Wireless Pers Commun
DOI 10.1007/s11277-007-9424-7

QoS-Aware Network-supported Architecture to Distribute Application Flows over Multiple Network Interfaces for B3G Users

Qi Wang · Tobias Hof · Fethi Filali · Robert Atkinson · John Dunlop · Eric Robert · Leire Aginako

Abstract Users in the Beyond-Third-Generation (B3G) wireless system expect to receive ubiquitous communication services with customised quality-of-service (QoS) commitments for different applications, preferably in a way as transparent as possible. Ideally, flows belonging to diverse applications can be automatically and optimally distributed (or handed off) among the most appropriate access networks for multihomed users. To contribute to realising this vision, we propose a novel architecture to achieve QoS-aware policy-based flow handoffs for multihomed users, especially those equipped with more than a single personal device. In this architecture, advanced network intelligence enables a personal gateway to handle flow distributions dynamically for all the devices behind it according to the applications' QoS requirements and the current available network resources. The essential procedures in this

Q. Wang (✉) · R. Atkinson · J. Dunlop
Mobile Communications Group, Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, G11XW, UK
e-mail: qwang@eee.strath.ac.uk

R. Atkinson
e-mail: r.atkinson@eee.strath.ac.uk

J. Dunlop
e-mail: j.dunlop@eee.strath.ac.uk

T. Hof · E. Robert
TAI Laboratory, Thales Communications S. A., 92704 Colombes, Paris, France

T. Hof
e-mail: Tobias.HOF@fr.thalesgroup.com

E. Robert
e-mail: Eric.ROBERT@fr.thalesgroup.com

F. Filali
Mobile Communications Department, Institut Eurecom, Nice, France
e-mail: Fethi.Filali@eurecom.fr

L. Aginako
Department of Strategy and Technology, Euskatel S. A., Bilbao, Spain
e-mail: laginako@euskaltel.es

architecture are described. Following that, the flow handoff delay is analysed and numerical results are illustrated. To prove the proposed concepts, up-to-date implementations with experimental results are also presented.

Keywords Multihoming · Flow handoff · NEMO · Delay analysis · QoS measurement

1 Introduction

Recent years have witnessed the remarkable emergence of a new communication paradigm, the Beyond-Third-Generation (B3G). B3G is integrating homogeneous and heterogeneous wireless networks such as 2G/2.5G/3G, Wi-Fi wireless local area network (WLAN) and the Worldwide Interoperability for Microwave Access (WiMAX) into a uniform platform based on IP with increasing popularity towards IPv6. Such a convergence of competing yet complementary access technologies has fostered the exciting vision of pervasive, permanent and personalised communication services. To facilitate the users to fully utilise available access networks, the market is shipping a range of smart devices such as laptops with several interfaces and cellular phones of dual or multiple modes. Therefore, it is possible for multihomed B3G users to have application flows with different quality-of-service (QoS) requirements distributed across these interfaces based on policies generated from intelligent network selection algorithms. This operation is referred to as a *flow handoff* in this paper. Preferably, a flow handoff should be mainly dealt with by the network intelligence [1] due to the complexity of the QoS measurements and computing-intensive algorithms. Nevertheless, such advanced flow handoff functionality is still unavailable. Furthermore, the set of devices carried by a user formulates a Personal Area Network (PAN). These devices often run incompatible operating systems such as different versions of Windows and Linux, and have different levels of computing capabilities and battery constraints. Therefore, it would be inefficient and impractical for each device to be installed with the desired flow handoff functionality. A powerful personal gateway (PG) would thus be desired to handle flow handoffs for all the devices in the PAN. To sum up, tremendous research needs to be undertaken to fulfil QoS-aware network-supported intelligent flow handoffs for B3G users.

In the Internet Engineering Task Force (IETF), the Network Mobility (NEMO) basic protocol [2] has been standardised to extend the mobility functionality in Mobile IPv6 (MIPv6) [3] to mobile routers (MRs) rather than single mobile hosts (MHs). This extension facilitates mobility of an entire mobile network. A MR acts as a gateway to the external Internet for devices located within the mobile network. A MR registers its care-of address (CoA), bound with its home address, with its NEMO home agent (HA), which is an enhanced MIPv6 HA aware of the prefix of the mobile network. Bidirectional tunnelling between the MR and its HA is then established for delivering application flows between the mobile network nodes (MNNs) behind the MR and their correspondent nodes (CNs). Naturally, a PG may be built on top of a MR. However, advanced flow handoff functionalities in the multihoming context have yet to be added and standardised. To tackle the flow handoff issue, the Mobile Nodes and Multiple Interfaces in IPv6 (MONAMI6) working group has produced a number of relevant Internet drafts. The MCoA draft [4] enables a multihomed mobile node (either MH or MR) to register multiple CoAs with its HA so that multiple bidirectional tunnels can simultaneously exist, a prerequisite for flow handoffs. The Flow Binding draft [5] extends MIPv6 and NEMO mobility messages including Binding Update (BU) and Binding Ack (BA) to enclose user-defined policies so that flows can be distributed according to the

policies. However, since a BU is always sent from a Mobile Node (MN, either a MH or a MR) to its HA the mechanism can only support user-initiated flow handoffs, which do not exploit the network intelligence. The Flow Distribution draft [6] uses flexible messages encoded in the Extensible Markup Language (XML) over Simple Object Access Protocol (SOAP) for policy exchange, which can be initiated by a MN or its HA and thus enables the potential introduction of network intelligence. Nonetheless, the Flow Distribution draft [6] is still a work in progress and needs considerable further research and development. The Nautilus6 project [7] is implementing NEMO and the MCoA draft; however, dynamic policy-based flow handoffs have not been implemented. Furthermore, no network intelligence has been incorporated into their architecture.

In addition to the above IP-layer approach, there are other emerging proposals attempting to support multihomed users from other layers. Amongst them, the Stream Control Transmission Protocol (SCTP) [8] appears the most promising protocol that enables multihoming in the transport layer and deals with handoffs with extensions such as mSCTP [9]. Nevertheless, running in the transport layer the SCTP multihoming would only manage to manipulate applications that are based on SCTP rather than TCP or UDP. Consequently, the overwhelming majority of the current and legacy applications can hardly benefit.

In the EU IST FP6 MULTINET project [10], we are constructing advanced service delivery architecture for B3G users by integrating the most appropriate existing protocols with crucial enhancements as well as introducing novel techniques in network intelligence. In this paper, we report the state-of-the-art achievements of this project on QoS-aware network-supported flow-handoff management. The remainder of the paper is organised as follows. Section 2 presents the proposed architecture with a focus on signalling procedures. Subsequently in Sect. 3, the flow handoff delay is analysed with a mathematical model developed; and numerical results are demonstrated. In Sect. 4, the proof-of-concept implementations, together with experimental results, are described. The paper is concluded in Sect. 5.

2 Proposed Architecture

2.1 Reference Network Model

For the following design and analysis, the network model is illustrated by Fig. 1. The model is a simplified version of the MULTINET reference architecture [10], which reflects a real-world practical deployment.

In this architecture, the PG is a NEMO MR that is enhanced with flow handoff functionalities (to be described in the following subsections); and it is equipped with multiple network interfaces e.g., IF1 and IF2. The PG, together with the PAN, is visiting a foreign domain. The HA of the PG is located in the home domain and collaborates with the PG and a serving Common Radio Resource Manager (CRRM) for flow handoff support. Symmetric flow binding policies are dynamically stored, synchronised and enforced in both the PG and the HA for distributing the identified uplink and downlink flows, respectively. A correspondent node (CN) is stationary in a third domain. All the domains are interconnected through a common IP core network. In addition, pure IPv6 is assumed in this architecture for B3G although IPv6–IPv4 interoperability is being investigated for the IPv4–IPv6 transition period.

In the foreign domain, two access routers AR1 and AR2, together with integrated access points, provide wireless access to the PG. A QoS measurement entity, called wimeter, is collocated with each access point. The Radio Resource Manager (RRM) resident in each

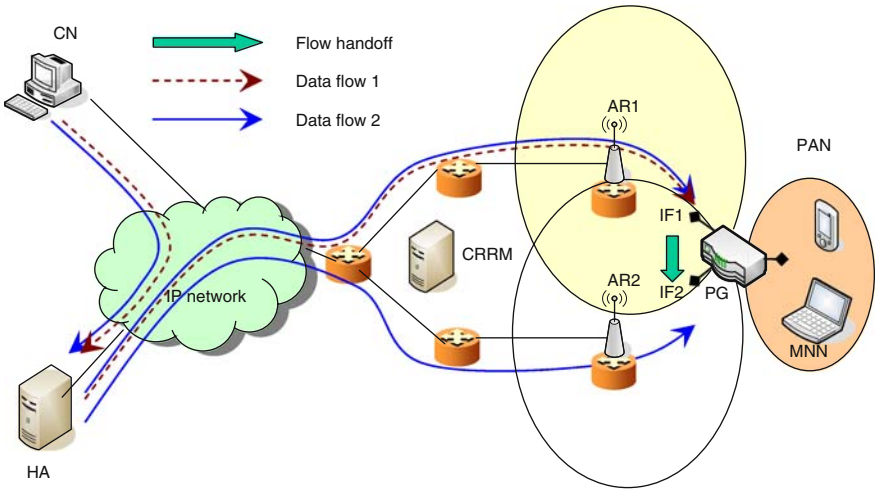


Fig. 1 Network model

AR reports to a serving CRRM. The intelligent network selection algorithms running at the CRRM determine if a flow handoff should be triggered based on the output of the intelligent network selection algorithms. There is no constraint on the physical location of the CRRM: it can be located wherever appropriate as chosen by the service provider. For instance, more than one foreign domain may share a CRRM in a gradual deployment. More detailed definitions of the components in the architecture can be found in [10].

Based on the targeted user scenarios defined in the MULTINET project, we concentrate on QoS-oriented flow handoffs for nomadic users, who do not experience movement-triggered IP handoffs during the session's lifetime. Standard IP security schemes such as IPsec and/or HTTPS are applicable to the proposed schemes for secure signalling. Additional Authentication, Authorisation and Accounting (AAA) mechanisms are not explicitly addressed in this paper.

In the scenario depicted in Fig. 1, prior to a flow handoff two flows are being transmitted through AR1 and IF1; after a flow handoff one of the flows is redirected to AR2 and IF2. Bidirectional tunnelling between the PG and the HA is shown as supported in the NEMO base protocol. Figure 1 only illustrates the downlink flows for clarity.

2.2 Registration

In order to leverage the benefits of the multihoming flow handoff service, a PAN must perform a registration procedure as shown in Fig. 2. When the multihomed PG is activated in a foreign domain, its multiple interfaces may configure global IPv6 addresses using IPv6 stateless auto-configuration or other means such as the Dynamic Host Configuration Protocol version 6 (DHCPv6). The PG then sends a single BU message to register its multiple CoAs with its HA, which replies with a BA message to indicate whether the binding update is successful. This bulk registration mode is defined in the MCoA draft [4].

