Using the EAP Framework for Fast Media Independent Handover Authentication[*]

Antonio Izquierdo
antonio.manzanares@nist.gov

Nada Golmie
nada.golmie@nist.gov

Katrin Hoeper
katrin.hoeper@nist.gov

Lidong Chen
lily.chen@nist.gov

National Institute of Standards and Technology
100 Bureau drive, stop 8920
Gaithersburg, MD 20899-8920, USA

ABSTRACT

In this paper we compare different authentication techniques that may be used in order to reduce the time and resources required to perform a handover: namely, re-authentication and pre-authentication. We provide a detailed analysis of each mechanism, and highlight key performance trade-offs for select scenarios and parameters of interest.

Categories and Subject Descriptors

B.8.2 [**Performance and Reliability**]: Performance Analysis and Design Aids; E.3 [**Data Encryption**]: Standards

General Terms

Security, Standardization, Performance

Keywords

Authentication, Pre-authentication, Re-authentication, Handover, EAP, Performance

1. INTRODUCTION

Key characteristics of today's networking environment are the availability of different types of networks and the ability to roam across these networks on an as needed basis. As network heterogeneity has become more present, user mobility has also increased, making roaming from one network to another a common practice. However, in many cases roaming often implies a temporary service disruption, which is unacceptable for many user applications that have stringent quality of service requirements, such as voice and video. Thus, a number of tools and mechanisms are provided in order to minimize service disruptions. At a minimum, these mechanisms should include mobility protocols to manage the network transitions, tools for continuously monitoring the network status, and mechanisms for fast authentication in the new network. Efficient mobility mechanisms provide the means to reduce the operations to be performed during the network entry. Some of these protocols include Mobile IPv6 [11] and Fast Mobile IPv6 [12] that improve the performance of the layer-3 handover. Similarly, there are mechanisms that aim to improve the layer-2 handover, such as the handover optimizations introduced in 802.16e [9] and the fast Base Station (BS) transition in 802.11r [8].

Although authentication may be considered as an intrinsic part of network entry, it is always treated separately due to its many security implications. Furthermore, and as we shall see in the later sections of this article, authentication involves complex operations requiring considerable amount of time and computational resources. Thus, reducing the authentication time is critical to the overall mobility performance and to achieving seamless handovers.

In this article we focus on studying efficient mechanisms for fast authentication in heterogeneous networks. We analyze re-authentication [13] and pre-authentication [16] in the context of a media independent framework such as the Extensible Authentication Protocol (EAP) [2] and compare these schemes to full authentication as performed at connection set-up time. For each technique we provide a detailed analysis and the means to estimate the authentication latency in terms of key network parameters, such as the propagation delay between nodes, the number of messages exchanged, and the cryptographic processing time required.

We also analyze the relevance of the authentication in handovers, especially in optimized ones, in order to lay the basis for further studies in this area. Although each wireless technology provides its own means for optimizing handovers between networks using that specific technology, our focus is on media independent handovers, which can be optimized using the IEEE 802.21 ([10]). This draft standard defines mechanisms to notify about the imminent loss of connection, so that both the mobile node and the network may take the necessary steps to perform a handover to a new network in a seamless way.

The remainder of this paper is organized as follows. Section 2 presents an overview of authentication in heteroge-

[*]This research is supported by the NIST/ Office of Law Enforcement Standards (OLES)

WICON '08 Maui, Hawaii USA
Copyright 2008 ICST 978-963-9799-36-3 ...$5.00.

Digital Object Identifier: 10.4108/ICST.WICON2008.4814
http://dx.doi.org/10.4108/ICST.WICON2008.4814

neous networks. In section 3 we describe and analyze each authentication scheme considered. In section 4 we introduce the simulation environment used to obtain the results presented in section 5. In Section 6 we will study the relation between the authentication techniques and the optimized handovers. Finally, section 7 has our concluding remarks.

2. AUTHENTICATION IN HETEROGENEOUS NETWORKS

In this section we provide a brief overview of how authentication is achieved in heterogeneous networks. Our goal is to provide insights on current practices and point out similarities found in various authentication techniques used in heterogeneous networks.

Currently, each network technology defines its own authentication mechanism that will allow the Point of Attachment (PoA) to validate the device's credentials, and grant or deny access to the network. Considering authentication mechanisms defined in IEEE 802.11r, IEEE 802.16e, and Universal Mobile Telecommunications System (UMTS) [1], it is easy to see that, although the basic steps are comparable, all three technologies use different messages, cryptographic operations, and key derivation schemes.

However, in spite of these differences, there are a number of similarities when it comes to achieving fast authentication. The basic idea is to reuse the cryptographic information negotiated in previous authentications, so that most of the exchanged messages can be omitted during a fast handover. For example, in IEEE 802.16e the Pairwise Master Key (PKM) and the Traffic Encryption Key (TEK) can be reused, and in IEEE 802.11r during a fast Basic Service Set (BSS) transition the first level Pairwise Master Key (PMK-R0) is reused.

