An Identifier-based Architecture for Native Vertical Handover Support

Walter Wong, Marcus Giraldi and Maurício F. Magalhães
School of Electrical and Computer Engineering (FEEC)
State University of Campinas (UNICAMP)
Campinas/São Paulo - Brazil
{wong, marcus, mauricio}@dca.fee.unicamp.br

Fábio L. Verdi
Federal University of São Carlos (UFSCar) Sorocaba, Brazil
verdi@ufscar.br

Abstract—The growing demand for ubiquitous computing lead manufactures to develop multi-band devices supporting different network access technologies, such as GSM, Wi-fi, and UMTS. Although these devices support different access technologies, they lack of native vertical handover support, breaking any ongoing connections whenever users switch between access technologies.

In this paper we present a next-generation Internet architecture with native vertical handover support, providing mechanisms to support mobility between different access technologies without disruptions. The architecture introduces an identification layer that decouples the host identification from its location, enabling native mobility support for legacy applications. The identification layer employs cryptographic identifiers to identify end-hosts over the Internet and to provide security mechanisms for identity management during vertical handovers. As a proof of concept, a prototype was implemented and evaluated in different vertical handover scenarios in GSM, Wi-fi and wired domains.

Keywords—Security Protocol, Mobility, Heterogeneous Networks, Next-generation Architectures

I. INTRODUCTION

The growing demand for ubiquitous computing lead manufactures to develop devices with multiple network interfaces to access different wireless networks. As a consequence, users have a broader range of options to choose the best available network to connect based on the users’ current needs, such as higher data rates for a file download or wider area coverage for mobile devices. Two classes of wireless technology are mainly supported by these devices: IEEE 802.11 (Wi-fi) and Global System for Mobile Communication (GSM). The first one offers higher data rates (up to 54 Mbps in the 802.11ag modes) but provides small coverage area, restricted to Wi-fi hotspots. The second one offers wider area coverage with high mobility, but with lower data rates.

Although these devices support different wireless technologies, they lack of native vertical handover support, which is the process of communication maintenance of mobile hosts while they move from one technology domain (e.g. GSM, Wi-fi, UMTS) to another. One of the root causes is the IP semantic overload problem [1], [2], where the IP address is used as end-host identifier in the transport layer and as topological identifier in the network layer. Therefore, whenever one host moves from one technology domain to another, it changes its topological identifier (its current IP address), but also changes its end-host identifier. Any connections bound to the previous topological identifier are broken and cannot be naturally restored. Also, changes in the IP address in a mobile environment opens security issues, since there is not any credential binding between the end-host identifier with its identity.

In this paper we present a next-generation Internet architecture extending our previous works [3]–[5] to provide native vertical handover support, allowing mobile devices to move from one technology domain to another without any connection disruptions. The vertical handover support is provided by the introduction of an identification layer decoupling the host identification from its location, enabling the native mobility support for legacy applications and solving the IP semantic overload problem, as shown in Fig. 1. The identification layer provides stable identifiers to legacy applications while it dynamically binds to the network layer, allowing the deployment of the mobility feature. In order to provide identity management during vertical handovers, the architecture uses cryptographic identifiers to identify end-hosts over the Internet. These identifiers are generated from the cryptographic hash over the public key, binding the end-host identity with its private key.

Figure 1. (a) Current protocol stack. (b) Protocol stack with the identification layer.

The architecture can be incrementally deployed over the Internet using a Distributed Hash Table (DHT) and does not require any modifications in the operating system or in the legacy applications. As a proof of concept, we implemented
a prototype to evaluate the handover support in a real environment in different vertical handover scenarios using GSM, Wi-fi and wired networks.

The rest of the paper is organized as follows. Section 2 presents the background and summarize the related work. Section 3 describes the next-generation architecture and its features to support the vertical handover. Section 4 presents the implementation design of the proposed architecture. Section 5 shows the prototype evaluation and discusses the experimental results. Finally, Section 6 concludes the paper.

