Distributed AAA Framework for MANET: performance analysis

LOgiciels-**R**éseaux	**Sondes LARAFA** **Maryline LAURENT-MAKNAVICIUS**	09009-LOR 2009

Abstract

Access control AAA infrastructures are traditionally used by the service providers so as to charge their subscribers. Given the easiness and the cheapness of MANET deployment and provided that charging is possible, service providers are likely to offer their services over MANET. In previous works [1] and [2], we presented a distributed AAA framework for MANET. We propose to evaluate the overhead of this framework authentication protocol by modeling and simulating typical cases that are fairly representative of the reality and can easily be extended. Results show that when routes are already established, the authentication overhead is about three times smaller than the overhead due to routing. Hence the overhead doesn't appear to be an impediment to distributed AAA infrastructures implementation.
Key-words: Mobile Ad-Hoc Networks (MANET), access control, Distributed Coordination Function (DCF) basic access mechanism, modelization, simulations.

Résumé

Les infrastructures de contrôle d'accès de type AAA sont traditionnellement utilisées par les prestataires de services afin de facturer leurs abonnés. Compte tenu de la facilité de déploiement de MANET et de leur bas prix, et à condition que la facturation des abonnés soit possible, les prestataires de services sont susceptibles d'offrir leurs services au-dessus de MANET. Dans de précédents travaux de recherche [1, 2], nous avons présenté une architecture AAA distribuée pour MANET. Nous nous proposons d'évaluer les délais introduits par l'exécution du protocole d'authentification de cette architecture, et cela grâce à la modélisation et à la simulation de cas typiques qui sont assez représentatifs de la réalité et qui peuvent être facilement étendus. Les résultats montrent que lorsque les routes sont déjà établies, le délai d'authentification est environ trois fois plus petit que la surcharge due au routage. Ainsi, le délai d'authentification ne semble pas être un obstacle à la mise en oeuvre des infrastructures AAA distribuées.
Mots-clés : Réseaux ad-hoc, contrôle d'accès, DCF, modéelisation, simulations.

—oOo—

Sondes LARAFA	Maryline LAURENT-MAKNAVICIUS
Doctorant	Professeur

TELECOM & Management SudParis - Département LOR - CNRS
9 rue Charles Fourier 91011 Evry Cedex
{sondes.larafa|Maryline.Maknavicius}@it-sudParis.eu

Contents

List of Figures

List of Tables

Introduction

Mobile Ad-hoc Networks (MANET) are basically wireless networks where terminals are mobile and contribute themselves to the routing operations of the network. MANET are self-configuring and infrastructure-less networks, with no need for any centralized entities and operators' management.

The easiness of deployment and the resulting financial gain are among the most interesting features of ad-hoc networks. Service providers and network operators are highly likely to take advantage of ad-hoc networks by providing to ad-hoc customers their ordinary and newly-defined services. Therefore access control infrastructures (e.g. AAA: Authentication, Authorization, and Accounting) are a hot topic in this kind of networks as they will help to support subscribers charging.

In two previous articles[1] and [2], we proposed a theoretical AAA framework that allows a joining node JN to authenticate itself to a group of AAA servers in a MANET. When the authentication succeeds, the servers deliver an Access Token to the JN thanks to which the neighboring nodes can check the legitimacy of the JN, before granting access.

An authentication protocol is executed between the JN and the AAA servers during the authentication phase. This protocol is located at the upper-layer of the TCP/IP stack. It is so strongly conditioned by the performance of the underlying layers. Moreover the more the number of the AAA servers increases, the bigger the number of exchanged messages is. Hence, there is an interest analyzing the overhead inherent to this protocol. This is the purpose of our report.

The report is outlined as follows. In the first chapter, we give a reminder of the distributed AAA infrastructure that we detailed in [1] and [2]. In addition we present some important features of the basic access mechanism of DCF (Distributed Coordination Function) in order to understand what follows. The two following chapters deal with the evaluation of the overhead using two methods: modelization and simulations. The second chapter is dedicated to the modelization part where we make some hypothesis to establish our model. The third chapter is dedicated to the simulations part. NS simulations led us to adjust our model and validate it thereafter. The non-modified model remains, however, more faithful to the reality and performs better than the modified model. In the third chapter, two simulation scenarios were considered, the case where nodes are motionless and the case where nodes move. Finally, the fourth chapter examines the influence of

the computing time within the nodes on the total overhead due to the authentication protocol.

1 State of the Art

AAA protocols are classically implemented in the application layer e.g. Radius [3] is an application over UDP and Diameter [4] is an application over TCP. Likely, our AAA protocol is running at the application layer over UDP. You will find an overview of this AAA authentication protocol in the section 1.1.

In order to build a model for it, we were conducted to have a closer look to the lower layers operations, especially those of the MAC layer, because the transmitted messages can be subject to collisions or to broken routes. This is addressed by the IEEE 802.11 standard [5] that provides for retransmissions by means of the Distributed Coordination Function (DCF). We were interested in the basic access mechanism which is one of the two DCF technics. It is briefly summarized in the section 1.2.

1.1 Authentication Protocol within a MANET Distributed AAA infrastructure

A centralized AAA infrastructure is traditionally composed of a AAA server, a AAA client located in a Network Access Server, and a client (a subscriber) who authenticates itself to the AAA server via the AAA client before accessing to the operator's network. To distribute this architecture and make access control possible in ad-hoc networks, we replaced the single server by a group of AAA servers and we placed the AAA client directly into the client (subscriber) device. As such, an ad-hoc node (any node from the ad-hoc network) is either a AAA server or a AAA client. AAA clients and AAA servers form the distributed AAA framework.

An authentication protocol takes place between a AAA client, e.g. a Joining Node JN, and the group of AAA servers. Both parties authenticate themselves using RSA asymmetric cryptography. During the authentication phase, the JN connects to the AAA servers. Actually, by means of threshold cryptography ([1, 6, 7]), it requests authentication to at least a threshold number of them. The ad-hoc network may bootstrap this number by following a procedure defined by the ad-hoc network-exploiting operator. Its value can later change according to the network evolution. For the sake of simplicity, we take the threshold number equal to the number of AAA servers in this report.

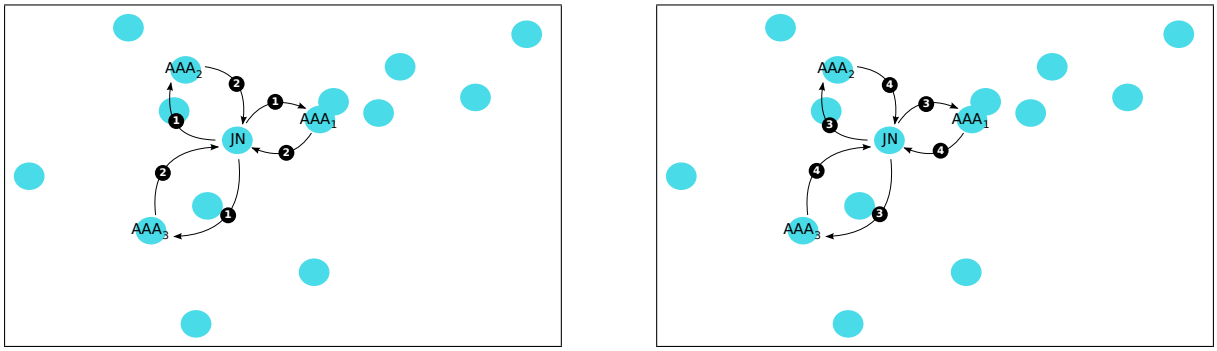

Figure 1.1: Four-way authentication protocol

Figure 1.1 shows the execution of the authentication protocol between a JN and three AAA servers:
(1) JN sends to each server a request for authentication that includes its identity (present in its public key certificate), MSG1: $\{ID_{JN}\}$.
(2) The servers respond with a challenge in the form of a random number [1], MSG2: $\{R_{AAA}\}$.
(3) JN generates a random number R_{JN}. Then it signs, using its RSA private key, both random numbers in addition to the identity of the group of AAA servers (ID_{AAA}). Next, it answers each server sending this signature accompanied by its public key certificate, its random number, and the identity of the AAA service, MSG3: $\{cert_{JN}, R_{JN}, ID_{AAA}, Sign_{JN}(R_{JN}, R_{AAA}, ID_{AAA})\}$.
(4) If the servers succeed to decipher JN's signature and to establish the integrity, each one of them computes a signature piece [7] using its RSA key-share [6] on both random numbers and on the identity of JN (this is one of the threshold cryptography aspects). They also generate an access token T_{JN} for the JN that is sent with the signature pieces accompanied by the public key certificate of the AAA service and the identity of the JN, MSG4: $\{cert_{AAA}, ID_{JN}, Sign_{AAA}(R_{AAA}, R_{JN}, ID_{JN}), T_{JN}\}$
These steps are inspired from the ISO-three way protocol (ISO [9798-3] [8]) that we adapted to our distributed context.