Extending these principles to heterogeneous networks implies reusing cryptographic material previously negotiated for a different network. This can only be achieved using a common authentication architecture such as EAP, that provides a technology-independent framework for running authentication protocols. This framework consists of a set of messages and formats to transport the authentication messages, which are defined by the specific authentication algorithm used, called 'EAP method'. These methods define the messages to be sent by each peer and the way in which, upon successful authentication, a cryptographic key is agreed.

The EAP framework also defines several entities that play different roles during the authentication process: the supplicant, the Network Access Server (NAS), the authenticator and the server. The **supplicant** is the device that needs to be authenticated. The **NAS** is the device that provides network access to the supplicant. The **authenticator** receives the authentication requests forwarded by the NAS, and starts the authentication exchange with the supplicant. After the authentication process has started, the authenticator plays the role of a cache for the server. The **server** is the entity that validates the supplicant's credentials and grants or revokes the network access. The relationship between these entities is purely hierarchical, as an EAP server may have several related authenticators, which in turn have several NASs forwarding messages and providing access services to many supplicants.

In wireless networks, the NAS represents the PoA, which is the base station in IEEE 802.16 or the Access Point (AP) in IEEE 802.11, and the supplicant is the mobile node (MN). The physical location or the type of device of the authenticator and the server are not specified as they are usually located in the backbone network.

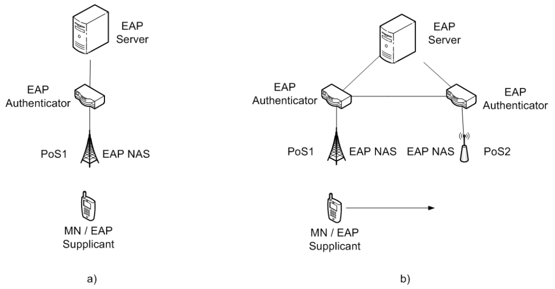


Figure 1: Base Network Topology

A base topology including all the entities described above is shown in Figure 1a. When considering handovers, this topology is easily extended to Figure 1b, where both the current and target networks share the same server in order to allow for cryptographic information reuse.

3. EAP-BASED FAST AUTHENTICATION

We consider two EAP-based fast authentication schemes: namely, re-authentication and pre-authentication, and compare them to full authentication. All three schemes are described in details and their performance characterized using the authentication signaling and the EAP method latency metrics. The authentication signaling latency, l_{Sign}, is defined as the time elapsed between the sending of the first authentication message until the reception of the ACK for the last authentication message during the network entry. On the other hand, the EAP method latency, l_{EAP}, is the time elapsed between the sending of the EAP Start message until the reception of either the EAP SUCCESS or EAP FAILURE messages.

3.1 Full authentication

When using full authentication, the mobile device carries out a full EAP exchange with the corresponding EAP server in the backbone network. This exchange takes place using specific layer-2 authentication messages that transport the EAP information between the mobile node and the PoA, which in turn transmits these messages over the wired network to the EAP authenticator. This transmission is usually achieved by encapsulating the EAP contents in other protocols, such as RADIUS.

The distinctive characteristic of this authentication scheme is that there is no optimization performed over the specification of the EAP method. As we shall see in section 5, this leads to large handover delays, due to the number of messages exchanged between the peer and the server. The EAP message flow of a full authentication is shown in Fig. 2a. It involves the exchange of the EAP Start messages between the mobile node and the target PoA, the execution of the full EAP method exchange between the mobile node and the EAP server, and the distribution of the cryptographic key from the EAP server to the EAP authenticator and the PoA. This key distribution involves a Request / Response exchange between the EAP entities in the hierarchy, and

Digital Object Identifier: 10.4108/ICST.WICON2008.4814
http://dx.doi.org/10.4108/ICST.WICON2008.4814

Table 1: Notation used in the expressions of the security latency

m_{EAP}	Number of messages required for the execution of the EAP method, not including the EAP Start signaling
m_{Auth}	Number of messages in the authentication signaling of the network, not including the EAP exchange messages
c	Time required for all parties to perform all the cryptographic operations for the EAP exchange, including the key derivation
d_{MP_C}	Average propagation delay between the MN and the PoA in the current network
d_{MP_T}	Average propagation delay between the MN and the PoA in the target network
d_{PA_T}	Average propagation delay between the target PoA and its associated EAP Authenticator
$d_{A_CA_T}$	Average propagation delay between the current and target EAP Authenticators
d_{PS_T}	Average propagation delay between the target PoA and its EAP Server
$d_{P_CP_T}$	Average propagation delay between the current and target PoAs

ensures that every node gets and derives a set of keys, as defined by the EAP method in use. This whole exchange is equivalent in terms of delay to a Request / Response exchange between the target PoA and the EAP server. Each of these phases may involve processing delays, especially when the nodes have to perform cryptographic operations. This cryptographic delay is also considered for our characterization of the authentication latencies.

