Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 40 (2014) 26 – 36

Procedia Computer Science

2014 International conference on selected topics in Mobile and Wireless Networking

Client-based and Cross-Layer Optimized Flow Mobility for Android Devices in Heterogeneous Femtocell/Wi-Fi Networks*

Norbert Varga[a], László Bokor[a*], Stephane Bouroz[b], Benoit Lecroart[b], András Takács[c]

[a]*Budapest University of Technology and Economics, Department of Networked Systems and Services (HIT), Multimedia Networks and Services Laboratory (MediaNets), Magyar Tudósok krt. 2, H-1117, Budapest Hungary*

[b]*NEC Technologies (UK), Immeuble Le capitole 55 avenue des champs pierreux, 92000 Nanterre, France*

[c]*Hungarian Academy of Sciences, Computer and Automation Research Institute, Kende u. 13-17, H-1111, Budapest, Hungary*

Abstract

The number of subscribers accessing Internet resources from mobile and wireless devices has been increasing continually since i-mode, the first mobile Internet service launched in 1999. The handling and support of dramatic growth of mobile data traffic create serious challenges for the network operators. Due to the spreading of WLAN networks and the proliferation of multi-access devices, offloading from 3G to Wi-Fi seems to be a promising step towards the solution. To solve the bandwidth limitation and coverage issues in 3G/4G environments, femtocells became key players. These facts motivate the design and development of femtocell/Wi-Fi offloading schemes. Aiming to support advanced offloading in heterogeneous networks, in this paper we propose a client-based, cross-layer optimized flow mobility architecture for Android devices in femtocell/Wi-Fi access environments. The paper presents the design, implementation and evaluation details of the aforementioned mechanisms.

Peer-review under responsibility of organizing committee of Fourth International Conference on Selected Topics in Mobile & Wireless Networking (MoWNet'2014)

Keywords: Femtocell/Wi-Fi offloading, Android, MIPv6, Flow Bindings, decision engine, heterogenous networks, evaluation

* Corresponding author. Tel.: +36-1-463-2048; fax: +36-1-463-3263.
E-mail address: bokorl@hit.bme.hu

1877-0509 © 2014 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/3.0/).
Peer-review under responsibility of organizing committee of Fourth International Conference on Selected Topics in Mobile & Wireless Networking (MoWNet'2014)
doi:10.1016/j.procs.2014.10.028

1. Introduction

Recently the number of mobile users accessing wireless and mobile Internet services has been increasing spectacularly. The overall mobile data traffic is expected to grow nearly 11 fold between 2013 and 2018 [1]. Network operators must adapt for this traffic explosion. The traditional mechanisms to expand the capacity of the network require high-cost, large-scale modifications; however, the goal of the network operators is the optimization of usage of the available network resources with low-cost investments. Due to the spreading of WLAN networks and the proliferation of multi-access (3G/4G and Wi-Fi) mobile devices network operators are capable to design cost-effective resource management strategies based on data offloading from 3G/4G to Wi-Fi. These strategies could be even more efficient if decisions are made on the user flow level, based on application requirements in means of Quality of Service (QoS) and/or Quality of Experience (QoE) [2]. However, dynamic migration of ongoing sessions between different radio access technologies requires special mobility management solutions. These mechanisms can be divided into two main groups, namely network-based and client-based approaches [3]. Client-based approaches provide higher level of freedom to users by allowing them completely maintain every aspect of handover decision and execution based on the context information available at the terminal side. On the contrary, in case of network-based mobility management the overall control falls into the hands of operators: by decreasing freedom of choice at the user side, network and traffic management can be enhanced from a completely operator point of view. This work considers a client-based solution.

Network efficiency can be further increased by reducing the mobile data traffic in 3G macro network segments using femtocells. Femtocells are able to expand cell coverage and extend radio resources. As integrated femtocell/Wi-Fi networks are getting more and more widespread, femtocell/Wi-Fi offloading schemes will also come into picture. Femtocell/Wi-Fi offloading is able to move data traffic from femtocell radio interface to Wi-Fi interface and with this scheme network operators can alleviate the load of the network while also providing higher data rates to the end users [4]. The growth of heterogeneous and overlapping wireless access networks demand to design and implement aforementioned algorithms, which are able to exploit the available network resources. These facts motivated us to design and develop an extensive, client-based, flow-aware, cross-layer optimized mobility management scheme to Android Smartphones, and evaluate our proposed mechanism in a femtocell/Wi-Fi based testbed environment.

The rest of paper is organized as follows. In Section II we present the related work on the existing solutions for cross-layer optimized flow mobility in femtocell environments. Section 3 introduces our client-based, cross-layer optimized offloading scheme in details. Also the overall femtocell testbed setup is depicted. Section 4 introduces our measurement scenarios and performance evaluation results. In Section 5 we conclude the paper and describe our future work.

