UPC

epsc Escola Politècnica Superior de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

MASTER THESIS

TITLE: Evaluation of the impact of mobility models on handover in WLAN indoor environments

MASTER DEGREE: Master in Science in Telecommunication Engineering & Management

AUTHOR: Jesús Martínez Navarro

DIRECTOR: Enrica Valeria Zola

DATE: January, 20th 2009

Title: Evaluation of the impact of mobility models on handover in WLAN indoor environments

Author: Jesús Martínez Navarro

Director: Enrica Valeria Zola

Date: January, 20th 2009

Overview

Nowadays, mobility models are used to simulate realistic movements produced by the users of a wireless or a mobile network.

The aim of this project is to evaluate the impact of mobility models on handover process in Wireless Local Area Networks (WLAN) indoor environments, using the network simulator NS-2.

This document contains a previous theoretical characterization of the basic mobility models and their application in the network simulator NS-2, a study of the infrastructure mode support and a modification of the current handover algorithm.

The study focuses on two mobility models in WLAN indoor environments: one where movements are completely random (i.e. Random Waypoint) and another one where next step depends on previous movements (i.e. Gauss-Markov). In order to support infrastructure mode operation in WLAN, a new patch with modifications of the source code is applied. Furthermore, a first approach for the development of a new handover algorithm is presented.

Simulations are run in two different scenarios: one with 4 Access Points (APs) offering full coverage in the simulated area, and another one with 8 APs, simulating overcoverage in order to guarantee higher capacity for a higher density of users. Results are presented showing a comparative of the cell residence time (CRT) for each scenario. From the results obtained, it is possible to see that the CRT changes depending on the algorithm and the mobility model applied, being lower when the Random Waypoint model is applied. On the other hand, the new handover algorithm designed from the current implementation leads to a decrease in the average CRT.

INDEX

Acronyms list

ACK → Acknowledgement
ACRP → Association Reply
ACRQ → Association Request
AP → Access Point
AUTH → Authentication
BCN → Beacon frame
BSS → Basic Service Set
BSSID → Basic Service Set Identifier
CV → Coefficient of variation
CONSER → Collaborative Simulation for Education and Research
CRT → Cell Residence Time
DARPA → Defense Advanced Research Projects Agency
ESS → Extended Service Set
ESSID → Extended Service Set Identifier
FCC → Federal Communications Commission
GM → Gauss-Markov
HO → Handover
IBSS → Independent Basic Service Set
ISM → Industrial, Scientific and Medical band
LAN → Local Area Network
NIC → Network Interface Card
NS → Network Simulator
RSS → Received Signal Strength
RWP → Random Waypoint
SAMAN → Simulation Augmented by Measurement and Analysis for Networks
STA → Station
UDP → User Datagram Protocol
WLAN → Wireless Local Area Network

INTRODUCTION

The aim of this project is to evaluate the impact of mobility models in WLAN indoor environments on handover process, running simulations in infrastructure mode and providing a new handover algorithm.

The project is divided into four sections. The first chapter is an introduction to WLANs and mobility models. The second section contains information about the network simulator NS-2 and the modifications applied to achieve the infrastructure mode. Firstly, the current 802.11 implementation in NS-2 will be checked, applying a new patch and changing the source code in order to achieve infrastructure mode simulations. This operation mode will be tested in different configurations.

Chapter 3 is dedicated to the study of handovers and the scenarios applied in the final simulations. Two different scenarios will be created with 4 APs and 8 APs for the simulations. The scenario with 8 APs simulates a network where the number of users rises and it is necessary to offer more capacity. Furthermore, two mobility models will be applied: Random Waypoint and Gauss-Markov. In the first one, movements are completely random and in the second one, next step depends on previous movements. Results are displayed in chapter 4, showing comparative tables and graphics of the average cell residence time and its distribution for each scenario.

Finally, some conclusions will be exposed examining the results obtained to describe the impact that these models have on the network parameters, and adding possible future work related to the current project.

CHAPTER 1. 802.11 WLANs

1.1. Introduction and history

Wireless Local Area Network (WLAN) is a communication system of wireless data, commonly used as an alternative to the wired LANs or as an extension of them. It uses radiofrequency technology that allows more mobility to the user because of the absence of cables.

WLANs are subject to interference from other electromagnetic sources and can be affected by other interference which causes performance degradation. The power usage is limited, too.

The development of radio-based LAN components was possible in 1985, when the Federal Communications Commission (FCC) authorized the public use of the Industrial, Scientific and Medical (ISM) bands; between 902 MHz and 5.85 GHz. People began developing radios and access points without a standard [1].

And in the late 1980s, the IEEE 802 working group developed the Wireless LAN (Local Area Network) Medium Access Control and Physical Layer specifications. The standard was published in 1997, concretely on November 18th.

1.2. Architecture

The 802.11 standard defines two operating modes in a wireless network: Ad-Hoc mode and infrastructure mode. While the infrastructure mode setups devices to communicate through an access point, the Ad-Hoc mode works with a direct communication between them.

1.2.1. Ad-Hoc mode

Ad-Hoc mode allows the radio network interface card (NIC) to operate in an Independent Basic Service Set (IBSS) network configuration [2]. As it does not requires access points, devices communicate directly with each other in a peer-to-peer mode. Working on this mode, it is possible to form a wireless LAN by only using the devices to connect, without access points or cables.

The use of Ad-Hoc mode does not affect the physical layer because only affects the protocols. For that reason, the MAC Layer has all the responsibility. During the process, the active Ad-Hoc station establishes an IBSS and start sending beacons. These frames are needed for the synchronization. The other Ad-Hoc stations, once receive a beacon, they can accept the IBSS parameters and join the network. All stations that join the network must send a beacon frame periodically if it does not hear a beacon from another station.

This form of communication is useful in medical applications that require fast communications and there is no time to connect cables and devices.

Figure 1.1 shows how the Ad-Hoc mode works.

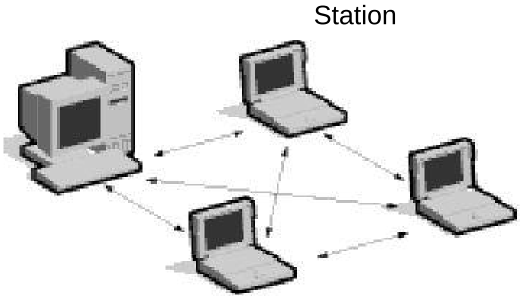

Fig 1.1 WLAN in Ad-Hoc mode [3]

1.2.2. Infrastructure mode

Infrastructure mode allows wireless devices to communicate with each other or to communicate with a wired network.

The Basic Service Set (BSS) is the basic building block, composed by one AP together and its associated stations (STAs) [2]. The AP controls the stations within that BSS. Each BSS is identified by a BSSID (Basic Service Set Identifier), which corresponds to the access points' MAC address. An Extended Service Set (ESS) is a set of two or more BSSs that form a single subnetwork, and it is identified by an ESSID (Extended Service Set Identifier), which show the network's name

The cell sends a *probe request* when a station joins it, which contains the ESSID of the cell. The AP sends beacon frames (giving its BSSID, characteristics ...) and checks the ESSID. The next step for the AP is to send a response with data and information on its traffic load, whenever the given ESSID matches the traffic volume request found in the beacon.

As most corporate wireless LANs requires access to the wired LAN, they operate in this mode in order to use these services.

In figure 1.2 there is an illustration of the infrastructure mode.

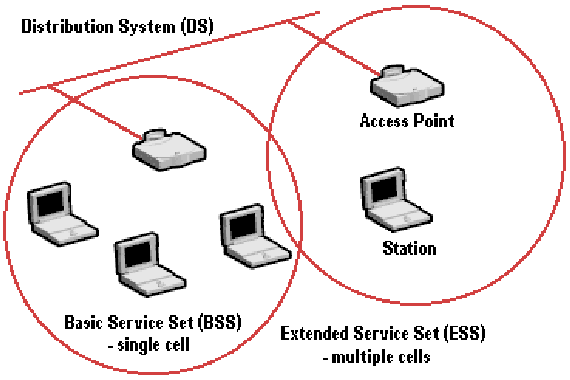

Fig. 1.2 WLAN in Infrastructure mode [3]

1.2.3. Ad-Hoc mode vs. infrastructure mode

As a summary, the main differences between the two modes are the next [4]:

- Ad-Hoc mode is easier to set up due it is not necessary an AP.

- The high power of an AP is used by the infrastructure mode to cover wide areas, while Ad-Hoc mode connections are limited.

- Changes in an Ad-Hoc topology can influence on the throughput. Learning to manage weaknesses or wireless strengths with infrastructure mode is easier to new users.

- In an Ad-Hoc network, as each device is trying to use the same channel, it will cause more interference than in an infrastructure network.

- Network delays is another problem in the Ad-Hoc mode, since there is no control over the path the data takes and the data is sent through other computers.

1.3. Mobility models

Mobility models are used to describe the movements of the mobile nodes at every moment, defining their position, speed and acceleration. According to the dependence of the movements, these mobility models can be classified in Entity mobility models (independent movements) or Group mobility models (dependent movements).

The main Entity mobility models are: Random Walk, Random Waypoint, Random Direction, a Boundless Simulation Area, Gauss-Markov and City Section. On the other hand, the Group mobility models are: Exponential

Correlated Random, Column Mobility Model, Nomadic Community Mobility Model, Pursue Mobility Model and Reference Point Group Mobility Model.

This Master Thesis only works with Entity mobility models, concretely with Random Waypoint Model and Gauss-Markov model.

1.3.1. Random Waypoint Mobility Model

In the Random Waypoint model [5], a node selects a random position and moves along a zigzag line from one point to another one. As this pattern does not have memory, every new movement is different from the previous one, choosing a new velocity and direction. It is possible to include pause times between the movements.

A realistic application of the Random Waypoint model could be any room where people move randomly: an airport, a library, etc.

Next figure, 1.3, illustrates how a mobile node moves when applying the Random Waypoint model.

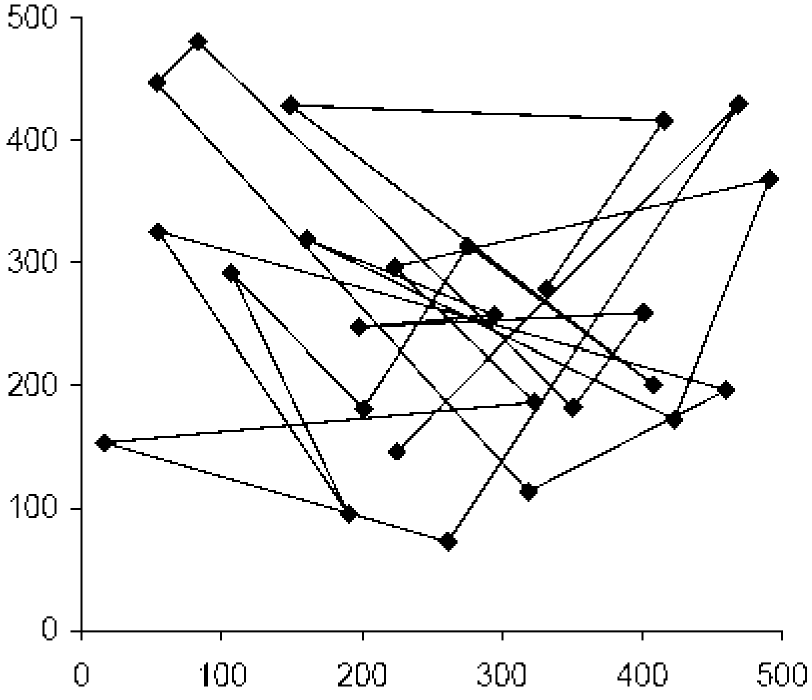

Fig. 1.3 Random Waypoint model [5]

1.3.2. Gauss-Markov Mobility Model

This mobility model adapts different levels of randomness by using a tuning parameter [4]. Once the MN is fixed with a speed and direction, movements occur at fixed intervals of time *n*, updating the two parameters. However, the new values are calculated according the previous values at *n-1* instance and a random variable. The next equations are followed by this model:

$$s_n = \alpha s_{n-1} + (1-\alpha)\bar{s} + \sqrt{(1-\alpha^2)}s_{x_{n-1}} \qquad \textbf{(1.1)}$$

$$d_n = \alpha d_{n-1} + (1-\alpha)\bar{d} + \sqrt{(1-\alpha^2)}d_{x_{n-1}} \qquad \textbf{(1.2)}$$

SS_n and d_n are the new speed and direction. The random values are obtained by changing the value of α, the tuning parameter, between 0 and 1. While a null value of the tuning parameter causes totally random values, α = *1* leads to a linear motion. As for the new locations calculated depending on the previous location, follow the next formula:

$$x = x_{n-1} + s_{n-1}\cos d_{n-1} \qquad \textbf{(1.3)}$$

$$y = y_{n-1} + s_{n-1}\sin d_{n-1} \qquad \textbf{(1.4)}$$

It is important to note that MNs are forced to not remain near the edges of the grid. Figure 1.4 shows an example of a MN moving according this pattern.

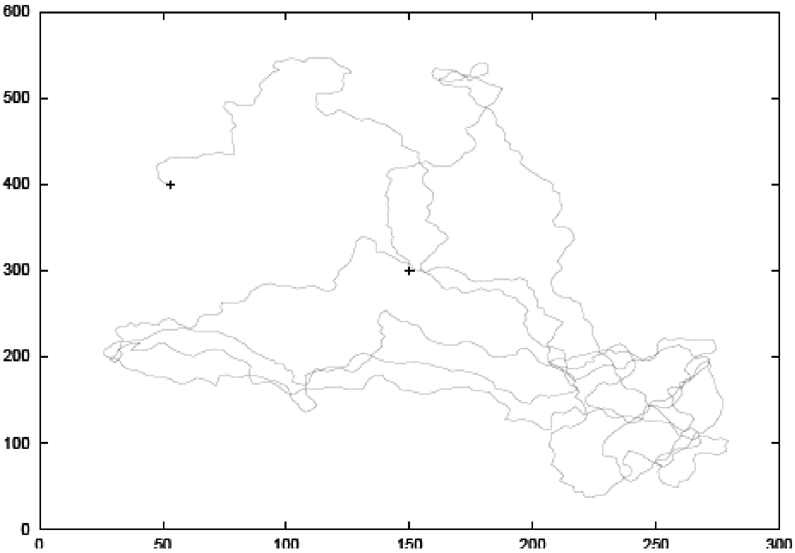

Fig. 1.4 MN using the Gauss-Markov Mobility Model [5]

As can be seen in the distribution, this model eliminates the sudden stops.

CHAPTER 2. NETWORK SIMULATOR NS-2

2.1. Introduction

The simulator chosen for this project is the network simulator NS-2, version 2.31 [6].

NS is a discrete event simulator targeted at networking research which provides support for simulation of TCP, routing, and multicast protocols both wired and wireless networks. This is probably the most extended simulator in research and education.

REAL network simulator is the origin of ns. Currently, ns development is support through DARPA (Defense Advanced Research Projects Agency) with SAMAN (Simulation Augmented by Measurement and Analysis for Networks) and through NSF with CONSER (Collaborative Simulation for Education and research). However, NS is not a finished product and users develop the software every day.

The simulator consists of a core written in C++ which can be run simply by typing ns in the command line. The specific interface oTcl, object-oriented Tcl, is used to act on the simulator. While C++ handles a big volume of data in a short execute time, oTcl reconfigures simulation scenarios.

2.2. Installation of NS-2

The first requirement to build ns is to have a C++ compiler [6]. There are two ways to build ns: from all the pieces or all at once. The allinone package requires 320 MB of disk space, so building ns from all the pieces can save some disk space.
Ns-2 can run in both Windows and Linux.