Considering all of these factors, the latency of the EAP method exchange is expressed as:

$$l_{EAP} = 2 \cdot d_{MP_T} + (d_{MP_T} + d_{PS_T}) \cdot m_{EAP} + 2 \cdot d_{PS_T} + c$$

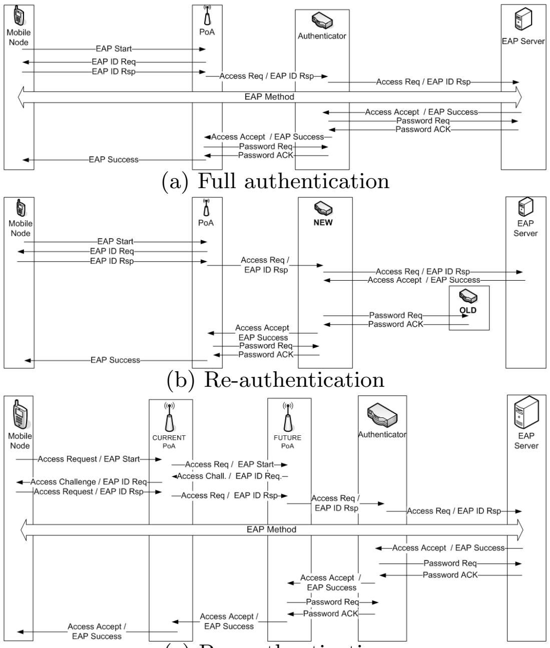
where $2 \cdot d_{MP_T}$ is the delay of the initial EAP Start message exchange; $(d_{MP_T} + d_{PS_T}) \cdot m_{EAP}$ is the time it takes to complete the full EAP exchange between the mobile device and the EAP server; $2 \cdot d_{PS_T}$ is the delay introduced by the key distribution and c is the total delay introduced by the cryptographic operations.

However, the EAP authentication is just one step in the network authentication process, which involves other media dependent operations. For example, in IEEE 802.11 networks the mobile node has to go through the Open Authentication, the Association and the Four-Way Handshake, while in IEEE 802.16 the only required additional step is the TEK transfer.

These operations include the exchange of request and response messages between the mobile node and the target PoA. Thus, the total latency for the authentication signaling phase is expressed as:

$$l_{Sign} = l_{EAP} + d_{MP_T} \cdot m_{Auth}$$

with $d_{MP_T} \cdot m_{Auth}$ being the delay for exchanging technology dependent authentication messages.



(a) Full authentication

(b) Re-authentication

(c) Pre-authentication

Figure 2: Message flows for the different authentication schemes

3.2 Re-authentication

Re-authentication is an optimization over full authentication, based on the reuse of information already exchanged between the EAP peer and server when performing a subsequent authentication with the same EAP server.

There are several proposals being considered for how to perform re-authentication, such as [3], [7], and [13]. In this study, we use the re-authentication scheme proposed in the IETF HOKEY working group in [13]. This scheme presents a minimum set of modifications to the original EAP framework, while providing a complete key sharing and derivation model that allows security services to be maintained during a handover. Although we have considered other re-authentication schemes in the course of our work such [7], they are omitted here due to a lack of space.

As shown in Fig. 2b, the EAP messages are reduced to the EAP Start, Identification (ID), and result (SUCCESS/FAILURE) messages exchanged between the EAP peer and the server. The result message also carries information about the new derived key and is propagated back to the peer through the authenticator and the PoA. A final message is exchanged between the old and the new authenticator in order to transfer the keys that the old authenticator got in the previous authentication. So, for re-authentication, the EAP latency becomes:

$$\begin{aligned} l_{EAP} = 2 \cdot d_{MP_T} + 2 \cdot (d_{MP_T} + d_{PS_T}) + \\ +2 \cdot d_{A_CA_T} + 2 \cdot d_{PA_T} + c_{Reauth} \end{aligned}$$

where c_{Reauth} is the total delay introduced by the cryptographic operations in this reduced EAP exchange. Since most of the cryptographic operations are eliminated, and the key derivation operations are simpler than computing new keys, we can say that $c_{Reauth} < c$. Also, we note that in the case of re-authentication the EAP method exchange $(2 \cdot (d_{MP_T} + d_{PS_T}))$ consists of only two messages, instead of the variable number (m_{EAP}) that we saw in the full authentication case. This makes re-authentication EAP method-independent.

