Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für
Technische Informatik und
Kommunikationsnetze

Resilience of wireless mesh networks to node misbehaviors

Semester Thesis

November 2009-March 2010

Author: Georgios Nomikos

Professors: Bernhard Plattner, Maria Papadopouli

Advisors: Dr. Merkourios Karaliopoulos, Gabriel Popa

Contents

Abstract

Wireless mesh networks (WMN) are a special type of mobile wireless ad-hoc networks. In WMNs each node can act as router and therefore can relay packets to its neighbors. Through the relaying process, packets find their way to their destination over a number of single hops through intermediate nodes. Mesh networks feature attractive properties: cost-effective deployment, dynamical self-configuration and self-healing. On the other hand, due exactly to their distributed and highly self-organizing properties significant challenges and threats to their normal operation, may arise due to node misbehaviors. The two main types of these misbehaviors are malice and selfishness. Results from literature suggest that they can easily cancel the operation of this kind of networks and their treatment is anything but trivial.

In this project, we try to look closer into the effects of these misbehaviors, only from multiple points of view. We are particularly concerned with the interplay of selfishness with the radio interference that is one of the main inherent capacity-limitation factors of these networks. Our approach is experimental in that we examine and analyze the reaction of these multihop systems through various sets of simulation runs. In all cases, we use a diversity of configurations for each experiment in order to assess the impact of the position(s), popularity and misbehavior intensity on the resilience of these networks.

The ultimate goal of this effort is to identify and simultaneously quantify the interactions of the misbehaved nodes. Furthermore, the study underlines the need for revisiting proposed mechanisms in literature aiming at coping with the misbehaved nodes, to cater for the interference factor as well.

1. Introduction

Nowadays, there is a lot of interest in having reliable communications. One of the most important aspects of that is improving the wireless networks. Wireless networks have increased the efficiency of fast communications, optimizing the process of sending and receiving huge bunches of data. Also, the wireless technology aims at increasing the possibility of the users to retain their connectivity without using any special external equipment. They are more flexible with high mobility, so that users are not restricted to highly importable PCs. Most of the times, the wireless networks are a cheap solution, easy to deploy and maintain.

In general, wireless networks are classified as single-hop and multi-hop networks. In a single-hop network, the client connects to the fixed base station or access point directly in one hop. Some well-known examples of single-hop wireless networks are WLANs and cellular networks. However, in a multi-hop wireless network, the source and destination nodes communicate in a multi-hop fashion. The well-known forms of multi-hop networks are ad hoc networks, sensor networks, and WMNs.

In this project, we are interested in the wireless mesh networks (WMNs). A WMN is a promising decentralized and simplified infrastructure, which offers wireless access for several emerging and commercially interesting applications at a much lower cost than classic Wi-Fi networks , e.g., broadband home networking, community and neighborhood networks, coordinated network management and intelligent transportation systems.

These wireless mesh multihop systems consist of mesh-mobile clients (e.g. laptops, palmtops), mesh routers and gateways, as shown in Figure 1.1, in order to cover a predefined geographic area where wireless access is desired. In a mesh network, each node can assist each other in transmitting packets through the network, behaving in this way as routers. In this case, there is no need for a base station or any other dedicated infrastructure. The relaying process depends on the fact that every participant has to take over only the transmission as far as the next neighboring node, according to the path to the destination.

Most of the times, the wireless mesh network nodes are small sized objects with built in replaceable batteries and high portability (such as laptops). In addition, these devices suffer from limited storage and computing power. Unfortunately, the technological progress on batteries is much slower than on electronics. When a node is mobile, it either communicates, moves, and then communicates again, or it communicates while moving. Hence, the necessary computational operations of the mobile stations are delayed.

As a result, in order for such a networking mechanism to be applicable in this type of networks, it must be simple, energy efficient, scalable and robust. The limitations on power consumption imposed by portable wireless radios, coupled with the fact that the communication infrastructure does not rely on the assistance of centralized stations, implies that terminals must communicate with each other either directly or indirectly using multihop routing techniques. Consequently, the implementation of traditional networking protocols such as routing and medium access control (MAC), without taking into account energy consumption, scalability, and fault tolerance, is not practical in these networks.

It's worth mentioning that the main difference between wireless mesh networks and mobile ad-hoc networks relies on the traffic pattern. In WMNs all the traffic is usually to or from a gateway, while in ad hoc networks, the traffic flows between arbitrary pairs of nodes [16].

With respect to the routing process, there are numerous routing protocols proposed for ad hoc networks. They are classified according to the information that the nodes maintain about their neighbors and the way that they acquire the appropriate routing information. So, there are topology-based routing protocols and position-based routing protocols. Topology-routing protocols consist of proactive and reactive protocols.

In general, topology-based protocols use routing tables or distribute link-state information. Routing tables are generally used by proactive protocols. It is a static way to maintain routes and metrics associated with those routes, to particular network destinations. The link-state information determines the status and connection type of each link and produces a calculated metric based on these factors in order to detect the most effective path to the destination. In proactive routing protocols, each node retains consistent up-to-date routing information for the rest of the nodes in the topology, no matter whether they have or not to send data. There exist two types of proactive protocols, the link state and the distance vector protocols. In link state, each node periodically lets the rest of the nodes know about the state of his links, flooding the network with messages containing link-state information. Hence, every node has the overall view of the entire network. In contrast, in distance vector protocols the nodes execute a distributed shortest path algorithm to determine the best route to every other node in the network. Furthermore, every node discovers its neighbors propagating their routing tables to the nodes that are in its range. By repeating the routing table exchange and routing table up-date steps, the system converges to a stable state, where each routing table contains correct routing information. Sometimes, proactive protocols are not so effective, because there are situations where even if a small number of nodes communicate all the existing nodes exchange routing packets to update their useless paths. Consequently, proactive protocols are not the best choice when there are a lot of nodes with high mobility, because they increase the routing overhead in very high levels. Hence, the throughput is very low. On the other hand, in the case of reactive or on demand routing protocols, the nodes do not maintain any information if that is not necessary. They take on setting up a path to their destination, only when they are really needed. In this way, they manage to avoid unnecessary wasting of resources. These protocols are a good solution when there is a topology with thousand of mobile nodes.

In position-based routing protocols, the nodes, in order to establish a route, use their own physical locations and the locations of the destinations. They use a mechanism, known as "location service", to exchange location information. Lying on this information, both source and intermediate nodes make the appropriate routing decisions to send the data [6].Because WMNs are multi-hop networks, the protocols designed for ad hoc networks also work well for WMNs. The main goal of those protocols is their quick adaptation to the change in a path when there is a path break due to mobility of the nodes. Current deployments of WMNs make use of routing protocols proposed for ad hoc networks such as AODV (Ad hoc On-Demand Distance Vector) and DSR (Dynamic Source Routing). It is worth mentioning that in WMNs the mesh routers have minimal mobility power and therefore

there are not strict power constraints, whereas the clients are mobile with limited power. For this purpose, it is crucial to choose efficient routing protocols. A wireless mesh configuration can be either hierarchical, flat or hybrid. In a hierarchical network, nodes are partitioned into groups, often called clusters. The dominant kinds of nodes in a cluster are the cluster-head node, the gateway (GW) node, and the cluster-member (internal) node. Cluster-head (CH) nodes basically emulate the functionalities of an AP. In a flat architecture there is no grouping and all nodes have equal responsibilities. Connections are established between nodes that are close enough to each other, in order to allow sufficient radio propagation conditions and establish connectivity.

On the other hand, the hybrid architecture is the combination of flat infrastructure and client meshing. In particular, mesh clients can access the network through mesh routers as well as through meshing with other mesh clients.

Generally, we focus on a distributed kind of network with the ability of self-configuration, self-healing and self-discovering which automatically establish and maintain network connectivity. Mobile devices can freely and dynamically self-organize in arbitrary and temporary network topologies. A wireless mesh network may be left unattended after deployment for days, months or even years in areas without any preexisting communication infrastructure. There is no need for a system administrator to organize how a message can be transmitted towards its destination, and it does not require manual configuration. Ideally, the network automatically finds the fastest and most reliable paths to send the data, even if some nodes are blocked or lose their signal. Hence, these technologies are an effective way to avoid and simultaneously confront a number of central points of failure and control.

Finally, taking into consideration these features of the wireless mesh networks, we can infer some significant benefits and characteristics for the end-to-end users. For instance, low up-front cost, easy network maintenance, robustness and reliable service coverage. [2], [3]

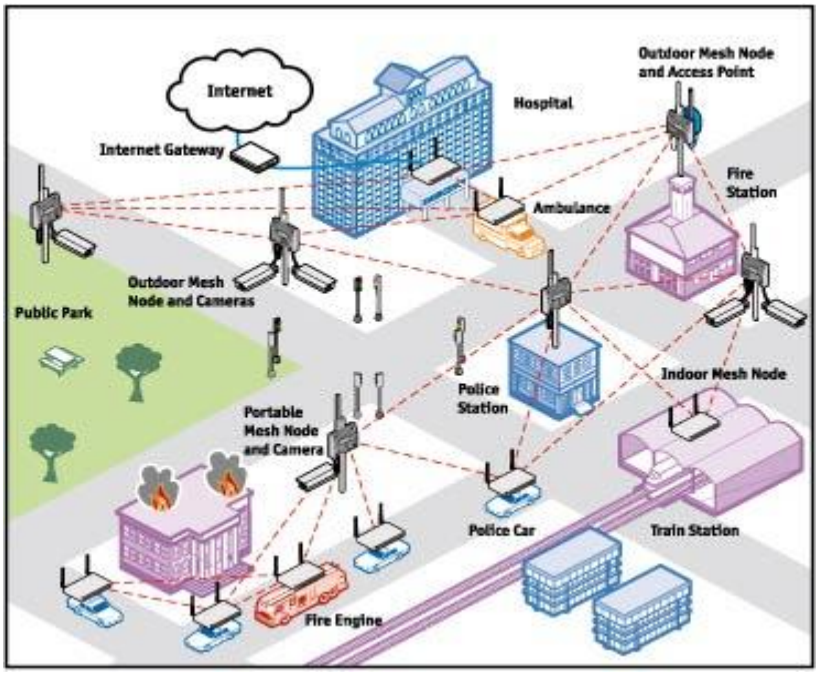

Figure 1.1 In a wireless mesh network, multiple nodes cooperate to relay a message to its destination. Obviously, all the nodes can communicate with every other node (http://www.shanix.com/security/wifi.html).