Subsequently, each activated MNN in the PAN can send a SOAP Registration Request to the PG, which then forwards the message to the serving CRRM. The PG has learnt the address of the serving CRRM from the extended Router Advertisement message in stateless auto-configuration (or the DHCPv6 Reply message assuming the Rapid Commit mode). This

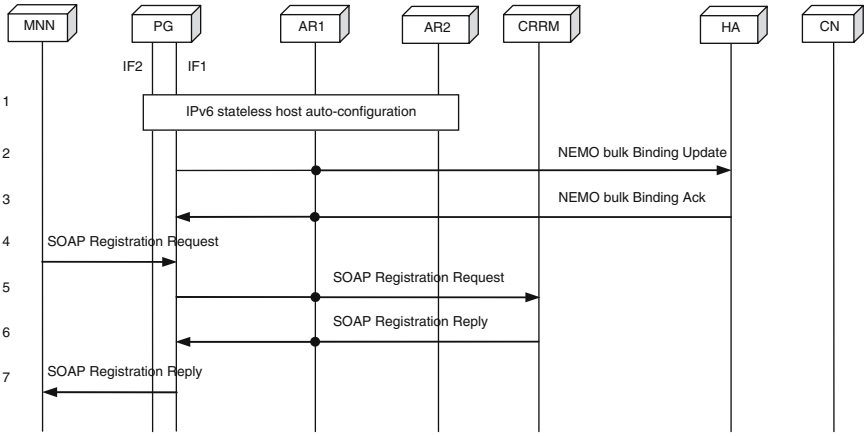


Fig. 2 Registration

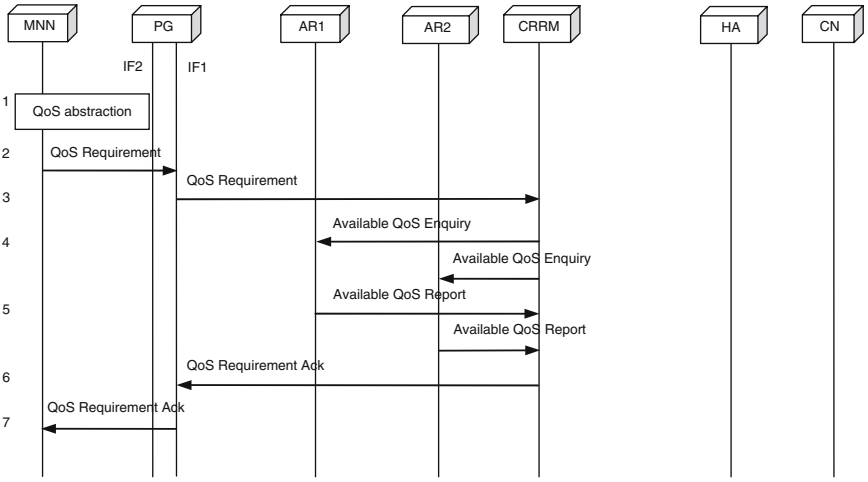


Fig. 3 QoS measurement and report

SOAP request message contains MNN-specific profiles including the terminal capabilities, application and network preferences, and addressing information (home address, the HA's address etc.). Later on, the intelligent algorithms in the CRRM will use these data, the QoS requirements of applications and the collected QoS information to determine if a flow handoff should be triggered and to formulate the corresponding policies if a flow handoff is decided. After processing the message, the CRRM replies the PG with a SOAP Registration Reply message, which is then forwarded to the MNN by the PG. The PG can also broadcast the address of the CRRM in the PAN so that each MNN can send the message to the address of the CRRM directly.

2.3 QoS Measurement

Figure 3 illustrates the signalling steps to establish QoS awareness in the architecture. Firstly, a MNN in the PAN characterises the QoS requirement of an application to be initiated. In

our design, a typical QoS requirement is expressed as the mean packet size, the required bandwidth and the priority of the application. Actually, the priority can be implicitly indicated based on a pre-mapping between application packet size and its corresponding QoS class. Secondly, the MNN sends the QoS Requirement message to its PG, which then forwards the message to the serving CRRM. The CRRM queries the intelligent network selection algorithms and the associated resource database for current resource availability within the access networks using the application packet size as a key input. If the estimated resources to meet the QoS requirement can be computed based on the current database records, the CRRM sends a QoS Requirement Ack message to the PG and the message is delivered to the MNN. From this message, the MNN can send the application flows to the most appropriate interface of the PG so that the flows are delivered over the most appropriate access network. If the resource estimation can not be made based on current records (e.g., the records are outdated), the CRRM sends an Available QoS Enquiry message to each of the involved ARs to request independent QoS status report from each AR. The RRM in each AR then performs QoS measurement and returns an available QoS Report message to the CRRM. The QoS measurement is typically in terms of available bandwidth. For CN-initiated applications, if such a service is unavailable the applications can be established through a random access network and redistributed in a later stage.

Due to the arrival of new application flows, the intelligent network selection algorithms running at the CRRM may determine that a flow handoff should be initiated for a subset of existing flows to optimally make use of the overall network resources distributed in the access networks. Detailed intelligent network selection algorithms are being defined and they are beyond the scope of this paper.

2.4 Flow Handoff

Once a flow handoff is determined by the intelligent network selection algorithms, the flow handoff procedure is initiated. Two schemes are designed as shown in Figs. 4 and 5, respectively.

In Scheme I, firstly the CRRM sends a SOAP-based Trigger message to the HA. In this Trigger, downlink policies for flow handoffs generated by the intelligent algorithms are enclosed. A policy includes a binding between an identifier of a flow (or a class of flows) and an identifier of the selected interface through which the identified flow would be transmitted. The processing action such as inserting the policy or flushing all policies is also associated with a policy so that the receiver of the policy can proceed accordingly. A flow can be identified by an arbitrary combination of five tuples: source address, destination address, source port number, destination port number, protocol number. Other fields of a packet may also be used for this purpose. An interface can be identified by the CoA of the interface or a Binding ID (BID) that is a unique number assigned to identify a pair of home address and CoA and defined in the MCoA draft [4]. BID is preferred since a policy that is expressed as binding of a flow identifier and a BID (e.g., port number = 80, BID = 200) does not need to be modified when the corresponding CoA changes mainly due to a user's movement. This choice should facilitate our future design for fully mobile scenarios.

On completing the processing of the Trigger, the HA replies a SOAP Trigger Ack message to the CRRM. Simultaneously, the HA also sends a SOAP Flow Handoff Request message to the PG to update the uplink flow handoff policies stored in the PG. Upon receiving the request, the PG parses the policies and double checks if all the involved interfaces are still available. If any of the involved interfaces has become unavailable, the PG rejects the policy updates; otherwise, the PG accepts and enforces the new policies. In either case, the reply

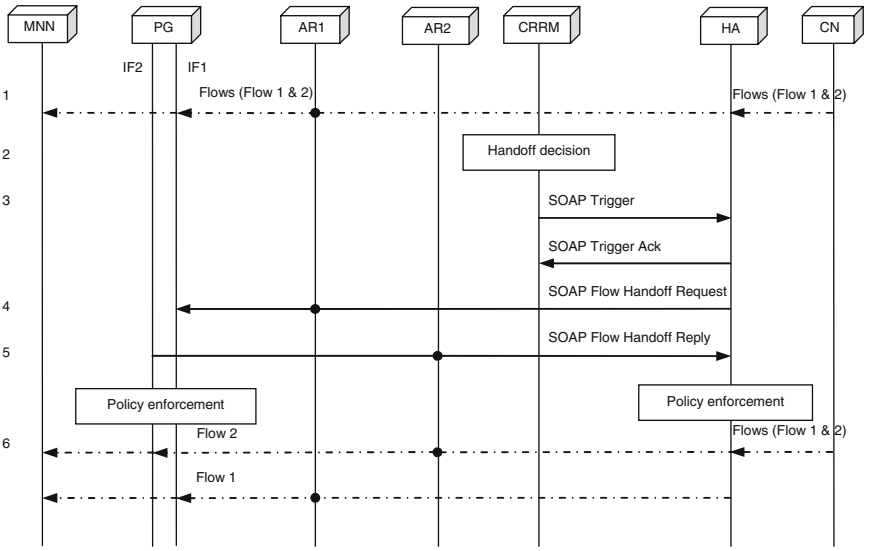


Fig. 4 Flow handoff (Scheme I)

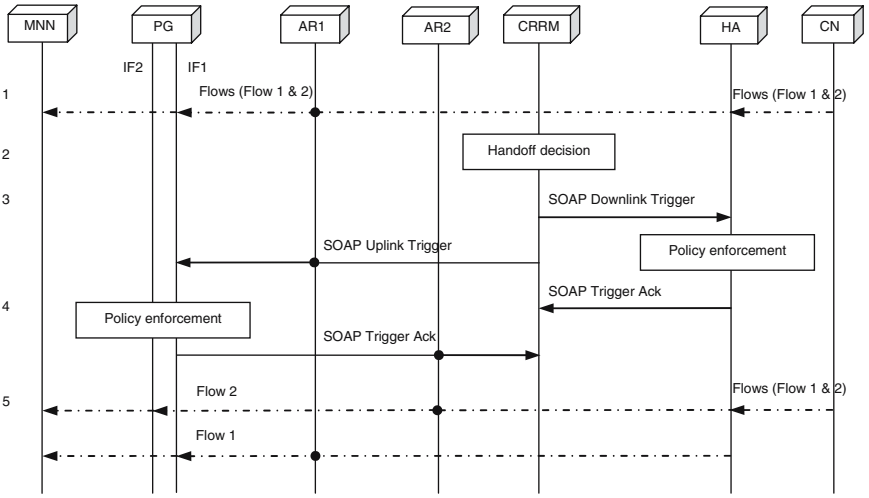


Fig. 5 Flow handoff (Scheme II)

is signalled in a SOAP Flow Handoff Reply message. Depending on the received reply, the HA abandons the proposed flow handoff or enforces the downlink policies to execute the flow handoff. To accelerate the procedure, the HA may enforce the downlink policies whilst waiting for the response from the PG if deemed appropriate (the HA can check the binding cache for currently valid interfaces). If the policies cannot be enforced successfully, the HA reports the reasons to the CRRM. Figure 4 demonstrates the scenario where the flow handoff is executed successfully. As an example, Flow 2 is handed over from IF1 (AR1) to IF2 (AR2).

In Scheme II, the CRRM generates and sends the downlink and uplink SOAP Trigger messages to the HA and the PG, respectively. The message sent to the HA contains the downlink

policies for traffic addressed to the mobile network, and the one for the MR contains uplink policies for outgoing traffic from the mobile network. The policies are then processed and enforced at the HA and the MR, respectively. If the policies cannot be executed successfully, errors are reported to the CRRM from the HA and the MR, respectively.

Note that symmetric downlink and uplink policies are assumed in Scheme I whilst no such restrictions are imposed in Scheme II. Furthermore, Scheme II does not require the additional capability of the HA to generate the uplink policies. On the other hand, in Scheme I the CRRM does not need to produce the uplink policies or communicate with a PG for policy signalling.

3 Delay Analysis and Results

In this section, we develop an analytical model for performance evaluation in terms of flow handoff delay.

