Towards multiple-exchange protocol use in distributed AAA frameworks for more autonomy in MANETs

Sondes Larafa, Maryline Laurent

Towards Multiple-Exchange Protocol Use in Distributed AAA Frameworks for More Autonomy in MANETs

Sondes LARAFA
Email: sondes.larafa@it-sudparis.eu

Maryline LAURENT
Email: maryline.laurent@it-sudparis.eu

CNRS Samovar UMR 5157, Institut Telecom, TELECOM SudParis, 9 rue Charles Fourier, 91000 EVRY, FRANCE

***Abstract*—In previous work, we designed a distributed AAA framework for MANETs for which we defined a simple and robust AAA authentication and authorization protocol whose specification carries some implementation latitude. The protocol, therefore, offers several options. In this paper, we propose some of the possible implementation options for which we conducted an analytical study and computationally intensive simulations to evaluate their performances. The objective is to provide guidelines for a fine tuning of this protocol.**

***Keywords*-Mobile Ad-hoc Networks (MANETs); Distributed systems architecture; Security; Modelling and simulation; Access control.**

I. Introduction

Mobile Ad-hoc Networks (MANETs) are wireless networks without an infrastructure – where the nodes involved are mobile and might come within the range of each other to communicate. Network algorithms and protocols are handled by the nodes themselves. Centralized authorities and administration management are not required especially once the network nodes are bootstrapped with the necessary information for network maintenance.

MANETs were initially conceived for specific purposes such as military [1] and rescue operations [2]. However, since IEEE standard 802.11 approval [3], and thanks to the spread of wireless devices, service providers have predicted commercializing public ad hoc network-based services [4], being attracted principally by the ease of deployment and the potential financial gain from MANETs [5].

MANETs are sufficiently developed and tested to guarantee the smooth running of the protocol stack (MAC layer [3] and routing protocols [6][7][8]), however the lack of a business model and an AAA framework (Authentication, Authorization, and Accounting) is still a major issue for service deployment.

In a previous work [9], we proposed a distributed AAA framework, termed *Dist-AAA*, for authorizing service access in MANETs and we focused on the definition of a suitable AA protocol (authentication and authorization) in order to give access only to the authorized clients. The conception of this distributed AAA framework followed some design guidelines of well known centralized AAA infrastructures, like Diameter [10], by using EAP (Extensible Authentication Protocol [11]) for carrying the authentication credentials. In this way, operators can extend their networks with a MANET and can mandate the MANET distributed AAA framework to support AAA operations. Moreover, some nodes from a MANET can jointly offer a service, for example a multimedia service, in a distributed fashion and deploy a distributed AAA service to control the access to it. Finally, in an emergency scenario, the most reliable nodes that is to say those responsible for the operation can take the role of AAA servers and distribute this service among them.

Dist-AAA is distributed on n nodes, termed AAA servers, using a (n,th) threshold cryptography scheme as defined by Shamir [12] and developed by Shoup [13]. The AA protocol supports *mutual* authentication between a client and the servers and enables servers to limit service access to successfuly authenticated clients only. In [14], we studied the performances of the AA protocol in a simple case where $th = n$ and where the protocol is executed between one single client and an increasing number of servers.

In this paper, we investigate performance when $th \leqslant n$ within a more realistic scenario i.e. within a MANET composed of one hundred nodes randomly moving in a squared area. Under these conditions, we presumed that the protocol specifications carry some implementation latitude, therefore we identified several implementation options.

In the next section, we give an overview of related works and point out our main contributions. In Sect. IV, we describe our proposals for implementation options: the first option introduces two supplementary thresholds, in addition to the cryptographic threshold th, the second option is to renew the authentication process in accordance with a multiple-attempt *back-off algorithm* if a first attempt of authentication failed. The last option is about making a client start the second protocol exchange before the end of the first one i.e. before the arrival of *sufficient* responses from the solicited servers, which is the opposite of the initial approach [9]. Sect. V to VII highlight our methodology to set suitable parameters for the first and the second options so that the protocol performances can be optimal. The seventh section compares the performances of the third option approach with the initial approach.

II. Related Works and Main Contributions

So far, many authentication solutions in ad hoc networks have been defined and studied, [15] examines 21 authentication solutions and [16] surveys 13 of them. Both articles propose different criteria to classify them and they provide a thorough analysis of the state of the art up to 2005. The authentication solutions presented were not designed for AAA infrastructures and are not suitable for such frameworks. That is why, [15] points towards the need for defining a framework responsible for authentication management in MANETs. Such a framework can also be used for to authorizing and accounting purposes.