The resulting signaling latency is the same as the one computed for the full authentication case:

$$l_{Sign} = l_{EAP} + d_{MP_T} \cdot m_{Auth}$$

Thus, reducing the number of messages exchanged and simplifying the cryptographic operations leads to a lower authentication signaling latency.

3.3 Pre-authentication

Pre-authentication is another optimization, which aims at reducing the resources and time needed for authentication during handovers. The main advantage of pre-authentication is that cryptographic material is not reused, hence it may be considered more secure. This also removes the constraint of having a common EAP server for both the current and target networks.

In order to make use of pre-authentication, the mobile device needs to perform a full EAP method exchange with the target network's EAP server via the target PoA well before a handover is initiated. To reach the target PoA the mobile node may use a layer-2 or layer-3 tunnel through the current PoA, depending on the network topology, as highlighted in Fig. 2c.

The EAP latency is computed as:

$$\begin{aligned} l_{EAP} = 2 \cdot (d_{MP_C} + d_{P_C P_T}) + \\ +(d_{MP_C} + d_{P_C P_T} + d_{PS_T}) \cdot m_{EAP} + \\ +2 \cdot d_{PS_T} + c + c_{PoA} \end{aligned}$$

In this case, the EAP latency is likely to increase compared to that of a full authentication, due to the additional hops the messages have to go through, and the additional cryptographic operations performed between the PoAs if they establish a secure tunnel to protect the information forwarding, represented as c_{PoA}.

Once this EAP message exchange is done, the mobile node is authenticated in the target network, so during the handover the EAP phase is skipped. However, all the remaining steps in the network entry phase are performed in order to effectively bind the authentication credentials to the mobile device. Given that, the authentication signaling latency is expressed as:

$$l_{Sign} = d_{MP_T} \cdot m_{Auth}$$

In this case, the authentication signaling latency during the network entry is significantly reduced. However, note that this improvement occurs only if the EAP method was successfully completed in advance, that is, before the mobile node lost the connection with the old network, or before the authentication with the new network starts, whichever happens first. If the pre-authentication phase is not completed before the handover starts, then the signaling latency of a full authentication applies. Note that the notation used in the analysis is summarized in Table 3.

4. SIMULATION ENVIRONMENT

Table 2: Simulation parameters

802.11 coverage area radius	50 m
802.16 coverage area radius	500 m
Key lifetime	More than the simulation time
Size of Diffie-Hellman keys	1024 bits
Size of symmetric keys	128 bits
Size of EAP IDs	64 bytes
Application traffic type	Constant Bit Rate
Application traffic rate	640 kb/s
Application traffic bursts	32 kb every 50 ms
Handover optimization	None

The simulation environment we use is based on the NS-2 [14] simulator, with extensions for the IEEE 802.16 model [15], and the authentication schemes described previously. For this study we will make use of IPv6 ([5]) stateless address auto-configuration, defined in [17].

In addition to the performance metrics defined previously, namely, the EAP and the authentication signaling latencies, we define two new metrics as follows. Let the **handover delay** be the time elapsed between the decision to switch to a new network and the traffic redirection to the new interface. Similarly, we define the **cryptographic processing delay** as the time used by the mobile node to perform cryptographic operations related to the authentication.

The parameters used in the simulations are shown in Table 4, and the network topology is based on the one shown in Fig. 1, with the following parameters:

- There is an 802.11 network and an 802.16 network, which share the EAP server.
- A common router between both networks and the backbone also hosts the EAP authenticator for both EAP NAS installed in both PoAs.
- The EAP server is connected to the EAP authenticator by a link with a 5 ms propagation delay.
- The link between the EAP authenticator and each of the PoAs have a 15 ms propagation delay.
- The mobile node starts either in the IEEE 802.11 network and moves towards the IEEE 802.16 network, or conversely starts in the IEEE 802.16 network and moves towards the IEEE 802.11 network.
- The application traffic is sent from the EAP server to the mobile device.

Additionally, in order to provide the simulator with realistic information about the cryptographic performance, we used as an example a Palm Tungsten T3[1] to obtain the time required to compute all the cryptographic operations. Although the performance results of the cryptographic processing are dependent on the choice of platform used, and may therefore vary from system to system, the

[1] This device was used as an example platform, and its use in this research does not constitute an endorsement by National Institute of Standards and Technology

Digital Object Identifier: 10.4108/ICST.WICON2008.4814
http://dx.doi.org/10.4108/ICST.WICON2008.4814

relative gain or loss in performance between different authentication schemes remains valid for any platform choice.

Regarding the simulation parameters, a topic which deserves special attention is the EAP methods used, as their complexity, security level and number of messages exchanged vary greatly from one method to another.