Wireless mesh networks are not yet ready for wide-scale deployment for two main reasons. The first one, as we said in the previous session, WMNs present severe capacity and delay constraints. The second reason is that the communications are wireless and therefore prone to interference. They do not have adequate security guarantees to succeed regional extension [5]. Due to interference, wireless mesh networks are highly vulnerable to security attacks. The attackers are capable of interfering from outside or inside. An outsider adversary can attack the communication of some nodes with the usage of wireless channels, while an insider targets at taking the control of some nodes within the network. Misbehaviors denote deviation from the full cooperative behavior, where nodes obey the nominal protocol operation and collaborate smoothly in transferring data in the network. There are two types of misbehaviors, malice and selfishness, reflecting different motivation for non–cooperative behavior.

Malicious misbehavior denotes that someone aims at disrupting the network or destroying it, in an extreme case. For instance, somebody can try to obtain information about or from another user of the network using illegitimate means and threatening his privacy in this way. A malicious node may want to overload the buffers of the nodes or provoke collisions when another node makes efforts to have a successful transmission of packets. Generally, they point to increase the resource consumption and therefore degrade the quality of the service provided by the network. The most frequent expressions of malice are eavesdropping, jamming, spoofing and denial of service attacks, replaying control packets. Consequently, there are quiet methods to incommode the communication between sender and receiver as well.

Selfish nodes, on the other hand, do not aim at hurting the overall network operation but rather try to maximize their own benefit, almost always at the expense of reduced performance for other users exploiting in this way common resources (radio spectrum). For instance, selfish nodes may be unwilling to spend battery life, CPU cycles, or available network bandwidth to forward packets not of direct interest to them, even though they expect others to forward packets on their behalf.

Both types of misbehaviors can reduce dramatically the performance of the wireless network and put the privacy and rights of individual users in danger. Unfortunately, it is too difficult to deploy effective security mechanisms to be protected from the attackers. They continually increase using cleverer and faster tools.

Hence, the aim of this Thesis is to experimentally assess the impact of node misbehaviors in WMNs and examine whether we can finally take advantage of these attitudes. Particularly, a lot of researchers have concentrated on analyzing the performance of different wireless network topologies under the presence of selfish or malice nodes. They have proposed protocols [10] or mechanisms ([12], [13], [14], [6]), some of them based on reputation systems, in order to detect and afterwards to confront the misbehaved nodes. However, they have not taken into consideration the case when the operation of a well-behaved node is forced to be similar with a malice or selfish node because of the interference of the adjacent nodes. For instance, due to a high a level of interference a node may suffer from successive collisions. This makes the nodes to drop packets until the medium becomes available again, adopting in some way a packet forwarding misbehavior. Thus, this aspect stimulated us to study the interplay of the misbehaved nodes with the interference of all the nodes within the network. We are going to quantify the capacity of the network when both misbehaved and well-behaved nodes coexist.

In the next sessions we demonstrate how some metrics are affected under a kind of misbehavior in different conditions. More precisely, section 2 presents a group of implementing scenarios using the ns2 simulator trying to make malice/selfish conditions, in section 3 we show and analyze the interaction between healthy and misbehaved nodes in network topologies and therefore the resilience of the wireless mesh nodes to some dangerous situations. [4], [5], [6]

2. Background: 802.11 protocol

One of the most popular wireless technology with which wireless mesh networks can be implemented is the 802.11 protocol. In wireless networks it is very significant to control which node can have access to the medium every time it wants to transmit a number of packets. Moreover, when data are exchanged between the sources and the destinations it is too possible to have collisions. For this purpose the 802.11 protocol uses the CSMA/CA, a medium sensing and collision avoidance mechanism. In brief, 802.11 uses a four-way RTS/CTS/DATA/ACK exchange in order to provide an optional collision reduction scheme with hidden stations. When a station wishes to send data packets, it sends an RTS (request to send) packet to the destination before transmitting data frames. The destination senses the medium, and if it is idle, it responds with a CTS (clear to send). Afterwards, the source sends the data, and waits for an ACK (acknowledgment) from the

destination. If a node overhears RTS or CTS frames, then the medium will be unavailable for a certain time, and avoids initiating new transmissions.

However, it is useful for the rest of nodes to know how much time they have to wait until they can contend for the medium again. For this job, there is the "network allocation vector" (NAV), which indicates the remaining time till the network will become available. It's worth mentioning that 802.11 RTS and CTS packets include the amount of time the medium will be busy. [1]

Another issue that provokes problems to the transmission of the nodes, is the phenomenon of interference. More precisely, interference can distort and disrupt any signal that travels from the source to the destination. There are two kinds of interference, the intra-flow interference and the inter-flow interference, as shown in the figure 2.1. The intra-flow interference obstructs the simultaneous communication of all the nodes in a traffic flow under a chain. We consider the network shown in the plot 2.1 where the node 1(source) originate a flow of data to the destination node 6. Due to the intra flow interference range, the transmission of the node 4 does not permit the simultaneous transmission of the node 1 to the node 2. For instance, node's 1 RTS transmission will interfere with the node's 4 packet transmission preventing node 2 from correctly receiving the RTS or sending the corresponding CTS to the node 1 [1]. However, the inter-flow interference determines the effect of the interference between flows that are in the interference range of each other. For example in the figure 2.1, we assume that the node 6 additionally start to send a flow of data to the node 12 (destination). As two flows of data coexist in the network, the interference range of the node 4 prevents with its transmission the transmission of the node 8 to the node 9, at the same time. Hence, two single chains of nodes suffer from the interference of each other, reducing the maximum utilization of the nodes.

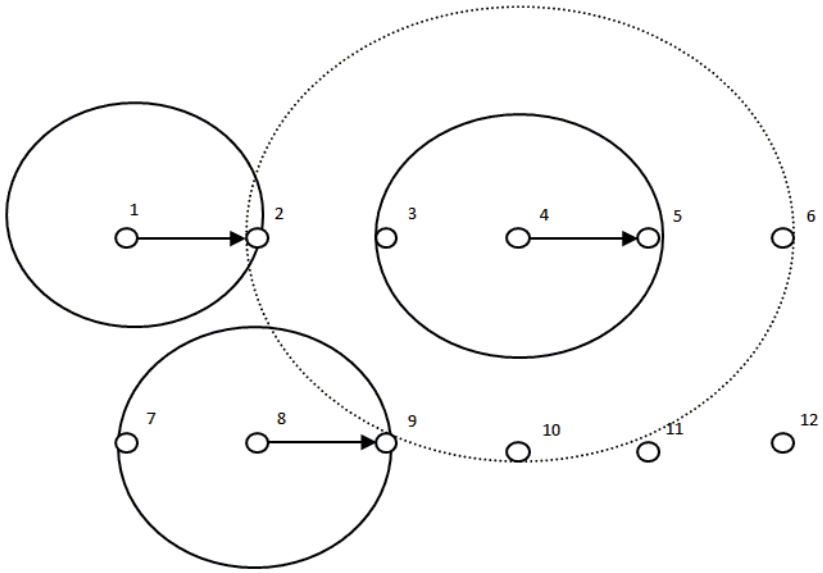

Figure 2.1: Inter and intra flow interference. The solid-line circle denotes a node's valid transmission range while the dotted line circle a node's interference range.

3. Simulation Methodology

This section describes a number of scenarios in order to illustrate the interactions between healthy and misbehaving nodes. Actually, it is an effort to present how some metrics e.g. throughput, delay, number of collisions, fluctuate under various combinations of these “bad” nodes. Our attempt is at showing how selfishness could in certain scenarios *benefit* the aggregate network throughput.

To begin with, the ns2 simulator was used to implement some simulations according to the topologies in Figure 3.1.

