Vertical Handover Support for Multimode Mobile Terminal using Multi-Homed MIPv4

Tansir Ahmed[†], Kyandoghere Kyamakya[*], Markus Ludwig[†], Kalenga Wa Ngoy Cyrille[**], and Kalombo Masimango Monique S.[‡]
[†]BenQ Mobile, CM ITM DT DC, Munich, Germany
[*]Institute of Informatics Systems (ISYS), University of Klagenfurt, Klagenfurt, Austria
[**]Faculté Polytechnique, Université de Kinshasa, Kinshasa, DR Congo
[‡]Institut Supérieur de Commerce, Kinshasa, DR Congo

Abstract

Basic Mobile IPv4 (MIPv4) operation constraints applications bound to the same Home Address (HoA) to use the same physical interface because traffic is only tunneled towards a single Care-of-Address (CoA) by the Home Agent (HA). This prevents different applications from using different physical interfaces. Solutions to address this problem have been proposed that require configuration of filters in the HA to tunnel traffic towards different CoAs but these require modifications to the network, and additional signaling between the mobile terminal (MT) and the network. This paper describes a solution to support multimode operation using MIPv4 with no changes to the network, as well as the multimode MT architecture supporting such MIPv4 operation.

I Introduction

Rapid growth of mobile computing and emergence of diverse wireless access technologies like GPRS, UMTS, WiFi, etc lead to a common desire – to be connected "anytime, anywhere, anyway, and to anything". In order to facilitate this demand the concept of all-IP wireless network beyond 3G comes forth, which provides IP-based communication on top of diverse wireless access technologies in heterogeneous environment. MIPv4 [1] delivers a global mobility solution that provides host mobility management for a diverse array of heterogeneous systems.

MIPv4, briefly described here (see Fig. 1), is designed for use on a single-mode MT. It allows an MT to be continually reachable at a well known home IP address, called HoA, regardless of any mobility events that change the MT's point of attachment to the network. The MT also has a CoA that represents its current point of attachment to the network. Within the user's home network, there is a HA that holds a mapping between the MT's HoA and its current CoA. Packets addressed to the HoA are intercepted by the HA and tunneled to the CoA, and therefore, the MT at its current location. When the MT moves it updates the HoA with its new CoA.

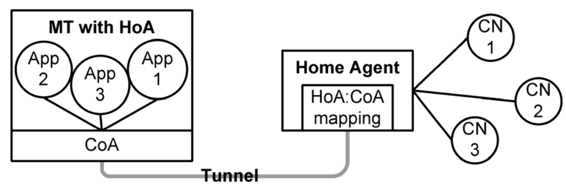

Fig. 1. Standard MIPv4

All traffic destined for the MT is tunneled to a single CoA, so effectively all active sessions are handed over together when the device changes point of attachment. For a single-mode MT where all applications use the same single interface this solution works adequately. However, when an MT has more than one interface, i.e. multimode MT, this solution means that all applications still have to use the same interface irrespective of their individual requirements, which is totally undesirable. This is because basic MIPv4 only supports a simple one-to-one mapping between HoA and CoA.

In order to combat the above constraint, [2] and [3] suggest MIPv4 extensions. In [2], HA and MT need to perform complex routing functions based on user profile. It also proposes a complex network management scheme for updating routing tables in HA and MT, handoff and single IP-multiple MAC mapping. In [3], HA needs to be informed of the application's QoS status and available resources on any interface to help it schedule packets using a complex scheduling algorithm. But it does not explain how the parameters are estimated and information conveyed; rather assume such a mechanism exists. [4] and [5] deal only with MIPv6 and propose some extensions to the standard protocol. The concept of having one HoA per interface in order to realize vertical handoff in multimode MT is proposed in [4], whereas, [5] specifies a set of rules (filters) for bindings that are transmitted to HA and corresponding node (CN), who in turn use this information to determine whether and where to route flows associated with MT.

The aim of our solution, described in section II, is to enhance current MIPv4 enabled MTs to support multimode operation (e.g. vertical handover) with no changes to the network or standard protocol. Section III describes the multimode MT architecture supporting such MIPv4 operation. Finally, section IV concludes the paper.