More recently, AAA services in MANETs have been addressed. In [17], an interesting business scenario for selling multimedia services in MANETs is proposed. Authors focus on non repudiation when charging clients. The charging is supervised by an AAA architecture in the operator's network in whose domain the selling and buying nodes belong. The solution is original, but is highly dependent on the operator's network. Moreover no performance evaluation is provided. [18] proposes to relieve the burden on different servers located in a MANET using peer to peer methods and especially JXTA. Progress on many issues has been made possible, but with respect to AAA servers, the gain in computing time comes at the cost of additional messages. Finally, [19] considers the extension of the Kerberos authorization service of the operator's network to the ad hoc network in order to allow single nodes to offer multimedia services. The solution depends on the centralized Kerberos server which acts as a trusted third party and does not address distribution on the ad hoc network nodes themselves.

Moreover, threshold cryptography that we intend to use for AAA service distribution within a MANET, has already been investigated as a means of distributing a certification authority for key management in MANETs. This research work known as the MOCA framework (MObile Certificate Authority) [20] was conducted by Yi and Kravets, and led to defining a *single exchange* protocol. Through the introduction of β-unicast and the safety margin used, Yi and Kravets noted the importance of addressing more than th servers to improve their protocol success ratio. As far as we know, no analytical studies have yet been undertaken to identify the fine tuning of the number of addressed servers.

In this paper, an analytical study is performed on a *2-exchange* AA protocol that supports authentication and authorization of nodes and that leads each of them to possess an access token. Note that our mathematical method is relevant to any other protocols composed of two exchanges and executed between a node and a group of servers. The token is a kind of passport that a client carries in his traffic to prove that he was authorized to access the services that he already paid for. Note that the literature reports no other work referring to a protocol providing such an access method in a single exchange. Moreover, solutions like MOCA, which are of real interest as previously noted, seem to offer an alternative in using a single exchange protocol, but a careful study of what it really provides (i.e. certificates) demonstrates that at least one more exchange is necessary to actually reach the authentication level. As such, our 2-exchange protocol is to be considered more or less minimal.

We subsequently validate our mathematical analysis when $th = 5$ and $th = 10$ (cf. Sect. V): a novel simulation approach that introduces ambient traffic in addition to the AA traffic was conducted (cf. Sect. VI-A2). Our contributions, amongst other things, include the definition of an algorithm for renewing AA attempts in case of failures and the proposal of interleaving messages of the two exchanges. Both lead to improvements in protocol performance.

III. AAA Exchanges overview

At this point, a client that we have called a joining node (JN), needs to contact at least a threshold number, th, of servers out of a total of n AAA servers for two exchanges in order to ensure both authentication and authorization. Each exchange consists in a *request* and a *reply*. The *1st request* is in fact composed of at least th *unicast* messages, each of them addressed to one server and traveling on a *multi-hop route*. The *1st reply* is composed of *unicast* responses coming from the solicited servers and traveling also on *multi-hop routes*. The *2nd request* and the *2nd reply* are similarly formed. The process is considered successful if the *1streply* and the *2nd reply* are composed of at least th responses. Here is the composition of each kind of message:

1) *1st request*: JN AA-Requests containing its identity ID$_{JN}$ encapsulated in an EAP message.
2) *1st reply*: servers AA-Answers each one containing a random number encapsulated in an EAP message. Assuming that servers are noted srv$_1$, srv$_2$, ..., srv$_n$, the AA-Answer of srv$_i$ contains R$_i$.
3) *2nd request*: JN AA-Requests each one containing its public certificate, a random number R$_{JN}$ generated by JN, the identity ID$_{AAA}$ of the AAA service and a signature, all of them encapsulated in an EAP message. The signature destined to srv$_i$ is computed on (R$_{JN}$, R$_i$, ID$_{AAA}$).
4) *2nd reply*: each server verifies the validity of the information sent by the JN and responds only after a successful authentication. Its AA-Answer contains the certificate of the AAA service, ID$_{JN}$ and its partial signature on (R$_i$, R$_{JN}$, ID$_{JN}$) computed with its key share [12][13], all of them encapsulated in an EAP Success message. This message also contains a part of the access token that JN will use to prove its authorization.