3.1 Assumptions and Parameters

The flow handoff delay is defined as the average time elapsed between the time instance when an IP-layer flow handoff decision is made and the time instance when the flow binding policies are synchronised between the PG and the HA so that the subsequent packets of the selected flows can be distributed amongst the interfaces in each transmission direction. This delay corresponds to the time taken to fulfil Steps 3 to 5 in Fig. 4 for Scheme I, or Step 3 in Fig. 5 for Scheme II. During the flow handoff delay period, QoS may be temporally degraded due to lack of network resource (e.g., bandwidth) for the identified flows that should be redirected to another access network.

In the following, we list the assumptions and parameters for the following analysis.
Assumptions:

- The involved signalling entities are interconnected through wired links with the exception of the PG and the ARs, which communicate through one-hop wireless links as typically deployed in an infrastructure-based wireless network. Since effectively wired links are by far more reliable than their wireless counterparts, it is fair to assume that the wired hops in the system are error-free and the packet loss is only caused by the error-prone wireless hop between a PG and a serving AR.
- SOAP can run over either HTTP/TCP or UDP directly. SOAP over UDP is assumed in the analysis for efficient signalling in wireless networking context [11].
- Since UDP is an unreliable transport protocol, practically SOAP over UDP (at a PG, the HA, or the CRRM) needs to employ its own timer-based retransmission mechanism such as in the Session Initiation Protocol (SIP) [12], in MIPv6 and in NEMO. For each SOAP message, there are no additional underlying retransmission (or other error correction) mechanisms. This is common in WLAN systems. The sender of a request message keeps on retransmitting the request when a local predefined timer expires until it receives a matching reply message from the receiver or aborts the operation when the predefined maximum timer is reached; the receiver of a request sends back or retransmits a reply only upon the receipt of the request (either the initial one or a retransmitted copy). Figure 6 demonstrates a retransmission scenario where a PG attempts to send a request to the HA or the CRRM via an AR. τ_i is the $(i+1)$th retransmission timer. In this specific example, the transaction succeeds on the fourth transmission (i.e., the third retransmission) trials from the PG's perspective.

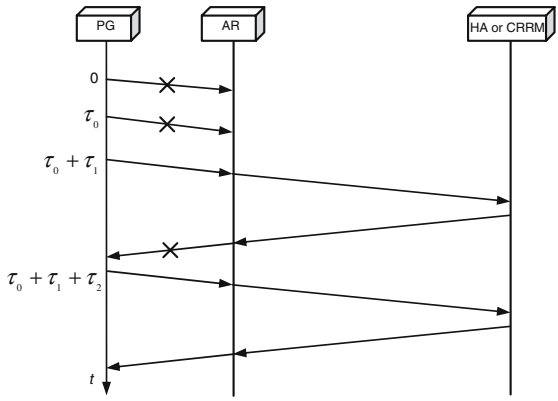


Fig. 6 Timer-based retransmissions

Parameters:

β_i: The bit error rate (BER) of the wireless hop that packet i (a packet or a non-oversized message of type i) is being transmitted over ($\beta_i < 1$)
α_i: The packet loss rate of packet i
$q_{i,i+1}$: The probability that a transaction consisting of a pair of request and reply packets i and $i + 1$ fails (We denote packet $i + 1$ be the reply to packet i that is a request.)
L_i: The length of packet i (in the unit of bytes; link-layer length unless otherwise stated)
H_{A-B}: The distance (in terms of hops) between entities A and B, or between B and A (non-directional)
N_{RM}: The maximum number of transmission trials
τ_0: The initial retransmission timer
r: The coefficient for a retransmission mechanism based on exponential back-off
$t_{i(j)}t_{i,i+1(j)}$: The delay incurred by the jth transmission of packet i or both packets i and $i + 1$
$T_i(h)$: The average one-way delay for packet i along a path of h hops (with retransmissions considered)
$T_{i,i+1}(h)$: The average round-trip delay for packet i and $i + 1$ along a path of h hops in each direction (with retransmissions considered)
$\overleftrightarrow{T}_i(h)$: The average one-way delay of packet i along a path of h hops on a successful transmission
B_k^{wl}, B_k^{wd}: The bandwidth of the kth wireless or wired hop
T_k^{wl}, T_k^{wd}: The average propagation latency of the kth wireless or wired hop
$T_i^{(e)}$: The average processing delay of packet i at entity e along the end-to-end route
$E_i^{(h)}$: The set of entities along the end-to-end route of h hops for packet i
$\lambda_i^{(e)}$: The mean arrival rate of packet i at entity e
$\mu_i^{(e)}$: The mean processing rate of packet i at entity e
$\rho_i^{(e)}$: The mean utilisation for packet i at entity e ($\rho_i^{(e)} = \lambda_i^{(e)}/\mu_i^{(e)}$)
$\sigma_i^{(e)}$: The variance of processing time for packet i at entity e

3.2 Handoff Delay Analysis

As shown in Fig. 4, a flow handoff in Scheme I starts when the downlink policies are generated by the CRRM and completes when the HA receives the response from the PG (after the

uplink policies are enforced at the PG) and the HA finishes enforcing the downlink policies. Therefore, the flow handoff delay in Scheme I is an accumulative delay as follows:

$$T_{\mathrm{HO}}^{I} = T_{\mathrm{SOAP\_Trigger}}(H_{\mathrm{CRRM-HA}}) + T_{\mathrm{SOAP\_Req,SOAP\_Rep}}(H_{\mathrm{HA-PG}}), \tag{1}$$

where $T_{\mathrm{SOAP\_Trigger}}(H_{\mathrm{CRRM-HA}})$ is the delay to signal and process the SOAP Trigger message from the CRRM to the HA, and $T_{\mathrm{SOAP\_Req,SOAP\_Rep}}(H_{\mathrm{HA-PG}})$ is the joint delay to signal and process the SOAP Flow Handoff Request and Reply messages between the HA and the PG assuming that the HA waits for the response to enforce the downlink policies.

According to Fig. 5, a flow handoff in Scheme II is initiated with both downlink and uplink policies are generated at the CRRM and is finished when both the downlink and the uplink policies have been enforced by the HA and the PG, respectively. Thus, the flow handoff delay in Scheme II is determined by either the delay between the CRRM and the HA or that between the CRRM and the PG, whichever is longer. Therefore, this delay can be expressed as

$$T_{\mathrm{HO}}^{II} = \max\left\{T_{\mathrm{SOAP\_Trigger}}(H_{\mathrm{CRRM-HA}}), T_{\mathrm{SOAP\_Trigger}}(H_{\mathrm{CRRM-PG}})\right\}, \tag{2}$$

where $T_{\mathrm{SOAP\_Trigger}}(H_{\mathrm{CRRM-HA}})$ and $T_{\mathrm{SOAP\_Trigger}}(H_{\mathrm{CRRM-PG}})$ is the delay to signal and process the SOAP Trigger message from the CRRM to the HA and to the PG, respectively.

In the following, we derive the formulae of $T_{\mathrm{SOAP\_Trigger}}(H_{\mathrm{CRRM-HA}})$ and $T_{\mathrm{SOAP\_Req,SOAP\_Rep}}(H_{\mathrm{HA-PG}})$ in Scheme I; and $T_{\mathrm{SOAP\_Trigger}}(H_{\mathrm{CRRM-HA}})$ and $T_{\mathrm{SOAP\_Trigger}}(H_{\mathrm{CRRM-PG}})$ in Scheme II. The first step is to establish a couple of commonly used functions. First of all, a successful transmission of packet i requires that every bit of the packet is correctly received. Therefore, the packet loss rate of packet i is given by

$$\alpha_i = 1 - (1 - \beta_i)^{8 \cdot L_i}, \tag{3}$$

where $8 \cdot L_i$ is the length of packet i in the unit of bits.

As widely employed elsewhere such as in SIP, MIPv6 and NEMO, an exponential back-off algorithm is assumed for retransmissions and thus the timer for the ith transmission is given by

$$\tau_i = r^i \cdot \tau_0. \tag{4}$$

In the flow handoff procedure of Scheme I, once the Trigger is received and processed the HA can send the Flow Handoff Request to synchronise the policies at the PG regardless of the Trigger Ack it is sending to the CRRM. Therefore, $T_{\mathrm{SOAP\_Trigger}}(H_{\mathrm{CRRM-HA}})$ in Scheme I can be identified independently from the Trigger Ack. The same is true for $T_{\mathrm{SOAP\_Trigger}}(H_{\mathrm{CRRM-HA}})$ and $T_{\mathrm{SOAP\_Trigger}}(H_{\mathrm{CRRM-PG}})$ in Scheme II. Statistically, the mean delay for a request packet i (regardless of the reply) over h hops including a wireless hop is given by

$$\begin{aligned}
T_i(h) &= \sum_{j=1}^{N_{RM}} p_{i(j)} \cdot t_{i(j)} \\
&= (1-\alpha_i) \cdot \overleftrightarrow{T}_i(h) + \alpha_i \cdot (1-\alpha_i) \cdot \left(\overleftrightarrow{T}_i(h) + \tau_0\right) + \alpha_i^2 \cdot (1-\alpha_i) \cdot \left(\overleftrightarrow{T}_i(h) + \tau_0 + \tau_1\right) \\
&\quad + \cdots + \alpha_i^{N_{RM}-1} \cdot (1-\alpha_i) \cdot \left(\overleftrightarrow{T}_i(h) + \tau_0 + \tau_1 + \cdots + \tau_{N_{RM}-1}\right) \\
&= \begin{cases} (1-\alpha_i) \cdot \sum_{j=0}^{N_{RM}-1} \left\{\alpha_i^j \cdot \left[\overleftrightarrow{T}_i(h) + \frac{\tau_0 \cdot (1-r^j)}{1-r}\right]\right\} & r \neq 1; \\ (1-\alpha_i) \cdot \sum_{j=0}^{N_{RM}-1} \left[\alpha_i^j \cdot (\overleftrightarrow{T}_i(h) + j \cdot \tau_0)\right] & r = 1. \end{cases}
\end{aligned} \tag{5}$$

Accordingly, $T_{\text{SOAP\_Trigger}}(H_{\text{CRRM-HA}})$ can be obtained from (5) by letting $\alpha_i = 0$ (no wireless hop between the CRRM and the HA as assumed).

