Study and Evaluation of the MOBIKE Protocol as a Mobility Solution for Femtocell Networks[1]

Patricia Noriega-Vivas, Celeste Campo, Carlos García-Rubio
Department of Telematic Engineering
University Carlos III of Madrid
Email: {pnoriega, celeste, cgr}@it.uc3m.es

***Abstract*—Femtocells can be used to improve the indoor coverage and bandwidth of 3G cellular networks in homes and buildings. They are connected to the core network by means of a fixed broadband connection such as DSL or cable, thus they have been designed to be placed in a fixed location. Public transportation systems could attract more passengers by setting up femtocells, however the currently defined architecture does not support mobility. In this paper first we identify what modifications are needed on the existing femtocell architecture and then we propose improvements to support mobility. Moreover, we present the MOBIKE protocol as a mobility solution for femtocell networks and create a testbed to evaluate the handover performance in terms of handover delay and packet losses under several handover scenarios.**

***Index Terms*—femtocell architecture, mobile femtocell, MOBIKE, IKEv2, IPsec.**

I. Introduction

Femtocells are small, low-cost and low-power cellular base stations, typically designed for use in a home or small business (e.g., a holiday cottage) to improve indoor coverage and bandwidth, and also to off-load traffic from the existing macrocell network [2]. Nowadays, femtocells are usually deployed by the customers, they have a fixed location (i.e., they do not move), and they always connect to the 3G core network using a ciphered IP tunnel through the Internet connection provided by a Digital Subscriber Line (DSL) or cable router. However, femtocells could also be interesting in other scenarios.

Trains, buses or trams could provide faster data speeds and better user experience to theirs passengers setting up femtocells. Moreover, providing Internet access to passengers more travelers could be attracted due to the possibility of being able to get the most of their journey time, using their mobile phones and computers.

In recent years some solutions have been proposed to support Internet connectivity on trains. Most of them are focused on using a single technology to connect the train to the outside. For instance, Aguado *et al.* [3] propose a network architecture based on WiMAX since it is able to support mobility at speeds up to 500 km/h. Other approaches are based on satellite architectures, IEEE 802.11 or emerging standards as IEEE 802.20. Most solutions provide a single access terminal per train and therefore the connection is shared between passengers. Latter architectures are efficient since they avoid many users performing simultaneously handover procedures.

There are also other approaches focused on using several technologies instead of using single-based architectures. For example, Rodríguez *et al.* [4] argue that a more efficient approach is to use a multitude of wireless technologies simultaneously and they present a system to provide Internet access on trains by combining several wireless interfaces.

The work we present in this paper is focused on supporting mobility on femtocell networks by suppressing the original fixed interface and setting a pool of heterogeneous wireless interfaces in its place. To reach this objective, it will be necessary to provide mechanisms to perform handovers between technologies ensuring thus continuity of service to the users. Hence, handovers between different technologies (inter-handover) are expected but also between interfaces of the same technology (intra-handovers). In addition, different Internet service providers could be used to obtain redundant links and thus reliable systems.

This paper proposes some modifications to the existing femtocell architecture. On one hand, we propose to include a new message for the existing Home Node B Application Part (HNBAP) protocol to help in the mobility management. On the other hand, we propose the use of the MOBIKE protocol to support mobility between femtocells and the operator's core network. Moreover, we evaluate its feasibility by creating an experimental testbed which simulates a scenario where trains have femtocells embedded. As far as the authors know, there is no proposal to support Internet connectivity on trains by using femtocells.

The rest of this paper is structured as follows. Section II summarizes the state of the art related to this work. The femtocell architecture and the main protocols implemented by them are included. Next in Section III, we propose some changes on the femtocell architecture to support mobility and also the use of MOBIKE in several network entities to manage the tunnel reestablishment and thus achieve continuity of service. Then in Section IV we create a testbed to evaluate the feasibility of MOBIKE under several handover scenarios and afterwards in Section V we discuss the results obtained. In Section VI the conclusions and future work are drawn. Finally, in the Appendix we describe the limitations encountered on the tools employed to develop and test the experiment.

[1] This paper is an extended version of the preliminary work presented at the "The Second International Conference on Access Networks (ACCESS 2011)" on a specific femtocell track [1] in June 2011.

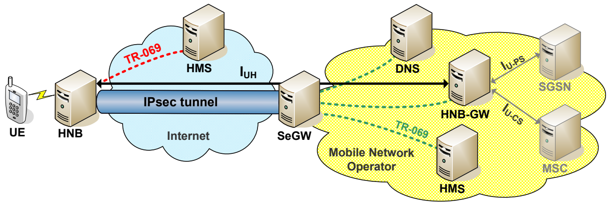


Fig. 1. 3G femtocell network overview

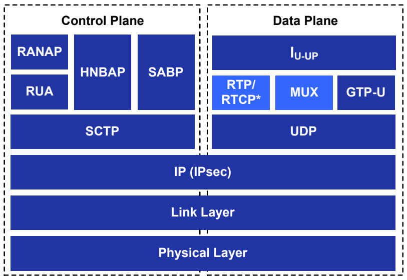


Fig. 2. I_{UH} protocol stack

II. Femtocells and Mobility

In this section we review the state of the art related with the operation of femtocells and the protocols they implement. First, the elements and interfaces for 3G femtocell networks are introduced. Then we present the IKEv2 protocol, employed by the femtocells and the core network to establish and maintain the IPsec tunnel between them and MOBIKE, an extension to IKEv2 to support mobility, which is currently not used in femtocells.

A. Femtocell architecture

Femtocells are envisaged to be deployed by home and enterprise users at their premises. This new deployment model requires the new network architecture depicted in Figure 1.

In this section, we briefly describe the new network elements and interfaces introduced by 3GPP in [5] for 3G femtocell networks.

1) Home Node B (HNB): The HNB is the femtocell. It serves User Equipment (UE) traffic by means of the UMTS U_U interface [6], and sends it to the core network through the I_{UH} interface. It contains part, or all the functionality normally associated to a Radio Network Controller (RNC) and supports HNB and UE registration procedures over the I_{UH} interface.

2) Home Node B Gateway (HNB-GW): This element terminates the I_{UH} interface and acts as a concentrator to aggregate a large number of HNBs. It is seen as a RNC by the core network which communicates with it using the existing I_{U-CS}, I_{U-PS} interfaces [6].

3) I_{UH} interface: This interface connects the HNB with the HNB-GW. It defines two new protocols in the control plane to address the differences between HNBs and the original I_U interface:

- Home Node B Application Part (HNBAP) [7]: it provides functions for registering UEs and HNBs into the network, error handling and group management.
- RANAP User Adaptation (RUA) [8]: it provides the signaling service between HNB and HNB-GW in the control plane. It is used to send RANAP messages in a transparent way. It also provides error handling functions.

Figure 2 shows the protocol stack defined for the new I_{UH} interface.

4) HNB Management System (HMS): The HMS is a management server that configures the HNB according to the operator's policy. It is composed of a TR-069 manager [9] and a file server. The Broadband Forum [10] in collaboration with Femto Forum [11] have defined the TR-196 [12] technical report, an extension to the TR-069 protocol that incorporates a new data model for femtocells.

The HMS performs location verification and assigns local access information to the femtocells when they are powered up. This information includes the entities needed by the femtocell to provide service: the Serving Security Gateway (S-SeGW), the Serving HMS (S-HMS) and optionally the HNB-GW.

The HMS may be located inside the operator's core network (accessible on the Intranet) or outside of it (accessible on the Internet). 3GPP defines two kind of HMS:

- Initial HMS (I-HMS): it may provide location verification and assign the S-HMS, S-SeGW and optionally HNB-GW to the HNB.
- Serving HMS (S-HMS): it has new functions such as performance and fault updates, and assigns the HNB-GW during the HNB registration procedure if the I-HMS did not provide it.