JN combines the received partial signatures and obtains the signature of the AAA service. If the validity of this signature and the other received information is proved, the

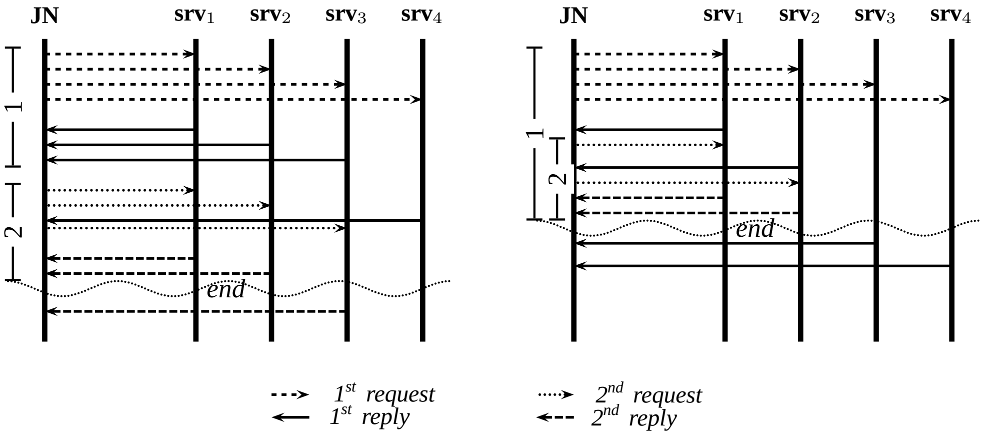

Figure 1. $th_M = 4$, $th_i = 3$ and $th = 2$. On the left-hand side: *separation* case. On the right-hand side: *interleaving* case

AAA service is correctly authenticated. JN combines then the access token parts in order to henceforward insert it in its traffic.

IV. Options for Protocol Implementation

The previous article [9] focused on safety aspects of the protocol and presented the cryptographic content of the protocol messages to ensure the AA functions. It did not specify an implementation that considers performance aspects in terms of AA time and AA success ratio. The objective of this section is to identify and detail some possible implementation options that will be evaluated later with respect to the parameters chosen.

A. Introducing Two New Thresholds: th_M and th_i

The system signing key is shared between n servers in such a way that a set of at least $th \geqslant 2$ servers can achieve signing, and thus authentication and authorization in our case. For a given n, there is a trade-off between secrecy (th should be maximum) and reliability (th should be minimum). It is commonly admitted, as it was first pointed out in [12], that a good compromise would be to guarantee normal operating when at most $th-1$ servers fail. Then, a good relation between n and th is $\lfloor n/2 \rfloor \geqslant th - 1$. Therefore, we assume that $2 \leqslant th \leqslant \lfloor n/2 \rfloor + 1$ throughout this article.

There is also a trade-off between security and convenience. Increasing the threshold value th achieves better security, since a mobile adversary [21] has to compromise at least th servers to cause the system to break down. But the latter becomes inconvenient because the chance of a JN to get th responses decreases and the protocol overhead increases.

Once the value range of th is fixed and these points have been identified, a question is still pending: how many servers should a JN contact to maximize its AA success ratio or, alternatively, to minimize the protocol overhead? For minimizing the overhead, th is the obvious solution, for maximizing the success ratio, n is the obvious one. But th obviously minimizes the success ratio and n maximizes the overhead. So it seems natural to choose an intermediate number, say th_M. JN contacts th_M servers and waits for th responses.

This might be optimal for the first exchange, but is not sufficient for the whole process because we need to define the number of servers to contact during the second exchange. Obviously, it is unnecessary to contact the same th_M servers but to choose instead a lower number from those who responded. For the same reasons as previously argued, contacting only th servers is also excluded. So an intermediate number, th_i, between th and th_M needs to be defined.

To summarize, we define two thresholds th_M and th_i such that $th < th_i < th_M \leqslant n$. JN contacts th_M servers during the first exchange. It waits for at least th_i responses. It then contacts th_i servers, for example those which were the fastest to respond and waits for th responses to complete authentication and authorization. Using an analytical approach, Sect. V shows how to select the values of these thresholds to obtain a high AA success ratio.

B. Introducing Multiple AA Attempts

Reaching 100% of success in a unique AA attempt is clearly unrealistic. That is why, in case of failure, the AA process can be renewed in accordance with an algorithm that we call a *back-off algorithm,* based on three parameters: the waiting time before giving up an attempt, the waiting time before initiating a new one, and the maximum number of attempts. For the sake of simplicity, we set the waiting time before giving up an attempt equal to the waiting time before initiating a new one and we denote it t_{max}. We determine t_{max} and the maximum number of attempts, denoted K, respectively in Sect. VII and Sect. VIII.