In contrast, the HA can only start to execute the flow distribution after the Flow Handoff Request is agreed on by the PG through the Flow Handoff Reply. As demonstrated in Fig. 6, the probability that a transmission fails (when either the request is lost or the request is received whereas the reply is lost) and thus a retransmission is incurred is given by

$$q_{i,i+1} = 1 - (1 - \alpha_i) \cdot (1 - \alpha_{i+1}) = \alpha_i + (1 - \alpha_i) \cdot \alpha_{i+1}. \tag{6}$$

Consequently, the mean delay for such a pair of request and reply packets i and $i + 1$ over h hops including a wireless hop is given by

$$\begin{aligned}
T_{i,i+1}(h) &= \sum_{j=1}^{N_{RM}} p_{i,i+1(j)} \cdot t_{i,i+1(j)} \\
&= (1 - q_{i,i+1}) \cdot \left(\overleftrightarrow{T}_i(h) + \overleftrightarrow{T}_{i+1}(h)\right) \\
&\quad + q_{i,i+1} \cdot (1 - q_{i,i+1}) \cdot \left(\overleftrightarrow{T}_i(h) + \overleftrightarrow{T}_{i+1}(h) + \tau_0\right) \\
&\quad + q_{i,i+1}^2 \cdot (1 - q_{i,i+1}) \cdot \left(\overleftrightarrow{T}_i(h) + \overleftrightarrow{T}_{i+1}(h) + \tau_0 + \tau_1\right) + \cdots \\
&\quad + q_{i,i+1}^{N_{RM}-1} \cdot (1 - q_{i,i+1}) \cdot \left(\overleftrightarrow{T}_i(h) + \overleftrightarrow{T}_{i+1}(h) + \tau_0 + \tau_1 + \cdots + \tau_{N_{RM}-1}\right) \\
&= \begin{cases} (1 - q_{i,i+1}) \cdot \sum_{j=0}^{N_{RM}-1} \left\{ q_{i,i+1}^j \cdot \left[\left(\overleftrightarrow{T}_i(h) + \overleftrightarrow{T}_{i+1}(h)\right) + \frac{\tau_0 \cdot (1 - r^j)}{1 - r} \right] \right\} & r \neq 1; \\ (1 - q_{i,i+1}) \cdot \sum_{j=0}^{N_{RM}-1} \left\{ q_{i,i+1}^j \cdot \left[\left(\overleftrightarrow{T}_i(h) + \overleftrightarrow{T}_{i+1}(h)\right) + j \cdot \tau_0 \right] \right\} & r = 1. \end{cases}
\end{aligned} \tag{7}$$

$T_{\text{SOAP\_Req,SOAP\_Rep}}(H_{\text{HA-PG}})$ can thus be obtained from (7).

Next, by generalising the formulae provided in [13] we can estimate the one-way delay of packet i along a path of h hops on a successful trial as

$$\overleftrightarrow{T}_i(h) = \sum_{k=1}^{j} \left(\frac{L_i}{B_k^{\text{wl}}} + T_k^{\text{wl}} \right) + \sum_{k=1}^{h-j} \left(\frac{L_i}{B_k^{\text{wd}}} + T_k^{\text{wd}} \right) + \sum_{e \in E_i^{(h)}} T_i^{(e)}. \tag{8}$$

The first and second terms are transmission and propagation delays over the j wireless hops and the $(h - j)$ wired hops along the path, respectively; the third term is the accumulative processing delays incurred at the $(h + 1)$ entities including $(h - 1)$ routers, one source entity and one destination entity along the path. Note that if there are no wireless or wired hops involved, the first or second term of the right hand in (8) becomes zero and j or $(h - j)$ is replaced with h, respectively. Given the reference network model and our assumptions, it is clear that $j = 1$ for signalling between a PG and the HA or the CRRM (via ARs), and $j = 0$ between a CRRM and a HA. $\overleftrightarrow{T}_i(h)$ is also the average delay for packet i that is transmitted over h wired hops. Actually, it can be derived that $T_i(h) \to \overleftrightarrow{T}_i(h)$ when $\alpha_i \to 0$ in (5) and $T_{i,i+1}(h) \to \overleftrightarrow{T}_i(h) + \overleftrightarrow{T}_{i+1}(h)$ when $q_{i,i+1} \to 0$ (with $\alpha_i, \alpha_{i+1} \to 0$) in (7).

Regarding processing delays, typically each of the involved end signalling entities can be modelled as an M/G/1 system [14]. Hence, according to the Pollaczek–Khinchin mean value formulae in the queuing theory, the average processing delay (including waiting time and

actual processing time) is given by

$$T_i^{(e)} = \frac{1}{\mu_i^{(e)}} + \frac{\lambda_i^{(e)} \cdot (1/(\mu_i^{(e)})^2 + (\sigma_i^{(e)})^2)}{2(1 - \rho_i^{(e)})}, \quad e \in \{\text{PG, HA, CRRM}\}. \tag{9}$$

In addition, the average routing delay at an intermediate router (denoted by $T_i^{(R)}$) is assumed to be insignificant and be a constant [13]. The total end-to-end processing delay over h hops between entity A and entity B is given by

$$\sum_{e \in E_i^{(h)}} T_i^{(e)} = T_i^{(e_A)} + (h-1) \cdot T_i^{(R)} + T_i^{(e_B)}. \tag{10}$$

Finally, we estimate the length of the SOAP messages. We assume that the SOAP messages are textual-based and use IPsec Encapsulating Security Payload (ESP) transport mode for secure signalling. In addition, we assume the SEED-CBC algorithm [15] for encryption, and the HMAC-MD5-96 algorithm [16] for authentication. The IPv6 and UDP headers are assumed 40 bytes and 8 bytes, respectively. Accordingly, the length of an IPv6-level SOAP message consisting of n policies or replies can be derived as

$$L_{\text{SOAP}_i}(n) = 76 + \left\lceil \frac{10 + L_{\text{SOAP}_i\text{-hdr-payload}}(n)}{16} \right\rceil \times 16, \tag{11}$$

where $L_{\text{SOAP}_i\text{-hdr-payload}}(n)$ is the total length of the SOAP's own header and payload, and varies from the number of policies (in a Trigger or Flow Handoff Request message) or replies (every policy is replied in a Flow Handoff Reply message). Since SOAP is an application-level protocol, its length can hardly be precisely identified according to the message definition only. We estimate $L_{\text{SOAP}_i\text{-hdr-payload}}(n)$ based on empirical values regarding the length of a void (skeleton) SOAP-over-UDP message from [11] and express it as

$$\begin{aligned} L_{\text{SOAP}_i\text{-hdr-payload}}(n) &= (L_{\text{void-SOAP}_i\text{-UDP}} - L_{\text{UDP-hdr}}) + L_{\text{SOAP}_i\text{-hdr-extra}} \\ &\quad + L_{\text{SOAP}_i\text{-body}} \\ &\cong 450 + L_{\text{SOAP}_i\text{-body}} \\ &\cong 450 + \sum_{m=1}^{n} L_{\text{SOAP}_i\text{-body-m}}, \end{aligned} \tag{12}$$

where $L_{\text{void-SOAP}_i\text{-UDP}}$ is the length of a void SOAP message over UDP (including the length of the UDP header, $L_{\text{UDP-hdr}}$), $L_{\text{SOAP}_i\text{-body}}$ is the length of the body part of a SOAP message, $L_{\text{SOAP}_i\text{-hdr-extra}}$ is the length difference introduced by the header part of a SOAP message compared with that of a void SOAP message, and $L_{\text{SOAP}_i\text{-body-m}}$ is the length of a policy or a reply. $L_{\text{SOAP}_i\text{-body}}$ is estimated as the accumulative length of the n policies or replies enclosed in the body part of a SOAP message.

It is noted that an oversized message needs to be transmitted through more than one link-layer frame [17]. A message is "oversized" if its link-layer length is larger than the frame length limit or its network-layer length is larger than the IP maximum transmission unit (MTU). In the context of B3G broadband wireless system, the length limit of a frame is usually sufficiently large. For instance, in IEEE 802.11 standard the frame length limit in an 11-Mbps channel with frame duration of 3.5 ms is $\lfloor 11 \times 10^6 \times 3.5 \times 10^{-3}/8 \rfloor = 4{,}812$ bytes. Regarding the MTU in IPv6, the minimum MTU is standardised as 1,280 bytes. Given the range of typical values used in this paper, each of the involved messages can fit in a single frame for transmission. Additionally, the added length of the 802.11 link layer due to frame

Table 1 Default values for the input parameters

Parameter	Default value
n	4
N_{RM}	7
τ_0	0.5 s
r	2
β_i	10^{-5}
$L_{\text{SOAP}_i-\text{body}-\text{m}}$ (one policy, trigger, or reply)	100, 100, 15 (bytes)
$H_{\text{HA}-\text{PG}}$, $H_{\text{CRRM}-\text{HA}}$, $H_{\text{CRRM}-\text{PG}}$	20, 15, 35
$\mu_{\text{SOAP}}^{(\text{PG})}$, $\mu_{\text{SOAP}}^{(\text{HA})}$, $\mu_{\text{SOAP}}^{(\text{CRRM})}$	0.02 (messages/ms)
$\lambda_{\text{SOAP}}^{(\text{PG})}$, $\lambda_{\text{SOAP}}^{(\text{HA})}$, $\lambda_{\text{SOAP}}^{(\text{CRRM})}$	2, 2000, 2000 (messages/h)
$\sigma_{\text{SOAP}}^{(\text{PG})}$, $\sigma_{\text{SOAP}}^{(\text{HA})}$, $\sigma_{\text{SOAP}}^{(\text{CRRM})}$	0.3 ($\left(\sigma_i^{(e)}\right)^2$ in ms)
B_k^{wl}, B_k^{wd}	11, 100 (Mbps) [12]
T_k^{wl}, T_k^{wd}, $T_i^{(\text{R})}$	0.5, 2.0, 0.001 (ms) [12]

header and trailer is assumed 28 bytes including a 24-byte header and a 4-byte frame check sequence.

3.3 Analytical Results

To demonstrate numerical results and theoretically examine the flow handoff performances of Scheme I and Scheme II in various scenarios, Table 1 lists the default average values (unless stated otherwise) for the involved input parameters.

As can be seen from Table 1, we assume that on average four policies are involved in each flow handoff in the following discussions. It is noted from our implementation that the number of policies directly influences the processing delay of the corresponding SOAP message as will be discussed in the next section. For the four-policy case, the mean measured SOAP message processing delay is around 50 ms and thus we assume that the mean processing rate at all the signalling entities (PG, HA or CRRM) is $\mu = 0.02$ messages/ms. We also assume that the SOAP message arrival rate at a PG is a fraction of that at a CRRM or a HA since each CRRM or HA serves numerous PGs. The distance (in terms of hops) between the HA and the PG/CRRM is sufficiently large to represent a large-scale setting, where the PG is not in the home domain. Based on these assumptions and default values, we present selected numerical results as follows. Note that both signalling delays and processing delays contribute to the overall flow handoff delays, and hence we explore both of them, respectively.

Figure 7 illustrates the effects of the BER on the flow handoff delays given the processing delays at the signalling entities remain unchanged (i.e., message processing rate etc. are default values in Table 1). Figure 7 investigates the performances of Scheme I and Scheme II in two scenarios that employ two different typical initial retransmission timers (τ_0), 0.5 s and 1.0 s. These two timers are standardised as recommended values for message retransmissions over UDP in SIP and MIPv6, respectively. As shown in Fig. 7, the flow handoff delays increase with the rise of the BER since increasingly more retransmissions are invoked (more retransmissions, higher signalling delays). However, the increases are insignificant and delays are well under 1,000 ms (410–570 ms in Scheme I, 170–260 ms in Scheme II) in both scenarios when the BER is not larger than 10^{-5}. The differences between the two cases are also small. The reason is that when the BER is small retransmissions rarely occur and thus the processing delays dominate the overall delays. Only when the BER is greater than

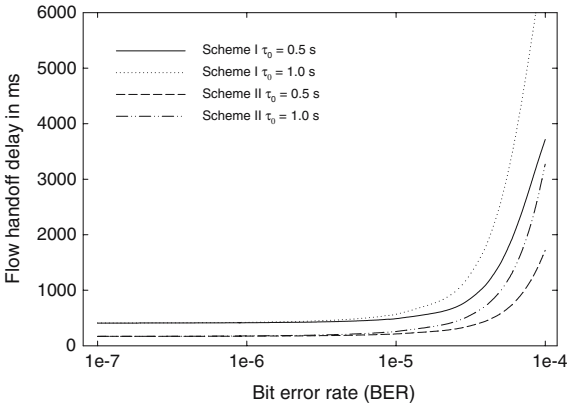


Fig. 7 Flow handoff delay vs. BER with different initial retransmission timers

approximately 2×10^{-5}, the delays in both cases start to grow exponentially and soon reach several seconds, which real-time applications may not be able to tolerate. Between the two scenarios in both schemes, the case of 0.5 s yields an evidently slower growth. When the two schemes are compared with each other, the delays generated in Scheme II are consistently lower than those in Scheme I. Intuitively, this difference results from the fact that Scheme II adopts a parallel signalling approach whilst Scheme I signals the policies serially. This finding is commonly observed in the remaining analytical results and thus it would not be repeated.