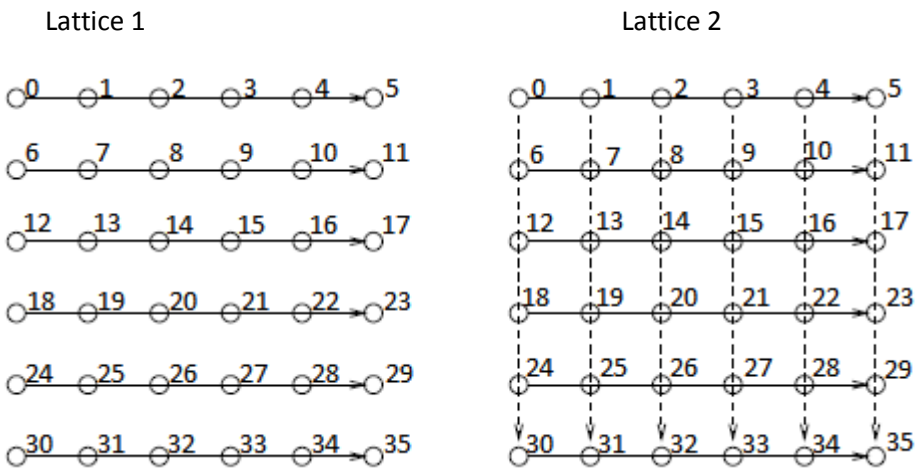

Figure 3.1: Lattice network topologies, indicating horizontal flows on the left, and both horizontal and vertical on the right [1].

More precisely, all the nodes are stationary and separated from their east, west, north and south radio neighbors by 200 meters, settled in a grid of 1000m X 1000m. We used the default values about the transmission and interference ranges of the simulator, which are 250 meters and 550 meters respectively. The Figure 3.1 shows the exact positions of the nodes.

All the nodes are tuned according to the 802.11 MAC protocol of the simulator at 1 Mbps. In the network layer we used the default values for the wireless configuration mode. Moreover, the ns2 simulator uses some radio propagation models, such as the free space model, two-ray ground reflection model, shadowing model, in order to predict the received signal power of each packet [15].In our simulations we used the two-ray ground reflection model. The details of the different scenarios for the set of misbehaving nodes are presented in later sections. We have used as routing protocols the AODV (Ad hoc On-demand Distance Vector) and the DSDV (Destination-Sequenced Distance Vector), as representatives of reactive and proactive routing protocols, respectively, and UDP as the transport protocol. We selected this topology in order to have the flexibility to give rise to various forms of radio interference (intra-flow interference, inter-flow interference) as well as hard contention either at packet level in the queues of nodes or at the MAC frame level, while competing for access to the medium.

The duration of each simulation run was 300 seconds. . In order to have more reliable results we let the flows start randomly within the time interval between 0 and 1 second using a Uniform probabilistic way. Results are drawn taking into account only the last 290 seconds of the simulation time.

3.1. Misbehavior Model

In this project we tried to emulate the packet-forwarding misbehavior. In particular, we used the DropTail queue of ns2-simulator. It is an already implemented queue which applies FIFO scheduling and drop-on-overflow buffer management. We only modified the method 'enque' from the class DropTail, integrating a probabilistic way to drop packets. The user has the possibility to specify the dropping rate of the receiving packets, for a node, in the source code of the simulation with the variable SELFISH RATE. Hence, when a node has a packet to enque, a random number is drawn from a uniform distribution that spans from 0 to 1. The selfish rate has also a range from 0 to 1. If the random number is smaller than the SELFISH RATE value, the packet is dropped. Otherwise the packet becomes enqued.

3.2. Misbehavior scenarios

3.2.1. Scenario 1 : Intra flow interference scenario—Lattice 1

In this scenario, we estimate the throughput (1) per flow for the lattice 1. We activate selectively horizontal chains of nodes to avoid inter-flow interference in the network since every third chain can operate without inter-flow interference. The configuration of the simulator for this scenario is described in Table 3.1:

$$\text{Throughput} = \frac{\#Successfully\ received\ \ packets\ of\ \ flow}{time\ of\ \ simulation\ run} \qquad (1)$$

Table 3.1 Configuration of the simulator for the scenario 1 simulations

Packet size	Sending interval	Number of simulation runs	Transport protocol	Routing protocol
64/500/1500 (bytes)	1 packet/0.001-0.1 (seconds)	50	UDP	AODV

For this simulation the first flow of data starts from the west node (source node 0) in the first line to the east node in the same line (destination node 5) and the second flow from node 18 to the node 23.

3.2.2. Scenario 2: Intra and Inter flow interference—Lattice 1

In this simulation set we are interested in the average aggregate throughput (2) for the square lattice 1. As traffic matrix, we adopt the scheme with the six horizontal flows as we can see in the right lattice of the figure 1. The related simulator configuration is summarized in Table 3.2:

$$\text{Average throughput} = \frac{\sum_{i=1}^{N} \#Successfully\ received\ \ packets\ of\ \ flow\ i}{time\ of\ \ simulation\ run \cdot N} \quad \text{(N: number of flows)} \qquad (2)$$

Table 3.2 Configuration of the simulator for the scenario 2, 3 simulations.

Packet size	Sending interval	Number of simulation runs	Transport protocol	Routing protocol
64/500/1500 (bytes)	1 packet/0.001-0.1 (seconds)	5	UDP	AODV

In both scenarios 1 and 2, we choose a range of values for the sending interval in order to assess the impact of the low and high levels of intra-flow interference without misbehaving nodes.

3.2.3. Scenario 3: Intra and Inter flow interference—Selfishness—Lattice 1

A natural extension of the previous scenario is to examine how the average capacity of the network varies when substituting healthy nodes by misbehaving nodes. For this purpose, we attach to these nodes the aforementioned misbehavior model and originate horizontal traffic flows moving from the left edge to the right edge (Figure 3.1, lattice 1). We run four different simulations with this traffic pattern configuring each time one more intermediate column of nodes between source and destination according to our misbehavior model. For instance, the first time we chose the nodes 1,7,13,19,25,31, the second time the nodes 2,8,14,20,26,32 and so forth. The only difference with the configuration of nodes in scenario 2 is that now we only use 64 bytes as packet size rather than the full set packet sizes considered earlier.

3.2.4. Scenario 4 : Intra and Inter flow interference—Lattice 2

In this scenario we take into account the right-hand lattice in figure 3.1. At the first step, we measure the average flow throughput. As it is shown in the lattice 2, all traffic originates from the top and left edges of the network, and is forwarded downward and rightward to the opposite edges. The configuration of the nodes is described in Table 3.3, which is the same for the scenarios 5 and 6 as well. The combination of this scenario with the scenario 5 helps to determine the throughput under the use of healthy and misbehaved nodes with a heavy network load.

Table 3.3 Configuration of the simulator for the scenario 4, 5, 6 simulations.

Packet size	Sending interval	Number of simulation runs	Transport protocol	Routing protocol
64 (bytes)	0.1 (seconds)	100	UDP	AODV/DSDV

3.2.5. Scenario 5: Intra and Inter flow interference—Selfishness —Lattice 2

Since we have measured the average flow throughput in the above simulation, now we assume different traffic patterns in lattice 2 with misbehaving nodes in order to evaluate the behavior of the network under a variable number of those nodes and dropping rates as well. Hence, our alternative considerations about the traffic patterns are:

5.1) We set as misbehaving nodes in turn:

1. Firstly, those nodes in the diagonal line (nodes 7, 14, 21, 28).
2. Then, the nodes in the second diagonal line (nodes 10,15,20,25)
3. Finally, we have made the inside square of the lattice 2 (nodes 7, 8, 9, 10, 13,14, 15, 16, 19, 20, 21, 22, 25, 26, 27, 28) to be configured according to our packet dropping model.

5.2) Some other traffic assumptions with more impact on the peripherals flows instead of the inside flows about the average throughput and the interference are:

1) Firstly, we enabled the dropping rate only for the node 1.
2) Both nodes 1 and 2 are misbehaving.
3) We also add node 8.
4) The final set of the misbehaving nodes are the nodes 1, 2, 8, 14.

3.2.6. Scenario 6: Intra and Inter flow interference—Selfishness —Lattice 2

For this scenario, we follow the same scenario with 5.1 with respect to the selected misbehaving nodes. Rather than throughput, we estimate some other metrics, such as end-to-end delay, number of collisions and which paths are followed by the data packets. We configure the node routing functionality with the AODV and the DSDV routing protocol. It is necessary to mention that for each simulation in this project about 35000 IDs of packets with data were produced by the sources. Hence, we chose one thousand packets with ID 15000 till 16000 in order to analyze their paths to the destinations. We chose these packets in order to be sure that network has been stabilized. In addition, due to the fact that the analysis was time-consuming we decided to examine only 1000 packets.

4. Simulation results

In this section, we present our results from the simulations we run for each of the six scenarios we describe above.

