

České vysoké učení technické v Praze
Fakulta elektrotechnická



Diplomová práce

Simulation of Wireless Communication Networks

Martin Selecký

Vedoucí práce: Ing. Milan Rollo

Studijní program: Elektrotechnika a informatika strukturovaný
magisterský

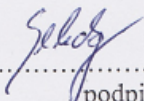
Obor: Kybernetika a měření

Leden 2010

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

V Praze dne*21.12.2009*.....

..........
podpis

Acknowledgements

I would like to thank all the people who helped me in any way with this master thesis. Especially, I would like to thank to my supervisor Ing. Milan Rollo and to Ing. David Šišlák for their advice, helpful ideas, and their time.

Abstrakt

Cílem této práce je navrhnout a implementovat simulátor bezdrátových sítí, který bude použit v multi-agentní platformě A-globe pro simulaci bezdrátové komunikace. Tento simulátor by měl umožnit specifikovat přenosové výkony, citlivosti přijímačů a zesílení antén u simulovaných entit a dále také definovat překážky neprostupné pro rádiový signál - buď manuálně uživatelem, nebo pomocí extrakce z výškové mapy simulovaného prostředí.

V této práci se budu zabývat problematikou přenosu rádiového signálu a simulace bezdrátových sítí, přiblížím základní prvky multi-agentní platformy A-globe a popíšu návrh a vlastnosti nového simulátoru.

Vlastnosti navrženého simulátoru byly testovány a porovnány s vlastnostmi open-source simulátoru SWANS++ a dosahovaly velmi podobných výsledků.

Abstract

The aim of this work is to design and implement a new wireless network simulator that will be used in multi-agent platform A-globe for simulation of wireless communication. This simulator should enable specification of transmission powers, sensitivity thresholds and antenna gains of the simulated entities. Furthermore, it should enable definition of obstacles for radio signal - either manually by the user or by extracting the obstacles' positions from provided height-map of the environment.

In this work, I will deal with the problems of radio frequency signal propagation and wireless network simulations, I will give an overview of the main concepts of the multi-agent platform A-globe, and I will describe the design and features of the new simulator. The accuracy of the simulator was tested and compared to open-source simulator SWANS++, giving very similar results.

DIPLOMA THESIS ASSIGNMENT

Student: Bc. Martin Selecký
Study programme: Electrical Engineering and Information Technology
Specialisation: Cybernetics and Measurement – Artificial Intelligence
Title of Diploma Thesis: Simulation of Wireless Communication Networks

Guidelines:

1. Investigate existing discrete event simulators of wireless communication networks.
2. Discuss their basic principles and their applicability for real-time simulations.
3. Study the architecture of communication transport layer in the A-Globe agent platform developed at the Department of Cybernetics.
4. Design a wireless network simulator suitable for real-time simulations. This simulator should take into account path loss model, model of the environment, radio signal strength and interference among transmitters.
5. Implement this simulator as a plug-in module into the A-Globe platform to allow scalable simulations of real-life scenarios featuring wireless communication among scenario actors.
6. Carry out a set of experiments to compare macro-level behavior of your module with another wireless network simulator like e.g. SWANS++ or ns2.

Bibliography/Sources: Will be provided by the supervisor.

Diploma Thesis Supervisor: Ing. Milan Rollo

Valid until: the end of the winter semester of academic year 2010/2011



prof. Ing. Vladimír Mařík, CSc.
Head of Department



doc. Ing. Boris Šimák, CSc.
Head

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: Bc. Martin Selecký

Studijní program: Elektrotechnika a informatika (magisterský), strukturovaný

Obor: Kybernetika a měření, blok KM2 – Umělá inteligence

Název tématu: Simulace bezdrátových komunikačních sítí

Pokyny pro vypracování:

1. Seznamte se se stávajícími simulátory bezdrátových komunikačních sítí založených na diskrétních událostech.
2. Popište jejich základní principy a možnost jejich využití v simulátorech pracujících v reálném čase.
3. Prostudujte architekturu komunikační vrstvy multiagentní platformy A-Globe vyvinuté na katedře kybernetiky ČVUT.
4. Navrhněte simulátor sítí vhodný pro simulátory pracující v reálném čase. Tento simulátor by měl pracovat s modelem šíření radiového signálu, modelem prostředí, silou signálu a vzájemným rušením mezi jednotlivými jednotkami.
5. Naimplementujte tento simulátor jako plug-in modul do platformy A-Globe tak, aby umožnil škálovatelné simulace reálných scénářů zahrnujících bezdrátovou komunikaci mezi entitami.
6. Experimentálně porovnejte vlastnosti navrženého modulu s jiným existujícím simulátorem (např. SWANS++ nebo ns2) na makro úrovni.

Seznam odborné literatury: Dodá vedoucí práce.

Vedoucí diplomové práce: Ing. Milan Rollo

Platnost zadání: do konce zimního semestru 2010/2011



prof. Ing. Vladimír Mařík, CSc.
vedoucí katedry



doc. Ing. Boris Šimák, CSc.
děkan

Table of Contents

List of figures	xi
List of tables	xiii
1 Introduction	1
1.1 Wireless Network Communication	1
1.2 Motivation	2
1.3 Contribution	2
1.4 Thesis Organization	3
2 Problem Formulation	4
2.1 Radio Frequency Signal Propagation	4
2.1.1 Scattering	5
2.1.2 Diffraction	6
2.1.3 Refraction	7
2.1.4 Reflection	7
2.2 Path Loss Models	8
2.2.1 The free space propagation model	8
2.2.2 The two-ray ground model	8
2.2.3 The log-distance path loss model	9
2.2.4 Probabilistic propagation models	9
2.3 Interference	10
3 IEEE 802.11 Standard	12
3.1 Physical Layer	12
3.1.1 Maximum Information Rate	12
3.1.2 Operation Modes	13
3.2 Medium-Access Control Layer	13
3.3 Protocols	15
4 Wireless Network Simulators	19
4.1 The Accuracy Of Wireless Network Simulators	19
4.1.1 Level of details	19
4.1.2 Mobility models	20
4.1.3 Radio propagation models	20
4.2 An overview of the most frequent simulators	21
4.3 The event-driven simulation versus real-time simulation	24

5	A-globe Agent Platform	25
5.1	About the platform	25
5.2	A-globe Architecture	26
5.3	Message Transport	28
5.3.1	Message Hooks	29
5.4	Scenario Support in A-globe	29
5.4.1	Container Reachability	29
6	Module Design	32
6.1	Server-Client Architecture	33
6.1.1	Server side	33
6.1.2	Client side	34
6.2	Station Reachability	35
6.2.1	Signal strength	36
6.2.2	Obstacles	37
6.3	Message Transport Simulation	37
6.3.1	Physical layer simulation	39
6.3.2	MAC layer	40
6.3.3	Interference and retransmission of messages	41
7	Experiments	44
7.1	WLAN Simulation Characteristics	44
7.1.1	Experiment's testbed description	44
7.1.2	Results	45
7.1.2.1	Message delivery delay	45
7.1.2.2	Ratio of the lost messages	46
7.1.2.3	Network throughput	47
7.2	Computational Resource Requirements	47
8	Conclusions	50
9	Bibliography	51
A	Content of the enclosed CD	I

List of Figures

2.1	Fresnel zone between transmitter and receiver, (borrowed from http://www.danets.com/wifi-images)	5
2.2	Example of particle scattering - elastic two particle collision (borrowed from http://www.scitopics.com)	6
2.3	Diffraction of a plane wave (a) from a wide slit, (b) from a wide obstacle, (c) from a very narrow slit (much narrower than the signal's wavelength), (d) from a slit with width comparable to the signal's wavelength. (borrowed from http://cs.wikipedia.org/wiki/Difrakce)	6
2.4	Wave refraction where $n_2 > n_1$, (resp. $v_1 > v_2$). (borrowed from http://en.wikipedia.org/wiki/Refraction)	7
2.5	The two-ray propagation model - the signal from transmitter reaches receiver through the direct path, and through a ground reflected path.	9
2.6	Problems of wireless communication (a) node 4 as a hidden node to node 1, (b) node 3 as an exposed node by node 1	11
3.1	Wireless mode (a) with base station (access point) (b) without base station (ad hoc mode).	13
3.2	Virtual channel sensing when A is sending a frame to B - from topological point of view (a), and from the view of the channel allocation (b).	14
5.1	Some of projects employing the A-globe. (a) NAIMT scenario, (b) AGENTFLY scenario	25
5.2	System architecture	26
5.3	Agent container architecture	27
5.4	Message flow	28
5.5	A-globe simulation structure	30
5.6	Reachability information propagation	30
6.1	Simulation state diagram	32
6.2	System architecture	33
6.3	The mechanism of the message transport in <i>aglobex.wireless</i>	34
6.4	The ranges used in wireless simulation	35
6.5	Examples of opaque obstacles: (a) user defined zone, (b) obstacles given by a height map	38
6.6	Basic scheme of the message transport simulation	39
6.7	Example of exponential increase of CW, borrowed from [21]	42
7.1	Nodes' placement used for the testing of simulator's characteristics	45

7.2	Average delivery delays of messages in milliseconds	46
7.3	Ratio of lost messages	47
7.4	Network throughput in individual simulators in <i>kbps</i>	48
7.5	Comparison of the CPU time (a) and memory requirements (b) with visualisation enabled	48
7.6	Comparison of the CPU time (a) and memory requirements (b) with visualisation disabled	49

List of Tables

- 2.1 Values of the distance-power gradient in different environments (borrowed from [39]) 10
- 3.1 Overview of IEEE 802.11 standards. 16
- 4.1 Granularity of simulators, programming language they are based on and licences they use. Data for the table were taken from [19] 20

1 Introduction

1.1 Wireless Network Communication

In general, wireless networking refers to the use of radio or infrared frequency signals to transfer information or share resources between devices. The importance of wireless networks grows with rising numbers of laptops, netbooks, Wi-Fi cell phones, handheld PCs, PDAs and other devices that employ wireless technology. They also have wide utilization in wireless sensor networks which are spatially distributed autonomous sensors used for cooperative monitoring of physical or environmental conditions. Also, recently the scientific and industrial community in the area of telecommunications deeply investigates a promising area of noninfrastructure-based wireless networks - Mobile Ad-hoc Networks (MANETs), a self-configuring wireless network of mobile devices - which will play an important role in the near future of communication.

There are some major differences in characteristics showed by wireless networks when compared to the wired ones. The most important ones are

- Higher interference - signal can interfere with itself because of multipath propagation, devices can interfere with each other because of broadcast nature of the transmissions, radio signals can interfere with other electrical devices and infrared signals interfere with sunlight and heat sources and are blocked by non-transparent materials.
- Lower transmission rates and bandwidth ability.
- Variable network conditions - devices may move, interference can cause higher data loss rates, signal power attenuates with distance, signal quality is influenced by the weather and so on.
- Limited computing and energy resources when the devices run on battery.

When new protocols or applications for wireless systems are developed, its functionality needs to be tested. Real hardware for testing is unfortunately rather expensive, testing conditions are usually not reproducible and experiment monitoring is difficult. That is why real testbeds are not very suitable for experiments. On the other hand, none of the mentioned problems of real testbeds is applicable to wireless network simulators. There exist numerous network simulators ([33],[42],[3],[26],[29] and others) which vary in attributes like the maximal number of entities they can model, in their granularity, i.e. level of details they use, in their processing power requirements and in other properties. These simulators, however, mostly do not support co-simulation facilities (with some exceptions like TrueTime [1] or Jane simulator [30]) that would simulate the latency-related aspects of the network communication in combination with the node computations and the dynamics of the environment.

1.2 Motivation

The main motivation of this work was to add a facility of wireless communication simulation to multi-agent platform A-globe [23] developed at Agent Technology Centre (ATG) in Prague.

Although there are various wireless network simulators currently in use, none of them is suitable for being employed in the A-globe platform. The reasons are the following:

- Importing of an existing simulator to the A-globe at its message transport level would need complex changes to the simulator's and A-globe's structure.
- Most of the simulators do not allow individual hosts to move at their will, they only utilize randomized mobility models of the hosts' movement which the hosts must follow. In A-globe simulations, however, the host's decision making about its movement is highly required feature.
- While the simulator should model wireless communication in A-globe simulations, it needs to run in real time to enable co-simulation facilities. Most of the existing simulators do not provide these facilities.

1.3 Contribution

In this work I will present a design of a multi-agent wireless network simulator that was projected as a plug-in module (called *aglobex.wireless*) to A-globe platform with the following features:

- Provides model of the physical layer - signal propagation model, path loss model dependent on the environment, estimates signal interference and supports node mobility.
- Supports basic IEEE 802.11 MAC protocols, such as CSMA/CA and virtual carrier sensing.
- Enables importing of a map of the physical environment and its utilization for communication availability.
- Enables co-simulation of network communication and node computations in real-time.
- Module is suited for A-globe message transporting protocols (as described further in chapter 5) and thanks to its agent nature supports parallel computations and enables distribution of the simulation on distinct computers.

1.4 Thesis Organization

This thesis is organized as follows. In chapter 2, I will formulate the problem of radio frequency (RF) signal propagation, describe path loss model of RF signal and discuss effects of signals' interference. In chapter 3, I will describe the IEEE 802.11 standard, its specifications and features and its MAC layer protocol. In chapter 4, I will give an overview of existing wireless network simulators and describe the advantages and disadvantages of simulators working in the real-time domain over more common discrete-event simulators. As I mentioned above, *aglobex.wireless* module is intended to be used in A-globe agent platform. I will give a general overview of this platform and describe its message transport protocols in chapter 5. Then in chapter 6, I will present the actual design of the *aglobex.wireless* module, describe its architecture and used environmental and signal propagation models. Chapter 7 contains results of experiments that examine the scalability of the module and test its simulation accuracy by comparing it to another wireless network simulator, SWANS++ [2]. Finally, chapter 8 concludes this thesis and gives an overview of future work.

2 Problem Formulation

When simulating wireless networks, one of the most substantial tasks is to model the wireless channel. Even though, both radio and infrared frequency signals are used for wireless communication, in the following I will focus on radio frequency signal propagation since it is the most common means of wireless communication nowadays.

Because the radio frequency (RF) signal, as well as light waves, is subject to various physical phenomena like *reflection*, *refraction*, *diffraction*, *scattering*, *attenuation* or *interference*, its modelling is a very complex task. Wireless network simulators usually do not model all the details because firstly, not all of them have such a significant effect on the simulation results, and secondly, modelling all of them would make the simulation very slow and cumbersome. Nevertheless, it is important to know what these phenomena are and how they influence the signal propagation. In this chapter, I will describe RF signal propagation, as well as the physical processes that influence it. Further, I will present some of the most used models of path loss (how signal power is attenuated with distance from receiver) and the principle of signals interference and problems connected with it, as it is very common in wireless networks.

2.1 Radio Frequency Signal Propagation

RF signal is radiated in straight lines but unlike the light, bulk of its energy is carried in the space around the straight line connecting the transmitter and the receiver. This ellipsoidal space is divided into concentric regions called *Fresnel zones* (see Figure 2.1). Radius of n-th zone in some point P can be estimated according to equation 2.1. In the first Fresnel zone, there is carried 90% of signal's energy. If this zone is obstructed by some obstacles, quality of the transmission decreases significantly.

$$F_N = \sqrt{\frac{N\lambda D_1 D_2}{D_1 + D_2}} \quad (2.1)$$

where F_N is n-th zone's radius in point P ,

λ is wavelength,

D_1 and D_2 are distances from transmitter and receiver.