II Multi-homed MIPv4

In the solution, shown in Fig. 2, an MT is allocated multiple HoAs, where the HoA implicitly identifies a HA. Each HoA is associated with a single CoA at any one point in time. Applications with a common set of attributes, e.g. similar QoS requirements, similar cost constraints, etc are grouped together and use the same HoA for their communications. When this CoA changes, the update to the mapping in the HA ensures that traffic for that group of applications is routed to the current point of attachment, but does not force any other applications using alternative HoAs to handover as well.

The solution makes use of the concept of a MIPv4 virtual interface (VIF). In single mode operation of MIPv4, there is a MIPv4 VIF between the application and the physical interface (PIF). VIF is concerned with modifying the packets generated by the application (either by re-writing packet headers, or by encapsulation) to include the CoA instead of the HoA. VIF is bound to a PIF, which is used by all applications bound to that HoA. On a multimode MT, a VIF can be bound to different PIFs to adapt to interface availability and application requirements, but with a single PIF at any one point in time. On the contrary, a PIF can have more than one VIF associated with it. The number of MIPv4 VIFs is limited to the number of HoAs available on the MT, with a one-to-one mapping between VIF and HoA.

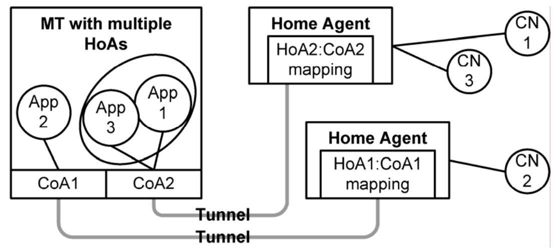

Fig. 2. MIPv4 support for multimode MTs

In this solution, the operation of MIPv4 protocol remains unchanged. The implementation of MIPv4 in HA is the same as that for basic MIPv4, i.e. HA intercepts any traffic destined for the HoA, and simply tunnels it to the current CoA it has in its mapping table. However, on the terminal side certain functionalities need to be implemented. Firstly, the implementation of MIPv4 has to be multimode enabled, i.e. it has to handle multiple HAs and interwork with the multimode functions described in the next section. Secondly, applications must be assigned to the most appropriate VIF (and therefore bind to the appropriate HoA). Finally, some means to group applications with similar requirements are needed, as this application group will move between PIFs as an aggregate.

III MIPv4 enabled terminal architecture

A. Functional modules

Over the past few years we have developed a reference architecture (see Fig. 3) for use within multimode MTs [6], which has been extended here to become MIPv4 enabled.

The Light Network Capability Discovery (LNCD) module periodically monitors all interfaces and discovers the capabilities of any active interface. It stores the Network Profiles (NP) into the Storage. NP contains information about all interfaces in the MT including the VIFs. The Storage also stores other specific groups of information, namely, Application Profiles (AP), User Profiles (UP), Connection Profiles (CP), and Running Application Information (RAI). All applications inside the MT are classified into five service types, namely, voice, real-time streaming, non-real-time streaming, interactive and background services. User preference scores, QoS requirements and other preferences of individual type of application are stored together in AP, where individual service type is identified by the application type tag [6]. UP contains information about user preferences, whereas CP contains information regarding active connections (e.g. port, transport protocol). Finally, RAI contains information related to active applications. Sample data structures for the Storage are shown in Fig. 4.

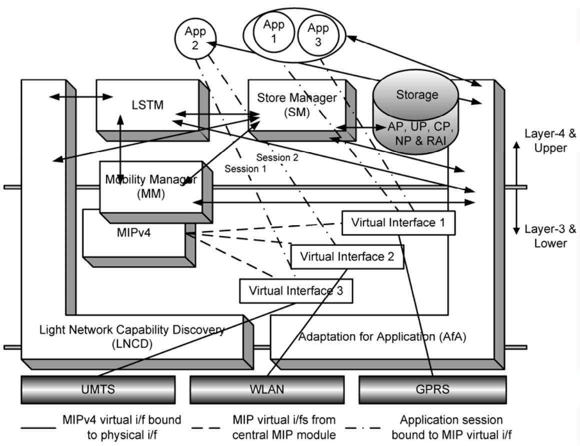

Fig. 3. MIPv4 enabled multimode MT architecture

The Store Manager (SM) acts as a gateway for information storing and retrieval from the Storage. It informs the Mobility Manager (MM) as soon as the applications on the VIF change. A very important functionality of SM is the execution of the Requirements Aggregation Function (RAF) discussed in the next subsection. MM interfaces MIPv4 to the multimode functions within the terminal, and provides certain management functions that are needed to handle application groups. It takes care of binding of VIF to PIF, as well as handover of VIF between PIFs.