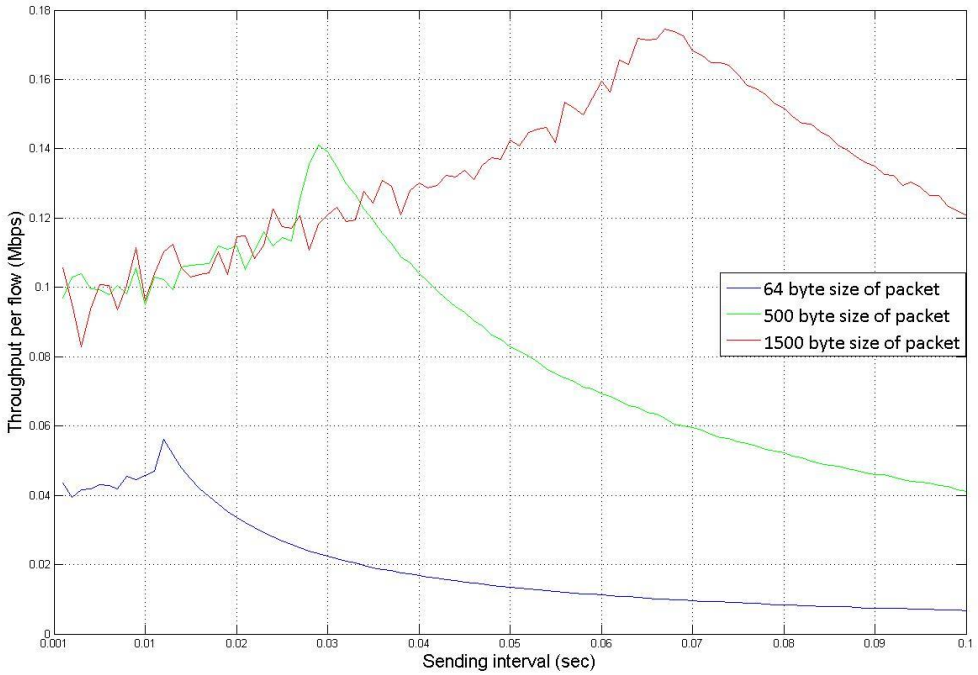

Figure 4.1: Throughput per flow for the square lattice 1 (scenario 1).

In figure 4.1, we only plot the throughput for the first data flow since we got almost the same results for the second flow. The sending interval of the source nodes is shown in the X axis and the throughput in the Y axis. This interval determines the time interval between successive packet emissions from the source. We chose three packet sizes, 1500 byte (red line), 500 byte (green line) and 64 byte (blue line), and a range of sending time intervals in order to depict how much the throughput varies when changing the traffic load of the network and therefore the level of the intra-flow interference. More precisely, we can see in figure 4.1 that, under light traffic load, increase of the packet size results in increase of the throughput as well. However, when the source starts to send 1 packet/0.1 second and accelerating towards 1 packet/0.001 sec, we notice a different behavior of the throughput. Firstly, the throughput increments for each packet size without the network suffering from heavy traffic load. Also, since this type of the topology does not exhibit inter-flow interference, the throughput of the flows grows by becoming more aggressive in claiming the available channel bandwidth. Afterwards, when the sending interval reaches certain values for each packet size, the throughput drops down. This reaction is due to the effect of the intra-flow interference; the transmission rate and the capacity of the network, at that time, are adequate to cover the demands of the network nodes. Especially, we can notice (figure 4.1) that as the packet size reduces, these crucial data rates become higher. For instance, the throughput begins to reduce

in the case of 1500 byte, having 0.065 packet time interval, while with the 500 byte and the 64 byte packet size, we have 0.028 (sec) and 0.013 (sec) as sending intervals, respectively. The intra-flow interference kicks in when a minimum threshold of traffic load is reached. Hence, for smaller packet sizes, we need higher data rates to create the critical traffic load within the network. Moreover, it's worth mentioning that when the sending data rate reaches the maximum value in our simulation run (0.001sec time interval), the network seems to stabilize its throughput. This happens because the intra-flow interference and the transmission time do not permit the nodes to forward more packets.

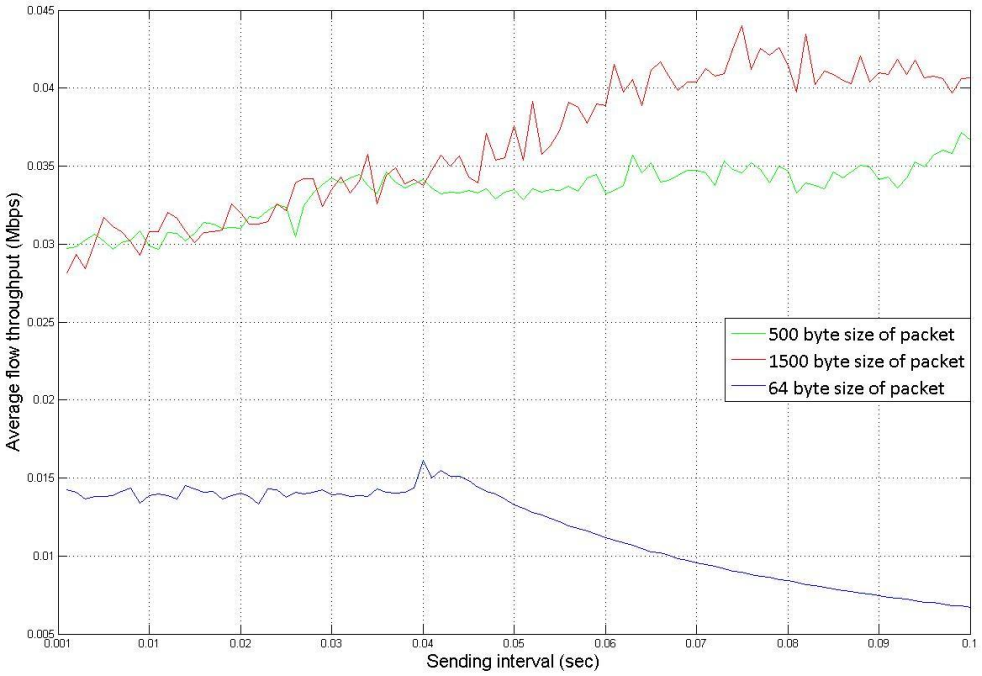

Figure 4.2: Average flow throughput in square lattice networks with horizontal data streams only (scenario 2).

The next figure 4.2 presents the average throughput of the six horizontal flows for the lattice 1 in figure 3.1 with different packet sizes. This figure presents a similar behavior of the throughput with that captured in the figure 4.1. The goal was to show the potential fluctuations of throughput, including the inter-flow and the intra-flow interference. Hence, the results of the 1500 and 500 byte as packet sizes (red and green lines) demonstrate our expectation; the increase of the data rate provokes the reduction of the throughput under these conditions. On the other hand, with the use of 64 byte packet sizes, the throughput increases until a certain value of the sending interval. This implies that level of the interference is not adequate in order to influence the overall throughput of the network, until a specific traffic load. After this threshold, as it is shown in the plot (blue line), the throughput remains constant. Thus, the inter- and intra-flow interference seems to take place drastically preventing the network to take advantage of the higher sending rates.

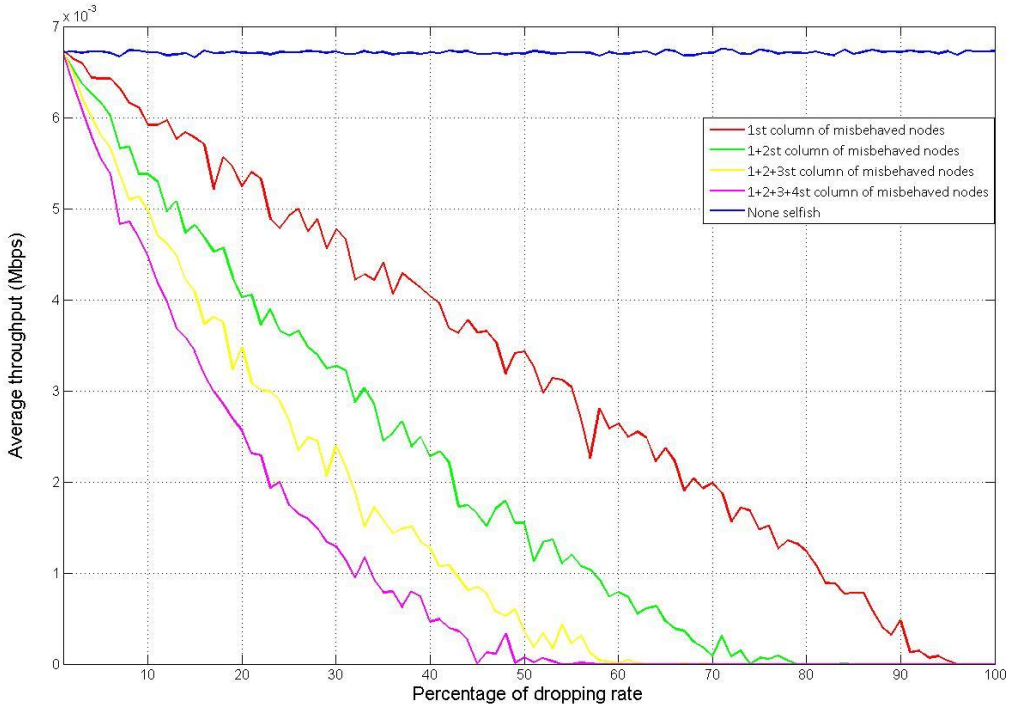

Figure 4.3: Flow average throughput as a function of the dropping rate in lattice 1 (scenario 3).

In the next figures we present some results about the throughput including our misbehavior model.

The figure 4.3 shows the average throughput under the presence of the misbehaved nodes. The X axis represents the percentage of dropping rate of the misbehaved nodes and the Y axis the average throughput. Each colored line represents the configuration of the selected scenario. This plot demonstrates that the choice of specific nodes as misbehaving ones can reduce dramatically the throughput of the network. More precisely, the colorful lines, except from the blue one, indicate decrease of the throughput as the percentage of the dropping rate increases. Furthermore, the figure 4.3 demonstrates that when more misbehaved nodes are used, more rapidly the throughput decreases and finally reaches the zero value. This happens because the routing protocol is not able to find alternative paths in order to avoid those nodes, due to the selected topology. Hence, the intermediate misbehaving nodes do not forward any packet to their neighbors and therefore to the destination.

