Les Cahiers du GERAD ISSN: 0711–2440

A simulation framework for network performance evaluation of large-scale RF-mesh AMIs

F. Malandra
B. Sansò

G–2016–19

March 2016

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

A simulation framework for network performance evaluation of large-scale RF-mesh AMIs

Filippo Malandra
Brunilde Sansò

GERAD & Department of Electrical Engineering, Polytechnique Montréal, Montréal (Québec) Canada, H3C 3A7

filippo.malandra@polymtl.ca
brunilde.sanso@polymtl.ca

March 2016

Les Cahiers du GERAD
G–2016–19

Abstract: Advanced Metering Infrastructures (AMIs) lay the foundation of a wide variety of smart grid applications. Despite their low capacity, RF-mesh systems are an increasingly popular option for AMI implementation. In this paper, we present a simulator framework of realistic RF-Mesh AMI with the objective of analyzing the performance of this kind of system, and defining the feasibility of possible smart grid applications. The framework yields a simulation tool, implemented on Java and Python, that permits the simulation of large scale AMI, while taking into account important technical details in a reasonable amount of time.

Key Words: Wireless Mesh Network, FHSS, ALOHA, performance analysis, RF-mesh.

1 Introduction

RF-MESH Advanced Metering Infrastructure (AMI) are increasingly popular and they can be used as a transmission support for different types of smart-grid applications. However, the issue of how to properly evaluate their performance is a roadblock to be able to assess system behaviour when all those applications are functional. Despite the fact that existing simulation platforms are extensively used in general networking, some issues and gaps were observed in the RF-AMI performance literature. First, there is *a lack of dedicated modules* designed exclusively to the RF-mesh technology in affordable software. Second, some important *modelling details* such as FHSS are not implemented. Third, there is the issue of *low scalability*, as a high percentage of the reviewed papers simulated wireless systems up to few hundreds of smart meters, a consistently lower scale compared with currently implemented RF-mesh AMI. Moreover, many articles did not focus on the topological and geographical aspects of the problem that are an important item to better analyze the peculiarities of RF-mesh AMI. Finally, there is a lack of publicly available information that makes it difficult to compare and reproduce different situations and results. Some authors do not share details of their simulations, others use software that are not open-source.

To fill the gap with currently available solutions, we propose a simulation framework based on a detailed performance analysis of large-scale RF-mesh AMI systems. The methodology permits to calculate the probability of collision of packets and the end-to-end delays in different areas, analyzing eventual geographical bottlenecks of the system. Wireless interference and its negative effects on performance are carefully accounted for, as well as other performance indexes, such as the activity time, that can be interesting to limit the public fears on RF exposure. The framework is highly scalable and allows the study of large scale geographical regions composed of thousands of smart meters, running different types of applications.

The reminder of this document is structured as follows. Section 2 describes the state of the art in AMI simulators. The main features of a RF-Mesh system are presented in Section 3. The Simulation framework is presented in Section 4. Numerical results obtained with the developed simulation tool are studied in Section 5 wheres in Section 6, the conclusions and some discussions are presented.

2 State of the art in RF-mesh AMI simulator studies

Network simulators are usually designed to recreate the performance behavior of the system taking into account network layer protocols and conditions. Therefore, when working with RF-systems, cross-layer modules have to be integrated to commercial and/or open source simulation codes to be able to study the network layer performance while also considering the lower layer functionalities and phenomena. Reviewing the literature on RF-mesh AMI simulator studies, we found that most work relied on OPNET, others on NS-2 and NS-3, some considered OMNET++ and finally, there was a proposal for CASTALIA.

OPNET is a commercial network simulator with a nice GUI and a wide variety of modules to simulate different wired and wireless technologies and protocols. Therefore, many authors, [1, 2, 3, 4, 5, 6], performed OPNET-based simulations of AMI systems. They carefully took into consideration some aspects of AMI implementations (e.g., routing, propagating condition, QoS requirements). However, the network sizes used in the published studies were too small to retain the large scale phenomena that are present when deploying real life RF-mesh AMI.

Others (e.g., [7, 8, 9]) carried out simulations using NS-2 and NS-3, two very popular open-source discrete-event network simulators. The two simulators allows for a high precision representation of the physical channel and complex propagation models (i.e., Nakagami-Rayleigh in [8] and HATA COST231 in [9]) but, again, only small size networks (around 60 nodes) were considered in the two contributions and the time horizon was too short (e.g., 20 seconds in [9]) to provide general considerations for AMI performance. [7], however, used 350-node instances but their analysis was more focused on the performance of a AMI under a cyber-attack and not necessarily on general application performance issues.

The free discrete event-driven network simulation software OMNET++ is also used in the RF-mesh AMI performance literature. In particular, [10] compared different routing solutions and [11] analyzed a smart meter communication system very similar to the one under study in this work. Despite the large scale nature of the instances, in [10] GIS data were not used and the positions of smart meters were randomly distributed in a given area. This choice reduces the fitness to actually implemented AMI because smart meters are usually installed in residential addresses, which are not randomly distributed in space. This drawback does not apply to the inspiring work of [11], where GIS data were used to build consistent large scale instances; however, the author mainly focused on hop count statistics whereas just few or no details were provided on some important performance indexes, such as latency or the amount of time spent in transmission by smart meters.

CASTALIA, another open-source discrete-event-driven network simulator, was used in [12] to analyze the AMI performance. Nevertheless, there was again the issue of instance size that was in the order of few hundred nodes. Moreover, the performance analysis only encompassed the hop count and an end-to-end delay histogram, not considering the impact on the performance of geographical (e.g., the position) and topological (e.g., the number of neighbors) parameters, thoroughly evaluated in this work.

As a conclusion, the study of the literature on AMI simulations shows that modifying general purpose networking simulators leads to some important shortcomings to capture the performance of large-scale systems. As a result, and differently from the aforementioned studies, we decided to part from implementing yet another module of an existing network simulator and rather propose a new framework that captures the most important performance features while allowing the study of large-scale deployed AMI-systems.