5) Security Gateway (SeGW): It terminates the IPsec tunnel established with the HNB, provides mutual authentication, encryption, data integrity and access to the HNB-GW and S-HMS (and also the I-HMS if it is accessible on the Intranet). After successful mutual authentication between the femtocell and SeGW, the SeGW connects the femtocell to the operator's core network and any connection between the femtocell and the core network is tunnelled through this entity.

The SeGW is a logically separated entity and it can be implemented as a separate physical element or into others such as the HNB-GW. 3GPP defines two kind of SeGW:

- Initial Security Gateway (I-SeGW): its URL may be factory programmed in the HNB to allow the establishment of the IPsec tunnel with the I-HMS.
- Serving Security Gateway (S-SeGW): it terminates the IPsec tunnel and implements a forwarding function to inject IP packets into the mobile network operator (Intranet) that allows the communication with the HNB-GW, S-HMS and other network elements.

B. IKEv2

IPsec [13] is a protocol suite for securing IP communications by authenticating and encrypting each IP packet of a session. The IPsec architecture uses the concept of Security Association (SA), a one-way connection that provides security services for the traffic carried by it using the Encapsulating Security Payload (ESP) [14] or Authentication Header (AH) [15] protocols.

IKEv2 [16] is a component of IPsec. It is used to perform mutual authentication between two parties, to establish and to maintain dynamically SAs for ESP or AH protocols. Several scenarios are included by IKEv2, however we focus in one particular case: when an endpoint is connected with a security gateway using the tunnel mode of IPsec, since this is the scenario deployed by femtocell networks on the I_{UH} interface.

According with the IKEv2 terminology, we use the term “initiator” to refer the party who initiates the first IKE security association and “responder” to the other peer.

Figure 1 shows the scenario mentioned above. The HNB (or femtocell) is connected through an IPsec tunnel with the SeGW that is located within the mobile network operator. All traffic generated by the femtocell (user data and control packets) is securely protected by IPsec. The SeGW receives that traffic and then forwards through the mobile network operator without protection.

The IKEv2 protocol uses request/response pairs and every pair is called *exchange*. The first exchange in an IKEv2 session is the IKE_SA_INIT in which security parameters for the IKE SA are negotiated. If this exchange is completed, the second exchange, IKE_AUTH, will try to set up a SA for the ESP or AH protocols. These exchanges are known as Phase 1 of IKEv1.

Nonetheless, peers involved in an IKEv2 session may desire to transmit control messages to each other in order to report notifications or errors. To reach this behavior, IKEv2 defines an INFORMATIONAL exchange that only can be sent after the initial exchanges. Hence every message sent at this point is cryptographically protected with the negotiated keys.

Messages that belong to the INFORMATIONAL exchange contain zero or more Notify, Delete and Configuration payloads. They have to be confirmed sending some response to the sender, even with an empty message. Otherwise the sender will assume that the message has been lost in the network and will retransmit it.

The Notify Payload is used to transmit informational data such as state information or error conditions (e.g., specify why a SA could not be established). Every type of message has a concrete value specified within the Notify Payload. However, IANA [17] reserves value ranges for future use.

Some reserved values have been used to create extensions to IKEv2 and thus provide new capabilities. For instance, in RFC 5685 [18] it is defined a “Redirect Mechanism for IKEv2” that allows a VPN gateway that is overloaded or it is being shut down for maintenance to redirect a client to attach another gateway. Another interesting extension to IKEv2 using Notify payloads is MOBIKE [19], [20] and it is presented in the next section.

C. MOBIKE

IKEv2 itself does not provide any mobility support. For this purpose MOBIKE is defined as an extension to the existing IKEv2 protocol to provide secure mobility. This is why it has a similar behavior as IKEv2 regarding to the message exchange: responses are sent to the same address and port from which the request was received.

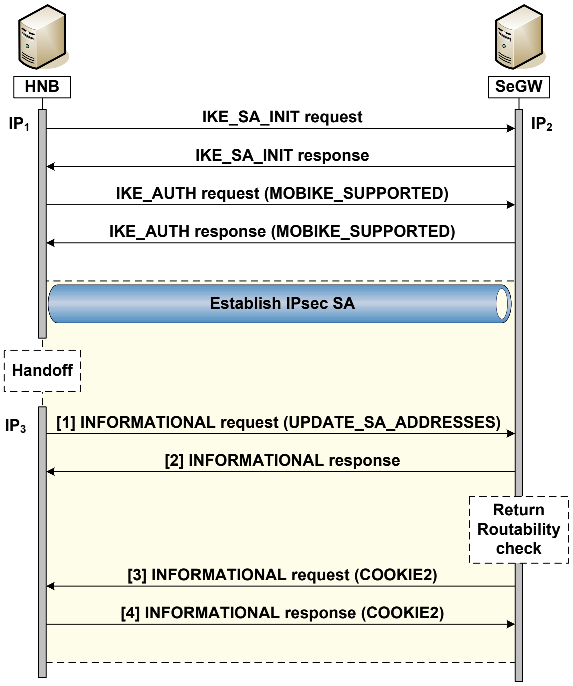

HNB
SeGW
IP1
IP2
IKE_SA_INIT request
IKE_SA_INIT response
IKE_AUTH request (MOBIKE_SUPPORTED)
IKE_AUTH response (MOBIKE_SUPPORTED)
Establish IPsec SA
Handoff
IP3
[1] INFORMATIONAL request (UPDATE_SA_ADDRESSES)
[2] INFORMATIONAL response
Return Routability check
[3] INFORMATIONAL request (COOKIE2)
[4] INFORMATIONAL response (COOKIE2)

Fig. 3. MOBIKE handover procedure

MOBIKE can update the IP addresses associated with an IPsec SA using an internal API (usually PF_KEY [21]) that provides access to the Security Association Database (SAD) and the Security Policy Database (SPD). The IP addresses associated with IKE and IPsec SAs can be changed without need for disconnecting the existing SAs and establishing new ones. Hence continuity of service can be achieved. Moreover, the applications running inside the MOBIKE-controlled IPsec tunnel might not detect the movement since their IP addresses remain constant. However, it only covers the tunnel mode of IPsec.

Although messages involved in the handover procedure are protected by the keys negotiated in the first IKEv2 exchange, updating the IP address of an IPsec tunnel lead to new security considerations. To address them, MOBIKE includes two new features: the “return routability check” allows a peer can verify if the other party IP address is available and therefore can receive packets. Conversely with “NAT prohibition” it is assured that IP addresses have not been modified by intermediate hosts such as NATs or translation agents.

MOBIKE allows a peer to have several IP addresses, e.g., a road-warrior with different wireless interfaces such as UMTS, WiMAX or Wi-Fi. However, the decision of which IP address is used for a SA is made by the initiator and how it makes that decision is beyond of the scope of this protocol.

Furthermore, it provides multihoming features to allow traffic movement between different network interfaces if for instance, the one that is being used stops working. However, it does not support load balancing between interfaces or simultaneous mobility, i.e., just one side can move (the initiator).

1) Mobility support: Figure 3 shows an example of the MOBIKE operation in a mobility environment. The initiator host is a mobile node and has two wireless interfaces which may be or not of the same technology. The responder host, on the contrary, is a fixed node and has only one interface reachable by IP_2.

To start the communication, first the initiator establishes an IKE SA with the responder by means of the IKE_SA_INIT exchange. These messages negotiate cryptographic algorithms, exchange nonces and do a Diffie-Hellman exchange to generate keying material for the IKE SA. Thereafter, all messages are encrypted by the session key.