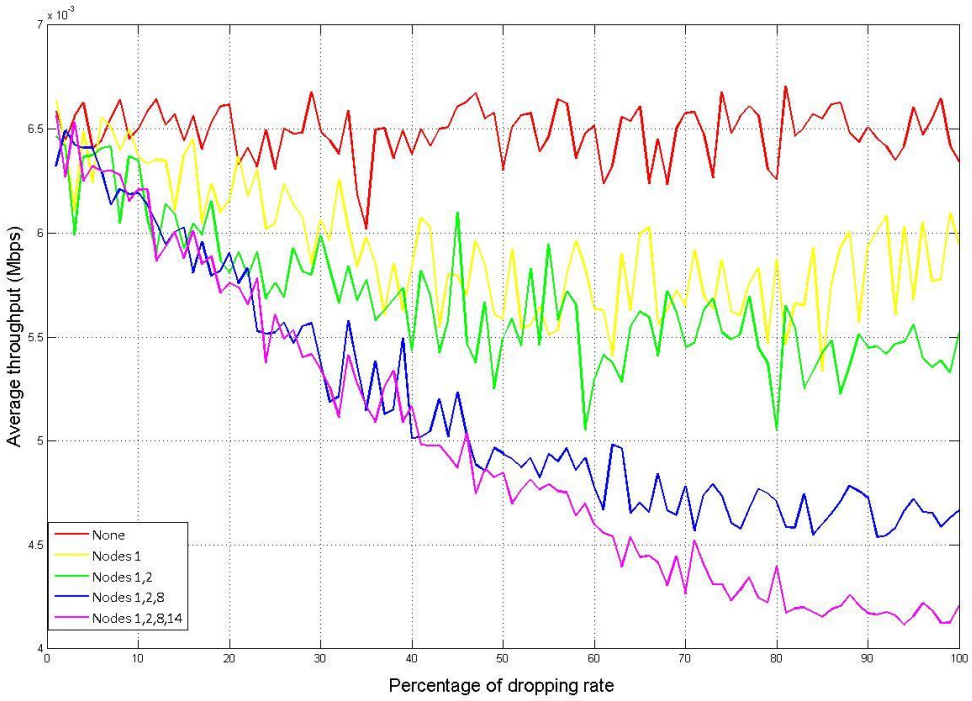

Figure 4.4: Flow average throughput of lattice 2 (scenarios 4, 5.2).

In figure 4.4 we present the average flow throughput of lattice 2 using traffic patterns with misbehaving nodes amongst the peripheral nodes (scenarios 4, 5.2) and AODV as routing protocol. The Y axis shows the percentage of dropping rate of the selected misbehaved nodes and in the X axis we have the value of throughput. In this plot we have integrated the scenarios 4, 5.2. Hence, each colored line represents the throughput under different sets of misbehaving nodes. Likewise with the previous figure 4.3, the throughput decreases and especially more rapidly, as we increase the number of certain misbehaved nodes. In contrary with the previous plot (figure 4.3), even if we use a number of misbehaving nodes, the throughput do not reaches the zero value. We selected the misbehaving nodes in such a way that the routing protocol is able to discover available-alternative paths to the destination ignoring somehow the misbehaved nodes.

As we expected, we observed a fall of the throughput when increasing the dropping rate (figure 4.3, 4.4) for both lattices. This reduction of the throughput becomes steeper as we increase the number of misbehaving nodes. It happens, as we said before, because the packets do not have the possibility to find alternative paths in order to avoid those unhealthy stations. Hence, a lower number of packets are distributed to the network.

The rest of our results are in contrary with what we expected. Below, there is a thorough description concerning those results.

In general, a satisfying number of surveys have shown that the throughput in the network is directly related to the number of the collisions that occur over a wireless or not topology. As the level of throughput increases, it is expected to have fewer and fewer collisions and the other way around.

Thus, we have two inversely proportional values. However, in our results we do not notice this pattern. Figures 4.5, 4.6 demonstrate that the behavior of the throughput, in relation to the number of collisions (figure 4.9), does not follow the above assumption. More precisely, in Figure 4.6, where the DSDV routing protocol is used, we would expect the throughput to increase and at the same time the number of collisions to decrease for each of the dropping rates, respectively. Moreover, the figures 4.5, 4.9, where the AODV routing protocol is used, show that the level of throughput does not affect the number of collisions. Thus, it is necessary to give some plausible hypotheses that explain our results.

First of all, it is appropriate to highlight that we have used UDP as a transport protocol in our simulations. As a consequence, we do not have retransmissions whenever the packets drop. This factor weakens the above assumption about the relationship between the throughput and the collisions. While there are many potential reasons for the nodes to drop packets, such as misbehaviors, availability of medium, inter-flow interference, intra-flow interference etc., they do not retransmit the dropped packets. Hence, this reaction assists the network to have low levels of congestion and contention, and therefore fewer collisions. In our experiments, under this scenario, we examined how much the throughput, the delay, the collisions and the paths of the packets change under different dropping rates by using a proactive (DSDV) and a reactive (AODV) protocol.

The AODV protocol is a reactive routing protocol. This protocol can be characterized as “on-demand”, because every node searches for a route to the destination when it has to send packets. In other words, none of the nodes sustain routing information if there is no need.

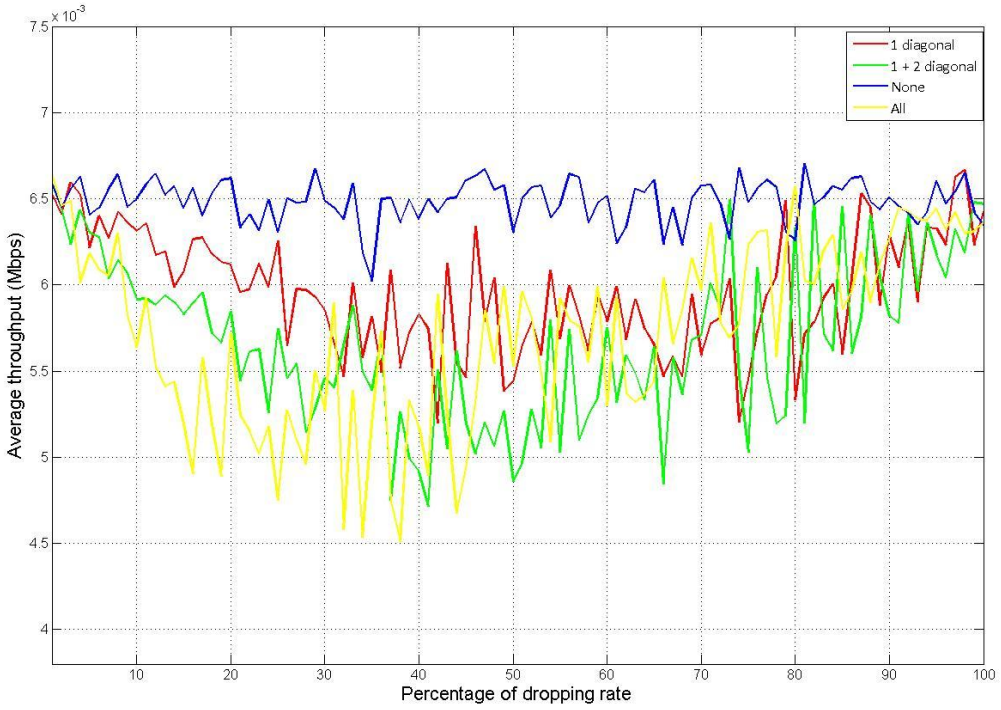

Figure 4.5: Average flow throughput of lattice 2 under AODV (scenarios 4, 5.1).

So, the figure 4.5 presents the average flow throughput of lattice 2 figure 3.1 using traffic patterns with misbehaved nodes in the inside square (scenarios 4, 5.1) and AODV as routing protocol. The Y axis shows the percentage of dropping rate of the selected misbehaved nodes and in the X axis we have the value of throughput. In this plot we have integrated the scenarios 4, 5.1.

We can see a different situation when compared to the results of the previous simulations. Whereas one might expect to see the throughput declining monotonically with the dropping rate throughout its value range, we notice that for each scenario with misbehaving nodes, the throughput rehabilitates after the 40-50 percent of dropping rate and ends up to have the same value with that at the beginning of the simulations. Moreover, it is obvious that as we augment the number of misbehaving nodes the throughput decreases with lower percentage of dropping rate each time.

The second protocol that was used is the DSDV proactive protocol. All the nodes have routing, up-to-date tables with paths to the rest of the nodes within the network topology. They maintain and exchange routing information even if they do not have packets to transmit.