The first step to install NS-2 is to download the compressed file ns-allinone-2.31.tar.gz, available in the web page. This file contains all the libraries needed for the simulator. These packages are the next:

- Tcl release 8.4.14
- Tk release 8.4.14
- Otcl release 1.13
- TclC release 1.19
- Ns release 2.31
- Nam release 1.13
- Xgraph version 12
- CWeb version 3.4g
- SGB version 1.0
- Gt-im gt-itm y sgb2ns 1.1

- Zlib version 1.2.3

Once the ns-allinone-2.31.tar.gz has been downloaded, the next step is to uncompress it using the command:

```
tar –zxvf ns-allinone-2.31
```

And the generated folder ns-allinone-2.31 will contain the installation script, which have to be executed:

```
cd ns-allinone-2.31
./intall
```

Finally, the environment variables have to be added to the PATH.
Additionally, to check possible installation errors, the next command can be useful.

```
./validate
```

At this moment, to check that there is any problem with the installation, it is useful to execute an example:

```
cd home/ns-allinone-2.31/NS-2.31/tcl/ex
ns simple.tcl
```

2.3. TCL script example

In this section, it is going to be created a simple example to simulate a basic topology, setting up nodes and links, sending data from one node to another, and starting a *nam* simulation to visualize it [7].

The tcl script can be written in any text editor like Emacs or Kate. This first example will be named “example.tcl”.

The first step is to create a simulator object:

```
set ns [new Simulator]
```

Then, it is necessary to open a file which will be used for the *nam* trace data, firstly opening the file “out.nam” for writing and then, creating the trace file where all the relevant data in the simulation will be written.

```
Set nf [open out.nam w]
$ns namtrace-all $nf
```

After that, the “finish” procedure is in charge of closing the trace file and starting *nam* as follows:

```
proc finish {} {
global ns nsf
$ns flush-trace
close $nf
nam out.nam &
exit 0
}
```

The next two lines tell the simulator object to execute the “finish” procedure after 5.0 seconds of simulation time and then, to start the simulation.

```
$ns at 5.0 “finish”
$ns run
```

The written code can be useful to start any topology and, to start with a simple one, it is going to be created two nodes and one link. This code will be inserted just before the “proc finish” line.

```
set n0 [ns node]
set n1 [ns node]
```

Two nodes have been created, assigning their handles “n0” and “n1”. In order to connect them it is necessary to add:

```
$ns duplex-link $n0 $n1 1Mb 10 ms
DropTail
```

This line tells the simulator object to connect the two nodes with a duplex link with 1 Megabit bandwidth, a delay of 10 ms and a DropTail queue.

Once the script is saved, it is possible to execute it with the command “ns example.tcl” and to see a *nam* output like the next:

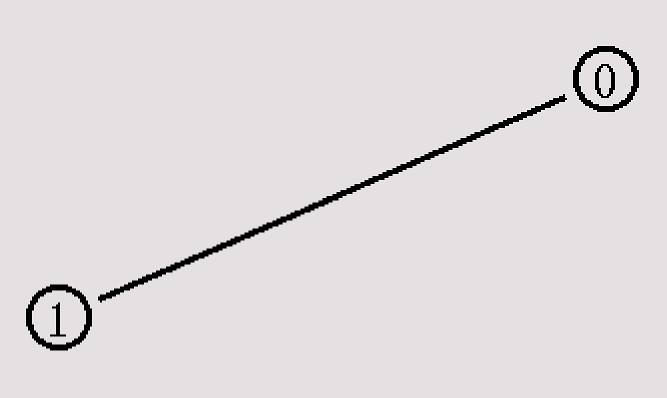

Fig 2.1 Two nodes, one link [6]

But, by now, it is only possible to look at the topology. So, the next step is to send some data from node n0 to node n1, from one “agent” to another one (ns uses “agents” to send and receive data).

```
#Create a UDP agent and attach it to node n0
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0

# Create a CBR traffic source and attach it to udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
```

The lines above create a UDP (User Datagram Protocol) agent and attach it to the node n0, then attach a CBR (constant bit rate) traffic generator to the UDP agent. Packet size is fixed to 500 bytes and a packet will be sent every 0.005 seconds. The lines below are added in order to create a Null agent which acts as traffic sink for node n1:

```
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0
```

Finally, it is necessary to connect the two agents with each other and to tell the CBR agent the moment to send data and the moment to stop sending it.

```
$ns connect $udp0 $null0
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
```

Saving the script and typing again “ns example.tcl”, at this moment is possible to see the simulation in the *nam* window, how node 0 starts sending data packets to node 1 after 0.5 simulation seconds. It is seen like figure 2.2:

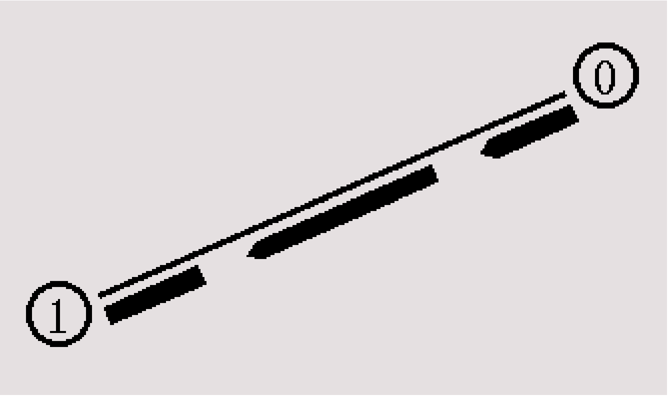

Fig. 2.2 Two nodes, one link and sending data [7]

2.4. Mobility Tools

There are different ways to generate movement for MNs in NS-2. *Setdest* and *Bonnmotion*, used in this project, are exposed in this section.

2.4.1. Setdest

Setdest tool is used to generate the position of nodes, their moving speed and moving directions. It is integrated in the package ns-allinone-2.31 [6] and it follows this syntax:

```
setdest  -v   -n   -s   -m   -M   -t  -P   -p   -x   -y  >
scenario-name.tcl
```

Where the meaning of the parameters are the next:

v → Version: 1 = 1999 CMU, 2 = 2003 UM
n → Number of nodes in the scenario
s → Speed distribution: 1 = uniform, 2 = normal
m → Minimum speed in m/s
M → Maximum speed in m/s
T → Simulation time in seconds
P → Pause type: 1 = constant, 2 = uniform
p → Pause time in seconds
x → Length of the scenario in the x axis
y → Length of the scenario in the y axis

In this project, this tool is used to generate the Random Waypoint mobility model. On the other hand, *Bonnmotion* tool is used to generate the Gauss-Markov mobility model because *setdest* does not support it.

2.4.2. Bonnmotion

Bonnmotion [8] is a Java software developed in the Communication Systems group at the Institute of Computer Science IV of the University of Bonn [8]. It creates mobility scenarios and analyzes them. The scenarios can be exported for NS-2 and for GlomoSim/QualNet, too. It supports the Random Waypoint model, the Gauss-Markov model, the Manhattan Grid model and the Reference Point Group model.

Once the software is installed, to start generating mobility the two command lines to apply are the next:

```
bm -f scenario1 RandomWaypoint -n  -d -x -y
bm NSFile -f scenario1
```

The defining parameters are:

n → Number of nodes
d → Duration, simulation time in seconds
x → Length of the scenario in the x axis
y → Length of the scenario in the y axis
uniformSpeed → Speed distribution: true = uniform, false = normal

2.5. Infrastructure mode support for IEEE 802.11 implementation in NS-2

Infrastructure mode simulations are not supported in the existing 802.11 implementation in NS-2.31. The patch applied [9] [10] supports Passive and

Active Scanning, Authentication, Association, Inter-AP communication and mobility support (Handoff). This section outlines the changes which have to be made to the existing NS-2.31 802.11 implementation, to enable infrastructure mode support.

2.5.1. Changes in NS-2

- **Beacon Frames and Passive Scanning**

A node configured as an AP will start transmitting beacon frames every Beacon Interval using **BeaconTimer**.

sendBEACON() function builds the beacon packet and **chk_pktBEACON()** transmits it. The main fields are Address and BSSI information, Timestamp and Beacon Interval. All the relevant information is stored by **recvBEACON()**, the beacon reception function.

If Scan Type is configured as PASSIVE, the RSS (Received Signal Strength) of all beacons received during **ChannelTime** are stores and the best AP is determined using **passive_scan()**. It exists a linked list containing the list of APs and beacon powers, too. Once passive scanning is over, STAs can move on to authentication and association.

- **Probe frames and Active Scanning**

When the ScanType of a STA is ACTIVE, it sends out a broadcast Probe Request, which will be received by all APs in range. This probe request transmissions are made through the functions **sendPROBEREQ()** and **check_pktPROBEREQ()**; the APs use **recvPROBEREQ** to receive them and evoke unicast Probe Responses, containing the same information as a Beacon frame, using **sendPROBEREP()** and **check_pktPROBEREP()**.

At this moment **recvPROBEREP** is called to store the information in the probe frame.

To handle active scanning was created **Probe timer**, which can be run for the time intervals **MinChannelTime** and **MaxChannelTime**. MinChannelTime is the time to declare a channel empty and channel will be declarer devoid of APs if CCA (Clear Channel Assessment) busy is not indicated for this time. On the other hand, MaxChannelTime will be handled if CCA busy was not indicated within MinChannelTime, collecting all Probe Responses

After expiring Probe Timer, using **active_scan()**, the STA processes all the probe responses and selects the strongest AP based on RSS measurements.

- **Authentication**

The function in charge of starting the authentication process of each STA is **checkAssocAuthStatus()**. This process takes place when the best AP has been determined and the scanning is over.

Each STA sends out an Authentication frame to its strongest AP containing sequence number 1 and the specific BSSID. This frame is built by **sendAUTHENTICATE()** function and the transmission function is **check_pktAUTHENTICATE()**.

Once the AP receives the authentication frame using **recvAUTHENTICATE()**, it replies with an authentication frame of sequence number 2. Each STA, on reception of the second authentication frame, sets its own authenticated flag and sends an ACK (acknowledgement) to the AP.

If the authentication process is successful, the client sets the AP's MAC address as its **BSSID** and is ready for association. On the other hand, when the AP receives the ACK from the authenticating node, it sets the authentication state of the STA in a linked list.

- **Association**

On this stage, once authentication process is finished, each STA sends the Association Request frame by using **sendASSOCREQ()** and **check_pktASSOCREQ()**.

The AP, on receiving the Association request frame using **recvASSOCREQ()**, sends out the Association response, which will contain the Association ID, too. The function in charge of this process are **sendASSOCREP()** and **check_pktASSOCREP()**.

Each STA receives Association response frames using **recvASSOCREP()** and sends an acknowledgement to the AP. From this moment it will be able to transfer data packets in the BSS.

The AP updates its client list when receives the ACK and STA will be able to transfer data in the BSS through the AP. Packets are filtered using the client state table, discarding packets proceeds from a client which doesn't belong to the BSS.

2.5.2. Inter-AP communication and mobility support

Thanks to the infrastructure mode support, clients of the same or different BSS can exchange data. If there is a connection between two clients of different BSSs, the AP will forward the packets from one BSS to another one. However, if the two clients belong to the same BSS, data will be exchange through one AP. Although it has been simulated inter-AP communication on a wired distribution system, it is possible to achieve communication on a Wireless distribution medium.

Moreover, using a multicast method, it is possible to transfer data across multiple BSSs by the **ToDs** and **FromDs** bits.

The handover process begins when a client moves from one BSS to another one. It is detected by a decrease in RSS (Receive Signal Strength). After three consecutive retransmissions of the same data, it is supposed that the client is out of range of its AP. In Section 3.1, different network configurations are analyzed in order to check the infrastructure mode operation.

Figure 2.3 shows all the functions used for the infrastructure mode support. They are divided in two columns, depending on wether they belong to the AP or to the STA.

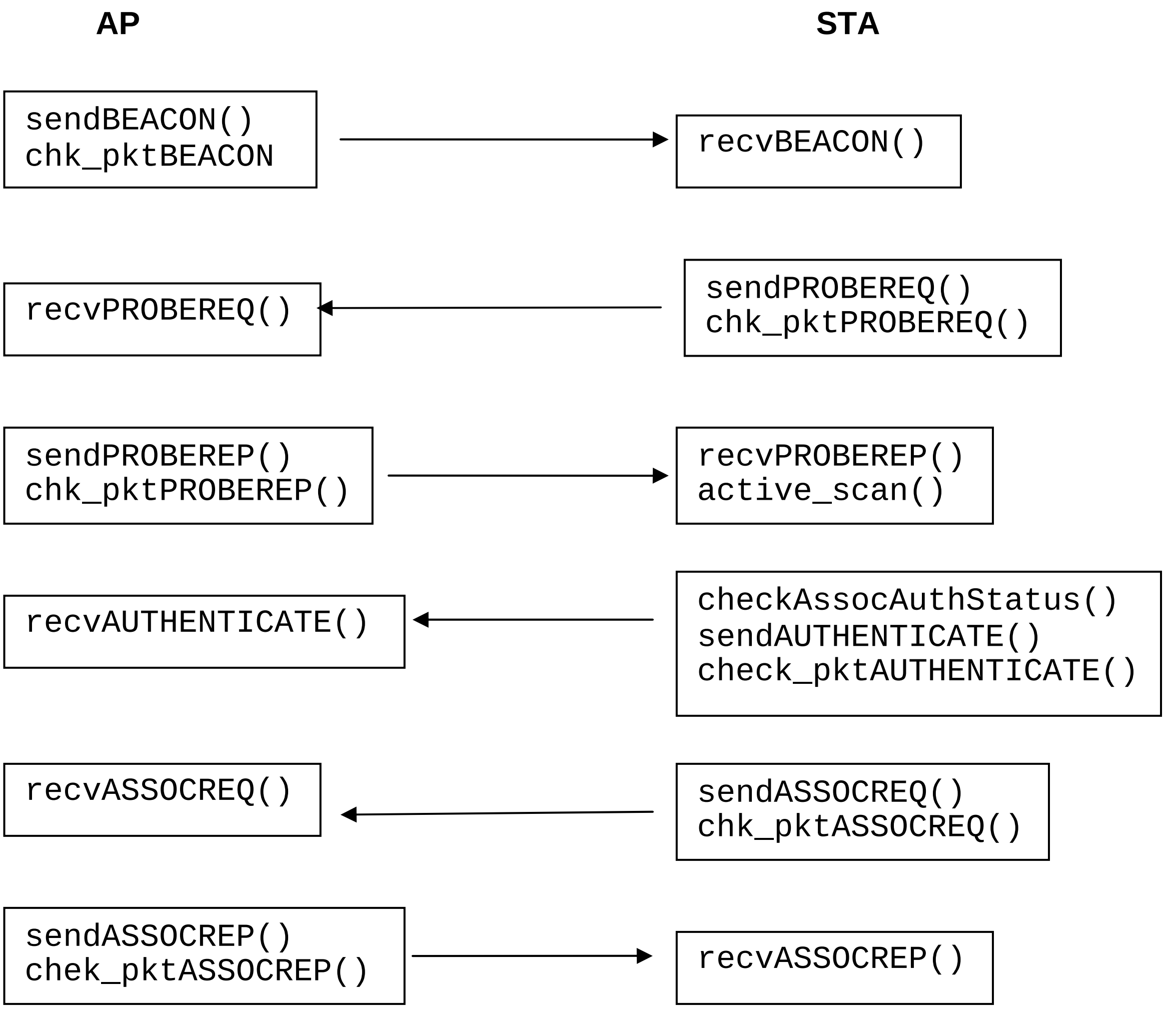

Fig. 2.3 Infrastructure mode support functions

CHAPTER 3. SIMULATION: STUDY OF HANDOVERS

3.1. Scenarios tested for infrastructure mode

Having applied the patch described on chapter 2 and in order to check the correct operation of the infrastructure mode, four different scenarios have been tested. Mobility has been applied to force handovers, too.

3.1.1. Scenario 1: 2 static nodes, 1 AP

The first scenario checks the infrastructure mode operation in a simple topology composed by one AP and 2 mobile nodes without movement. While node 1 is the AP, nodes 0 and 2 represent the two MNs. We expect that data sent from node 2 to 0 go through the AP.