Next, in the IKE_AUTH exchange, peers transmit their identities, prove knowledge of the secrets and finally establish the first CHILD_SA for the ESP protocol. Three notify payloads are introduced by MOBIKE to be used in the IKE_AUTH exchange. The `MOBIKE_SUPPORTED` payload to inform the other peer that supports MOBIKE; the `ADDITIONAL_IP_ADDRESSES` which contains an available IP where the sender can also receive packets and the `NO_ADDITIONAL_ADDRESSES` to inform that there isn't any additional IP available.

In the example, the initiator has obtained two IP addresses through both interfaces, IP_1 and IP_3. However, its preferred address is IP_1 and it reports this information to the responder by initiating the connection from that IP.

Then, in the IKE_AUTH exchange, both peers announce they support MOBIKE by including the `MOBIKE_SUPPORTED` payload. Also they may inform that they have available secondary addresses by sending an `ADDITIONAL_IP_ADDRESSES` payload for each one. Thus the initiator sends one payload that contains IP_3 while responder includes the `NO_ADDITIONAL_ADDRESSES` since it is a single-homed host.

Finally, the IKE and IPsec SAs have been established between IP_1 and IP_2 and two new entries are created in both SADs (one for the uplink and one for the downlink) including the source and destination IP and some other useful parameters such as the protocol used (ESP), the Security Parameter Index (SPI) or the SA lifetime. Then, protected data traverse the IPsec SAs.

During the session, the initiator peer may want to change its IP address due to mobility or some failure. To that end, MOBIKE defines two notify payloads to be used within the INFORMATIONAL exchange.

The `UPDATE_SA_ADDRESSES` notify payload is sent from an available IP to set it as the preferred address and it does not convey any data. The receiver does not need to know in advance that address to perform the handover, i.e., it is not needed to associate it previously. This payload can be sent only by initiators since the MOBIKE mobility support does not provide a "rendezvous" mechanism, i.e., only one peer can move (the initiator) while the other is considered stable. As the initiator is the one that changes its point of attachment, it is the peer responsible to decide when (and how) perform a handover procedure.

The `COOKIE2` payload is used to ensure that responders cannot generate the correct response for a concrete handover request without seeing it. For that purpose, initiators send a random value to responders that they must send back in the response. If both values do not match, the IKE SA must be closed.

Now, let us assume that the initiator peer of our example decides to perform a handover to its secondary address, IP_3, and also it requires the return routability check before updating the IPsec SAs. The handover procedure would proceed as follows: first, the initiator updates the IKE SA with the new address (IP_3) and sends the `UPDATE_SA_ADDRESSES` payload from it. When the responder receives that message, it checks the new address pair (IP_2, IP_3) according to the local policy and if that pair is acceptable, it updates the IKE SA and then sends a reply including the same `COOKIE2` payload sent by the initiator. Otherwise, it replies with a message containing the `UNACCEPTABLE_ADDRESSES` payload, also defined by MOBIKE. Once the return routability check is completed, both initiator and responder update the IPsec SAs stored in their SADs and traffic is conveyed between IP_2 and IP_3.

III. Changes to femtocells

The objectives of this section are first, identify the mobility limitations on the existing 3G femtocell architecture and second, propose solutions to support mobility on the protocols implemented by femtocells and the core network.

According with 3GPP specifications [22] when femtocells are powered up a discovery procedure is triggered to obtain local access information according to its own location and identity. This information consists of the entities that it needs to provide the service: the S-HMS, S-SeGW and HNB-GW. Then, the femtocell establishes a SCTP session with the HNB-GW and registers itself sending a `HNB REGISTER REQUEST` message. This is known as the HNB registration procedure.

Similarly, when an UE connects with a femtocell for the first time, an UE registration procedure is triggered to perform access control for that UE in the HNB-GW. If the operation is successful, a specific context identifier is assigned to that UE to be used between the femtocell and HNB-GW.

During the HNB registration procedure, the femtocell reports to the HNB-GW that it is reachable at a particular IP address and located in a particular venue. If the femtocell moves across heterogeneous networks, frequently IP address and venue changes, and thus connectivity losses, are expected.

To support mobility, the femtocell should be able to update its IP address and location to avoid connectivity losses but also to be reachable by the operator, who may want to detach a femtocell anytime or perform software updates. Specifically, two elements on the existing femtocell architecture should be updated to support mobility: the HMS, responsible for provision and remote management of femtocells; and the HNB-GW, which has context information for each femtocell and the UEs attached to it.

To reach this objective, we propose the addition of a new HNBAP message that updates the HNB location and IP address in the HNB-GW. This message may be named `HNB REGISTER UPDATE` and may have the parameters presented in Table I. Basically, this message extends the existing `HNB REGISTER REQUEST` to update the IP address in which the femtocell is reachable anytime and its location. Consequently,

TABLE I
PROPOSED HNB REGISTER UPDATE MESSAGE

PARAMETER	PRESENCE
Message type	Mandatory
HNB Identity	Mandatory
HNB Location Information	Mandatory
New IP address	Mandatory
PLMN-ID	Mandatory
Cell-ID	Mandatory
LAC	Mandatory
RAC	Mandatory
SAC	Mandatory

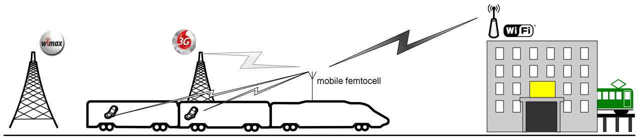


Fig. 4. Applicability scenario: Connectivity on trains

we also propose the addition of a `HBN REGISTER UPDATE ACCEPT` and `HBN REGISTER UPDATE REJECT` message to indicate if the operation was successful or not.

In addition to extend the HNBAP protocol to support mobility, we also propose the use of MOBIKE between the femtocell and the SeGW to maintain the IPsec tunnel during handovers. Using MOBIKE over the I_{UH} interface minor changes are needed. In fact, only two hosts (femtocell and SeGW) would require an upgrade to include the MOBIKE protocol in their existing IKEv2 implementations. Thus, the impact on the currently defined architecture would be minimal.

IV. EXPERIMENTAL TESTBED

The growth in wireless communication technologies over the last years opens several opportunities for supporting communication on trains. For example, users in a stationary train can have Internet access through the existing cellular networks or WLANs located in the train stations. However, when the train starts to move new problems emerge due to the handovers needed to maintain the continuity of service along the journey.

In such scenario, we propose to employ femtocells on trains to provide Internet connectivity to the passengers. The idea is to change the currently defined femtocell architecture to move femtocells through a heterogeneous network without losing connectivity with the core network. Thus, the femtocell would connect outside using the most suitable technology at a given time. For instance, if the train is stopped in a station it could connect with a Wi-Fi network. If on the contrary the train moves, it could connect with an existing cellular technology (e.g., UMTS, LTE or WiMAX) or even satellite (see Figure 4).

To that end, we propose the use of MOBIKE for the reasons mentioned in previous sections to support mobility between the femtocell and the core network, i.e., the security gateway. In this section we present the testbed created to simulate the train scenario. Later in Section V, we discuss the results obtained from that testbed.

TABLE II
TESTBED MAIN FEATURES

HOST	INITIATOR	RESPONDER	ROUTER
Kernel version	2.6.26.1	2.6.38	2.6.26.1
Network interfaces	2	1	2
Description	Mobile femtocell (HNB)	SeGW (core network)	Router (network conditions)
Software installed	StrongSWAN 4.5.3rc2	StrongSWAN 4.5.3rc2	Dummynet

Nowadays, there exist several IKEv2 open source software implementations. OpenSWAN [23], IKEv2 Project [24] or Racoon2 [25] support most features of IKEv2 but they don't provide the MOBIKE extension and also they are no developing it at this point in time. On the other hand, the OpenIKEv2 [26] project was developing the MOBIKE extension in early 2011 but now it is stopped.