The Light Session Transfer Management (LSTM) module acts as a middleware between upper and lower layers and processes the vertical handover decision algorithm [6]. The algorithm selects suitable VIF and PIF when requested by the application and the MM, respectively. The hierarchical decision making approach of the decision model is illustrated in Fig. 5. In this approach, available options (wireless networks with network contexts) are compared in terms of each predefined objectives (defined by users and terminal contexts) in order to determine their relative suitability. Similarly, the objectives are compared with each other in order to determine their relative importance. Finally, the best option is determined on the basis of these two sets of data.

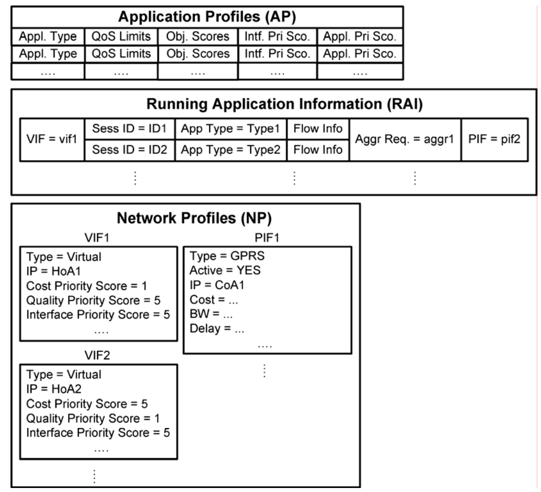

Fig. 4. Sample data structures for the Storage

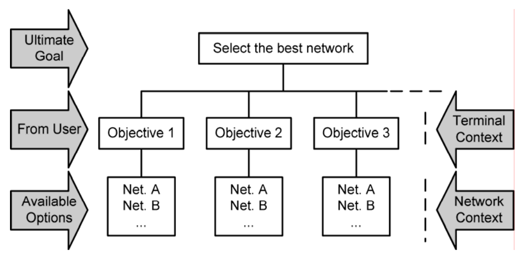

Fig. 5. Decision hierarchy

The Adaptation for Application (AfA) module interfaces all applications to the multimode functions within the terminal. It also manages IP addresses associated with VIFs and PIFs, and provides appropriate addresses when requested. In the process, it retrieves CoAs from PIFs after their address configuration, whereas HoAs for VIFs are preconfigured ones. Our architecture consists of three VIFs (i.e. three HoAs), each representing one specific objective among the user defined objectives – "Lowest Cost", "Best Quality" and "Desired Interface", as defined in [6]. Consequently, each VIF is assigned with the highest objective priority score (i.e. score 1) for a specific objective, as shown in Fig. 4.

B. Requirements aggregation function (RAF)

The vertical handover decision algorithm takes into account priority and objective scores (defined by users) and QoS requirements (predefined upper and lower limits) associated with each running application [6]. RAF calculates overall scores and QoS limits (upper and lower) of a group of applications, running together on a VIF, from the scores and limits provided by each application. This component is implemented inside SM. The aggregate functions for each parameter required for the decision algorithm are the following:

1) Mean bandwidth: Mean bandwidth is measured in bits per second. It correlates to the token rate *r* of the token bucket specification in [7]. Each application requires its own mean bandwidth value. Therefore, if the interface is to support all the applications at once, it must be able to support the total of all the bandwidth requirements. This applies to both the minimum (lower limit) and the preferable (upper limit) requirements because the mean bandwidth is that which is expected to be used by the application under normal running conditions. Consequently, for *n* number of applications:

$$agg(BW)_l = \sum_{i=1}^{n} BW_{il} \qquad (1)$$

$$agg(BW)_u = \sum_{i=1}^{n} BW_{iu} \qquad (2)$$

In (1) and (2), $agg(BW)_l$ and $agg(BW)_u$, respectively, are the aggregate lower and upper limits for mean bandwidth, and BW_{il} and BW_{iu}, respectively, are the lower and upper limits of mean bandwidth for the i^{th} application.