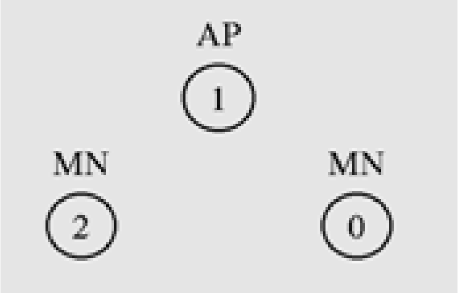

Fig. 3.1 2 MNs and 1 AP

In the following trace, data sent (i.e. *s*) and received (i.e. *r*) at each node is displayed. After the authentication (AUTH) and association request and response (ACRQ and ACRP, respectively) processes, at 5.0 s node 2 starts to send data to node 0 (i.e. cbr). At 5.05 s it is possible to see how packets sent by MN 2 are received by the AP 1 and then redirected to MN 0. Each frame must be acknowledged (i.e. ACK). Beacon frames (i.e. BCN) are sent periodically by the AP in order to set channel parameters among its clients.

```
s 0.001216047 _1_ MAC  --- 0 BCN 72 [0 ffffffff 1 533f]
r 0.001792085 _2_ MAC  --- 0 BCN 72 [0 ffffffff 1 533f]
r 0.001792094 _0_ MAC  --- 0 BCN 72 [0 ffffffff 1 533f]
s 0.016560000 _0_ MAC  --- 0 AUTH 58 [0 1 0 0]
r 0.017024047 _1_ MAC  --- 0 AUTH 58 [0 1 0 0]
s 0.017074047 _1_ MAC  --- 0 AUTH 58 [13a 0 1 0]
r 0.017538094 _0_ MAC  --- 0 AUTH 58 [13a 0 1 0]
s 0.017548094 _0_ MAC  --- 0 ACK 38 [0 1 0 0]
r 0.017852141 _1_ MAC  --- 0 ACK 38 [0 1 0 0]
s 0.029062094 _0_ MAC  --- 0 ACRQ 52 [0 1 0 0]
r 0.029478141 _1_ MAC  --- 0 ACRQ 52 [0 1 0 0]
s 0.029728141 _1_ MAC  --- 0 ACRP 54 [0 0 1 0]
```

```
r 0.030160189 _0_ MAC  --- 0 ACRP 54 [0 0 1 0]                    → MN 0 is associated with AP1
s 0.030170189 _0_ MAC  --- 0 ACK 38 [0 1 0 0]
r 0.030474236 _1_ MAC  --- 0 ACK 38 [0 1 0 0]
s 1.016680000 _2_ MAC  --- 0 AUTH 58 [0 1 2 0]
r 1.017144038 _1_ MAC  --- 0 AUTH 58 [0 1 2 0]
s 1.017394038 _1_ MAC  --- 0 AUTH 58 [13a 2 1 0]
r 1.017858075 _2_ MAC  --- 0 AUTH 58 [13a 2 1 0]
s 1.029442075 _2_ MAC  --- 0 ACRQ 52 [0 1 2 0]
r 1.029858113 _1_ MAC  --- 0 ACRQ 52 [0 1 2 0]
s 1.029928113 _1_ MAC  --- 0 ACRP 54 [0 2 1 0]
r 1.030360151 _2_ MAC  --- 0 ACRP 54 [0 2 1 0]                    → MN 2 is associated with AP1
```

→ Packets 1 and 2 are sent from MN 2 to AP 1 and then redirected to MN 0

```
s 5.050415000 _2_ MAC  --- 1 cbr 570 [13a 1 2 800] ------ [2:0 0:0 32 0] [1] 0 0
r 5.051004129 _1_ MAC  --- 1 cbr 512 [13a 1 2 800] ------ [2:0 0:0 32 0] [1] 1 0
s 5.051014129 _1_ MAC  --- 0 ACK 38 [0 2 0 0]
r 5.051318166 _2_ MAC  --- 0 ACK 38 [0 2 0 0]
s 5.051968129 _1_ MAC  --- 1 cbr 570 [13a 0 1 800] ------ [2:0 0:0 32 0] [1] 1 0
r 5.052557267 _0_ MAC  --- 1 cbr 512 [13a 0 1 800] ------ [2:0 0:0 32 0] [1] 2 0
s 5.052567267 _0_ MAC  --- 0 ACK 38 [0 1 0 0]
r 5.052871314 _1_ MAC  --- 0 ACK 38 [0 1 0 0]
s 5.100630000 _1_ MAC  --- 0 BCN 72 [0 ffffffff 1 1440]
r 5.101206038 _2_ MAC  --- 0 BCN 72 [0 ffffffff 1 1440]
r 5.101206047 _0_ MAC  --- 0 BCN 72 [0 ffffffff 1 1440]
s 5.101296038 _2_ MAC  --- 2 cbr 570 [13a 1 2 800] ------ [2:0 0:0 32 0] [2] 0 0
r 5.101885166 _1_ MAC  --- 2 cbr 512 [13a 1 2 800] ------ [2:0 0:0 32 0] [2] 1 0
s 5.101895166 _1_ MAC  --- 0 ACK 38 [0 2 0 0]
r 5.102199204 _2_ MAC  --- 0 ACK 38 [0 2 0 0]
s 5.102849166 _1_ MAC  --- 2 cbr 570 [13a 0 1 800] ------ [2:0 0:0 32 0] [2] 1 0
r 5.103438304 _0_ MAC  --- 2 cbr 512 [13a 0 1 800] ------ [2:0 0:0 32 0] [2] 2 0
s 5.103448304 _0_ MAC  --- 0 ACK 38 [0 1 0 0]
r 5.103752352 _1_ MAC  --- 0 ACK 38 [0 1 0 0]
```

3.1.2. Scenario 2: 2 MNs, 4 APs and 1 switch

The second scenario is composed by 2 MNs, 4 APs and a wired node (switch). The infrastructure mode operation is checked when MNs associate to different APs. Three different subscenarios are analyzed: statics nodes in the first one; MN moving among APs in the other two subscenarios. We want to test if roaming is supported.

For all subscenarios, Node 0 is the switch. APs are represented by nodes 1, 2, 3 and 4 and mobile nodes are 5 and 6.

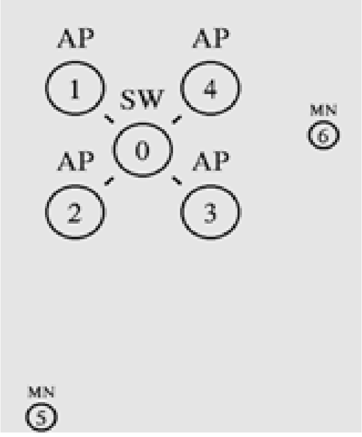

Fig. 3.2 2 MNs, 4 APs and 1 switch

3.1.2.1. Static nodes connected to different AP

At 2.0 s CBR traffic starts from node 5 to node 6. Firstly, the MN 5 is associated to the AP 2 and the MN 6 is associated to the AP 4. Packets sent by the MN 5, the transmitter node, to the MN 6 follow the sequence 5-2-0-4-6, providing infrastructure mode operation.

```
s 0.017166137 _5_ MAC  --- 0 AUTH 58 [0 1 4 0]
r 0.017630275 _2_ MAC  --- 0 AUTH 58 [0 1 4 0]
s 0.018260275 _2_ MAC  --- 0 AUTH 58 [13a 4 1 0]
r 0.018724412 _5_ MAC  --- 0 AUTH 58 [13a 4 1 0]
s 0.030448412 _5_ MAC  --- 0 ACRQ 52 [0 1 4 0]
r 0.030864550 _2_ MAC  --- 0 ACRQ 52 [0 1 4 0]
s 0.030954550 _2_ MAC  --- 0 ACRP 54 [0 4 1 0]
r 0.031386687 _5_ MAC  --- 0 ACRP 54 [0 4 1 0]                    → MN 5 is associated with AP 2
s 0.018734412 _5_ MAC  --- 0 ACK 38 [0 1 0 0]
r 0.019038550 _2_ MAC  --- 0 ACK 38 [0 1 0 0]
s 1.120490000 _6_ MAC  --- 0 AUTH 58 [0 3 5 0]
r 1.120954137 _4_ MAC  --- 0 AUTH 58 [0 3 5 0]
s 1.121384137 _4_ MAC  --- 0 AUTH 58 [13a 5 3 0]
r 1.121848275 _6_ MAC  --- 0 AUTH 58 [13a 5 3 0]
s 1.121858275 _6_ MAC  --- 0 ACK 38 [0 3 0 0]
r 1.122162412 _4_ MAC  --- 0 ACK 38 [0 3 0 0]
s 1.133692275 _6_ MAC  --- 0 ACRQ 52 [0 3 5 0]
r 1.134108412 _4_ MAC  --- 0 ACRQ 52 [0 3 5 0]
s 1.134478412 _4_ MAC  --- 0 ACRP 54 [0 5 3 0]
r 1.134910550 _6_ MAC  --- 0 ACRP 54 [0 5 3 0]                    → MN 6 is associated with AP4
s 1.134920550 _6_ MAC  --- 0 ACK 38 [0 3 0 0]
r 1.135224687 _4_ MAC  --- 0 ACK 38 [0 3 0 0]
```

→ *Packet 25 follows the sequence 5-2-0-4-6*

```
s 2.004681781 _5_ MAC  --- 25 cbr 288 [13a 1 4 800] ------- [4194305:2 16777217:3 32 8388608] [1] 0 0
r 2.005065919 _2_ MAC  --- 25 cbr 230 [13a 1 4 800] ------- [4194305:2 16777217:3 32 8388608] [1] 1 0
s 2.005075919 _2_ MAC  --- 0 ACK 38 [0 4 0 0]
+ 2.005091 2 0 cbr 230 ------- 0 1.0.1.2 4.0.1.3 1 25
- 2.005091 2 0 cbr 230 ------- 0 1.0.1.2 4.0.1.3 1 25
r 2.005380056 _5_ MAC  --- 0 ACK 38 [0 4 0 0]
r 2.007459 2 0 cbr 230 ------- 0 1.0.1.2 4.0.1.3 1 25
+ 2.007459 0 4 cbr 230 ------- 0 1.0.1.2 4.0.1.3 1 25
- 2.007459 0 4 cbr 230 ------- 0 1.0.1.2 4.0.1.3 1 25
r 2.009827 0 4 cbr 230 ------- 0 1.0.1.2 4.0.1.3 1 25
```

```
s 2.012188310 _4_ MAC  --- 25 cbr 288 [13a 5 3 800] ------- [4194305:2 16777217:3 29 16777217] [1] 1
0
r 2.012572447 _6_ MAC  --- 25 cbr 230 [13a 5 3 800] ------- [4194305:2 16777217:3 29 16777217] [1] 2 0
```

3.1.2.2. Mobile nodes connect to the same AP

At 5.0 s the MN 5 starts to move (i.e. *M*) from its initial position to another one near AP 4, which causes the communication between the two wireless nodes go through this AP (i.e. MN5 performs a handover from AP 2 to AP 4). Since the mobile nodes are connected to the same AP, it is not necessary to redirect the information to the switch and the new sequence is 5-4-6.

```
M 5.00000 4194305 (140.00, 60.00, 0.00), (470.00, 90.00), 30.00        → Movement of MN 5
s 13.721944669 _5_ MAC  --- 0 AUTH 58 [0 3 4 0]
r 13.722408842 _4_ MAC  --- 0 AUTH 58 [0 3 4 0]
s 13.722818842 _4_ MAC  --- 0 AUTH 58 [13a 4 3 0]
r 13.723283015 _5_ MAC  --- 0 AUTH 58 [13a 4 3 0]
s 13.736722435 _5_ MAC  --- 0 ACRQ 52 [0 3 4 0]
r 13.737138607 _4_ MAC  --- 0 ACRQ 52 [0 3 4 0]
s 13.737308607 _4_ MAC  --- 0 ACRP 54 [0 4 3 0]
r 13.737740779 _5_ MAC  --- 0 ACRP 54 [0 4 3 0]        → MN 5 is associated with AP 4
s 13.737750779 _5_ MAC  --- 0 ACK 38 [0 3 0 0]
r 13.738054951 _4_ MAC  --- 0 ACK 38 [0 3 0 0]
```

→ Packet 3243 sent by MN5 is received by AP 4 and the redirected to MN 6

```
s 13.741265648 _5_ MAC  --- 3243 cbr 288 [13a 3 4 800] ------- [4194305:2 ... 16777217:3 32 16777216]
[3130] 0 0
r 13.741649820 _4_ MAC  --- 3243 cbr 230 [13a 3 4 800] ------- [4194305:2 16777217:3 32 16777216]
[3130] 1 0
s 13.742093820 _4_ MAC  --- 3243 cbr 288 [13a 5 3 800] ------- [4194305:2 16777217:3 31 16777217]
[3130] 1 0
r 13.742477957 _6_ MAC  --- 3243 cbr 230 [13a 5 3 800] ------- [4194305:2 16777217:3 31 16777217]
[3130] 2 0
```

3.1.2.3. Mobile nodes connect to different APs

MN 5 simulates the movement towards a new position near AP 3 and the handover is realized from AP 2 to AP 3. At this moment, packets follow the sequence 5-3-0-4-6 and the infrastructure mode is working, too.

```
M 5.00000 4194305 (160.00, 90.00, 0.00), (290.00, 90.00), 30.00
s 13.695579411 _5_ MAC  --- 0 ACK 38 [0 2 0 0]
r 13.695883458 _3_ MAC  --- 0 ACK 38 [0 2 0 0]
s 13.700270000 _3_ MAC  --- 0 BCN 72 [0 ffffffff 2 2b40]
r 13.700846047 _5_ MAC  --- 0 BCN 72 [0 ffffffff 2 2b40]
s 13.701340500 _2_ MAC  --- 0 BCN 72 [0 ffffffff 1 2b40]
s 13.702591500 _4_ MAC  --- 0 BCN 72 [0 ffffffff 3 2b40]
r 13.703167637 _6_ MAC  --- 0 BCN 72 [0 ffffffff 3 2b40]
s 13.710649961 _5_ MAC  --- 0 AUTH 58 [0 2 4 0]
r 13.711114008 _3_ MAC  --- 0 AUTH 58 [0 2 4 0]
s 13.711684008 _3_ MAC  --- 0 AUTH 58 [13a 4 2 0]
r 13.712148055 _5_ MAC  --- 0 AUTH 58 [13a 4 2 0]
s 13.730021849 _5_ MAC  --- 0 ACRQ 52 [0 2 4 0]
r 13.730437897 _3_ MAC  --- 0 ACRQ 52 [0 2 4 0]
s 13.730907897 _3_ MAC  --- 0 ACRP 54 [0 4 2 0]
r 13.731339944 _5_ MAC  --- 0 ACRP 54 [0 4 2 0]        → MN 5 is associated with AP 3
s 13.731349944 _5_ MAC  --- 0 ACK 38 [0 2 0 0]
```

```
r 13.731653991 _3_ MAC  --- 0 ACK 38 [0 2 0 0]
```

→ *Packet 3662 follows the sequence 5-3-0-4-6*

```
s 15.250967276 _5_ MAC  --- 3662 cbr 288 [13a 2 4 800] - [419305:2 16777217:3 32 1258212] [3533] 0
0
r 15.251351323 _3_ MAC  --- 3662 cbr 230 [13a 2 4 800] - [419305:2 16777217:3 32 12582912] [3533] 1
0
s 15.251361323 _3_ MAC  --- 0 ACK 38 [0 4 0 0]
+ 15.251376 3 0 cbr 230 ------- 0 1.0.1.2 4.0.1.3 3533 3662
- 15.251376 3 0 cbr 230 ------- 0 1.0.1.2 4.0.1.3 3533 3662
r 15.253744 3 0 cbr 230 ------- 0 1.0.1.2 4.0.1.3 3533 3662
+ 15.253744 0 4 cbr 230 ------- 0 1.0.1.2 4.0.1.3 3533 3662
- 15.253744 0 4 cbr 230 ------- 0 1.0.1.2 4.0.1.3 3533 3662
r 15.256112 0 4 cbr 230 ------- 0 1.0.1.2 4.0.1.3 3533 3662
s 15.256667323 _4_ MAC  --- 3662 cbr 288 [13a 5 3 800] ------- [4194305:2 16777217:3 29 16777217]
[3533] 1 0
r 15.257051461 _6_ MAC  --- 3662 cbr 230 [13a 5 3 800] ------- [4194305:2 16777217:3 29 16777217]
[3533] 2 0
```

3.1.3. Scenario 3: 1 static node, 1 AP, 1 switch and 1 static host

The third scenario tests the infrastructure mode between a MN and a static host.
Node 0 represents the switch, which is connected to the AP and to a static host. There is traffic between the mobile node and the static host connected to the switch.