EAP Methods

Our results have been obtained using two different EAP methods, with different levels of complexity: the *EAP Generalized Pre-Shared Key* (EAP-GPSK, [4]) and the *EAP Tunneled TLS Authentication Protocol version 1 with MD5 challenge-response authentication* (EAP-TTLSv1-MD5, [6]). EAP-GPSK is defined in an Internet draft, and aims at providing a lightweight EAP method with well-defined security requirements (such as mutual authentication), while keeping it as simple and efficient as possible. The resulting EAP method only requires four messages to be exchanged between the EAP server and the EAP peer.

EAP-TTLSv1-MD5 is more complex. First, the EAP server and the EAP peer establish a secure tunnel using a TLS-like message exchanges. Once the tunnel is ready, both entities perform an EAP authentication using a different method (in this case, MD5-Challenge, which only requires two messages to be sent). Besides the added complexity of the additional messages and the tunneling of one EAP method within a secure tunnel negotiated with a different EAP method, the cryptographic operations are more complex than those of GPSK.

By using these two different EAP methods our goal is to gain additional insights on how different authentication schemes deal with different levels of complexity and cryptographic processing. Additionally, as the different available EAP methods differ greatly regarding their complexity and performance, by using these two EAP methods we aim to study the behavior of a "fast" method (GPSK) and a "slow" one (TTLS-MD5). Thus, the results obtained are orientative of the results we would get using other EAP methods.

5. SIMULATION RESULTS

In this section we analyze the performance of the authentication schemes, using the simulation environment we defined previously. The results represent averages obtained over 100 simulations, with the associated standard deviations. All latencies are given in milliseconds, unless stated otherwise.

5.1 Authentication Signaling

The time required for the authentication signaling (as defined in section 3) with each of the authentication schemes is shown in Table 5.1 for the EAP-GPSK method, and Table 5.1 for the EAP-TTLSv1-MD5[2].

There are several points to observe in the results. First, it is important to note that both the re-authentication and pre-authentication schemes constitute a significant improvement (70 % or more) over the full authentication case, regardless of the EAP method chosen. Second, we observe the same differences uncovered by the analysis in section 3. Since for re-authentication, the EAP message exchange is significantly reduced, this leads to lower EAP and authentication signaling latencies. In the case of pre-authentication, although the EAP message exchange may take longer to complete, it is carried out before a handover is started. As a result, the authentication signaling latency during the handover is even lower than in the case of re-authentication.

Table 3: Authentication Signaling latency with EAP-GPSK (ms)

802.11 Security	**Full Auth.**	**Re-Auth.**	**Indirect Pre-Auth**
Signaling Latency	194.33 ±0.672	46.59 ±0.510	7.44 ±0.371
EAP Latency	192.47 ±0.608	45.07 ±0.417	422.42 ±0.136
802.16 Security	**Full Auth.**	**Re-Auth.**	**Indirect Pre-Auth**
Signaling Latency	235.42 ±0.013	70.42 ±0.001	10.42 ±0.171
EAP Latency	226.37 ±0.001	61.37 ±0.001	422.42 ±0.136

Table 4: Authentication Signaling latency with EAP-TTLSv1-MD5 (ms)

802.11 Security	**Full Auth.**	**Re-Auth.**	**Indirect Pre-Auth**
Signaling Latency	31352.37 ±0.751	46.59 ±0.450	7.44 ±0.371
EAP Latency	31350.49 ±0.705	45.07 ±0.395	31802.67 ±0.366
802.16 Security	**Full Auth.**	**Re-Auth.**	**Indirect Pre-Auth**
Signaling Latency	31445.42 ±0.001	70.42 ±0.014	10.42 ±0.171
EAP Latency	31436.18 ±0.001	61.37 ±0.001	31892.35 ±0.366

Also, in Table 5.1 we can see that when using pre-authentication the mobile node has to start the EAP message exchange more than 30 seconds prior to the start of a handover in order to complete it in time. In some situations, this may be hard to achieve since it is not always possible to decide which target network to use that much time in advance. On the other hand, by comparing the re-authentication data in Table 5.1 and Table 5.1 we can see that, as we mentioned previously, when using re-authentication the latencies are not affected by the choice of the EAP method. This means that when using pre-authentication the mobile node has to decide which network it will move to well in advance, so it has enough time to perform the entire EAP message exchange, which can be time consuming.

5.2 Cryptographic Processing Delay

As cryptographic operations are usually computationally extensive, it is important to consider how these operations may be reduced when using either authentication scheme. The cryptographic delay is defined as the time spent by the mobile node to perform different cryptographic operations during the authentication phase. The values of this delay when using each of these schemes can be seen in Table 5.2 (for EAP-GPSK) and Table 5.2 (for EAP-TTLSv1-MD5).