StrongSWAN [27] is currently the only open source IKEv2 implementation supporting MOBIKE and therefore we use it for this work to study the feasibility of using MOBIKE on femtocell networks. From now on we use the MOBIKE terminology to refer the entities involved in the experiment, i.e., the mobile node (or femtocell) is called the initiator and the fixed node (or SeGW) the responder.

We conducted experiments on several Linux hosts whose main features are shown in Table II. The initiator represents the femtocell in our assumption since it is the host which moves through the heterogeneous network. It has two network interfaces to communicate with the outside world and has installed the StrongSWAN software to establish the IPsec tunnel with the responder. The responder has only one network interface. It terminates the IPsec tunnel with the initiator and can be seen as the SeGW in the femtocell architecture. Finally, the router is intended to simulate network conditions, hence it has installed Dummynet [28], a widely used tool able to simulate bandwidth, delay and packet losses among others by configuring rules.

The experiments are focused on evaluating the handover performance when an initiator establishes an IPsec tunnel and then it loses connectivity due to mobility. We measure the data traffic delay and losses when the handover occurs to discuss the validity of this solution depending on the concrete application. Specifically, we base on the experiment proposed by Ł Budzisz *et al.* in [29] to configure our testbed. That work studies the feasibility of using the SCTP failover mechanism for handover between different type of wireless networks: WLAN, UMTS and GSM EDGE (GERAN).

For better understanding, we show in Figure 5 an overview of our experiment in which three hosts are involved. The responder is the fixed node and terminates the IPsec tunnel. It is located in an operator's core network and is accessible on the Internet. The initiator is the mobile node. It has several network interfaces, one per wireless technology, connected with the router whereby the data traffic is sent. Finally, the router simulates network conditions and communicates the initiator with the responder. On one side it is connected with the responder by means of the fixed network, simulated by Dummynet (100 Mbps and 5 ms). The other side simulates

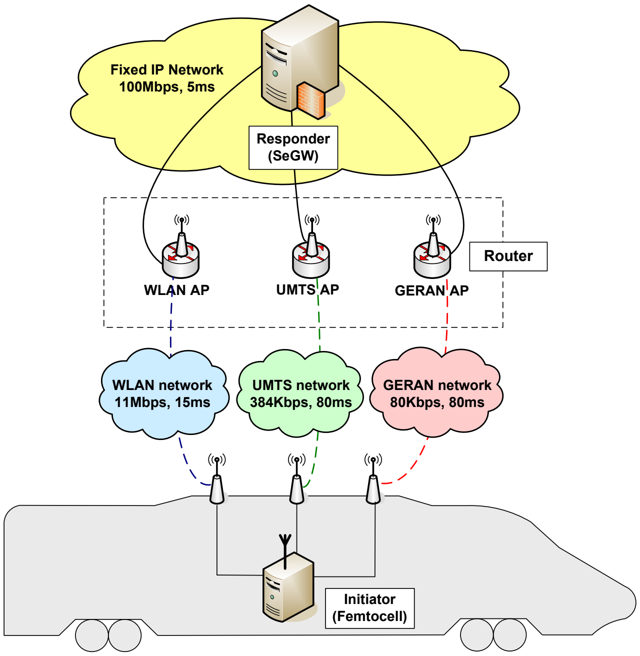


Fig. 5. Experiment overview

TABLE III
NETWORK PARAMETERS

NETWORK	BANDWIDTH	PROPAGATION DELAY
WLAN	11 Mbps	15ms
UMTS	384 Kbps	80 ms
GERAN	80 Kbps	80 ms
Fixed network	100 Mbps	5 ms

the three wireless technologies (GERAN, UMTS and WLAN) connected with the initiator. The parameters employed for the link simulation are shown in Table III. These values are taken from [29].

To evaluate the handover efficiency when moving from one wireless network to another, we have created tests with three different traffic sources: CBR-12.2, CBR-45 and CBR-Max. CBR-12.2 has a send rate of 12.2 Kbps since it is the highest rate provided by the AMR decoder [30], widely used in UMTS and GSM networks to encode voice traffic. Besides, CBR-45 sends 45 Kbps and CBR-Max sends with 70% of the wireless network bandwidth. Both traffic classes have been taken also from aforementioned work.

According the femtocell architecture, the data traversing the IPsec tunnel between the femtocell and its SeGW is sent over the I$_{UH}$ interface presented in Section II-A. The I$_{UH}$ protocol stack defines UDP as the single transport protocol used in the user plane. Since this work is centered on moving the tunnel established between the femtocell and its SeGW we just send CBR traffic over UDP. The properties of the three traffic sources employed in the experiment are listed in Table IV.

To evaluate the handover performance, we assume the handover delay as the time elapsed between the first data packet tunneled through the destination network and the last data packet tunneled through the home network when using MOBIKE. In such situation, two delays are expected: the one introduced by the inherent features of the wireless networks (propagation and transmission delays) and the one added by the MOBIKE implementation.

The handover scenarios contemplated for this work are the following: UMTS to WLAN, UMTS to GERAN, WLAN to UMTS, WLAN to GERAN, GERAN to UMTS and GERAN to WLAN. Next section presents the results obtained when sending the three types of traffic sources over all scenarios.

V. Result analysis

To analyze the handover performance between two different wireless networks we configure a test that consists of sending data traffic from the primary interface and at a given rate. After 5 seconds, the secondary interface detects connectivity from another wireless technology and obtains a new IP address. Three seconds later, the primary interface loses connectivity and as a result, a handover is triggered to the secondary interface.

In this section, we evaluate the latency and packet loss associated with the handover procedure between different wireless technologies. Each test is repeated 30 times and then, we compute the mean and 95% confidence interval of the measured handover delays and traffic loss.

We present the results according several scenarios depending on the home network propagation delay (D_{hn}) and the destination network propagation delay (D_{dn}). Therefore we contemplate three different scenarios.

In the first one, the home network propagation delay (15 ms) is lower than the destination one (80 ms). WLAN to UMTS and WLAN to GERAN handovers are grouped in this scenario. The second one corresponds to the opposite case, where the home network propagation delay (80 ms) is higher than the destination delay (15 ms). As an example GERAN or UMTS to WLAN handovers cover this case. Finally in the last scenario, both home and destination network propagation delays are the same (80 ms), as for example in the GERAN to UMTS handover and vice versa.

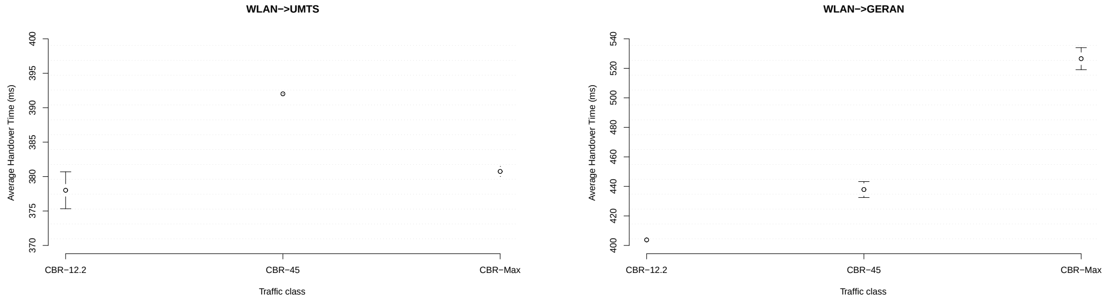
A. Scenario 1: $D_{hn} < D_{dn}$

Figure 6 illustrates a comparison between two handover procedures performed from WLAN to GERAN when the initiator sends CBR-12.2 traffic. The red and dotted bars represent the packets received by the responder ideally, i.e., the delay added in that handover is produced by the inherent features of the networks crossed (transmission and propagation delays). On the other hand, the black bars represent the packets received by the responder when using MOBIKE. In such case two delays are added, the one associated with the networks crossed and also the delay produced by the MOBIKE implementation.