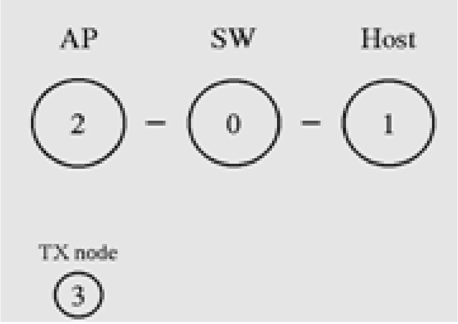

Fig. 3.3 1 MN, 1 AP and a static host connected to the switch

The trace file shows the association between the node 3 and the only AP (2), and how packets are transmitted from node 3 to host 1 via the AP 2 and the switch 0 (3-2-0-1).

```
s 0.000090000 _2_ MAC  --- 0 BCN 72 [0 ffffffff 0 173f]
r 0.000666067 _3_ MAC  --- 0 BCN 72 [0 ffffffff 0 173f]
s 0.017166067 _3_ MAC  --- 0 AUTH 58 [0 0 1 0]
r 0.017630133 _2_ MAC  --- 0 AUTH 58 [0 0 1 0]
s 0.017720133 _2_ MAC  --- 0 AUTH 58 [13a 1 0 0]
r 0.018184200 _3_ MAC  --- 0 AUTH 58 [13a 1 0 0]
s 0.030088200 _3_ MAC  --- 0 ACRQ 52 [0 0 1 0]
r 0.030504267 _2_ MAC  --- 0 ACRQ 52 [0 0 1 0]
```

```
s 0.030714267 _2_ MAC  --- 0 ACRP 54 [0 1 0 0]
r 0.031146333 _3_ MAC  --- 0 ACRP 54 [0 1 0 0]                    → MN 3 is associated to AP 2
s 0.031156333 _3_ MAC  --- 0 ACK 38 [0 0 0 0]
r 0.031460400 _2_ MAC  --- 0 ACK 38 [0 0 0 0]
```

→ Packet 7 follows the sequence 3-2-0-1

```
s 2.000095000 _3_ MAC  --- 7 cbr 288 [13a 0 1 800] ------- [4194305:2 2048:1 32 4194304] [0] 0 0
r 2.000479067 _2_ MAC  --- 7 cbr 230 [13a 0 1 800] ------- [4194305:2 2048:1 32 4194304] [0] 1 0
s 2.000489067 _2_ MAC  --- 0 ACK 38 [0 1 0 0]
+ 2.000504 2 0 cbr 230 ------- 0 1.0.1.2 0.1.0.1 0 7
- 2.000504 2 0 cbr 230 ------- 0 1.0.1.2 0.1.0.1 0 7
s 2.001323067 _2_ MAC  --- 0 BCN 72 [0 ffffffff 0 40]
r 2.001899133 _3_ MAC  --- 0 BCN 72 [0 ffffffff 0 40]
r 2.002872 2 0 cbr 230 ------- 0 1.0.1.2 0.1.0.1 0 7
+ 2.002872 0 1 cbr 230 ------- 0 1.0.1.2 0.1.0.1 0 7
- 2.002872 0 1 cbr 230 ------- 0 1.0.1.2 0.1.0.1 0 7
r 2.00524 0 1 cbr 230 ------- 0 1.0.1.2 0.1.0.1 0 7
```

3.1.4. Scenario 4: 1 MN, 4 APs, 1 switch and 1 static host

As for this last simulation example, the scenario 4 is similar to the scenario 3 but it is composed by 4 APs instead of one AP. Once again, the infrastructure mode is tested while the MN is moving from one AP to another one.

A static host is connected to the switch and the mobile node moves along the simulation area. Nodes 0 and 1 represent the switch and the fix host respectively. APs are defined by nodes 2, 3, 4 and 5 and the transmitter mobile node is node 6.

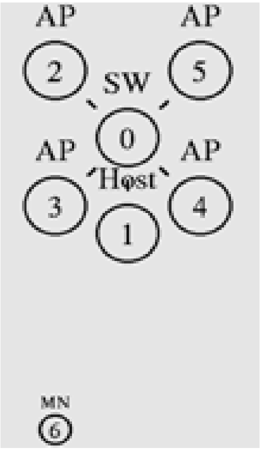

Fig. 3.4 1 MN, 4 APs and a static host connected to the switch

Firstly associated with AP 3, MN 6 moves along the simulation area and communication between it and the static host (1) remains working. Handover is performed through all the APs depending on MN 6 position.

```
s 0.020409035 _6_ MAC  --- 0 AUTH 58 [0 1 4 0]
r 0.020873168 _3_ MAC  --- 0 AUTH 58 [0 1 4 0]
```

```
s 0.021503168 _3_ MAC  --- 0 AUTH 58 [13a 4 1 0]
r 0.021967302 _6_ MAC  --- 0 AUTH 58 [13a 4 1 0]
s 0.021977302 _6_ MAC  --- 0 ACK 38 [0 1 0 0]
r 0.022281435 _3_ MAC  --- 0 ACK 38 [0 1 0 0]
s 0.033731302 _6_ MAC  --- 0 ACRQ 52 [0 1 4 0]
r 0.034147435 _3_ MAC  --- 0 ACRQ 52 [0 1 4 0]
s 0.034737435 _3_ MAC  --- 0 ACRP 54 [0 4 1 0]
r 0.035169568 _6_ MAC  --- 0 ACRP 54 [0 4 1 0]              → MN 6 is associated to AP 3
s 0.035179568 _6_ MAC  --- 0 ACK 38 [0 1 0 0]
r 0.035483702 _3_ MAC  --- 0 ACK 38 [0 1 0 0]
```

→ Packet 19 follows the sequence 6-3-0-1

```
s 2.000155000 _6_ MAC  --- 19 cbr 288 [13a 1 4 800] ------- [4194305:2 2048:1 32 8388608] [0] 0 0
r 2.000539133 _3_ MAC  --- 19 cbr 230 [13a 1 4 800] ------- [4194305:2 2048:1 32 8388608] [0] 1 0
s 2.000549133 _3_ MAC  --- 0 ACK 38 [0 4 0 0]
+ 2.000564 3 0 cbr 230 ------- 0 1.0.1.2 0.1.0.1 0 19
- 2.000564 3 0 cbr 230 ------- 0 1.0.1.2 0.1.0.1 0 19
r 2.002932 3 0 cbr 230 ------- 0 1.0.1.2 0.1.0.1 0 19
+ 2.002932 0 1 cbr 230 ------- 0 1.0.1.2 0.1.0.1 0 19
- 2.002932 0 1 cbr 230 ------- 0 1.0.1.2 0.1.0.1 0 19
r 2.0053 0 1 cbr 230 ------- 0 1.0.1.2 0.1.0.1 0 19
```

```
M 5.00000 4194305 (150.00, 60.00, 0.00), (350.00, 100.00), 30.00      → Movement of MN 6
s 12.660352032 _6_ MAC  --- 0 AUTH 58 [0 2 4 0]
r 12.660816199 _4_ MAC  --- 0 AUTH 58 [0 2 4 0]
s 12.661246199 _4_ MAC  --- 0 AUTH 58 [13a 4 2 0]
r 12.661710365 _6_ MAC  --- 0 AUTH 58 [13a 4 2 0]
s 12.661720365 _6_ MAC  --- 0 ACK 38 [0 2 0 0]
r 12.662024532 _4_ MAC  --- 0 ACK 38 [0 2 0 0]
s 12.673474365 _6_ MAC  --- 0 ACRQ 52 [0 2 4 0]
r 12.673890532 _4_ MAC  --- 0 ACRQ 52 [0 2 4 0]
s 12.674160532 _4_ MAC  --- 0 ACRP 54 [0 4 2 0]
r 12.674592699 _6_ MAC  --- 0 ACRP 54 [0 4 2 0]         → MN 6 is associated with AP 4
s 12.674602699 _6_ MAC  --- 0 ACK 38 [0 2 0 0]
r 12.674906865 _4_ MAC  --- 0 ACK 38 [0 2 0 0]
```

→ Packet 3909 follows the sequence 6-4-0-1

```
s 16.236757515 _6_ MAC  --- 3909 cbr 288 [13a 2 4 800] ----- [4194305:2 2048:1 32 12582912] [3796] 0 0
r 16.237141682 _4_ MAC  --- 3909 cbr 230 [13a 2 4 800] ----- [4194305:2 2048:1 32 12582912] [3796] 1 0
+ 16.237167 4 0 cbr 230 ------- 0 1.0.1.2 0.1.0.1 3796 3909
- 16.237167 4 0 cbr 230 ------- 0 1.0.1.2 0.1.0.1 3796 3909
r 16.239535 4 0 cbr 230 ------- 0 1.0.1.2 0.1.0.1 3796 3909
+ 16.239535 0 1 cbr 230 ------- 0 1.0.1.2 0.1.0.1 3796 3909
- 16.239535 0 1 cbr 230 ------- 0 1.0.1.2 0.1.0.1 3796 3909
r 16.241903 0 1 cbr 230 ------- 0 1.0.1.2 0.1.0.1 3796 3909
```

```
M 17.00000 4194305 (350.00, 100.00, 0.00), (450.00, 100.00), 50.00    → Movement of MN 6
s 24.458081545 _6_ MAC  --- 0 AUTH 58 [0 3 4 0]
r 24.458545545 _5_ MAC  --- 0 AUTH 58 [0 3 4 0]
s 24.459175545 _5_ MAC  --- 0 AUTH 58 [13a 4 3 0]
r 24.459639545 _6_ MAC  --- 0 AUTH 58 [13a 4 3 0]
s 24.459649545 _6_ MAC  --- 0 ACK 38 [0 3 0 0]
r 24.459953545 _5_ MAC  --- 0 ACK 38 [0 3 0 0]
s 24.480323545 _6_ MAC  --- 0 ACRQ 52 [0 3 4 0]
r 24.480739545 _5_ MAC  --- 0 ACRQ 52 [0 3 4 0]
s 24.480889545 _5_ MAC  --- 0 ACRP 54 [0 4 3 0]
r 24.481321545 _6_ MAC  --- 0 ACRP 54 [0 4 3 0]            → MN 6 is associated with AP 5
s 24.481331545 _6_ MAC  --- 0 ACK 38 [0 3 0 0]
r 24.481635545 _5_ MAC  --- 0 ACK 38 [0 3 0 0]
```

→ Packet 6163 follows the sequence 6-5-0-1

```
s 24.515905727 _6_ MAC  --- 6163 cbr 288 [13a 3 4 800] ----- [4194305:2 2048:1 32 16777216] [5997] 0
0
r 24.516289727 _5_ MAC  --- 6163 cbr 230 [13a 3 4 800] ----- [4194305:2 2048:1 32 16777216] [5997] 1
0
+ 24.516315 5 0 cbr 230 ------- 0 1.0.1.2 0.1.0.1 5997 6163
- 24.516315 5 0 cbr 230 ------- 0 1.0.1.2 0.1.0.1 5997 6163
32 16777216] [5998] 1 0
s 24.518245727 _5_ MAC  --- 0 ACK 38 [0 4 0 0]
r 24.518683 5 0 cbr 230 ------- 0 1.0.1.2 0.1.0.1 5997 6163
+ 24.518683 0 1 cbr 230 ------- 0 1.0.1.2 0.1.0.1 5997 6163
- 24.518683 0 1 cbr 230 ------- 0 1.0.1.2 0.1.0.1 5997 6163
r 24.521051 0 1 cbr 230 ------- 0 1.0.1.2 0.1.0.1 5997 6163

M 30.00000 4194305 (450.00, 100.00, 0.00), (2.00, 100.00), 90.00          → Movement of MN 6
s 31.156560630 _6_ MAC  --- 0 ACRQ 52 [0 2 4 0]
r 31.156976783 _4_ MAC  --- 0 ACRQ 52 [0 2 4 0]
s 31.157346783 _4_ MAC  --- 0 ACRP 54 [0 4 2 0]
r 31.157778936 _6_ MAC  --- 0 ACRP 54 [0 4 2 0]                   → MN 6 is associated with AP4
s 32.830189960 _6_ MAC  --- 0 ACRQ 52 [0 1 4 0]
r 32.830606111 _3_ MAC  --- 0 ACRQ 52 [0 1 4 0]
s 32.830776111 _3_ MAC  --- 0 ACRP 54 [0 4 1 0]
r 32.831208261 _6_ MAC  --- 0 ACRP 54 [0 4 1 0]                   → MN 6 is associated with AP 3
s 36.453443863 _6_ MAC  --- 0 AUTH 58 [0 0 4 0]
r 36.453907863 _2_ MAC  --- 0 AUTH 58 [0 0 4 0]
s 36.453997863 _2_ MAC  --- 0 AUTH 58 [13a 4 0 0]
r 36.454461863 _6_ MAC  --- 0 AUTH 58 [13a 4 0 0]
s 36.454471863 _6_ MAC  --- 0 ACK 38 [0 0 0 0]
r 36.454775863 _2_ MAC  --- 0 ACK 38 [0 0 0 0]
s 36.468265863 _6_ MAC  --- 0 ACRQ 52 [0 0 4 0]
r 36.468681863 _2_ MAC  --- 0 ACRQ 52 [0 0 4 0]
s 36.469231863 _2_ MAC  --- 0 ACRP 54 [0 4 0 0]
r 36.469663863 _6_ MAC  --- 0 ACRP 54 [0 4 0 0]                   → MN 6 is associated with AP 2
s 36.469673863 _6_ MAC  --- 0 ACK 38 [0 0 0 0]
r 36.469977863 _2_ MAC  --- 0 ACK 38 [0 0 0 0]

→ Packet 10332 follows the sequence 6-2-0-1

s 39.733928187 _6_ MAC  --- 10332 cbr 288 [13a 0 4 800] --- [4194305:2 2048:1 32 4194304] [10061] 0
r 39.734312187 _2_ MAC  --- 10332 cbr 230 [13a 0 4 800] ---- [4194305:2 2048:1 32 4194304] [10061] 1
s 39.734322187 _2_ MAC  --- 0 ACK 38 [0 4 0 0]
+ 39.734337 2 0 cbr 230 ------- 0 1.0.1.2 0.1.0.1 10061 10332
- 39.734337 2 0 cbr 230 ------- 0 1.0.1.2 0.1.0.1 10061 10332
r 39.734626187 _6_ MAC  --- 0 ACK 38 [0 4 0 0]
r 39.736705 2 0 cbr 230 ------- 0 1.0.1.2 0.1.0.1 10061 10332
+ 39.736705 0 1 cbr 230 ------- 0 1.0.1.2 0.1.0.1 10061 10332
- 39.736705 0 1 cbr 230 ------- 0 1.0.1.2 0.1.0.1 10061 10332
r 39.739073 0 1 cbr 230 ------- 0 1.0.1.2 0.1.0.1 10061 10332
```

3.2. Describing scenarios and handover strategies

Simulation of 24 hours will be realized in order to get enough samples. Random Waypoint and Gauss-Markov mobility models will be generate to evaluate the cell residence time (CRT). In the next sections the scenarios applied and the different handover strategies are described.

3.2.1. Scenarios

Simulation area is a square with side of 280 meters. Two different scenarios are applied.

Figure 3.5 illustrates the scenario with 4 APs, where the coverage area of each cell is 100 meters guaranteeing a full coverage of the simulated area. The coordinates of each AP are represented in table 3.1:

Table 3.1 AP coordinates in the scenario with 4 APs

Coordinates	x (m)	y (m)
AP1	75	205
AP2	205	205
AP3	75	75
AP4	205	75

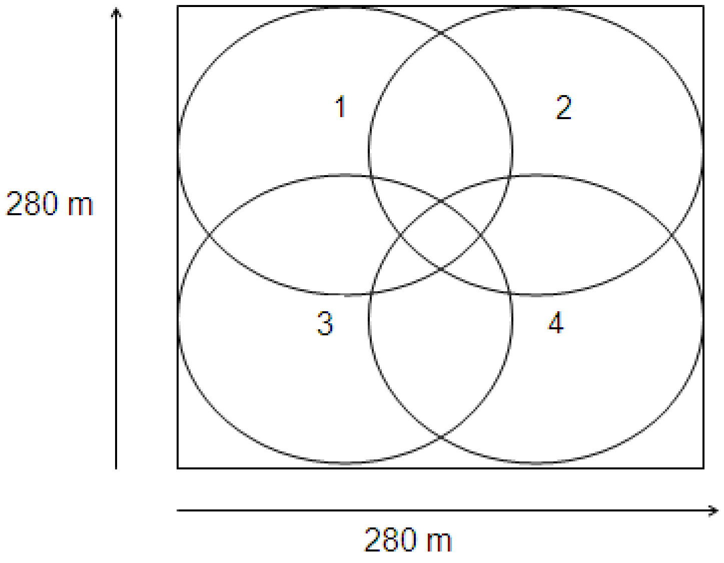

Fig. 3.5 Scenario with 4 APs

On the other hand, figure 3.6 represents a scenario with 8 APs. This scenario simulates a network where the number of users increases and it is important to provide more capacity. Since the area is the same than in the scenario with 4 APs, the coverage range in this case has been reduced to 75 meters to decrease the overlapping effect while still offering full coverage. Table 3.2 represents the AP coordinates for this scenario.