Furthermore, for a succesful data reception, strength of the transmitted signal needs to be greater than the background noise at the receiver. The background noise can be caused either by devices operating in the same frequency band (such as radios, microwave ovens, cordless telephones or Bluetooth devices in the 2.4 GHz band) or it can be caused by the transmitting stations when their signals get corrupted by the environment and become unusable (e.g. when the multi-path propagated signal interferes with itself).

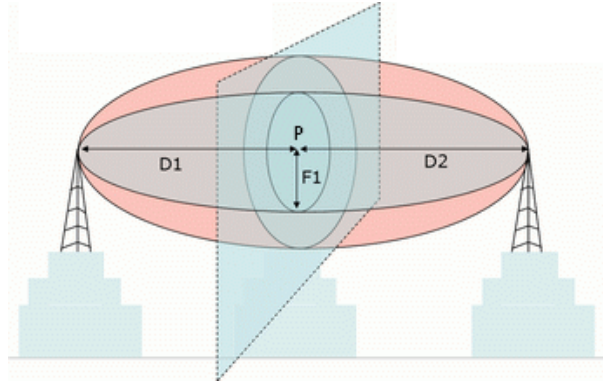


Figure 2.1: Fresnel zone between transmitter and receiver, (borrowed from <http://www.danets.com/wifi-images>)

The difference between the signal's power and the noise power is known as *signal-to-noise-ratio* (SNR) and can be estimated as

$$SNR[dB] = S[dB] - N[dB] \quad (2.2)$$

where S is the signal's power in dB and N is the noise power in dB .

The necessary values of SNR are different for various data rates and modulation techniques. These will be in more detail described in chapter 3.

In this context, I should mention the difference between three different units - dB , dBm and dBi .

- dB is a unit of relative measure of two different power levels. It can be computed as

$$x = 10 \cdot \log \frac{P_1}{P_2} \quad (2.3)$$

- dBm is a unit of a signal power computed according to the equation 2.3 compared to the power of 1 mW .
- dBi is a common reference unit for antennas. It states the gain of an antenna as referenced to an isotropic source (perfect omnidirectional radiator).

As mentioned above, the RF signal transmission is affected by various physical processes, some of which I am about to describe shortly in the following.

2.1.1 Scattering

Scattering is a physical process that causes waves (or moving particles) to deviate from a straight trajectory because of some non-uniformities (or small objects, compared to the signal wavelength) in the medium through which they pass. An example of scattering can be a two particle collision, shown in Figure 2.2.

Non-uniformities which can cause scattering (called *scatterers* or *scattering centers*), are for example, particles, droplets, bubbles, surface roughness, cells in organisms, textile fibers in clothing or foliage and street signs.

Apparently, the level of details needed for modelling the scattering is very high and the final effects on the general RF signal propagation are insignificant.

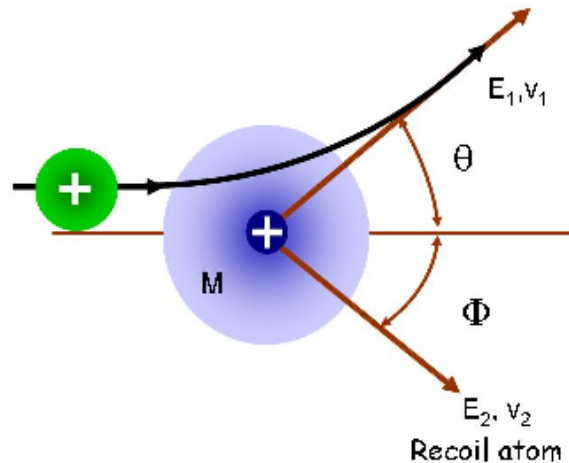


Figure 2.2: Example of particle scattering - elastic two particle collision (borrowed from <http://www.scitopics.com>)

2.1.2 Diffraction

Wave diffraction refers to a phenomenon which occurs when a wave encounters an obstacle. Waves can bend around small obstacles or spread out past small openings. The process is explained by the *Huygens-Fresnel principle* - each point of an advancing wave front is in fact the center of a fresh disturbance and the source of a new train of waves. The effects of wave diffractions are shown in Figure 2.3.

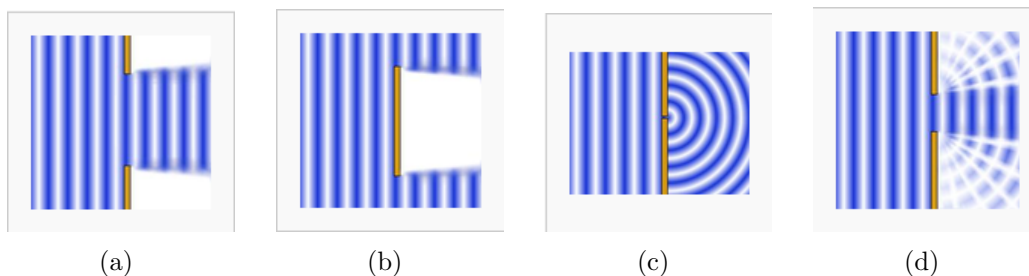


Figure 2.3: Diffraction of a plane wave (a) from a wide slit, (b) from a wide obstacle, (c) from a very narrow slit (much narrower than the signal's wavelength), (d) from a slit with width comparable to the signal's wavelength. (borrowed from <http://cs.wikipedia.org/wiki/Difrakce>)

In wireless radio communication, diffraction is mostly caused by objects with very sharp edges that lie on the radio path between the transmitter and the receiver. It permits wireless communication between points where there is no line-of-sight path due to obstacles that are opaque to radio waves. Its effects can be significant but still, none of present simulators take it into account in its physical layer's model.

2.1.3 Refraction

Change in direction of a wave due to its velocity is referred to as *refraction*. This happens when a wave is passing from one medium to another (see Figure 2.4).

Refraction is described by *Snell's law*, which states that the angle of incidence Θ_1 is related to the angle of refraction Θ_2 by

$$\frac{\sin \Theta_1}{\sin \Theta_2} = \frac{v_1}{v_2} = \frac{n_2}{n_1} \quad (2.4)$$

where v_1 and v_2 are the wave velocities in the respective media, and n_1 and n_2 are the *refractive indices*.

Modelling of refraction could be meaningful in environments with significantly different media through which radio signal would pass. However, none of the current wireless simulators implements refraction models because of their complexity and lack of real scenarios where refraction would have significant impact on the signal's propagation.

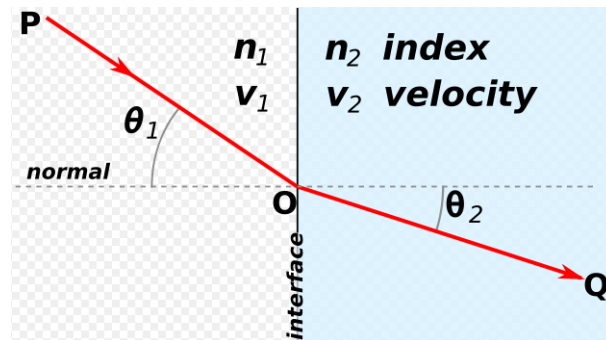


Figure 2.4: Wave refraction where $n_2 > n_1$, (resp. $v_1 > v_2$). (borrowed from <http://en.wikipedia.org/wiki/Refraction>)

2.1.4 Reflection

Reflection is a special case of refraction where the change in direction of a wavefront at an interface between two different media leads to return of the wave into the medium from which it originated. The area of the interface needs to be large, compared to the signal's wavelength, for reflection to occur.

In wireless radio communication, the most common case of reflection occurs when a transmitter or receiver are not high enough and the transmitted radio signal reflects from the ground. This situation is taken into account in a *two-ray ground model* described in section 2.2.2.

2.2 Path Loss Models

Path loss is the reduction in power density (*attenuation*) undergone by an electromagnetic wave in transit between a transmitter and a receiver. Path loss is a major component in the analysis and design of wireless networks and it is an essential part of all wireless network simulators.

Path loss models describe the attenuation between transmitting and receiving antennas as a function of the propagation distance and other parameters. In the following, the most common path loss models are briefly described. For more details, see [36].

2.2.1 The free space propagation model

This model is used to estimate the signal power at receivers when the path (meaning the first *Fresnel zone*) between the transmitter and the receiver is unobstructed. The waves obey the *inverse-square law* which states that the power density of a wave is proportional to the inverse of the squared distance from the source.

The received signal power can be computed according to equation

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \quad (2.5)$$

where $P_r(d)$ is the signal power received at distance d from the transmitter, G_t is the transmitter antenna gain, G_r is the receiver antenna gain, λ is the wavelength in meters and L is the system loss factor not related to propagation.

Equation 2.5 can be simplified as

$$P_r(d) = C_f \frac{P_t}{d^2} \quad (2.6)$$

where C_f is a free space constant that depends on the transceivers' characteristics.

2.2.2 The two-ray ground model

The two-ray ground model improves the accuracy of the free space propagation model by considering two propagation trajectories - the direct line from the transmitter to the receiver and the path of ground reflected ray (see Figure 2.5). More specifically, although the two-ray ground model gives more accurate prediction than the free space model at

long distances (as shown in [36]), it does not give good results for short distances due to the oscillation caused by the constructive and destructive combination of the two rays.

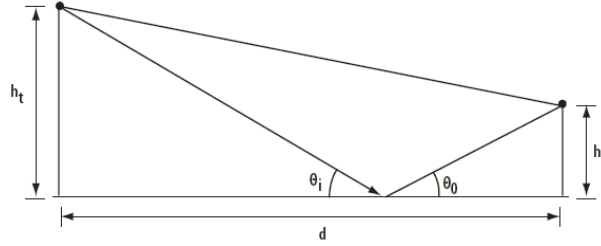


Figure 2.5: The two-ray propagation model - the signal from transmitter reaches receiver through the direct path, and through a ground reflected path.

The received signal power at distance d then can be predicted by

$$P_r(d) = P_t G_t G_r \frac{h_t^2 h_r^2}{d^4 L} \quad (2.7)$$

where h_t is the transmitter's antenna height and h_r is the receiver's antenna height.

Equation 2.7 can be simplified analogously to 2.5 by abstracting the features of the transceivers. We then get

$$P_r(d) = C_t \frac{P_t}{d^4} \quad (2.8)$$

C_t is a two-ray ground constant that depends on the transceivers' characteristics.

2.2.3 The log-distance path loss model

The log-distance generalizes the path loss and takes environmental factors into account. It has been derived combining analytical methods and empirical methods based on field measurements and reverse curve fitting on the measured data.

According to this model, the received signal power is proportional to the distance between the transmitter and the receiver d raised to exponent α which is called the *path loss exponent*, or *distance-power gradient*.

$$P_r(d) \propto \frac{P_t}{d^\alpha} \quad (2.9)$$

The *path loss exponent* depends on the environment characteristics and it has been experimentally measured in various scenarios. Some of its values are stated in Table 2.1.

2.2.4 Probabilistic propagation models

Although, the log-distance model predicts the received signal's power at a certain distance well, it predicts only an average signal's power value. The actual signal's intensity can

Environment	α
Free space	2
Urban area	2.7 - 3.5
Indoor LOS	1.6 - 1.8
Indoor no LOS	4 - 6

Table 2.1: Values of the distance-power gradient in different environments (borrowed from [39])

vary from the average value a lot. To model this variability of wireless channel, various probabilistic models were proposed. In these models, the radio coverage area is not a circle anymore, but instead, the existence of a wireless channel is a random event.

There are two classes of probabilistic models - *large-scale* models and *small-scale*, or sometimes called *fading* models.

The large-scale models are used for modelling signal strength variations over long distances. Most known large-scale model is *log-normal shadowing model* that models the path loss as a random variable with log-normal distribution centered around the mean value estimated by 2.9.

On the contrary, the fading models are used for modelling signal strength variations over very small distances. The most known fading model is *Rayleigh model* that models the path loss as a random variable with Rayleigh distribution.

2.3 Interference

Interference is an addition of two or more waves that results in a new wave pattern. This happens when the waves have the same or nearly the same frequencies. In wireless communication, when two or more signals interfere, the transferred data gets corrupted and fails to be delivered correctly.

More precisely, what matters for correct data delivery is interference at the receiver. This is a problem in wireless networks because there is no way the transmitter could know the current situation at the receiver's surroundings. The sender only knows about the channel usage in its neighbourhood which leads to two problems typical for wireless communication - the *hidden station* problem and the *exposed station* problem.

For better understanding of these problems, three ranges with respect to the transmitting station are defined:

- **Transmission range** within which a transmission can be successfully captured.
- **Interference range** within which the other transmitting stations will be interfering with the particular transmitting station, thus suffering a decrease of SNR and possible loss of information.
- **Carrier sense range** within which the other stations can detect a transmission

and postpone transmissions of their own.

In the Figure 2.6a node 1 wants to communicate with node 2, not knowing about the node 4 concurrently transmitting packet to node 5 (node 4 is outside the 1's carrier sense range). Because node 4 is in the interference range of the node 2, there happens to be interference of signals from nodes 4 and 1 at node 2 and the packet from node 1 gets corrupted. Because the transmitting node 4 was hidden from node 1, this situation is called the hidden station problem.

On the other hand, Figure 2.6b shows the exposed station problem. Now, node 3 wants to transmit packet to node 4 while node 1 is transmitting packet to node 0. Node 3 senses the transmission from 1 (node 1 is in 3's carrier sense range) and it falsely concludes that it may not send the packet to the node 4, even though the two transmissions would not be affected by interference. Such transmission would cause bad packet reception only in the zone between nodes 1 and 3.

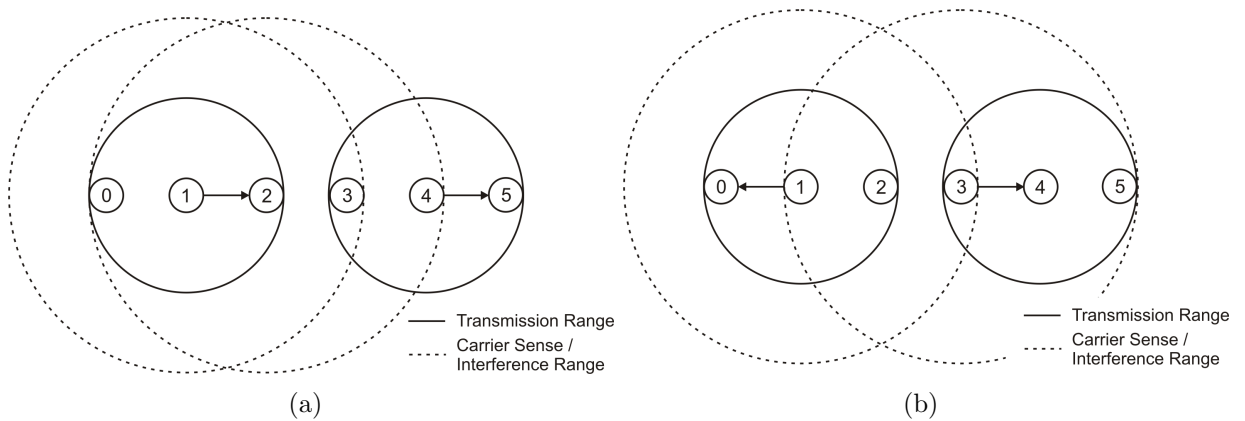


Figure 2.6: Problems of wireless communication (a) node 4 as a hidden node to node 1, (b) node 3 as an exposed node by node 1

According to Gupta and Kumar [16], signal reception in presence of other interfering signals abide by the following rules:

1. The radio locks on a strongest signal of all and the others are added to the cumulative noise level. This is known as a *capture effect* [28].
2. If the power of the strongest signal is no longer stronger than the receiver's sensitivity threshold or stronger than the cumulative noise level by SNR threshold, the radio drops receiving that signal. This is a collision.