2. Background and Related Work

Mobile Internet traffic is growing dramatically due to the penetration of multi-access smartphone devices, data-hungry mobile entertainment services like video, music, games, and new application types, such as social media, M2M (machine-to-machine) and C-ITS (Cooperative Intelligent Transport Systems) [5]. This fact forced the standardization bodies to design and develop wireless standards, namely 3G UMTS, LTE, LTE-A, WiMAX, 802.11n/ac/ad WLANS, etc. The complementary characteristics of the above architectures motivate network operators to integrate them in a supplementary and overlapping manner. The data traffic offloading between 3GPP (Third-Generation Partnership Project) access networks and WLAN networks has been recognized as a key mechanism to exploit the available network resources in an efficient way. Authors of [6] introduce the basic conception of 3G/Wi-Fi seamless offloading and an application layer based switching scheme. In [7] the architecture and the protocol stack of an Ethernet-based offloading technology and a testbed environment are presented. Also the measurement results of their system are introduced here. Local IP Access (LIPA) and the Selected IP Traffic Offloading (SIPTO) solutions can also play important roles at network operators in realizing cost-efficient offloading techniques [8]. In [9] the two aforementioned technologies are discussed, however this paper only introduces theoretical results and doesn't contain real implementations and measurements. Likewise the Multi Access PDN Connectivity (MAPCON) [10] provides a solution, which allows the mobile terminal to establish more PDN connections to different access networks (3GPP and also non 3GPP accesses are supported).

The articles above present recommendations for data offloading mechanisms between 3G/4G and Wi-Fi. To expand the radio resources of legacy cellular mobile networks the femtocells are turning into a promising solution. Several researches in the femtocell topic have been published. The vast majority of these articles discuss the architecture of femtocells and the detailed mechanism of handovers between femto and macro cells (e.g., [11], [12]). Jaehoon Roh et al. [13] propose multiple femtocell traffic offloading scheme and analyze the performance of the proposed scheme. However, our scope within this work is more close to existing femtocell/Wi-Fi offloading schemes and the flow-aware decision algorithms. The IP flow mobility and seamless Wireless Local Area Network offload (IFOM) standard in Rel-10 [14] has been created to ensure fine-grained and seamless offload strategies. IFOM has been designed to define different IP flows belonging to the same PDN connection and registering to different network interfaces. Benefits of IFOM can be exploited efficiently only if the User Equipment (UE) is capable to communicate via 3GPP access and WLAN simultaneously. This solution is based either on Dual-Stack Mobile IPv6 as per 3GPP Rel-8 (DSMIPv6) [15], thus the IP address preservation and session continuity for mobile users in the course of their movement is guaranteed during the movement of the UE. Also Proxy Mobile IPv6 (PMIPv6) could be used for IFOM purposes [16], however it has not yet been standardized. Further optimization can be achieved by using intelligent decision engines, which are capable to assign application flows to the appropriate interface. In Rel-8 the Access Network Discovery and Selection Function (ANDSF) assists to the UE to discover wireless access networks and provides routing policies, rules and discovery information to facilitate the appropriate network selection for the UE ([17], [18]).

All the mechanisms introduced above are designed to manage handovers by the network operators. Contrarily we designed and implemented a client-based mobility management scheme based on MIP6D-NG [19], which is a client-based, multi-access Mobile IPv6 implementation with various extensions and an advanced cross-layer communication API. Both the network discovery mechanism and the flow mobility algorithms are handled by our highly customized Android Smartphone having MIP6D-NG integrated; the network operators have no influence on the selection and decision process. The first publicly available Flow Bindings implementation was designed for Linux distributions by the authors of [20], however their implementation supported only NEMO [21] environments, regular mobile nodes were not able to register or update network flows. Most of the papers in the subject discuss the definition and management of different flows in protocol level [22]. In our solution the advanced toolset of MIPD6-NG solves all the protocol level questions of flow mobility management by relying on the MCoA [23] and Flow Bindings [24] RFCs, so we do not detail them in this paper. Instead, we focus on the flow-aware offloading schemes based on a built-in decision engine. In [25] a multi-criteria decision engine is presented based on network cost, signal strength, packet loss and predefined weight of the flows, however this paper introduces only theoretical results and doesn't contain evaluation based on real implementations. Francois Hoguet et al. [26] showed a Linux based flow mobility environment on the basis of UMIP's MIPv6 implementation† and the possibilities of porting it to

† UMIP: http://www.umip.org/

Android Smartphones. Although their paper introduces a real implementation, it does provide neither flow mobility management nor complex decision engine. Ricardo Silva et al. [27] examine the mobility management on Android systems. They created a custom Android ROM to use the 3G and Wi-Fi interfaces simultaneously. IEEE 802.21 Media Independent Handover framework [28] is applied to support IPv6 based mobility. From this article also the flow mobility and the flow based decision mechanism are missing compared to our architecture.