After the first 8 seconds the data is sent through the WLAN network. Then, we force a handover to the destination network, i.e., GERAN. In both cases, the last packet sent through the home network is seen at the second 8.079 with a sequence number of 337. Ideally, packet 338 is received through the GERAN network 67.176 ms delayed due to the propagation and transmission delays associated with GERAN. On the other

TABLE IV
PROPERTIES OF THE TRAFFIC TYPES

TRAFFIC	DESCRIPTION	PACKET SIZE	SEND RATE	TIME BETWEEN PACKETS
CBR-12.2	Constant Bit Rate	32 Bytes	12.2 Kbps	20.983 ms
CBR-45	Constant Bit Rate	200 Bytes	45 Kbps	36.8 ms
CBR-Max-G	Constant Bit Rate	1410 Bytes	0.7× GERAN bandwidth	201.428 ms
CBR-Max-U	Constant Bit Rate	1410 Bytes	0.7× UMTS bandwidth	41.964 ms
CBR-Max-W	Constant Bit Rate	1410 Bytes	0.7× WLAN bandwidth	1.464 ms



(a) WLAN to UMTS

(b) WLAN to GERAN

Fig. 7. Handover delay when $D_{hn} < D_{dn}$

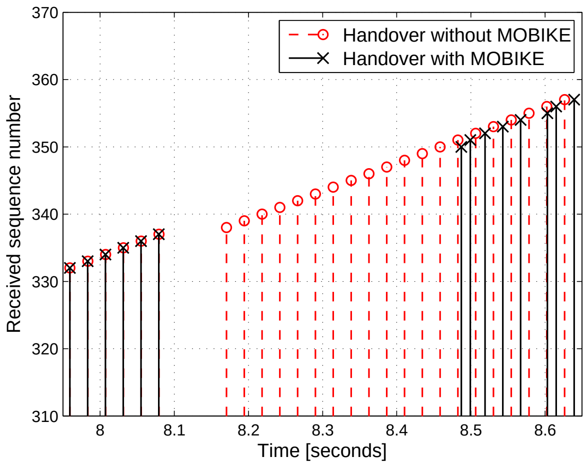


Fig. 6. WLAN to GERAN handover example (12.2 Kbps)

hand, 12 packets are lost during the handover when using MOBIKE (more information on packet loss in the Appendix). Thus, the first packet received from the destination network appears at second 8.482 with a sequence number of 350. Here, the handover delay takes 403.31 ms which is the time elapsed between packet 350 and packet 337 (black bars).

Now, we present and discuss two concrete applications for this scenario after conducting 30 tests on each traffic class: the WLAN to UMTS and the WLAN to GERAN handovers.

- WLAN to UMTS
  In this scenario, the home network bandwidth is much faster than destination one (almost 29 times more). Moreover, only CBR-12.2 and CBR-45 traffic can be carried to the destination network without overloading it.
  The CBR-12.2 measurements indicate that on average, the handover delay is 378.009±2.683 ms. That values are much larger than the ones obtained in the opposite case and it makes sense. Before the initiator loses connectivity, the responder is receiving the data at the same rate that the initiator sends (i.e., 12.2 Kbps). Then, after handover completes, the traffic is sent through the UMTS network which introduces an additional delay on that traffic.
  The values obtained for CBR-45 traffic are slightly larger than CBR-12.2 ones. Thus, the average handover delay measured is 392.014±0.005 ms.
  Finally, the CBR-Max measurements led to an average handover delay of 380.743±0.732 ms.
  Figure 7(a) shows the average handover delay for the different traffic sources for the WLAN to UMTS scenario.
- WLAN to GERAN
  This handover scenario is similar the previous one but more restrictive due to the GERAN bandwidth. As before, CBR-12.2 and CBR-45 traffic classes can be carried through the GERAN network without overloading but CBR-Max traffic, that suppose a data rate of 7.7 Mbps, cannot.
  Sending 12.2 Kbps from WLAN to GERAN we obtain that on average the handover delay takes 403.747±1.172 ms. Conversely, we measure an average handover delay of 437.864±5.417 ms when sending 45 Kbps.
  The CBR-Max measurements shows the largest delay, 526.564±7.463 ms of average handover delay. Hence, it may be unacceptable for some applications.
  The UMTS bandwidth is 4.8 times larger than GERAN one. This fact is reflected on the handover delays which are higher on the WLAN to GERAN handover on all the traffic classes. Figure 7(b) illustrates the measurements mentioned before.

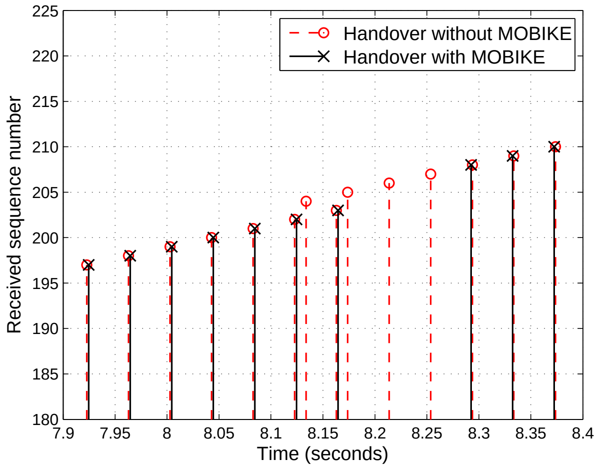


Fig. 8. UMTS to WLAN handover example (45 Kbps)

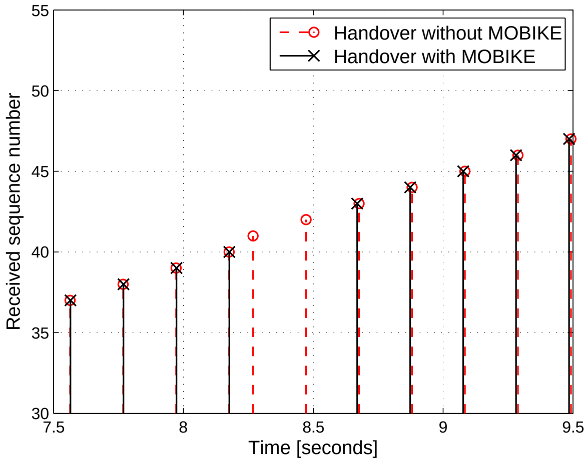


Fig. 10. GERAN to UMTS handover example (CBR-Max: 56 Kbps)

B. Scenario 2: $D_{hn} > D_{dn}$

Our second scenario represents those handovers in which the home network has a propagation delay of 80 ms and the destination one 15 ms. For that reason, it is likely that some packets sent through the destination network will arrive earlier than the latest sent by the home network (i.e., they can anticipate).

Figure 8 shows an UMTS to WLAN handover example, in which CBR-45 traffic is sent. Ideally packet number 203 would arrive at second 8.162, 29.09 ms latter than packet 204 which arrives at second 8.133 (red bars). We compute 128.50 ms of handover delay and 4 lost packets.

Below we discuss the handover delay for UMTS to WLAN and GERAN to WLAN handovers according to the traffic sources.

- UMTS to WLAN
  In this scenario the propagation delay of the home network (UMTS) is greater than the destination one (WLAN). The values obtained for the different traffic sources are shown in Figure 9(a).
  When the initiator sends 12.2 Kbps to the responder, the average handover delay obtained is 100.543±0.852 ms. Also, when the data rate is 45 Kbps, the average handover delay increases to 128.800±0.719 ms and last, if sends 268.8 Kbps (70% of the UMTS network bandwidth) the average time slightly decreases to 128.011±0,012 ms.
  The handover delays obtained from the CBR-45 and CBR-Max measurements are very close probably because of the waiting time set on both classes. CBR-45 waits for 36.8 ms between each packet whereas CBR-Max waits for 41.964 ms, slightly more than 5 ms.
- GERAN to WLAN
  Tests performed over the GERAN to WLAN scenario achieve better results in terms of handover delay than those obtained over the UMTS to WLAN in both CBR-12 and CBR-45 traffic classes. Figure 9(b) shows the results. Sending CBR-12 traffic the handover delay takes on average 94.142±1.341 ms. Results increase lightly when sending CBR-45 traffic, taking an average handover delay of 104.937±0.641 ms.
  Finally, the CBR-Max traffic does not implies a lower handover delay as it happens on the UMTS to WLAN handover. Indeed, we obtain the highest values of all classes being the average handover delay 208.582±1.503 ms. These values don't decrease the handover delay with respect to the CBR-45 due to the packet rate, which is not enough to appreciate that reduction.