IEEE 802.11 standard provides some solutions to interference problems in the transmitters' MAC layer protocols which I will describe in the chapter 3.

3 IEEE 802.11 Standard

As stated before, the *aglobex.wireless* simulator was designed for simulation of wireless communication of A-globe's entities using radio frequency signal. Thus, the mechanisms and protocols of the most widespread standard for wireless local area network (WLAN) computer communication, IEEE 802.11 [21], were employed in the simulator.

IEEE 802.11 is a set of Wi-Fi standards for WLANs that defines their standard physical (PHY) and medium-access control (MAC) layers in the publicly available ISM bands. It also describes the functions and services required by an IEEE 802.11-compliant device to operate within ad hoc and infrastructure networks, as well as the aspects of station mobility within those networks. Furthermore, it also defines several privacy protocols (WEP, WPA and WPA2) to provide the same security level as the wired Ethernet.

In this chapter I will briefly discuss the specifications of individual standards of 802.11 family, show two possible operation modes and describe some of the MAC layer channel access protocols.

3.1 Physical Layer

IEEE 802.11-compliant devices communicate using radio frequency signals in 2.4 GHz or 5 GHz ISM bands with bandwidth of about 20 Mhz. The signal is modulated by *direct-sequence spread spectrum* (DSSS) technique at 2.4 GHz or by *orthogonal frequency-division multiplexing* (OFDM) technique at 5 GHz. These modulation techniques offer various bits per symbol values, thus providing various transmission speeds and various susceptibility to noise.

3.1.1 Maximum Information Rate

According to information theory, maximum amount of error-free digital data (information) that can be transmitted with a specified bandwidth in the presence of noise or interfering signals is bounded by two phenomena - the Hartley's law and the Shannon's Theorem. The maximal transmission data rate can be estimated according to the *Hartley's law*

$$R \leq 2B \log_2 M \quad (3.1)$$

where R is an achievable line rate in bits per second, B is the analog bandwidth and M is the number of distinct signal levels given by the modulation.

On the other hand, by increasing the number of signal levels, the data-rate cannot be increased forever. At some point, the receiver would not be able to distinguish the signal levels because of the noise. The upper bound to the capacity of a link is given by the

Shannon's Theorem

$$C = B \log_2 \left(1 + \frac{S}{N} \right) \quad (3.2)$$

where C is the achievable channel capacity in bits per second, S is the total received signal power over the bandwidth in watts and N is the total noise or interference power over the bandwidth in watts.

3.1.2 Operation Modes

IEEE 802.11 standard can work in two modes: either in the presence of a base station (*access point* - AP) with a regular client-server structure called IM-BSS (Infrastructure Mode Basic Service Set) or in an *ad-hoc* mode with the absence of a base station. This operational mode is called IBSS (Independent Basic Service Set). See Figure 3.1. Operating in ad hoc mode allows all wireless devices within range of each other to discover and communicate in peer-to-peer fashion. In order to communicate, all wireless adapters must use the same SSID (Service Set Identifier) and the same channel number.

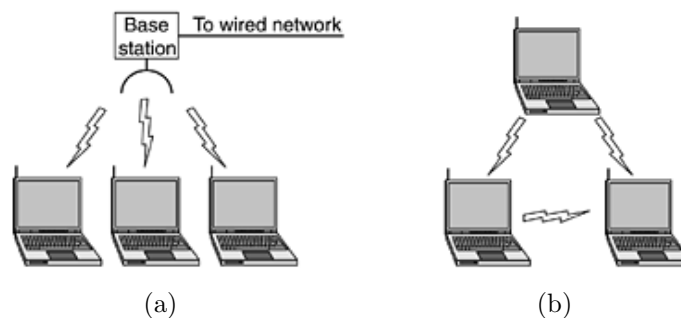


Figure 3.1: Wireless mode (a) with base station (access point) (b) without base station (ad hoc mode).

3.2 Medium-Access Control Layer

802.11 standard defines three MAC channel access methods - *Distributed coordination function* (DCF), *Point coordination function* (PCF) and *Hybrid coordination function* (HCF).

DCF

DCF can be used either in the Infrastructure Mode or in IBSS ad hoc mode. It is based on the *Carrier Sense Multiple Access - Collision Avoidance* (CSMA-CA) MAC protocol and must be implemented in all 802.11 stations.

The CSMA/CA protocol uses a *random backoff* procedure to reduce the collision probability between multiple stations accessing a medium. This means that just after the medium becomes idle (as indicated by the carrier sense (CS) function) the station waits for a random time period before transmitting. Furthermore, all unicast traffic uses positive acknowledgments (ACK frame) of frames - when the sender receives no ACK, frame retransmission is scheduled.

CS is performed by physical and virtual mechanisms. When either mechanism indicates a busy medium, the medium is considered busy, otherwise, it should be considered idle. The *physical CS mechanism* is provided by the PHY. In principle, each station, when not transmitting, senses the medium and if some signal with power higher than station's sensitivity threshold is received, the medium is considered busy. The details of physical CS are provided in the individual PHY specifications.

The *virtual CS mechanism* is provided by the MAC layer. It works on a principle of distributing RTS (Ready-to-send) and CTS (Clear-to-send) frames by the transmitting and the receiving stations, prior to the actual data frame. The RTS and CTS frames contain a *Duration field* that specifies the amount of time that the channel is to be reserved for transmitting the actual data frame and the returning ACK frame. All stations that receive either RTS (transmitted by the originating station) or CTS (transmitted by the destination station) learn about the future medium reservation and set the *Network Allocation Vector* (NAV) signal - kind of reminder to keep quiet for a certain period of time. Virtual channel sensing is illustrated in Figure 3.2.

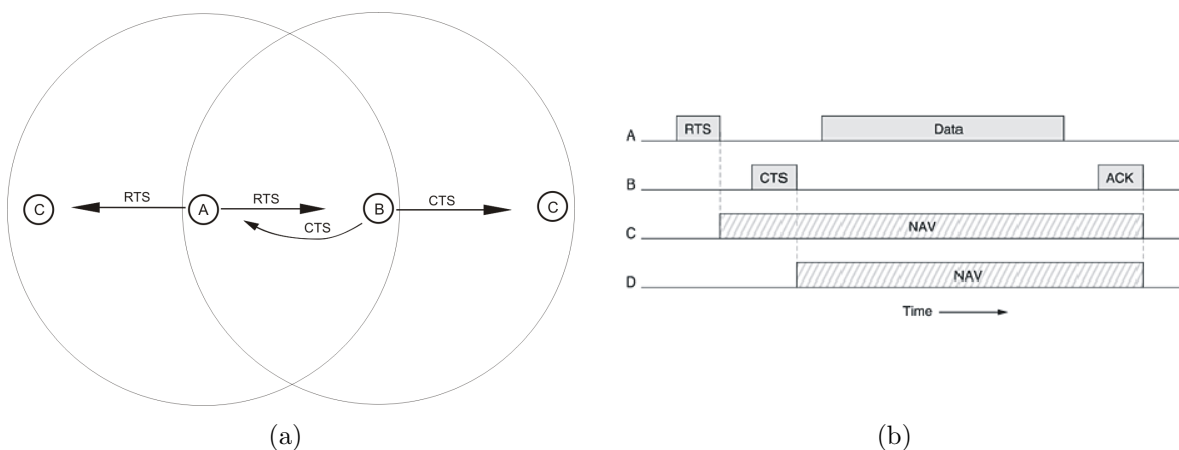


Figure 3.2: Virtual channel sensing when A is sending a frame to B - from topological point of view (a), and from the view of the channel allocation (b).

The RTS/CTS exchange enables fast collision detection. If the returned CTS is not detected by the originating station, the station may repeat the process more quickly

than if the long data frame had been transmitted and a returned ACK frame had not been detected. This mechanism is supposed to be solution to hidden station problem as stated in 2.3.

The RTS/CTS mechanism cannot be used for broadcast or multicast transmissions because there are potentially multiple concurrent senders of the CTS frames. The RTS/CTS mechanism need not be used for every transmission. Stations can be configured to use RTS/CTS either always, never, or only on frames longer than a specified length. Because the additional RTS and CTS frames add significant overhead (as measured for example by Basagni et al. in [4]), the mechanism is not always justified, especially for short data frames.

PCF

In PCF mode, which is applicable only on infrastructure network configurations, the base station polls the other stations, asking them if they have anything to send. Since transmission order is completely controlled by the base station in this mode, no collisions ever occur.

The basic mechanism is for the base station to broadcast a beacon frame periodically (10 to 100 times per second). The beacon frame contains system parameters, such as clock synchronization. It also invites new stations to sign up for polling service. Once a station has signed up for polling service at a certain rate, it is effectively guaranteed a certain fraction of the bandwidth, thus making it possible to give quality-of-service guarantees. PCF mode implementation is optional for WiFi devices.

HCF

HCF access method is only applicable on QoS (Quality of Service) network configurations and should be implemented in all QoS stations. It combines functions from the DCF and PCF with some enhanced QoS-specific mechanisms. The HCF uses both contention-based channel access method (similar to DCF method), called the *enhanced distributed channel access* (EDCA) mechanism for contention-based transfer, and controlled channel access (similar to PCF method), referred to as the HCF controlled channel access (HCCA) mechanism, for contention-free transfer.

More details on HCF can be found in [21].

3.3 Protocols

The original IEEE 802.11 standard was formed to unite the physical (PHY) and medium-access control (MAC) layers for WLANs in the publicly available ISM bands. This

standard provided three different PHY layer options - infrared, frequency hopping spread spectrum (FHSS) at 2.4 GHz, and direct sequence spread spectrum (DSSS) at 2.4 GHz. Since its creation in 1997, various extensions called Task Groups have been formed. These new extensions abandoned the infrared and FHSS PHY layer options and defined higher data rates, better reliability and other features.

In this part, I will introduce some of the IEEE 802.11 variants (Task Groups). The overview of the most important standards can be seen in Table 3.1.

Standard	Year	Band [GHz]	Maximum rate [Mbit/s]	PHY layer
original IEEE 802.11	1997	2.4	2	DSSS / FHSS
IEEE 802.11a	1999	5	54	OFDM
IEEE 802.11b	1999	2.4	11	DSSS
IEEE 802.11g	2003	2.4	54	OFDM
IEEE 802.11n	2009	2.4 / 5	600	MIMO

Table 3.1: Overview of IEEE 802.11 standards.

IEEE 802.11b

802.11b is a direct extension of the modulation technique defined in the original standard. It was designed to increase the maximum bit rate in 2.4 GHz frequency range from 2 Mbps to 11 Mbps while maintaining interoperability with the original standard. The standard keeps the original MAC layer but redefines the PHY layer (unlike the original that used DSSS and FHSS modulations, 802.11b works only with DSSS) to increase the spectral efficiency. The supported bit rates are 11, 5.5, 2 and 1Mbps.

802.11b devices suffer interference from other products operating in the 2.4 GHz band, like microwave ovens, Bluetooth devices, baby monitors and cordless telephones.

IEEE 802.11a

802.11a keeps the original MAC layer but the PHY layer was reworked to provide rates up to 54 Mbps in 5 GHz band. It employs OFDM spectrum spreading technology that is more efficient at working with high bit rates than DSSS technology. The supported bit rates are 54, 48, 36, 24, 18, 12, 9 and 6 Mbps, thus the *task group a* compliant devices are incompatible with *task group b* compliant devices.

Because the 5 GHz band is relatively uncrowded and since the available band at Unlicensed National Information Infrastructure (U-NII) is about 300 Mhz (eight nonoverlapping 20MHz bands available, compared to three with 802.11b), IEEE 802.11a suffers from the interference much less than 2.4 GHz band protocols.

Although the signal energy at 5 GHz is more likely to be absorbed than the 2.4 GHz signal and its power is attenuated much more over distance, its performance is superior

to the 802.11b (as shown by researchers at Atheros Communication [10]).

IEEE 802.11g

802.11g task group is fully backwards compatible with the 802.11b protocol. It is working in 2.4 GHz band and enables transmissions up to 54 Mbps. The standard uses DSSS modulation technology at 802.11b data rates (11, 5.5, 2, and 1 Mbps) for the interoperability mode while for the higher rates (54, 48, 36, 24, 18, 12, 9, and 6 Mbps), it employs the OFDM technology.

The drawback of working in the same frequency band as 802.11b is that it is influenced by the same interference sources in the crowded 2.4 GHz band and unlike 802.11a, it still has the three-channel restriction.

IEEE 802.11n

802.11n is a new WiFi standard that enables transmissions at a maximum of 600 Mbps. This is achieved by adding multiple-input multiple-output technology (MIMO) and 40 MHz channels to the PHY layer, and frame aggregation to the MAC layer (to deal with 802.11 protocol overheads).

MIMO technology provides *Spatial Division Multiplexing* (SDM) technology that multiplexes multiple independent data streams, transferred simultaneously within one frequency channel, using multiple antennas. Thanks to this, the data throughput increases significantly.

The standard works in both 2.4 and 5 GHz bands and 802.11n-compliant devices are backwards compatible with 802.11a/b/g devices, however, in their presence the standard's performance decreases significantly.

Other significant IEEE 802.11 task groups

IEEE 802.11d standard is also referred to as the Global Harmonization standard. It is used in countries where systems using other standards of the IEEE 802.11 family are not allowed to operate. Country-specific information is included in a beacon transmissions to inform receiving stations about the available parts of the spectrum, maximum transmission power and other radio constraints.

IEEE 802.11h standard was originally designed to address European regulations. It solves problems like interference with satellites and radars using the same 5 GHz frequency band. In principle, stations that happen to detect transmission should limit their transmitting power, or free the channel where the transmission was detected.

IEEE 802.11e defines a set of Quality of Service (QoS) enhancements for WLAN applications through modifications to the MAC layer. It redefines both the centrally controlled

channel access and the contention-based channel access of CSMA/CA by including packet priorities. Packets with higher priority then have access benefits over the lower-priority packets in a Differentiated Services manner.

4 Wireless Network Simulators

Communication in wireless networks is a complex task with many variable parameters to be taken into account, such as signal propagation, interferences, obstacles and noise in the environment, or hosts' movement. Also, unlike the wired networks, problems like power constraints, decreased reliability or typically higher density and number of nodes need to be faced. Because of these factors, it is inevitable to thoroughly test the behaviour of all newly designed protocols and applications before using them in practise. Testing can be done either by implementing and deploying of the developed concepts on real hardware (testbeds) or by using wireless network simulators. Most researchers today prefer simulators to real-size testbeds, although testbeds might provide the most accurate results. The reasons are the cost of hardware, testbeds' limited monitoring and debugging possibilities, high effort needed to create an experiment's environment resembling the real scenario, as well as the lack of reproducibility of experiments on real hardware. Therefore the use of testbeds is applicable for smaller networks in the final stages before concept's deployment, while in the first stages of the implementation phase and with the growing size of experimented networks, grows the importance of software-based simulation.