A similar story can be revealed when different retransmission coefficients (r) are studied (not illustrated here for brevity). To sum up, these results indicate that the flow handoff delays are within an acceptable range (though Scheme II outperforms Scheme I) as long as the wireless channel condition is normal with a BER not significantly worse than 10^{-5} and different values in the initial retransmission timer or coefficient do not make much difference. Note that 10^{-5} is the typical par value in Wi-Fi networks and should be satisfied normally.

Figure 8 further demonstrates the effects of SOAP message arrival rate at a PG $\left(\lambda_{\text{SOAP}}^{(\text{PG})}\right)$ on the flow handoff delays in the default wireless channel condition (i.e., BER = 10^{-5}). In Scheme I, the SOAP Flow Handoff Request message sent from the HA to the PG may generate a flow handoff subject to the PG's confirmation. This is also true for the SOAP Trigger

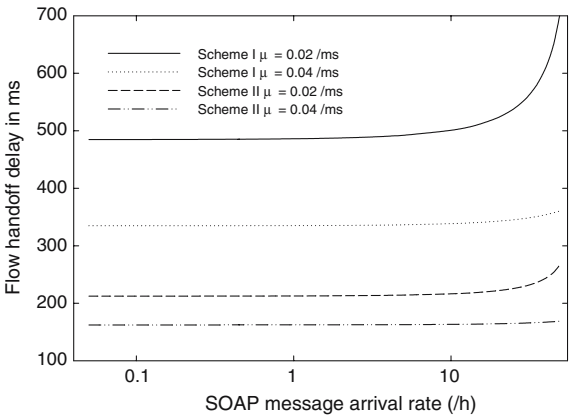


Fig. 8 Flow handoff delay vs. SOAP message arrival rate

message from the CRRM to the PG in Scheme II. Therefore, Fig. 8 also indirectly indicates how the flow handoff arrival rate (a fraction of the SOAP arrival rate) at a PG would affect the flow handoff delays. Generally, different processing rates of SOAP messages would incur different processing delays, which in turn account for the overall flow handoff delays. With the default processing rate ($\mu = 0.02$), the flow handoff delays rise significantly in Scheme I (from about 550 to 720 ms) only when the SOAP message arrival rate reaches well above 10 messages per hour for a given PG. In contrast, the delays in Scheme II do not increase significantly (from about 230 to 270 ms). When the message arrival rate is lower, the flow handoff delays are consistently around 500 ms and 220 ms in Scheme I and Scheme II, respectively; and the increases are not obvious (480–540 ms in Scheme I, 210–230 ms in Scheme II) with the growth of the message arrival rate.

It is expected that the processing delays can be significantly reduced with machines of higher specifications and/or new techniques in processing the SOAP messages, e.g., the binary characterisation [18] being investigated in the World Wide Web Consortium (W3C). To illustrate this promising perspective, Fig. 8 also plots two curves corresponding to an accelerated processing scenario where the processing rate is doubled ($\mu = 0.04$). Accordingly, the delays in both schemes drop radically (to 330–360 ms in Scheme I, to 160–170 ms in Scheme II) within the whole range of concerned message arrival rate.

4 Proof-of-Concept Implementation and Experimental Results

To validate the proposed architecture, proof-of-concept implementations and experiments have been carried out. In this section, we report the practical aspect and focus on the core functionalities.

4.1 Overview

A proof-of-concept testbed resembling the reference network model shown in Fig. 1 in principle has been set up. The handoff execution functionality was built upon the NEMO implementation for Linux (NEPL) with integrated MCoA support from Nautilus6 [7]. The Nautilus6 implementation consists of both user-space enhancements of the basic NEPL package and a kernel-space patch for Linux 2.6.15.

To enable advanced and dynamic policy-based flow handoff management, a framework was designed and implemented on the testbed for policy generation, transmission, reception, processing and enforcement. The testbed allowed us to test our implementation and to validate the interactions of the sub-modules for flow handoff support according to the changing policies. Moreover, the testbed permitted preliminary measurements on implementation performances. Currently, the multihomed PG is connected to the HA via two 802.11b/g WLAN ARs. Future work would port the implementations on an evolved wireless testbed by incorporating additional heterogeneous systems such as WiMAX although the current setup is sufficient for proof of concept in the IP level (L3).

With respect to the MCoA-enabled NEPL implementation, our implementation represents a complementary module as it enables advanced use of the multiple CoAs. From the application traffic's perspective, the multiple tunnels between the HA and the PG can be deemed as multiple bit pipes provided by MCoA whereas our implementation allows the dynamic redirection of traffic flows amongst these bit pipes. The design of our implementation and the operation mode were kept aligned with NEPL to facilitate the deployment of these two modules in parallel.

Furthermore, a tool named wimeter was designed and developed to estimate the interested QoS parameters such as traffic load and available bandwidth, which can then be used as input to the intelligent network selection algorithms to trigger flow handoffs. This tool has been independently validated in an 802.11 WLAN environment. Recently, the core functionalities described in this section have been successfully demonstrated to a panel of expert reviewers.

4.2 QoS Measurement

The wimeter tool was designed to measure the interested QoS metrics as input to the intelligent network selection algorithms to trigger QoS-aware flow handoffs. It captures and analyses on real-time the frames going to or coming from a pre-configured access point in order to estimate the traffic load and to compute the available bandwidth. To operate in an accurate manner, this tool requires as input the application packet size, the source and destination link sending rate, and the transfer mode (unicast, multicast, or broadcast). Figures 9 and 10 demonstrate a couple of performance test examples for verification of the tool.

The corresponding experiments were conducted for a single constant-bit-rate (CBR) traffic in an 802.11 WLAN. Figure 9 depicts the results obtained for the scenario where the source is increasing its sending rate from 0 to 10 Mbps. When the source sending rate increases,

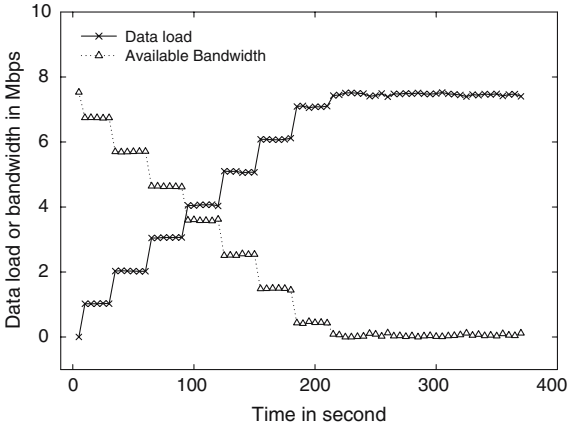


Fig. 9 Data load and available bandwidth measurement vs. sending rate over time

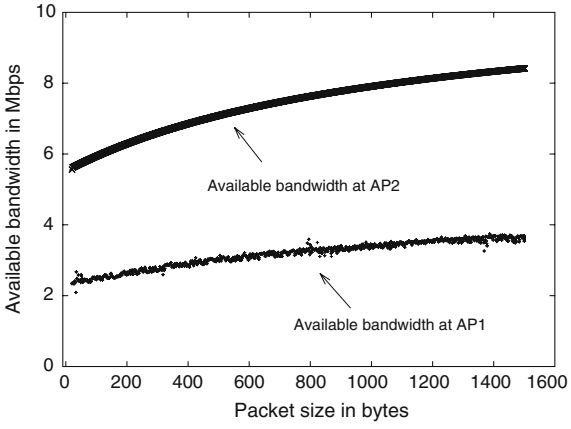


Fig. 10 Available bandwidth measurement vs. packet size

wimeter is able to consistently capture the new behaviour of the source and to correctly estimate the source data load (throughput) as well as the available bandwidth. Furthermore, the available bandwidth drops to 0 when the data throughput is close or larger than the saturation throughput (about 7.4 Mbps). Hence, when the source sending rate is 8, 9, 10 and 11 Mbps the data throughput and the available bandwidth remains constant.

In Fig. 10, we plot the variations of the available bandwidth versus different packet sizes for two access points. In this scenario, there is a CBR connection established toward the first access point (AP1) at the rate of 4 Mbps, whilst there is no connection using the second access (AP2). Two main observations can be made from this figure. First, as expected the difference between the available bandwidth at AP2 and AP1 is around 4 Mbps which is the rate of the active connection at AP1. This difference remain valid whatever the packet size for which we determine the available bandwidth. Second, the available bandwidth increases with the packet size given that sending large packets induces less time spent on backoff and defer periods.

4.3 Policy Processing

The integration of the QoS-measurement-based trigger is being implemented. To test the trigger, the CRRM is currently simplified as a policy generation engine, which dynamically generates policies and then sends the policies to the Policy Reception and Enforcement Module (PREM) instances on the HA and on the PG using either Scheme I or Scheme II. The policies are coded in XML in a similar syntax as proposed in the Flow Distribution draft [6] and transmitted from the policy generation engine to the HA and the PG over a web service. The web service client and server modules were developed using the Hypertext Preprocessor 5 (PHP5) [19], which has integrated C-based SOAP support.

The overall operation cycle of the PREM is shown in Fig. 11. Upon receiving a SOAP message, the PREM parses the message and derives the policies and their corresponding processing actions. Currently, two crucial actions including policy insertion and flush have been implemented though additional actions can be added later. For each policy, the PREM constructs and passes a command chain to ip6tables. Finally, the updated policies are enforced and ongoing or future application flows that are identified in the policies are distributed over desired interfaces.