Once the JN successfully validates the integrity of the servers signature pieces (by combining them first [7]), the mutual authentication between the JN and the servers is considered as successful. JN is henceforward authorized to access the network. The legitimacy of its future traffic will be controlled by the relaying nodes before they will route it. This control will be done by checking the validity of JN's access token.

So far, authentication and authorization have been addressed in this framework. The accounting function is not yet supported, but, as a hot topic, it will be addressed in future works.

1.2 Basic Access Mechanism of the Distributed Coordination Function

As standardized by the 802.11 protocol [5], Distributed Coordination Function (DCF), a fundamental mechanism to access the medium, is based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol. It describes two technics to employ for packet transmission. The basic access mechanism is the default technic and it consists in a two way handshaking mechanism. The Request-to-send/Clear-to-send (RTS/CTS) mechanism is the second technic and it consists in a four-way protocol. In this report, we employed the basic access mechanism because it is more appropriate to our context since authentication packets length does not exceed 3000 bytes. RTS/CTS mechanism is rather applicable for longer packets.

Before transmitting a new packet, a node monitors the channel activity. If the channel is idle for a period of time equal to a Distributed Inter-Frame Space (DIFS), the node transmits. Otherwise, if the channel is sensed busy (either immediately or during the DIFS), the node persists to monitor the channel until it becomes idle for a DIFS. At this point, the node generates a random backoff time during which it waits before transmitting (this is the Collision Avoidance feature of the protocol), to minimize the probability of collision with packets being transmitted by the other nodes. In addition, to avoid channel capture, a node must wait a random backoff time between two consecutive new packet transmissions, even if the medium is sensed idle in the DIFS time.

For efficiency reasons, DCF employs a discrete-time backoff scale. The time immediately following an idle DIFS is slotted, and a node is allowed to transmit only at the beginning of each slot time. The slot time size, θ, depends on the physical layer and is set equal to the time needed at any node to detect the transmission of a packet from any other node.

Table 1.1: Contention window values for the three PHY specified by the 802.11 standard: FHSS (Frequency-Hopping Spread Spectrum), DSSS (Direct Sequence Spread Spectrum), and IR (Infra-Red)

PHY	Slot Time (θ)	CW_{min}	CW_{max}
FHSS	50 μs	16	1024
DSSS	20 μs	32	1024
IR	8 μs	64	1024

DCF adopts binary exponential backoff rules. Before each packet transmission, a random number r is uniformly chosen in the range (0, $cw - 1$), and the node waits for the backoff time $r\theta$. The value cw is called the contention window, and depends on the number of transmissions failed for the packet. At the first transmission attempt, cw is set

equal to the minimum contention window, CW_{min}. After each unsuccessful transmission, cw is doubled up to a maximum value $CWmax = 2^m CW_{min}$. The values CW_{min} and CW_{max} are PHY-specific and are summarized in table 1.1.

The backoff time counter is decremented as long as the channel is sensed idle, "frozen" when a transmission is detected on the channel, and reactivated when the channel is sensed idle again for more than a DIFS. The node transmits when the backoff time reaches zero.

The CSMA/CA mechanism does not rely on the capability of the nodes to detect a collision by hearing their own transmission. That is why an ACK is transmitted by the destination node to notify the successful packet reception. The ACK is immediately sent at the end of the packet reception and after a period of time called Short Inter-Frame Space (SIFS). As the SIFS is shorter than a DIFS, no other node will detect the channel idle until the end of the ACK. If the source node does not receive the ACK within a specified ACK_Timeout, or if it detects the transmission of a different packet on the channel, it reschedules the packet transmission [9].

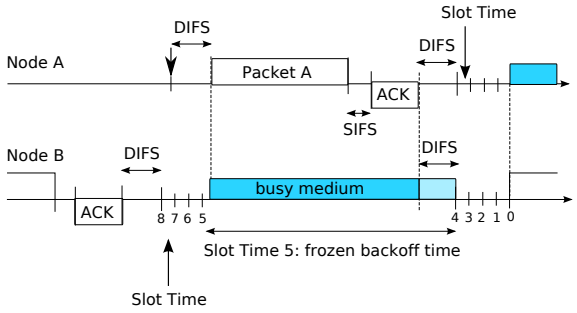

Figure 1.2: DCF basic access mechanism: example

Figure 1.2 illustrates the operations of the basic access mechanism. Two nodes A and B share the same wireless channel. At the end of the packet transmission, node B waits for a DIFS and then chooses a backoff time equal to 8, before transmitting the next packet. We assume that the first packet of node A arrives at the time indicated with an arrow in the figure. After a DIFS, the packet is transmitted. Note that the transmission of packet A occurs in the middle of the Slot Time corresponding to a backoff value, for node B, equal to 5. As a consequence of the channel sensed busy, the backoff time is frozen to its value 5, and the backoff counter decrements again only when the channel is sensed idle for a DIFS [9].

2 Protocol Modelization for a Theoretical Evaluation of the Authentication Overhead

The present chapter outlines the reasoning for building a model and computing the overhead of the authentication protocol exposed in section 1.1. It starts by analyzing the events sequence at the nodes from the construction of the first message MSG1 by the JN until its reception by one of the AAA servers, call it AAA_j (cf. Figure 2.1). Once the overhead of MSG1 with one single server is known, the reasoning simply applies to the other three messages of the protocol, MSG2, MSG3, and MSG4, and for the remaining servers, AAA_1, AAA_2, ..., AAA_n if n is the number of servers.

Our reasoning is based on some schemes that follow the following guidelines:

1. In a static network, the routes between the nodes are considered as fixed. We will represent them by straight segments.

2. Along a given route there maybe some relay nodes. The transmission delay between two successive nodes is due to hardware and software treatment but, according to the high velocity of electromagnetic waves (3 μs/km), is quite independent from the distance between them. Hence, nodes will be represented equally spaced along the segment. The segment will appear cut into shorter segments that define the hops. The length of a route is simply the number of hops it involves.

3. Routes between the JN and the AAA servers maybe be of different lengths . As a matter of fact, we could simply consider that all the routes have the same length equal to the length of the longest one. This will give an upper bound of the delays, which is sufficient for our purpose.

4. Finally, the scheme abstracting the connections between the JN and the AAA servers appears as an equal-length star-shaped graph, which is not very accurate comparing to the real geometry of the net but sufficient for the correctness of the reasoning.

In this work, computations were made by mean of the MAPLE Computer Algebra

System (CAS). We do not report all the stages of this computations but only a few significant results.

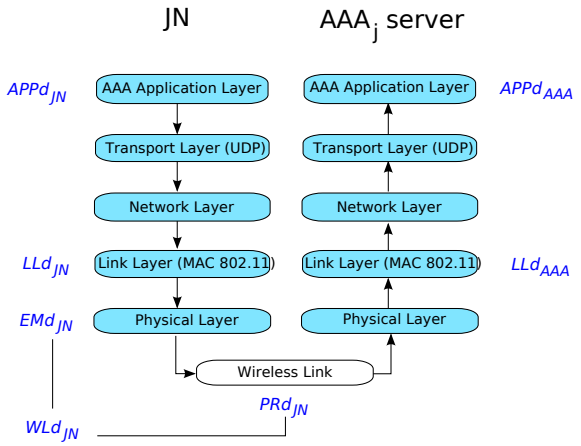

Figure 2.1: Events sequence of the first message MSG1

Figure 2.1 illustrates the events sequence when the JN and the server are one-hop away from each other. At the JN:

1. AAA Application Layer generates the first message containing the identity of the JN. The delay is $APPd_{JN}$ (Application delay).

2. After going through the Transport and the Network Layers, which takes a negligible time, the resulting packet enters the buffer of the Link Layer. The time spent in the Link Layer is LLd_{JN} (Link Layer delay).

3. During transmission on the Wireless Link, the packet might be subject to collisions or broken routes. Transmission delay (WLd_{JN}) has to take into account the possible packet retransmissions [5], as well as the emission delay (EMd_{JN}) and the propagation delay PRd_{JN}. Considering the speed of electromagnetic waves in the air, the latter is actually insignificant (about 3 μs/km)

At the AAA_j server:

1. Packets coming from the JN are placed in a buffer of the Link Layer. A packet is treated after LLd_{AAA} time.

2. After going through the Network and the Transport Layers, the packet is processed by the AAA Application Layer during $APPd_{AAA}$.