3 Description of the RF-mesh system

The RF-mesh system under study, akin to the one described in [13, 14], is composed of 1) *Data collectors* that are gateways between the NAN and the WAN and have enhanced capacity; 2) *Routers* that are used as relays between meters or between a meter and a data collector and 3) *Smart meters*. The system is characterized by six elements: 1) a large number of nodes (in the thousands for large deployments); very low throughput (typically 19.2 kbps for links between routers and data collectors, and 9.6 kbps for other links); 3) black box nature of the deployment (which forces to do some reverse engineering); 4) a MAC layer based on slotted ALOHA, 5) high interference mitigated by an FHSS ; 6) layer-based routing at the network level.

4 Simulation framework

4.1 Initialization phase

The first phase of the simulation framework, schematized in Figure 1, is the *Initialization*, whose main objective is to prepare the instance to be simulated. The user is given the possibility to choose the instance that best reflects its needs. The flexibility on this phase is very important to enable a broad set of performance analysis. Figure 2 shows that the initialization phases encompasses the nodes placement (Section 4.1.1), the link definition (see Section 4.1.2), and the routing (4.1.3).

4.1.1 Nodes placement

Let us define some useful mathematical notation that will be used in the rest of this section:

V	Set of nodes in the topology
M	Set of smart meter nodes
R	Set of routers
C	Set of data collectors
E	Set of links in the topology
$d(i,j)$	Euclidean distance between $i \in V$ and $j \in V$
R_m	Covering ray of smart meters
R_r	Covering ray of routers / collectors
$\nu(i)$	Set of neighbors of node i

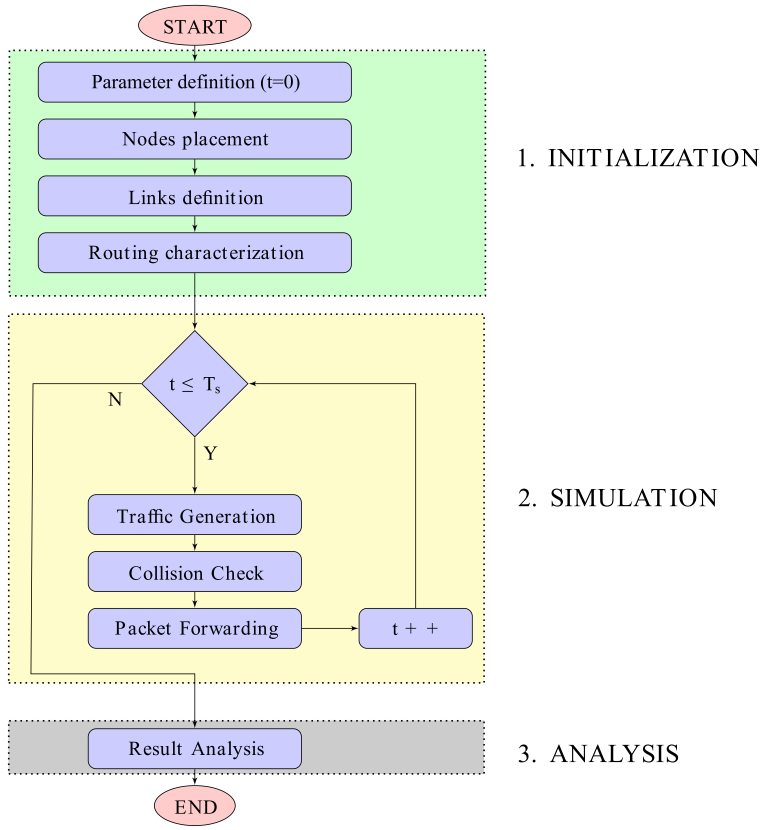


Figure 1: Simplified architecture of the simulator.

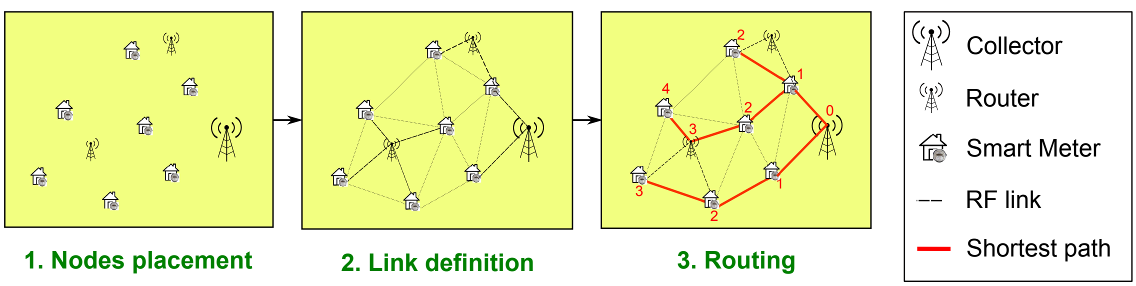


Figure 2: A scheme of the initialization phase with a simple topology with 10 nodes.

As already mentioned, three types of devices are foreseen: the smart meters M, the routers R, and the data collectors C. It is important to remark that every device $i \in V$ has a unique functionality and cannot be of two different kinds neither it can change its role in the course of a simulation: $M \cup R \cup C = V, M \cap R = M \cap C = C \cap R = \varnothing$. Every node is characterized by 1) its role, 2) a unique integer ID, and 3) its GPS coordinates. The ID notation follows the convention that the first node is a collector with ID equal to 0. The other IDs are incrementally assigned starting by any other data collectors [1], followed by the routers and finally the smart meters.

The topology to be used in the simulation is determined by the node placement and the link definition. Nodes position can be either manually selected in a map by means of a geographical tool we implemented, or directly provided in text files, structured to be used within the proposed framework. Since the number of smart meter is very large, it is not feasible to manually draw them on a map. It is rather more convenient

[1] At least one data collector is required but, as shown in the rest of the paper, the number of data collectors might be greater than 1.

to use publicly available data, or to retrieve their position by one of the various portals providing open GIS data.[2] On the other hand, the position of routers and collectors can be not only selected on a map, but also optimally determined by integrating a facility location model into the current framework.