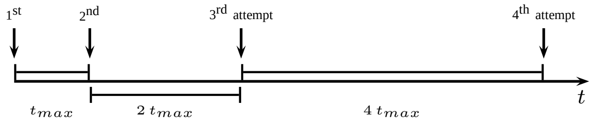

Figure 2. Multiple-attempt *back-off algorithm*

Figure 2 depicts the course of the *back-off algorithm*: if

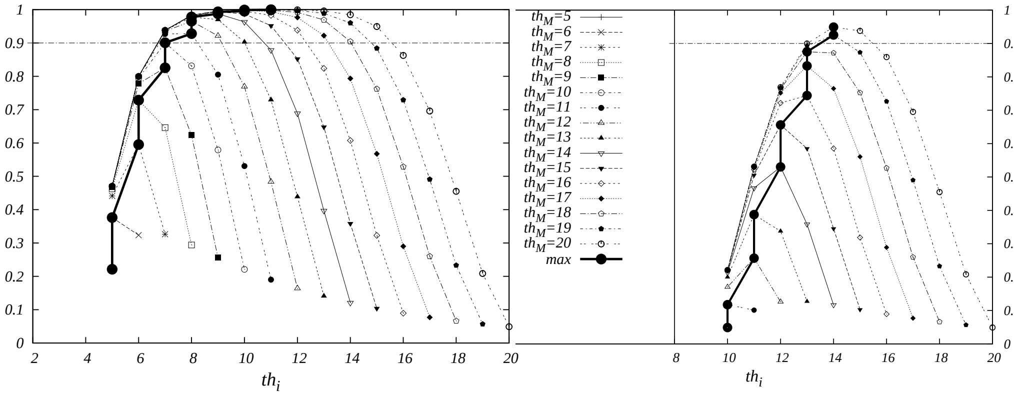

Figure 3. On the left-hand side: $\widehat{P_2}$ when $th = 5$, $th_M \in \{7, .., 20\}$. On the right-hand side: $\widehat{P_2}$ when $th = 10$, $th_M \in \{12, .., 20\}$

a first attempt fails after waiting t_{max}, JN initiates a second attempt. If it fails after waiting $2\,t_{max}$, JN tries a third time. Generally speaking, after initiating a *j**th* attempt, the JN waits $2^{j-1}\,t_{max}$ before trying again. If after j attempts, the authentication and authorization succeeds, the total process time is then between $(2^{j-1} - 1)\,t_{max}$ and $(2^j - 1)\,t_{max}$. Finally, if after K attempts there was no success, JN gives up authentication attempts. This algorithm allows unfavorable network conditions to be taken into account by increasingly spacing the new attempts. It also allows the success ratio to be raised.

C. Interleaving the First and Second Exchanges

The initial approach separates the first and the second exchanges. With the introduction of the thresholds th_i and th_M, we saw that a JN waits for the th_i server responses before initiating the second exchange. Another possible approach is to interleave exchanges by making a JN send a message to each responding server as soon as it receives its response, without waiting for the arrival of all the th_i responses. This is feasible because an exchange is done between a JN and a set of servers and not only a single server. We call *separation* the case where a JN waits until the end of the first exchange before initiating the second one and *interleaving* the other case.

Figure 1 shows that, in the *interleaving* case, an AA process might be successful with responses from less than th_i servers to the *1st request*. This might improve the process time and so t_{max} in addition to the success ratio because fewer messages are needed to complete authentication and authorization.

In Sect. V, for the sake of simplicity, we carry out our analysis in the case of *separation*. We do the same when estimating the maximum waiting time t_{max} in Sect. VI and the maximum number of attempts K in Sect. VII.

V. Setting Thresholds for a High Probability of the AA Success

From a purely theoretical point of view, we suppose that each solicited server responds with the same probability p. We also suppose that servers act independently. In this case, the probability of getting a reply from an exact number of th servers is p^{th}. This is a decreasing function of the cryptographic threshold th: *the probability of getting a reply decreases when the security increases*. To override this issue, a JN has to request $th_M \geq th$ servers and accept th or more responses. In this case, the probability becomes:

$$P_1(th, th_M) = \sum_{j=th}^{th_M} \binom{th_M}{j} p^j (1-p)^{th_M - j} \tag{1}$$

Now, we have to manage two successive exchanges. In the same manner, a JN asks th_M servers and validates the first exchange when th_i or more answers arrive. Next, for the second exchange, it asks th_i servers and completes the second exchange when it gets th or more answers. The probability of success is:

$$\begin{aligned} P_2(th, th_i, th_M) &= P_1(th, th_i) \cdot P_1(th_i, th_M) \\ &= \left(\sum_{j=th}^{th_i} \binom{th_i}{j} p^j (1-p)^{th_i - j} \right) \cdot \\ &\quad \left(\sum_{j=th_i}^{th_M} \binom{th_M}{j} p^j (1-p)^{th_M - j} \right) \end{aligned} \tag{2}$$

Our estimate of the probability p is $\hat{p}$=0.85 and it was obtained from previous simulations. On the left-hand side of Figure 3 are shown, for several values of $th_M \in [5, 20]$, typical variations of $\widehat{P_2}$, the estimate of P_2, as a function of $th_i \in [th, th_M]$ when $th = 5$, while on the right-hand side are shown, for several values of $th_M \in [10, 20]$, typical variations of $\widehat{P_2}$ when $th = 10$.

