



Copyright © IEEE.
Citation for the published paper:

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of BTH's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by sending a blank email message to pubs-permissions@ieee.org.

By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

PERIMETER: A User-Centric Mobility Framework

Karel De Vogeleer, Selim Ickin, David Erman, Markus Fiedler
School of Computing
Blekinge Institute of Technology
371 79 Karlskrona, Sweden
Email: {kdv, sic, der, mfi}@bth.se

Abstract—This demo shows a prototype of a user-centric mobility framework that provides handover for macro-mobility on handheld devices. The framework is designed for mobile-controlled handover and does not require modification of the Internet infrastructure. The end-users are so able to control the roaming process governed by user considerations in addition to business objectives.

To achieve seamless mobility UDP tunneling is used as a basis for the handover process. Additionally, measurements are performed on the tunnels to acquire Quality of Service (QoS) metrics and hence serve for roaming decision making processes.

I. INTRODUCTION

Mobility becomes increasingly popular. The advent of internet-enabled mobile devices such as smart phones and netbooks plays a major role in this. Market studies show that even in times of crises the demand for smart phones is not affected. Also, wireless networks are deployed at a fast pace. Not only because the end-user's demand for wireless access increases, but also the fact that wireless networks are more easily deployed than wired equivalents.

When traversing multiple geographic locations, mobile users might roam between networks. In order to maintain connectivity to the Internet and/or to maintain or increase their Quality of Experience (QoE), *i.e.*, their satisfaction. A mobility framework can assist the mobile user in acquiring context information and help the user to always provide the optimal connectivity with regards to the user's needs, *i.e.*, make them Always Best Connected (ABC).

Such mobility frameworks have been subjected to research for an extensive amount of time. Research papers can be traced back to the beginning of the '90s. Yet, up till today the mobile user does not have access to true mobility. It seems that Internet Service Providers (ISPs) are reluctant to provide true seamless mobility. Network-centric mobility solutions, however, are well studied and developed, *e.g.*, Mobile IP (MIP) [5], but not suitable for our case.

The mobility framework demonstrated in this demo is part of PERIMETER, a STREP project granted by EU FP7 [7]. PERIMETER's main objective is to provide a new paradigm for user-centric mobility in a multiple-access multiple-operator environment. [4]. PERIMETER tackles the seamless mobility problem from a user-centric perspective. As a result, the mobile-users are able to control their seamless roaming process governed by the user's considerations in addition to business objectives.

II. ARCHITECTURAL OVERVIEW

The architecture of the framework is depicted in Fig.1. Applications communicate over a regular TCP/IP stack. This way the mobility framework is transparent to the application layer and hence, legacy applications are able to be deployed on top of the framework. All network traffic is routed into a virtual interface, which is controlled by our mobility framework.

The virtual interface separates the TCP/IP stack from the mobile world, referred to as a Mobile Virtual Network (MVN). The interface provides a fixed point of attachment, an Internet Protocol (IP) address, for applications while the underlying physical network interfaces can change. Changing interfaces can occur when one roams to another network available on a different internet access technology, *e.g.*, from WLAN to 3G.

The virtual interfaces exchange data with each other via User Datagram Protocol (UDP) tunnels. A handover is then realized by relaying the UDP communication to another interface. UDP can handle changing interfaces gracefully as it is a stateless protocol. A binary signaling protocol is deployed on the Transport Layer to aid the handover process.

The UDP tunnel provides similar functionalities as Stream Control Transmission Protocol (SCTP) [6] with regards to multihoming support and handover signaling. UDP tunnels are, however, legacy application compatible and can penetrate NATs.

The UDP tunnel header is extended with measurement related information, which include a *timestamp* and *sequence number* [3]. The header fields are used to compute QoS metrics and can predict the satisfaction of the mobile users:

- Packet Delay Variation (PDV) is tracked and computed by timestamps from each incoming packet.

$$J = \sqrt{\frac{1}{N-1} \left[\sum_{n=1}^N (D_n^2) - N\bar{D}^2 \right]}, \quad (1)$$

where J is the PDV, D_n the one-way-delay, \bar{D}^2 the average delay, and N the number of processed packets. J is updated each time a packet arrives. Generally speaking, the larger the PDV the lower the QoE.

- Packet Drop Rate (PDR) tracks the packets lost per time unit. For acceptable QoE, the PDR must be minimized.
- Maximum Burst Size (MBS), the application layer seems not being able to cope gracefully with large data bursts. Avoiding burst is thus advised to maximize the QoE.