4.1.2 Link definition

The topology is then completed with the definition of the set E of links. In order to keep the complexity low, the definition of links is based on the distance between nodes. In particular two different covering rays were defined: R_m for smart meters, and R_r for routers and collectors. Numerical values of R_m and R_r vary according to the different scenarios: in urban environments, the covering rays are typically shorter than in rural ones, this is due to the harsh propagating condition with many obstacles in the path and also because of the concurrent transmissions of other wireless users in the same bandwidth (as already mentioned, these systems use the ISM bandwidth that is not reserved). Moreover, the covering ray of routers and collectors is sensitively higher than the ray of smart meters because of the use of better antennas and also because routers and collectors can exploit better propagating conditions since they are usually installed on the top of buildings.

The set of links $e_{ij} \in E$ is characterized as follows:

$$\begin{aligned} E = \{e_{ij} : (i, j \in V) \wedge (d(i,j) \leq R_m \wedge i, j \in M) \vee \\ \vee (d(i,j) \leq R_r, (i \in R \cup C) \vee (j \in R \cup C))\} \end{aligned} \tag{1}$$

where $d(i,j)$ is the distance between node i and j.

4.1.3 Routing

In RF-mesh systems, packets flow in just two directions: uplink (from a smart meter to the data collector) and downlink (from a collector to a smart meter). The communication between two smart meters is not possible for confidentiality reasons: messages are encrypted with a key that is shared between the data collector and each smart meter. The routing mechanism is the so-called *layer-based routing*, similar to the one presented in [15], and is based on an initial shortest path calculation. The shortest paths are bidirectional and are used as the initial routing mechanism. In particular, each smart meter $i \in M$ is assigned to the closest data collector $j \in C$ and the shortest path between them is used to route the packets from i to j and vice versa. After the shortest path definition, each node is assigned a layer, according to the layer-based routing paradigm, in order to be able to react to eventual changes in the topology (e.g., link losses, node's failures). In case a given node i with layer k looses connection with $(k-1)$-layer node (i.e., the node that follows i in its shortest path towards the data collector), i will choose one of its neighbors with the lowest layer number, and a path to the collector can be promptly reestablished. The shortest path routing is represented in Figure 2. The shortest path from smart meters to the data collector are computed (red lines) and the layer number of each node is displayed. The choice of a very basic static routing based on bidirectional shortest paths was driven by the necessity to keep low the computational burden of this simulator: layer-based routing is useful to embody the dynamic nature of wireless mesh networks. In a wireless system with infrequent route changes and very low data rates a mixed shortest-path/layer-based routing algorithm can be considered a good choice in the trade-off between simplicity and fitness to reality.

4.2 Simulation phase

A scheme of the logical structure of the nodes is reported in Figure 3. As shown in the two aforementioned figure, each node is equipped with 1) 2 logical antennas, 2) a FHSS module, 3) 2 separate buffers (one for incoming packets and one for packets waiting to be transmitted), 4) a packet analyzer, 5) a routing mechanism. However, the elements in the grey area (i.e., destination and traffic generator) are not implemented in routers but only in smart meters and collectors, which can be sink or source of traffic.

[2]Data for the Montreal island, used in our numerical results, can be found at http://donnees.ville.montreal.qc.ca/dataset/adresses-ponctuelles

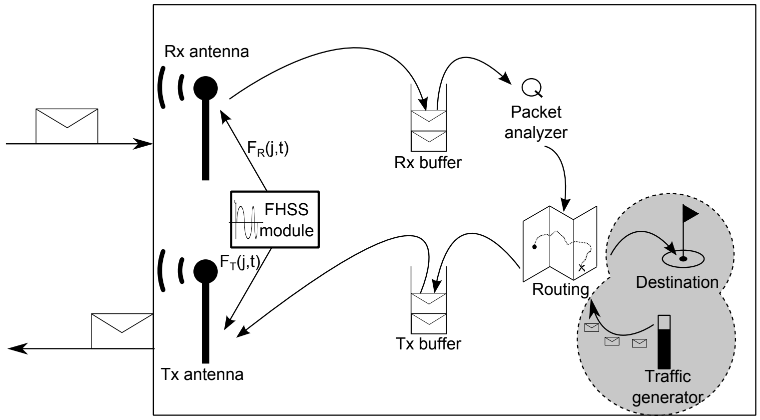


Figure 3: Logical structure of the different types of nodes in the simulator.

Let $F_r(j,t)$ and $F_t(j,t)$ be the frequency channels of the two antennas (respectively, transmitter and receiver) of node j at time t. Each node uses the same predetermined frequency channel sequence opportunely shifted in time. The FHSS module can be used to determine the channel $F_r(j,t)$ of all nodes $j \in V$ at each time interval t. The two antennas are two logical entities associated to the same physical antenna each node is equipped with. The presence of two different antennas is needed to describe the different modes of operation of nodes according to the FHSS protocol.

A given node i, at the beginning of a time slot, can be either in *idle* state or in *transmission* state. When in idle state, it is waiting to receive the packet and its antenna is tuned to the frequency $F_r(i,t)$ according to a predetermined sequence known to all the nodes. On the other hand, the transmitting frequency of node i at time t ($F_t(i,t)$) depends on the destination of the transmission: if node i wants to transmit a packet to node j at time t, it must tune its transmitting frequency to the receiving frequency of node j at time t, namely $F_t(i,t) = F_r(j,t)$. The computation of $F_r(j,t)$ is managed by the FHSS module. This mechanism is implemented to reduce the chances of collision: a collision in the transmission from node i to node j happens when another node is transmitting on the same channel $F_r(j,t)$.

The simulation phase, as shown in Figure 1, consists in the iteration, over the simulation horizon T_s, of three sub-phases (i.e., traffic generation, collision check, and packet forwarding) that are described in detail in the next subsections.