In this chapter I will give an overview of the most frequent wireless network simulators, principles and simulation techniques used in these simulators and describe how these simulators deal with accuracy. Overview of wireless network simulators has also been presented in [19], [31] and issues of accuracy have been studied in [9] and [17].

4.1 The Accuracy Of Wireless Network Simulators

In general, no network simulator is accurate. All simulators work with some abstraction of the real world which in its nature causes a certain level of imprecision. When really good accuracy is needed, experiments should be, if possible, conducted on the real devices, using testbeds. Otherwise, researchers have to manage with simulators and their inaccuracies. In the following, I am about to specify individual causes of imprecision.

4.1.1 Level of details

Selecting the correct level of detail (granularity) for a simulation is a difficult problem. Apparently, the more detailed model (finer model granularity), the more accurate is the simulation. But on the contrary, it is clear, that it is not necessary and even achievable to model all the details (for example, up to the atoms and electrons).

As studied by Heidemann et al. [17], lack of detail may have serious impact on the obtained results and may cause the results wrong or misleading in two ways. Either they

are simply incorrect, or they are technically correct but provides an answer to irrelevant parts of the design space, thus making the results inapplicable. On the other hand, too much detail results in slow simulations and cumbersome simulators. The granularity must be chosen with care and ideally suited according to the needs of the simulated application.

As no metrics for granularity levels is available, Hogue et al. [19] defined the granularity as **finest** < **finer** < **fine** < **medium** < **application-level**. Granularity levels for some of actual network simulators, as well as their programming languages and licence they use are shown in table 4.1.

Name	Granularity	Interface	Licence
ns-2	Finest	C++/OTCL	Open source
GloMoSim	Fine	Parsec (C-based)	Open source
J-Sim	Fine	Java	Open source
OMNet++	Medium	C++	Free for academic and educational use
OPNet	Fine	C	Commercial
SWANS	Medium	Java	Open source

Table 4.1: Granularity of simulators, programming language they are based on and licences they use. Data for the table were taken from [19]

4.1.2 Mobility models

Mobility models for simulating mobile networks are another source of imprecision. They are probabilistic approximation of hosts' movement and they have been surveyed by Camp et al. [8] and Lin et al. [32]. Nodes' mobility impacts the whole network performance in various ways. It can significantly increase capacity, as shown in [15] or it can have an impact on network connectivity, as shown in [11]. Random Waypoint mobility models [6] used to be common mobility models until their harmful impact was revealed. This was the fact that even though the speed of a mobile station was chosen uniformly at random from a given interval, the speed of a node in steady state was not necessarily from the same distribution (it was caused by slower nodes traveling for a longer time). Since then new more realistic mobility models have been suggested, like the Obstacle Mobility Model [27], the Graph-based Mobility Model [41], the Group Mobility Model [20] or the UDel model [7].

4.1.3 Radio propagation models

Modeling of the radio waves propagation is one of the most difficult task of the wireless networks simulation. As mentioned in chapter 2, radio waves, similarly to light waves, are affected by reflection, refraction, diffraction, absorption and scattering. Some simulators

take reflection and absorption into account, but so far, there are no good implementations of the other three phenomena. Also, the UDel project [7] models radio waves by using ray-tracing and Markov chains, but these techniques are highly time-consuming and decelerate the simulation up to 100 times. Furthermore, many simulators use highly simplifying assumptions about circularity of radio's transmission area and about the simulation's environment (they do not consider obstacles like hills or walls, or nodes' heights).

4.2 An overview of the most frequent simulators

According to Hogue et al. [19], the literature mentions less than twenty wireless network simulators currently in use, not considering sensor network simulators. The first wireless network simulator appeared with the wireless extensions to ns-2 by the Monarch CMU projects in 1992. Since that time numerous other simulators have appeared and some of them are now widely used. While most of the simulators allow modelling of the lowest network layers, more and more people focus on the highest levels, developing new applications and concepts, e.g. for mobile ad-hoc networks (MANETs). For example, Helbrück and Fischer developed ANSim [18] to analyze structural properties of the MANETs, and Jane simulator [30] has been used for studying ad-hoc gaming. In the following, I am about to give general description of some of the most common simulators.

ns-2

ns-2 [33] is the most popular network simulator. Its behaviour is highly trusted within the networking community and it has become de facto standard for network simulation. It is based on a combination of C++ and OTcl languages. In general, C++ is used for implementing protocols and extending the ns-2 library, while OTcl is used to create and control the simulation environment itself, including the selection of output data.

Thanks to its popularity and open source licence, many extensions are available enlarging the already huge pool of available features. ns-2 offers a large number of external protocols already implemented, as well as various mobility and path loss models or energy model. Thanks to its emulation facility, ns-2 can also be connected to a real network and capture live packets just like a common node or inject packets into the live network. Downsides of ns-2's high accuracy are its lack of modularity and its inherent complexity. Adding new components or protocols or modifying old ones is a hard task that requires good knowledge of the simulator. Its complexity and high consumption of computational resources leads to lack of scalability, preventing execution of simulations with more than a few hundred nodes.

GloMoSim

GloMoSim [42] is a discrete-event simulator developed at UCLA (California, USA) in 1998 and it is the second most popular wireless network simulator. It is written in Parsec, an extension of C for parallel programming, allowing the simulation to run on shared-memory symmetric processor computers.

The parallelization is achieved by splitting the network into subnetworks with similar number of nodes. Each of these subnetworks is then simulated by distinct processors.

GloMoSim's network model does not define every network node as entity because it would lead to too many objects and would slow down the simulation. Instead, in GloMoSim, objects called entities represent network layers. These entities then interchange messages simulating network events crossing the network layer stack. By using these principles, GloMoSim can simulate networks of tens of thousands hosts.

GloMoSim stopped releasing updates in 2000. Instead, it is now updated as a commercial simulator called QualNet.

SWANS++

"Scalable Wireless Ad Hoc Network Simulator" (SWANS) [3] is a complete Java-based library for simulation of MANETs running on the JiST engine. JiST is a discrete-event platform that transforms Java virtual machine into a scheduler for events. This allows each entity in simulation to be in charge of determining the time needed for its execution, instead of following some shared clock.

In SWANS, environmental models and protocol layers are represented as object entities. The `Field` entity provides node mobility and radio propagation. Nodes consist of a number of entities implementing various protocol layers, where the radio entity is connected to the `Field` entity. Packets then traverse the protocol stack entities usually as simple references, at virtually no cost.

In terms of quality, SWANS has been compared to GloMoSim and Schoch et al. [40] compared it also to ns-2 with very similar results in the geographic routing protocol, but considerably faster with much lower consumption of computational resources. In the same paper, JiST/SWANS was compared to GloMoSim, in terms of speed and memory consumption, again resulting with better performance of JiST/SWANS.

J-Sim

J-Sim [29] is a component-based, compositional simulation environment developed at Ohio and Illinois Universities. It is written in Java and originally it was designed for simulation of wired networks. For wireless network simulations, J-Sim offers the Wireless Extension that supports IEEE802.11 MAC protocol implementation.

In J-Sim, each network entity - a node, a link or a protocol - is a component. Each component can be atomic or composed of other components. Components communicate through their ports, by connection types 1-to-1, 1-to-many or many-to-many. The result is, that this component based architecture scales better than the object oriented model used by ns-2 and other simulators.

OMNet++

OMNeT++ (Objective Modular Network Testbed in C++) [25] is a well-designed discrete-event simulation environment capable of simulating any system that can be mapped to active components communicating by passing messages.

Like J-Sim, it has a component-based, modular and extensible architecture. The basic entity in OMNeT++ is a module. Modules can be atomic or they can be composed of submodules. Modules communicate with each other via messages through gates (equivalent to ports in J-Sim, but they support only one-to-one communication). Gates are linked to each other using connections. A connection can be associated with a propagation delay, error rate and data rate.

For simulation of wireless ad hoc networks, OMNeT++ uses external extensions INET Framework and the Mobility Framework. The simulator includes parallel simulation functionality, thus being able to model networks made of tens of thousands stations.

OPNet

OPNET (Optimized Network Engineering Tools) Modeler [12] is a well-established commercial discrete-event simulator developed at MIT. It provides an environment for studying and designing networks, devices, protocols and applications.

OPNet Modeler defines a network as a collection of sub-models representing sub-networks or nodes, thus employing hierarchical modeling. Developing of models is done in a hierarchical way using a four-level structure. *Network level* handles topology modeling and overall configuration, *node level* deals with internal structures of nodes like transmitters and receivers, *process layer* models functionalities of node level devices as finite state machines, and finally *Proto-C layer* is where coding of model behavior takes place in Proto-C language (an extension of C).

For modeling, simulation, and analysis of wireless networks, the simulator uses Wireless Suite extension. It allows models of RF signal propagation, interference, transmitter/receiver characteristics, node mobility including handover, and the interconnection with wired transport networks. There is an extensive list of supported wireless networking technologies such as MANET, IEEE 802.11, 3G, Ultra Wide Band, IEEE 802.16, Bluetooth, and Satellite. An interesting feature of OPNet is its ability to execute and monitor

several scenarios in a concurrent manner.

4.3 The event-driven simulation versus real-time simulation

Discrete-event simulators represent the world as a chronological sequence of events. These events occur in a discrete point in time and change state of the system. The simulation uses clock to keep track of the current simulation time, but this does not flows continuously, instead, it skips an event start time to the next event start time as the simulation proceeds.

Real-time simulators, on the other hand, employ continuous time clock and process events as they come.

From these properties accrue following advantages and disadvantages of these systems:

- Thanks to their clock independent to real time, discrete-event simulators are able to model long-lasting scenarios in shorter time than they actually last.
- Discrete simulators can model more complex scenarios because they are not bounded by strict time restrictions as the real-time simulators.
- Discrete simulations are easier to implement and they can benefit from optimization techniques like parallelism and computation distribution or *staged simulation* (a technique that improves the performance by identifying and eliminating redundant computations in discrete-event simulators).
- Real-time systems, as opposed to discrete-event simulators, thanks to their real-time clock, are able to support co-simulation features - they can simulate the network along with the latency-related aspects of the wireless communication in combination with the node computations and the dynamics of the environment.

5 A-globe Agent Platform

In this chapter I will describe a multi-agent platform *A-globe* [23], its fundamental parts and some of its functions and mechanisms I used in my work.

5.1 About the platform

A-globe is a Java-based multi-agent platform developed at Agent Technology Center (ATG) at the Department of Cybernetics, CTU Prague. The platform is designed for experimental scenarios testing with a possibility of simulation of agents' position and communication inaccessibility. Hosted agents are provided with communication infrastructure, migrating functions, agent stores, deploy service, directory services, etc.

Since its creation in year 2001, A-globe was successfully deployed to various industrial applications. For example, it was used in NAIMT (Naval Automation and Information Management Technology) [38] project for simulating underwater mine-sweeping operations (see Figure 5.1(a)), furthermore, it is used in CAMNEP (Cooperative Adaptive Mechanism for Network Protection) intrusion detection system [37] or in AGENTFLY project[22], a system for air-traffic control of several autonomous aerial vehicles (manned as well as unmanned)(see Figure 5.1(b)). Besides these projects, A-globe is also used by company CADENCE Design Systems for simulation and chipset design process modelling or by academic institutions like University of Edinburgh, Florida Institute for Human and Machine Cognition or Masaryk University in Czech Republic.

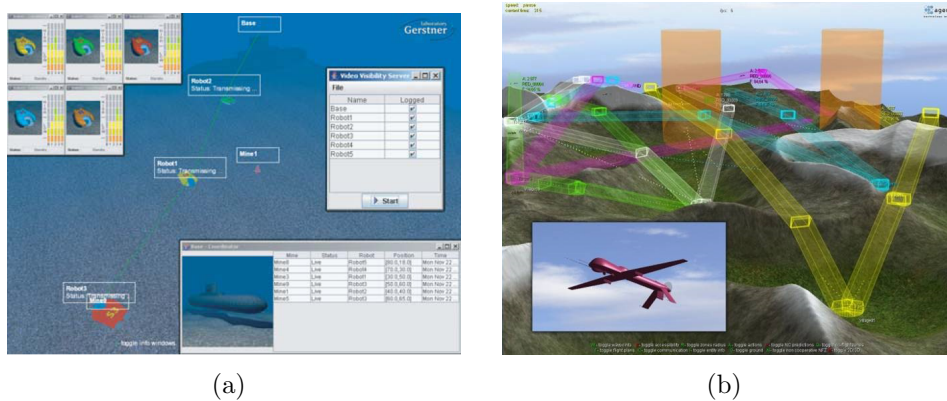


Figure 5.1: Some of projects employing the A-globe. (a) NAIMT scenario, (b) AGENTFLY scenario

A-globe platform is FIPA [13] compliant on the ACL level while it does not support the FIPA specification for inter-platform communication which would add computational overheads that are not necessary for closed systems. Thanks to avoiding these overheads, the platform achieves great scalability, simulation support and agent mobility.

Communication in A-globe is very fast and the platform is relatively lightweight. As studied in [23], it strongly outperforms other multi-agent platforms like FIPA-OS [35], ZEUS [34] or JADE [5] in memory requirements and message delivery times.

5.2 A-globe Architecture

The system architecture is shown in Figure 5.2. Its main parts are the following.

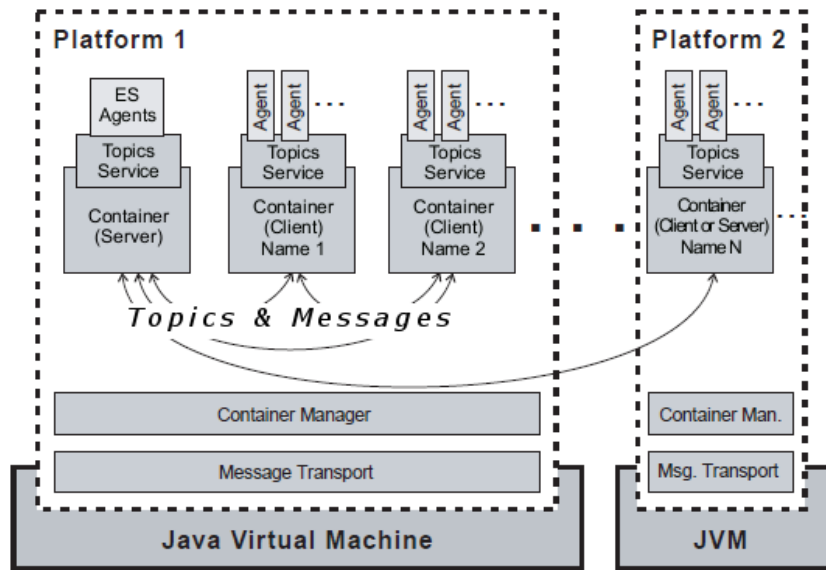


Figure 5.2: System architecture

Agent Platform - provides basic components for running one or more agent containers. More than one platform can run simultaneously on a single host, each in its own JVM instance. The main components of the platform for ensuring the rest of system functionality are Container Manager and Message Transport.

Container Manager handles starting, execution and finishing of agent containers.

Message Transport ensures an efficient exchange of messages between two agent containers running either within a single agent platform or on distinct agent platforms.

Agent Container hosts agents and services, entities that are able to send and receive messages, and provides them with some low level functions like message transport, agent management, or service management. High level functionality, like agent deployment, migration, directory facilitator, etc. is provided as standard container services.