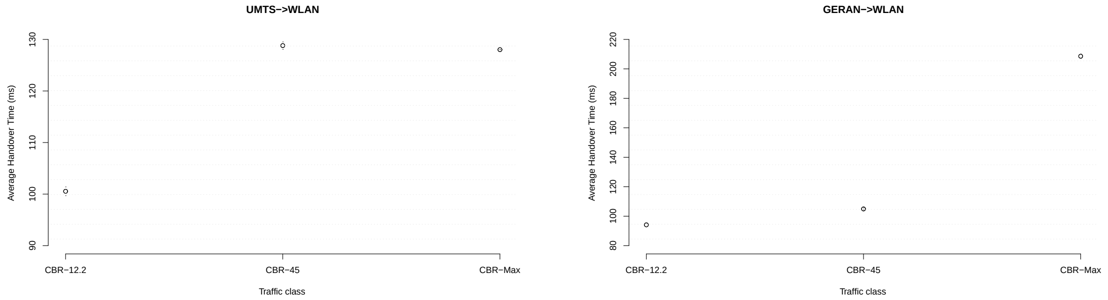
C. Scenario 3: $D_{hn} = D_{dn}$

The last scenario implies that both home and destination networks have the same propagation delay. The example shown in Figure 10 represents the handover between GERAN and UMTS, both configured with a propagation delay of 80 ms. To evaluate the ideal and practical handover we send CBR-Max traffic which corresponds in this case to 56 Kbps.

In the ideal case, packet number 41 arrives earlier than if it had been sent through the home network due to UMTS, which has a bandwidth greater than GERAN. On the other hand, the practical case (using MOBIKE) implies 2 lost packets and 504.02 ms of handover delay (time elapsed between packet 40 and 43).

Now, we discuss the handover delays for UMTS to GERAN and GERAN to UMTS handovers resulting from the use of MOBIKE.

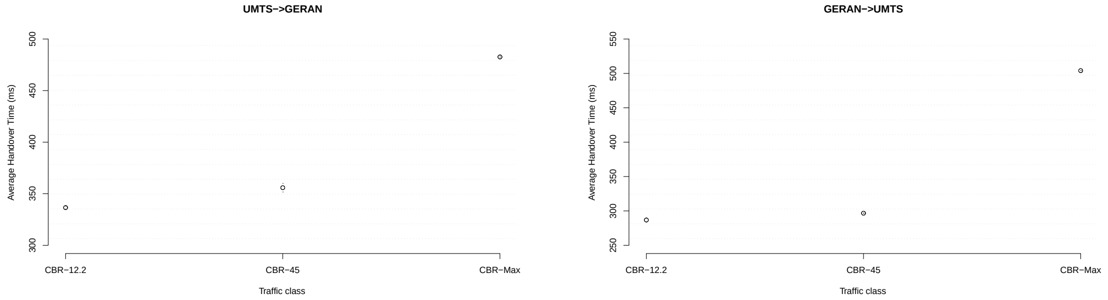
- UMTS to GERAN
  Propagation delays are set to be the same in UMTS and GERAN. However, UMTS bandwidth is almost 5 times faster than GERAN.
  We measured an average handover delay of 336.555±1.399 ms for CBR-12.2 traffic. Also, an increase is observed for CBR-45 traffic where the average handover delay takes a value of 355.846±4.172 ms.
  On the other hand, the CBR-Max traffic takes an average handover delay of 482.635±0.718 ms, much larger than CBR-45. Figure 11(a) shows these results.



(a) UMTS to WLAN

(b) GERAN to WLAN

Fig. 9. Handover delay when $D_{hn} > D_{dn}$



(a) UMTS to GERAN

(b) GERAN to UMTS

Fig. 11. Handover delay when $D_{hn} = D_{dn}$

- GERAN to UMTS handover
  Both GERAN and UMTS networks have 80 ms of propagation delay configured. In Figure 11(b) we show the values obtained for all traffic classes.
  On average the CBR-12.2 traffic takes a handover delay of 286.782±2.894 ms and CBR-45 296.664±0.567 ms. Comparing with the previous scenario, both values are below the better result obtained by the UMTS to GERAN, which takes more than 336 ms.
  CBR-Max traffic takes on average a handover delay of 504.028±0.008 ms, much larger than previous classes probably because of the large waiting time established between each packet (201.428 ms).

D. Packet loss

The mobility solution offered by StrongSWAN implies a packet loss during the handover procedure due to the time needed to re-establish the tunnel. Although this issue will be studied in detail later (see Appendix), in this section we present and discuss the measurements performed in terms of packet loss under each scenario.

We consider one packet is lost if it did not reach the destination or if it was sent outside the tunnel[1]. Hence, we calculate the packet loss by reading the sequence number of the first data packet tunneled from the destination network and the sequence number of the last packet tunneled from the home network.

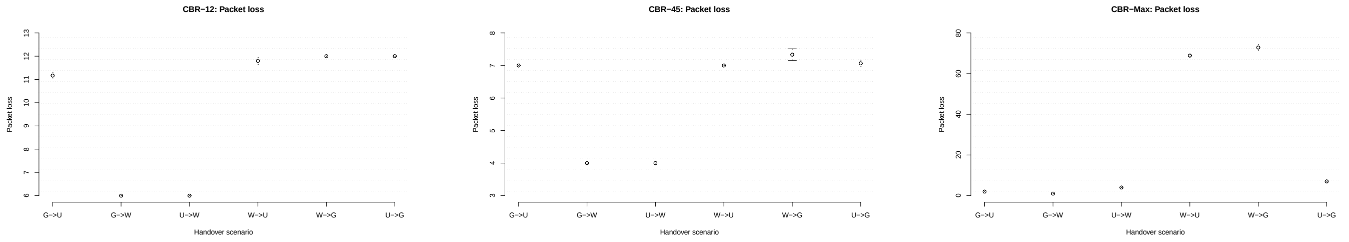
Figures 12(a), 12(b) and 12(c) show the mean and 95% confidence interval of the measured packet loss for each handover scenario. The horizontal axis represents the concrete handover scenario whereas the vertical axis the number of packets lost.

As shown in figures, similar values are obtained in those handover pairs with same destination network and traffic. For instance, in CBR-12.2 the values obtained from the UMTS to WLAN handover and GERAN to WLAN are the same (6 packets) and also the ones obtained from the WLAN to GERAN and UMTS to GERAN handovers (12 packets). Moreover, the average packet loss measured on the GERAN to UMTS and WLAN to UMTS handovers are very close: 11.166±0.141 packets on average in the GERAN to UMTS scenario and 11.8±0.159 packets in the WLAN to UMTS scenario.

In a similar way, the CBR-45 results show the same behavior. When WLAN is the destination network 4 packets are lost due to the handover. Also, the same is noticed when UMTS is the destination network. In such case, 7 packets lost. On the contrary, values are close when GERAN is the destination network. 7.33±0.179 packets are lost on average if WLAN is the home network and 7.066±0.094 if on the contrary is UMTS.