II. BACKGROUND AND RELATED WORK

In this section we present the handover classification in horizontal (mobility within the same wireless technology) and vertical (mobility among different wireless technologies, e.g., Wi-fi to GSM) approaches. Then we summarize the related work, outlining the main points.

A. Handover Classification

The handover procedure is defined as the process of communication maintenance without disruptions when users move from one network technology to another [6]. The commonest handover procedure is called **Horizontal Handover**, which is the handover procedure among the same network technology (e.g. Wi-fi to Wi-fi), illustrated in Fig. 2.

The second handover case studied is called **Vertical Handover**, which is the handover procedure among different access technologies. In this procedure, mobile devices switch from one network interface attached to one wireless technology to another. The **Vertical Handover** can be classified based on the characteristics of the domain (**upward** or **downward**) and the decision to make the handover process (**Imperative** or **Alternative**) [7], [8]. An **upward vertical handover** occurs when a mobile device moves from a high data rate network with small coverage to a wider coverage and lower data rate (for instance, from Wi-fi to GSM). On the other hand, the **downward vertical handover** occurs when a mobile device moves from a low data rate network with high coverage to a higher data rate with smaller coverage (e.g. GSM to Wi-fi).

The second classification considers the handover decision: it can be an **imperative vertical handover**, where a mobile device moves due to a low signal from the access point or base station, or an **alternative vertical handover**, where mobile devices switch the access technology due to other criteria, such as lower cost or higher data rate. Figure 3 illustrates an example of **upward** and **downward** vertical handovers.

B. Other Mobility Approaches

The Mobile IP (MIP) [9] is a network layer mobility support solution, introducing two IP addresses for the end-host mobility support: the **Home Address** and the **Care-of Address**. The first one represents the static end-host identifier and rarely changes, even during mobility events. The latter one is the temporary address that an end-host receives when it arrives in a new network. Hence, legacy applications bind to the **Home Address** and send packets directly to the correspondent node (CN) if the node is located in the same network. Otherwise, packets are sent to the destination **Home Agent** and then they are encapsulated to the destination **Care-of Address**. The handover procedure is transparent for legacy applications, since applications bind to the **Home address** and they are not aware of the changes in the **Care-of Address**.

The Takeover [10] protocol employs a mechanism that delegates the handover process to a neighbor node that is already in the destination domain. The mobile node first searches an available neighbor and sends a **takeover request** message to the neighbor. The neighbor starts the pre-authentication and pre-registration procedure in the destination domain and replies with all the parameters to the mobile node. Then, the mobile node configures itself with the new parameters and seamlessly move to the new wireless domain.

The Universal Seamless Handoff Architecture (USHA) [11] proposes a vertical handover solution using an IP tunneling technique. Each mobile has a pair of virtual/fixed IP addresses and uses them to create a tunnel to a Handover Server (HS) located in the Internet. Whenever a mobility event occurs, the mobile host is responsible for switching
between the underlying physical connection to the virtual tunnel in the new interface and also notifying the HS about the new IP address. The HS then updates the mobile hosts tunnel settings and any subsequent data packets are correctly forwarded.

In [12], the authors propose a technique to support seamless vertical handovers using multi-homed mobile access points. Mobile nodes have multiple pre-established and independent paths, regardless whether the node is using it or not. Whenever a handover event is triggered, the mobile node switches from the current access technology to the visiting one which is has already setup the access. The multi-homing feature used to support the vertical handover is provided by the same mechanism employed in the Mobile IP.

This mechanism benefits from the removal of the setup time during the handover, since all protocol interactions required to access the visiting technology domain is previously configured. In [13], the authors present an anticipated vertical handover mechanism based on Mobile IP, where mobile nodes detecting new wireless networks start the registration process before breaking the connectivity (make-before-break). It is assumed that different access technologies (e.g. UMTS, WLAN) are available in the same region and there is an overlap between them to support vertical handovers with reduced or without packet losses.

Finally, we should mention the recent IEEE effort named Media Independent Handover Services (MIHs) [14] which defines an standard for supporting handover between different heterogeneous IEEE 802 technologies. Such standard proposes the usage of an intermediate layer common for all technologies. To some extend, our identifier-based layer is an instance of the MIH proposal and anticipates challenges and advantages of having such layer by implementing a prototype evaluated in a real heterogeneous handover scenario.