One or more agent containers can run within single agent platform. Running several containers on one platform rapidly decreases system resources requirements because only single JVM is used.

The agent container structure is shown in Figure 5.3.

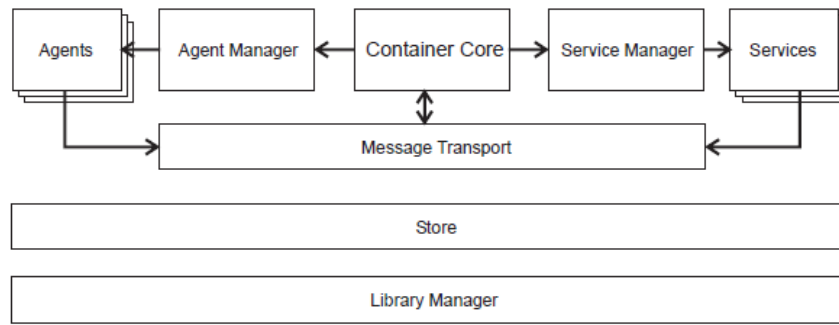


Figure 5.3: Agent container architecture

The container composes of following components:

- **Container Core** - starts up and shuts down all container components.
- **Store** - is an XML file where startup configuration and parameters of each container component, agent or service are stored. These data persist in the store after the platform is shut down and can be again loaded in the future.
- **Library Manager** - handles loading and usage of libraries installed in the container.
- **Message Transport** - manages sending and receiving of messages from, to, and within the container.
- **Agent Manager** - handles agents creation, execution and removal. It routes incoming messages to the agents, and manages agents migration.
- **Service Manager** - starts and stops container's services takes care of their interfacing to other container components.

Furthermore, the container hosts following entities.

- **Agents** - are autonomous entities residing in the containers. Each of them is running in its own thread and they can communicate with each other using messages. There are either *actor agents*, simulated entities that play roles in the simulated world, or the *environment simulation agents* (ES) that do not participate in the simulated scenario and instead, they take care of the environment simulation. They reside on server containers and communicate with other agents by *topic messaging* - messaging used for environment simulation.
- **Services** - services to be used by the container's agents for various purposes like for other agents localization (*DirectoryService*), sending and receiving topics (*TopicsService*), migration (*AgentMoverService* and *DeployService*) etc. Agents can

communicate with the service by messages or via *service shell*. There are two types of services - user services and system services. System services are automatically started by the container and form a part of the container infrastructure (agent mover, visibility service, directory services etc.). The user services can be started by the user or by an agent/service.

5.3 Message Transport

Agents in A-globe can communicate with each other by two means - either by *topics* or by *messages*.

Messages can be exchanged between agents or services that are located on the same container or on mutually reachable containers. Containers reachability is managed by the *visibility service* that, if running, considers distances between agents to decide about their mutual reachability. The message structure respects FIPA-ACL [14] specifications and the messages themselves can be unicast or multirecipient (can be send using multicast). Messages between two agent containers from different A-globe platforms are exchanged via a shared TCP/IP connection that is established when the first message is sent. Messages between two agent containers located in the same agent platform are passed via the platform-level message transport component and the flow of messages within the same container is shown in Figure 5.4. The message hooks shown in the figure will be described further in this section.

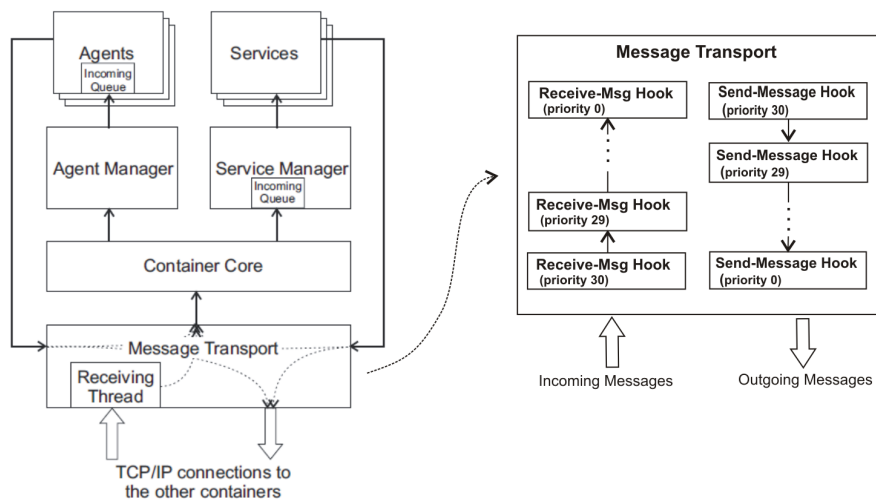


Figure 5.4: Message flow

Topics are system messages that are reliably delivered regardless of the container reachability. They are used for distribution of system information like configuration parameters, entity position updates, reachability information, etc. To receive a particular topic, agen-

t/service need to subscribe it first. Each agent/service can handle one or more topics and one topic can be handled by more than one agent/service.

5.3.1 Message Hooks

As shown in Figure 5.4, messages pass through constructs called *message hooks* on their way to the receivers. These hooks, if registered at the container-level message transport component, intercept messages that are being sent/received on the client container. These intercepted messages can be then delayed, modified, or processed in various ways. There are two kinds of message hooks - *send message hooks* for intercepting outgoing messages and *receive message hooks* for intercepting incoming messages. Each of the hooks has its *priority* that determines the order in which the hooks process the message. Messages are processed by the hooks in descending priority order, i.e. a message is firstly intercepted by the hook with the maximum priority and lastly by the hook with the minimum priority.

No more than one send message hook and one receive message hook with a particular priority level can be registered at the container message transport component. As shown in the Figure 5.4, priority levels can vary from 0 to 30. So far, hooks with two priorities are reserved. Priority level 2 hooks are used by the *visibility service* to prevent messages from being delivered to unreachable containers and priority level 1 hooks are used by the *Sniffer* agent to visualize the message exchange.

5.4 Scenario Support in A-globe

A-globe is primarily aimed at large scale, real world simulations. To support this goal, it includes a special infrastructure for environmental simulation.

As I mentioned earlier, in A-globe simulations there are two kinds of agents - *actor agents* that simulate acting entities in the simulated world and *environment simulation (ES) agents* that implement the simulated world. ES agents, for example, take care of managing position and movement of actor agents (*Position Updates Provider*), handles entity reachability and their crashes (*Visibility-Crash Agent*), or takes care of entity visualisation (*Operator Agent*). To communicate with each other and with the action agents, the ES agents use the topic messaging. Simulation structure example is shown in Figure 5.5.

5.4.1 Container Reachability

One of the most important A-globe's simulation features, especially in terms of the aim of this thesis, is the container reachability. Container reachability should simulate

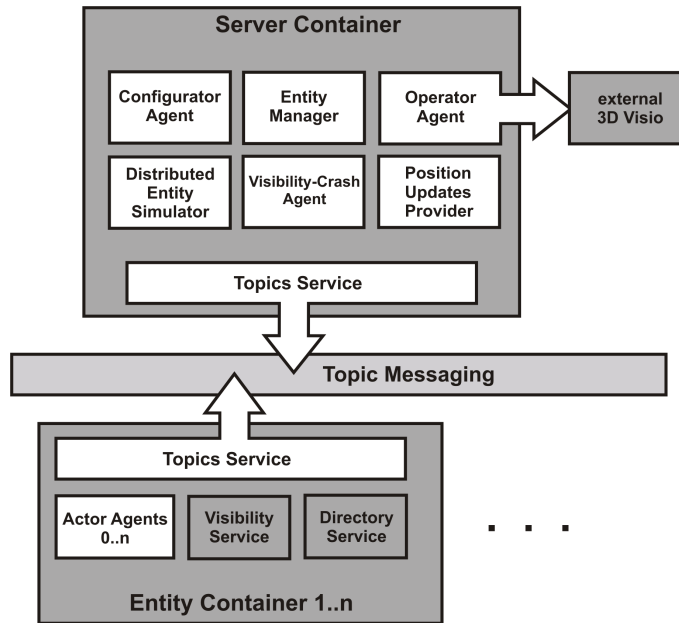


Figure 5.5: A-globe simulation structure

communication inaccessibility for reasons of too large distances or obstacles between entities. Thus, entities in containers that are not mutually reachable, cannot exchange messages with each other as it would be in the real world.

The basic mechanism of processing container reachability in A-globe scenarios is shown in Figure 5.6.

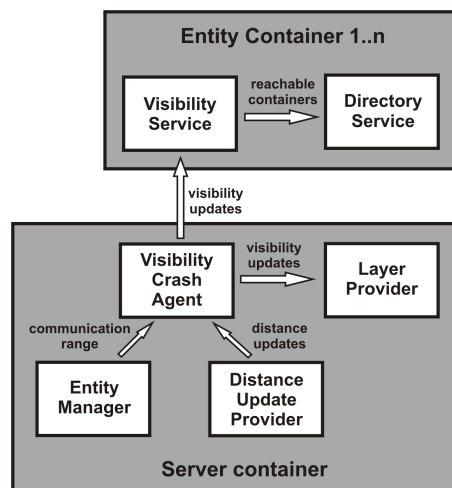


Figure 5.6: Reachability information propagation

Entities can be configured to have *communication range*. Entities that are further than this range are then considered inaccessible for the particular entity. Positions and mutual distances of entities are fed by *Position/Distance Updates Provider* to *Visibility Crash*

Simulator. This simulator, besides handling entities crashes, estimates mutual reachability taking the communication range into account, and sends them as *visibility updates* to the *Visibility Services* of individual entities and to a *Layer Provider* that (if present) visualizes them. The *Visibility Service* than informs the local *Directory Service* about newly reachable or unreachable containers. However, the main task of the *Visibility Service* is to intercept and process outgoing messages by the message hooks. Using the reachability information, the service either sends the message further if the receiver is from a reachable container or prevents the message from being send successfully if the receiver resides in an unreachable container.

6 Module Design

The *aglobex.wireless* module is neither *discrete-event* nor pure *real-time* simulator. Instead, it works on a hybrid principle. As real-time simulators, it uses continuous time clock, shared with the clock of the simulated entities and enables co-simulation of various entity behaviour. On the other hand, its simulation features are not carried out continuously. They are triggered by two events.

As a first case of discrete behavior, the module handles entities' movement and computes container reachability information only when a change in entities' positions occurs. The other discrete behavior is triggered as periodically scheduled A-globe's event. The scheduling period is specified by parameter *INFORM_PERIOD* which is optionally adjustable during the simulation. By default, it is set to 50 milliseconds. In this scheduled event, the simulator processes all the intercepted messages. It checks the reachability of the message receivers, delays the message transmission according to the size of the message and the occupancy of the frequency channel, and handles interference of signals and message retransmissions. The basic state diagram of the simulation is shown on Figure 6.1.

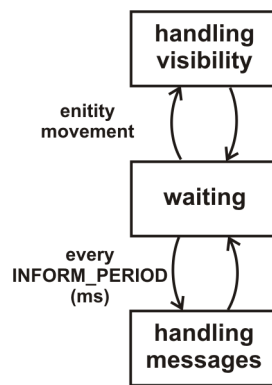


Figure 6.1: Simulation state diagram

In the following, I will describe the overall design of the *aglobex.wireless* module. First, I will discuss the client-server architecture - what are the tasks of client and server parts and how they interact. Then I am about to describe the new way of container reachability estimation that is more accurate than the one used in A-globe, because it takes wireless transceiver's characteristics and the map of the environment into account. Finally, I will show, how the actual message transport is solved in the *aglobex.wireless* module.

6.1 Server-Client Architecture

The module is designed in server-client fashion. It handles A-globe's containers reachability and message transport and for that purpose uses some of the A-globe's structures, modifies some of them, or completely replaces some of them. However, the main parts of the module are *Wireless Client Service* and *Wireless Server* which are discussed below. The *aglobex.wireless* also takes advantage of an A-globe feature provided by the *Distributed Entity Provider* that enables distribution of the simulation to several platforms according to *world partitions* which split entities into several simulated regions depending on the entities positions, thus enabling distribution of the simulation on more hosts. The module structure is shown in Figure 6.2.

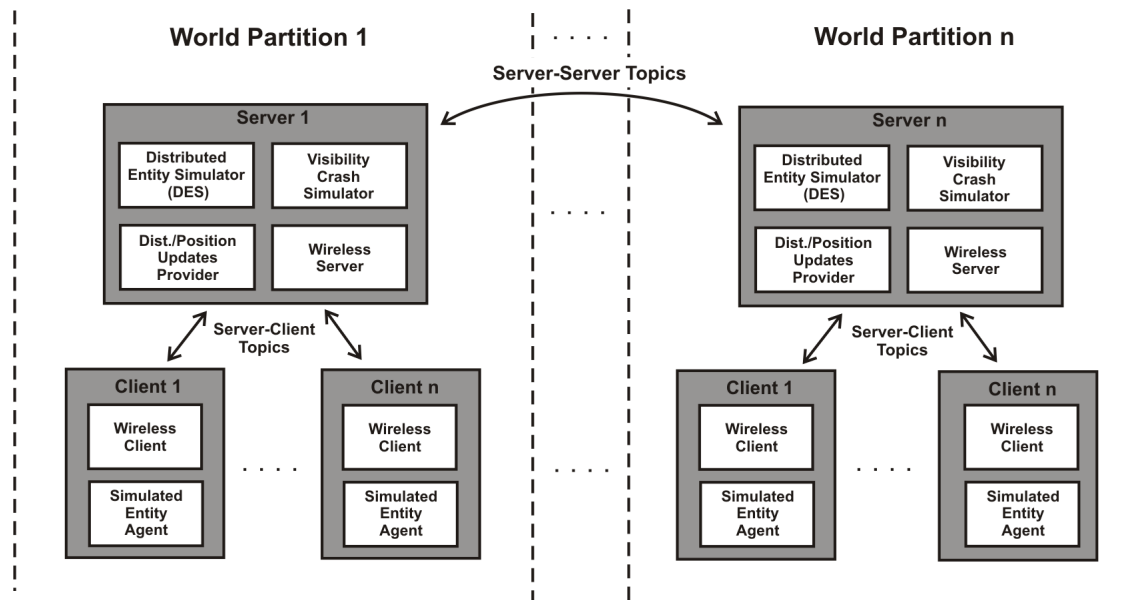


Figure 6.2: System architecture

6.1.1 Server side

The server side of the wireless module, represented by *Wireless Server Agent* and *Wireless Server Simulator*, handles two tasks. First one is to process positions and movement of the simulated entities¹ and to estimate their reachability properties depending on their mutual distances and taking obstacles from the map of the environment into account. This function is a replacement of a part of A-globe's *Visibility Crash Simulator* which was modified to work together with the *Wireless Server* and provide only the handling of entities crashes. How the reachability information is estimated, is in detail described in section 6.2. As its second function, the server processes all the *Wireless Message Infos*

¹In the following, simulated entities, containers and hosts, refers to the same thing

coming from *Wireless Client Services*. It delays the messages from wireless hosts taking the message length and the wireless channel usage into account, and prunes messages' receivers according to their reachability from the senders. As mentioned above, A-globe's messages can be sent only using unicast or multicast, thus also the module can process only unicast and multicast messages. Message broadcasting can be realized by multicasting a message to all containers reachable from its sender. The message handling is further described in section 6.3. The server has knowledge of wireless hosts's parameters like their maximal transmission power, sensitivity threshold, minimal required SNR, or their actual position. The wireless parameters are specified in a configuration XML file and the position information is provided by *Position Updates Provider*. All of these data is stored in *IHost* class representing individual wireless hosts.