When $th = 5$, $\widehat{P_2}$ exceeds 90% for $th_M \geq 10$ and $th_i \geq 7$. Though increasing th_M improves $\widehat{P_2}$, it is not worthwhile choosing $th_M \geq 11$ because maximum values reached by $\widehat{P_2}$ flatten out (see the thick curve *"max"*). The relative gain is in fact no longer significant. It is between around 3% from $th_M = 10$ to $th_M = 11$ and 0.04% from $th_M = 19$ to $th_M = 20$. Taking higher values of th_M will rather cause needless overhead. When $th = 10$, the relative gain decreases as we increase th_M and for appropriate th_i (see the thick curve *"max"*). It is between around 5% from $th_M = 18$ to $th_M = 19$ and 2.5% from $th_M = 19$ to $th_M = 20$.

Therefore in the following sections, we select $th_M = 11$ and $th_i = 8$ when $th = 5$, as well as $th_M = 19$ and $th_i = 14$ when $th = 10$.

VI. Estimating the Maximum Waiting Time t_{max}

A. Simulation Settings

1) General Settings: We developed the protocol implementation options on ns-2.34 and we evaluated them by simulating what would be considered as a small town network. For wireless transmissions, we set the Phy/WirelessPhy and Mac/802_11 ns modules to simulate an ORiNOCO 802.11b card as indicated in [22][23]. There are no channel errors. The propagation model used is the two-ray ground reflection.

The network is multi-hop and composed of 100 wireless nodes moving on a 600mx600m area at a maximum speed of 10m/s and a pause time of 0 sec. 20% of them are AAA servers, which we believe is largely enough, and the other 80 nodes are JNs. The most widely used node mobility model in the literature is the *Random Waypoint Model* (RWM) [24]. However, several authors reported issues when using *a priori* uniform distribution of nodes in the square [25] and a zero based uniform distribution of the speed [26]. So we used the package *mobgen-ss* [27] that allowed us to define a stationary distribution of both the positions and the speeds of nodes. We ran the simulator once but for a long enough time because this stationary distribution guarantees the process ergodicity.

We assume that the bootstrapping phase had already taken place. So the JNs already know the addresses of the servers, the parameters n and th, as well as th_M and th_i. Also, AAA servers share a public certificate and each JN has its own public certificate. Finally, JNs have been previously authenticated and authorized, as well.

When a node needs to authenticate again, it randomly chooses th_M AAA servers among the 20 servers in order to solicit them. Once authenticated, the node is authorized and assigned a token that permits access to the network services for a limited period of time. We fix this period to 1 hour, after which the node needs to authenticate and to ask for a new access token again. The simulation duration is 01:06:40 i.e. 4000 sec, which allows to observe nodes behavior after expiry of their tokens at least once and at most twice (cf. table I).

Table I
Summary of setting parameters

Parameter	Value
Surface area	600mx600m
Mobility model	Steady state RWM
Node speed	mean=5.2m/s, delta=4.75m/s, max=10m/s
Pause time	mean=0 sec , delta= 0 sec
Propagation model	Two-ray ground reflection
Wireless card	ORiNOCO 802.11b
Routing protocol	AODV
Nbr of nodes	100
Nbr of servers	20
Nbr of JNs	80
Thresholds	$\{th = 5,\ th_i = 8,\ th_M = 11\}$, $\{th = 10,\ th_i = 14,\ th_M = 19\}$
Access token deadline	1h
Simulation time	4000 sec i.e. 01:06:40
Auth. req. time	Exponentially drawn(mean=600 sec)
Nbr of CBR generators	50
CBR traffic rate	$r = 19.7$ Kbps

2) Traffic Model: The arrival time of nodes' AA requests follows an exponential distribution of mean 600 sec. So each node practically authenticates once during the first hour of the simulation. Furthermore, to be closer to real network traffic, the simulations were performed in a *moderately charged network operating at a "cruise speed"*. That is why some traffic (data, signaling, etc) differing from the AA traffic was injected and simulated by a Constant Bit Rate (CBR) traffic[1] whose rate was set by applying a bisection method explained below.