3. Cross-Layer Optimized Offloading Scheme

Fig. 1 presents the architecture of the proposed highly customized Android-based system, where cross-layer information transfer plays an essential role. We introduce each part of the system in a bottom-up approach.

Our architecture requires special kernel configuration extended with Mobil IPv6 support, MIP6D-NG patches and also kernel module modifications. To apply these changes in the kernel level of the system we had to recompile the whole kernel source. In Section 4.1 we present this work in details.

As Fig. 1 shows the native layer contains all of the used, cross-complied native binaries and associated libraries such as Lighttpd, Pingm6, and Socat. Also MIP6D-NG binaries and libraries are located in this layer.

For multi-access communication, the Mobile Node (MN) needs the ability to communicate via two (3G and Wi-Fi) network interfaces (with IPv6 support) simultaneously, however even the newest Android OS versions (Android 4.4) do not allow the simultaneous usage of them. This fact forced us to modify the application framework layer. An overall Android OS and kernel source code recompiling is required to apply aforementioned modifications (see details in Section 4.1).

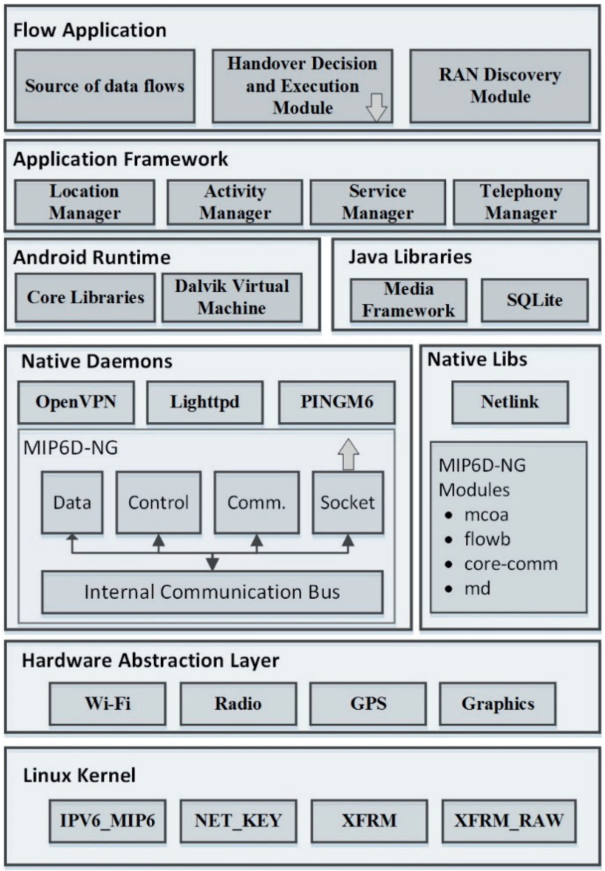

Figure 1 - The proposed Android architecture for Femto/Wi-Fi offloading

In the Java layer we devised and implemented a modular Android application comprising three main parts. The first is the so-called Radio Access Network Discovery Module (RANDM), which is designed to measure the different parameters from multiple layers of the available networks (e.g., signal strength, delay, and packet loss). The Handover Decision and Execution module (HDEM) can be divided into two parts: Handover Decision (HDM) and Handover Execution Module (HEM). HEM communicates with the native MIP6D-NG daemon, creates and sends flow register and flow update messages [20] induced by the advanced decision algorithm. The register message allocates and initializes a new flow entry to the selected network interface while the update message updates an existing flow by the Flow Identifier (FID). For the cross-layer information exchange a socket based communication scheme was designed and developed. HDM decides about the necessity of the handoff based on the decision algorithm (see later in more detail). HDM directs the HEM to send flow register or update command to MIP6D-NG. The HDM is a modular, exchangeable part of the architecture, thus we can modify the offloading decision scheme easily.

Fig. 2 presents the operation of our cross-layer optimized offloading scheme incorporating the decision algorithm. The most important input parameters are the actual measurement data, the static information obtained during the network measurements in the currently used networks, and the user preferences.