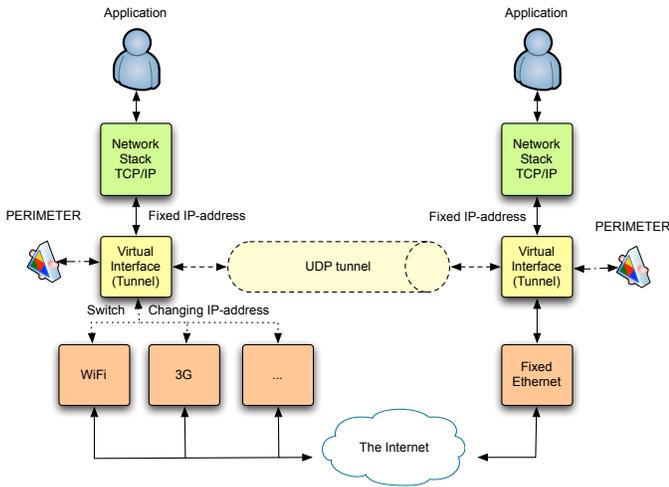


Fig. 1. Architectural overview of the mobility framework. Applications communicate over a conventional TCP/IP stack while the virtual interface separates the stack from the mobile world. The virtual interfaces communicate with each other over a UDP tunnel. Seamless handover is then achieved by relaying the tunnels over another network interface.

With this information, decision makers can predict the future network state and decide to take proper actions when network performance deteriorates. Such an action could be a vertical handover.

III. IMPLEMENTATION

Our prototype is implemented on a Linux platform, and is able to run on Google's Android [2] platform. The motivation for Linux is rather straightforward; the operating system is accessible, *i.e.*, has an open-source network stack, and thus allows for relatively easy modifications to the Transport Layer. In addition to this, Android provides a useful set of tools to manage the handheld devices' resource.

To provide seamless handovers, handovers must minimize overhead to avoid packet losses and service interruptions. Therefore we opted to implement our vertical handover algorithms in the Linux kernel as a module, a virtual network interface. This approach minimizes changes to the original path that data would take when traversing the network stack and decreases processing load as context switching is minimized compared to conventional user-space VPN solutions.

The virtual interface is partly based on the IP tunneling driver. All outgoing traffic passing through the virtual driver is multiplexed into the appropriate tunnel with the aid of a handover routing table. Incoming traffic is similarly collected by the virtual interface and forwarded to the higher layers in the network stack. To do this, the multiple routing table feature must be enabled in the Linux kernel to allow for connections over multiple interfaces. By default this is disabled in Android even though it is a multihoming device. Hence, the Linux kernel must be recompiled with the multiple routing table setting.

Communication between the framework, residing in the kernel, and any controlling process in user-space happens through a UDP socket interface or the `/proc` file system.

IV. DEMO

In this demo we demonstrate user-centric vertical handover including monitoring QoS metrics of ongoing connections. We demonstrate this by streaming videos from a streaming server to a mobile node over the Internet. The mobile node is connected to the Internet through 3G, Fixed Ethernet and/or WLAN. We show that we are able to retrieve the QoS of ongoing connections and switch the communication from WLAN to 3G and visa versa, *i.e.*, vertical handover, without interruption of the video stream.

The Darwin Streaming Server (DSS) framework [1] is used for streaming the video to the Internet. The videos used for streaming are Moving Picture Experts Group 4 (MPEG-4) compressed with different dimensions, frame rates and bit rates. The DSS uses the Real Time Streaming Protocol (RTSP) protocol for video dissemination.

V. CONCLUSION

In this demo we demonstrated the feasibility of a user-centric mobility framework. As ISPs do not provide true seamless mobility, a user-centric mobility framework is desirable.

Mobility is achieved by tunneling data over the UDP protocol which is prone to failure when switching interfaces during handover. QoS metrics are observed during communication to predict network performance and detect network failures. In such a way our mobility framework is able to achieve the ABC principle.

We proved that the prototype is working, however, there are many opportunities to improve the system. For example some parts in the low-level code could be optimized to yield better performance, *e.g.*, IP address lookups in tables can be accelerated. Moreover, decision making criteria must be optimized and fine-tuned. Also, the system is prone to security issues, which must be addressed. Handover signaling is for example not encrypted nor authenticated and hence open for malicious activities.

ACKNOWLEDGEMENT

We would like to thank PERIMETER, a STREP project, granted by EU FP7, and MOBICOME, a EUREKA project, granted by VINNOVA for the funding of our research in vertical handover technologies.

REFERENCES

- [1] Apple. <http://developer.apple.com/opensource/server/streaming>, 2010.
- [2] Google. <http://www.android.com/>, 2009.
- [3] Selim Ickin, Karel De Vogeleer, Markus Fiedler, and David Erman. The effects of packet delay variation on the perceptual quality of video. In *The 4th Workshop On User MOBility and Vehicular Networks 2010 (IEEE LCN On-MOVE)*, Denver, Colorado, USA, 2010.
- [4] Perimeter. <http://www.ict-perimeter.eu/>, 2009.
- [5] C. Perkins. IP Mobility Support for IPv4. RFC 3344 (Proposed Standard), August 2002. Updated by RFC 4721.
- [6] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson. Stream Control Transmission Protocol. RFC 2960 (Proposed Standard), October 2000. Updated by RFC 3309.
- [7] Karel De Vogeleer, David Erman, Markus Fiedler, and Adrian Popescu. A lightweight macro-mobility framework. In *2nd International ICST Conference on Mobile Lightweight Wireless Systems*, May 2010.