Thereby, the delay d_{1_j} for the first packet generation, transmission to AAA_j and processing is:

$$d_{1_j} = (APPd_{JN} + LLd_{JN} + WLd_{JN}) + (LLd_{AAA} + APPd_{AAA})$$

2.1 Model Features

From now, we suppose that computing operations within the nodes (so within JN and AAA_j) is fast enough to neglect the delays $APPd_{JN}$ and $APPd_{AAA}$. We also suppose that there is practically no other packets, except the authentication packets, in the Link Layers of the nodes, so LLd_{JN} and LLd_{AAA} are negligible, too. Thereby:

$$d_{1_j} = WLd_{JN}$$

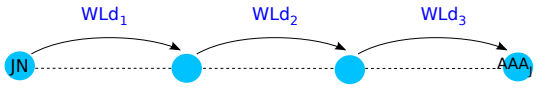

Figure 2.2: Delay on a 3-hop link

Besides, let us consider the case where the JN and AAA_j are not necessarily at one hop from each other but at a number of hops $hops = 3$ (cf. Figure 2.2); and suppose that the routes are fixed once and for all between all the nodes, so between JN and AAA_j (we shall relax this hypothesis in section 3.3). Hence the relaying nodes do routing operations in a fixed time supposed equal to zero. Thus:

$$WLd_{JN} = WLd_1 + WLd_2 + WLd_3$$

So, if *hops* is any number of hops:

$$WLd_{JN} = WLd_1 + WLd_2 + ... + WLd_{hops}$$

and:

$$d_{1_j} = \sum_{k=1}^{hops} WLd_k \tag{2.1}$$

The delay WLd_k is a positive random variable having as a distribution function F_{WLd_k}. It takes into consideration the prospective retransmissions of a packet as described by the DCF basic access mechanism. The maximum number of retransmissions is equal to seven as defined in the IEEE 802.11 specifications [5]. If p is the probability of retransmission for a packet in the wireless channel, and X the number of retransmissions (X is a discrete random variable that covers the values of the set $\{1..7\}$), then:

$$\begin{cases} P(X = i) = p^i(1-p) \ for \ 0 \leq i \leq 6 \\ P(X = 7) = p^7 \\ P(X = i) = 0 \ for \ i \geq 8 \end{cases}$$

Table 2.1: Probability computing using the binary exponential backoff rules

Event	Probability	Event / Condition	Conditional Probability
$X = 0$	$P(X = 0) = 1 - p$	$\{WLd_k \leq t\}$ $\{X = 0\}$	$P(\{WLd_k \leq t\}\|\{X = 0\}) = DIFS$ $+ P(\{EMd \leq t\}) + SIFS + EMd_{ACK}$
$X = 1$	$P(X = 0) = p(1 - p)$	$\{WLd_k \leq t\}$ $\{X = 1\}$	$P(\{WLd_k \leq t\}\|\{X = 1\}) = DIFS$ $+ 2\ P(\{EMd \leq t\}) + \frac{1}{2}CW_{min}\theta$ $+ DIFS + ACK\_Timeout + SIFS + EMd_{ACK}$
$X = 2$	$P(X = 0) = p^2(1 - p)$	$\{WLd_k \leq t\}$ {X=2}	$P(\{WLd_k \leq t\}\|\{X = 2\}) = 3 \cdot DIFS$ $+ 3\ P(\{EMd \leq t\}) + \frac{1}{2}\sum_{j=0}^{1} 2^j \cdot CW_{min}\theta$ $+ 2\ ACK\_Timeout + SIFS + EMd_{ACK}$
$X = 3$	$P(X = 0) = p^3(1 - p)$	$\{WLd_k\ \leq t\}$ $\{X = 3\}$	$P(\{WLd_k \leq t\}\|\{X = 3\}) = 4 \cdot DIFS$ $+ 4\ P(\{EMd \leq t\}) + \frac{1}{2}\sum_{j=0}^{2} 2^j \cdot CW_{min}\theta$ $+ 3\ ACK\_Timeout + SIFS + EMd_{ACK}$
$X = 4$	$P(X = 0) = p^4(1 - p)$	$\{WLd_k \leq t\}$ $\{X = 4\}$	$P(\{WLd_k \leq t\}\|\{X = 4\}) = 5 \cdot DIFS$ $+ 5\ P(\{EMd \leq t\}) + \frac{1}{2}\sum_{j=0}^{3} 2^j \cdot CW_{min}\theta$ $+ 4\ ACK\_Timeout + SIFS + EMd_{ACK}$
$X = 5$	$P(X = 0) = p^5(1 - p)$	$\{WLd_k \leq t\}$ $\{X = 5\}$	$P(\{WLd_k \leq t\}\|\{X = 5\}) = 6 \cdot DIFS$ $+ 6\ P(\{EMd \leq t\}) + \frac{1}{2}\sum_{j=0}^{4} 2^j \cdot CW_{min}\theta$ $+ 5\ ACK\_Timeout + SIFS + EMd_{ACK}$
$X = 6$	$P(X = 0) = p^6(1 - p)$	$\{WLd_k \leq t\}$ {X=6}	$P(\{WLd_k \leq t\}\|\{X = 6\}) = 7 \cdot DIFS$ $+ 7\ P(\{EMd \leq t\}) + \frac{1}{2}\sum_{j=0}^{5} 2^j \cdot CW_{min}\theta$ $+ 6\ ACK\_Timeout + SIFS + EMd_{ACK}$
$X = 7$	$P(X = 0) = p^7(1 - p)$	$\{WLd_k \leq t\}$ $\{X = 7\}$	$P(\{WLd_k \leq t\}\|\{X = 7\}) = 8 \cdot DIFS$ $+ 8\ P(\{EMd \leq t\}) + \frac{1}{2}\sum_{j=0}^{6} 2^j \cdot CW_{min}\theta$ $+ 7\ ACK\_Timeout + SIFS + EMd_{ACK}$

The Total Probability Law [10] allows to write the following formula:

$$F_{WLd_k}(t) = P(WLd_k \leq t) = \sum_{i=0}^{7} P(\{WLd_k \leq t\}|\{X = i\}) \cdot P(\{X = i\}) \tag{2.2}$$

Given the binary exponential backoff rules (cf. section 1.2) and as it is exposed in table 2.1:

$$\begin{aligned} P(\{WLd_k \leq t\}|\{X = i\}) &= (i+1)DIFS + (i+1) \cdot P(\{EMd \leq t\}) + \frac{1}{2}\sum_{j=0}^{i-1} 2^j \cdot CW_{min}\theta \\ &\quad + i\ ACK\_Timeout + SIFS + EMd_{ACK} \end{aligned}$$

where $DIFS$, θ, $SIFS$, and $ACK\_Timeout$ are DCF timers, CW_{min} is the minimum contention window, EMd_{ACK} is the emission time of an ACK at the byte-rate of 1 Mbps [5], and EMd the emission time of a message of l bytes at the byte-rate λ.

If we suppose that the emission time necessary to deliver one byte is a positive continuous random variable, following an exponential distribution with parameter λ (the average byte-rate), then the necessary mean time to deliver l bytes is l/λ. Since l indicates the length of an authentication message, l is large enough (cf. table 2.2) to apply the Central Limit Theorem [11]. Thus, the emission time of l bytes is a positive continuous random variable, EMd, following a gaussian distribution of mean l/λ and variance l/λ^2 [11].

Hence, $\forall i \in [0, 7]$, WLd_k, conditionally in {X=i}, is a positive random variable following a gaussian distribution of mean μ_i and variance σ_i^2 where:

$$\mu_i = (i+1)DIFS + (i+1)\frac{l}{\lambda} + \frac{1}{2}\sum_{j=0}^{i-1} 2^j \cdot CW_{min}\theta + i\ ACK\_Timeout + SIFS + EMd_{ACK}$$

and

$$\sigma_i^2 = (i+1) \cdot \frac{l}{\lambda^2}$$

Consequently, using the classical erf function [12]:

$$F_{WLd_k}(t) = \frac{1}{2} + \frac{1}{2}\sum_{i=0}^{6} p^i(1-p) \cdot \mathrm{erf}\left(\frac{1}{2}\frac{\sqrt{2}(t-\mu_i)}{\sigma_i}\right) + \frac{1}{2}p^7 \cdot \mathrm{erf}\left(\frac{1}{2}\frac{\sqrt{2}(t-\mu_7)}{\sigma_7}\right) \tag{2.3}$$

with a mean:

$$\mu_{WLd} = \sum_{i=0}^{6} p^i(1-p)\mu_i + p^7\mu_7$$

and a variance:

$$\begin{aligned}\sigma^2_{WLd} = {} & (-\mu_6^2 + 2\mu_7\mu_6 - \mu_7^2)p^{14} + (-2\mu_7\mu_6 + 2\mu_6^2 - 2\mu_6\mu_5 + 2\mu_7\mu_5)p^{13} \\ & + (-2\mu_6\mu_4 - 2\mu_7\mu_5 + 4\mu_6\mu_5 - \mu_6^2\mu_5^2 + 2\mu_7\mu_4)p^12 + (-2\mu_7\mu_4 \\ & + 2\mu_5^2 - 2\mu_6\mu_3 - 2\mu_6\mu_5 - 2\mu_5\mu_4 + 4\mu_6\mu_4 + 2\mu_7\mu_3)p^11 + (-2\mu_7 \\ & \mu_3 + 2\mu_7\mu_2 - 2\mu_6\mu_2 + 4\mu_6\mu_3 - \mu_5^2 - 2\mu_6\mu_4 + 4\mu_5\mu_4 - \mu_4^2 \\ & - 2\mu_5\mu_3)p^10 + (2\mu_7\mu_1 - 2\mu_4\mu_3 - 2\mu_7\mu_2 + 2\mu_4^2 - 2\mu_6\mu_3 + 4\mu_6 \\ & \mu_2 + 4\mu_5\mu_3 - 2\mu_5\mu_2 - 2\mu_5\mu_4 - 2\mu_6\mu_1)p^9 + (4\mu_4\mu_3 - \mu_3^2 + 4 \\ & \mu_6\mu_1\mu_4^2 + 4\mu_5\mu_2 - 2\mu_6\mu_0 + 2\mu_7\mu_0 - 2\mu_5\mu_3 - 2\mu_5\mu_1 \\ & - 2\mu_4\mu_2 - 2\mu_6\mu_2 - 2\mu_7\mu_1)p^8 + (4\mu_6\mu_0\sigma_6^2 - 2\mu_6\mu_1 - 2\mu_7 \\ & \mu_0 - \mu_6^2 + \mu_7^2 - 2\mu_4\mu_3 + 4\mu_5\mu_1 - 2\mu_5\mu_0 + 4\mu_4\mu_2 - 2\mu_5\mu_2 \\ & - 2\mu_3\mu_2 + \sigma_7^2 + 2\mu_3^2 - 2\mu_4\mu_1)p^7 + (\mu_6^2 - 2\mu_3\mu_1 - \mu_5^2 + 4\mu_5 \\ & \mu_0 + \sigma_6^2 - \mu_3^2 - \mu_2^2 - \sigma_5^2 - 2\mu_6\mu_0 - 2\mu_4\mu_2 + 4\mu_3\mu_2 + 4\mu_4 \\ & \mu_1 - 2\mu_5\mu_1 - 2\mu_4\mu_0)p^6 + (\sigma_5^2 + 4\mu_4\mu_0 - 2\mu_3\mu_0 - 2\mu_5\mu_0 \\ & - 2\mu_2\mu_1 + \mu_5^2 + 4\mu_3\mu_1 - 2\mu_3\mu_2 - 2\mu_4\mu_1 + 2\mu_2^2 - \mu_4^2 - \sigma_4^2)p^5 \\ & + (-2\mu_3\mu_1 - \mu_1^2 - \mu_2^2 - \mu_3^2 + \sigma_4^2 + \mu_4^2 - 2\mu_4\mu_0 + 4\mu_2\mu_1 - \sigma_3^2 \\ & - 2\mu_2\mu_0 + 4\mu_3\mu_0)p^4 + (-2\mu_3\mu_0 + 4\mu_2\mu_0 + \sigma_3^2 - 2\mu_1\mu_0 + 2\mu_1^2 \\ & - 2\mu_2\mu_1\mu_2^2 + \mu_3^2 - \sigma_2^2)p^3 + (-\sigma_1^2 - 2\mu_1^2 + \sigma_2^2 + \mu_2^2 + 4\mu_1 \\ & \mu_0 - 2\mu_2\mu_0 - \mu_0^2)p^2 + (\sigma_1^2 + \mu_1^2 - 2\mu_1\mu_0 + \mu_0^2\sigma_0^2)p + \sigma_0^2\end{aligned}$$

μ_{WLd_1} and $\sigma^2_{WLd_1}$ are independent from k because all the wireless links are assumed identical.

Accordingly, the positive random variables $\{WLd_k\}_{1\leq k\leq hops}$ follow the same probability law. Since each transmission of a packet on a specific hop is independent from the transmission of the same packet on another hop, these random variables are independent and the Central Limit Theorem [11] applies again. Hence, given the equality (2.1), d_{1_j} follows a gaussian distribution of mean:

$$\mu_{d_{1_j}} = hops \cdot \mu_{WLd_1}$$

and variance:

$$\sigma^2_{d_{1_j}} = hops \cdot \sigma^2_{WLd_1}$$

Now, if d_{2_j}, d_{3_j} and d_{4_j} are respectively the delays for the second, the third and the fourth message of the authentication protocol, d_{2_j}, d_{3_j} and d_{4_j} have similar distribution

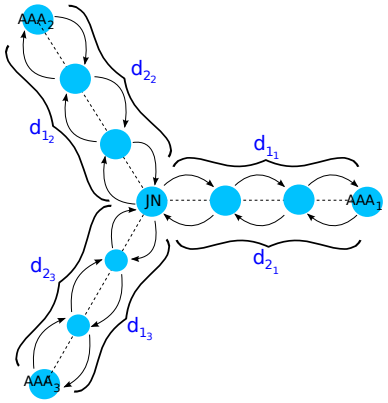

Figure 2.3: First round-trip of the authentication protocol ($n = 3$, $hops = 3$)

functions as d_{1_j} (only the length of their corresponding messages differs). Let $d_{12_j} = d_{1_j} + d_{2_j}$ and $d_{34_j} = d_{3_j} + d_{4_j}$ be the delay for respectively the first and the second round-trip of the protocol through the server AAA_j, and n the number of AAA servers. The delays $\{d_{12_j}\}_{1 \leq j \leq n}$ (respectively $\{d_{34_j}\}_{1 \leq j \leq n}$) are different for each server (because the transmissions on the links between the JN and the AAA servers can not be exactly the same for each link and at any moment), however they follow the same probability law i.e. they have the same gaussian distribution function $F_{d_{12}}$ of mean $\mu_{12} = \mu_{d_{1_j}} + \mu_{d_{2_j}}$ (respectively) and variance $\sigma_{12}^2 = \sigma_{d_{1_j}}^2 + \sigma_{d_{2_j}}^2$:

$$F_{d_{12}}(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}\sigma_{12}} \mathrm{e}^{\frac{-(\mathrm{t}-\mu_{12})^2}{2\sigma_{12}^2}} \mathrm{dt} \tag{2.4}$$

Also, suppose that the number of hops between the JN and each server is the same i.e. equal to *hops* (cf. the example of Figure 2.3 when $hops = 3$ and $n = 3$). During the first round-trip and for threshold cryptography reasons, JN has to wait for all the servers responses before triggering the second round-trip with all the servers. The total delay $D_{12_{max}}$ to achieve the first round-trip with all the servers is the maximum of the $\{d_{12_j}\}_{1 \leq j \leq n}$:

$$D_{12_{max}} = max\{d_{12_j}\}_{1 \leq j \leq n} \tag{2.5}$$

As $\{d_{12_j}\}_{1 \leq j \leq n}$ follow the same probability law, the distribution function of $D_{12_{max}}$ is:

$$F_{D_{12_{max}}} = (F_{d_{12}})^n \tag{2.6}$$

If $D_{34_{max}}$indicates the total delay to achieve the second exchange with all the servers and D the total delay for an achieved authentication, then:

$$D = D_{12_{max}} + D_{34_{max}} \tag{2.7}$$

Since the expected value (or mean [11]) is a linear operator:

$$E(D) = E(D_{12_{max}}) + E(D_{34_{max}}) \tag{2.8}$$

where E(D) is the expected value of the total delay i.e. the authentication protocol overhead. Its expression and its profile are given in the next section.

Summary of our assumptions:

1. threshold number equal to servers number n
2. authentication method uses RSA cryptography
3. computing time within the nodes is negligible
4. Link Layer treating time is negligible
5. same number of hops between the JN and each AAA server
6. routes are already established
7. retransmission probability (p) equal 0.1

2.2 Overhead Evaluation

The overhead $E(D)$ depends on the length of the four messages, the byte-rate, the spread spectrum technic employed by the Physical layer, the probability of retransmissions, the number of hops, and the number of servers. Their values are summarized in table 2.2 except for the number of servers and the number of hops that we varied: $n \in \{1, ..., 6\}$ and $hops \in \{1, ..., 10\}$. The spread technic we employed is DSSS. The length of messages were indicated according to their content (cf. section 1.1) and following the example given in [13].