It has been observed through repeated experiments that the processing delay of the whole process increases in a roughly linear way with the increase of the number of policies that are encoded in the SOAP message. Figure 12 depicts this observation in the range of 2–10

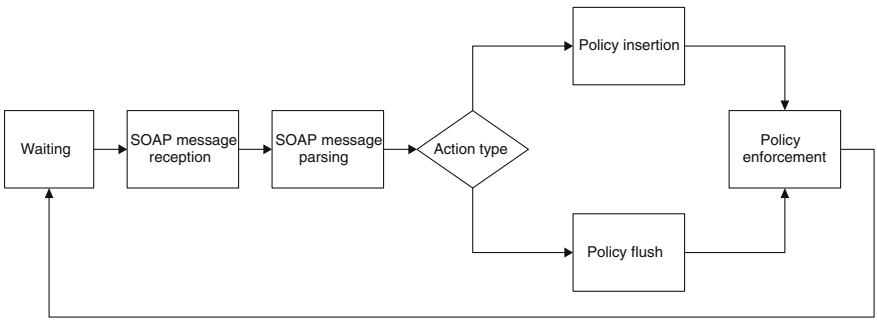


Fig. 11 Policy reception and enforcement module

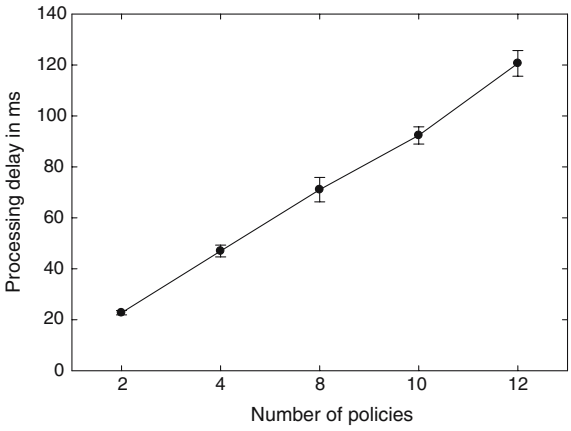


Fig. 12 Effect of policy numbers on processing delay

polices, resulting in processing delays from about 20–120 ms. Homogeneous policies (in the form of a protocol number and a BID binding) were applied in each case to exclude the effects of difference in policy complexity on the delay. The measurements were gathered from a Pentium M (1.73 GHz, 2 GB RAM) laptop.

Figure 13 further shows the detailed interactions between the daemons NEPL and PREM and the native Linux routing engine and the firewall in the control plane. As presented in Fig. 13, the user-space parts of NEPL and PREM are only involved for the handling of signalling message, extracting the essential information and passing it to the relevant kernel-space modules. Incoming Binding Updates with the Binding Unique Identifier sub-option are parsed and validated by the MCoA-enabled NEPL daemon. The resulting routes to the Mobile Router and the Mobile Network Prefixes are installed into the Linux routing table whose number is the BID. This means that in contrast to the basic NEPL implementation where all NEMO-related routes are installed in the same routing table, the MCoA implementation activates one routing table per BID, which is directly identified by the BID number. Regarding the PREM, it parses incoming policy files, extracts the relevant information and constructs the

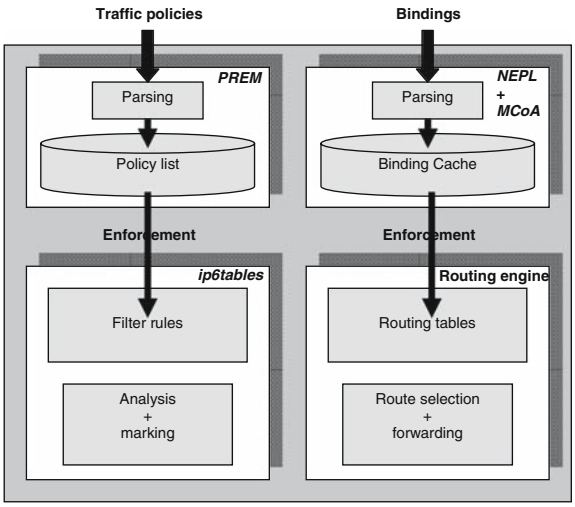


Fig. 13 Reception and processing of flow handoff messages

command chain for ip6tables with the correct syntax. Then, it enforces the policy by passing the command chain over to ip6tables.

4.4 Handoff Execution

Regarding the user data, incoming data packets are analysed, marked and forwarded according to the currently active filter rules and bindings as shown in Fig. 14. Ip6tables analyses incoming packets and if they match one of the traffic rules in the PREROUTING chain, they are marked with the corresponding BID in the MARK target. Thanks to the kernel patch provided by the MCoA implementation, this marker can be taken into account for the packet forwarding; and thus the packet can be redirected to the routing table whose index is the BID. Consequently, the route lookup is done on the correct routing table first and the packets are forwarded to the desired CoA. Note that on HAs, there can be multiple routes for several mobile networks in a given routing table, as different MRs (and thus PGs) might use the same BID. Nevertheless, the packets are forwarded correctly simply by using the packet destination address as search key amongst the different routes for the same BID.

To demonstrate the flow handoffs between interfaces, we show a case study here. Figure 15 represents a HTTP-based video stream experiencing traffic policy-based flow handoffs between two egress interfaces of a PG. Using the software Ethereal, we represent the IPv6 packets passing through the two egress interfaces as shown in Fig. 15. The time is expressed in seconds on the X axis, and the traffic in expressed in number of packets on the Y axis.

By analysing Fig. 15 step by step, we can reconstruct the different handoff processes. The HTTP video streaming traffic starts on interface IF1. The HTTP streaming traffic is sent from a server on the fixed network to a MNN in the PAN, and the TCP acknowledgements (ACKs) are returned by the MNN. HTTP packets are represented by the higher and the ACKs by the lower curves, respectively. In addition, we can see that the signalling traffic volume is very low compared wtih the HTTP streaming and the TCP ACK traffic, as expected. Signalling traffic is observed on both interfaces as it includes the periodical Router Advertisements sent by the ARs as well as the BUs and BAs to refresh the bindings. These messages are not subject to flow binding policies as they are not forwarded by the HA or are addressed to a MNN.

At the beginning of the transmission, all traffic is sent over IF1 as it is defined as the default egress interface of the PG. At $t1$, the policy generation engine sends a trigger message to the HA and to the PG with the new policies that associates HTTP traffic to interface IF2.

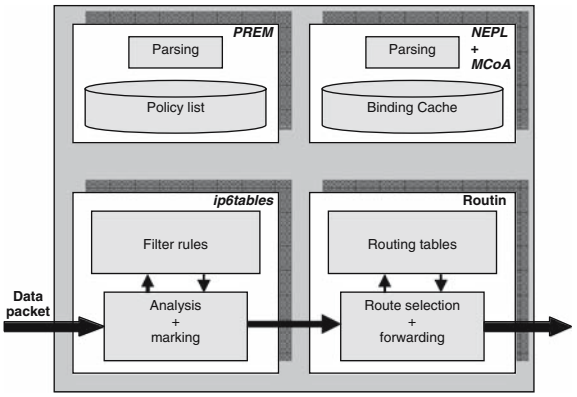


Fig. 14 Policy-based routing for data packets

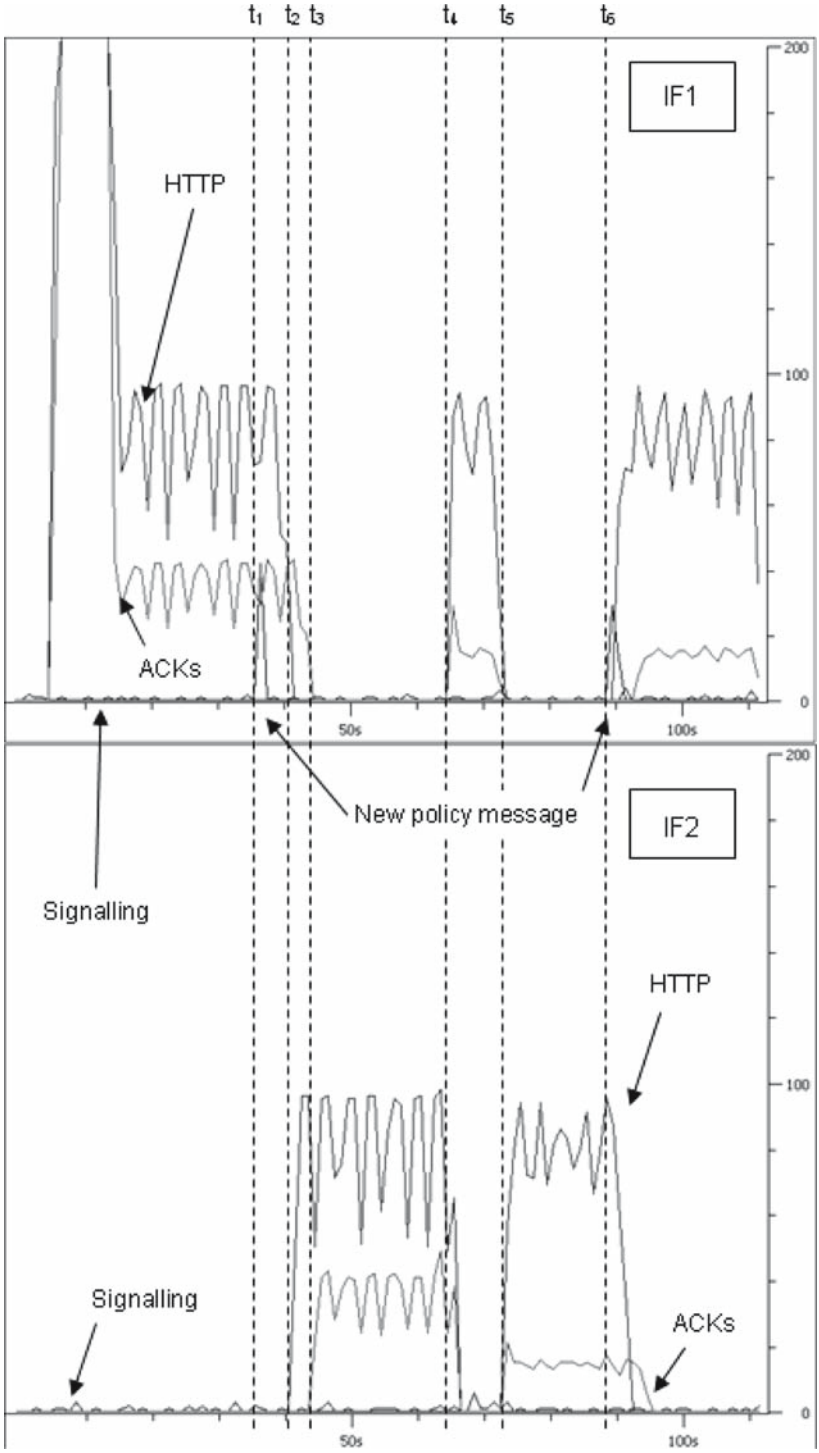


Fig. 15 Traffic captures on the PG for a video streaming with a flow handoff

This signalling traffic is also sent through IF1 since there is no explicit policy for this type of traffic.

At $t2$ and then at $t3$, the policies are enforced at the HA and the PG, respectively, and the HTTP packets as well as the ACKs are handed off to IF2 accordingly. As the trigger message is sent to the PG via the HA, it arrives with an additional delay in the PG. Consequently, the ACKs routed by the PG are handed off later than the incoming HTTP packets. However, this temporarily unsymmetrical traffic flow does not impact the end-to-end TCP connection, as neither the original IP header nor the TCP payload is modified and all packets always go through the HA and the PG.

At $t4$, IF2 performs a handoff from one access network to another. During the handoff, the traffic is automatically redirected to IF1 in order to provide a smother handoff and a higher perceived QoS to the user. At $t5$, this interface handoff is finished and the HTTP and TCP flows are switched back to IF2 according to the active policies.