[2] Note that the computation of the Diffie-Hellman agreement in TTLS took 30813 ms on the Tungsten platform chosen

Table 5: Cryptographic delay with EAP-GPSK (ms)

802.11	**Full Auth.**	**Re-Auth.**	**Indirect Pre-Auth**
EAP latency	192.47 ±0.650	45.07 ±0.498	192.47 ±0.965
Crypto. delay	17.48 ±0.1	1.02 ±0.001	17.48 ±0.1
802.16	**Full Auth.**	**Re-Auth.**	**Indirect Pre-Auth**
EAP latency	226.37 ±0.011	61.37 ±0.001	226.37 ±0.011
Crypto. delay	17.48 ±0.1	1.02 ±0.001	17.48 ±0.1

Table 6: Cryptographic delay with EAP-TTLSv1-MD5 (ms)

802.11	**Full Auth.**	**Re-Auth.**	**Indirect Pre-Auth**
EAP latency	31350.49 ±0.722	45.07 ±0.438	31350.49 ±1.215
Crypto. delay	30884.22 ±0.708	1.02 ±0.001	30884.22 ±1.203
802.16	**Full Auth.**	**Re-Auth.**	**Indirect Pre-Auth**
EAP latency	31436.18 ±0.001	61.37 ±0.001	31436.18 ±0.121
Crypto. delay	30884.22 ±0.708	1.02 ±0.001	30884.22 ±1.203

These results are consistent with the authentication schemes analysis presented in section 3. Since pre-authentication does not shorten or eliminate any step in the authentication process, the cryptographic delay is the same as in the case of a full authentication. However, in the case of re-authentication, there are less operations carried out, and thus, the cryptographic processing delay is reduced. This may be an important consideration in the context of the resources available on the mobile device.

As we mentioned earlier, we have to note that the use of pre-authentication may increase the number of cryptographic operations performed, although we have not considered this issue in the simulation models. This would be due to the establishment of a secure tunnel between the mobile device and the target PoA, in order to securely transmit the EAP messages through the backbone network and the current PoA.

6. HANDOVER IMPACT

Our next step is the study of the impact of these authentication schemes on the total handover time. The handover delay represents the time elapsed between when a decision to handover is executed until the traffic is redirected to the new interface. We consider a base case with no handover optimization whatsoever, to move later onto optimized handovers that use the link triggers defined in IEEE 802.21.

Table 7: Handover delay without optimizations (ms)

802.11	**Full Auth.**	**Re-Auth.**	**Indirect Pre-Auth**
GPSK	12190.13 ±143.712	11997.84 ±139.682	11957.87 ±140.989
TTLSv1-MD5	43364.81 ±128.436	11997.84 ±137.051	43364.81 ±128.436
802.16	**Full Auth.**	**Re-Auth.**	**Indirect Pre-Auth**
GPSK	13194.64 ±110.118	13024.64 ±105.15	12094.64 ±112.482
TTLSv1-MD5	44399.64 ±115.285	13024.64 ±116.460	44399.64 ±115.285

6.1 Handover without optimizations

First of all, we will analyze the handover performance without using any optimization. In this case, the mobile node first has to detect that the current link is down, in order to start looking for a new one. This happens when the current network prefix expires without having received a new Router Advertisement message from the current Access Router (in this case, our PoA). When this event happens, the device will start looking for a new network in all the available interfaces, selecting the best available network, performing the network entry (including the authentication messages), and waiting for a Router Advertisement message that will provide the network prefix to use. Finally, the traffic will have to be redirected to the new address.

In our simulations, the Router Advertisement messages are sent every 2 seconds, and the information they carry expires after 18 seconds. With this configuration, the handover delay is shown in Table 7.

We observe that re-authentication reduces the handover delay in a predictable way, regardless of the EAP method used, since the actual message exchange of the method is omitted. Also pre-authentication reduces the handover delay as long as the EAP message exchange is completed before the handover starts. We can see these two situations with EAP-GPSK, which completes before a handover, and EAP-TTLSv1-MD5, which does not complete before the handover starts.

6.2 Handover with link triggers

Next in our analysis is the performance of the handover using link triggers. These triggers notify the mobile node of the loss of connectivity in the current link (*Link Down (LD)* trigger), the establishment of a connection in a given interface (*Link Up (LU)* trigger), and the progressive loss of the current link (*Link Going Down (LGD)* trigger). This trigger notifies the mobile node that the current connection on the given interface is about to be lost. This prediction can be made based on parameters such as the signal strength, topology configuration, quality of service parameters, etc.... With this notification the mobile node can start looking for new networks on other interfaces, in order to have a new link ready by the time the current link goes definitely down. In Figure 3 we can see how the different events are generated as a mobile node moves from one network to another: when the signal first goes under the LGD threshold, a LGD event is generated; if the signal continues decaying, it will eventually