**III. ARCHITECTURE PROPOSAL**

This section presents the identifier-based architecture and discusses its main requirements in order to support native vertical handover. First, we discuss the identification layer and its built-in security features. Second, we present the mechanism to support the vertical handover procedures. Third, we discuss the transparent support of legacy applications. Finally, we discuss the security mechanism to protect the name resolution in the architecture.

**Identification Layer**

The architecture introduces the identification layer to provide stable and reliable identifiers to end-hosts over the Internet. The new layer is located between the network and transport layers, detaching end-host identification from its location. As a result, the IP semantic overload is solved, resulting in the native deployment of the mobility feature. These identifiers are based on the cryptographic identifiers proposed in [2] and have special properties to allow entities to self-claim their identification. The end-host identifier is represented by its public key and the hash of public key is a 128-bit end-host network identifier (ID). As a result, the identification layer identifies the end-hosts, and the network layer is strictly responsible for the packet delivery service and does not identify end-hosts anymore.

**Handover Support**

Although the mobility feature comes natively from the identity-locator split, additional mechanisms are required to fully support mobility. One of these mechanisms is the location management of the mobile nodes. Mobile nodes must have a common place to update their location after a mobility event, maintaining its reachability over the Internet. The location management is achieved through the use of the Domain Name Service (DNS) and a rendezvous mechanism responsible for the node’s location storage. Some proposals [15] consider the DNS as not suitable for dynamic information storage, e.g. mobile node’s IP address, since DNS relies heavily on caching. Dynamic networks, where nodes move in a short period of time, may require zero TTL DNS registries, increasing the load on root servers. This is a trade-off between scalability and fast locator updates. Since the DNS was not conceived for mobile environment, we propose the separation of the name resolution from the locator update mechanism. In order to overcome the DNS limitations, the architecture uses a Distributed Hash Table (DHT) located in the Internet to provide the location management feature. The DHT provides a distributed database that can be accessed from anywhere and is also fault-tolerant, robust and implements a standard API.

Whenever a mobile device bootstraps or moves, the node updates its current IP address in the nearest DHT server in order to be reachable after the mobility event. One optimization to avoid constant queries in the DHT for the current destination end-host locator is the usage of identifier to locator mapping caches. The cache-based mechanism is particularly interesting in vertical handover scenarios where mobile nodes receive a flow of data using stateless protocols, such as UDP. Normally the mobile node does not send any piggyback message to the peer node transmitting the data flow, since it is unidirectional. Thus, the peer node is not aware about any mobility events of the mobile node. With the timeout mechanism, the peer node periodically queries the DHT for a updated mapping of the mobile node’s locator, maintaining a correct ID to locator association even when the mobile node moves. There is also another advantage in this mechanism when stateful protocols are employed, such as TCP. As TCP is bidirectional, both nodes exchange packets that can be used to update their peer’s ID mapping by inspecting the current mapping in the identification layer. Thus, both mobile nodes reduce the number of queries at the DHT and the signaling needed to maintain the end-to-end connectivity.
Transparency

Legacy applications are transparently supported through legacy name resolution proxies and legacy packets interceptors. The former one intercepts all name resolutions from legacy applications and translates to identifiers resolution. Given a FQDN of an end-host, the proxy will request a TXT type registry in the DNS containing the destination ID and the destination Gateway ID (IDWG) of the end-host. Based on the type of the name resolution (A or AAAA), the proxy will return either a 32-bit identifier or a 128-bit identifier. The 32-bit identifier is derived from the ID and is used to support legacy IPv4 applications. The legacy packets interceptor is necessary since all legacy packets need an identification header to use the provided services. Legacy packets are captured by the Iptables module and inserted in our architecture. Every packet receives an Identification Header (IDH) and is sent to the destination end-host. Once arriving at the destination, the IDH is removed and the packet is delivered transparently to the legacy application.