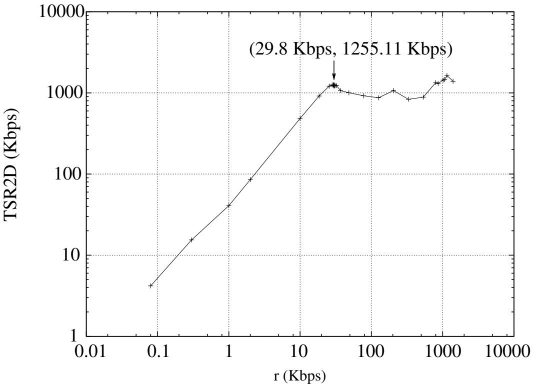

Figure 4. Total per Second Received Data at Destinations (TSR2D) vs. r

All nodes are considered having similar roles. As they randomly move, it is worthless to randomly choose end nodes of connections. As such, assuming that nodes IDs are $(n_k)_{k \in \{0,..,99\}}$, it was sufficient to set up the following:

[1] CBR traffic is more or less realistic than Poisson process traffic but allows to considerably decrease the number of events during each simulation for narrow deviations. As a matter of fact a typical single simulation use between 30 and 45 minutes of computer time and it would otherwise be multiplied by ten.

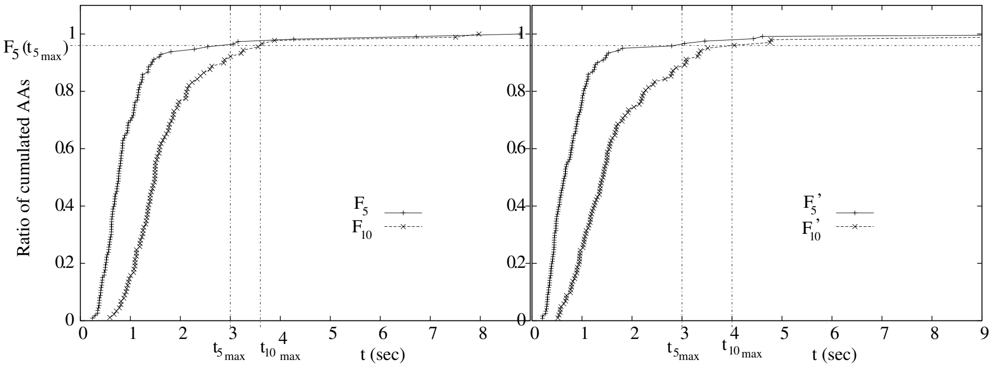

Figure 5. Cumulated AAs for $th = 5$ and $th = 10$. On the left-hand side: *separation case*. On the right-hand side: *interleaving case*

node n_k emits a CBR at rate *r* towards the node n_{n-k-1} for $k \in [0, 49]$. Therefore, 50 CBR traffic generators, each at *r* Kbps, were created and 50 connections were established.

To appreciate what a *moderately charged network* is, we first determined what a fully charged network would be, and then reduced it arbitrarily to $\frac{2}{3}$. We did that in the above defined network but with no AA traffic. With $r = 0.08$ Kbps, we got a Total per Second Received Data at Destinations (TSR2D) of 4.19 Kbps. With $r = 1400$ Kbps, the TSR2D was near 1390 Kbps. With $r = 700.04$ Kbps, the TSR2D was around 900 Kbps. With $r = 350.02$ Kbps, the TSR2D was near to the previous value. Full results of the TSR2D vs *r* are given in Figure 4. A tedious but straightforward use of bisection method led us to determine $r = 29.88$ Kbps as the rate where the network became fully charged; TSR2D no longer increases linearly and network performance weaken. Thus, we choose $\frac{2}{3}$ of this value as the CBR rate, i.e. ***r*=19.7 Kbps**.

B. Computing t_{max} and Validating the Probability of the AA Success

Considering first of all that the AA processes are executed in a single attempt without applying the *back-off algorithm*, simulations show that the experimental estimation of the probability of success, i.e. the success ratio is around 91% for $th = 5$, which is very close to the analytical estimation i.e. 92.8% (cf. Sect. V). However, it is around 72% for $th = 10$, which is not very close to the analytical estimation i.e. 92%. The reason is that the flurry of responses arriving to JN from the servers for larger th increases the collisions ratio by a sent AAA message (cf. the last column of table III), and so decreases the probability p of server response. Meanwhile, the success ratio for $th = 10$ is algebraically more sensible to the estimation of p than the success ratio for $th = 5$. In fact, if $p = 0.8$ rather than 0.86 as it was the case in Sect. V, it would have been equal to 72%!

Moreover, we denote F_5 and F_{10} the cumulative distribution function of the AA time for respectively $th = 5$ and $th = 10$. Their shape is drawn on the left-hand side of Figure 5. We can reasonably estimate t_{max} as the maximum waiting time where 96% of successful AAs were achieved, then $t_{max} = t_{5_{max}} = 3$ sec for $th = 5$ and $t_{max} = t_{10_{max}} = 3.8$ sec for $th = 10$ (i.e. $F_5(t_{5_{max}}) = F_{10}(t_{10_{max}}) = 0.96$). These values will be used in the next sections when applying the multiple-attempt *back-off algorithm*.