Digital Object Identifier: 10.4108/ICST.WICON2008.4814
http://dx.doi.org/10.4108/ICST.WICON2008.4814

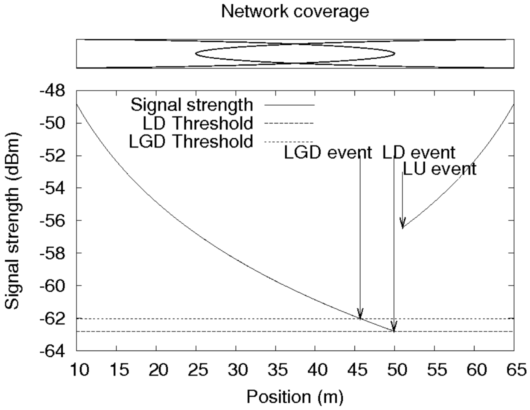


Figure 3: Generation of link events

go under the reception threshold, which will trigger a LD event; finally, as the mobile node eventually connects to the new network, we can see that a LU event is generated.

In our simulations the threshold to generate a LGD event has been configured so that the mobile node will have enough time to fully complete the entry in the new network using full authentication. This means that, while the LGD event is generated when the signal strength falls under the 90 % of the carrier sense level if we are using EAP-GPSK, when using EAP-TTLSv1-MD5 this parameter has been adjusted to the 20 % of the signal strength. If this parameter wasn't adapted to the required handover time, our results would show that the disruption time is still quite significant, althought it may have been reduced partially. Thus, there is an interesting area of research in the mechanisms that the mobile node may use to properly estimate this handover time to dynamically adjust the hreshold level.

With the information these triggers provide, the mobile node does not have to rely on timeouts or periodic notifications to learn that the connection in a given interface is up or down. Consequently, in the handover the mobile node can start looking for available networks right away, and after performing the network entry it doesn't have to wait for the periodic Router Advertisement, and can instead request the needed information sending a Router Solicitation message.

With this configuration the observed handover delays in our simulations are shown in Table 8. It is important to note the difference between the Handover delay and the disruption time (shown in Table 9) when using link triggers, as the mobile node starts performing the handover while it is still connected to the old network.

We can see that the disruption time is always 0, as we adjusted the threshold of the LGD event to allow for the full execution of the network entry procedure in the time it took the mobile node to lose the connection. However, with the time required to perform the TTLSv1 authentication, this may lead to unreal threshold values that will make the mobile node to start searching for new networks even when close to the PoA of the current network. Other methods for generating the Link Going Down events (such as the proposed in [18]) may also prevent this problem by predicting beforehand the required time and adjusting the threshold accordingly.

Table 8: Handover delay using link triggers (ms)

802.11	**Full Auth.**	**Re-Auth.**	**Indirect Pre-Auth**
GPSK	920.17 ±1.224	719.23 ±1.151	677.48 ±0.925
TTLSv1-MD5	32080.23 ±1.998	719.23 ±1.151	32080.23 ±1.98
802.16	**Full Auth.**	**Re-Auth.**	**Indirect Pre-Auth**
GPSK	1158.64 ±1.205	995.64 ±1.102	935.64 ±1.012
TTLSv1-MD5	32342.64 ±1.176	995.64 ±1.102	32342.64 ±1.023

Table 9: Disruption time using link triggers (ms)

802.11	**Full Auth.**	**Re-Auth.**	**Indirect Pre-Auth**
GPSK	0 ±0	0 ±0	0 ±0
TTLSv1-MD5	20080.23 ±1.998	0 ±0	20080.23 ±1.98
802.16	**Full Auth.**	**Re-Auth.**	**Indirect Pre-Auth**
GPSK	0 ±0	0 ±0	0 ±0
TTLSv1-MD5	20342.64 ±1.176	0 ±0	20342.64 ±1.023

As we can see from the results shown in this section, the use of link triggers improves significantly the handover delay and disruption time in heterogeneous handovers. However, these improvements also make the authentication become a more relevant part in the handover process. By comparing the authentication signaling time and the handover delays, we can see that the more optimized the handover, the more relevant the authentication signaling delay is. For example, when using GPSK, a full authentication only represents a mere 1.59 % of the handover delay if we don't use lik triggers, but this value rises to more than 21 % when using link triggers. In the case of TTLSv1 the results are much more significant, as a full authentication requires a 72.3 % of a non-optimized handover and over 97.5 % of an optimized one. However, using re-authentication reduces this values to just 0.39 % and 6.48 % respectively, regardless of the EAP method used.