Taking into account the parameters of table 2.2, (2.8) gives:

$$\begin{aligned} E(D) \quad = \quad & \int_{-\infty}^{+\infty} 866.7284917\ t\ (-0.5000000002\ \mathrm{erf}(\frac{0.6144945009\ 10^{-8}(-0.25\ 10^{12}\ t + 0.287777519\ 10^{9}\ hops)}{\sqrt{hops}}) \\ & + 0.5000000002)^{(n-1)}\ n\mathrm{e}^{(-\frac{0.3776034916\ 10^{-16}(-0.25\ 10^{12}\mathrm{t}+0.287777519\ 10^{9}\mathrm{hops})^2}{\mathrm{hops}})}/\sqrt{\mathrm{hops}}\ \mathrm{dt} \\ & + \int_{-\infty}^{+\infty} 385.8159001\ t\ (-0.5000000000\ \mathrm{erf}(\frac{0.3419204391\ 10^{-8}(-0.2\ 10^{12}\ t + 0.747232111\ 10^{9}\ hops)}{\sqrt{hops}}) \\ & + 0.5000000000)^{(n-1)}\ n\mathrm{e}^{(-\frac{0.1169095867\ 10^{-16}(-0.2\ 10^{12}\mathrm{t}+0.747232111\ 10^{9}\mathrm{hops})^2}{\mathrm{hops}})}/\sqrt{\mathrm{hops}}\ \mathrm{dt} \end{aligned}$$

Table 2.2: Parameter values used in the model

Parameter	Value
1st message length ($l1$)	287 bytes
2nd message length ($l2$)	32 bytes
3rd message length ($l3$)	1593 bytes
4th message length ($l4$)	1925 bytes
byte-rate (λ)	11 Mbps
SIFS	10 μs
DIFS	50 μs
SlotTime (θ)	20 μs
CWmin	32
ACK_Timeout	334 μs
ACK message length	304 bits
ACK emission time (EMd_{ACK})	304 μs
retransmission probability (p)	0.1

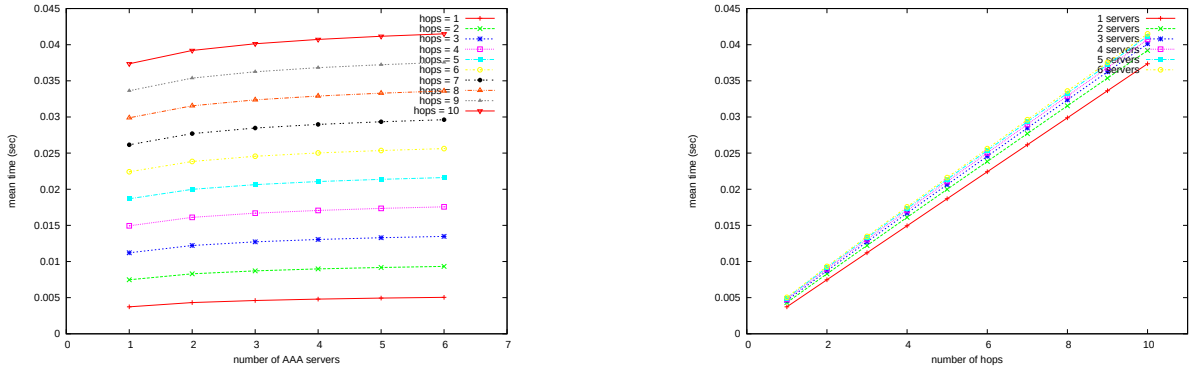

Figure 2.4: Max model. On the left-hand side: overhead vs. #AAA servers. On the right-hand side: overhead vs. #hops

Figure 2.4 depicts the evolution of the overhead $E(D)$ when the number of AAA servers and the number of hops increase. As expected, $E(D)$ increases when n rises and when *hops* rises. The form of the curves when n increases is not exactly a line segment whereas that of the curves when *hops* increases is roughly a line segment. The values range is between about 0.003 sec for $n = 1$ and $hops = 1$ and 0.04 sec for $n = 6$ and $hops = 10$, which is largely acceptable from the QoS point of view. However, it still necessary to confront these results to the simulations. This is treated in the next chapter.

3 Protocol Simulation for a Practical Evaluation of the Authentication Overhead

Simulations were conducted on NS-2.33 [14]. In a nutshell, NS-2 is a discrete event network simulator that provides several configuration possibilities for wireless scenarios. It can correctly simulate a hundred wireless nodes in one run. The core of NS-2 is written in C++. OTCL scripts are used to specify scenarios.

We mainly wrote two OTCL scripts for our simulations. In the first one, we assumed that the routes were already established as for the model of the chapter 2 and that the nodes don't move. This first scenario allowed us to validate our model. In the second one, nodes were set into motion to evaluate the impact of the movement on the overhead.

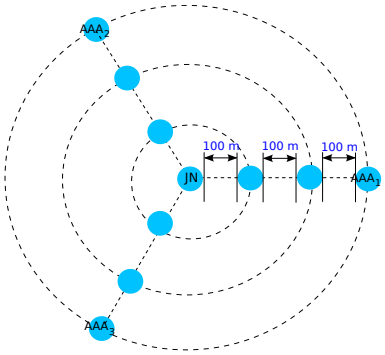

Figure 3.1: Nodes placement in simulations flat-grid ($n = 3$, $hops = 3$)

We also wrote a C program to automatically place the nodes at a given number of hops and for a given number of servers (cf. Figure 3.1). Nodes are placed on concentric circles of the same center: the joining node JN. Servers are on the outermost circle of radius $100 \cdot hops$ meters. They are placed in such a way that angles are equal between them. Relaying nodes are at the intersection of the lines joining the JN to the servers with the circles of radius $r \in \{100, ..., 100 \cdot (hops - 1), 100 \cdot hops\}$.

Moreover, we wrote an OTCL script to generate automatically the authentication traffic between the JN and the AAA servers. The authentication traffic is UDP traffic. Lengths of UDP packets are those defined in table 2.2. Finally, a shell script ran each

scenario for different values of n and *hops*.

The following section gives some important NS commands that were used in OTCL scripts. It also specifies some relevant parameters for the simulations.

3.1 Simulation Parameters

Table 3.1 summarizes the simulation parameters. To be sure that the obtained results are not influenced by the simulation duration or length of the queue, their values were chosen sufficiently large: simulation duration: 600 sec, length of the queue: 1000 packets.

Table 3.1: Simulation parameters

Parameters	**NS command**	**Value**
Ad-hoc routing protocol	$ns_ node-config -adhocRouting	AODV
MAC protocol	$ns_ node-config -macType	Mac/802_11
Queue type	$ns_ node-config -ifqType	Queue/DropTail/PriQueue
Length of the queue	$ns_ node-config -ifqLen	1000 packets
Antenna type	$ns_ node-config -antType	Antenna/OmniAntenna
Propagation model	$ns_ node-config -propType	Propagation/FreeSpace [14]
Node range	Phy/WirelessPhy set RXThresh_ 8.5457e-09	150 meter
DCF technic	Mac/802_11 set RTSThreshold_ 3000	basic access technic
PHY spread-spectrum	(default NS model)	DSSS
Byte-rate	Mac/802_11 set dataRate_	11Mbps
Node coordinations	set X_, set Y_	(depends on n and *hops*)
A node (nd) speed (v) towards a position (x,y)	$nd setdest x y v	(depends on n and *hops*)
Simulations duration (d)	$ns_ at d "$ns_ halt"	600 sec

3.2 Motionless Nodes Scenario

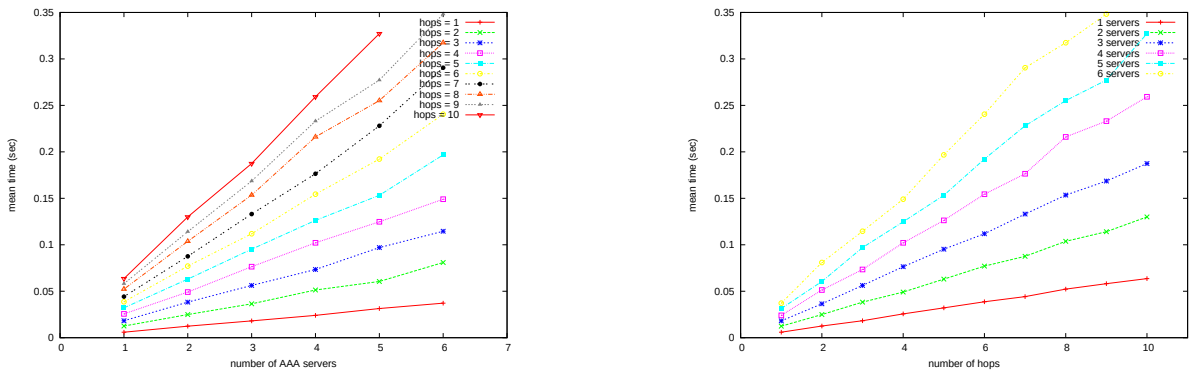

Figure 3.2: Motionless simulations. On the left-hand side: overhead vs. #AAA servers. On the right-hand side: overhead vs. #hops

This scenario utilizes fixed routes during the whole simulation time. A route from the JN to a AAA server is composed of the nodes at the intersections between the line joining

the JN to the server and the concentric circles (cf. Figure 3.1). We ran NS for each couple (n, $hops$) and traced the results showed in Figure 3.2.

Figure 3.3: On the left-hand side: actions execution in the real case. On the right-hand side: actions execution in NS simulations

The resulting curves are increasing like for the model (cf. chapter 2). However their shape and the values range are significantly different. This stems essentially from a main trait of NS that is "sequentiality". NS indeed can not simulate two actions that overlap [14]. Figure 3.3 illustrates that when the servers AAA_i and AAA_j respond to JN (during the first round-trip). AAA_i sends its response, MSG2, to the JN at t_1. The transmission ends at t_3. AAA_j sends its response MSG2 at t_2 ($t_1 \leq t_2 \leq t_3$). In the real case, this response is effectively sent at t_2. In NS simulations, it can not be sent until the end of MSG2 of AAA_i i.e at t_3.