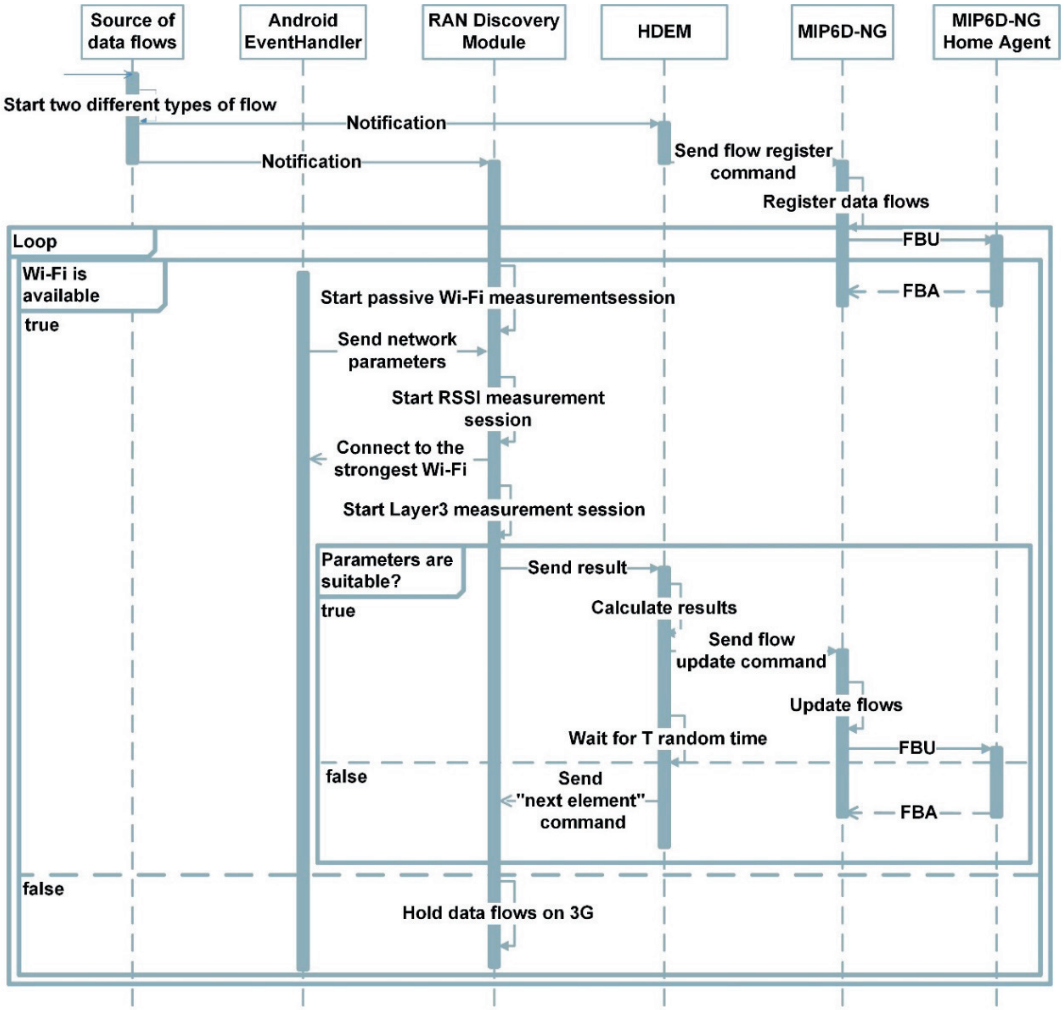

Figure 2 - The proposed cross-layer optimized flow mobility mechanism

The first step of the algorithm is checking the available wireless access networks. The default interface is the 3G access, therefore the system registers data flows to the 3G interface using cross-layer communication between the application and the network layers. After this step and if there is at least one available Wi-Fi network, the algorithm starts the phase of passive measurements of Wi-Fi networks. If there are no available WLANs, the algorithm holds the flows on the 3G interface and waits for the appearance of new Wi-Fi access points. Otherwise starts the cross-layer measurements, in which it measures the signal strength from link-layer, and packet loss, RTT and jitter from network layer. If our decision engine does not find the parameters of the current measured network suitable for the application flow's QoS profile, the scheme starts to measure the next available network. If the measured QoS values are appropriate, the MN connects to this Wi-Fi network and moves the corresponding flows to the Wi-Fi interface based on the flows QoS profiles. After that, the application waits for a random time to avoid the ping-pong effect similarly to the solution applied in [29]. (Note, that in situations where the MN moves around the border of wireless accesses, a series of unnecessary handovers may occur during a very short time, such creating the so-called ping-pong effect). Cross-layer communication mechanism allows us to trigger and execute flow updates in a different layer of the stack which further increases the efficiency of our system. In each case when MIP6D-NG executes a flow registration or update, sends a Flow Binding Update (FBU) message to Home Agent, who sends back a Flow Binding Acknowledgement (FBA) message according to [24]. The third and last part of the Java layer application is the Source of Data Flows which serves as a simple traffic generator: produces an UDP audio stream and/or a TCP file transfer.

4. Overall Testbed Architecture

Fig. 3 presents the overall architecture of our testbed environment designed and implemented for real-life femtocell-based evaluation of advanced cross-layer optimized, flow level mobility management protocols and algorithms. In this section we introduce the main parts of our testbed.