2) Delay, jitter and error rate: Delay (D) is measured in milliseconds. Value (v) of delay is acceptable for application i if $v < D_i$ for all i, which implies that $v < \min\{D_1, D_2, ..., D_n\}$. Jitter (in milliseconds) and error rate (bit error rate, BER) are also measured using the same principle. So, for *n* number of applications grouped together:

$$agg(X)_l = \min\{X_{1l}, X_{2l}, ..., X_{nl}\} \qquad (3)$$

$$agg(X)_u = \min\{X_{1u}, X_{2u}, ..., X_{nu}\} \qquad (4)$$

In (3) and (4), X can be delay (D), or jitter (J), or error rate (BER). Here, $agg(X)_l$ and $agg(X)_u$, respectively, are the aggregate lower and upper limits of parameter X, and X_{il} and X_{iu}, respectively, are the lower and upper limits of parameter X for the i^{th} application in the group.

3) Application priority score: The application priority score is a single value between 1 and 9 associated with each application [6]. The priority value increases with the importance of the application. In order to ensure that important applications do not suffer because of aggregation, the aggregate application priority should reflect the highest individual priority among the group. Therefore, for *n* number of applications grouped together:

$$agg(AP) = \max\{AP_1, AP_2, ..., AP_n\} \qquad (5)$$

In (5), $agg(AP)_l$ and AP_i, respectively, are the aggregate application priority and the individual application priority for the i^{th} application.

4) Interface priority scores: The interface priority scores are a set of scores between 1 and 9 associated with each application that represents the relative priorities among active interfaces in the MT [6]. The aggregate interface priority should consider individual interface priorities defined for each active interface by each application. Therefore, the aggregate interface priority for each active interface should be the average value of the individual priority scores specified for that particular interface by each application. That means for *n* number of applications and *m* number of active interfaces:

$$agg(IP) = \begin{bmatrix} \text{Avg}(IP_{11}, IP_{21}, ..., IP_{n1}) \\ \text{Avg}(IP_{12}, IP_{22}, ..., IP_{n2}) \\ ...... \quad ...... \quad ...... \quad ...... \\ \text{Avg}(IP_{1m}, IP_{2m}, ..., IP_{nm}) \end{bmatrix} \qquad (6)$$

In (6), $agg(IP)$ and IP_{ij}, respectively, are the aggregate interface priority scores and the individual interface priority score of the i^{th} application in the group for the j^{th} type of interface.

5) Objective priority scores: The objective priority scores are a set of scores between 1 and 9 associated with each application that represents the relative priorities among user defined objectives [6]. The aggregate objective priority scores are calculated in the same way as described for the interface priority scores. Therefore, for *n* number of applications and *m* number of objectives:

$$agg(OP) = \begin{bmatrix} \text{Avg}(OP_{11}, OP_{21}, ..., OP_{n1}) \\ \text{Avg}(OP_{12}, OP_{22}, ..., OP_{n2}) \\ \cdots\cdots \quad \cdots\cdots \quad \cdots\cdots \quad \cdots\cdots \\ \text{Avg}(OP_{1m}, OP_{2m}, ..., OP_{nm}) \end{bmatrix} \quad (7)$$

In (7), *agg(OP)* and OP_{ij}, respectively, are the aggregate objective priority scores and the individual objective priority score of the i^{th} application in the group for the j^{th} objective.

C. Operational phases

In our architecture, it is assumed that the MT only operates using MIPv4 collocated care-of-addresses (as opposed to operating with a Foreign Agent).