At $t6$, another policy message is received. This one is a message that flushes all flow binding policies and thus the packets are switched to the default interface IF1 again.

This comprehensive scenario validates the capability of the implementation to dynamically hand off flows by using the multiple interfaces of the PG according to the policies and also the possibility to use one egress interface as backup for the other. Similar experiments as described above were repeated and observed for numerous times, and no perceptible disruptions were experienced during the whole course of these experiments from a number of different users' perspectives.

4.5 Handoff Signalling Performance

To further evaluate the objective performance of the dynamic policy-based flow handoffs, we measured the handoff signalling delays and traffic loads in Scheme I and Scheme II; and we compared the corresponding results as shown in Figs. 16 and 17, respectively. In these experiments, SOAP over HTTP/TCP was implemented to complement the analytical results in the context of SOAP over UDP. Regarding the hardware settings, the policy generation engine (i.e., the preliminary version of the CRRM), the HA and the PG uses a Pentium 4 (2.00 GHz, 768 MB RAM) PC, a Pentium D (2.80 GHz, 1 GB RAM) PC and a VIA Nehemiah (1.20 GHz, 512 MB RAM) PC, respectively. The PG is connected to the HA directly whilst the HA is one hop away from two Wi-Fi ARs that providing multiple wireless connections to the PG. The bit rate of the wireless hop between the HA and the PG is set to be 11 Mbps. One policy (in the form of a full five-tuple combination) and its symmetric counterpart are generated, signalled, processed and enforced in each experiment. Experiments were repeated and the mean values are reported here.

The handoff signalling delays include a three-way TCP handshake for each connection establishment between the nodes and the end-to-end round trip time for SOAP signalling initiated by the policy generation engine. The delays exclude the latency for fetching the web services description language (WSDL) files that define the proposed network services since that latency is only incurred at the first time before policies are ever transferred or when the local WSDL caches expire (the default expiration time is 24 h). Thus, the defined delays are more typical ones. Note that these experiment measurements shown in Fig. 16 cannot be compared with the analytical values directly since the analyses take into account a large-scale deployment scenario (and exclude the acknowledgement delay to the policy generation engine) whilst the experiments were performed on a local testbed for proof-of-concept assessments. Nevertheless, the experiment results reveal more implementation specific insights. Two cases were considered in implementing Scheme II. In one case (Scheme

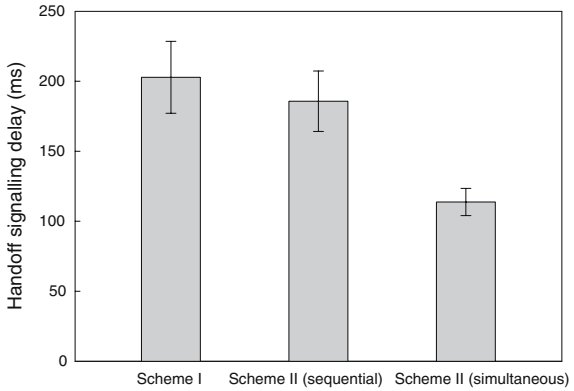


Fig. 16 Flow handoff signalling delays in Scheme I and Scheme II

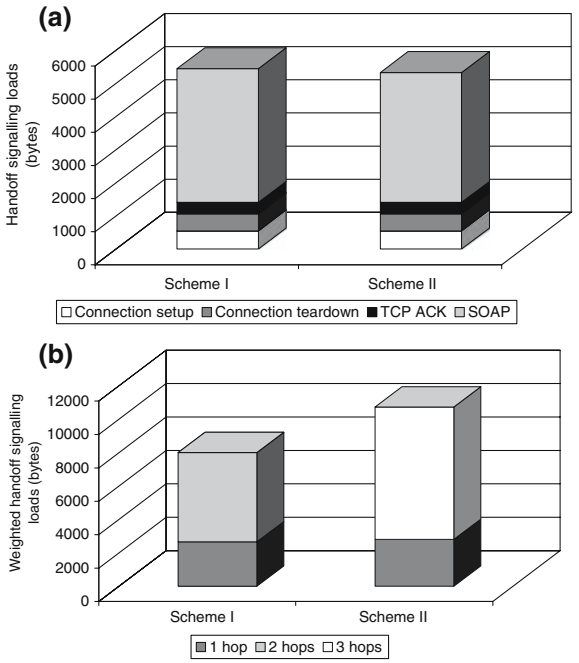


Fig. 17 Flow handoff signalling loads in Scheme I and Scheme II: (**a**) nominal loads, (**b**) weighted loads

II sequential), the policy for the HA and the symmetric policy for the PG are transferred to the HA and the PG sequentially from the policy generation engine. In another case (Scheme II simultaneous), each policy is sent simultaneously. In the PHP5 SOAP over HTTP, a TCP ACK is expected after a SOAP message is sent before another SOAP message can be sent. Therefore, Scheme II (sequential) is a natural choice and thus has been implemented and tested. Scheme II (sequential) results in a mean delay of 186 ms, slightly better than Scheme I (203 ms) with an 8% reduction. In contrast, in theory Scheme II (simultaneous) could be significantly faster than Scheme I if the policies could be transmitted simultaneously somehow. From an analysis of the Scheme II (sequential) trace file, the mean delay in Scheme II (simultaneous) would be 114 ms, 44% shorter than that of Scheme I. Furthermore, there are variations in the measurements in both schemes as an effect of the periodical background signalling traffic bursts for NEMO binding refreshes and so on as well as the fluctuations in

the quality of the wireless channel. Despite the differences in those cases, both schemes are quick enough to support dynamic flow handoffs.

The handoff signalling loads are the total signalling traffic generated from the SOAP messages, the corresponding TCP ACK messages and the TCP messages for connection establishments and releases in each experiment in the unit of bytes. We examine two kinds of loads: the nominal loads and the weighted loads, without or with the travel distances of the messages taken into account, respectively. As shown in Fig. 17a, the nominal signalling loads in both schemes are effectively the same: both are around 5,400 bytes. The proportions of the loads are also similarly distributed: the SOAP messages accounts for the majority of the loads (74%) whilst the TCP overheads contribute to the remaining (10% for connection setups, 10% for connection teardowns and 6% for TCP ACKs, from bottom to top in the stacked bar chart). This observation results from the fact that in either scheme a couple of TCP connections are required for the pair of SOAP policy message exchanges. On the other hand, from the system's perspective the weighted loads from Scheme I and Scheme II can be significantly different. The weighted loads of a message are calculated as the product of the length of the message and its travel distance (in terms of hops) from the source to the destination [20]. Given the setting of the testbed, as illustrated in Fig. 17b Scheme II yields 34% more loads in contrast to Scheme I because the messages in Scheme II travel over more hops. Therefore, although Scheme II tends to be faster it is important to deploy the system carefully to minimise the weighted traffic loads.

From the above experiment results, we may conclude that the two proposed schemes perform effectively as expected in supporting the policy-based flow handoffs; and the handoff signalling delays and traffic loads in both schemes are well acceptable. Further experiments are underway to examine the scalability of the proposed schemes.

4.6 Related Work and Comparisions

With the preliminary experiments described above, we have validated that the proposed MULTINET architecture supports a range of advanced capabilities that are desired by B3G users. These comprehensive capabilities include NEMO-based mobility, MCoA-enabled multihoming, dynamic XML-coded policy signalling, and network-supported QoS measurement to trigger sophisticated flow handoffs amongst multiple interfaces. The development of the proposed architecture has benefited from a number of existing implementations and proposals whilst significant extensions have been carried out. Table 2 summarises the similarities and the differences between the proposed architecture and a number of closely related work based on the network layer.

Amongst the relevant work, a couple of implementations are closely related to the MULTINET approach. The Mobile IPv6 for Linux (MIPL) [21] has implemented basic multihoming support through an interface preference mechanism so that a multihomed MH can use its multiple interfaces sequentially. Nevertheless, simultaneous use of multiple interfaces has not been defined. Furthermore, policy signalling or QoS awareness are still missing. As aforementioned, the Nautilus NEPL plus MCoA implementations [7] serve as building blocks for the mobility and multihoming support in the proposed Schemes I and II. The available Nautilus implementation itself, however, has not addressed dynamic policy signalling or QoS measurement.

The Flow Binding [5] and Flow Distribution [6] drafts proposed two mechanisms to signal the policies for flow handoffs; however, an implementation of neither is available. Moreover, network-supported QoS awareness is beyond the scope of both drafts. Regarding the policy (or trigger) initiator(s), only the MN is enabled in [5] whilst both the HA and the MN are

Table 2 Comparison with related work

	Mobility	Multihoming	Policy signalling	Policy/trigger initiator	QoS measurement	Source code
MIPL [21]	MIPv6	Limited	N/A	N/A	N/A	Available
NEPL + MCoA [7]	NEMO	MCoA	N/A	Command line	N/A	Available
Flow binding draft [5]	MIPv6/NEMO	MCoA	BU	MN	N/A	N/A
Flow distribution draft [6]	MIPv6/NEMO	MCoA	XML/SOAP	HA, MN	N/A	N/A
Hybrid flow handoff [22]	MIPv6/HMIPv6	MCoA	BU, BRR, ICMPv6	Network, MH	wimeter	To Do
Schemes I, II	NEMO	MCoA	XML/SOAP	Network	wimeter	Developed

enabled in [6]. In our schemes, a dedicated network entity (CRRM) is proposed to issue policies and it can be decoupled from a HA and be deployed anywhere as deemed appropriate by the service provider. The policies are determined by intelligent network selection algorithms, which in turn utilise real-time network measurements and user profiles as input. Therefore, the MULTINET approach provides a complete solution to QoS-aware policy-based flow handoff management.

In a previous work [22], we have also investigated a hybrid approach to manage both network- and user-initiated flow handoffs by enhancing MIPv6 or its variant Hierarchical MIPv6 (HMIPv6) [23] for multihomed MHs. Similar to [5], a MH can initiate a trigger via a BU message. From the network side, a trigger encoded in ICMPv6 is sent from an intelligent network server to the HA. The existing Binding Refresh Request (BRR) message is extended with a mobility option introduced to accommodate policies and the extended BRR is allowed to be sent from a HA to a MH. In future work, this alternative approach would be implemented and compared with the proposed Scheme I and Scheme II.

5 Conclusion

It is widely envisioned that the B3G users would have high expectation on ubiquitous and personalised services with acceptable QoS control. The MULTINET project aims to provide enabling technologies to fulfil a seamless and customised service paradigm. In this paper, we have presented a QoS-aware network-supported architecture as an emerging solution to the challenges foreseen by both academia and industry. System entities are introduced, signalling procedures are presented, and up-to-date implementations are described with future work identified. Both theoretical analyses and proof-of-concept experiments are conducted to evaluate the performance of the proposed architecture, and the results demonstrate that the proposed architecture is promising and solid. The analytical results show that the flow handoff delays in typical WLAN conditions are satisfactory, whilst the preliminary experiments have validated the design of the core functionalities and provided pilot assessment of the handoff performances.