4.1. Mobile Node

In our proposed testbed environment the UE entity is realized by an Android Smartphone, namely a HTC Desire S device. This Smartphone must be able to run the MIP6D-NG daemon and requires special kernel environment. Therefore we modified the kernel part with the required extensions referred in Section 3. The porting of MIP6D-NG to Android systems was a non-trivial task, because it required libraries and header files that do not exist on Android OS or if exists, differ from their original GNU Linux implementations. To make up the missing requirements we created a cross-compiler toolchain which contains the ARM compatible versions of all the necessary components. We made the aforementioned compiler pack based on the NDK stand-alone toolchain and extended with our own libraries and header files. MIP6D-NG requires multi-access communications via two (3G and Wi-Fi) network interfaces (with IPv6 support) simultaneously. Despite the fact that recent Android devices usually possess multiple radio interfaces, the Android OS is currently pushing a solution which saves battery power so only one interface can be active at the same time. In fact, the built-in mechanisms for network interface management in Android phones are very simple: if a 3G interface is active and Wi-Fi is available, the 3G will shut down, while if only a 3G network is available, then the Wi-Fi interface will be in down state. To change the mechanism described above it was necessary to modify the source code of the Service module of the Android OS managing network connections. The Service module contains the ConnectivityService.java where the NetworkStateTrackerHandler class is responsible for the state management of network interfaces: a switch-case statement contains the implementation of each scenario. We implemented a new statement as an extension: if the 3G interface is active and Wi-Fi is available, then 3G should remain active, therefore real multi-access became usable. It meant that the Android OS itself also required modifications. Another issue to be solved was that the 3G interface doesn't support native IPv6 on most Android devices. In order to solve this problem we configured an OpenVPN connection with a bridged interface on the Android Smartphone. The OpenVPN server is located on a router, which provides an appropriate IPv6 prefix for the Android Smartphone's 3G interface through the OpenVPN tunnel.

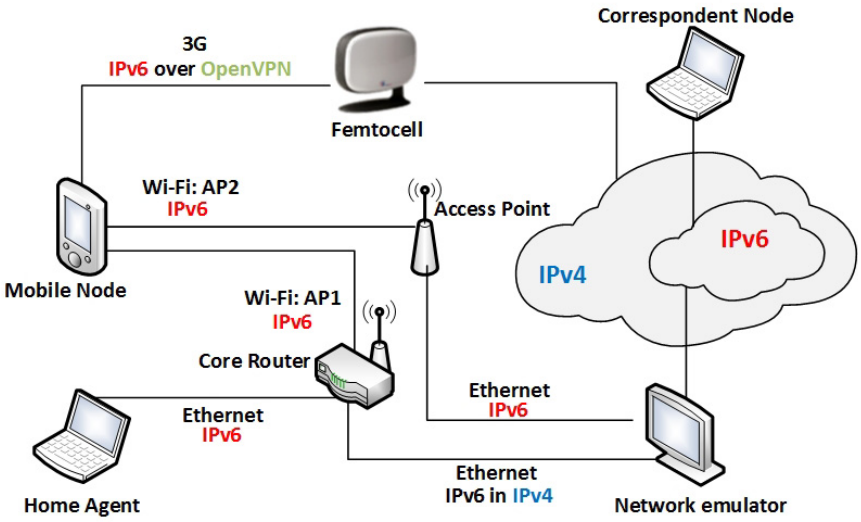

Figure 3 – Overall testbed setup

In order to perform the required modifications inside the source code of the Android OS and the kernel, a build environment was created in which we were able to make a custom ROM image with our MIP6D-NG ready kernel source code and with our modified Android OS code. We used CyanogenMod[‡] Android sources and Andromadus[§] kernel tree distribution as a base code platform for our extensions. The result is a highly customized Android 4.1.2 and Kernel 3.0.57 with the appropriate patches and settings.

To measure the different parameters of the network in the Java layer we use built-in APIs, and external binaries. TelephonyManager API provides the signal strength. The packet loss and delay are calculated from the output of Pingm6. To run Pingm6 (which is not a so called system binary but a part of the MIP6D-NG distribution package) from Java layer we had to use an external library, the RootCommands. The HEM module of our application is able to direct the Android OS to connect an available WiFi network using the WifiConfiguration and WifiManager APIs.

4.2. Femtocell

The femtocell in the architecture provided 3G access over UMTS band 1 with HSPA support. It supports HSUPA up to category 6 allowing 5.76 Mbps max. in uplink, and HSDPA up to category 10 allowing 14Mbits

‡ CyanogenMod github: https://github.com/CyanogenMod

§ Andromadus github: https://github.com/Andromadus max. in downlink. The applied solution is typically used by network operators as a residential gateway to extend network coverage inside a building either for an enterprise or an end users home. Its transmission power can be set up to 5dBm. The femtocell within our architecture is controlled by its own network infrastructure that is embedded in an associated PC (not depicted in Fig. 3). The controller software provides the 3G core network infrastructure features for authenticating the subscribers (i.e., the mobile devices), for providing voice call services, and for accessing to the IP networks.