6.1.2 Client side

On the client side, the *Wireless Client Service* is responsible for the wireless module operation. It provides the module with two functions. First, it distributes container reachability information received from the *Wireless Server*, to the local directory service. This function corresponds to the function of A-globe's *Visibility Service*. The second task of the service is intercepting messages going from the container by *send message hooks*. After the message interception, the receivers of the message are tested for reachability. If they are reachable from the originating container, the message is stored and a *Wireless Message Info* is extracted from the message, containing only the information about the sender, desired receivers, message length and the signal power of the message transmission. This message info structure is then sent to the server for processing. When the server responds back with a "message_ready" topic, the stored message is passed on to pruned receivers specified in the topics content. The mechanism is shown in Figure 6.3.

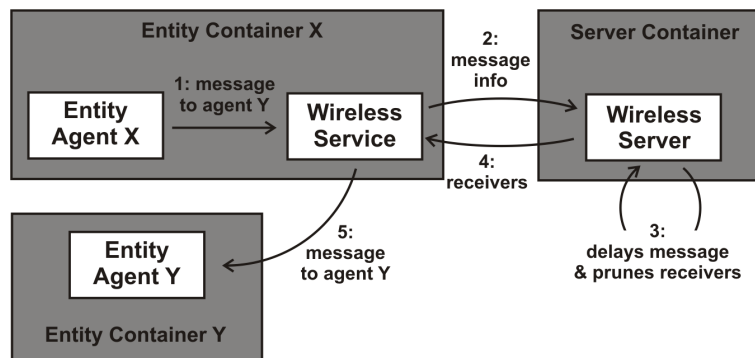


Figure 6.3: The mechanism of the message transport in *aglobex.wireless*

6.2 Station Reachability

As mentioned in section 5.4.1, standard A-globe container reachability is estimated in *Visibility Crash Simulator* depending on entities distances and the *communication range* parameter. The wireless module increases reachability estimation accuracy by taking the radio signal propagation, wireless characteristics like receiver's sensitivity, transmission power, channel's bandwidth, etc., and map of the environment into consideration.

The module, similar to *ns-2* simulator [33], defines three types of ranges, see Figure 6.4.

- The transmission range (*TX_Range*) is the range (with respect to the transmitting station) within which a transmission can be successfully captured. The range is mainly determined by transmission power, sensitivity threshold, and radio propagation properties.
- The interference range (*IF_Range*) is the range (with respect to the transmitting station) within which the other transmitting stations will be interfering with the particular transmitting station, thus suffering a decrease of SNR and possible loss of information.
- The physical carrier sensing range (*PCS_Range*) is the range (with respect to the transmitting station) within which the other stations can detect a transmission and postpone transmissions of their own.

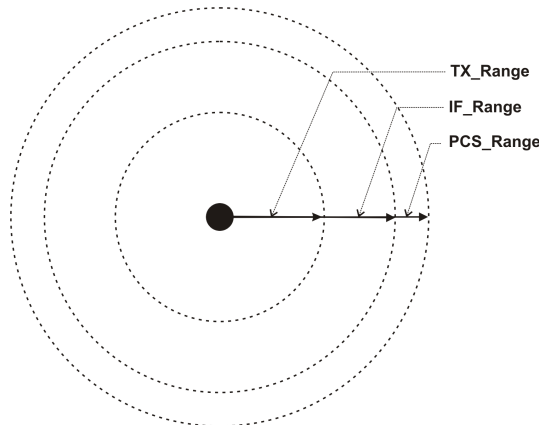


Figure 6.4: The ranges used in wireless simulation

The following relation exists between the ranges: $TX_Range \leq IF_Range \leq PCS_Range$. It helps in explaining the results obtained in experiments, e.g. [4]. In *aglobex.wireless* module the ranges are set as $PCS_Range = 2TX_Range$ and $IF_Range = 1.8TX_Range$. PCS_Range is set larger than the IF_Range to avoid simultaneous interfering transmissions.

To estimate the transmission range of a station, the module combines equations 2.5 and 2.9, giving

$$P_r(d) = \frac{P_t G_t G_r}{L} \left(\frac{\lambda}{4\pi d} \right)^\alpha \quad (6.1)$$

While we are trying to find the maximum distance at which the receiving stations will be able to receive the signal correctly, we should select the power at receiver P_r equal to the receiver's sensitivity threshold.

$$d = \sqrt[\alpha]{\frac{P_t G_t G_r}{S_{th} L} \frac{c}{4\pi f}} \quad (6.2)$$

where c is the speed of light in vacuum, f is the signal frequency and S_{th} is the sensitivity threshold of the receiver.

However, the sensitivity thresholds may vary for individual receivers and so in reality, there is no such thing as circular transmission range. We only roughly estimate the transmission range by assuming the receivers's sensitivity thresholds are the same and are equal to the sender's sensitivity threshold. That is why the *transmission*, *interference*, and *physical carrier sensing ranges* are not used for estimating stations' reachability. Instead, they are used only for time effective search for interfering hosts and detection of busy channel. The *transmission range* is also used for visualization of approximate transmission radius.

The actual stations' reachability is estimated individually for every host. There are two cases when containers are not mutually reachable. Either they are too far away from each other, so that the signal strength at receiver is lower than receiver's sensitivity threshold, or there is an obstacle between the hosts that is opaque for the radio signal. These two cases are discussed below.

6.2.1 Signal strength

The most common reason a pair of hosts cannot communicate with each other is their large distance. Because of the effects of phenomena discussed in the chapter 2, the radio frequency signal is attenuated with distance. Therefore, if two nodes are too far away from each other, the transmission signal will be too weak and the transmitted information will be lost.

The simulator computes signal power at receiver according to the equation 6.1 and if the power is lower than receiver's predefined *sensitivity threshold* the transmission fails. Another cause of transmission failure is noise - either a background noise or a signal from other simultaneously transmitting stations. When the signal-to-noise ratio (SNR) value decreases below receiver's predefined minimal required SNR value, the transmission fails as well.

Every time an entity (wireless host) moves, a reachability matrix containing link quality between each pair of hosts is computed. The link quality is a value representing maximum possible data rate in *Mbps* with that given two hosts can communicate with each other. It depends on two things. Firstly, the signal power at receiver is checked whether it is higher than the receiver's sensitivity threshold (according to the equation 6.1) and whether the SNR value is higher than the required SNR. If these conditions do not hold, the link quality is considered zero. The second thing that influences the link quality is the modulation data-rate. So far, the used data-rate is manually predefined in a configuration file, but in the future work, automatic adaptation of the data-rate according to SNR and signal strength at the receiver will be enabled.

If the link between two hosts has zero quality, the hosts are considered mutually unreachable.

6.2.2 Obstacles

Another reason for containers to be mutually unreachable for communication is an obstacle opaque for the radio frequency signal between them.

The module enables to define such obstacles in two ways (see Figure 6.5). The first way is to manually define a *zone*. These *zones* are cylinders with variable radius and infinite height which can be placed anywhere on the map of the simulated world. Their original use is to simulate the no-flight zones in the AGENTFLY project. An example of user defined zone is shown in Figure 6.5a.

The other way to define obstacles is height-map. The simulation scenario can be provided by an image that represents a 2D height-map of the world. According to the brightness levels of the image, the surface height moves between predefined minimal and maximal height and so called *surface zones* are created. This feature is again taken from the AGENTFLY project, where these zones are used to simulate the world's surface. By this mechanism, arbitrary shaped obstacles can be defined, see Figure 6.5b.

These zone-related obstacles provide line-of-sight reachability checking. However, as stated in the section 2.1, the radio frequency signal needs the whole first Fresnel zone to be unobstructed for an error free transmission. This feature is not supported in the *aglobex.wireless* module yet. Further, the module also does not support semi-opaque obstacles yet.

6.3 Message Transport Simulation

The module's principle of operation is to intercept entities' messages, postpone them according to PHY and MAC layer characteristics and then send them only to those receivers that are reachable. In the following I am about to closely describe the main

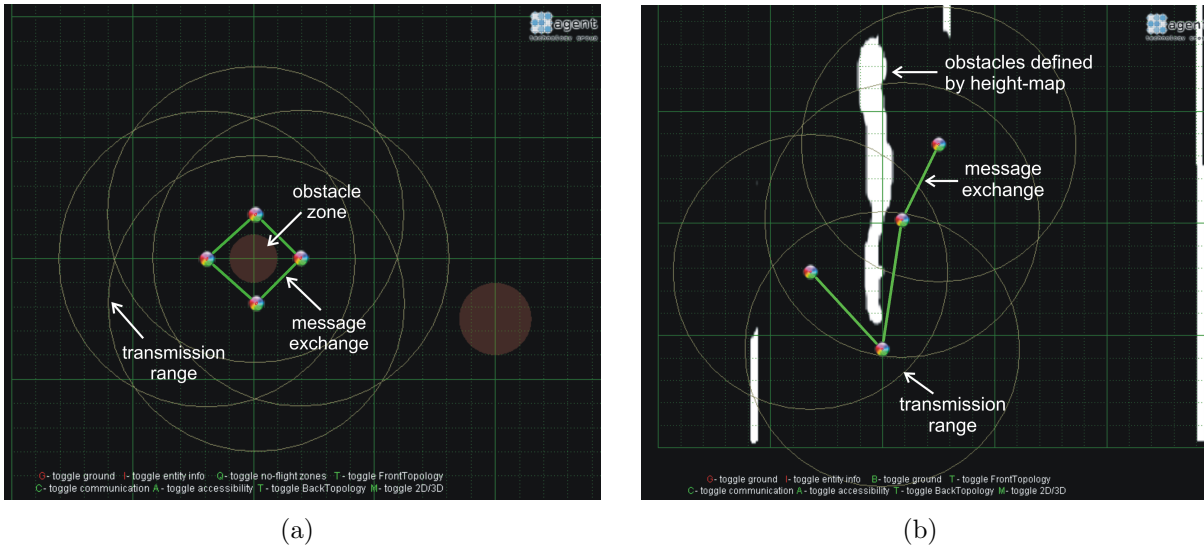


Figure 6.5: Examples of opaque obstacles: (a) user defined zone, (b) obstacles given by a height map

mechanism of the message processing.

Every incoming message is processed by the server in the following steps:

1. Receivers of the message are estimated according to their reachability.
2. Message is processed in MAC layer - algorithm shown in Listings 6.1 is applied.
3. Transmissions are tested for interference according to the algorithm shown in Listings 6.2
4. Unicast messages, receivers of which were removed in the previous steps are re-transmitted if their maximal retransmission limit was not exceeded.
5. Senders are informed about receivers the message can be sent to.

As mentioned before, the processing of messages at server side is scheduled in *INFORM_PERIOD* milliseconds intervals. The simulator processes hosts' messages and puts them one after another, simulating the backoff due to carrier sensing, until the sum of their durations reaches the *INFORM_PERIOD* time. The duration of message transport is estimated according to the equation 6.3. Because of this discrete behavior, the messages may be delayed little more than in reality. For example, even if a message arrives at the beginning of the *INFORM_PERIOD* lasting interval, it still cannot be sent sooner than at the end of the interval thus it is artificially delayed. Such delay can theoretically be in an interval from 0 milliseconds to $(INFORM_PERIOD - message_duration)$ milliseconds.

The placement of the messages into the *INFORM_PERIOD* interval is influenced by two things. Firstly, if transmitters are not in PCS_Ranges of each other, their messages may

overlap in time (several queues in current interval in Figure 6.6 that represent overlapping messages). And secondly, messages, that start in current interval but their duration overlaps to the next interval, are processed in the next interval, but their duration is lowered by the time, they spent in the current interval (see the overlapping messages in Figure 6.6).

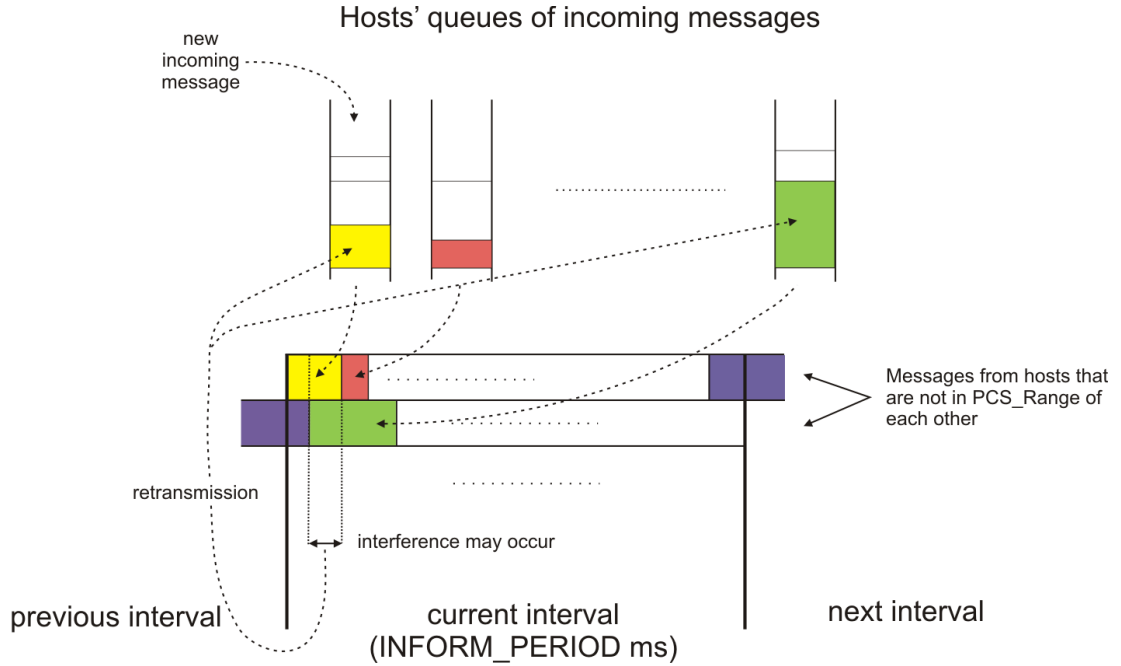


Figure 6.6: Basic scheme of the message transport simulation

6.3.1 Physical layer simulation

At the physical layer level, the simulator employs the log-distance path loss model according to the equation 6.1. This equation is used to verify the accessibility of all desired receivers by the mechanism of comparing signal strength at the receiver and receiver's sensitivity threshold described earlier. The path loss exponent, as well as, the sensitivity threshold, maximal transmission power and required SNR values can be defined in XML configuration files. In these files, the used channel bandwidth and antenna gain of the hosts can be specified too. According to the bandwidth, duration of the message transport can be counted as

$$t_M = \frac{size_M}{B_w C_o} \quad (6.3)$$

where t_M is the time in seconds, the message is processed,

$size_M$ is the size of the message in bites,

B_w is the channel's bandwidth in bits per second,

and C_o is a coefficient that represents the overhead of the link layer's and MAC layer's protocols.

6.3.2 MAC layer

At the MAC layer level, the simulator employs the 802.11 MAC protocol with the utilization of virtual carrier sensing and retransmissions of undelivered frames for unicast messages and classical carrier sensing for multi-recipient messages. However, because the A-globe's message transport does not support fragmentation of messages, neither the *aglobex.wireless* module fragments the intercepted messages. Instead, the messages are processed as solid packets.