Security

For the security requirement, we use cryptographic identifiers and secure mechanisms to add and modify entries in the DNS. Cryptographic identifiers enable transaction authentication, since identifiers can be checked against the public key provided by a Public Key Infrastructure (PKI) [16]. Secure insertion or modification in the DNS are provided by the adoption of the DNSSEC extension [17]. As DNS holds fairly static information about identifiers, nodes must authenticate before any modification in the DNS server, preventing attacks such as DNS cache poisoning. Mobile nodes establish secure connections during the bootstrap process or whenever a node arrives in a new domain. Nodes send their certificates to peer nodes to verify the authenticity, and Diffie-Hellman parameters are exchanged to establish a symmetric session key, providing end-to-end authentication and confidentiality.

IV. IMPLEMENTATION

This section presents the implementation of the identifier-based architecture. Figure 4 shows the architecture implementation proposal, mapping the proposed features into functional modules. The functionalities are divided in external and internal modules. The external modules are components providing services for the end-hosts. The internal modules are components providing services within each end-host, being divided in data and control planes. Data plane modules are involved in the data delivery, identifier management and flat routing. The control plane is responsible for signaling messages exchanged between components of the architecture, such as security associations and locator update messages.

The external modules provide services for the mobile nodes of the architecture, such as legacy name resolution, identifier to locator resolution and service discovery. The legacy name resolution is separated in two steps: the resolution of a name (FQDN) into the end-host identifier (ID) and the resolution of the identifier into one or more locators. The first step is performed by querying the DNS and the second step is done by querying the Distributed Hash Table (DHT) located in the Internet.

The internal modules provide services to support mobility, legacy applications and security. We discuss them through an example. First, legacy applications request a typical name resolution that is intercepted by the Legacy Application Support module. This module queries the standard DNS and returns the destination end-host identifier, acting as a proxy that translates standard queries (A or AAAA) into name to identifier resolution (type TXT). The DNS TXT type in our architecture returns the destination end-host identifier.

Legacy applications bind to the ID (or the 32-bit ID) instead of the locator and send these packets to the network (Fig. 5a). Such legacy application packets are captured by the Legacy Application Support module and sent to the Identification Layer module. This module inserts the identification header (IDH) and queries its internal cache to find the destination end-host identifier. The cache was previously populated by the Legacy Application Support module during the name resolution procedure. After receiving the IDH, the legacy packet is forwarded to the Security module, which queries for the security association bound to the target ID in the Security module. These security associations are exchanged using the SEC_CTRL message, employing the Diffie-Hellman algorithm to establish a shared key during the authentication phase. The security association also establishes a symmetric key to provide end-to-end integrity and confidentiality (Fig. 5b). After the packet is encrypted, it is forwarded to the Routing module, which resolves the ID to locator mapping in the local cache and sends to the network through IP + UDP encapsulation (Fig. 5c).

When packets arrive in the destination node, the reverse operation is performed by the Routing, Security and Ident-
Figure 5. Packets through the modules of the architecture.

V. EVALUATION

This section presents the prototype evaluation in different vertical handover scenarios. First we evaluated the prototype overhead in GSM and Wi-fi networks. Second, we evaluated the vertical handover support between GSM, Wi-fi and wired networks and analyzed the results. Finally, we evaluated the vertical handover support using Wi-fi, GSM and wired network in our campus.

The first prototype evaluation compared the prototype overhead in GSM and Wi-fi networks. We used one desktop server Pentium Core 2 Duo, 4 Gb RAM using Linux Debian and one notebook Pentium Core, 2 Gb RAM also using Linux Debian. In order to access the Wi-fi and GSM network, the notebook had a built-in 802.11ag wireless card and an external Sony-Ericsson MD300 GSM modem. The desktop had one interface connected to the Internet and the other one to an access point, as shown in Figure 6.

![Throughput measurement scenario.](image)

The first prototype evaluation compared the prototype overhead in GSM and Wi-fi networks. We used one desktop server Pentium Core 2 Duo, 4 Gb RAM using Linux Debian and one notebook Pentium Core, 2 Gb RAM also using Linux Debian. In order to access the Wi-fi and GSM network, the notebook had a built-in 802.11ag wireless card and an external Sony-Ericsson MD300 GSM modem. The desktop had one interface connected to the Internet and the other one to an access point, as shown in Figure 6.