VII. Setting the Maximum Number of Attempts K

Given P_2, the probability of AA success defined in Sect. V, and F, the cumulative distribution function of the AA time for a triplet (th, th_i, th_M), the probability that an attempt fails after waiting t_{max} is: $Q = 1 - P_2\, F(t_{max})$, and the probability of failure after K attempts is: Q^K.

$lim\, Q^K = 0$, so if a JN makes a sufficient number of attempts, it will finally authenticate. According to simulations, $K = 5$ for $th = 5$ and $K = 10$ for $th = 10$ are largely enough to reach 100% of success. Table II illustrates the fraction of additional successful AA processes for each number of attempts between 1 and K.

Table II
Fraction of successful AA processes vs number of attempts

#attempts thr.	1	2	3	4	5	6	7	8	9	10
th =5, th_M=11, th_i=8	0.847	0.105	0.016	0.016	0.016	0	0	0	0	0
th =10, th_M=19, th_i=14	0.613	0.137	0.145	0.089	0	0	0	0.008	.008	0

VIII. Contribution of the *Interleaving* Approach over the *Separation* Approach

Success ratios and number of attempts were slightly enhanced in the *interleaving* case (cf. table III). Figure 5 illustrates on the right-hand side F_5' and F_{10}', the cumulative distribution functions of the AA time for respectively $th = 5$ and $th = 10$ in the *interleaving* case. We notice that the AA time was not really improved. Moreover, for $th = 5$ (respectively $th = 10$) the mean AA time in *separation* case is 1.02 sec (respectively 2.28 sec) and in *interleaving* case is 0.93 sec (respectively 2.56 sec). This little enhancement for $th = 5$ and regression for $th = 10$ can be explained by the fact that even if the number of sent AAA messages per AA

Table III
Comparison of the *interleaving* vs *separation* approaches

Metrics Approach	P_2	$t_{max}(sec)$	K	sent AAA/auth.	col./sent AAA (%)
Separation, $th = 5$	0.91	3	5	39	63.84
Interleaving, $th = 5$	0.98	3	4	35	69.25
Separation, $th = 10$	0.72	3.8	10	78	75.18
Interleaving, $th = 10$	0.82	4	8	71	83.40

process were generally decreased in *interleaving*, collisions per sent AAA message, and so per AA process, were meanwhile increased (cf. table III): a message immediately sent by a JN after receiving a response from a server has higher risks to collide with the responses arriving from the other servers therefore increasing the contention.

IX. Discussion and Conclusion

In this paper, we proposed implementation options for an authentication and authorization protocol in a distributed AAA framework in MANET. We provided guidelines for setting their parameters and investigating their performances. The analytical study highlighted the procedure to be followed for optimally setting the number of servers to contact and getting a high AA success ratio. Though that analysis focused on a 2-exchange protocol, it should be noted that it can be utilized for protocols of one, two, three or more exchanges by considering the product of one, two, three or more factors of P_1 type, as defined in Sect. V. Multiple-exchange protocols can be needed for establishing stronger trust between a distributed service and a client if sensitive traffic is involved (billing traffic, secret information, etc). When servers and link capacities are high enough, they can also be used for sending data traffic (multimedia, files, etc) from several servers to a client (e.g. Mesh networks, MANET comprising laptops, etc). The simulation results confirmed the predicted probability of success and served to define the necessary parameters of a multiple-attempt *back-off algorithm*. This algorithm applies in case of unsuccessful AA processes to reach 100% of success.

Besides, interleaving exchanges slightly enhanced the success ratio and the maximum number of attempts K, but they insignificantly improved the AA process time. As such, further investigations are needed in this direction.

We believe that our work is novel because it provides a methodology to optimize the AA success ratio of a 2-exchange protocol by first defining new thresholds, then designing a multiple-attempt *back-off algorithm*, and finally proposing the *interleaving* case. Our approach is original. It combines analytical and simulation studies. It also introduces the injection of ambient traffic in addition to the authentication and authorization traffic to investigate the protocol performances. Our results demonstrate that the process of regular authentications and authorizations does not charge the network more than 1% of the total throughput.

In the future, we will improve our analytical study to better fit to the cases where the cryptographic threshold th is large. Until now, we focused on the two first As of AAA. This work might be extended to the last A i.e. accounting by using the access token expiry time.

Acknowledgment

The authors are thankful to Chafika Sait for the extensive simulation works that she conducted. They also wish to thank ANR (Agence Nationale de la Recherche) for financing the TLCOM MobiSEND project. They would like to thank the anonymous reviewers for their valuable comments and suggestions.

References

[1] B. Leiner, R. Ruth, and A. Sastry, “Goals and challenges of the DARPA GloMo program,” *IEEE Personal Communications*, vol. 3, no. 6, pp. 34–43, 1996.