4.3. Home Agent and Core Router

The essential role of the Home Agent entity is to manage flow bindings for MN Care-of Addresses (CoA) and Home Address (HoA) according to the MCoA standard [23]. In the used terminology the flow is defined as a set of IP packets matching a traffic selector. A traffic selector can identify the source and destination IP addresses, transport protocol number, the source and destination port numbers and other fields in IP and higher-layer headers. The different flows are referred by the Flow Identifier (FID), which is a unique identifier. MIP6D-NG routes the incoming (outgoing) packets from HA (MN) to MN (HA) based on the defined router and rule policies for the diverse flows [24]. The Home Agent is realized by a Dell Inspiron 7720 notebook running a MIP6D-NG daemon configured for Home Agent functionality. This entity requires special kernel configuration, which means the need of a MIP6D-NG compatible kernel.

In our testbed the core router is an ASUS WL500 with DD-WRTv24 OS (CrushedHat distribution). Two OpenVPN daemons are running on this router. On one hand an OpenVPN Server provides an appropriate IPv6 address for the 3G connection of Android Smartphone using RADVD. On the other hand an OpenVPN client operates as an IPv6 over IPv4 or IPv6 over IPv6 tunnel, interconnecting the testbed with our University's IPv6 network, independently of the router's actual IP access. It means that the overall architecture could be portable and in the worst case only recovers legacy IPv4 connection for the core router. Wanulator network emulator node is also applied in the environment. This entity is a Linux distribution which allows us to manipulate the QoS parameters (e.g., delay, packet loss, jitter etc.) of the link to which it is connected (i.e., the Wi-Fi connections in the depicted setup). Using Wanulator we were able to evaluate different decision algorithms in any set of network QoS parameters.

5. Measurement scenarios and results

In order to present the feasibility of our scheme and to evaluate the proposed offloading algorithm, we implemented two measurement scenarios in the testbed. In the first scenario we measured the throughput of a HTTP video stream (H.264 encoded video with 854x480 resolution originated by a Lighttpd Webserver) over TCP transmitted from the mobile node towards a correspondent node, with and without cross-layer mobility support. We defined and registered two different types of data flows: the HTTP video stream over TCP and a VoIP flow over UDP. According to the QoS policies, the UDP flow is routed through the 3G interface during the entire measurement session, however the TCP flow is moved by our decision engine between the 3G and Wi-Fi accesses. The first and the second boxes on Fig. 4 (with captions of Wi-Fi and Femtocell) depict reference scenarios where both TCP and UDP flows are transferred via Wi-Fi or 3G respectively, without any flow handover initiated. Contrarily in the third box the application moves the TCP flow from 3G to Wi-Fi after 30 seconds (and from Wi-Fi to 3G in the fourth box). The measurement session took 90 seconds, the average bandwidth of the used Wi-Fi network was 300 KBps, and the femtocell was able to provide 100 KBps. The average bandwidth was calculated as the ratio of the amount of the transmitted data and the elapsed time. As Fig. 4 shows the execution of the vertical handover does not generate significant reduction of the throughput. If both TCP and UDP flows were assigned to the Wi-Fi interface the quality of VoIP and video stream was deteriorated, thus the separation of flows to different interfaces improved the quality of aforementioned applications. In this scenario the operation of the proposed flow mobility management scheme was the following between MN and HA:

- The proposed framework registered both TCP and UDP flows to the interface using the Femtocell network.
- MN sent Flow Binding Update (FBU) messages to the HA, which created the appropriate Binding Cache, flow policies and rules.
- After 30 seconds the decision module of our application moved only the TCP flow from Femtocell network to Wi-Fi.
- MN sent FBU to HA, HA update the current entities of Binding Cache, policy and rule tables.

