A. Benoit, Algorithmique des Réseaux et des Télécoms, *Chapitre 2: Réseaux Pair-à-Pair*. ENS Lyon, 2006.
5. G. Pujolle, *Les Réseaux*, EYROLLESParis, 2008.
6. R. Al King, *Localisation de Sources de Données et Optimisation de Requêtes Réparties en Environnement Pair-à-Pair*. Thèse de Doctorat, Université de Toulouse, 2010.
7. Y. Huang and W. Lee, Attack analysis and detection for ad hoc routing protocols. In *Proceedings of 7th International Symposium on Recent Advances in Intrusion Detection (RAID)*. Springer, 2004.
8. A. Esnault, *Systèmes Pair-à-Pair pour l'Informatique Opportuniste*. Réseaux et Télécommunications, Université de Bretagne Sud, 2017.
9. A. Hajami, *Sécurité du Routage dans les Réseaux sans Fil Spontanés*. Thèse de Doctorat, Université Mohammed V Souissi, Maroc, 2011.
10. N. Tabbane, S. Tabbane and A. Mehaoua, *Simulation et Mésure des Performances du Protocole de Routage AODV*. JTEA 2004, Hamamet, Tunisie, 2004.
11. Abdellaoui Rachid and Jean-Marc Robert, *SUOLSR: A New Solution to Thwart Attacks Against the OLSR Protocol*, Ecole de Technologie SupérieureMontreal, 2009.
12. P. Ning and K. Sun, How to misuse AODV: A case study of insider attacks against mobile ad hoc routing protocols. In *Proceedings of IEEE Systems, Man and Cybernetics Society, Information Assurance Workshop (IAW'03)*. IEEE, June 2003.
13. M. A. Ayachi, *Contributions à la Détection des Comportements Malhonnêtes dans les Réseaux Ad Hoc AODV par Analyse de la Confiance Implicite*. Thèse de Doctorat, Université de Rennes 1, 2011.
14. D. Raffo, *Security Schemes for the OLSR Protocol for Ad Hoc Networks*. Ph.D. Thesis, Université de Paris, 2005.
15. Maha Abdelhaq, et al., Security routing mechanism for black hole attack over AODV MANET routing protocol, *Australian Journal of Basic and Applied Sciences*, Vol. 5, No. 10, pp. 1137–1145, 2011.
16. K. Lakshmi, et al., Modified AODV protocol against black hole attacks in MANET, *International Journal of Engineering and Technology*, Vol. 2, No. 6, pp. 444–449, 2010.
17. H. Deng, W. Li and D. P. Agrawal, Routing security in wireless ad hoc networks. *Communications Magazine, IEEE,* October 2002.
18. M. Al-Shurman, S. Yoo, and S. Park, Black hole attack in mobile ad hoc networks. In *ACM Southeast Regional Conference*, 2004.
19. L. Tamilselvan and V. Sankaranarayanan, Prevention of black hole attack in MANET. In *The 2nd International Conference on Wireless Broadband and Ultra Wideband Communications, Aus Wireless*, 2007.
20. L. Himral, V. Vig, and N. Chand, Preventing AODV routing protocol from black hole attack, *International Journal of Engineering Science and Technology (IJEST)*, Vol.3, No. 5, pp. 3927–3932, 2011.
21. P. N. Raj and P. B. Swadas, DPRAODV: A dynamic learning system against black hole attack in AODV-based MANET, *IJCSI International Journal of Computer Science Issues*, Vol. 2, pp. 54–59, 2009.
22. S. C. Mandhata and S. N. Patro, A counter measure to black hole attack on AODV-based mobile ad-hoc networks, *International Journal of Computer and Communication Technology (IJCCT)*, Vol. 2, No. VI, pp. 37–42, 2011.
23. S. V. Mallapur and S. R. Patil, Survey on simulation tools for mobile ad-hoc networks, *International Journal of Computer Networks and Wireless Communication*, Vol. 2, No. 2, pp. 241–248, 2012.

24. V. Legrand and S. Ubda, *Vers un Modèle de Confiance pour les Objets Communicants: Une Approche Sociale*, Laboratoire CITI INRIA ARES, 2004.
25. N. Kalia and H. Sharma, *Detection of multiple black hole nodes attack in MANET by modifying the AODV protocol*, Lovely Professional University PhagwaraPunjab, 2016.
26. F. J. Ros and P. M. Ruiz, *Implementing a New Manet Unicast Routing Protocol in ns-2*, Dept. of Information and Communication Engineering, University of MurciaMurcia, 2004.
27. K. Fall and K. Varadhan, *The ns Manual*, University of CaliforniaBerkeley, 2011.
28. V. Vasanthi et al, A detailed study of mobility models in wireless sensor networks, *Journal of Theoretical and Applied Information Technology*, Vol. 33, No. 1, pp. 7–14, 2011.

Peter Ndajah received a Ph.D. in Information Science and Engineering from Niigata University, Japan in 2011. He is currently Head of School of Technology and Mathematics at the University of the Commonwealth Caribbean, Kingston, Jamaica. Dr. Ndajah has worked extensively in the areas of mathematics, physics and computer science and has a wide academic and industrial experience. He is also the recipient of several international scholarships in the course of his studies. These include scholarships from the International Centre for Theoretical Physics (ICTP) and the Mombukagakusho Japanese Government Scholarship (Monkasho). His research interests currently range from wireless communication and security, robotics, image and video processing, variational calculus to nonlinear functional analysis.

Abdoul Ousmane Matine is currently a Ph.D. student in Computer Science in the Engineering Science Faculty at the University of Abomey-Calavi (UAC). In 2009, he obtained his Bachelor's Degree in Computer Science from the National School of Applied Economics and Management (ENEAM-UAC), Cotonou. He received his Masters from the Institute for Training and Research in Computer Science (IFRI-UAC). He has worked on the security of networks and he continues his research in network security; Routing protocols; affective computing; modelling and emotion recognition, human–machine interactions, physiological signals analysis, image processing, robotics and contactless biometrics.

Mahouton Norbert Hounkonnou is a full Professor of Mathematics and Physics at the University of Abomey-Calavi, Cotonou, Benin. His research deals with electromagnetic theory, noncommutative and nonlinear mathematics including differential equations, operator theory, coherent states, quantization techniques, orthogonal polynomials, special functions, graph theory, nonassociative algebras, nonlinear systems, noncommutative field theories and geometric methods in physics. He is a visiting professor at several African, Asian, European, and North American Universities. Together with his peers at the international level, he founded the International Chair in Mathematical Physics and Applications (ICMPA—UNESCO Chair) of the University of Abomey-Calavi in which he created a multi-university masters degree and PhD programs in mathematics with connections, motivations, or applications to physics, or in physics with important relationships to mathematics. Professor Hounkonnou is the current President of the Benin National Academy of Sciences, Arts and Letters. His membership extends to the International Association of Mathematical Physics, American Mathematical Society, African Academy of Sciences (AAS), The World Academy of Sciences (TWAS), UNESCO Scientific Board for International Basic Sciences Programme (IBSP), and to many others.