Table 3.2 AP coordinates in the scenario with 8 APs

Coordinates	x (m)	y (m)
AP1	75	205
AP2	140	200
AP3	205	205
AP4	80	140
AP5	200	140
AP6	75	75
AP7	140	80
AP8	205	75

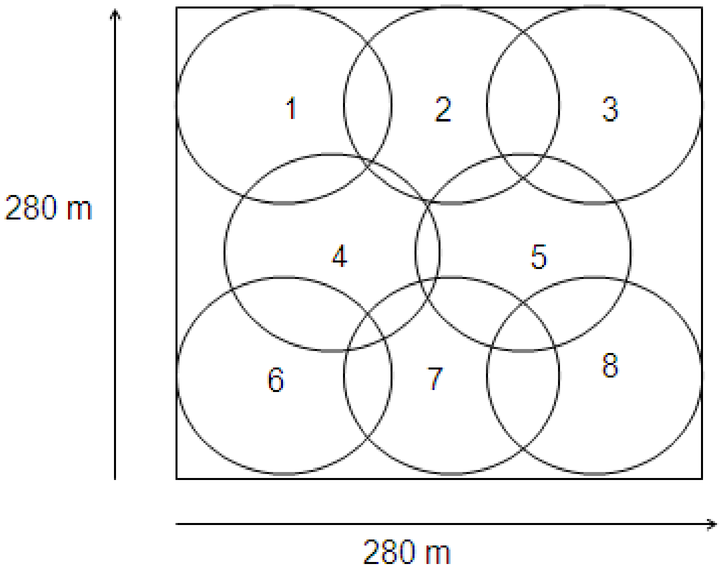

Fig. 3.6 Scenario with 8 APs

Eight different mobility patterns have been created: six from Random Waypoint model and two from Gauss-Markov. Both models allow setting speed distribution (i.e. uniform or normal), while only RWP allows setting pause times. The mobility patterns are represented as follows:

- RWP_u-0: Random Waypoint with uniform speed distribution and pause time set to 0 seconds.
- RWP_u-1: Random Waypoint with uniform speed distribution and pause time set to 1 second.
- RWP_u-10: Random Waypoint with uniform speed distribution and pause time set to 10 seconds.

- RWP_n-0: Random Waypoint with normal speed distribution and pause time set to 0 seconds.

- RWP_n-1: Random Waypoint with normal speed distribution and pause time set to 1 second.

- RWP_n-10: Random Waypoint with normal speed distribution and pause time set to 10 seconds.

- GM_u: Gauss-Markov with uniform speed distribution

- GM_n: Gauss-Markov with normal speed distribution

For each scenario, a MN will be move according to each mobility pattern while sending data to a static host through the switch. In this way, we will record information about handover process.

3.2.2. Handover algorithm

This subsection is divided in two parts: the first one refers to the current implementation of the handover algorithm in NS-2, and the second one is a study of the power received by a MN in different situations in order to design a new algorithm.

3.2.2.1. Current implementation in NS-2

As it has seen in section 2.5.2, the current implementation (algorithm A) of the handover process works as follows: when MN has to resend the same packet more than three times, it starts the scanning process in order to find out AP in range with better signal strength: from the new AP list, it select the best AP (i.e. the one with the highest received signal strength) in order to associate to it. In next section, the signal strength is analyzed in order to find out where the node is located when it starts a handover.

3.2.2.2. Study of RSS

In order to discover how the current handover strategy works and how to improve it, it has been necessary to study the Received Signal Strength (RSS) level of the mobile node while moving. Three situations have been analyzed: the MN located at the border of another cell, located at a middle point between two cells and located at the border of its cell. Finally, the average RSS found for each situation is displayed in table 3.3.

- Situation A: Mobile node at the border of another cell

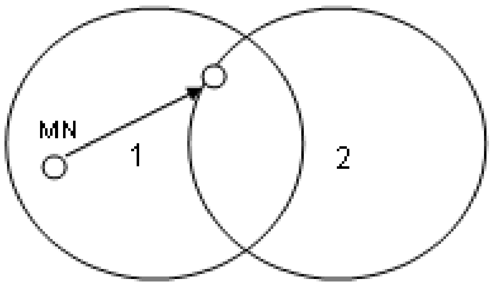

Fig. 3.7 MN at the border of another cell

- Situation B: Mobile node at a middle point between two cells

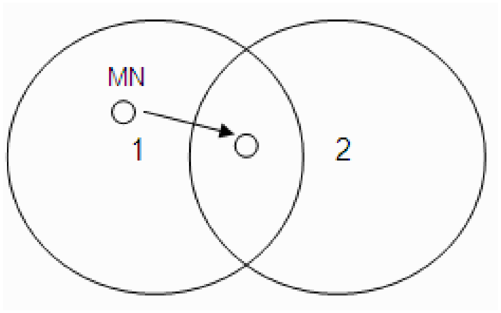

Fig. 3.8 MN at a middle point between two cells

- Situation C: Mobile node at the border of its cell

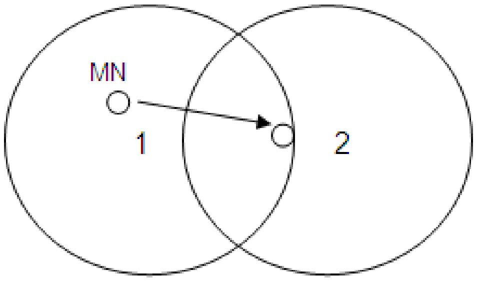

Fig. 3.9 MN at the border of its cell

Table 3.3 shows the average RSS, the variance and the standard deviation for the situations A, B, C and for the current implementation of the HO algorithm in NS-2.

Table 3.3 Study of RSS

	Average RSS (W)	Variance	Standard deviation
Situation A	$1{,}04\cdot10^{-8}$	$9{,}44\cdot10^{-19}$	$9{,}72\cdot10^{-10}$
Situation B	$2{,}31\cdot10^{-9}$	$4{,}404\cdot10^{-23}$	$6{,}636\cdot10^{-12}$
Situation C	$\mathbf{1{,}35\cdot10^{-9}}$	$2{,}689\cdot10^{-18}$	$1{,}639\cdot10^{-9}$
Current HO algorithm	$\mathbf{1{,}36\cdot10^{-9}}$	$2{,}594\cdot10^{-19}$	$1{,}61\cdot10^{-9}$

Results in table 3.3 shows that the RSS values obtained from the simulations applying the current HO algorithm correspond to the “situation C” (MN at the border of its cell).

From the previous study, a new algorithm (algorithm B) has been designed in order to analyze the impact of a different handover strategy over mobility. The MN will start looking for a new list of neighboring AP if the next conditions are accomplished: it has lost one data packet and it is located, at least, at a middle point between two cells. This last event is identified according to RSS values displayed in Table 3.3 (situation B: RSS = $2{,}31\cdot10^{-9}$ W). We want to stress here that this new algorithm is just a first step towards the design of a new HO strategy that will improve network performance.

In order to apply this new algorithm, “mac-802_11.cc” source file has been modified in NS-2.

CHAPTER 4. CELL RESIDENCE TIME

This last chapter provides statistics of the cell residence time (CRT) from results obtained from simulation with NS-2. Two different scenarios (i.e. 4 APs and 8 APs) have been analyzed, changing the mobility pattern and the handover algorithm.

4.1. Average CRT

In this section, the average CRT, the standard deviation and the coefficient of variation (CV) are provided for each HO algorithm and scenario.

4.1.1. Algorithm A

Table 4.1 and 4.2 show the results for the scenario with 4 APs and 8 APs respectively, using handover algorithm A. When the Random Waypoint mobility model is applied, average CRT increases and coefficients of variation decrease as pause time increases.

On the other hand, the application of Gauss-Markov mobility model provides higher CRT since this model calculates movements according to the previous ones generated and Random Waypoint generates random movements that change the trajectory of the MN suddenly.

Average CRT is lower in the scenario with 8 APs because the number of APs is twice and the simulation area does not change.

Table 4.1. Cell residence time in a scenario with 4 APs using handover algorithm A

4 APs Algorithm A	RWP_u-0	RWP_u-1	RWP_u-10	RWP_n-0	RWP_n-1	RWP_n10	GM_u	GM_n
Mean CRT (s)	168,85	174,98	180,97	163,56	184,03	195,50	280,16	272,39
Standard dev.	128,90	116,16	115,68	119,71	122,76	106,97	117,47	111,74
CV	0,763	0,664	0,639	0,732	0,667	0,547	0,419	0,41
Samples	512	494	477	528	469	442	308	317

Table 4.2. Cell residence time in a scenario with 8 APs using handover algorithm A

8 APs Algorithm A	RWP_u-0	RWP_u-1	RWP_u-10	RWP_n-0	RWP_n-1	RWP_n10	GM_u	GM_n
Mean CRT (s)	118,32	133,68	137,71	124,03	128,67	138,22	224,21	214,17
Standard dev.	93,08	96,58	101,68	98,20	101,12	103,59	93,89	86,48
CV	0,787	0,722	0,738	0,791	0,786	0,749	0,419	0,163
Samples	730	646	627	697	671	625	385	403

4.1.2. Algorithm B

Tables 4.3 and 4.4 show the results for the scenario with 4 APs and 8 APs respectively, using handover algorithm B. Once again, average CRT increases when pause time increases in Random Waypoint mobility model. Furthermore, GM provides higher CRT.

Table 4.3. Cell residence time in a scenario with 4 APs using handover algorithm B

4 APs Algorithm B	RWP_u-0	RWP_u-1	RWP_u-10	RWP_n-0	RWP_n-1	RWP_n10	GM_u	GM_n
Mean CRT (s)	88,73	93,75	102,68	87,86	92,59	103,29	205,43	199,38
Standard dev.	78,58	76,59	80,61	71,25	77,02	84,28	83,58	81,14
CV	0.886	0,817	0,785	0,811	0,832	0,816	0,407	0,407
Samples	974	922	841	983	933	836	421	433

Table 4.4. Cell residence time in a scenario with 8 APs using handover algorithm B

8 APs Algorithm B	RWP_u-0	RWP_u-1	RWP_u-10	RWP_n-0	RWP_n-1	RWP_n10	GM_u	GM_n
Mean CRT (s)	62,89	69,24	73,33	67,67	70,51	79,29	134,42	124,01
Standard dev.	48,66	49,67	53,29	57,42	55,80	67,56	61,64	49,21
CV	0,774	0,717	0,727	0,848	0,791	0,852	0,458	0,397
Samples	1374	1248	1178	1277	1225	1090	643	697

4.1.2. Comparison

Figure 4.1 compares the average CRT of the current HO algorithm implementation and the new one designed. Algorithm A is represented in black columns and algorithm B is represented in grey columns.

As it was supposed, the new algorithm leads to a decrease in the time the MN is in a determined cell because the handover process starts after one retransmission if the MN is located at a middle point between two cells. Then, if the power received from the actual AP is not good enough and a new AP with better RSS is available, the handover process will be realized. This situation causes an increase in the HO rate.

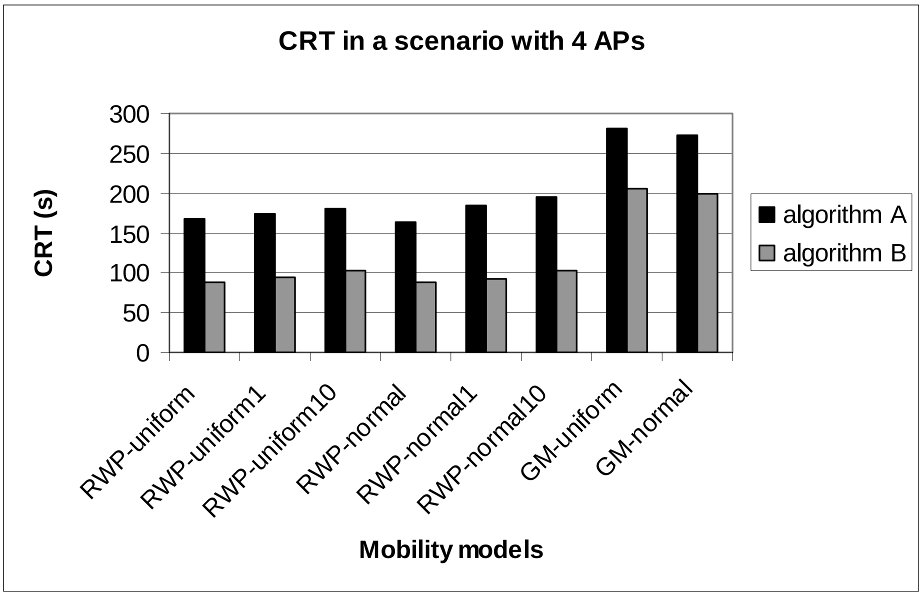

Fig. 4.1 CRT comparative in a scenario with 4 APs

Figure 4.2 illustrates the average CRT for the two handover algorithms for the scenario composed by 8 APs.

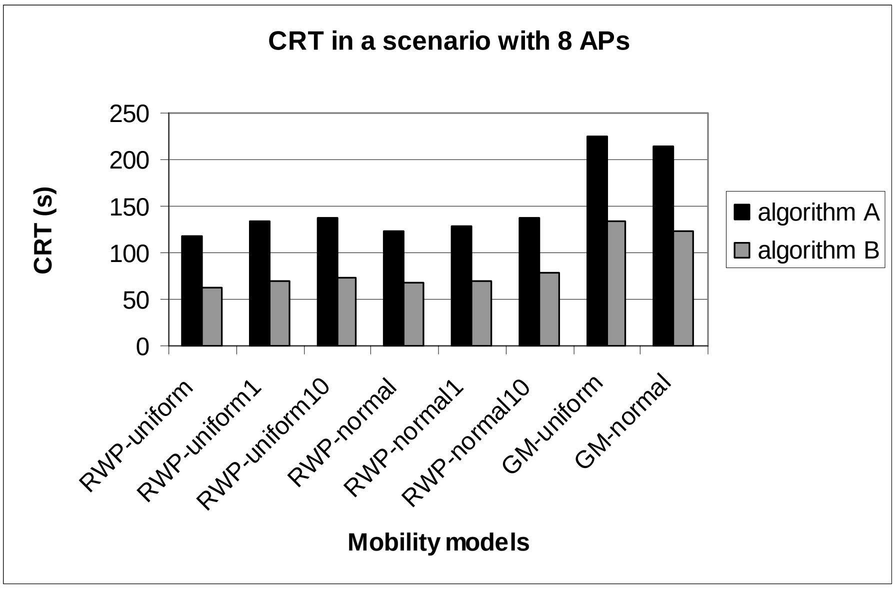

Fig. 4.2 CRT comparative in a scenario with 8 APs

What is common in the scenario with 4 APs and in the scenario with 8 APs is that the cell residence time is lower when the Random Waypoint mobility model is applied, causing a higher number of handovers. This event occurs because the Random Waypoint model generates random movements without memory and the Gauss-Markov model calculates new movements according to the previous ones and it causes more changes in the trajectory of a MN.

4.2. CRT distribution

Finally, the distribution of the CRT values is analyzed, showing the percentage of time the MN resides on a cell.

Figures 4.3 and 4.4 show the distribution of the CRT values in a scenario composed by 4 APs, when the Random Waypoint mobility model with pause time set to 0 seconds is applied. The two handover strategies, algorithms A and B, are represented. While algorithm A is represented in black columns, algorithm B is represented in grey columns.