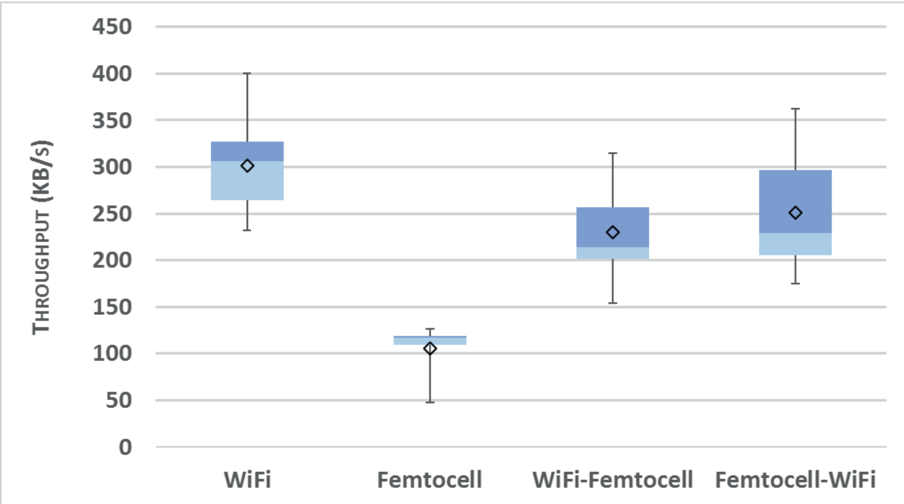

Figure 4 - Throughput measurement results

Fig. 5 presents the components of the average vertical handover latency measured during flow mobility events. The two considerable parts are the latency between the Java and the native layer, and the delay between the reception of the flow update command and the sending of FBU message towards the HA by the MIP6D-NG daemon. In our testbed the aforementioned latency values are quite significant thanks to performance limitations of the used HTC Desire S device: this is model is an old-version Smartphone (announced in 2010) optimized to Android 2.2. However, our modifications resulted in a highly-customized Android 4.1.2 requiring a lot more resources from the Smartphone hardware. This quite high latency doesn't imply serious practical issues, because during these procedures the registered flows will not be affected: data transfer will harmlessly continue till the third phase (FBU->FBA signaling) starts. This last component is the delay between the sent FBU and the received FBA messages. The average latency of this component is less than 1 seconds.

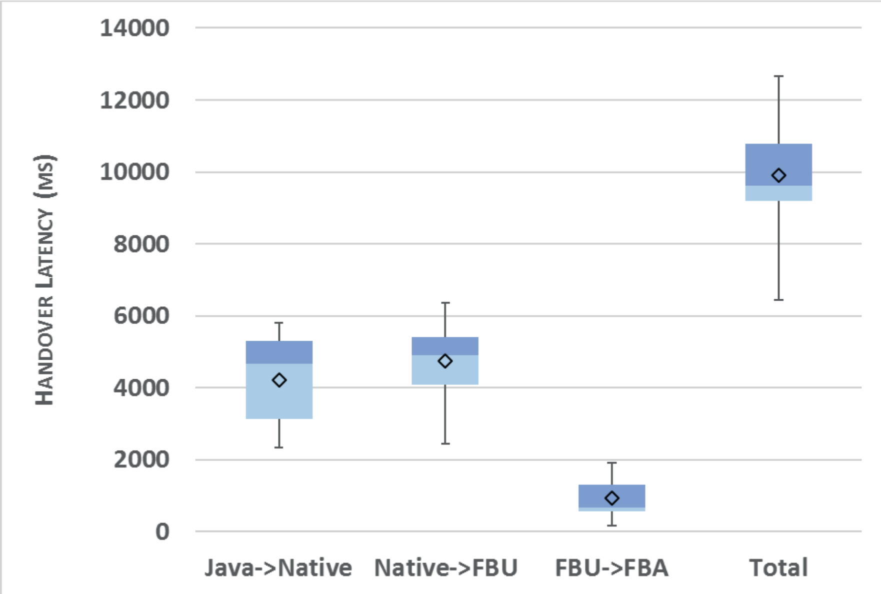

Figure 5 - The total vertical flow handover latency in its three main components

6. Conclusion

Current offloading standards are defined by 3GPP so that the operator can manage the way flows are routed either through the operator network or through the Internet. This allows the operator to pass the "important" traffic on its network, and the "best effort" traffic (e.g., Youtube) to the Internet. Our introduced client-based, flow-aware, cross-layer optimized offloading scheme would allow to offer similar type of services, but on overlay of the network operator, where either someone outside of the network operator, or the mobile itself can control how the flows of the mobile node would be routed using its multiple access. We confirmed the applicability of our solution by evaluating it in an integrated femtocell/Wi-Fi testbed environment with the help of extensive real-life measurements. As a part of our future activities we are planning to refine our algorithm (e.g., decreasing the measurement period), optimize our implementation (reduce inside Android signaling delays), combine our client-based approach with network-based mobility management techniques (e.g., by integrating Home Agent initiated handovers into the scheme), and also to further enhance our decision engine.

Acknowledgements

The research leading to these results has received funding from the European Union's Seventh Framework Programme ([FP7/2007-2013]) under grant agreement n° 288502 (CONCERTO project). The authors are grateful to the many individuals whose work made this research possible.

References

[1] Cisco Visual Networking Index, "Global Mobile Data Traffic Forecast Update, 2013–2018." 05-Feb-2014.

[2] N. Varga, L. Bokor, and A. Takács, "Android-based Testbed and Demonstration Environment for Cross-layer Optimized Flow Mobility," in *9th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities (TridentCom'2014), May 5–7, 2014 Guangzhou, People's Republic of China*, 2014.

[3] K.-T. Tran, V. K. Gondi, and N. Agoulmine, "Analysis of Client-Based and Network-Based Mobility Protocols," in *Communications (ICC), 2010 IEEE International Conference on*, 2010, pp. 1–5.