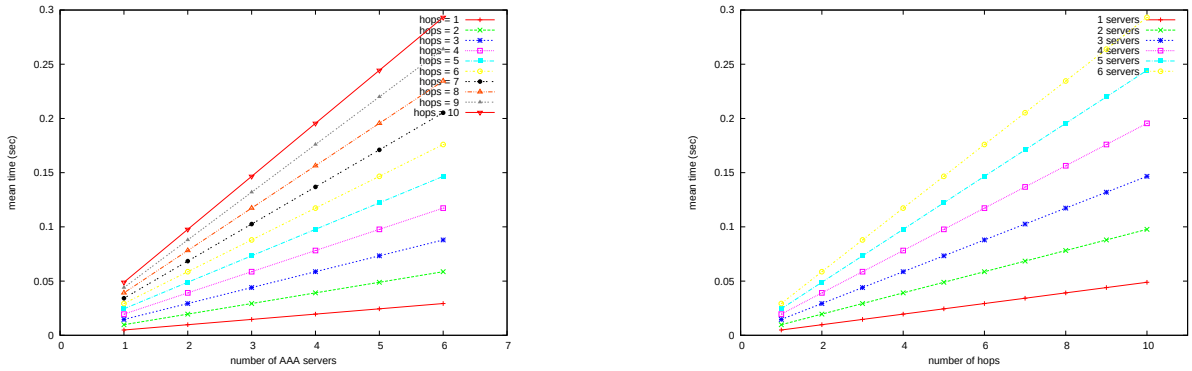

Figure 3.4: Sum model. On the left-hand side: overhead vs. #AAA servers. On the right-hand side: overhead vs. #hops

We modified the model to take the "sequentiality" feature into consideration and obtained the results of Figure 3.4. In the modified model, instead of computing the maximum of $\{d_{12_j}\}_{1 \leq j \leq n}$, we computed their sum. Hence the delay $D_{12_{sum}}$ to achieve the first round-trip with all the servers is:

$$D_{12_{sum}} = \sum_{j=1}^{n} d_{12_j}$$

As $\{d_{12_j}\}_{1 \leq j \leq n}$ follow the same probability law and that the expected value (or mean [11])

is a linear operator:

$$\begin{aligned} E(D_{12_{sum}}) &= E(\sum_{j=1}^{n} d_{1_j}) + E(\sum_{j=1}^{n} d_{2_j}) \\ &= n \cdot (hops \cdot \mu_{WLd_1}) + n \cdot (hops \cdot \mu_{WLd_2}) \\ &= n \cdot hops \cdot (\mu_{WLd_1} + \mu_{WLd_2}) \end{aligned}$$

Similarly the delay $D_{34_{sum}}$ to achieve the second round-trip with all the servers verifies:

$$\begin{aligned} E(D_{34_{sum}}) &= E(\sum_{j=1}^{n} d_{3_j}) + E(\sum_{j=1}^{n} d_{4_j}) \\ &= n \cdot (hops \cdot \mu_{WLd_3}) + n \cdot (hops \cdot \mu_{WLd_4}) \\ &= n \cdot hops \cdot (\mu_{WLd_3} + \mu_{WLd_4}) \end{aligned}$$

So the total delay for a successful authentication is:

$$\begin{aligned} E(D) &= E(D_{12_{sum}}) + E(D_{34_{sum}}) \\ &= n \cdot hops \cdot (\mu_{WLd_1} + \mu_{WLd_2} + \mu_{WLd_3} + \mu_{WLd_4}) \\ &= n\ hops\ (4\ p^3 ACK\_Timeout\ \lambda + 4\ p\ ACK\_Timeout\ \lambda\ + 2\ p\ CW_{min}\ \lambda \\ &\quad + 4\ p\ DIFS\ \lambda + 4\ p^5\ ACK\_Timeout\ \lambda + 4\ p^7 ACK\_Timeout\ \lambda \\ &\quad + 128\ p^7\ CW_{min}\ \lambda + 4\ p^4 ACK\_Timeout\ \lambda + 16\ p^4\ CWmin\ \lambda \\ &\quad + 4\ p^4\ DIFS\ \lambda + 32\ p^5\ CWmin\ \lambda + 4\ p^5\ DIFS\ \lambda \\ &\quad + 4\ p^2\ ACK\_Timeout\ \lambda + 4\ p^2\ CWmin\ \lambda + 4\ p^2\ DIFS\ \lambda \\ &\quad + 4\ p^6\ DIFS\ \lambda + 4\ p^6\ ACK\_Timeout\ \lambda + 64\ p^6 CWmin\ \lambda \\ &\quad + 8\ p^3\ CWmin\ \lambda + 4\ p^3\ DIFS\ \lambda + 4\ p^7\ DIFS\ \lambda + 4\ \lambda\ SIFS + l_4 \\ &\quad + l_2 + l_1 + l_3 + 4\ \lambda\ DIFS + 4\ \lambda\ Ed_{ACK} + p^3\ l_3 + p^6\ l_3 + p^4\ l_3 \\ &\quad + p^5\ l_4 + p^3\ l_4 + p^4\ l_4 + p^6\ l_4 + p^7\ l_4 + p^4\ l_2 + p\ l_2 + p^2\ l_3 \\ &\quad + p^5\ l_3 + p^7\ l_3 + p^6\ l_2 + p\ l_4 + p\ l_3 + p^2\ l_4 + p^5\ l_2 + p^7\ l_2 \\ &\quad + p^3\ l_2 + p^2\ l_1 + p^5\ l_1 + p^7\ l_1 + p^3\ l_1 + p^4\ l_1 + p^6\ l_1 + p\ l_1 \\ &\quad + p^2\ l_2)/\lambda \end{aligned}$$

The similarity between Figure 3.2 and Figure 3.4 is weighty. The range of values is slightly larger for the simulations. The largest difference of values is about 0.08 sec for $n = 6$ and $hops = 9$. This slight difference is due to the accumulated delays of transmissions and receptions of the packets between the layers of the nodes. In the model, we supposed that they were negligible. This is confirmed by the simulations with less than 0.08 sec.

In addition, since the simulation curves have almost the same shape as the "sum"

model curves, we deduce that this tiny difference is proportional to the number of hops and the number of servers: when the number of crossed nodes increases, the number of crossed layers rises, so that the number of transmissions and receptions between the layers rises, as well.

These findings are of great importance because they prove that our model is valid. They also prove that the authentication protocol is scalable for different numbers of servers and different numbers of hops. They remain valid for the "max" model where the maximum of the message delays through the servers was computed rather than their sum: the maximum is indeed at most equal to the sum.

Indeed, notice that the "max" model gives a better overhead than the "sum" model and its corresponding simulations. The overhead is almost nine times better for the "max" model than for the "sum" model.

Anyway, the second scenario simulations were also carried with NS to evaluate the impact of the nodes movement on the overhead.

3.3 Moving Nodes Scenario

This scenario simulations were also carried out with NS-2 to evaluate the impact of the nodes movement on the overhead. Initially routes are established likely to the previous section. At $t = 300sec$, nodes that are one-hop away from the JN, are put into motion in the trigonometric (anticlockwise) direction. At the same time, nodes that are two-hops away from the JN, are put into motion in the clockwise direction. Each node moves linearly at a speed of 5m/s towards the next node on the same circle as itself.

Trajectories were thought in a manner that nodes from the first and the second hop begin to move away from each others without disconnecting definitely from the network. At the end of the movement, they indeed place themselves at the previously defined intersections (cf. the beginning of chapter 3).

During the movement and when authentications are attempted, routes are recalculated by AODV [15]. Sometimes, some nodes find themselves beyond the scope of the other nodes and hence become unreachable. This can happen when $n \geq 3$ and $hops \geq 3$. For $n < 3$, all the nodes are situated on the same line whatever the value of $hops$ is, so moving nodes never become unreachable (when $n = 1$, there is one node at the first hop and one node at the second hop, so they do not move. When $n = 2$, there is a couple of nodes at the first hop and a couple of nodes at the second hop, all situated on the same line. Each couple nodes move towards each other, so remain within the scope of each other).