1) Assignment of application to VIF: In this phase (see Fig. 6), an application, during startup, is assigned to a VIF based on the decision of the decision algorithm (LSTM). At first, an application registers a session with AfA by informing its session ID and application type. Note that a single application can have multiple sessions of the same or different types. AfA then requests LSTM to select appropriate VIF for this type of application. LSTM requests SM to retrieve the objective priority scores set by the user for this specific type of application from the Storage, AP. Based on these scores, appropriate VIF is selected from the Storage, NP. As described earlier, since each VIF represents one specific objective and each type of application has its own objective preference, it is straight forward to map applications to appropriate VIF in accordance with their objective preferences. LSTM sends the selected VIF along with its HoA to AfA, which in turn sends the address to the application. At the same time, AfA updates the Storage, RAI, with session ID and type, assigned VIF, flow info, etc. Finally, the application binds itself to the VIF i.e. to the HoA.

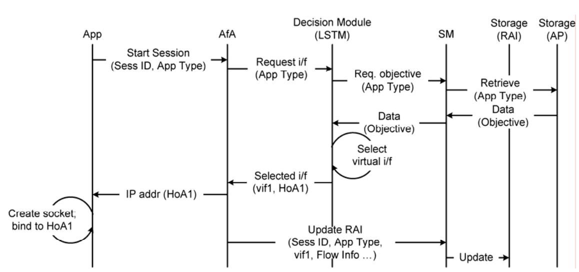

Fig. 6. Assignment of application to VIF

2) Binding of VIF to PIF: This phase (see Fig. 7) follows phase one when a new application is just bound to a VIF, which has no other associated applications and is not yet bound to a PIF, i.e. during startup. The whole phase is triggered by SM when it announces to MM that the applications on the VIF have changed. Upon request from MM, SM retrieves the requirements of the new application and the aggregate requirements of the applications currently bound to the VIF (if any) from the storage, AP and RAI, respectively. The aggregate requirements will be zero in this case. Using the retrieved data, SM calculates the new aggregate requirements of all the applications (only one application in this case) using the RAFs described earlier. MM gets the aggregate requirements from SM and checks whether the VIF is already associated with any PIF. Since the VIF is yet to be bound to any PIF, MM requests LSTM to select a suitable PIF, which fulfills the aggregate requirements of the applications bound to the VIF. Using the decision algorithm [6], LSTM selects a suitable PIF. MM then requests AfA for the IP address (CoA) configured for the selected PIF. AfA retrieves the CoA (either directly from the PIF or from the Storage, NP) and sends it to MM. Finally, MM binds the VIF to the selected PIF i.e. to the CoA, and updates the storage, RAI, with assigned VIF, aggregate requirements, selected PIF, etc.

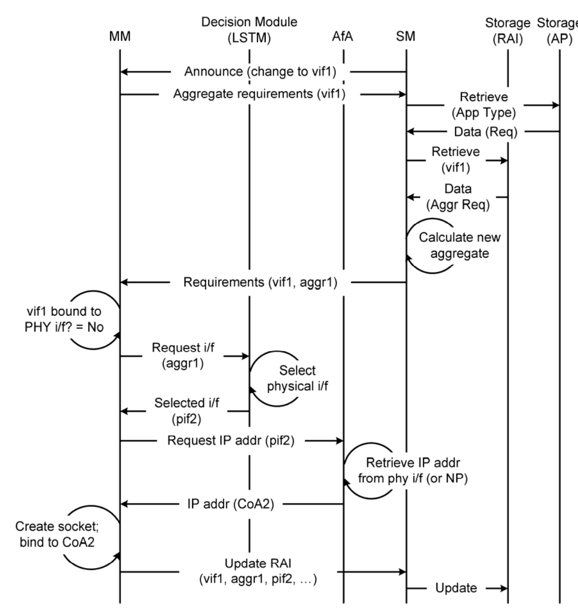

Fig. 7. Binding of VIF to PIF

3) Change of application on VIF: This phase illustrates the dynamic addition or deletion of an application to or from a VIF. This phase follows phase one when a new application is assigned to a VIF during runtime i.e. when the VIF, with one or more running applications, is already bound to a PIF. The trigger comes from SM that the applications on the VIF have changed (either a new application starts or an existing one ends). The execution of this phase is similar to that of phase two except that since the VIF is already bound to a PIF, MM needs not to request a PIF from the decision algorithm (LSTM). The operation is depicted in Fig. 8, where additional messages when an application ends are shown in dotted lines.