[4] Small Cell Forum, "Integrated Femto-WiFi (IFW) Networks." Small Cell Forum Ltd, 28-Feb-2012.

[5] L. Virág, J. Kovács, and A. Edelmayer, "Interacting Advanced ITS Communications with Low-Power Sensor Networks," *J. Wirel. Mob. Netw. UBIQUITOUS Comput. DEPENDABLE Appl.*, vol. 4, no. 3, pp. 79–96, 2013.

[6] Qualcomm Incorporated, "3G/Wi-Fi Seamless Offload." Qualcomm Incorporated, Mar-2010.

[7] Y.-N. Lin, W. Chen, S.-C. Tsai, and Y.-B. Lin, "Design and Implementation of an Offloading Technology for 3.5G Networks," in *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*, 2010, pp. 1–5.

[8] 3GPP TR 23.829, "Local IP Access and Selected IP Traffic Offload (Release 10)." Oct-2011.

[9] K. Samdanis, T. Taleb, and S. Schmid, "Traffic Offload Enhancements for eUTRAN," *Commun. Surv. Tutor. IEEE*, vol. 14, no. 3, pp. 884–896, Third 2012.

[10] 3GPP TR 23.861, "Network based IP flow mobility (Release 12)." Nov-2012.

[11] D. Knisely, T. Yoshizawa, and F. Favichia, "Standardization of femtocells in 3GPP," *Commun. Mag. IEEE*, vol. 47, no. 9, pp. 68–75, Sep. 2009.

[12] G. Horn, "3GPP Femtocells: Architecture and Protocols." Qualcomm Incorporated, Sep-2010.

[13] J. Roh, Y. Ji, Y. G. Lee, and T. Ahn, "Femtocell Traffic Offload Scheme for Core Networks," in *New Technologies, Mobility and Security (NTMS), 2011 4th IFIP International Conference on*, 2011, pp. 1–5.

[14] 3GPP TS 23.261, "IP flow mobility and seamless WLAN offload Stage 2." Mar-2012.

[15] H. Soliman, *Mobile IPv6 Support for Dual Stack Hosts and Routers*. IETF, 2009.

[16] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, *Proxy Mobile IPv6*. IETF, 2008.

[17] 3GPP TS 24.312, "Access Network Discovery and Selection Function (ANDSF) Management Object (MO), Rel. 11." Apr-2013.

[18] 3GPP TS 23.402, "Architecture enhancements for non-3GPP accesses", Rel. 11" .

[19] A. Takács and L. Bokor, "A Distributed Dynamic Mobility Architecture with Integral Cross-Layered and Context-Aware Interface for Reliable Provision of High Bitrate mHealth Services," in *Wireless Mobile Communication and Healthcare*, vol. 61, B. Godara and K. Nikita, Eds. Springer Berlin Heidelberg, 2013, pp. 369–379.

[20] T. Ropitault and N. Montavont, "Implementation of Flow Binding Mechanism," in *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on*, 2008, pp. 342–347.

[21] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, *Network Mobility (NEMO) Basic Support Protocol*. IETF, 2005.

[22] A. De La Oliva, C. J. Bernardos, M. Calderon, T. Melia, and J. C. Zuniga, "IP flow mobility: smart traffic offload for future wireless networks," *Commun. Mag. IEEE*, vol. 49, no. 10, pp. 124–132, Oct. 2011.

[23] R. Wakikawa, V. Devarapalli, G. Tsirtsis, T. Ernst, and K. Nagami, *Multiple Care-of Addresses Registration*. IETF, 2009.

[24] G. Tsirtsis, H. Soliman, N. Montavont, G. Giaretta, and K. Kuladinithi, *Flow Bindings in Mobile IPv6 and Network Mobility (NEMO) Basic Support*. IETF, 2011.

[25] Q. Wang, R. Atkinson, and J. Dunlop, "Design and evaluation of flow handoff signalling for multihomed mobile nodes in wireless overlay networks," *Comput. Netw.*, vol. 52, no. 8, pp. 1647 – 1674, 2008.

[26] F. Hoguet, "Network mobility for multi-homed Android mobile devices," Nicta, Eveleigh, Sydney, NSW, Australia , Compiagne, France, 2012.

[27] R. Silva, P. Carvalho, P. Sousa, and P. Neves, "Enabling Heterogeneous Mobility in Android Devices," *Mob. Netw. Appl.*, vol. 16, no. 4, pp. 518–528, 2011.

[28] IEEE, "IEEE Standard for Local and metropolitan area networks- Part 21: Media Independent Handover." IEEE, Jan-2009.

[29] T. Inzerilli, A. M. Vegni, A. Neri, and R. Cusani, "A Location-Based Vertical Handover Algorithm for Limitation of the Ping-Pong Effect," in *Networking and Communications, 2008. WIMOB '08. IEEE International Conference on Wireless and Mobile Computing,*, 2008, pp. 385–389.