4.2.1 Traffic characterization and the traffic generator module

At the beginning of a time slot, all the nodes that are equipped with a *traffic generator* can produce a packet according to a random distribution that depends on the type of the implemented applications. In fact, the characterization of traffic has a strong impact on the overall performance of the system. That characterization is determined by the types and number of applications using the system since every application has its own requirements in terms of packets exchange rate. So far, the range of applications that can be enabled by a RF-mesh system is limited due to severe bandwidth constraints limiting the throughput and the simultaneous transmissions of the multitude of nodes that leads to many collisions, and consequently retransmissions, causing high packet transmission delays. Nevertheless, for those applications that do not require very low delay, or whose bandwidth constraints are not so strict, such as meter-reading, demand-response, and load-management, the RF-mesh system is a good choice.

Let us denote by $\Omega = \{\alpha, \beta, \gamma\}$ the set of three different traffic types in the communication system under study. In this case, we have considered meter-reading, demand and broadcast traffic.

- α (*meter-reading*) consists in the random transmission of a packet from a smart meter to its associated collector. This event is Poisson-distributed with parameter λ_u.

β (*demand*) consists in the random transmission of packets from a data collector to one of its controlled meters. The generation of a packet is also Poisson-distributed but with a different parameter λ_d.

γ (*broadcast*) consists, in general, in the transmission of the same packet from the data collector to all smart meters. Although system privacy requirements prevent broadcast transmissions, which are replaced by unicast packets, in the rest of this paper we will refer to γ as a *broadcast* transmission.

Note that α flows in the uplink direction, while β and γ take place on the downlink. The coexistence of different kinds of applications is key for the success of a smart grid communication system: smart grid is constantly enlarging its application domain and a unique communication system would be convenient from both an economic and feasibility perspective.

4.2.2 The collision check module

When a given node i is attempting a transmission to one of its neighbors $j \in \nu(i)$, possible interferers of the transmission $i \rightarrow j$ are all the nodes $k \in \nu(j) : k \neq i, j$ who transmit on the same frequency channel as i. In order to identify the collisions, the dedicated *collision-check* module was implemented. At each time slot, a special check is done to see if nodes that are transmitting are interfering with simultaneous transmissions. Each transmission is identified by a boolean *collision-flag* variable, whose default value is *false*. When a packet collides, its *collision-flag* variable is set to *true* and it is scheduled for retransmission at the following time-slot.

4.2.3 The Packet-forwarding module

At the end of each time slot, after the *traffic generator* module has decided which node is transmitting and after the *collision-check* module has determined the presence of collision, it is necessary to update the Tx and Rx buffers. For this, we implemented the *Packet-forwarding* module, that, for each transmission from node i to node j, checks if a collision has been reported. Then, in case of collision, the packet stays in the Tx buffer of node i and a new transmission will be retried in one of the following time slots with probability p_r, as described at the end of Section 3. In case no collision was observed, the packet is deleted from the Tx buffer of node i and forwarded to the Rx buffer of node j. Then the *packet-analyzer* of node j verifies the destination of the packet: if the destination is j the packet is considered successfully received and packet statistics are updated; if j is not the destination the packet will passed to the *routing* module, in order to determine the next hop, and then written to the Tx buffer of node j.

5 Numerical results

To illustrate the interest and versatility of the proposed framework, we present in this section a set of results that were obtained by analyzing collision probability, delay and activity time in a realistic setting built from publicly available data.

5.1 Input parameters

Villeray, used for our tests, is a neighborhood within the city of Montreal. It was one of the three areas chosen for a pilot project of smart meter installation in Québec [16]. The starting topology was created with the topology generator described in details in Section 4.1 and the position of routers and collectors were extracted from [16]. The GPS coordinates of smart meters, not provided in the previous document, were extracted by a dataset[3] containing all the residential addresses in the city of Montreal. The topology is composed of 2 data collectors, 16 routers, and 6033 smart meters over an area of $2.7km^2$. In our simulations, we created 20 traffic scenarios with $1/\lambda_u = 1, 0.5, 0.25, 0.125$ hour and $1/\lambda_d = 4, 3, 2, 1, 0.5$ hour and no broadcast transmission. The effect of application γ on the performance were studied in 20 additional traffic scenarios with the same values of $1/\lambda_u$ and $1/\lambda_d$ but adding one broadcast transmission per day at a predetermined time slot. Relevant parameters that affect the simulation results are τ (i.e., the time slot duration), C_r and

[3]Provided by *Portail Données Ouvertes Montréal*, http://donnees.ville.montreal.qc.ca/dataset/adresses-ponctuelles

C_m (i.e., respectively the capacity of links between routers/collectors, and the capacity of all other links), L (i.e., the packet size in Bytes), and Θ (i.e., the buffer size expressed in number of packets). The average number of packets that can be simultaneously transmitted on the links between routers/collectors can be found as $\phi_r = \frac{C_r \cdot \tau}{L}$, and on the other links it is $\phi_m = \frac{C_m \cdot \tau}{L}$. For instance, with $C_m = 9.6kbps$, $C_r = 19.2kbps$, $\tau = 0.7s$ and $L = 100Bytes = 800bits$, we have $\phi_r = 16.8$ packets and $\phi_m = 8.4$ packets.

Additional notation, used in the rest of the performance analysis, is reported below:

$\nu(i)$	number of neighbors of node $i \in V$
$\xi(i)$	number of shortest path containing node $i \in V$
$N_t(i)$	total number of transmissions of node $i \in V$
N_t	overall number of transmissions
$N_c(i)$	total number of collisions experienced by node $i \in V$
N_c	total number of collisions by all the nodes $i \in V$
$\pi(i)$	collision probability at node i
π	overall collision probability

5.2 Collisions count

One of the elements that most affect the performance of a wireless network is the mutual interference of transmitting nodes, since they all use the same communication medium. FHSS protocol mitigates the effects of wireless interference by increasing the number of communication channels and consequently reducing the chances of collisions, as shown in [13, 14]. However, collisions are not completely eliminated because of the high number of neighbors, and consequently interferers, each wireless node has in RF-mesh AMI.

Numerical results are summarized in Table 1 for the 20 simulations run with $T_s = 123400$ time slots (roughly 24 hours). In the Table, values for N_t, N_c and π are reported according to the different values of λ_u (columns) and λ_d (rows) used in the simulations. Note that the average collisions probability ranges from 1.05% to 8.78%.