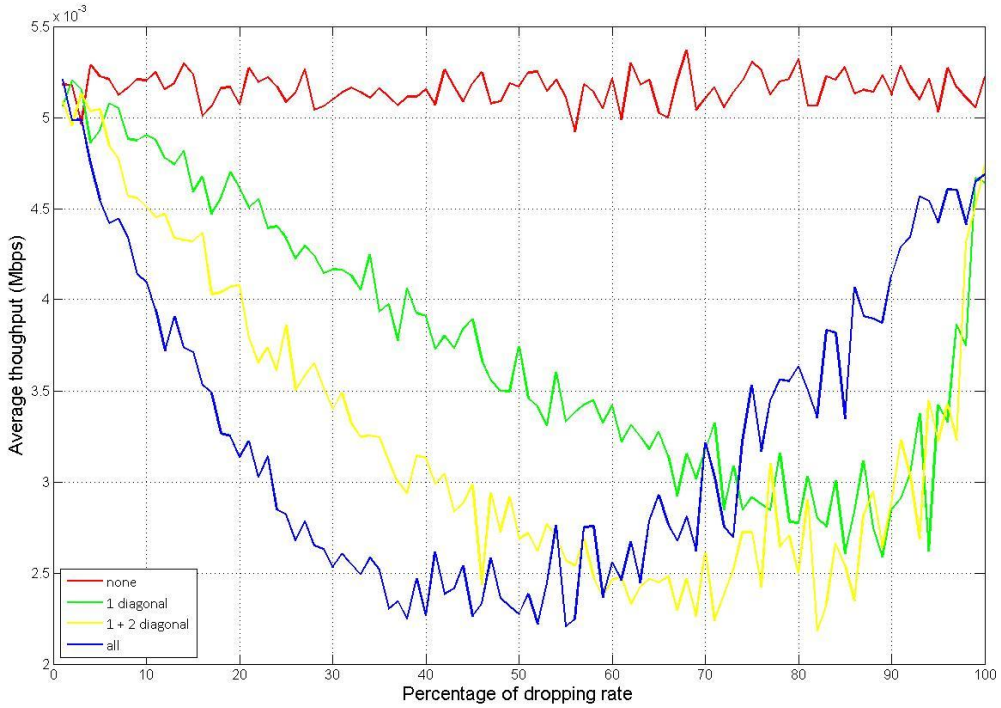

Figure 4.6: Average flow throughput of lattice 2 under DSDV (scenarios 4, 5.1).

Alike with the figure 4.5, the figure 4.6 depicts the average flow throughput of lattice 2 (figure 3.1) using traffic patterns with misbehaved nodes in the inside square (scenarios 4, 5.1) and DSDV as routing protocol. The Y axis shows the percentage of dropping rate of the selected misbehaved nodes and in the X axis we have the value of throughput. In this plot we have integrated the scenarios 4, 5.1.

In comparison with figure 4.5 we have about the same situation. The throughput decreases and afterwards increases again. However, it's worth noticing some particular differences with the previous plot (figure 4.5). First of all, in figure 4.5, when the throughput reaches the minimum level, it starts increasing again under the same dropping rates for each simulation. In figure 4.6 we do not have the same situation. The plot shows that as we increment the number of misbehaving nodes the throughput rehabilitates after lower dropping rates. Furthermore, the plot 4.5 indicates that all the simulations start and finish with almost the same value of throughput.

On the other hand, in figure 4.6 we can see that the throughput is lower at the end than at the beginning. The reduction of the throughput is due to the fact that the misbehaved nodes drop more and more packets until a certain dropping rate. Hence, the terminals receive fewer and fewer packets every time. Later on, we notice that the throughput increments again. This happens because there is a threshold in the frequency of dropping packages, so that the misbehaved nodes are not able to update their routing tables and have stale sequence numbers. In our case, this threshold is the value of the dropping rate where we have the minimum throughput. Consequently, as the protocol uses only the nodes that carry the most recent sequence numbers in order to find a path to

the destination, it starts to ignore them and consists in the healthy nodes (figure 4.8). Thus, the number of packets increases, whereas the dropped frames decrease. This reaction is based on the existence of alternative paths, which consist of nodes that can cover the demands of the protocol.

According to our results under the operation of the AODV protocol, the throughput follows the same behavior with that of the throughput of the DSDV protocol as described above (figure 4.5, 4.6 respectively). The reason of this situation is almost the same. The only difference is that the AODV routing protocol uses only the nodes, which respond to the process of request handshake and to the hello messages. If the nodes do not respond to these messages, the protocol considers that there are broken links with them and avoids to use them.

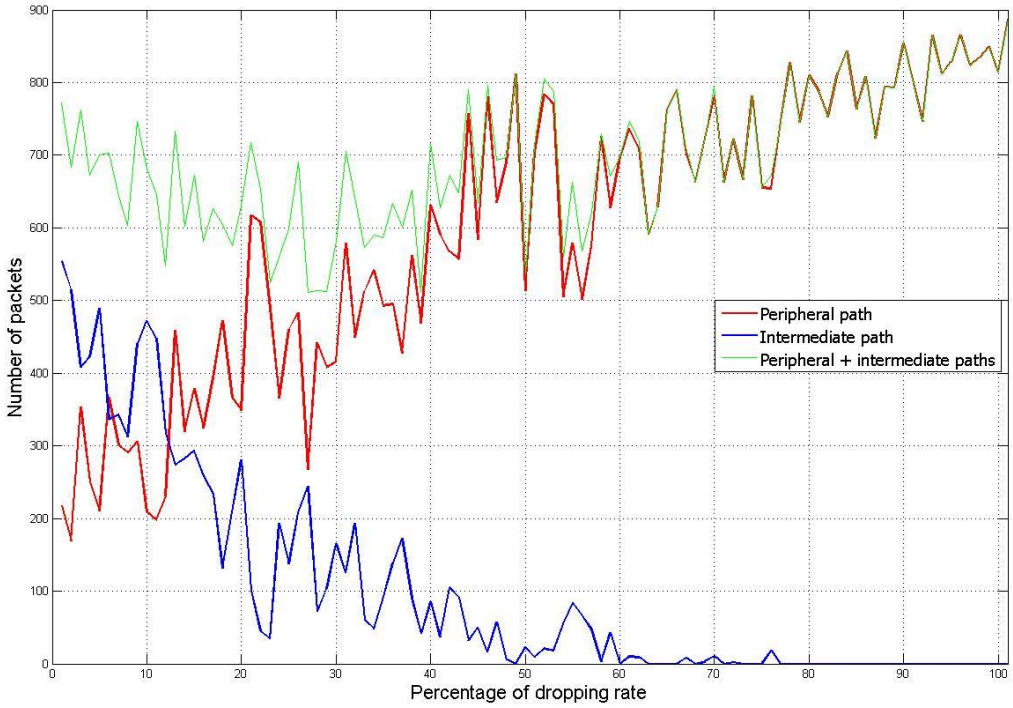

Figure 4.7: Path selection of the packets versus the dropping rate. AODV routing protocol.

Afterwards, we tried to examine thoroughly the path that the packets follow under the existence of the interference and the misbehaved nodes, in order to reach their destination.

In figure 4.7 we present the number of intermediate and peripheral paths of the packets according to the scenario 6 under the AODV as routing protocol. When we say intermediate paths we mean that the routing protocol has selected a path to the destination including at least one of the intermediate misbehaved nodes (7,8,9,10,12,13,14,15,16,18,19,20,21,22,25,26,27,28). On the other hand, the peripheral path are based on the cyclic nodes

(0,1,2,3,4,5,11,17,23,29,35,34,33,32,31,30,24,18,12,6) only. The X axis shows the percentage of dropping rate of the selected misbehaved nodes. The Y axis value is the number of packets that choose either intermediate paths to reach their destination or peripheral paths. Nevertheless we did not take into account the IDs of the packets which have as source stations the nodes 0,5,30 and as destinations 5, 30, 35. Especially, these packets travelled directly to the destinations using the minimum number of five hops. It is obvious from the lattice 2 in figure 3.1, that these packets are not affected from the selected misbehaved nodes. So, we wanted to analyze only the paths of the packets that they would choose intermediate nodes in normal conditions (without misbehaved nodes).

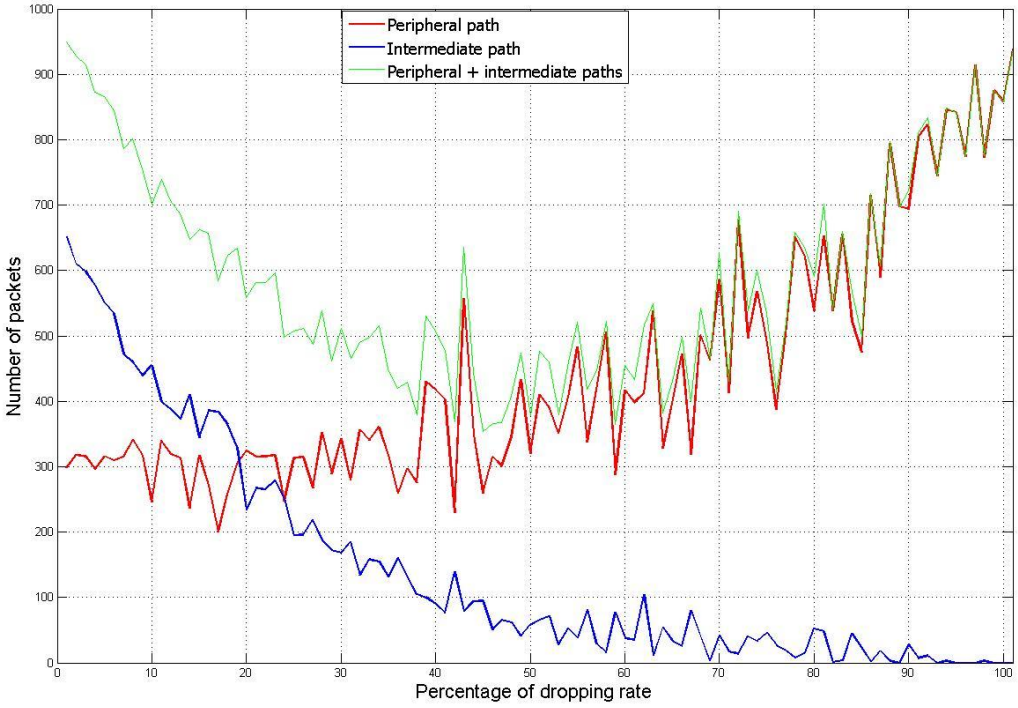

Figure 4.8: Path selection of the packets as a function of the dropping rate. DSDV routing protocol.