It is expected that the proposed multihoming and dynamic flow handoff mechanisms, integrated with QoS measurement means such as wimeter and intelligent network selection algorithms, constitute the base for providing efficient Always Best Connected (ABC) services as commercial offer to B3G customers.

Acknowledgements This work has been partly funded by the EU IST Project MULTINET: Enabler for Next Generation Service Delivery (No. IST-2005-027437). We would like to thank all the MULTINET project partners for their contributions during the development of various ideas presented in this paper. In particular, we would like to acknowledge Stephen Bell, Dr. Alisdair McDiarmid and Christopher Nicolson, all with the University of Strathclyde (USTR), for their contributions to the development of the USTR MULTINET testbed.

References

1. Yabusaki, M., Okagawa, T., & Imai, K. (2005). Mobility management in all-IP mobile network: End-to-end intelligence or network intelligence? *IEEE Communications Magazine, 43*(2), supl.16–supl.24.
2. Devarapalli, V., Wakikawa, R., Petrescu, A., & Thubert, P. (2005). Network mobility (NEMO) basic support protocol. IETF RFC 3963.
3. Johnson, D. B., Perkins, C., & Arkko, J. (2004). Mobility support in IPv6. IETF RFC 3775.
4. Wakikawa, R., Ernst, T., & Nagami, K. (2006). Multiple care-of addresses registration. IETF Internet Draft, <draft-wakikawa-mobileip-multiplecoa-05.txt>, work in progress.
5. Soliman, H., Montavont, N., Fikouras, N., & Kuladinithi, K. (2006). Flow bindings in mobile IPv6. IETF Internet Draft, <draft-soliman-monami6-flow-binding-02.txt>, work in progress.
6. Mitsuya, K., Tasaka, K., & Wakikawa, R. (2006). A schema fragment for flow distribution. IETF Internet Draft, <draft-mitsuya-monami6-flow-distribution-00.txt>, work in progress.
7. Nautilus6 project. Retrieved March 30, 2007, from http://www.nautilus6.org/nemo/.
8. Stewart, R., Xie, Q., Morneault, K., et al. (2000). Stream control transmission protocol. IETF RFC 2960.
9. Koh, S. J., Chang, M. J., & Lee, M. (2004). mSCTP for soft handover in transport layer. *IEEE Communications Letters, 8*, 189–191.
10. Lazaro, O., Gonzalez, A., Aginako, L., Hof, T., Sidoti, F., Vaquero, P., et al. (2006). *MULTINET: Enabler for next generation services*. Paper presented at the 17th Wireless World Research Forum (WWRF) Meeting, Heidelberg, Germany.
11. Phan, K. A., Tari, Z., & Bertok, P. (2006). *A benchmark on SOAP's transport protocols performance for mobile applications*. Paper presented at 2006 ACM Symposium on Applied Computing, Dijon, France.
12. Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., et al. (2002). SIP: Session initiation protocol. IETF RFC 3261.
13. Lo, S.-C., Lee, G., Chen, W.-T., & Liu, J.-C. (2004). Architecture for mobility and QoS support in all-IP wireless networks. *IEEE Journal on Selected Areas in Communications, 22*, 691–705.
14. Wang, W., & Akyildiz, I. F. (2001). A new signalling protocol for intersystem roaming in next generation wireless systems. *IEEE Journal on Selected Areas in Communications, 19*, 2040–2052.
15. Lee, H. J., Yoon, J. H., Lee, S. L., & Lee, J. I. (2005). The SEED cipher algorithm and its use with IPsec. IETF RFC 4196.
16. Madson, C., & Glenn R. (1998). The use of HMAC-MD5–96 within ESP and AH. IETF RFC 2403.
17. Banerjee, N., Wu W., Basu, K., & Das, S. K. (2004). Analysis of SIP-based mobility management in 4G wireless networks. *Computer Communications, 27*, 697–707.
18. Goldman, O., & Lenkov, D. (Eds.). (2005). XML binary characterisation. W3C Working Group Note, work in progress.
19. Hypertext Preprocessor. Retrieved March 30, 2007, from http://www.php.net/.
20. Wang, Q., & Abu-Rgheff, M. A. (2006). Mobility management architectures based on joint mobile IP and SIP protocols. *IEEE Wireless Communications, 13*, 68–76.
21. Mobile IPv6 for Linux (MIPL). Retrieved March 30, 2007, from http://mobile-ipv6.org/.
22. Wang, Q., Atkinson, R., Cromar, C., & Dunlop, J. (2007). *Hybrid user- and network-initiated flow handoff support for multihomed mobile hosts*. Paper presented at the 65th IEEE Vehicular Technology Conference (IEEE VTC2007-Spring), Dublin, Ireland.
23. Soliman, H., Catelluccia, C., Malki, K. E., & Bellier, L. (2005). Hierarchical mobile IPv6 mobility management (HMIPv6). IETF RFC 4140.

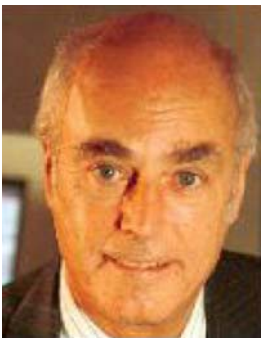
Author Biographies

Qi Wang received his B.E. in Electronic Engineering and M.E. in Communication and Electronic System from Dalian Maritime University, China, in 1995 and 1998, respectively; and his Ph.D. in Mobility Support Architectures for Next-Generation Wireless Networks from the University of Plymouth, UK, in 2006. He was granted a multi-year British ORS award for his Ph.D. programme. From 1998 to 2001, he was with the State Grid Corporation of China (Shandong) as an ICT engineer. Since 2006, he has been working on an EU IST FP6 project MULTINET as a postdoctoral research fellow with the University of Strathclyde, UK. His current research interests include IP networks, mobility management and multihoming support. Dr Wang is a member of IEEE.

Tobias Hof got the diploma in Electronical Engineering and Information Technologies of the University of Stuttgart in 2004 and the diploma in Telecommunications from the Ecole Nationale Supérieure des Télécommunications (ENST—Graduate School in Telecommunication in Paris) in 2005. He entered the department for Advanced Information Technologies of Thales Communications France in 2005. He is currently involved in French-funded (REMORA) and European IST FP6 (Multinet, Anemone) projects. In the context of these projects, he is working on topics such as IPv6, advanced mobility protocols and security aspects.

Fethi Filali received his Computer Science Engineering and DEA degrees from the National College of Informatics (ENSI) in 1998 and 1999, respectively. At the end of 1999, he joined the Planète research team at INRIA (National research institute in informatics and control) in France to prepare a Ph.D. in Computer Science which he has defended on November 2002. During 2003, he was an ATER (Attaché Temporaire d'Enseignement et de Recherche) at the Université of Nice France (UNSA) and he joined on September 2003 the Mobile Communications department of Institut Eurécom in France as an Assistant Professor. He is/was involved in several French-funded (Dipcast, Constellation, Rhodos, Cosinus, Airnet, WiNEM) and IST FP6 (Widens, Newcom, Daidalos, E2R, Multinet, Unite, Chorist) projects. In the context of some of these projects, he designed and developed an open, flexible and efficient architecture for the support of heterogeneous radio technologies. This architecture was integrated in Eurecom's wireless software-radio platform. His current research interests include WIMAX (802.16)-related communication mechanisms, QoS support in IEEE 802.11-based networks, sensor and actuator networks (SANETs), vehicle adhoc networks (VANETs), routing and TCP performance in wireless networks. He served as a technical reviewer of several international conferences and journals. Additionally, he is a member of IEEE and IEEE Communications Society.

Robert Atkinson is a Lecturer in the Institute for Communications and Signal Processing, University of Strathclyde, UK. He completed his Ph.D. in Mobile and Wireless Communications at Strathclyde in 2003. Throughout his time at the University he has worked on a variety of topics from Medium Access Control to Heterogeneous Networking. He is actively researching intelligent access network selection, user mobility solutions and Ad Hoc Networking. Dr. Atkinson is a member of IEEE and IET.



John Dunlop received the B.Sc. degree in electrical engineering from University College of Swansea in 1966 and the Ph.D. degree in telecommunications from the University of Wales in 1970. He is currently a Professor of Electronic Systems Engineering and Head of the Mobile Communications Group, University of Strathclyde, UK. He has recently completed a three-year term as a Director of the UK Virtual Centre of Excellence in Mobile and Personal Communications. He has been involved in research programs in communication systems and electronic systems engineering for more than two decades. This includes participation in RACE Definition Phase, RACE Mobile Communications Project (R1043), RACE Advanced Time Division Multiple Access ATDMA (R2084), and ACTS Mobile Communications Services for High Speed Trains MOSTRAIN (AC104) and as a full academic member of the UK Virtual Centre of Excellence in Mobile and Personal Communications. He has held several UK Engineering and Physical Sciences Research Council (EPSRC) awards on local area communications, underwater communications, and mobile communications and holds contracts from the Mobile VCE covering work in the areas of Networks and Services. He is also holder of several contracts with mobile communications companies. He is author and co-author of more than 150 scientific papers on Electronics Systems Engineering and Communications Engineering in international journals and conferences. He is co-author of *Telecommunications Engineering* which has been adopted as a standard text in many British Universities and a co-author of *Digital Mobile Communications and the TETRA System* (New York: Wiley, 1999).

Eric Robert holds a engineer degree in Electronics and a Master of Science in Network Architecture from the Ecole Nationale Supérieure des Télécommunications (ENST—Graduate School in Telecommunication located in Paris). After having worked for 4 years for a French telecommunication manufacturer specialised in broadband corporate access equipment, he joined Thales in 2001. He is a project manager of international project, ITEA Project, Mobilizing the Internet and national funded project, Infradio. He is expert in quality of service, QoS, traffic engineering, network mobility and IPv6.

Leire Aginako received her B.Sc. in Computer Engineering by the University of Deusto in 1993 and M.Sc. on Advanced Manufacturing Technologies by the Engineering School of the Basque University in 1994. She started her professional development in CARSA, an engineering company. She performed several responsibility tasks there, both in the Brussels delegation and Bilbao headquarters. She has managed and developed several RTD projects, in European, National and regional level, evaluated regional projects, managed evaluation of ESPRIT programme, supported the Basque IRC and managed the Delegation of the company in Brussels, since December 1999. Since March 2000, she is working in the RTD Department in Euskaltel S.A. (Telecom Operator). She is responsible of several projects at EU, National and regional level and support to the management of the department. Furthermore, since year 2000, she is expert evaluator for IST, Co-operation and SMEs and eTEN Programmes of the European Commission. She currently combines her responsibilities at Euskaltel, S.A. with her Ph.D. studies in the field of "IT and Communications in mobile networks". Her research interests lie in the area of mobility management and security in wireless networks. The areas of activity of the projects carried out are related to the following technological and innovation domains: IP Mobility for Wireless Communications and Location Based Services, eRural & eInclusion, eDemocracy, eLearning and eCommerce, eHealth, iTV Safe use of the Internet, Advanced Information Systems implementation in Public Administration, eLearning, Manufacturing Process Improvement and Software development process Improvement projects for SMEs, Technical support to the Basque Government related to their participation in the European IRC network, and several responsibilities within the Support to European Commision in the Evaluation management of ESPRIT programme, during FP4.