A separate analysis was carried out with the same levels of traffic but introducing broadcast demand γ. Numerical results for these simulations are also included in the lower part of Table 1. It is important to remark that the maximum value of the average collision probability is 8.36%, meaning that the impact of the introduction of broadcast traffic on the overall performance is not very high.

Collision probabilities in those simulations resulted to be slightly higher, on average, than in the previous set. The impact of γ on the performance does not appear to be remarkable, but its effect will become more evident in the delay analysis reported in the Section 5.3.

One of the 20 simulated scenarios (i.e., with $1/\lambda_d = 0.5$ hour and $1/\lambda_u = 0.25$ hour) has been separately studied in the rest of this section. The tool simulated a 24-hour period in 686.4 s with 500737generated (and

Table 1: Average collision probability according to different levels of $1/\lambda_u$ (rows) and $1/\lambda_d$ (columns).

$1/\lambda_d$	$1/\lambda_u$ (hours)												
	1			0.5			0.25			0.125			
(hours)	N_t	N_c	π	N_t	N_c	π	N_t	N_c	π	N_t	N_c	π	
4.0	464213	4887	**1.05E-2**	841563	13975	1.66E-2	1593447	49346	3.10E-2	3139517	204795	6.52E-2	α, β
3.0	498748	5947	1.19E-2	873846	16395	1.88E-2	1639268	59597	3.64E-2	3148433	170895	5.43E-2	
2.0	560698	7272	1.30E-2	940905	18234	1.94E-2	1706462	59395	3.48E-2	3222174	188071	5.84E-2	
1.0	750225	13371	1.78E-2	1128592	29458	2.61E-2	1901922	79180	4.16E-2	3450549	236048	6.84E-2	
0.5	1034249	28152	2.72E-2	1418194	47769	3.37E-2	2191815	106797	4.87E-2	3770543	330992	**8.78E-2**	
4.0	478006	7082	1.48E-2	860996	16963	1.97E-2	1618638	57341	3.54E-2	3138379	184388	5.88E-2	α, β, γ
3.0	508542	7245	1.42E-2	893388	18639	2.09E-2	1655539	57932	3.50E-2	3178781	199532	6.28E-2	
2.0	573215	9280	1.62E-2	953334	22233	2.33E-2	1719065	62465	3.63E-2	3268841	221671	6.78E-2	
1.0	763359	15297	2.00E-2	1139845	26392	2.32E-2	1922830	85055	4.42E-2	3444298	224795	6.53E-2	
0.5	1045155	29598	2.83E-2	1429997	52070	3.64E-2	2206510	116032	5.26E-2	3771909	315159	**8.36E-2**	

successfully received) packets, 2206510 transmissions and 116032 collisions, with an average network collision percentage of 5.26%.

A more accurate portray of the dynamics of the network is shown in Figure 4: a *heat-map* of all the nodes is reported, in which the collision percentages of nodes are grouped into 5 intervals corresponding to 5 different colors in the gray scale color-bar on the right. The ranges are not homogeneous in order to better represent the differences among nodes in certain areas. The perimeter of the area was also drawn in order to highlight the area analyzed in the simulation. As we can see, even though the average collision percentage of all nodes stays low even with high level of traffic, in the surroundings of the 2 data collectors the situation is considerably different. A high number of collisions is experienced and very high collision percentage are observed. The reason behind the increased number of collisions is to be found in the higher concentration of packet transmissions, in both uplink and downlink directions, in that area.

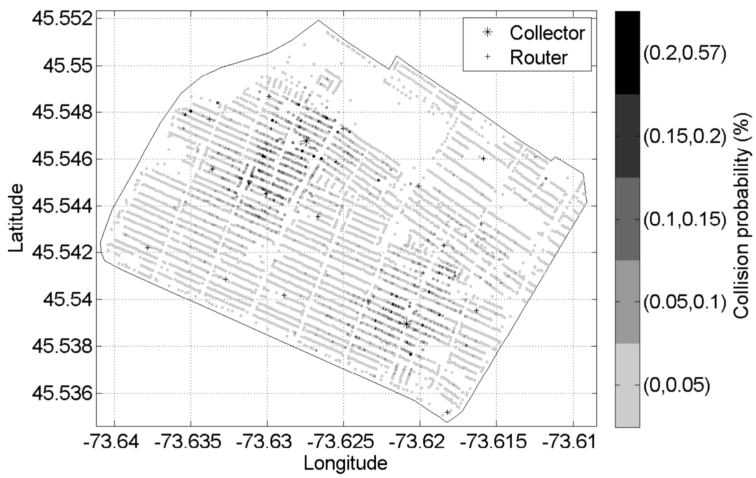


Figure 4: Heat-map of the collision probability in the scenario with $1/\lambda_d = 0.5$ h and $1/\lambda_u = 0.25$.

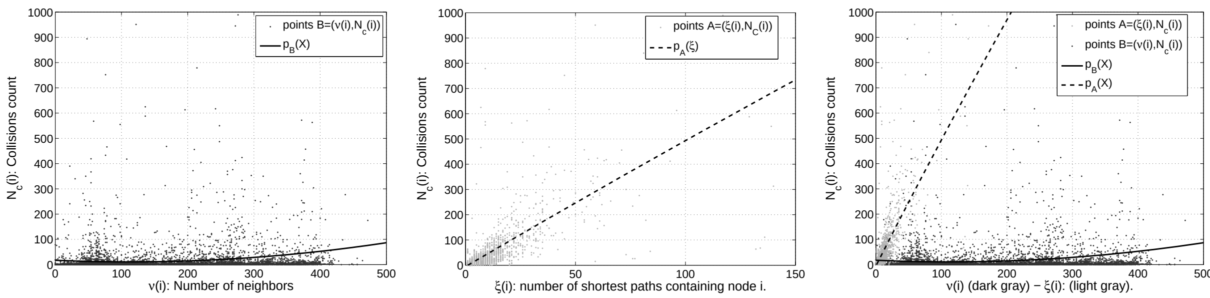
Moreover, a separate analysis was performed with the scope of investigating the effect of the two previously defined parameters $\nu(i)$ and $\xi(i)$ on the number of collisions per node $N_c(i)$. In Figure 5a, we can find a scatter plot of the points A_i, defined as $(\nu(i), N_c(i))$. In other words, each point corresponds to a node in the system and, in particular, its x and y components respectively amount to the number of neighbors $\nu(i)$ and to the count of collisions $N_c(i)$. The aforementioned picture shows that the collision count is quasi-uniformly spread over the range of distribution of $\nu(i)$. Furthermore, a regression analysis was performed and we found that the polynomial of 2^{nd} degree that best fitted the scatter plot was