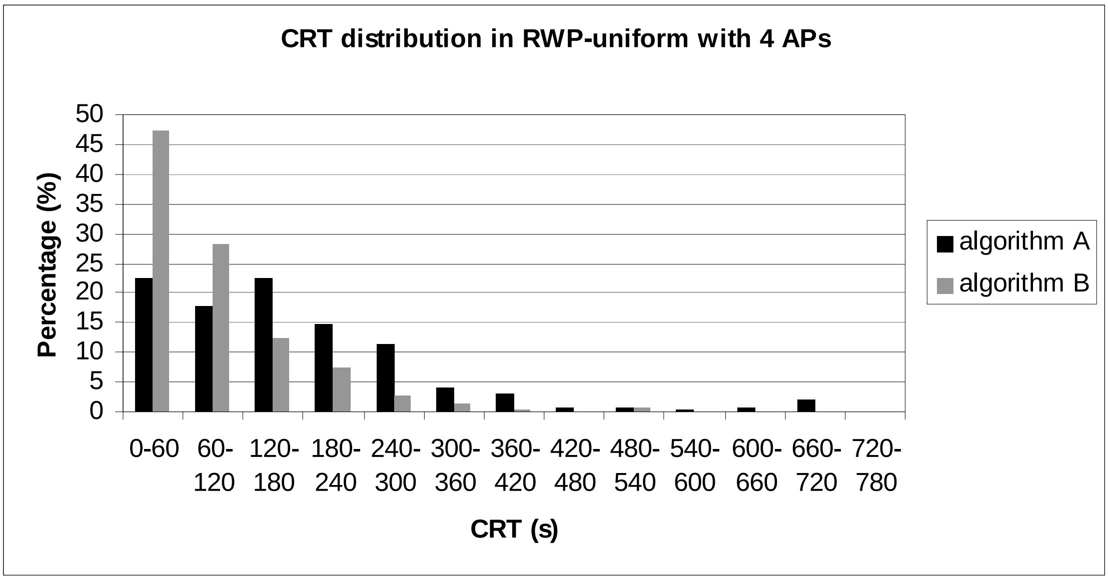

Fig. 4.3 CRT distribution using Random Waypoint uniform model with 4 APs

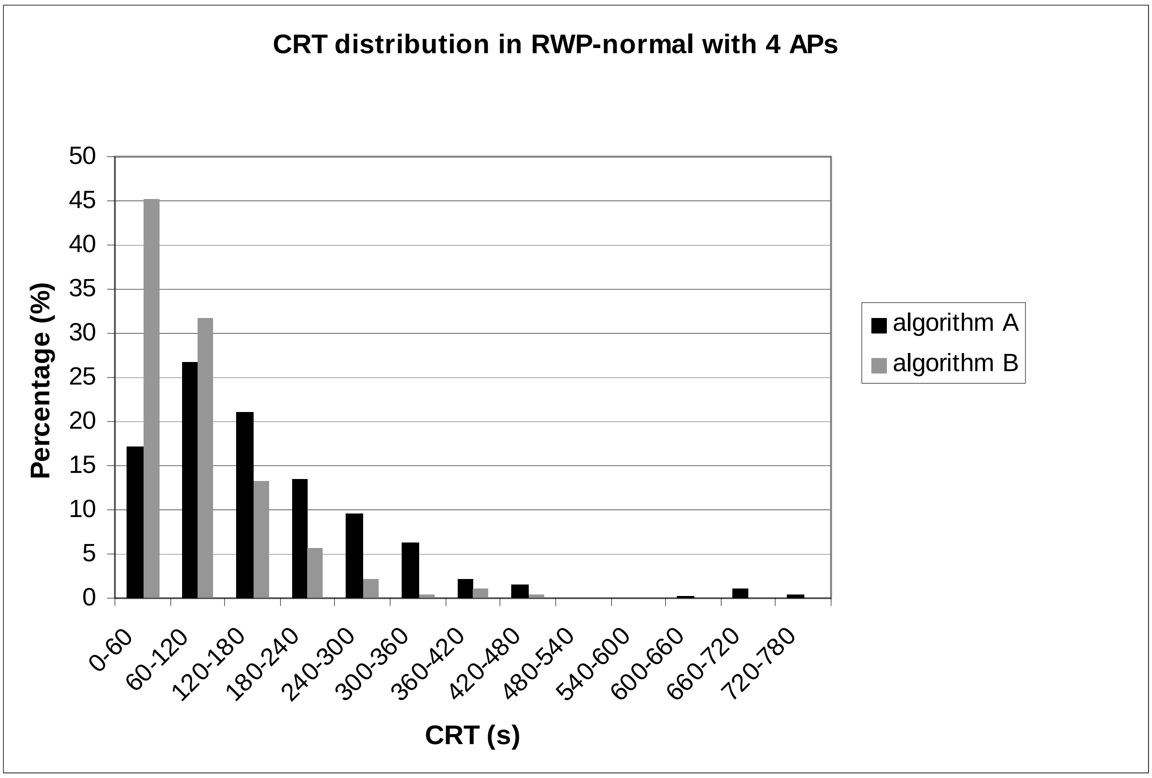

Fig. 4.4 CRT distribution using Random Waypoint normal model with 4 APs

The next two figures, 4.5 and 4.6, represent the distribution of the CRT values in a scenario with 8 APs when the Random Waypoint mobility model with pause time set to 0 seconds is applied.

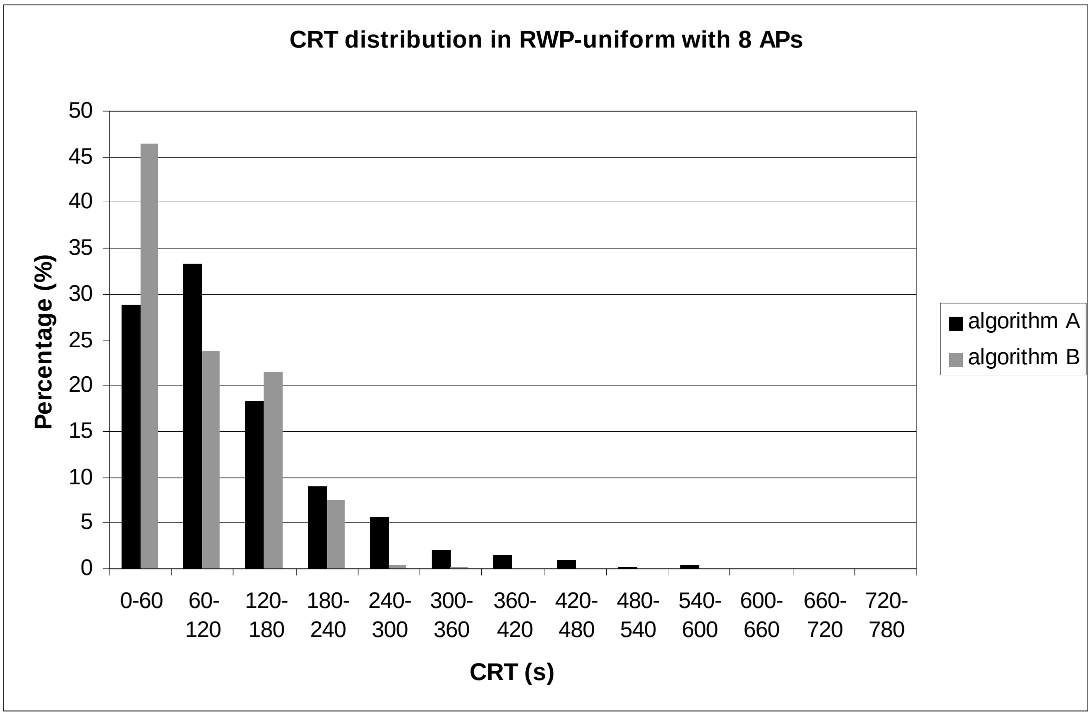

Fig. 4.5 CRT distribution using Random Waypoint uniform model with 8 APs

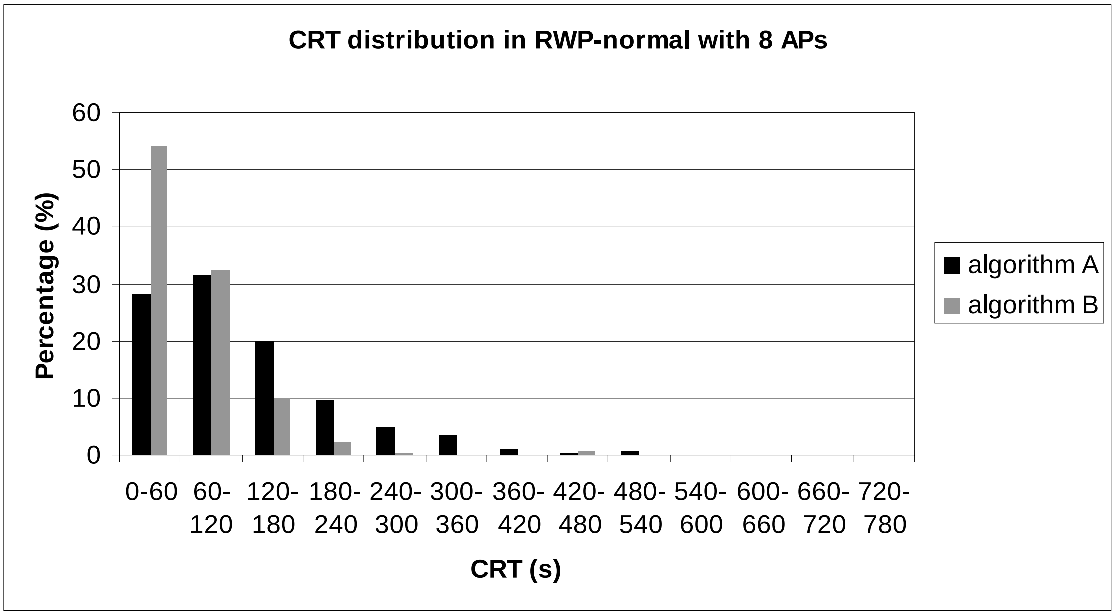

Fig. 4.6 CRT distribution using Random Waypoint normal model with 8 APs

As the graphics show, for each scenario shorter CRT appears when the second algorithm is used. Shorter cell residence times (i.e. less than 60 seconds) can be due to the ping-pong effect, where the MN associates and quickly disassociates among neighboring AP, since their RSS is continuously varying. Further analysis is needed in order to determine which the underlying effect is: this is out of the scope of this work.

The next figures illustrate the CRT distribution for the Gauss-Markov mobility model, firstly with 4 APs and then with 8 APs. Once again, the graphics compare CRT values in algorithms A and B. While figures 4.7 and 4.8 illustrate the comparative CRT distribution in a scenario with 4 APs, figures 4.9 and 4.10 represent it in the scenario with 8 APs. In both cases, pause time is set to 0 seconds.