This variable relevance of the authentication signaling is important, as we have mentioned how the lack of optimization in the authentication can prevent these handover improvements to provide the gain (disruption time, quality of service, etc. . .) expected unless the handover parameters

Digital Object Identifier: 10.4108/ICST.WICON2008.4814
http://dx.doi.org/10.4108/ICST.WICON2008.4814

are configured in ways that may damage the usability of the mobile node. Thus, handover enhancements and improvements have to also take into account the authentication as an important phase of the network entry.

Furthermore, we can see a tight interaction between link triggers and the authentication, as timely triggers will provide enough time for the mobile node to perform the network entry, regardless of the authentication technique used. Furthermore, the link triggers may prevent initiating the authentication procedure in networks that will not be visited by the mobile node (e.g., when using Pre-authentication, the mobile node wouldn't have to authenticate to every network the mobile node goes through, just those to which it actually connects).

7. CONCLUSIONS

The emergence of mobile and heterogeneous network environments is making roaming between networks more common. However, according to the results obtained in our simulations, many applications and environments cannot afford the service disruptions that typically occur when a full non-optimized authentication process is required. Therefore, it is necessary to consider optimizations to network authentication in order to expedite the authentication and achieve a true seamless mobility experience. In this article, we compare the performance of re-authentication and indirect pre-authentication showing that both proposals have several advantages over full authentication.

Re-authentication reduces the number of messages exchanged and cryptographic operations, while pre-authentication may provide the best optimization of the authentication signaling delay during the handover. Both schemes can be used independently and at different times in the same topology depending on the needs and constraints of the scenario considered.

Optimized handovers can benefit from these techniques, as the cost of performing full authentications is too high, even when "fast" methods are used. Correspondingly, the authentication techniques can benefit from the various handover optimization mechanisms in order to improve their performance.

8. REFERENCES

[1] 3gpp technical specification 33.102. 3g security - security architecture v7.1.0. Technical report, 3GPP, Dec. 2006.

[2] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz. RFC 3748: Extensible authentication protocol (eap), June 2004. Status: PROPOSED STANDARD.

[3] T. Aura and M. Roe. Reducing reauthentication delay in wireless networks. In *SECURECOMM '05: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, pages 139–148, Washington, DC, USA, 2005. IEEE Computer Society.

[4] T. Clancy. Eap generalized pre-shared key (eap-gpsk) method, internet draft, July 2008. Status: DRAFT.

[5] S. Deering and R. Hinden. RFC 2460: Internet protocol, version 6 (ipv6) specification, Dec. 1998. Status: DRAFT STANDARD.

[6] P. Funk and S. Blake-Wilson. Eap tunneled tls authentication protocol version 1 (eap-ttlsv1), internet draft, Mar. 2006. Status: DRAFT.

[7] A. Ganz, S. H. Park, and Z. Ganz. Robust re-authentication and key exchange protocol for IEEE 802.11wireless LANs. In *Military Communications Conference, 1998. MILCOM 98. Proceedings., IEEE*, volume 3, pages 1018–1022, Boston, MA, USA, Oct. 1998.

[8] Draft ieee 802.11r-2008. ieee draft for local and metropolitan area networks- part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. amendment 2: Fast bss transition, Jan. 2008.

[9] Ieee std 802.16e-2006. ieee standard for local and metropolitan area networks- part 16: Air interface for fixed and mobile broadband wireless access systems. amendment 2: Physical and medium access control layers for combined fixed and mobile operation in licensed bands and corrigendum 1, Feb. 2006.

[10] Draft ieee 802.21-2008. ieee standard for local and metropolitan area networks- part 21: Media independent handover services, June 2008.

[11] D. Johnson, C. Perkins, and J. Arkko. RFC 3775: Mobility support in ipv6, June 2004. Status: PROPOSED STANDARD.

[12] R. Koodli. RFC 5268: Mobile ipv6 fast handovers, June 2008. Status: PROPOSED STANDARD.

[13] V. Narayan and L. Doneti. Eap extensions for eap re-authentication protocol (erp), internet draft, Mar. 2008. Status: DRAFT.

[14] Ns-2 simulator, version 2.31, Mar. 2008. Retrieved on March 2008.

[15] N. I. of Standards and Technology. Seamless and secure mobility project, Mar. 2008.

[16] Y. Ohba. Eap pre-authentication problem statement, internet draft, June 2008. Status: DRAFT.

[17] S. Thomson, T. Narten, and T. Jinmei. RFC 4862: Ipv6 stateless address autoconfiguration, Sept. 2007. Status: DRAFT STANDARD.

[18] S.-J. Yoo, D. Cypher, and N. Golmie. LMS predictive link triggering for seamless handovers in heterogeneous wireless networks. In *Military Communications Conference, 2007. MILCOM 2007. IEEE*, pages 1–7, Orlando, FL, USA,, Oct. 2007.

Digital Object Identifier: 10.4108/ICST.WICON2008.4814
http://dx.doi.org/10.4108/ICST.WICON2008.4814