$$p_A(X) = a_0X^2 + a_1X + a_2 = 0.0005X^2 - 0.1X + 17.3$$

The polynomial curve was then drawn in Figure 5a: it shows how mildly $N_c(i)$ rises with respect to $\nu(i)$. The number of neighbors does not seems a major cause of collisions in the wireless system under study. On the other hand, in Figure 5b, we investigated the impact of the parameter $\xi(i)$ on the number of collisions $N_c(i)$. A scatter plot is shown containing the points $B(i) = (\xi(i), N_c(i))$. The points appear to be distributed in a much more restricted interval. As in the previous case, a regression analysis of the two variables $\xi(i)$ and $N_c(i)$ was performed:

$$p_B(X) = b_0X^2 + b_1X + b_2 = -0.001X^2 + 5.1X - 5.3$$

Moreover, the $p_B(X)$ is reported in Figure 5b (dashed line) highlighting a steep increase of the collision count in nodes that are included in a higher number of shortest paths (i.e., the $\xi(i)$ parameter). The comparison with the previous case (i.e., Figure 5a) appears even more clear in Figure 5c: the two series of points A_i and B_i are reported in the same graph and we can see how much steeper the B_i regression curve (dashed line) is with respect to the A_i regression line. The simulation results showed the routing parameter $\xi(i)$ to have a stronger impact on the collisions, and consequently on the performance, with respect to the topological parameter $\nu(i)$.



(a) (b) (c)

Figure 5: Scatter plots of $(\nu(i), N_c(i))$ in 5a, of $(\xi(i), N_c(i))$ in 5b, and comparison of the two in 5c.

5.3 Delay analysis

Delay is one of the most important parameters in network performance. It is calculated in the results analysis phase (see Section 4 for details about the structure of the simulator). If we denote by $T_d(i)$ the generation time slot of packet i and with $T_a(i)$ the arrival time slot at its destination we can generally define the delay $D(i)$, encountered by packet i, throughout the network as:

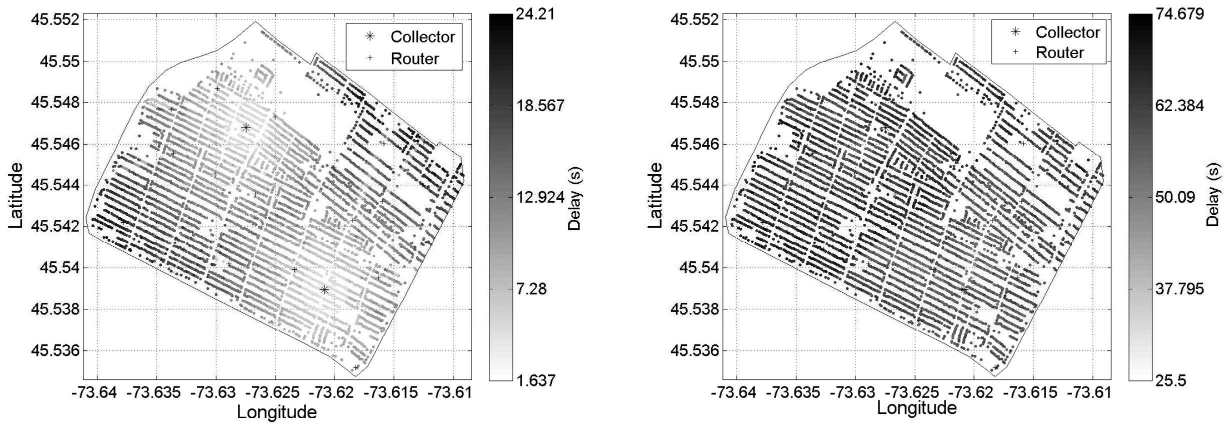
$$D(i) = T_a(i) - T_d(i) + 1 \tag{2}$$

It is also important to separately study the average delay for the three defined traffic applications: $D_\alpha(i)$ is the average delay of α-packets from node $i \in M$, $D_\beta(i)$ the average delay of β-packets directed to node $i \in M$, whereas D_α, D_β, and D_γ are the average delay of all the packets for respectively α, β, and γ. As for the collision count, we started with an analysis of traffic scenario with applications α and β. In the upper part of Table 2, we reported the values of D_α and D_β (no broadcast packet is transmitted so D_γ is not applicable) according to the previously defined ranges of variation of $1/\lambda_d$ and $1/\lambda_u$. As one can see, the delay in downlink slightly decreases as the packet generation rate increases but in the downlink direction higher delays are experienced. In particular, we can notice that when the mean packet generation time goes below 1 hour there is a steep increase of the average downlink delay, with values ranging from 31.95 s to 44.82 s.

In Figure 6, we reported the two heat maps of the delays (uplink on the left and downlink on the right) in the scenario with $1/\lambda_d = 0.5$ h and $1/\lambda_u = 0.25$ h. Note that we can see in the figures that the average delay in the downlink direction is sensitively higher than in the uplink (i.e., $D_\beta = 6.74$ s, $D_\alpha = 35.74$ s). The fact proved that the packet generation rate in the downlink has an higher impact on the delay, with respect to its uplink counterpart. Another remarkable element is that downlink delay resulted to be more homogeneous

Table 2: Average delay (expressed in seconds) for the different type of traffic.