This plot 4.8 presents the same type of results with the figure 4.7. The only difference is the routing protocol. These results produced using the DSDV as routing protocol. Nevertheless, this behavior of the packets implies that as the misbehavior intensity grows, the number of packets that gets routed over the original path drops and the packets get more strongly routed towards the peripheral path. But while they use the intermediate path, with adequately high dropping rates, they suffer from dropping and reduced throughput. In other words, the routing protocol in this case is a bit slow in deciding not to use such impaired paths and route over the peripheral path.

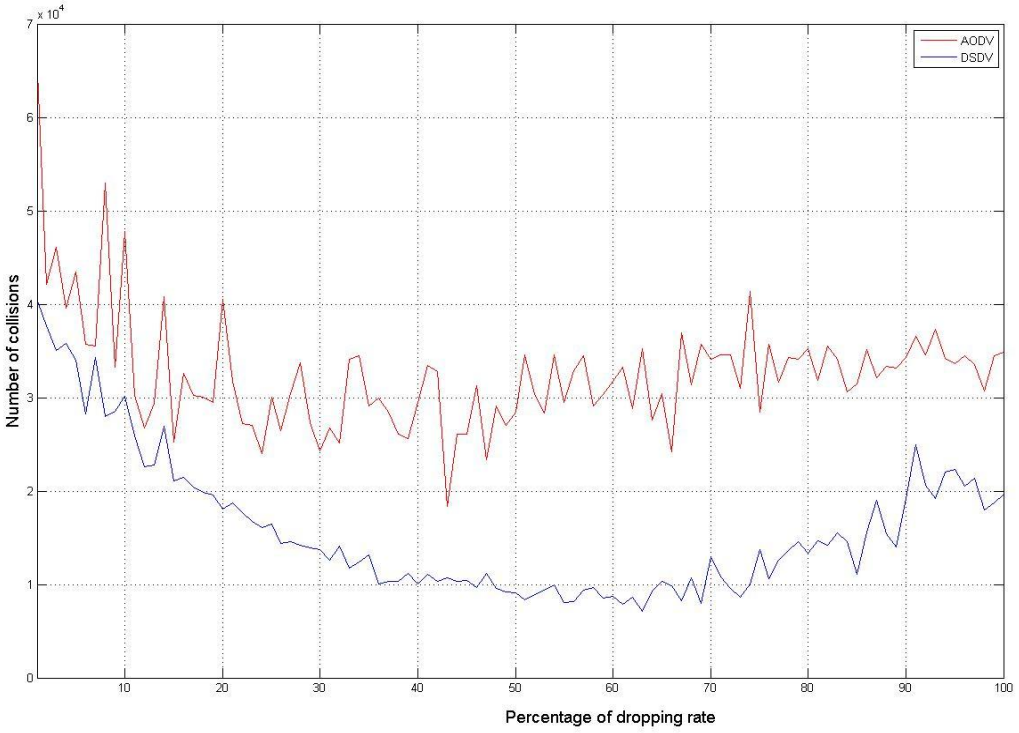

Figure 4.9: Number of collisions as a function of dropping rate under the scenario 6.

Moreover, we observed an unexpected behavior regarding the amount of collisions. In figure 4.9 we present the number of collisions as a function of dropping rate under the scenario 6. The red line shows the number of collisions for the AODV protocol, while the blue line the DSDV routing protocol. In the case of the DSDV protocol, when the traffic begins from the sources with zero percentage of dropping rate, the network reaches the maximum throughput. This implies that the throughput is kept under a specified number of collisions. Also, it is worth mentioning that the main routing overhead is not affected by changing the dropping rate. Particularly, the number of collisions that occur due to routing information are retained constant. This rate is the minimum value of collisions observed in figure 4.9. Because of the functionality of the DSDV protocol, each node, regardless of its participation in paths, informs its neighbors about its neighbors' characteristics (number of hops, sequence numbers). This type of routing update provokes a constant number of collisions. Hence, the rest of collisions are related to the number of packets that the nodes have to transmit. For this reason, the number of collisions decreases as the nodes have fewer packets to transmit. The collisions are concentrated on the intermediate and, especially, misbehaving nodes until they are ignored by the DSDV protocol. When the routing protocol starts to choose the peripheral healthy nodes, they tend to have more and more packets to transmit as they take over the propagation of the packets on behalf of the misbehaved nodes. So, the healthy nodes have in addition the collisions that the intermediate nodes would have, if they had to transmit their packets with zero dropping rate.

On the other hand, in AODV protocol, we can see that the collisions are maintained constant in general. We consider that the misbehaved nodes limit the efficiency of the routing protocol to

discover a path to the destination. They often drop crucial packets, which are necessary to set a route. Hence, the protocol sends repeatedly requests to find a new path. In order to prevent searching into the same misbehaving nodes without finally discovering a new one, it uses a blacklist with the failures of the nodes. A potential question at this point could be “why the level of collisions is the same instead of being reduced when the packets follow peripheral paths (figure 4.7) as an alternative and effective solution?” A plausible answer in this question is that the nodes have a short memory about the active paths. So, it is possible that when the nodes need again the same paths, they might have expired. Then, the sources repeat the process for a path request. However, during the delivery of the frames some misbehaving nodes may have been removed from the blacklist, and therefore continue to confuse the routing until they are detected again. Consequently, we have a kind of recycling of the collisions.

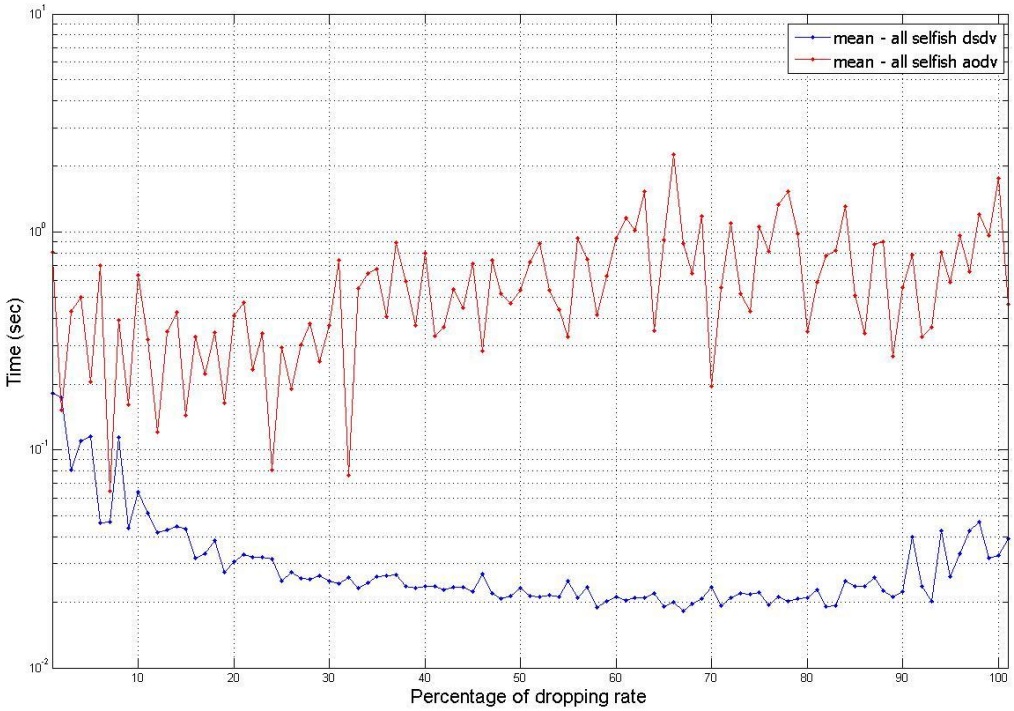

Figure 4.10: End-to-end delay for each routing protocol (scenario 6).

As far as the end-to-end delay is concerned, it is apparently constant (figure 4.10). In this plot we have the average end-to-end delay for each routing protocol (scenario 6). The Y axis (log scale) shows the average delay for different dropping rate(X axis) of the misbehaved nodes. Under the AODV routing protocol we have a slight increase of the end-to-end delay, while in the DSDV we notice a small decrease of the delay. In the DSDV, we assume that this happens because of the selected topology. If we take into account the exact positions of the nodes, we can conclude that every third chain can operate without inter flow interference. Therefore, we believe that there is a tradeoff between the interference and the number of hops. In the case where the intermediate nodes participate in the selected paths, we have smaller paths to the destination but a lot of nodes that can transmit at the same time. While, in the case of the peripheral nodes we have more nodes

that can propagate frames at the same time, we have bigger distances to the destinations. Finally, in the case of the AODV, this is not so strange as the number of hops to the destination increments and additionally, the reactive protocol spends a bit more time for the paths than the proactive protocol, with which the nodes have a priori ready the routes to the destinations. Also, the contention between the routing protocol and the misbehaved nodes, potentially, add some delay provoking extra backoffs.