The Iperf traffic generator\(^1\) was used to measure the total throughput in each network. For each measurement, we collected 25 samples and calculated the average value and standard deviation. Fig. 7 shows the total throughput without the prototype in GSM (85,46 Kbps) and Wi-fi (22,50 Mbps) networks, while using the prototype the throughput was 85,10 Kbps and 20,2 Mbps in GSM and Wi-fi networks respectively.

![Prototype overhead evaluation in GSM and Wi-fi networks.](image)

The experimental results show that the total overhead was 0,4% and 11,38% in GSM and Wi-fi scenarios respectively. These results are expected since the overhead of the prototype increases with the total number of packets encapsulated due to the processing overhead. Since the available bandwidth in GSM networks is lower than Wi-fi, fewer packets are encapsulated, resulting in lower overhead.

The second prototype evaluation analyzed the vertical handover between GSM, Wi-fi and wired networks in the topology illustrated in Fig. 8. In the Internet Core, we employed the Bamboo Distributed Hash Table (DHT) [18] to provide end-host identifier to locator resolution service. The desktop was directly connected to the Internet and the notebook has a built-in 802.11 wireless interface, a Sony-Ericsson MD300 GSM modem and a wired network interface attached to an access router to the Internet.

![Prototype evaluation in GSM, Wi-fi and wired networks.](image)

Step I evaluated the handover of the notebook from the GSM to a Wi-fi network (upward vertical handover). For all handover scenarios, the notebook started downloading a 2 MB file from the desktop server. Then, the notebook was

\(^1\)www.i-perf.org
forced to connect to other wireless technology (imperative vertical handover) and the handover time was measured. As future work we will implement a service to choose the best available network based on some criteria, such as higher data rate or lower cost. Finally, the notebook connected to the GSM modem and updated its current locator in the DHT to be reachable to other mobile nodes.

One optimization that occurred in our experimental evaluation was the automatic locator update of the notebook in the desktop due to the retransmission of the ACK message from the TCP. After the handover, the notebook sends ACK messages from the previously received data packets, updating the identifier to locator cache in the desktop. As identifiers cannot be forged, the server was secured to receive this kind of update.

Step II shows the vertical handover from a Wi-fi network to GSM (downward vertical handover). Step III shows the notebook initially connected to the desktop through a wired network then it switches to the GSM network. And finally, step IV shows the reverse path from the GSM to the wired network. Tab. I shows the average handover time of 25 samples for each described scenario.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Handover Time (s)</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I) GSM-&gt;WIFI</td>
<td>4.422303</td>
<td>0.940884</td>
</tr>
<tr>
<td>(II) WIFI-&gt;GSM</td>
<td>0.942619</td>
<td>0.100229</td>
</tr>
<tr>
<td>(III) Wired-&gt;GSM</td>
<td>0.997144</td>
<td>0.155440</td>
</tr>
<tr>
<td>(IV) GSM-&gt;Wired</td>
<td>4.634749</td>
<td>1.593503</td>
</tr>
</tbody>
</table>

Tab. I shows higher handover times for scenarios involving upward vertical handovers (GSM to Wi-fi and GSM to wired network) and lower handover times for the downward vertical handovers (opposite case). This result is expected due to the TCP ACK piggyback message during the transmissions. As the throughput of the Wi-fi and wired scenarios are higher than GSM networks, there are higher number of ACK messages to be transmitted from the notebook to the desktop, speeding up the location update procedure.

In order to analyze the traffic during the handover procedure, we used the Distributed Internet Traffic Generator (D-ITG) application to generate the traffic pattern in the four vertical handover scenarios. One sample of each vertical handover scenario was randomly selected and the traffic pattern was computed. Fig. 9 shows a data transfer from the desktop to the notebook when occurs an upward vertical handover. The graphic shows a traffic interruption between the interval time of 10 and 15 seconds and later it resumes normally.