For $hops < 3$ and whatever the value of n is, the network graph never becomes disconnected. The cases $n \in \{1, 2\}$ were already proved in the previous paragraph, and the analysis of the distances between the nodes for $n = 3$ can prove that, too (see below). Once this is true for $n \in \{1, 2, 3\}$, it is true when $n > 3$ given that the angles between the

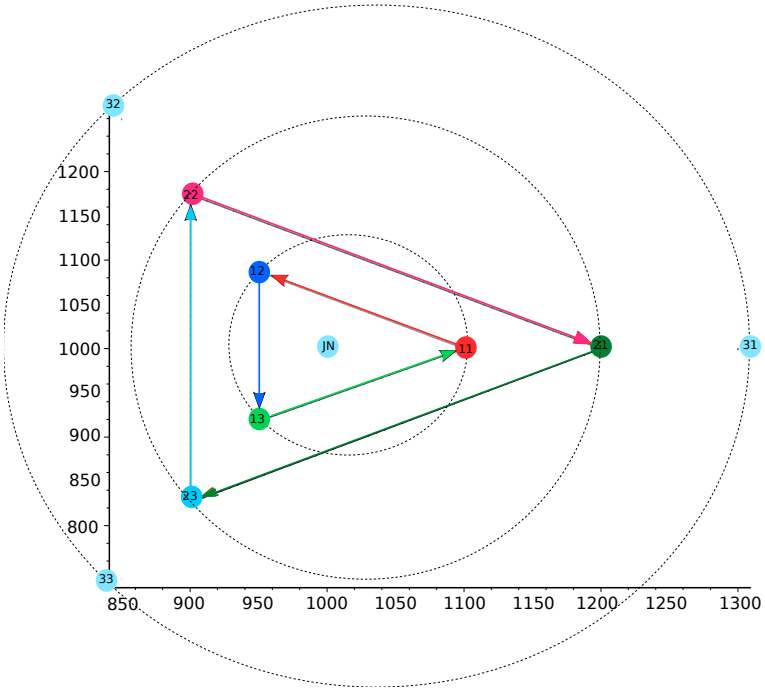

Figure 3.5: Trajectory of relaying nodes

lines joining the servers to the JN shrink as n increases.

Figure 3.5 depicts the case where $n = 3$ and $hops = 3$. Nodes 31, 32 and 33 designate the AAA servers. At the first hop, node 11 (respectively 12 and 13) moves towards node 12 (respectively 13 and 11). At the second hop, node 21 (respectively 22 and 23) moves towards node 23 (respectively 21 and 22). Beyond the second hop, nodes do not move.

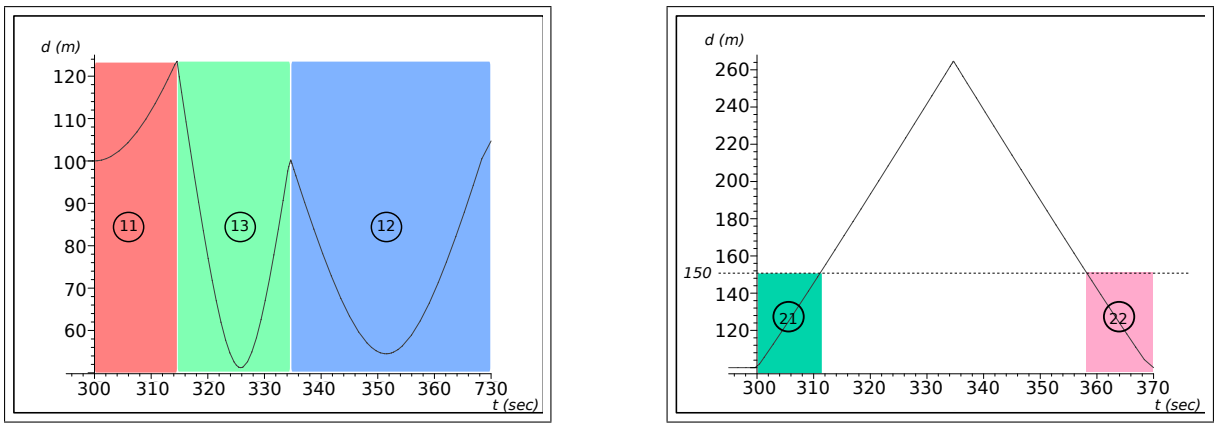

Figure 3.6: Influence of the relaying nodes movement on the network graph connectedness. On the left-hand side: minimum distance between node 21 and nodes 11, 12 and 13. On the right-hand side: minimum distance between node 31 and nodes 21 and 22

It is obvious that JN is never disconnected from nodes 11, 12 and 13 because they remain inside the circle of 100 meters radius (cf. table 3.1). Besides, the study of the distances between on one hand node 21 and on the other hand nodes 11, 12 and 13 demonstrates that 21 is never disconnected from the network, neither. As shown by

Figure 3.6, the minimum distance between node 21 and nodes 11, 12 and 13 is always less than 150 meters (i.e. nodes range). Between 300 sec and 314 sec, node 11 is the nearest node to 21. Between 315 sec and 334 sec, node 13 becomes the nearest node to 21. Finally, between 335 sec and 368 sec, node 12 becomes the nearest node to 21. Each time the nearest node to 21 changes, we observed that it is soon selected by AODV to become the relaying node between the JN and node 21. For symmetry reasons, this reasoning is valid for nodes 22 and 23.

Furthermore, the study of the distances between the node 31 and the nodes 21 and 22 demonstrates that 31 becomes disconnected from the network between 312 sec and 357 sec (cf. Figure 3.6). Message relaying from the JN to 31 is insured by 21 before 311 sec and by 22 after 358 sec. For symmetry reasons, the same reasoning applies for 32 and 33. Hence for $n = 3$ and $hops = 3$, authentication can not be successful from 312 sec to 357 sec because of the non connectedness of the network graph.

Note that the more n increases the shorter is the interval of disconnection and that disconnections occur for $hops \geq 3$.

3.4 Protocol Performance

For each couple (n,$hops$) where $n \in \{1..6\}$ and $hops \in \{1..10\}$ and at each second between the beginning and the end of the movement, an authentication was attempted. After that, we computed the success ratio τ defined as the number of successful authentications divided by the total number of authentications:

$$\tau = \frac{\#successful\ authentications}{\#attempted\ authentications}$$

The computed success ratio is equal to 0.69 i.e. 69% of the authentications were successful. The investigation of the trace files (produced by NS after each simulation) showed that unsuccessful authentications were often due to disconnectedness of the network graph, as explained in the previous section. In that case, authentication message is dropped by the source node and the error *NRTE: drop, no route is available* appears in the trace file as identified in the following example:

d -t 319.000000000 -Hs 0 -Hd -2 -Ni 0 -Nx 1000.00 -Ny 1000.00 -Nz 0.00 -Ne -1.000000 -Nl RTR -Nw **NRTE** -Ma 0 -Md 0 -Ms 0 -Mt 0 **-Is 0.2 -Id 3.0** -It udp -Il 347 -If 0 -Ii 0 -Iv 30 -Pn udp -Ps 3 -Pa 0 -Pf 0 -Po 0
where: *d* for drop, *-Is 0.* for the JN identifier, and *-Id j.* for the AAA_j identifier.

Unsuccessful authentications can be due to collisions as well. The authentication message

is dropped and the error *COL: drop, collision* appears in the trace file as illustrated here:

d -t 339.094004956 -Hs 7 -Hd 7 -Ni 7 -Nx 919.10 -Ny 1058.78 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw **COL** -Ma 13a -Md 7 -Ms 0 -Mt 800 **-Is 0.1 -Id 2.0** -It udp -Il 1711 -If 0 -Ii 0 -Iv 30 -Pn udp -Ps 4 -Pa 0 -Pf 0 -Po 0

When the number of retransmissions reaches seven (cf. section 1.2) for an authentication message, the latter is dropped according to the IEEE 802.11 802.11 standard [5]. The error that appears in the trace file is *RET: drop, retry count exceeded*, as shown below (note: *s* for send):

s -t 339.098445441 -Hs 7 -Hd 13 -Ni 7 -Nx 919.10 -Ny 1058.78 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw — -Ma 13a -Md d -Ms 7 -Mt 800 **-Is 0.1 -Id 2.0** -It udp -Il 1711 -If 0 -Ii 0 -Iv 29 -Pn udp -Ps 4 -Pa 0 -Pf 1 -Po 0

s -t 339.126988442 -Hs 7 -Hd 13 -Ni 7 -Nx 919.10 -Ny 1058.78 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw — -Ma 13a -Md d -Ms 7 -Mt 800 **-Is 0.1 -Id 2.0** -It udp -Il 1711 -If 0 -Ii 0 -Iv 29 -Pn udp -Ps 4 -Pa 0 -Pf 1 -Po 0

s -t 339.185584239 -Hs 7 -Hd 13 -Ni 7 -Nx 919.10 -Ny 1058.78 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw — -Ma 13a -Md d -Ms 7 -Mt 800 **-Is 0.1 -Id 2.0** -It udp -Il 1711 -If 0 -Ii 0 -Iv 29 -Pn udp -Ps 4 -Pa 0 -Pf 1 -Po 0

s -t 339.187391148 -Hs 7 -Hd 13 -Ni 7 -Nx 919.10 -Ny 1058.78 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw — -Ma 13a -Md d -Ms 7 -Mt 800 **-Is 0.1 -Id 2.0** -It udp -Il 1711 -If 0 -Ii 0 -Iv 29 -Pn udp -Ps 4 -Pa 0 -Pf 1 -Po 0