	$1/\lambda_u$ (hours)												
$1/\lambda_d$	1			0.5			0.25			0.125			
(hours)	D_β	D_α	D_γ	D_β	D_α	D_γ	D_β	D_α	D_γ	D_β	D_α	D_γ	
4.0	6.61	6.42	-	6.71	6.45	-	6.93	6.55	-	7.31	6.80	-	
3.0	6.77	6.44	-	6.81	6.47	-	6.97	6.58	-	7.36	6.72	-	
2.0	7.00	6.41	-	7.07	6.50	-	7.32	6.60	-	7.66	6.76	-	α, β
1.0	8.57	6.49	-	8.80	6.52	-	9.09	6.65	-	10.10	6.85	-	
0.5	**31.95**	6.61	-	33.04	6.65	-	35.74	6.74	-	**44.82**	7.04	-	
4.0	10.30	**6.43**	431.36	10.97	6.46	442.92	11.56	6.59	460.20	12.05	6.75	470.03	
3.0	11.59	6.43	458.12	11.26	6.50	438.41	11.67	6.59	458.58	12.27	6.78	476.40	
2.0	11.61	6.44	438.59	11.12	6.48	441.44	11.61	6.59	446.74	12.67	6.83	471.01	α, β, γ
1.0	12.98	6.53	437.79	13.69	6.52	450.29	14.50	6.66	455.62	15.27	6.81	448.57	
0.5	**37.27**	6.61	445.19	41.21	6.65	448.51	42.91	6.77	464.43	**48.96**	**7.01**	481.60	



(a) uplink (b) downlink

Figure 6: Heat-map of the delay in the scenario with $1/\lambda_d = 0.5$ h and $1/\lambda_u = 0.25$.

than uplink delay. Besides the visual acknowledge of the behavior from the maps of Figure 6, we defined and calculated the delay variation coefficients σ_u^* and σ_d^*, for respectively the uplink and the downlink, as the ratio between the standard deviation and the average, obtaining $\sigma_u^* = 0.495$ and $\sigma_d^* = 0.140$. The reason behind the higher homogeneity of the downlink direction can be found on the enhanced link congestion in the surroundings of the data collector. Nodes on these areas suffer the proximity to the collector and the limited buffer size that causes additional waiting times to the packet transmission.

Continuing the analysis, we investigated the impact of broadcast transmissions on D_α, D_β, and D_γ. In the lower part of Table 2, we reported the average delays for the three analyzed applications in a scenario with $1/\lambda_d = 0.5$ h and $1/\lambda_u = 0.25$ h and the simultaneous transmission of a packet to each and every smart meter. We can notice that the uplink communications are not sensitively affected by the introduction of a broadcast transmission: the variation interval of $D(\alpha)$ spans from 6.43 s to 7.01 s. On the other hand, a small increase in D_β is observed: suffice it to compare its average values in the case with broadcast (i.e., from 37.27 s to 48.96 s) to the corresponding average values without the broadcast (i.e., from 31.95 s to 44.82 s). D_γ resulted to be considerably higher than D_α and D_β.

5.4 Other performance parameters

Another sensitive parameter in the context of smart meter performance analysis is the activity time χ, which represents the fraction of the simulated time in which each node is actually transmitting a packet. The importance of this parameter is driven by the recent concerns about the electromagnetic impact on public health, issued by the massive installation of smart meters in residential premises. χ can be easily calculated as:

$$\chi(i) = \frac{\sum\limits_{\forall i \in V} N_t(i)}{T_s \cdot |V|} \tag{3}$$

Since $\chi(i)$ takes into account all the nodes in the network, and since the main concerns are related to the smart meters, we decided to define 3 additional parameters, χ_m, χ_r, and χ_c, representing the average activity time of respectively smart meters, routers, and data collectors. Numerical values of χ_m, χ_r and χ_c are reported in Table 3. We can notice that, as expected, the activity time increases as the packet generation rates increase: we observed that the variation of the activity time with respect to the variation of $1/\lambda_d$ is much more evident than to the variation of $1/\lambda_u$. However, numerical values of χ_m are always lower than 0.5%. This datum is interesting because it can be used to address and respond, as previously mentioned, to the public concerns about an excessive human exposure to wireless waves produced by smart meters. In fact, our results show that this kind of devices were in transmission mode for only a very low percentage

Table 3: Average activity times for smart meters (χ_m), routers (χ_r) and data collectors (χ_c) for the different type of traffic according to different levels of $1/\lambda_u$ (rows) and $1/\lambda_d$ (columns) without broadcast style traffic.

$1/\lambda_d$	$1/\lambda_u$ (hours)											
	1			0.5			0.25			0.125		
(hours)	χ_m	χ_r	χ_c	χ_m	χ_r	χ_c	χ_m	χ_r	χ_c	χ_m	χ_r	χ_c
4.0	**5.92E-4**	1.72E-3	8.17E-2	1.09E-3	3.11E-3	8.36E-2	2.10E-3	5.97E-3	8.56E-2	4.15E-3	1.21E-2	8.95E-2
3.0	6.28E-4	1.86E-3	1.11E-1	1.13E-3	3.29E-3	1.11E-1	2.15E-3	6.19E-3	1.14E-1	4.16E-3	1.19E-2	1.18E-1
2.0	6.92E-4	2.13E-3	1.67E-1	1.20E-3	3.42E-3	1.70E-1	2.22E-3	6.42E-3	1.75E-1	4.23E-3	1.23E-2	1.81E-1
1.0	8.89E-4	2.89E-3	3.34E-1	1.39E-3	4.31E-3	3.37E-1	2.42E-3	7.15E-3	3.44E-1	4.48E-3	1.33E-2	3.61E-1
0.5	1.21E-3	4.01E-3	4.99E-1	1.72E-3	5.49E-3	4.98E-1	2.76E-3	8.42E-3	5.00E-1	**4.86E-3**	1.46E-2	**4.98E-1**

of time, much lower than many others everyday wireless devices (e.g., smart-phones, and cordless phones). Particularly interesting is also the mean activity time of data collectors: in the scenario with $1/\lambda_u = 0.25$ hour and $1/\lambda_d = 0.5$ hour, the data collectors are active for nearly 50% of the time.