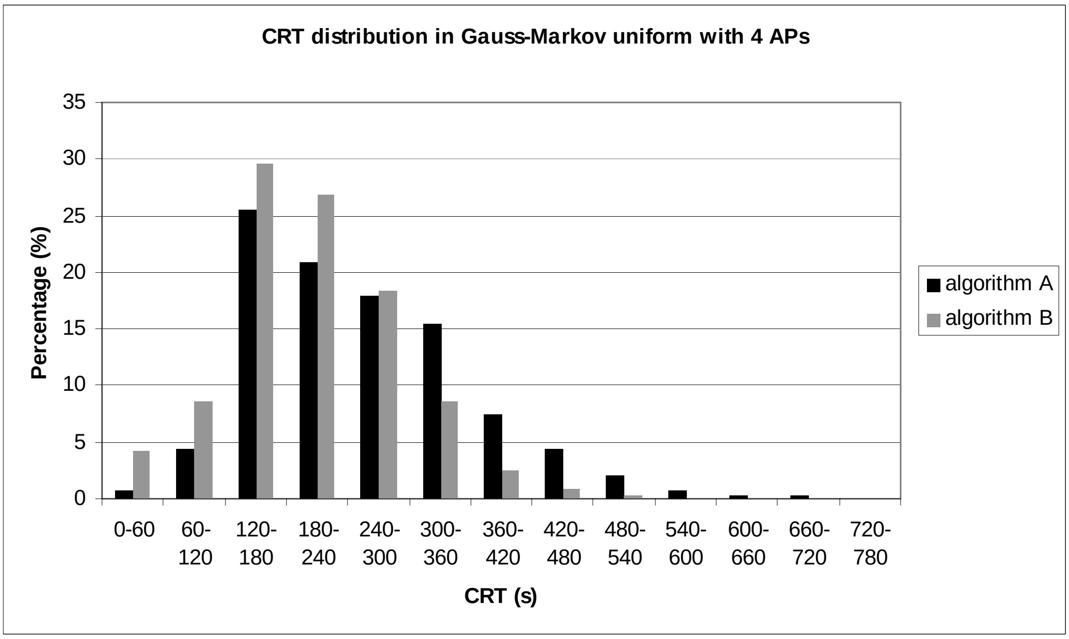

Fig. 4.7 CRT distribution using Gauss-Markov uniform model with 4 APs

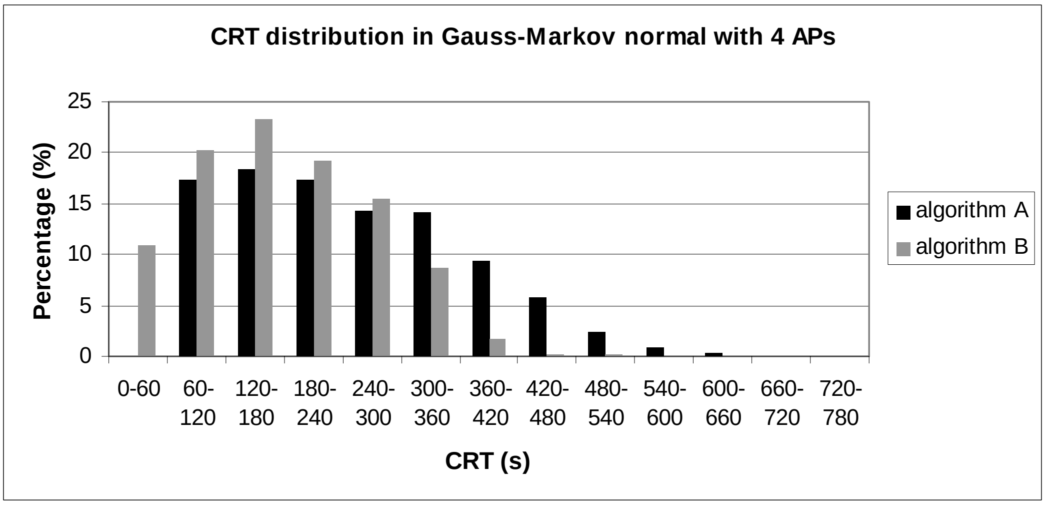

Fig. 4.8 CRT distribution using Gauss-Markov normal model with 4 APs

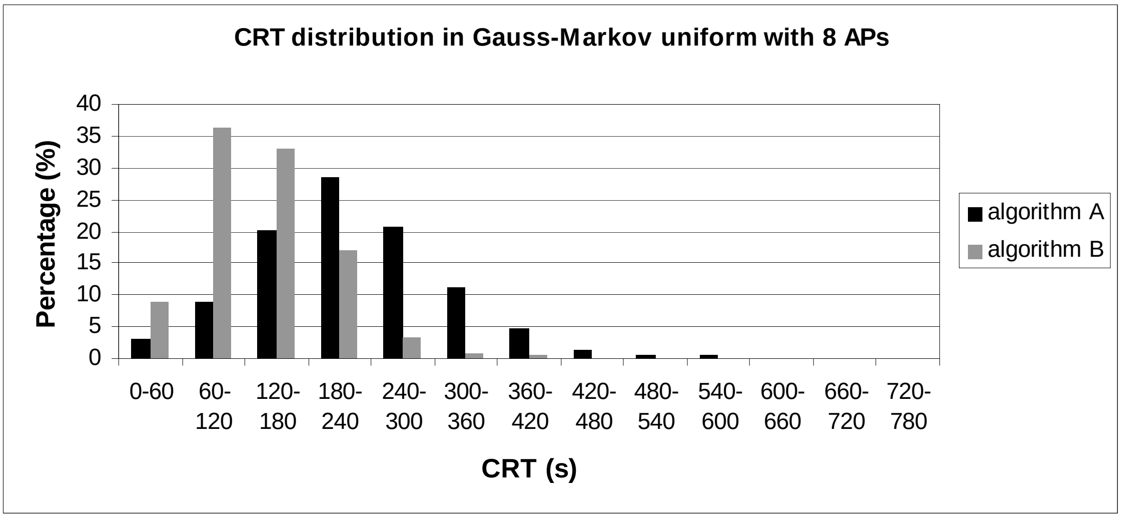

Fig. 4.9 CRT distribution using Gauss-Markov uniform model with 8 APs

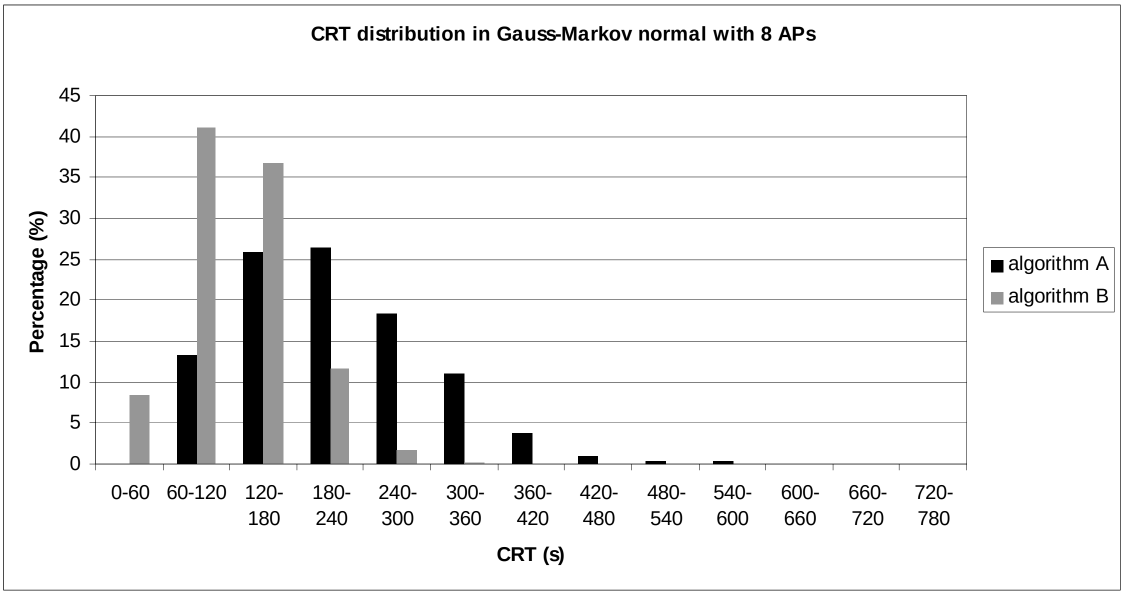

Fig. 4.10 CRT distribution using Gauss-Markov normal model with 8 APs

As the last four figures show, algorithm B causes shorter cell residence times which can be due to the ping-pong effect Once again, the application of the second algorithm causes CRT values less expanded and an increases the smaller CRT. However, the difference between the two algorithms in the CRT distribution is less significant when the Gauss-Markov model is applied in the scenario with 8 APs.

CHAPTER 5. CONCLUSIONS AND ENVIRONMENTAL IMPACT

5.1. Conclusions

In this project a study of infrastructure WLANs has been realized through the NS-2 simulator, applying different scenarios and mobility models.

The first step was to run simulations in infrastructure mode. However, it was checked that the current version of NS-2 does not support this simulation mode. After looking for many solutions, it was applied a patch for NS-2 infrastructure mode and some modifications in the source code to implement this configuration.

Once the infrastructure mode problem was solved and tested in different scenarios, the second part of the project was to create the scenarios for the final simulations: one of them with 4 APs, offering full coverage in the simulated area, and another one with 8 APs, simulating overcoverage in order to guarantee higher capacity for higher density of users. As for the mobility, the models applied were Random Waypoint mobility model and Gauss-Markov mobility model, changing the speed distribution. The aim was to evaluate the CRT applying totally random movements (Random Waypoint) and movements dependent on the previous movements (Gauss-Markov).

Another important task was to analyze how the current handover strategy works in order to change it designing a new algorithm. For this purpose, it was necessary a previous study of the RSS in order to determine the position of a MN when the handover is produced.

Finally, results were exposed in tables and graphically evaluating the cell residence time. CRT is lower if the MN moves according to a Random Waypoint mobility model than if the scenario is run with the Gauss-Markov model. On the other hand, the new algorithm is more restrictive than the current one and leads to a decrease in CRT. Moreover, using the new handover strategy causes CRT values less expanded.

To summarize, this project has applied infrastructure mode in NS-2 and has analyzed the impact of mobility models in WLAN indoor environments on handover process, achieving the main goals proposed initially. Furthermore, it has been created a new implementation of the handover algorithm to evaluate its impact on the mobility.

5.2. Environmental impact

The use of WLAN linked to this project does not have environmental impact since it is designed for indoor environments.

BIBLIOGRAPHY AND REFERENCES

[1] Andreu, F. Pellejero, I. and Lesta, A., “Labores de estandarización, Breve historia” in *Redes WLAN, Fundamentos y aplicaciones de seguridad*, pp. 17-18.

[2] WiFi modes of operation (802.11 or Wi-Fi). Web: http://en.kioskea.net/contents/wifi/wifimodes.php3 Last visit:12/12/08

[3] Crossroads, The ACM Student Magazine. Web: http://www.acm.org/crossroads/xrds9-4/wlan_abc.html Last visit: 24/12/08

[4] NETGEAR: Connect with Innovation. Web: http://kbserver.netgear.com/kb_web_files/N101519.asp Last visit: 5/12/08

[5] Camp, T., Boleng, J. and Davies V., “A survey of mobility models for ad hoc Network research”, in *Wireless Communication and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no 5, pp. 483-502, 2002.

[6] The Network Simulator – NS-2. Web: http://www.isi.edu/nsnam/ns/ Last visit: 30/11/08

[7] Marc Greis’ Tutorial for the UCB/LBNL/VINT Network Simulator “ns”. Web: http://www.isi.edu/nsnam/ns/tutorial/ Last visit:: 30/11/08

[8] Bonnmotion: A mobility scenario generation and analysis tool. Web: http://web.informatik.uni-bonn.de/IV/Mitarbeiter/dewaal/BonnMotion/ Last visit: 1/12/08

[9] “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”, Part 11 in *IEEE Standard 802.11*, 1999.

[10] Velayos, H. and Karlsson, G. “Techniques to Reduce IEEE 802.11b MAC Layer Handover Time,” in *IEEE ICC, vol. 7, pp. 3844-3848, June 2004.*

ANNEXES

Simulation scripts

The simulation scripts are shown in this section:

Scenario with 4 APs

```
#===========================================================
===========
# Parameters for 802.11 nodes
#===========================================================
===========

Antenna/OmniAntenna     set Gt_ 1           ;# Transmit antenna gain
Antenna/OmniAntenna     set Gr_ 1           ;# Receive antenna gain
Antenna/OmniAntenna    set ht_ 1.5          ;# Tx antenna heigth
Antenna/OmniAntenna    set hr_ 1.5          ;# Rx antenna heigth

Phy/WirelessPhy    set L_ 1.0           ;# System Loss Factor
Phy/WirelessPhy    set freq_ 2.412e9       ;# Channel 1: 2.412 GHz
Phy/WirelessPhy    set Pt_ 0.100          ;# Transmit Power (15dBm)
Phy/WirelessPhy       set CPThresh_ 10.0      ;# Collision threshold
Phy/WirelessPhy     set CSThresh_ 3.1622777e-12    ;# Carrier Sense Power (-
94dBm)
Phy/WirelessPhy    set RXThresh_ 9.79644e-10    ;# Receive Power Threshold

Mac/802_11           set basicRate_ 1Mb           ;# set this to 0 if want to use
bandwidth_ for
Mac/802_11       set dataRate_ 11Mb        ;# both control and data pkts
Mac/802_11       set CWMin_ 31            ;#
Mac/802_11       set CWMax_ 1023         ;#
Mac/802_11       set SlotTime_ 0.000020        ;# 20us
Mac/802_11       set SIFS_ 0.000010        ;# 10us
Mac/802_11       set PreambleLength_ 144     ;# 144 bit
Mac/802_11       set PLCPHeaderLength_ 48     ;# 48 bits
Mac/802_11       set PLCPDataRate_ 1.0e6     ;# 1Mbps
Mac/802_11          set RTSThreshold_ 2500           ;# bytes: changed so that
RTS/CTS has been disabled ENRI
Mac/802_11       set ShortRetryLPLCPimit_ 7     ;# retransmissions
Mac/802_11       set LongRetryLimit_ 4        ;# retransmissions

############################################################

set opt(chan)   Channel/WirelessChannel        ;# channel type
set opt(prop)   Propagation/TwoRayGround       ;# radio-propagation model
set opt(netif)  Phy/WirelessPhy              ;# network interface type
set opt(mac)    Mac/802_11                  ;# MAC type
set opt(ifq)    Queue/DropTail/PriQueue        ;# interface queue type
set opt(ll)     LL                             ;# link layer type
set opt(ant)    Antenna/OmniAntenna            ;# antenna model
set opt(ifqlen)         50                     ;# max packet in ifq
set opt(nn)             1                      ;# number of mobilenodes
set opt(adhocRouting)   DSDV                   ;# routing protocol
set opt(cp)     ""                             ;# cp file not used
set         opt(sc)                    "/home/TFC-jesus/ns-allinone-2.31/NS-
2.31/tcl/ex/802_11/Scenario-mobility.tcl"                      ;#/home/TFC-
jesus/scengen/scen/prueba"                     ;# node movement file.
set opt(x)      280                            ;# x coordinate of topology
set opt(y)      280                            ;# y coordinate of topology
set opt(seed)   0.0                            ;# random seed
set opt(stop)   86400                          ;# time to stop simulation

set num_wired_nodes      2
#set num_bs_nodes       4  ; this is not really used here.
#set opt(ftp1-start)      2.0
#
=============================================================

# check for boundary parameters and random seed
if { $opt(x) == 0 || $opt(y) == 0 } {
        puts "No X-Y boundary values given for wireless topology\n"
}
if {$opt(seed) > 0} {
        puts "Seeding Random number generator with $opt(seed)\n"
        ns-random $opt(seed)
}

# create simulator instance
set ns_   [new Simulator]

# set up for hierarchical routing
$ns_ node-config -addressType hierarchical

AddrParams set domain_num_ 5           ;# number of domains
lappend cluster_num 2 1 1 1 1          ;# number of clusters in each domain
AddrParams set cluster_num_ $cluster_num
lappend eilastlevel  1 1 1 2 1 1       ;# number of nodes in each cluster
AddrParams set nodes_num_ $eilastlevel ;# of each domain

set tracefd  [open handoverb1.tr w]
set namtrace [open handoverb1.nam w]
#$ns_ use-newtrace
$ns_ trace-all $tracefd
$ns_ namtrace-all-wireless $namtrace $opt(x) $opt(y)

# Create topography object
set topo   [new Topography]
```

```
# define topology
$topo load_flatgrid $opt(x) $opt(y)

# create God
create-god [expr $opt(nn) + 5]

#create wired nodes
set temp {0.0.0 0.1.0}          ;# hierarchical addresses
for {set i 0} {$i < $num_wired_nodes} {incr i} {
   set W($i) [$ns_ node [lindex $temp $i]]
}

$W(0) set X_ 140.0
$W(0) set Y_ 140.0
$W(0) set Z_ 0.0
$ns_ at 0.01 "$W(0) label \"SW\""

$W(1) set X_ 140.0
$W(1) set Y_ 120.0
$W(1) set Z_ 0.0
$ns_ at 0.01 "$W(1) label \"Host\""
set node_(1) $W(1)

# Configure for ForeignAgent and HomeAgent nodes
$ns_ node-config -mobileIP ON \
             -adhocRouting $opt(adhocRouting) \
             -llType $opt(ll) \
             -macType $opt(mac) \
             -ifqType $opt(ifq) \
             -ifqLen $opt(ifqlen) \
             -antType $opt(ant) \
             -propType $opt(prop) \
             -phyType $opt(netif) \
             -channelType $opt(chan) \
                 -topoInstance $topo \
             -wiredRouting ON \
                 -agentTrace OFF \
             -routerTrace OFF \
             -macTrace ON

# Create AP1, AP2, AP3 and AP4

set AP2 [$ns_ node 1.0.0]          ;# disable random motion
set AP3 [$ns_ node 2.0.0]
set AP4 [$ns_ node 3.0.0]
set AP5 [$ns_ node 4.0.0]

$AP2 random-motion 0
$AP3 random-motion 0
```

```
$AP4 random-motion 0
$AP5 random-motion 0


set mac_(2) [$AP2 getMac 0]
$mac_(2) ap [$mac_(2) id]
$mac_(2) set RTSThreshold_ 3000

set mac_(3) [$AP3 getMac 0]
$mac_(3) ap [$mac_(3) id]
$mac_(3) set RTSThreshold_ 3000

set mac_(4) [$AP4 getMac 0]
$mac_(4) ap [$mac_(4) id]
$mac_(4) set RTSThreshold_ 3000

set mac_(5) [$AP5 getMac 0]
$mac_(5) ap [$mac_(5) id]
$mac_(5) set RTSThreshold_ 3000

# Position (fixed) for base-station nodes (HA & FA).
$AP2 set X_ 75.00000000000
$AP2 set Y_ 205.0000000000
$AP2 set Z_ 0.000000000000

$AP3 set X_ 205.000000000000
$AP3 set Y_ 205.000000000000
$AP3 set Z_ 0.000000000000

$AP4 set X_ 205.000000000000
$AP4 set Y_ 75.000000000000
$AP4 set Z_ 0.000000000000

$AP5 set X_ 75.000000000000
$AP5 set Y_ 75.000000000000
$AP5 set Z_ 0.000000000000

$ns_ at 0.01 "$AP5 label \"AP\""
$ns_ at 0.01 "$AP2 label \"AP\""
$ns_ at 0.01 "$AP3 label \"AP\""
$ns_ at 0.01 "$AP4 label \"AP\""

# create a mobilenode that would be moving between HA and FA.
# note address of MH indicates its in the same domain as HA.
$ns_ node-config -wiredRouting OFF

set MN1 [$ns_ node 2.0.1]
set node_(6) $MN1

set mac_(6) [$node_(6) getMac 0]
```

```
$mac_(6) set RTSThreshold_ 3000
set AP4address [AddrParams addr2id [$AP4 node-addr]];#Asociación con el AP1
[$MN1 set regagent_] set home_agent_ $AP4address

$node_(6) random-motion 0

$ns_ at 0.01 "$MN1 label \"MN\""

$mac_(6) ScanType ACTIVE
#$mac_(2) ScanType ACTIVE

# create links between wired and BaseStation nodes

$ns_ duplex-link $W(0) $AP2 5Mb 2ms DropTail
$ns_ duplex-link $W(0) $AP3 5Mb 2ms DropTail
$ns_ duplex-link $W(0) $AP4 5Mb 2ms DropTail
$ns_ duplex-link $W(0) $AP5 5Mb 2ms DropTail
$ns_ duplex-link $W(0) $W(1) 5Mb 2ms DropTail

$ns_ duplex-link-op $W(0) $AP2 orient left-up
$ns_ duplex-link-op $W(0) $AP3 orient left-down
$ns_ duplex-link-op $W(0) $AP4 orient right-down
$ns_ duplex-link-op $W(0) $AP5 orient right-up
$ns_ duplex-link-op $W(0) $W(1) orient down

# setup TCP connections between a wired node and the MobileHost

set udp1(6) [new Agent/UDP]

$ns_ attach-agent $node_(6) $udp1(6)
set cbr1(6) [new Application/Traffic/CBR]
$cbr1(6) attach-agent $udp1(6)

set udp1(1) [new Agent/UDP]

$ns_ attach-agent $node_(1) $udp1(1)
set cbr1(1) [new Application/Traffic/CBR]
$cbr1(1) attach-agent $udp1(1)

set base0 [new Agent/Null]
$ns_ attach-agent $node_(1) $base0
$ns_ connect $udp1(6) $base0
$ns_ at 0.0 "$cbr1(6) start"

$ns_ at $opt(stop).0 "stop"
$ns_ at $opt(stop).0 "puts \"NS EXITING...\" ; $ns_ halt"

$ns_ at $opt(stop).0 "puts \"NS EXITING...\" ; $ns_ halt"
```

```
# source connection-pattern and node-movement scripts
if { $opt(cp) == "" } {
        puts "*** NOTE: no connection pattern specified."
      set opt(cp) "none"
} else {
        puts "Loading connection pattern..."
        source $opt(cp)
}
if { $opt(sc) == "" } {
        puts "*** NOTE: no scenario file specified."
      set opt(sc) "none"
} else {
        puts "Loading scenario file..."
        source $opt(sc)
        puts "Load complete..."
}

# Define initial node position in nam

for {set i 6} {$i < 7} {incr i} {

   # 20 defines the node size in nam, must adjust it according to your
   # scenario
   # The function must be called after mobility model is defined

   $ns_ initial_node_pos $node_($i) 5
}

# Tell all nodes when the simulation ends

$ns_ at $opt(stop).0 "$node_(6) reset";
$ns_ at $opt(stop).0 "$node_(1) reset";

$ns_ at $opt(stop).0 "$AP2 reset";
$ns_ at $opt(stop).0 "$AP3 reset";
$ns_ at $opt(stop).0 "$AP4 reset";
$ns_ at $opt(stop).0 "$AP5 reset";

$ns_ at $opt(stop).0002 "puts \"NS EXITING...\" ; $ns_ halt"
$ns_ at $opt(stop).0001 "stop"
proc stop {} {
   global ns_ tracefd namtrace
   close $tracefd
   close $namtrace
}

# some useful headers for tracefile
puts $tracefd "M 0.0 nn $opt(nn) x $opt(x) y $opt(y) rp \
        $opt(adhocRouting)"
```

```
puts $tracefd "M 0.0 sc $opt(sc) cp $opt(cp) seed $opt(seed)"
puts $tracefd "M 0.0 prop $opt(prop) ant $opt(ant)"

puts "Starting Simulation..."
$ns_ run
```