The implemented MAC protocol mechanism is shown in the following pseudocode.

Listing 6.1: Implementation of MAC protocol

```

active_msg := find host whose first message to go
                has the smallest startTime;
while (active_msg != null)
{
  if ((active_msg.startTime) overlaps current interval)
  {
    return all messages in current interval;
  }
  // apply reachability information
  pruneReceivers(active_msg);

  // count start and end times of the active_msg
  duration := (count message transmission duration)
                - active_msg.time_in_process;
  active_msg.startTime := max(active_msg.startTime,
                              current_interval.startTime);
  active_msg.endTime := active_msg.startTime + duration;

  if((active_msg.endTime) not in current interval)
  {
    // the message overlaps to the next interval
    active_msg.time_in_process := interval_end_time
                                - active_msg.startTime;
  }
  else
  {
    add active_msg to the messages in current interval;
    remove active_msg from message queue of its sender;
  }

  // model the carrier sensing
  for each host_1 in sensingRange(sender(active_msg))

```

```

{
    // carrier sense and receivers of RTS
    delayed_msg := host_1.nextMessage;
    delayed_msg.startTime := active_msg.endTime + backoffTime();
    if (active_msg is unicast)
    {
        for each host_2 in transmissionRange(host_1)
        {
            // receivers of CTS
            delayed_msg_2 := host_2.nextMessage;
            delayed_msg_2.startTime := active_msg.endTime + backoffTime();
        }
    }
}
active_msg := find host whose first message to go
                    has the smallest startTime;
}

```

The `backoffTime()` function is implemented according to IEEE 802.11 specifications. It is a pseudo-random function that is influenced by the number of message retransmissions that causes the contention window to be incremented. More precisely, according to [21]

$$\text{backoffTime} = \text{Random}() \times \text{aSlotTime}$$

$\text{Random}()$ = Pseudo-random integer drawn from a uniform distribution over the interval $[0, CW]$, where CW is an integer within the range of values of the PHY characteristics $aCWmin$ and $aCWmax$, $aCWmin \leq CW \leq aCWmax$. CW is the contention window parameter which exponentially grows from initial value $aCWmin$ to maximum of $aCWmax$ depending on the number of retransmissions. See Figure 6.7.

$aSlotTime$ = The value of the correspondingly named PHY characteristic.

Furthermore, each host has its queue of messages waiting to be transmitted. If this queue (or buffer) exceeds its predefined capacity, new messages coming to this queue will be dropped. By this mechanism, the packet losses caused by buffer overflows in routers is simulated.

6.3.3 Interference and retransmission of messages

The signal interference is handled according to the two rules stated in 2.3. If two or more transmissions overlap in time and space, only the strongest one is captured by all the receivers. The rest is added to the background noise and affect the SNR value. If the SNR value is lower than the required SNR value, even the strongest signal transmission fails to be received.

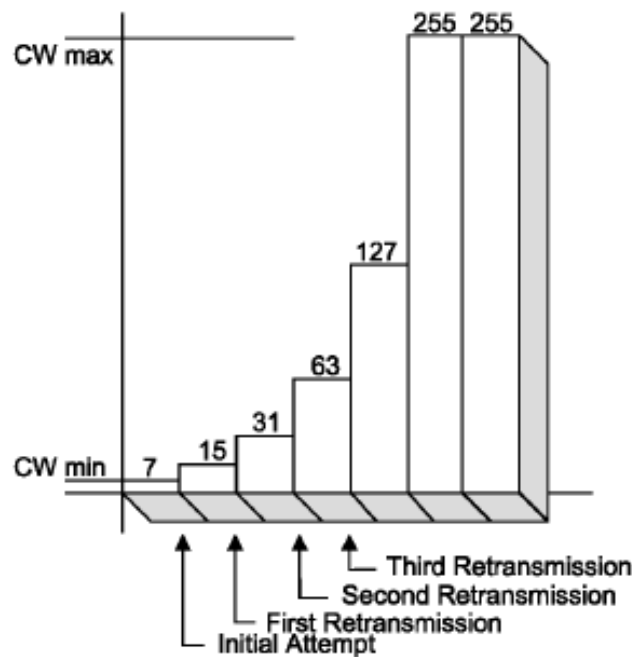


Figure 6.7: Example of exponential increase of CW, borrowed from [21]

The mechanism of estimating and handling signal interferences is shown in the following pseudocode:

Listing 6.2: Handling of signal interference

```

for each message M1 in current interval
{
  for each host in interferenceRange(sender(M1))
  {
    add signal strength of M1 to background noise of the host;
    for each message M2 in current interval other than M1
    {
      if (M1 and M2 overlap in time)
      {
        if(host is in interferenceRange(sender(M2)))
        {
          add signal strength of M2 to background noise of the host;
          if((M1 signal power at host > M2 signal power at host)
          {
            if(host is one of receivers of M2)
              remove host from receivers of M2;
          }
          else if(host is one of receivers of M1)
          {
            remove host from receivers of M1;
          }
        }
      }
    }
  }
}

```

```
        }
    }
}
if(host is still one of receivers of M1)
{
    // check the background noise
    if((M1 signal power)/(background noise - M1 signal power) < SNR)
        remove host from receivers of M1;
}
}
```

If a message fails to be delivered successfully, either because of the interference or because the receiving host moves out of the transmission range, there are two possible outcomes. Either the message is unicast and in that case it is retransmitted for several more times, thus simulating no arrived acknowledgements. After certain number of retransmissions, the message is considered as undeliverable and dropped. Or in the other case if the message is multi-recipient, there are no message acknowledgements and thus the undelivered message is lost.

7 Experiments

In this chapter, I am going to present the results of experiments that should verify the correct behavior of the *aglobex.wireless* simulator module. The experiments examined the wireless network simulation characteristics as well as the simulator's scalability and its computational resource requirements.

7.1 WLAN Simulation Characteristics

For testing of simulation characteristics, SWANS++ [2] simulator was used as a reference. SWANS++ is an extension to the SWANS simulator which was discussed in section 4.2. The main reasons for choosing this particular simulator were the following:

- SWANS++ is an open-source simulator with good users support and its authors still continue to develop new features.
- SWANS++ employs all the advantages of SWANS simulator like its accuracy (comparable to the accuracy of GloMoSim) and its simulation time management.
- The simulator is not too complex, thus changes of its behavior were easy to implement.
- The simulator code is written in Java so that there were no problems with its compilation.

The examined simulation properties were average delay in message delivery time, ratio of lost messages to all sent messages and the network throughput. All of these properties were measured as functions of the number of simulated transmitters.

7.1.1 Experiment's testbed description

The used testbed had following properties

- The modeled wireless transmitters were not moving for better comparison of the results, and they were placed in the modeled environment with 100 *m* spacing, according to the placement shown in Figure 7.1.
- Free space path loss model was used and there were no obstacles placed in the environment.
- The transmitters used 2 *GHz* frequency with 2 *Mbps* bandwidth.
- The transmission power was set to 1 *dBm*, antenna gain to 1 *dBi* and sensitivity threshold of the receivers was set to -81 *dB*. Thus, the transmission range encompassed only the nearest neighbors (as shown in Figure 7.1).
- The hosts were sending 1000 byte unicast messages to their neighbours with rate of 5 messages per second for the time of 5 minutes.

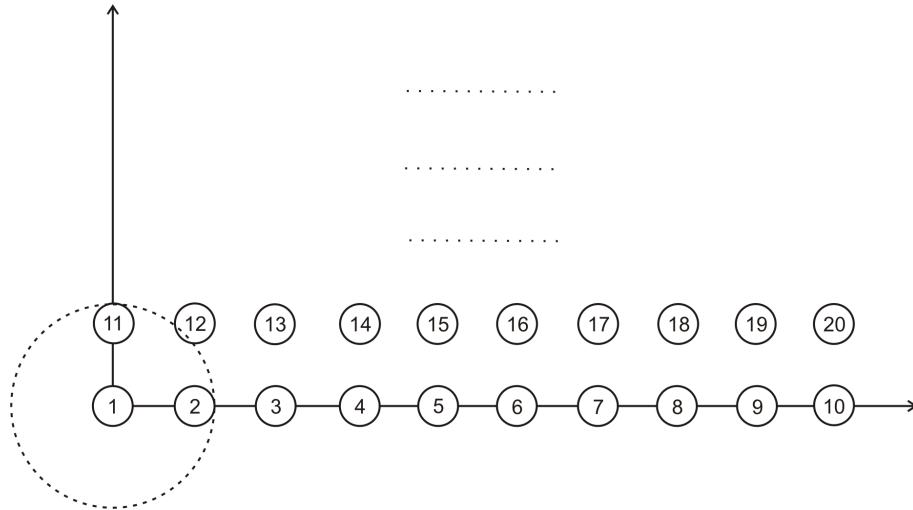


Figure 7.1: Nodes' placement used for the testing of simulator's characteristics

While *aglobex.wireless* provides simulation of first two layers (the physical and MAC layers) and SWANS++ in addition employs routing protocols, some minor changes needed to be done in the SWANS++ code - each transmitter was configured to send its messages only to one of its nearest neighbors so that no routing overhead was necessary.

Moreover, although the SWANS++ does not employ the log-distance path loss model, the measured hosts' transmitting ranges in both simulators, using the free space path loss models, were almost the same. The small difference in the ranges was caused by the background noise which SWANS++ simulator estimates from the temperature of the environment and which is not so significant to be considered in the *aglobex.wireless* module.

For better comparison of the simulators' behaviors, two modes of SWANS++ were tested - one using RTS/CTS mechanism in message exchange (marked as *swans RTS* in the following graphs) and the second one using no RTS/CTS frames (marked as *swans no_RTS* in the following graphs).

7.1.2 Results

7.1.2.1 Message delivery delay

As first, I measured the average time that was necessary for message transport. The delay between the message sending and its reception is influenced mainly by the channel utilization, i.e. how often the transmitters sense an idle channel and are allowed to transmit a message, and by the number of necessary retransmissions evoked by message losses due to interference (and in general case also by the receiver moving out of the sender's transmission range).

The graph of average delays in message delivery depending on the number of wireless hosts present in the scenario is shown in Figure 7.2.

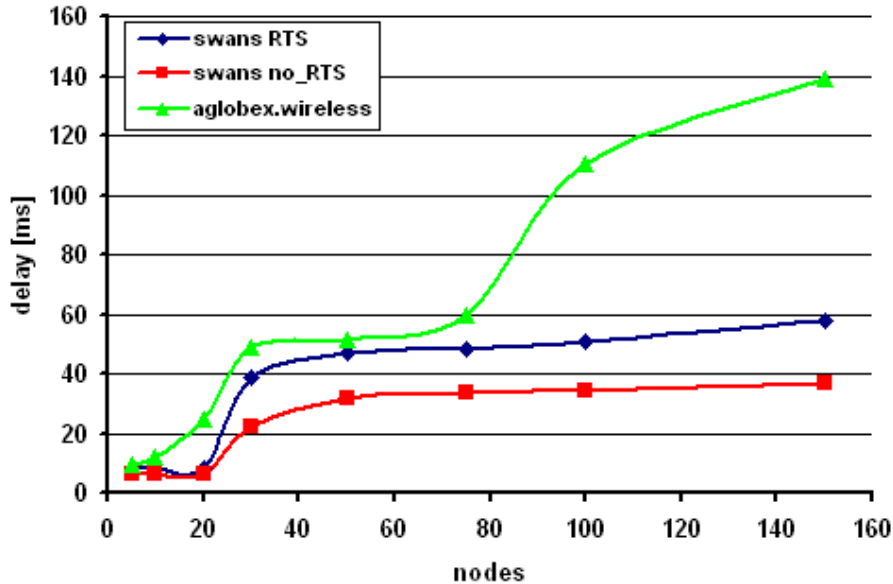


Figure 7.2: Average delivery delays of messages in milliseconds

It is noticeable that up to about 75 transmitters, the characteristics of *aglobex.wireless* module and SWANS++ simulator are similar, however with growing number of transmitters, the module's delays grow much more than the ones of SWANS++. This is caused by the fact that SWANS++ drops messages that are about to be sent by the MAC layer when the MAC protocol is not in the idle mode (i.e. the MAC layer is sending a frame or is expecting a frame). Instead, the *aglobex.wireless* module is queuing all the messages which causes the average delivery delay to accumulate. This effect can be suppressed by decreasing the capacity of the module's incoming message buffer.

The reason for slightly larger delays in delivery of the module's messages with lower number of transmitters in the scenario, is the artificial message delaying caused by the discrete nature of module's message handling which was discussed in the section 6.3.

7.1.2.2 Ratio of the lost messages

When a message is lost due to interference or receiver's movement out of the sender's transmission range, it is retransmitted for several times and if it still does not succeed to be delivered, it is dropped. The ratio of such dropped messages to all sent messages depending on the number of wireless hosts present in the scenario is shown in Figure 7.3.

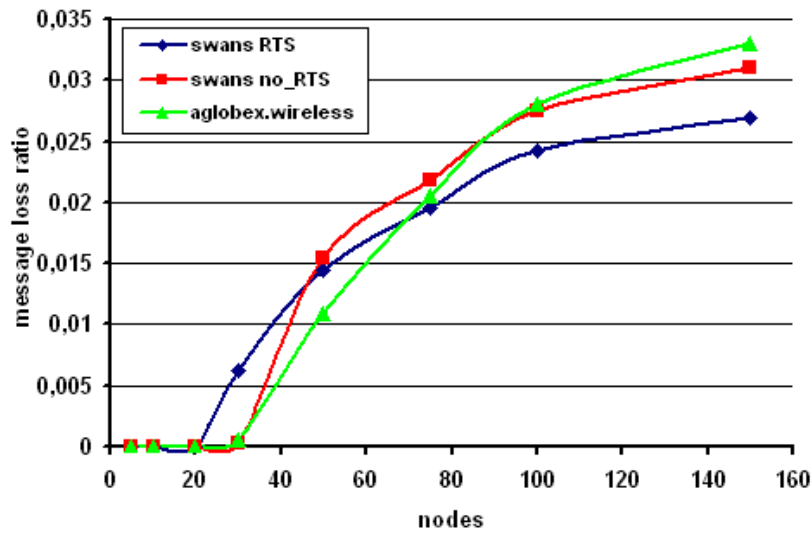


Figure 7.3: Ratio of lost messages

7.1.2.3 Network throughput

Finally, the overall network throughput was measured as a function of the number of transmitters. While the messages have size of 1000 bytes, the maximal throughput can be estimated as

$$throughput[kBps] = \frac{\text{max_}\# \text{ of messages sent in 5 min}}{5 \cdot 60}$$

With small number of the transmitters, the wireless channel manages to transmit all the sent messages even with some retransmissions because the hosts' sent rate is quite low. Thus the throughput of the network is equal to the maximal network throughput for the given number of nodes. With increasing the number of transmitters, the number of lost and dropped messages increases as well and the channel can no more send all the messages and the network throughput decreases.

The results of the simulation are shown in Figure 7.4.

The *aglobex.wireless* module behavior produces slightly higher network throughput than SWANS++ with smaller number of transmitters but it slowly approaches the SWANS++'s throughput with increasing number of nodes.