[2] M. Pužar, J. Andersson, T. Plagemann, and Y. Roudier, “Skimpy: A simple key management protocol for MANETs in emergency and rescue operations,” *Security and Privacy in Ad-hoc and Sensor Networks*, pp. 14–26, 2005.

[3] IEEE Computer Society, “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” IEEE Standard 802.11, June 1999.

[4] H. Moustafa, G. Bourdon, and Y. Gourhant, “AAA in vehicular communication on highways with ad hoc networking support: a proposed architecture,” in *Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*. ACM, 2005, p. 80.

[5] C. Chlamtac and J. Liu, “Mobile ad hoc networking: imperatives and challenges,” *Ad Hoc Networks*, pp. 13–64, July 2003.

[6] “IETF WG, Mobile Ad-hoc Networks (MANET),” http://datatracker.ietf.org/wg/manet.

[7] “IETF WG, Ad-Hoc Network Autoconfiguration (Autoconf),” http://datatracker.ietf.org/wg/autoconf.

[8] “Algorithms and protocols for wireless, mobile ad hoc networks,” A. Boukerche, Ed. Wiley-IEEE Press, 2009, p. 496.

[9] 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)*, Aachen, Germany, May 2009, pp. 75–86.

[10] P. Eronen, T. Hiller, and G. Zorn, "Diameter Extensible Authentication Protocol (EAP) Application (RFC 4072)," *IETF*, August 2005.

[11] B. Aboba, L. Blunk, J. Vollbrecht, and J. Carlson, "Extensible Authentication Protocol (EAP) (rfc 3748)," http://tools.ietf.org/html/rfc3748, June 2004.

[12] A. Shamir, "How to Share a Secret," in *Communications of the ACM*, vol. 22, November 1979, pp. 612–613.

[13] V. Shoup, "Practical Threshold Signatures," in *EUROCRYPT 2000*, vol. 1807, 2000, pp. 207–220.

[14] S. Larafa and M. Laurent, "Authentication protocol runtime evaluation in distributed AAA framework for Mobile Ad-Hoc Networks," in *2010 IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS)*, June 2010, pp. 277 – 281.

[15] N. Aboudagga, M. T. Refaei, M. Eltoweissy, L. A. DaSilva, and J.-J. Quisquater, "Authentication protocols for ad hoc networks: taxonomy and research issues," in *Q2SWinet '05: Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*. New York, NY, USA: ACM, 2005, pp. 96–104.

[16] Katrin Hoeper and Guang Gong, "Models of Authentication in Ad Hoc Networks and Their Related Network Properties ," 2004.

[17] G. Kounga and C. Schaefer, "Selling multimedia resources in ad hoc networks," in *IEEE Communications Magazine*, vol. 46, February 2008, pp. 126 – 131.

[18] O. Botero and H. Chaouchi, "Platform and experimentation of secure service location with P2P/Client-Server over ad hoc networks," in *Proceedings of the 2nd IFIP conference on Wireless days*, ser. WD'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 322–326.

[19] H. Moustafa, J. Forestier, and M. Chaari, "Distributed authentication for services commercialization in ad hoc networks," in *Mobility '09: Proceedings of the 6th International Conference on Mobile Technology, Application &#38; Systems*. New York, NY, USA: ACM, 2009, pp. 1–8.

[20] S. Yi and R. H. Kravets, "MOCA: Mobile Certificate Authority for Wireless Ad Hoc Networks," in *The Second Annual PKI Research Workshop (PKI)*, 2003.

[21] R. Ostrovsky and M. Yung, "How to withstand mobile virus attacks (extended abstract)," in *Proceedings of the tenth annual ACM symposium on Principles of distributed computing*. ACM, 1991, pp. 51–59.

[22] W. Xiuchao, "Simulate 802.11 b channel within ns2," *National University of Singapore, Singapore*, 2004.

[23] J. Robinson, "Making ns-2 simulate an 802.11b link," April 2005.

[24] S. Kurkowski, T. Camp, and M. Colagrosso, "Manet simulation studies: the incredibles," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 9, no. 4, pp. 50–61, 2005.

[25] C. Bettstetter, "Smooth is better than sharp: a random mobility model for simulation of wireless networks," in *MSWIM '01: Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*. New York, NY, USA: ACM, 2001, pp. 19–27.

[26] J. Yoon, M. Liu, and B. Noble, "Random waypoint considered harmful," in *IEEE INFOCOM*, vol. 2. Citeseer, 2003, pp. 1312–1321.

[27] W. Navidi, T. Camp, and N. Bauer, "Improving the accuracy of random waypoint simulations through steady-state initialization," in *15th International Conference on Modeling and Simulation (MS'04)*, March 2004, pp. 319–326.