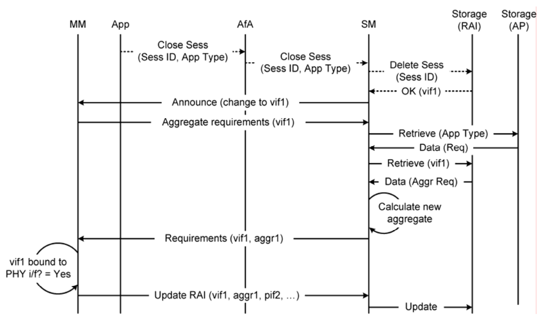

Fig. 8. Change of application on VIF

4) Handover of VIF: In this phase (see Fig. 9), it is assumed that MM receives a trigger from some external module, which suggests that a change of interface may be beneficial. In such a case, it requests SM to retrieve the aggregate requirements of the applications currently using the PIF. SM retrieves the aggregated requirements from the Storage, RAI, along with the associated VIF and sends the data to MM.

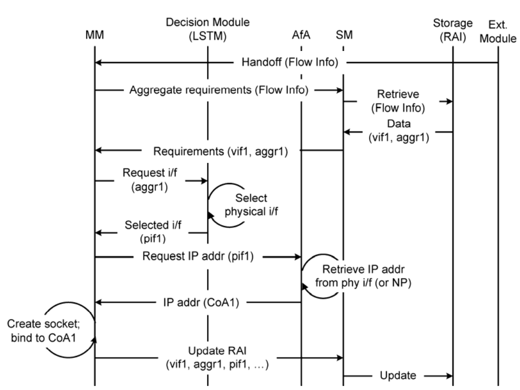

Fig. 9. Handover of VIF

Then, upon request from MM, LSTM selects a suitable PIF, which fulfills the aggregate requirements of the group of applications using the decision algorithm [6]. MM then requests AfA for the IP address (CoA) configured for the selected PIF. AfA retrieves the CoA (either directly from the PIF or from the Storage, NP) and sends it to MM. Finally, MM binds the VIF to the selected PIF i.e. to the CoA, and updates the storage, RAI, with the new PIF, VIF, etc.

IV Conclusions

In this paper, new enhancements to current MIPv4 enabled MTs has been presented. The multi-homed MIPv4 solution supports multimode operation for MTs with no changes to the network. In addition, the multimode MT architecture supporting such MIPv4 operation is elaborately discussed, and operational mechanisms are explained with message flow diagrams. Furthermore, a special function (RAF) is developed in order to represent user preferences and QoS requirements for applications, bound as a group, in a way that is perceivable by the context-aware vertical handover decision algorithm already designed, implemented and simulated [6]. Looking towards the future, it is hoped that all these solutions would surely give a strong platform and valuable aid for further development in the area where providing global mobility to multimode MTs for a diverse array of heterogeneous systems and internet applications is of prime importance.

References

[1] C. Perkins, “IP Mobility Support for IPv4”, *RFC 3344*, August 2002.

[2] W. C. Y. Lee, D. J. Y. Lee, “Mobile IP^2”, in *Proc. of PIMRC’01*, Oct. 2001, vol. 1, pp. A-88-A-92.

[3] K. Chebrolu, R. Rao, “Communication using Multiple Wireless Interfaces”, in *Proc. of WCNC’02*, March 2002, vol. 1, pp. 327-331.

[4] N. Montavont *et al.*, “Mobile IPv6 for Multiple Interfaces (MMI)”, *IETF Internet Draft*, July 2005, draft-montavont-mip6-mmi-02.txt.

[5] K. Kuladinithi *et al.*, “Filters for Mobile IPv6 Bindings (NOMADv6)”, *IETF Internet Draft*, May 2004, draft-nomadv6-mobileip-filters-02.txt.

[6] T. Ahmed *et al.*, “Architecture of a Context-Aware Vertical Handover Decision Model and Its Performance Analysis for GPRS-WiFi Handover”, in *Proc. of ISCC’06*, June 2006, pp. 795-801.

[7] S. Shenker, J. Wroclawski, “General Characterization Parameters for Integrated Service Network Elements”, *RFC 2215*, September 1997.