The latter case (CBR-Max) differs from the previous ones due to the data rate, which depends on the home network bandwidth. For that reason, none of the results matches any others. When GERAN is the home network, 56 Kbps is the data rate configured. The packet loss measured during

[1] On a real femtocell network, those packets sent out of the IPsec tunnel will be dropped by the endpoints.



(a) CBR-12.2

(b) CBR-45

(c) CBR-Max

Fig. 12. Packet loss under different traffic sources: CBR-12.2, CBR-45 and CBR-Max

handovers is 2 packets when UMTS is the destination network and 1 packet if on the contrary, it is WLAN the destination one. It makes sense since in the former case the handover takes more time than the latter and thus more packets are lost. Similarly, if UMTS is the home network two values are observed: 4 packets on average are lost during handovers to the WLAN network and 7 packets if it is GERAN the destination network. The latter value is larger than the previous one due to the handover delay measured for that scenario, which is almost 4 times greater than the former. Finally, WLAN acting as a home network implies that 7.7 Mbps are sent in the CBR-Max class. Bigger packet loss is expected due to the large data rate sent. Results show packets losses of 68.866±0.305 packets on average when the handover ends in the UMTS network and 72.833±1.503 packets if it ends in GERAN.

VI. Conclusion and future work

In this paper we present the MOBIKE protocol as a mobility solution for femtocell networks and specifically to provide seamless handovers under heterogeneous networks. We explain a laboratory experimental setup where we simulate two elements of the femtocell architecture: the femtocell (mobile node) and the security gateway (fixed node).

In section IV we present the applicability scenario for this work consisting of provide Internet connectivity by means of femtocells to passengers travel by train. Those passengers could spent their journey time surfing on the Internet, watching their favorite TV series or talking on the phone. All these applications have different requirements in terms of delay and packet loss and therefore not all the scenarios presented above may be feasible. Real time communications have strict requirements on delay, jitter and packet loss. For instance, voice and interactive video require delays not exceeding 150 ms [31]. Ideally (without MOBIKE) all scenarios satisfy this requirements. However, the practical results show that only two handover scenarios satisfy them: the UMTS to WLAN handover for all traffic classes and the GERAN to WLAN for CBR-12.2 and CBR-45. These results reveal that StrongSWAN is not fast enough to be used for real time applications in most handover scenarios. The time spent by the implementation to reestablish the tunnel is high and it needs to be decreased. This includes the time spent to detect a path failure due to mobility and the time spent on update the IP addresses in the SAD. On the other hand streaming-video applications are not delay sensitive due to the buffering. Delay should not take more than 4 to 5 seconds, depending on the buffering capabilities. This requirement is so lenient that all the proposed scenarios satisfy them.

The handover delay may be dependent of the waiting time between packets. However, we can not conclude yet due to the tests performed in which we change the waiting time between packets and the data rate for each traffic class. We will perform new tests in a future to conclude with this statement. To this end, two approaches will be studied according the parameter fixed. First we will set a fixed bit rate whereas the waiting time between packets (and thus the packet size) will be varied. Second the waiting time will be fixed whereas the bit rate will be varied.

Most of the Internet connections rely on TCP, e.g., data transfer, web browsing, remote management applications…In this work we only consider UDP as transport protocol since the I_{UH} protocol stack defines it on the user plane. However, it would be interesting evaluate the MOBIKE performance when sending TCP traffic. Future works will focus on sending that kind of traffic.

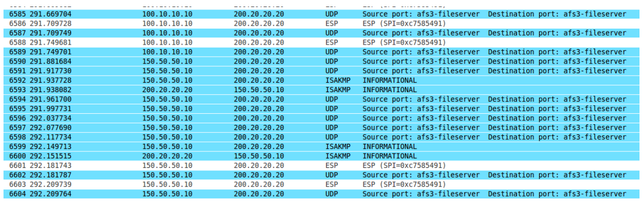
Regarding packet loss we can conclude that greater packet loss is obtained when more packets per second are sent. MOBIKE introduces a delay to reestablish the connection and during that period of time every packet sent is lost. Hence, the packet rate (and therefore the waiting time between packets) affects the packet loss. The amount of packet loss that is acceptable depends on the type of data being sent. For example, for VoIP traffic, the only effect seen due to the occasional dropped packet is jitter, and therefore missing one or two packets occasionally will not affect the quality of the conversation. However, losses between 5% and 10% of the total packet stream will affect the quality significantly [32]. In order to avoid packet loss a method to hold data packets during handover procedures may be implemented.

Appendix

During the course of this work we have detected some limitations in the tools employed. In this section we want to enumerate those limitations and unwanted behaviors introduced by the StrongSWAN and Dummynet implementations.

A. *StrongSWAN tunnel update*

Every time a handover procedure is initiated by StrongSWAN, the SAs involved in the communication are updated. During that upgrade the application level is still



```
6585 291.669704    100.10.10.10    200.20.20.20    UDP     Source port: afs3-fileserver  Destination port: afs3-fileserver
6586 291.709728    100.10.10.10    200.20.20.20    ESP     ESP (SPI=0xc7585491)
6587 291.709749    100.10.10.10    200.20.20.20    UDP     Source port: afs3-fileserver  Destination port: afs3-fileserver
6588 291.749681    100.10.10.10    200.20.20.20    ESP     ESP (SPI=0xc7585491)
6589 291.749701    100.10.10.10    200.20.20.20    UDP     Source port: afs3-fileserver  Destination port: afs3-fileserver
6590 291.881684    150.50.50.10    200.20.20.20    UDP     Source port: afs3-fileserver  Destination port: afs3-fileserver
6591 291.917730    150.50.50.10    200.20.20.20    UDP     Source port: afs3-fileserver  Destination port: afs3-fileserver
6592 291.937728    150.50.50.10    200.20.20.20    ISAKMP  INFORMATIONAL
6593 291.938082    200.20.20.20    150.50.50.10    ISAKMP  INFORMATIONAL
6594 291.961700    150.50.50.10    200.20.20.20    UDP     Source port: afs3-fileserver  Destination port: afs3-fileserver
6595 291.997731    150.50.50.10    200.20.20.20    UDP     Source port: afs3-fileserver  Destination port: afs3-fileserver
6596 292.037734    150.50.50.10    200.20.20.20    UDP     Source port: afs3-fileserver  Destination port: afs3-fileserver
6597 292.077690    150.50.50.10    200.20.20.20    UDP     Source port: afs3-fileserver  Destination port: afs3-fileserver
6598 292.117734    150.50.50.10    200.20.20.20    UDP     Source port: afs3-fileserver  Destination port: afs3-fileserver
6599 292.149713    150.50.50.10    200.20.20.20    ISAKMP  INFORMATIONAL
6600 292.151515    200.20.20.20    150.50.50.10    ISAKMP  INFORMATIONAL
6601 292.181743    150.50.50.10    200.20.20.20    ESP     ESP (SPI=0xc7585491)
6602 292.181787    150.50.50.10    200.20.20.20    UDP     Source port: afs3-fileserver  Destination port: afs3-fileserver
6603 292.209739    150.50.50.10    200.20.20.20    ESP     ESP (SPI=0xc7585491)
6604 292.209764    150.50.50.10    200.20.20.20    UDP     Source port: afs3-fileserver  Destination port: afs3-fileserver
```

Fig. 13. StrongSWAN handover limitation

sending traffic but it does not reach the destination properly. We have detected a packet loss during that time where packets sent from a secondary IP address are lost until the initiator SAs are updated. When the initiator loses connectivity from its primary address (see Figure 13), the default route associated with that interface is also lost. Then, the UDP socket request the kernel for an available IP address and changes its source IP for one available secondary address (150.50.50.10 in the figure). Next, the socket starts to send the traffic from that IP but MOBIKE does not send inside it the tunnel.