7.2 Computational Resource Requirements

Besides the accuracy of the wireless model, the simulator scalability is another important issue. To estimate the *aglobex.wireless* module's scalability, I measured CPU time consumption and memory requirements of the simulation with the module and compared

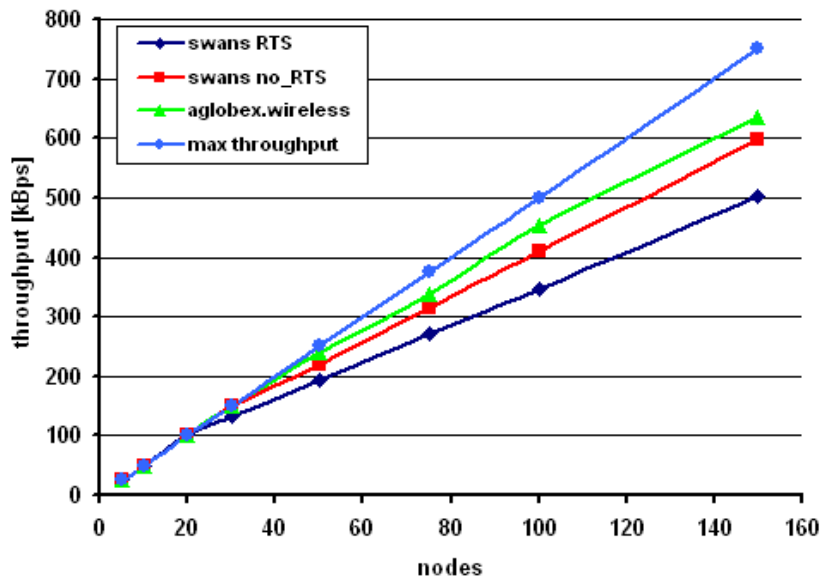


Figure 7.4: Network throughput in individual simulators in *kbps*

them to the requirements of the original simulation run without the module. The testing PC was Intel Core2 Duo CPU at 1,86GHz with 3GB RAM.

For the testing, the same settings as in the previous experiments were used, only the message size was lowered to 8 bytes, so that new incoming messages would be processed faster and would not make large queues.

The requirements of resources depending on the number of nodes are shown in Figures 7.5 where the simulation was running with visualisation enabled on the same computer and in Figure 7.6 where the visualisation was disabled on the simulating computer and was running on a different computer.

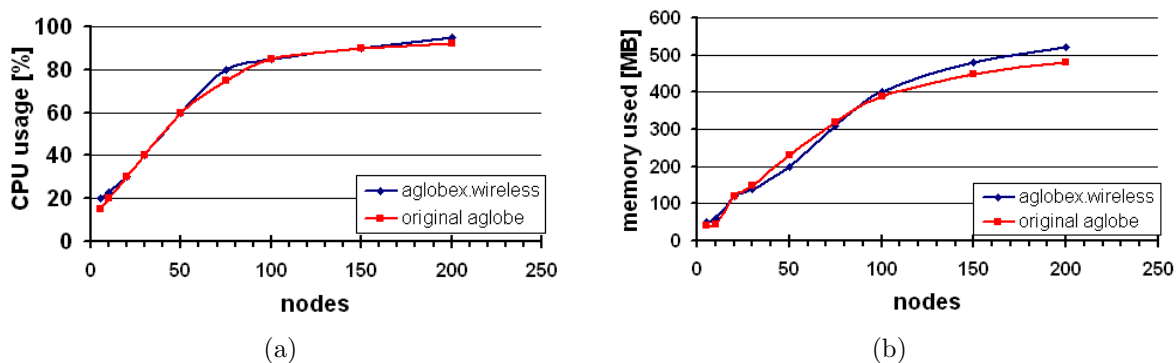


Figure 7.5: Comparison of the CPU time (a) and memory requirements (b) with visualisation enabled

The results show that with visualisation, the simulation with the wireless module has

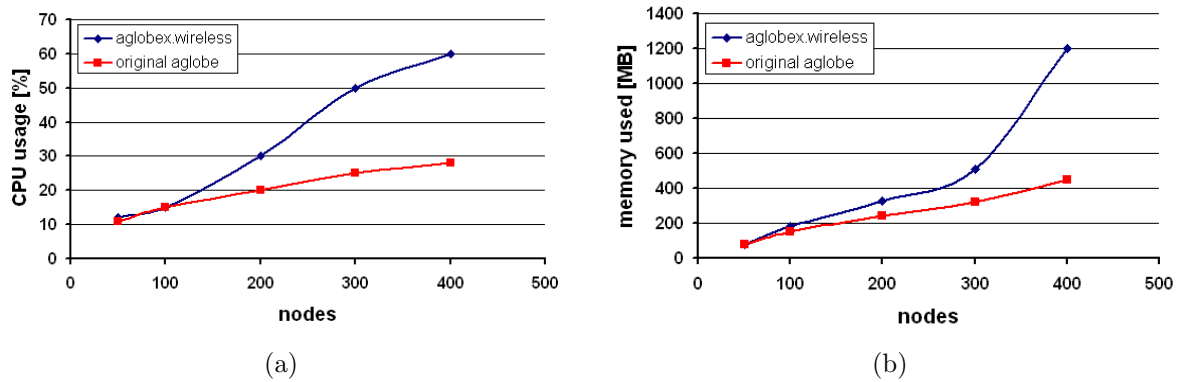


Figure 7.6: Comparison of the CPU time (a) and memory requirements (b) with visualisation disabled

only slightly worse but similar scalability as the original A-globe's simulation without it. However, only up to 200 nodes could be simulated with the visualisation enabled on the same computer and CPU time requirements almost reach 100%. The visualisation of the communication took most of the processing time.

With the visualisation running on a different computer, the requirements of the *aglobex.wireless* module were conspicuously higher than those of original A-globe simulation. This was caused by the fact, that from certain number of simulated nodes, the module was unable to manage modelling of the communication in the *INFORM_PERIOD* interval. Thus, the cached information and memory objects could not be released, the memory requirements rose and the overall performance dropped.

To overcome this scalability problem some optimization of the module's code will need to be done.

8 Conclusions

In this work, I presented a new wireless network simulator capable of modelling first two network layers - the physical layer and medium-access control layer. This simulator was designed as a plug-in module (called *aglobex.wireless*) to multi-agent platform A-globe and as such, it utilizes its multi-agent principles and its messaging and accessibility mechanisms.

The simulation is based on a hybrid architecture - it is triggered by discrete events, yet it enables co-simulation of the network communication and nodes' computations in real time.

The simulator enables users to specify the communication data rate, signal propagation characteristics of the environment in the form of *distance-power gradient* or background noise, and each simulated wireless host can be provided by particular transmitting powers, antenna gains, receiving sensitivity thresholds and required minimal SNR values. Capacity of the hosts' incoming message buffer can be specified as well to simulate packet losses caused by buffer overflows.

The accuracy performance of the simulator was tested against an open-source simulator SWANS++, which is comparable to GloMoSim [42], one of the best known WLAN simulators, in the terms of accuracy. In the test, both simulators achieved very similar results, meaning the *aglobex.wireless* simulator should provide the A-globe simulations with precise enough wireless simulation facility.

Thanks to its multi-agent nature, the module supports parallel computations and enables distribution of the simulation on distinct computers, thus considerably improving the simulation scalability.

Future Work

Even though the simulation can be distributed to numerous distinct computers thus providing very good scalability, its consumption of computational resources is much higher than that of A-globe. In the future work, optimization of the module's code will be necessary so that it would not slow down A-globe's simulations.

Furthermore, new features that were mentioned in previous chapters can be implemented, such as adaptation of the communication data-rate, consideration of Fresnel zones for estimation of the quality of received signal, various opacity levels of the obstacles, or use of directional antennas by the simulated entities.

In the nearest future, the *aglobex.module* is planned to be deployed on some of the A-globe's projects.

9 Bibliography

- [1] M. Andersson, D. Henriksson, A. Cervin, and K. Arzen. Simulation of wireless networked control systems. *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 476–481, 2005.
- [2] AquaLab. Swans++ - extensions to the scalable wireless ad-hoc network simulator. online: <http://www.aqualab.cs.northwestern.edu/projects/swans++>.
- [3] R. Barr. An efficient, unifying approach to simulation using virtual machines. *PhD thesis*, 2004.
- [4] S. Basagni, M. Conti, S. Giordano, and I. Stojmenović. *Mobile Ad Hoc Networking*. Wiley - Interscience, 2004.
- [5] F. Bellifemine, G. Rimassa, and A. Poggi. Jade - a fipa-compliant agent framework. *Proceedings of 4th International Conference on the Practical Applications of Intelligent Agents and Multi-Agent Technology, London*, 1999.
- [6] C. Bettstetter, G. Resta, and P. Santi. The node distribution of the random waypoint mobility model for wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, pages 257–269, 2003.
- [7] S. Bohacek and V. Sridhara. The udel models - manet mobility and path loss in an urban. (*In Submission*), 2004.
- [8] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, pages 483–502, 2002.
- [9] D. Cavin, Y. Sasson, and A. Schiper. On the accuracy of manet simulators. *POMC '02: Proceedings of the second ACM international workshop on Principles of mobile computing*, pages 38–43, 2002.
- [10] J. Chen and J. Gilbert. Measured performance of 5 ghz 802.11a wireless lan systems. *Atheros Communications White Paper*, 2001. <http://www.atheros.com/pt>.
- [11] T. Chu and I. Nikolaidis. On the artifacts of random waypoint simulations. *First International Workshop on Wired/Wireless Internet Communications*, 2002.
- [12] F. Desbrandes, S. Bertolotti, and L. Dunand. Opnet 2.4: An environment for communication network modeling and simulation. *Proceedings of European Simulation Symposium. Society for Computer Simulation*, pages 64–74, 1993.

- [13] Fipa: Foundation for intelligent physical agents. online: <http://www.fipa.org>.
- [14] Fipa-acl: Fipa agent communication language overview. foundation for intelligent physical agents, (2009). online: <http://www.fipa.org/specs/fipa00037>.
- [15] M. Grossglauser and D. Tse. Mobility increases the capacity of ad-hoc wireless networks. *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 1360–1369, 2001.
- [16] P. Gupta and P. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, pages 388–404, 2000.
- [17] J. Heidemann, N. Bulusu, J. Elson, C. Intanagonwiwat, K. Lan, Y. Xu, W. Ye, D. Estrin, and R. Govindan. Effects of detail in wireless network simulation. *Proceedings of the SCS Multiconference on Distributed Simulation*, pages 3–11, 2001.
- [18] H. Hellbrück and S. Fischer. Towards analysis and simulation of ad-hoc networks. *ICWN02: Proceedings of the International Conference on Wireless Networks*, pages 69–75, 2002.
- [19] H. Hogie, P. Bouvry, and F. Guinand. An overview of manets simulation. *Electronic Notes in Theoretical Computer Science 150*, pages 81–101, 2006. Online: www.elsevier.com/locate/entcs.
- [20] X. Hong, M. Gerla, G. Pei, and C. Chiang. A group mobility model for ad hoc wireless networks. *Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 53–60, 1999.
- [21] Ieee standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements, part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications, 2007. online: <http://standards.ieee.org/>.
- [22] D. Šišlák, M. Pěchouček, P. Volf, D. Pavlíček, J. Samek, V. Mařík, and P. Losiewicz. Agentfly: Towards multi-agent technology in free flight air traffic control. *AgentsDefence Industry Application of Autonomous Agents and Multi-Agent Systems*, 2008. Birkhäuser Verlag Publishing.
- [23] D. Šišlák, M. Reháč, M. Pěchouček, M. Rollo, and D. Pavlíček. A-globe: Agent development platform with inaccessibility and mobility support, 2004. online: <http://agents.felk.cvut.cz/cgi-bin/docarc/public.pl>.
- [24] D. Šišlák, M. Rollo, and M. Pěchouček. A-globe manual v1.1. online: <http://agents.felk.cvut.cz/aglobe>.

- [25] S. Imre, Keszei, Horváth, Hollós, Barta, and Kujbus. Simulation environment for ad-hoc networks in omnet++. *IST Mobile Summit 2001*, pages 135–140, 2001.
- [26] E. I. S. Institute. The nab (network in a box) wireless network simulator. online: <http://nab.epfl.ch>.
- [27] A. Jardosh, E. Belding-Royer, K. Almeroth, and S. Suri. Real world environment models for mobile ad hoc networks. *Journal on Special Areas in Communications - Special Issue on Wireless Ad hoc Networks*, 2005.
- [28] A. Kochut, A. Vasan, A. Shankar, and A. Agrawala. Sniffing out the correct physical layer capture model in 802.11b. *12th International Conference on Network Protocols (INCP'04)*, 2004.
- [29] N. U. C. Laboratory. Javasilim's users guide. online: <http://javasilim.ncl.ac.uk>.
- [30] J. Lehnert, D. Görgen, H. Frey, and P. Sturm. A scalable workbench for implementing and evaluating distributed applications in mobile ad hoc networks. *WMC'04: Western Simulation MultiConference*, pages 154–161, 2004.
- [31] J. Lessmann, P. Janacik, L. Lachev, and D. Orfanus. Comparative study of wireless network simulators. *Seventh International Conference on Networking*, pages 517–523, 2008.
- [32] G. Lin, G. Noubir, and R. Rajaraman. Mobility models for ad hoc network simulation. *IEEE INFOCOM 2004*, 2004.
- [33] The network simulator. ns-2. online: <http://www.isi.edu/nsnam/ns>.
- [34] H. Nwana, D. Ndumu, L. Lee, and J. Collis. Zeus: A tool-kit for building distributed multi-agent systems. *Applied Artificial Intelligence Journal* 13, pages 129–186, 1999.
- [35] S. Poslad, P. Buckle, and R. Hadingham. The fipa-os agent platform: Open source for open standards. *Proceedings of 5th International Conference on the Practical Applications of Intelligent Agents and Multi-Agent Technology, Manchester*, pages 355–368, 2000.
- [36] T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, Upper Saddle River, NJ, second edition, 2002.
- [37] M. Reháč, M. Pěchouček, K. Bartoš, M. Grill, P. Čeleda, and V. Krmíček. Camnep: An intrusion detection system for high-speed networks. *International Conference on Autonomous Agents*, 2008.

- [38] M. Rollo, P. Novák, and P. Jisl. Simulation of underwater surveillance by a team of autonomous robots. *HOLOMAS*, 2005.
- [39] P. Santi. *Topology Control in Wireless Ad Hoc and Sensor Networks*. John Wiley & Sons, Ltd., 2005.
- [40] E. Schoch, M. Feiri, F. Kargl, and M. Weber. Simulation of ad hoc networks: ns-2 compared to jist/swans. *Proceedings of the First International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools 2008)*, 2008.
- [41] J. Tian, J. Hahner, C. Becker, I. Stepanov, and K. Rothermel. Graph-based mobility model for mobile ad hoc network simulation. *Annual Simulation Symposium*, pages 337–344, 2002.
- [42] X. Zeng, R. Bagrodia, and M. Gerla. Glomosim: A library for parallel simulation of large-scale wireless networks. *Workshop on Parallel and Distributed Simulation*, pages 154–161, 1998.

A Content of the enclosed CD

The CD attached to this work contains source codes of the *aglobex.wireless* module as well as other files required for running a demo of wireless communication. Further, there is the module's documentation, LaTeX codes of this work and the work itself as a pdf file.

- Directory `source_codes` – contains source codes of the *aglobex.wireless* module and other A-globe projects required for running a wireless communication demo. The demo can be started by running the `test.bat` file.
- Directory `javadoc` – contains module's Java documentation.
- Directory `latex` – contains LaTeX files of this work.