5.5 Computational time

RF-mesh systems are usually large scale systems with thousands of node and the computational burden can be a serious impairment to be carefully taken into account. In our case, however, the average time to perform a simulation of 6051-node instances over a simulated time of 24 hours felt in the interval between 11 and 12 minutes (i.e., 711 seconds), obtained on a machine with a processor *AMD A8-4500M APU* working at 1.90 GHz. This is a very important result because it proves the capacity to simulate large-scale systems in a considerably short time. It is important to remark that the elapsed time of 711 seconds amounts to the 0.8% of the total simulated time of 86400 seconds: on average, 8 ms are necessary to simulate 1 second of operation of the network.

6 Conclusion

As smart grids proliferate around the world and AMI systems are widely deployed, more and more applications will likely be introduced. Therefore, it is essential for the power utility to have the appropriate tools to be able to assess the performance of the system and, therefore, the impact of the applications on their system. In this paper, we have presented a framework to assess the performance of AMI systems that removes some of the roadblocks, such as scalability or adaptability, that could be found in previous work related to AMI performance.

The framework yielded an effective simulation tool that provides great flexibility in the selection of system features (e.g., the area under consideration, the number of nodes). Moreover, the implemented topology generation tool grants the possibility to create customized instances, and can be used to both choosing manually the location of routers and collectors, and in combination with an optimization tool. The observed computational efficiency permits the analysis of instances of thousands of nodes in a very reasonable time.

The numerical results used to illustrate the framework flexibility have nonetheless brought to light some insights with respect to the AMI setting in a densely populated NAN. There are difficulties in the handling of broadcast traffic, and there is a performance degradation in the surroundings of the collectors, where multiple traffic strings coexist and mutually disturb each other. Furthermore, the uplink delay is not particularly affected by the broadcast traffic, and that the choice of routing affects the collision probability more than the number of neighbors. It would be interesting to compare those findings with other types of AMI deployment.

References

[1] D. Chen, J. Brown, and J. Y. Khan, 6LoWPAN based neighborhood area network for a smart grid communication infrastructure, in Proc. 5th International Conference on Ubiquitous and Future Networks (ICUFN 2013), 2013, 576–581.

[2] D. Chen, J. Brown, and J. Y. Khan, Performance analysis of a distributed 6LoWPAN network for the smart grid applications, in Proc. 9th IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing (IEEE ISSNIP 2014), 2014.

[3] M. Islam, M. M. Uddin, M. A. A. Mamun, and M. A. Kader, Performance analysis of AMI distributed area network using WiMAX technology, in Proc. 9th International Forum on Strategic Technology (IFOST 2014), 2014.

[4] D. Bian, M. Kuzlu, M. Pipattanasomporn, and S. Rahman, Analysis of communication schemes for advanced metering infrastructure (AMI), in Proc. 2014 IEEE PES General Meeting. IEEE, 2014, 5.

[5] A. Patel, J. Aparicio, N. Tas, M. Loiacono, and J. Rosca, Assessing communications technology options for smart grid applications, in Proc. IEEE International Conference on Smart Grid Communications (SmartGridComm 2011). IEEE, 2011, 126–131.

[6] S. Panchadcharam, G. A. Taylor, Q. Ni, I. Pisica, and S. Fateri, Performance evaluation of smart metering infrastructure using simulation tool, in Proc. 47th International Universities Power Engineering Conference (UPEC 2012), 2012, 1–6.

[7] A. AlMajali, A. Viswanathan, and C. Neuman, Analyzing resiliency of the smart grid communication architectures under cyber attack, in Proc. 5th Workshop on Cyber Security Experimentation and Test, 2012.

[8] M. Yigit, E. A. Yoney, and V. C. Gungor, Performance of mac protocols for wireless sensor networks in harsh smart grid environment, in Proc. 1st International Black Sea Conference on Communications and Networking (BlackSeaCom 2013), July 2013, 50–53.

[9] F. Aalamifar, L. Lampe, S. Bavarian, and E. Crozier, WiMAX technology in smart distribution networks: Architecture, modeling, and applications, in Proc. IEEE PES Transmission and Distribution Conference and Exposition, 2014.

[10] G. Iyer, P. Agrawal, and R. S. Cardozo, Performance comparison of routing protocols over smart utility networks: A simulation study, in Proc. 5th IEEE International Workshop on Management of Emerging Networks and Services (IEEE Globecom Workshops), December 2013, 969–73.

[11] B. Lichtensteiger, B. Bjelajac, C. Muller, and C. Wietfeld, RF-mesh systems for smart metering: system architecture and performance, in Proc. First IEEE International Conference on Smart Grid Communications (SmartGridComm 2010), 2010, 379–384.

[12] G. Iyer, P. Agrawal, E. Monnerie, and R. Cardozo, Performance analysis of wireless mesh routing protocols for smart utility networks, in Proc. IEEE International Conference on Smart Grid Communications (SmartGridComm 2011), Oct 2011, 114–119.

[13] F. Malandra and B. Sansò, Analytical performance analysis of a large-scale rf-mesh smart meter communication system, in Proc. IEEE Innovative Smart Grid Technologies Conference (ISGT 2015), Feb 2015.

[14] F. Malandra and B. Sansò, PeRF-Mesh: a performance analysis tool for large scale RF-mesh-based smart meter networks with FHSS, in Proc. IEEE International Conference on Smart Grid Communications (SmartGridComm 2015), Miami, USA, Nov. 2015.

[15] X. Chen, Z. Dai, W. Li, and H. Shi, A layer-based routing protocol for heterogeneous wireless sensor networks, in Proc. IEEE International Conference on Communications (ICC 2012), June 2012, 228–232.

[16] Hydroquébec, Remote meter-reading project phase 1, online, Régie de l'énergie du Québec, May 2012, http://publicsde.regie-energie.qc.ca/projets/34/DocPrj/R-3770-2011-B-0157-TRAD-DOC-2012_05_23.pdf.