This behavior is produced by the IKEv2 keying daemon, *charon*, when it updates an IPsec SA. Charon first deletes and then re-adds the policies in the kernel. Within the short time-frame during which no matching policy is installed in the kernel, unencrypted packets could have been transmitted. This means that StrongSWAN does not hold data packets until the handovers are completed. However, it does not have to be aware of that since the MOBIKE specification does not include that feature.

To avert this behavior we have installed DROP policies in the kernel by which unencrypted packets are denied to leave the host, thus they have seen as packet loss since the responder never receives them.

B. Binding interfaces on StrongSWAN

On heterogeneous networks, selecting the best technology every time a handover is needed is an important issue. StrongSWAN allows binding multiple interfaces on a specific IKE SA, however, it does not provide interface selection to specify which interface will be the next to perform the handover. When the handover is triggered, the routing table is checked and the interface chosen will be the interface appearing the first in the routing table.

Although the user application may control the interface selection by changing the subnet mask in the routing table, it would be interesting if StrongSWAN provided user commands to specify which interface will be the next used after handover procedures.

C. Dummynet buffering limitations

Figure 14 shows an execution example when CBR-Max traffic is sent from UMTS to GERAN. The horizontal axis represents the execution time whereas the vertical axis the sequence number received by the responder. First and during 8 seconds, packets are transmitted through UMTS at 268.8 Kbps and then a handover is triggered to a slower network. Due to the bandwidth reduction, packets are now represented with a larger space between them since the destination network cannot convey the data at the same data rate as sent. Seconds latter, an increase in the slope of the line is shown because some packets were dropped due to congestion, i.e., one packet is received out of each *N*. Specifically, 72 packets are received through the GERAN network and then, at second 20.404 the congestion appears. This effect leads to the network to drop the 67.56% of all the packets sent.

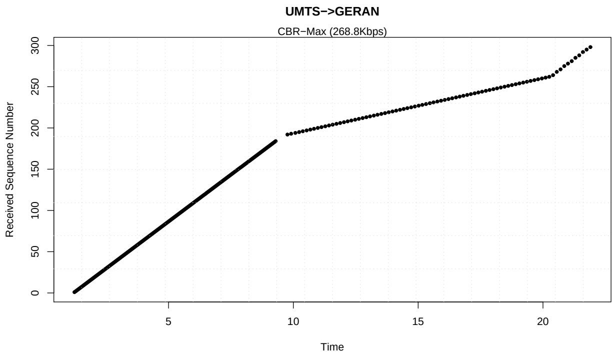


Fig. 14. UMTS to GERAN handover (CBR-Max)

The congestion is caused by Dummynet, which has a limited buffer configured. The Dummynet queue length can be changed by means of user commands, however, 100 bytes is the maximum size allowed.

Acknowledgment

The authors want to thank Alberto Cortés-Martín (aka *Linux Black Belt*) for his helpful discussions and precious advices on Linux systems and programming. Also thank Marta Carbone for the prompt support received about Dummynet and Alicia Rodríguez-Carrión for her excellent skills on MATLAB computing.

References

[1] P. Noriega-Vivas, C. Campo, C. Garcia-Rubio, and E. Garcia-Lozano, "Supporting l3 femtocell mobility using the mobike protocol," in *The Second International Conference on Access Networks*, june 2011.

[2] G. de la Roche and J. Zhang, *Femtocells: Technologies and Deployment*. Wiley, december 2009.

[3] M. Aguado, O. Onandi, P. Agustin, M. Higuero, and E. Jacob Taquet, "Wimax on rails," *Vehicular Technology Magazine, IEEE*, vol. 3, no. 3, pp. 47 –56, sept. 2008.

[4] P. Rodriguez, R. Chakravorty, J. Chesterfield, I. Pratt, and S. Banerjee, "Mar: a commuter router infrastructure for the mobile internet," in *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, ser. MobiSys '04. New York, NY, USA: ACM, 2004, pp. 217–230. [Online]. Available: http://doi.acm.org/10.1145/990064.990091

[5] *TS 25.467: UTRAN architecture for 3G Home Node B (HNB); Stage 2 (Release 9)*, 3GPP, june 2010.

[6] *TS 25.401: UTRAN overall description (Release 10)*, 3GPP, june 2011.

[7] *TS 25.469: UTRAN Iuh interface Home Node B (HNB) Application Part (HNBAP) signalling (Release 9)*, 3GPP, september 2010.

[8] *TS 25.468: UTRAN Iuh Interface RANAP User Adaption (RUA) signalling (Release 9)*, 3GPP, september 2010.

[9] *TR-069: CPE WAN Management Protocol v1.1, Version: Issue 1 Amendment 2*, Broadband Forum, december 2007.

[10] *Broadband Forum*, http://www.broadband-forum.org, Last Accessed: 1 april, 2011.

[11] *Femto Forum*, http://www.femtoforum.org, Last Accessed: 1 april, 2011.

[12] *TR-196: Femto Access Point Service Data Model*, Broadband Forum, april 2009.

[13] S. Kent, *Security Architecture for the Internet Protocol (RFC 4301)*, december 2005.

[14] S. Kent., *IP Encapsulating Security Payload (ESP) (RFC 4303)*, december 2005.

[15] S. Kent, *IP Authentication Header (AH) (RFC 4302)*, december 2005.

[16] C. G. Kaufman and P. Hoffman, *Internet Key Exchange Protocol Version 2 (IKEv2) (RFC 5996)*, september 2010.

[17] *IANA: Internet Assigned Numbers Authority*, http://www.iana.org, Last Accessed: 28 february, 2011.

[18] V. Devarapalli and K. Weniger, *Redirect Mechanism for the Internet Key Exchange Protocol Version 2 (IKEv2) (RFC 5685)*, november 2009.

[19] P. Eronen, *IKEv2 Mobility and Multihoming Protocol (MOBIKE) (RFC 4555)*, june 2006.

[20] T. Kivinen and H. Tschofenig, *Design of the IKEv2 Mobility and Multihoming (MOBIKE) Protocol (RFC 4621)*, august 2006.

[21] C. Metz and B. Phan, *PF KEY Key Management API, Version 2 (RFC 2367)*, july 1998.

[22] *TS 32.583: Telecommunication management; Home Node B (HNB) Operations, Administration, Maintenance and Provisioning (OAM&P); Procedure flows for Type 1 interface HNB to HNB Management System (HMS) (Release 9)*, 3GPP, december 2009.

[23] *Openswan*, http://www.openswan.org, Last Accessed: 20 january, 2011.

[24] *IKEv2 Project*, http://sourceforge.net/projects/ikev2, Last Accessed: 20 august, 2011.

[25] *Racoon2*, http://www.racoon2.wide.ad.jp/w/?Racoon2, Last Accessed: 20 august, 2011.

[26] *OpenIKEv2*, http://openikev2.sourceforge.net, Last Accessed: 21 august, 2011.

[27] *strongSWAN: The OpenSource IPsec-based VPN Solution for Linux*, http://www.strongswan.org, Last Accessed: 9 may, 2011.

[28] *Dummynet*, http://info.iet.unipi.it/~luigi/dummynet, Last Accessed: 24 august, 2011.

[29] L. Budzisz, R. Ferrús, A. Brunstrom, K. J. Grinnemo, R. Fracchia, G. Galante, and F. Casadevall, "Towards transport-layer mobility: Evolution of sctp multihoming," *Comput. Commun.*, vol. 31, pp. 980–998, March 2008.

[30] "Digital cellular telecommunications system (phase 2+) (gsm); adaptive multi-rate (amr) speech transcoding (gsm 06.90 version 7.2.1 release 1998)," 1998.

[31] T. Szigeti and C. Hattingh, *End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs (Networking Technology)*. Cisco Press, 2004.

[32] K. C. Mansfield and J. L. Antonakos, *Computer Networking for LANs to WANs: Hardware, Software and Security*, 1st ed. Delmar Learning, 2009.