5. Related Work

Marti *et al.* [9] propose two mechanisms to mitigate the node misbehaviors in mobile ad-hoc networks, called the “watchdog” and the “pathrater”. The watchdog mechanism identifies the misbehaving nodes comparing the overheard packets with the recently sent packets which are maintained in a buffer by the watchdog. When a packet exceeds a certain remaining period in the buffer, then the node, who is responsible for forwarding on the packet, is punished with an increment of his failure ratio. If a node reaches a maximum threshold, the watchdog sends a message of misbehavior to the source for the certain node. The pathrater owns information about misbehaving nodes and link reliability data. Hence, each node allocates a rating to every other node it knows about in the network, in order to choose the most reliable route between the source and the destination. Yang *et al.* [10] have used the watchdog mechanism in order to implement the AODV-S routing protocol, an extension of the AODV. With this combination, this protocol has the capability to detect problematic packet forwarding misbehaviors such as packet dropping, packet duplicating, and network layer packet jamming. In our project, as we showed, we used topologies integrating nodes with packet dropping misbehavior. Also, we demonstrated that the routing protocols that we used (AODV, DSDV) begin to isolate the misbehaving nodes, as the packet dropping rate increases. Hence, the AODV-S could be viewed as an alternative solution to avoid those misbehaviors.

In [6], the authors present DOMINO (Detection Of greedy behavior in the MAC layer of IEEE 802.11public Networks), another technique to detect misbehaving nodes. Domino is entirely integrated in the AP and periodically collects traffic traces of active users. According to an implemented modular architecture, DOMINO analyzes the traces and aims at determining some behavior anomalies in the traffic. Based on the results from the analysis, a Decision Making Component (DMC) takes place in order to handle the situation. Although the DOMINO is a robust mechanism, which can protect the network from a variety of attacks, it suffers from some severe vulnerabilities. For instance, the DOMINO does not work properly under hidden terminals, it can be used to create hybrid attacks and it may be exploited relying on the “adaptive cheating”.

Some other kinds of mechanisms to detect packet forwarding misbehaviors are those which are based on reputation schemes. Particularly, He,Wu, and Khosla [12] have proposed the SORI, a Secure and Objective Reputation-based Incentive scheme. It uses a watchdog-like mechanism for monitoring. Its reputation metrics are based on the number of packets forwarded by and for the neighboring nodes. Hence, every node keeps reputation ratings for every other node, which are sent through the local network. SORI additionally implements a hash-chain based authentication for propagated reputation ratings.

Furthermore, Michiardi and Molva [13] presented the CORE, a COllaborative REputation mechanism that uses a watchdog component for monitoring. According to the reputation system of the CORE, the reputation values depend on the observations (subjective reputation), the positive reports by others (indirect reputation), and the task-specific behavior (functional reputation). The nodes are classified into *requestors, providers or trusted entities* based on their behaviors in order to have a peaceful coexistence inside the network.

A similar approach is described in [14]. In that paper the authors propose the CONFIDANT, a Cooperation Of Nodes Fairness In Dynamic Ad-hoc NeTworks reputation system, which aims at not only detecting and avoiding, but also isolating misbehaving nodes. The CONFIDANT protocol relies on the following components in each node: a neighborhood monitor, which identifies deviations from the normal routing behavior, a trust manager, which sends and receives alarm messages to and from other trust managers, a reputation system, which rates other nodes according to their observed or reported behavior, and a path manager, that maintains path rankings and performs specific actions when processing routing messages that involve misbehaving nodes.

In general, both reputation systems, CONFIDANT and CORE, cannot be considered as a solution to our packet-forwarding misbehavior. As these mechanisms cannot determine the reason that a packet is dropped, they are not able to discriminate the dropped packets due to misbehaved nodes and especially the intra/inter flow interference. Likewise, even if the incentive scheme SORI is highly possible to detect selfish nodes, the objectively measured reputation of the nodes does not work well under a heavy network load. This is because SORI cannot take into account the packet collisions for the estimation of the nodes reputation. Hence, we cannot utilize the SORI to address our problem with misbehaving nodes including a large amount of packet collisions due to the intra and inter flow interference. Nevertheless, according to the implementation of the SORI, it uses a margin δ to avoid punishing well-behaving nodes which drop packets due to some phenomena such as collisions, rather than selfishness. They have shown that the appropriate configuration of δ permits SORI to work well for a light or medium network load.

Pietro Michiardi and Refik Molva [11] provide a simulation analysis based on mobile ad-hoc networks. They have implemented three models of selfishness using mobile nodes. They attempt to assess the performance of the network in terms of global throughput and delay with relation to the number of nodes and their mobility. In our work, in contrast, we examine the performance of the average throughput and the delay under stationary nodes and different model of misbehavior. We used the AODV and the DSDV as routing protocols, instead of the DSR. When the nodes use the DSR, they do not retain routing tables. Hence, as the existence of the misbehaved nodes usually gives the impression to the routing protocol that there are broken links with their neighbors, and the route maintenance mechanism cannot locally repair a path during a path discovery, the routing overhead increases radically. Additionally, our misbehaved nodes do not distinguish to drop, between packets of data or routing packets. This means that they participate in the route discovery phase of the selected routing protocol, performing the packet forwarding function as well.

6. Conclusions

In this project, we examine the impact of the misbehaved nodes over a number of traffic patterns. In general, we tried to implement a plethora of topologies under a certain kind of malice or selfishness. Firstly, we ran simulations integrating only intra- and inter-flow interference. Afterwards, implementing our misbehavior model we analyzed some crucial metrics such as throughput, number of collisions and end-to-end delay in order to assess the effect of the interference in combination with the misbehaved nodes. Moreover, we demonstrated that the existence of the misbehaved nodes forces the routing protocol to change the active paths.

After the analysis of the results, we can conclude that the routing protocols are adapted to the topologies in order to reach the optimal result. We saw that there are topologies which permit to overcome the effect of the misbehaved nodes and others which do not. When we started the simulations we expected to observe a sequential reduction of the capacity of the network which was proved not to be the case under certain circumstances. Furthermore, some results indicated that the rehabilitation of the effectiveness of the network costs, by increasing the end-to-end delay or the number of collisions. Even if we have chosen very different routing protocols, finally we didn't find significant discrepancies in the behavior of the nodes.

References

[1] J. Li, C. Blake, D.S.J. De Couto, H.I. Lee, R. Morris, Capacity of ad hoc wireless networks, in*: ACM Annual International Conference on Mobile Computing and Networking (MOBICOM),* 2001, pp. 61–69.

[2] Ekram Hossain, Kin K. Leung, *Wireless Mesh Networks, Architectures and Protocols*,

Springer, ISBN-10: 0387688382, ISBN-13: 978-0387688381.

[3] Aggelou G. *Wireless Mesh Networking*. New York, NY: McGraw-Hill Professional, 2008

[4] Y. Zhang, J. Zheng, and H. Hu, *Security in Wireless Mesh Networks*,CRC Press, 2008

[5] N. Ben Salem, and J.P. Hubaux, "Securing wireless mesh networks", IEEE *Wireless Communications*, Vol. 13, No. 2, pp.50-55, April 2006.

[6] L. Buttyan and J.-P. Hubaux. *Security and Cooperation in Wireless Networks: Thwarting Malicious and Selfish Behavior in the Age of Ubiquitous Computing*. Cambridge University Press, 2007.

[7] C. Perkins, E. Belding-Royer, and S. Das. Ad-hoc on-demand distance vector (AODV) routing. Internet RFC 3561, July 2003.routing. Internet RFC 3561, July 2003.

[8] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of the SIGCOMM'94 Conference on Communications Architectures, Protocols, and Applications*, August 1994.

[9] S. Marti, T. Giuli, K. Lai and M. Baker, Mitigating routing misbehavior in mobile ad hoc networks, in: *International Conference on Mobile Computing and Networking (MobiCom 2000)* (2000) pp. 254.265.

[10] H. Yang, X. Meng, and S. Lu, .Self-organized Network Layer Security in Mobile Ad Hoc Networks,. in *WiSe*, 2002.

[11] P. Michiardi, R. Molva. Simulation-based Analysis of Security Exposures in Mobile Ad Hoc Networks. European Wireless Conference, 2002.

[12] Qi He, Dapeng Wu, and Pradeep Khosla. SORI: A secure and objective reputation-based incentive scheme for ad hoc networks. In *IEEE Wireless Communications and Networking Conference (WCNC 2004), Atlanta, GA, USA*, March 2004.

[13] Pietro Michiardi and Refik Molva. CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. Sixth IFIP conference on security communications, and multimedia (CMS 2002), Portoroz, Slovenia., 2002.

[14] S. Buchegger and J.-Y. Le Boudec. Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes – Fairness in Distributed Ad-hoc NeTworks). In *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, Lausanne, Switzerland, June 2002.

[15] K. Fall, The ns Manual (formely ns Notes and Documentation), January 6,2009, http://www.isi.edu/nsnam/ns/ns-documentation.html

[16] J. Jun and M. L. Sichitiu, “The nominal capacity of wireless mesh networks,” *IEEE Wireless Communications*, vol. 10, no. 5, pp. 8–14, Oct. 2003.