s -t 339.221528837 -Hs 7 -Hd 13 -Ni 7 -Nx 919.10 -Ny 1058.78 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw — -Ma 13a -Md d -Ms 7 -Mt 800 **-Is 0.1 -Id 2.0** -It udp -Il 1711 -If 0 -Ii 0 -Iv 29 -Pn udp -Ps 4 -Pa 0 -Pf 1 -Po 0

s -t 339.234775746 -Hs 7 -Hd 13 -Ni 7 -Nx 919.10 -Ny 1058.78 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw — -Ma 13a -Md d -Ms 7 -Mt 800 **-Is 0.1 -Id 2.0** -It udp -Il 1711 -If 0 -Ii 0 -Iv 29 -Pn udp -Ps 4 -Pa 0 -Pf 1 -Po 0

s -t 339.243102655 -Hs 7 -Hd 13 -Ni 7 -Nx 919.10 -Ny 1058.78 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw — -Ma 13a -Md d -Ms 7 -Mt 800 **-Is 0.1 -Id 2.0** -It udp -Il 1711 -If 0 -Ii 0 -Iv 29 -Pn udp -Ps 4 -Pa 0 -Pf 1 -Po 0

d -t 339.244839564 -Hs 7 -Hd 13 -Ni 7 -Nx 919.10 -Ny 1058.78 -Nz 0.00 -Ne -1.000000 -Nl MAC -Nw **RET** -Ma 13a -Md d -Ms 7 -Mt 800 **-Is 0.1 -Id 2.0** -It udp -Il 1653 -If 0 -Ii 0 -Iv 29 -Pn udp -Ps 4 -Pa 0 -Pf 1 -Po 0

For the successful authentications, we computed the overhead then the ratio ρ for each

couple (n,$hops$). ρ is defined as:

$$\rho = \frac{overhead\ in\ motionless\ scenario}{overhead\ in\ moving\ nodes\ scenario}$$

The mean value of ρ **was computed and is equal to 0.23**, which means that a successful authentication in the moving nodes scenario produces approximately four times more overhead than the authentication in the motionless nodes scenario. As such, the routing overhead is three times more important than the authentication overhead in the motionless scenario.

The values of τ and ρ demonstrate that the success of an authentication and the amount of its overhead are conditioned by the routing process. However when the authentication is successful, its overhead is lower than 1.4 sec.

4 Impact of the Computing Time Overhead within the Motionless Scenario

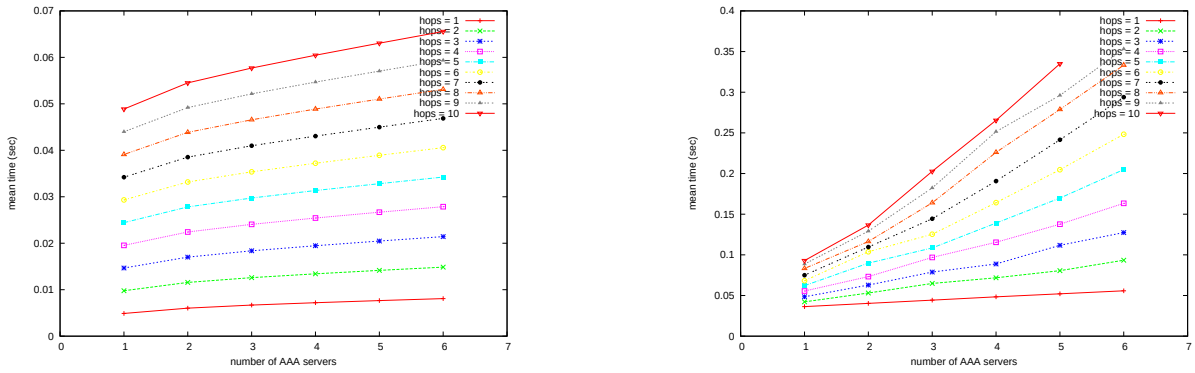

Figure 4.1: Impact of the computing time on the authentication overhead for the motionless scenario. On the left-hand side, in the "max" model. On the right-hand side, with the simulation

In addition to the routing overhead, we evaluated the computing mean time of the authentication messages content on a machine of 2 GHz processor:

- Random number generation: 0.000 ms
- RSA signature generation: 9.883 ms
- RSA signature verification: 0.391 ms
- Token generation: 9.873 ms

These values were integrated to the "max" model and the motionless nodes scenario (corresponding to the "sum" model) (cf. Figure 4.1). In the "max" model, the overhead rose by about 0.025 sec. However it increased by about 0.5 sec in the motionless simulation. This is logical because in the "max" model the computing time is added once, however in the motionless simulations corresponding to the "sum" model, it is added n times. Hence, again, there is an interest for the "max" model.

Conclusions and Future Works

In the report, we analyzed the overhead resulting from the authentication of a joining node by a distributed AAA infrastructure within a mobile ad-hoc network. The built model demonstrates that when routes are already established, the overhead increases as the number of servers rises and as the number of hops rises, as well. Its value does not exceed 40 milliseconds for a maximum of 6 servers and 10 hops. Simulations that were conducted following to the model pointed out that NS-2 is not the ideal simulator for distributed contexts where parallelism is needed. We modified the model to fit NS-2 simulation results. The obtained overhead values had again an upward trend as the number of servers and the number of hops increase, however their range is about nine times bigger. We conclude that because the simulations fit to the "sum" model and that this model has the same foundations as the "max" model, the "max" model is valid because the maximum is at most equal to the sum.

When some nodes are moving and AODV has to re-establish the routes, on average, the range of the produced overhead is multiplied by four for the successful authentications. The routing overhead is hence three times larger than the authentication overhead produced in the motionless case. On average, we can expect a full authentication accomplishment after at most 1.4 seconds for a number of servers less than 6 and a number of hops less than 10 (excluding the cases where routes can not be established because some nodes have been momentarily disconnected from the network). So the protocol is scalable when nodes move, too. This leads us to think that the authentication overhead won't be one of the serious impediments to distributed AAA framework implementations. Troubles will be more related to the AAA framework initialization and the accounting accomplishment.

Concerning the authentications that haven't been successful, their re-initialization has to be considered so that the necessary time to their completion can be estimated. The case when the number of joining nodes increases is to be treated, as well. Later, a trade-off has to be found between the number of AAA servers to use and the maximum accepted overhead.

Acknowledgment

We are thankful to ANR (Agence Nationale de la Recherche) financing the TLCOM MobiSEND project.

We are also thankful to the CEREGMIA department of the UAG (Universite des Antilles et de la Guyane) for their collaboration during the years 2007 and 2008.

Bibliography

[1] S. Larafa, M. Laurent-Maknavicius, and H. Chaouchi. Light and distributed AAA scheme for mobile ad-hoc networks. In *Proc. First Workshop on Security of Autonomous and Spontaneous Networks (SETOP 2008)*, pages 93–103, Loctudy, France, october 2008.

[2] S. Larafa and M. Laurent-Maknavicius. Protocols for distributed AAA framework in mobile ad-hoc networks. In *Proc. Workshop on Mobile and Wireless Networks Security (MWNS 2009)*, pages 75–86, Aachen, Germany, May 2009.

[3] C. Rigney, S. Willens, and A. Rubens. Remote authentication dial in user service (RADIUS). RFC 2865, June 2000.

[4] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko. Diameter base protocol. RFC 3588, September 2003. `http://tools.ietf.org/html/rfc3588`.

[5] IEEE Computer Society. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. IEEE Standard 802.11, June 1999.

[6] A. Shamir. How to share a secret. In *Communications of the ACM*, volume 22, pages 612–613, November 1979.

[7] V. Shoup. Practical threshold signatures. In *EUROCRYPT 2000*, volume 1807, pages 207–220, 2000.

[8] ISO [9798-3]. `http://www.iso.org/iso/fr/search.htm?qt=9798-3&published=on&active_tab=standards`.

[9] Giuseppe Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18:535–547, March 2000.

[10] Daniel Zwillinger and Stephen Kokoska. *CRC Standard Probability and Statistics Tables and Formulae*. CRC Press, 2000.

[11] Maurice Kendall, A. Stuart, and J.K. Ord. *The Advanced Theory of Statistics*, volume 1. Wiley, 2009.

[12] Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, New York, 1964.

[13] Alexander Sotirov, Marc Stevens, Jacob Appelbaum, Arjen Lenstra, David Molnar, Dag Arne Osvik, and Benne de Weger. Creating a rogue CA certificate. `http://www.win.tue.nl/hashclash/rogue-ca/`.

[14] UC Berkeley, LBL, USC/ISI, and Xerox PARC. The NS manual (formerly ns notes and documentation), January 2009. http://www.isi.edu/nsnam/ns/ns-documentation.html.

[15] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (AODV) routing. RFC 3561, July 2003. http://tools.ietf.org/html/rfc3561.