Scenario with 8 APs

```
#=================================================================
# Parameters for 802.11 nodes
#=================================================================

Antenna/OmniAntenna     set Gt_ 1            ;# Transmit antenna gain
Antenna/OmniAntenna     set Gr_ 1            ;# Receive antenna gain
Antenna/OmniAntenna    set ht_ 1.5            ;# Tx antenna heigth
Antenna/OmniAntenna    set hr_ 1.5            ;# Rx antenna heigth

Phy/WirelessPhy     set L_ 1.0            ;# System Loss Factor
Phy/WirelessPhy     set freq_ 2.412e9        ;# Channel 1: 2.412 GHz
Phy/WirelessPhy     set Pt_ 0.075;#0.100            ;# Transmit Power (15dBm)
Phy/WirelessPhy        set CPThresh_ 10.0        ;# Collision threshold
Phy/WirelessPhy     set CSThresh_ 3.1622777e-12    ;# Carrier Sense Power (-
94dBm)
Phy/WirelessPhy     set RXThresh_ 9.79644e-10    ;# Receive Power Threshold

Mac/802_11            set basicRate_ 1Mb            ;# set this to 0 if want to use
bandwidth_ for
Mac/802_11        set dataRate_ 11Mb        ;# both control and data pkts
Mac/802_11        set CWMin_ 31            ;
Mac/802_11        set CWMax_ 1023        ;#
Mac/802_11        set SlotTime_ 0.000020        ;# 20us
Mac/802_11        set SIFS_ 0.000010        ;# 10us
Mac/802_11        set PreambleLength_ 144     ;# 144 bit
Mac/802_11        set PLCPHeaderLength_ 48     ;# 48 bits
Mac/802_11        set PLCPDataRate_ 1.0e6     ;# 1Mbps
Mac/802_11           set RTSThreshold_ 2500            ;# bytes: changed so that
RTS/CTS has been disabled ENRI
Mac/802_11        set ShortRetryLPLCPimit_ 7     ;# retransmissions
Mac/802_11        set LongRetryLimit_ 4        ;# retransmissions

#################################################################
################################

set opt(chan)   Channel/WirelessChannel        ;# channel type
set opt(prop)   Propagation/TwoRayGround       ;# radio-propagation model
set opt(netif)  Phy/WirelessPhy              ;# network interface type
set opt(mac)    Mac/802_11                   ;# MAC type
set opt(ifq)    Queue/DropTail/PriQueue        ;# interface queue type
set opt(ll)     LL                         ;# link layer type
set opt(ant)    Antenna/OmniAntenna           ;# antenna model
set opt(ifqlen)        50                  ;# max packet in ifq
set opt(nn)            1                   ;# number of mobilenodes
set opt(adhocRouting)   DSDV                ;# routing protocol

set opt(cp)     ""                         ;# cp file not used
set         opt(sc)                              "/home/TFC-jesus/ns-allinone-2.31/NS-
2.31/tcl/ex/802_11/gaussmarkov-uniform1.ns_movements"          ;#/home/TFC-
jesus/scengen/scen/prueba"                        ;# node movement file.

set opt(x)      280                        ;# x coordinate of topology
set opt(y)      280                        ;# y coordinate of topology
set opt(seed)   0.0                        ;# random seed
set opt(stop)   86400                ;# time to stop simulation

set num_wired_nodes      2
#set num_bs_nodes       8  ; this is not really used here.
#set opt(ftp1-start)      2.0
#
============================================================
==========

# check for boundary parameters and random seed
if { $opt(x) == 0 || $opt(y) == 0 } {
	puts "No X-Y boundary values given for wireless topology\n"
}
if {$opt(seed) > 0} {
	puts "Seeding Random number generator with $opt(seed)\n"
	ns-random $opt(seed)
}

# create simulator instance
set ns_   [new Simulator]

# set up for hierarchical routing
$ns_ node-config -addressType hierarchical

AddrParams set domain_num_ 9          ;# number of domains
lappend cluster_num 2 1 1 1 1 1 1 1 1             ;# number of clusters in each
domain
AddrParams set cluster_num_ $cluster_num
lappend eilastlevel  1 1 1 2 1 1 1 1 1 1        ;# number of nodes in each cluster
AddrParams set nodes_num_ $eilastlevel ;# of each domain

set tracefd  [open 8handover.tr w]
set namtrace [open 8handover.nam w]
#$ns_ use-newtrace
$ns_ trace-all $tracefd
$ns_ namtrace-all-wireless $namtrace $opt(x) $opt(y)
```

```
# Create topography object
set topo   [new Topography]

# define topology
$topo load_flatgrid $opt(x) $opt(y)

# create God
create-god [expr $opt(nn) + 9]

#create wired nodes
set temp {0.0.0 0.1.0}           ;# hierarchical addresses
for {set i 0} {$i < $num_wired_nodes} {incr i} {
   set W($i) [$ns_ node [lindex $temp $i]]
}

$W(0) set X_ 140.0
$W(0) set Y_ 140.0
$W(0) set Z_ 0.0
$ns_ at 0.01 "$W(0) label \"SW\""

$W(1) set X_ 140.0
$W(1) set Y_ 120.0
$W(1) set Z_ 0.0
$ns_ at 0.01 "$W(1) label \"Host\""
set node_(1) $W(1)

# Configure for ForeignAgent and HomeAgent nodes
$ns_ node-config -mobileIP ON \
             -adhocRouting $opt(adhocRouting) \
             -llType $opt(ll) \
             -macType $opt(mac) \
             -ifqType $opt(ifq) \
             -ifqLen $opt(ifqlen) \
             -antType $opt(ant) \
             -propType $opt(prop) \
             -phyType $opt(netif) \
             -channelType $opt(chan) \
                 -topoInstance $topo \
             -wiredRouting ON \
                 -agentTrace OFF \
             -routerTrace OFF \
             -macTrace ON



# Create AP2, AP3, AP4, AP5, AP6, AP7, AP8 and AP9

set AP2 [$ns_ node 1.0.0]          ;# disable random motion
set AP3 [$ns_ node 2.0.0]
set AP4 [$ns_ node 3.0.0]
set AP5 [$ns_ node 4.0.0]
set AP6 [$ns_ node 5.0.0]
set AP7 [$ns_ node 6.0.0]
set AP8 [$ns_ node 7.0.0]
set AP9 [$ns_ node 8.0.0]

$AP2 random-motion 0
$AP3 random-motion 0
$AP4 random-motion 0
$AP5 random-motion 0
$AP6 random-motion 0
$AP7 random-motion 0
$AP8 random-motion 0
$AP9 random-motion 0

set mac_(2) [$AP2 getMac 0]
$mac_(2) ap [$mac_(2) id]
$mac_(2) set RTSThreshold_ 3000

set mac_(3) [$AP3 getMac 0]
$mac_(3) ap [$mac_(3) id]
$mac_(3) set RTSThreshold_ 3000

set mac_(4) [$AP4 getMac 0]
$mac_(4) ap [$mac_(4) id]
$mac_(4) set RTSThreshold_ 3000

set mac_(5) [$AP5 getMac 0]
$mac_(5) ap [$mac_(5) id]
$mac_(5) set RTSThreshold_ 3000

set mac_(6) [$AP6 getMac 0]
$mac_(6) ap [$mac_(6) id]
$mac_(6) set RTSThreshold_ 3000

set mac_(7) [$AP7 getMac 0]
$mac_(7) ap [$mac_(7) id]
$mac_(7) set RTSThreshold_ 3000

set mac_(8) [$AP8 getMac 0]
$mac_(8) ap [$mac_(8) id]
$mac_(8) set RTSThreshold_ 3000

set mac_(9) [$AP9 getMac 0]
$mac_(9) ap [$mac_(9) id]
$mac_(9) set RTSThreshold_ 3000

# Position (fixed) for base-station nodes.

$AP2 set X_ 75.00000000000
```

```
$AP2 set Y_ 205.0000000000
$AP2 set Z_ 0.000000000000

$AP3 set X_ 205.000000000000
$AP3 set Y_ 205.000000000000
$AP3 set Z_ 0.000000000000

$AP4 set X_ 205.000000000000
$AP4 set Y_ 75.000000000000
$AP4 set Z_ 0.000000000000

$AP5 set X_ 75.000000000000
$AP5 set Y_ 75.000000000000
$AP5 set Z_ 0.000000000000

$AP6 set X_ 80.000000000000
$AP6 set Y_ 140.000000000000
$AP6 set Z_ 0.000000000000

$AP7 set X_ 140.000000000000
$AP7 set Y_ 200.000000000000
$AP7 set Z_ 0.000000000000

$AP8 set X_ 200.000000000000
$AP8 set Y_ 140.000000000000
$AP8 set Z_ 0.000000000000

$AP9 set X_ 140.000000000000
$AP9 set Y_ 80.000000000000
$AP9 set Z_ 0.000000000000

$ns_ at 0.01 "$AP5 label \"AP\""
$ns_ at 0.01 "$AP2 label \"AP\""
$ns_ at 0.01 "$AP3 label \"AP\""
$ns_ at 0.01 "$AP4 label \"AP\""
$ns_ at 0.01 "$AP6 label \"AP\""
$ns_ at 0.01 "$AP7 label \"AP\""
$ns_ at 0.01 "$AP8 label \"AP\""
$ns_ at 0.01 "$AP9 label \"AP\""

# create a mobilenode that would be moving.
$ns_ node-config -wiredRouting OFF

set MN1 [$ns_ node 2.0.1]
set node_(10) $MN1

set mac_(10) [$node_(10) getMac 0]

$mac_(10) set RTSThreshold_ 3000
```

```
set AP4address [AddrParams addr2id [$AP4 node-addr]];
[$MN1 set regagent_] set home_agent_ $AP4address

$node_(10) random-motion 0
$ns_ at 0.01 "$MN1 label \"MN\""
$mac_(10) ScanType ACTIVE
# create links between wired and BaseStation nodes

$ns_ duplex-link $W(0) $AP2 5Mb 2ms DropTail
$ns_ duplex-link $W(0) $AP3 5Mb 2ms DropTail
$ns_ duplex-link $W(0) $AP4 5Mb 2ms DropTail
$ns_ duplex-link $W(0) $AP5 5Mb 2ms DropTail
$ns_ duplex-link $W(0) $AP6 5Mb 2ms DropTail
$ns_ duplex-link $W(0) $AP7 5Mb 2ms DropTail
$ns_ duplex-link $W(0) $AP8 5Mb 2ms DropTail
$ns_ duplex-link $W(0) $AP9 5Mb 2ms DropTail
$ns_ duplex-link $W(0) $W(1) 5Mb 2ms DropTail

$ns_ duplex-link-op $W(0) $AP2 orient left-up
$ns_ duplex-link-op $W(0) $AP3 orient left-down
$ns_ duplex-link-op $W(0) $AP4 orient right-down
$ns_ duplex-link-op $W(0) $AP5 orient right-up
$ns_ duplex-link-op $W(0) $AP6 orient left
$ns_ duplex-link-op $W(0) $AP7 orient down
$ns_ duplex-link-op $W(0) $AP8 orient right
$ns_ duplex-link-op $W(0) $AP9 orient up
$ns_ duplex-link-op $W(0) $W(1) orient down

# setup TCP connections between a wired node and the MobileHost


set udp1(10) [new Agent/UDP]

$ns_ attach-agent $node_(10) $udp1(10)
set cbr1(10) [new Application/Traffic/CBR]
$cbr1(10) attach-agent $udp1(10)


set udp1(1) [new Agent/UDP]

$ns_ attach-agent $node_(1) $udp1(1)
set cbr1(1) [new Application/Traffic/CBR]
$cbr1(1) attach-agent $udp1(1)

set base0 [new Agent/Null]

$ns_ attach-agent $node_(1) $base0

$ns_ connect $udp1(10) $base0
$ns_ at 0.0 "$cbr1(10) start"
```

```
$ns_ at $opt(stop).0 "stop"
$ns_ at $opt(stop).0 "puts \"NS EXITING...\" ; $ns_ halt"

$ns_ at $opt(stop).0 "puts \"NS EXITING...\" ; $ns_ halt"


# source connection-pattern and node-movement scripts
if { $opt(cp) == "" } {
        puts "*** NOTE: no connection pattern specified."
      set opt(cp) "none"
} else {
        puts "Loading connection pattern..."
        source $opt(cp)
}
if { $opt(sc) == "" } {
        puts "*** NOTE: no scenario file specified."
      set opt(sc) "none"
} else {
        puts "Loading scenario file..."
        source $opt(sc)
        puts "Load complete..."
}

# Define initial node position in nam

for {set i 10} {$i < 11} {incr i} {

   # 20 defines the node size in nam, must adjust it according to your
   # scenario
   # The function must be called after mobility model is defined

   $ns_ initial_node_pos $node_($i) 5
}

# Tell all nodes when the simulation ends

$ns_ at $opt(stop).0 "$node_(10) reset";
$ns_ at $opt(stop).0 "$node_(1) reset";

$ns_ at $opt(stop).0 "$AP2 reset";
$ns_ at $opt(stop).0 "$AP3 reset";
$ns_ at $opt(stop).0 "$AP4 reset";
$ns_ at $opt(stop).0 "$AP5 reset";
$ns_ at $opt(stop).0 "$AP6 reset";
$ns_ at $opt(stop).0 "$AP7 reset";
$ns_ at $opt(stop).0 "$AP8 reset";
$ns_ at $opt(stop).0 "$AP9 reset";

$ns_ at $opt(stop).0002 "puts \"NS EXITING...\" ; $ns_ halt"
$ns_ at $opt(stop).0001 "stop"
```

```
proc stop {} {
    global ns_ tracefd namtrace
    close $tracefd
    close $namtrace
}

# some useful headers for tracefile
puts $tracefd "M 0.0 nn $opt(nn) x $opt(x) y $opt(y) rp \
          $opt(adhocRouting)"
puts $tracefd "M 0.0 sc $opt(sc) cp $opt(cp) seed $opt(seed)"
puts $tracefd "M 0.0 prop $opt(prop) ant $opt(ant)"

puts "Starting Simulation..."
$ns_ run
```