Fig. 10 shows the handover from the Wi-fi network to the GSM. As previously mentioned, due to the higher data rates, the notebook transmits more TCP ACK messages, speeding up the convergence of the communication. Due to the faster convergence, there is almost no impact in the file transfer time.

Fig. 11 shows the handover from a wired network to the GSM. The figure shows a small decrease of the transmission rate near the 4 seconds due to the vertical handover.

Finally, Fig. 12 shows the handover from the GSM to the wired network. The Figure shows an interruption of the transmission due to the handover process between 10 and 15 seconds.

The third prototype evaluation analyzed the vertical handover support between Wi-fi, GSM and wired network in our campus, shown in Fig. 13. The evaluation scenario started with a notebook connected to a Wi-fi hotspot in the Faculty of Electrical and Computer Engineering (FEEC) and started to download a Linux distribution. Later the notebook was taken out of the laboratory performed an imperative downward vertical handover to connect to the GSM network due to the low signal power of the Wi-fi hotspot. Then we...
got into a car and drove approximately 1.2 kilometers to the Institute of Language Studies (IEL) in the campus while the notebook was downloading the file. Finally, we got in the laboratory and performed an imperative upward vertical handover to connect to the wired network, finishing the file download. During the evaluation, the file was downloaded in the Wi-fi, GSM and wired networks without interruption. The prototype supports this scenario because legacy applications bind to the identification layer, and the network layer is free to change its address from one network (e.g. Wi-fi) to another one (e.g. GSM).

VI. DISCUSSION

In this section, we compare our architecture with other approaches, outlining the main differences between them. First, Mobile IP provides vertical handover support, but does not solve the semantic overload problem, using IP addresses as end-host identifiers. Hence, these IP addresses do not indicate the correct end-host location in the network, resulting in errors for applications that collect information regarding its current latency to a target destination. The Takeover protocol lacks security mechanisms to protect the handover process since it relies on the goodwill of the neighbor node to correctly perform the handover process. The USHA proposal employs fixed and virtual addresses, but also does not propose a mechanism to ensure the authentication of the new locator after a vertical handover procedure. The main difference between the approaches described above and the architecture presented in this paper is that our identifier-based architecture overcome these limitations by introducing a new layer between the network and transport layers to solve the dual semantic of the IP address. Also, our architecture has native security mechanisms inherited from the cryptographic identifiers, providing end-host authentication after the handover event.

The architecture presented in this paper supports native handover with network connectivity and application functionality. The first feature is the most basic one and several architectures support it. However, the native handover with application functionality is supported by only few proposals. One of the most known is the Mobile IP protocol that uses an anchor point (Home Agent) to give mobility transparency to the applications. In our proposal, it is not necessary to have such an anchor point since all the features are supported by the architecture itself through the identification layer. This makes the mobility management easier and paves the path
towards a next generation mobility framework where the functions are built-in artifacts of the architecture.

VII. CONCLUSION

This paper presented a next-generation internetworking architecture to provide native vertical handover support. The architecture solves the IP semantic overload problem by introducing the Identification Layer, detaching the end-host identification from its location in the network (IP address). Also, the architecture employs cryptographic identifiers to provide stable and reliable identifiers for the end-hosts over the Internet. These identifiers provide strong binding between the end-host identifier with is identification, since it is based on the public key cryptography. Legacy applications are also transparently supported by the architecture without any modifications in their source code.

To validate the proposed architecture, we implemented a prototype and performed a set of tests analyzing the vertical handover between Wi-fi, GSM and wired network. The prototype had an overhead of 0.4% and 11.38% in GSM and Wi-fi scenarios respectively. The proposed architecture can also be incrementally deployed over the Internet. Mobile hosts interested in joining the architecture need to register their identifiers in the DHT to communicate with each other. After joining the identifier-based overlay network, they are able to use all the proposed benefits of the architecture without any modifications.

ACKNOWLEDGMENT

The authors would like to thank CAPES and Ericsson for their financial support.

REFERENCES


