Policy-Based Multihoming Support in the MULTINET Architecture

Qi Wang, Robert Atkinson, Jorge Espi Aleman, John Dunlop
Mobile Communications Group
Department of Electronic and Electrical Engineering
University of Strathclyde
Glasgow G1 1XW, UK

<qwang, r.atkinson, jorge.espi, j.dunlop>@eee.strath.ac.uk

Abstract

Multihoming can be exploited to provide advanced service delivery for users who can access to multiple networks concurrently. The EU MULTINET architecture envisions flexible and dynamic policy-based multihoming support for nomadic users with a personal area network. This paper presents the design and working implementation of this highly desirable functionality, focusing on the essential policy signalling and flow redistribution procedure. Experimental results derived from the implementation have validated the proposed design.

Introduction

With the expansion of overlay wireless networks and multi-interfaced user devices, simultaneous access to multiple networks has become increasingly feasible and popular for mobile users. Nevertheless, such a multihoming behaviour needs to be cautiously manoeuvred for optimised service delivery. The EU IST FP6 project MULTINET concentrates on policy-based multihoming support architecture that is aware of Quality of Service (QoS) enabled by intelligent network selection algorithms. The primary user scenario is targeted to nomadic workers who roam to a visited site for high-tech machinery maintenance. Such a worker is equipped with a Personal Area Network (PAN) composed of a set of communication devices that are managed by a multihomed Personal Gateway (PG).

Providing mobility to a PAN is supported by the IETF Network Mobility (NEMO) protocol [1]. NEMO is an extension to Mobile IPv6 (MIPv6) [2], which is the de facto standard IPv6 mobility management protocol for single mobile nodes. In MIPv6, a mobile node is known by its long-term Home Address (HoA) obtained from its home network; when visiting foreign networks it obtains and registers a Care-of Address (CoA), bound to the HoA, with its Home Agent (HA) and optionally with its correspondent nodes (CNs) if the Route Optimization is enabled. NEMO extends MIPv6 to handle the mobility of an entire moving network such as a PAN through mobile routers. To enable multihoming, the multiple CoAs allow more than one CoA to be bound with a single HoA [3]. Policy-based flow distribution mechanisms [4][5] are also being investigated in the IETF although further design and analysis are still needed. The MULTINET architecture further extends and integrates these building blocks, together with intelligent network selection and other supporting techniques, for advanced QoS-aware policy-based multihoming support.

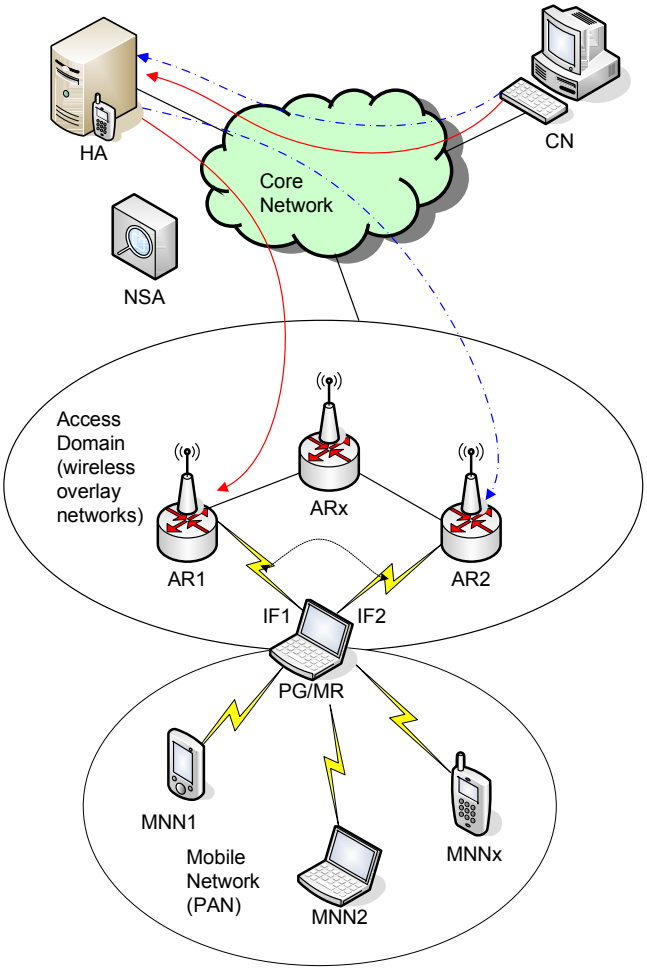
The remainder of the paper is structured as follows. We start with the introduction of the MULTINET architecture. We then present the proposed design of the policy-based multihoming support, followed by the description of an implementation and its experimental validation. Finally, we conclude the paper.

The MULTINET Architecture

The MULTINET architecture, as illustrated in Figure 1, is based upon the NEMO paradigm. The foreign access domain comprises homogeneous or heterogeneous wireless overlay networks whose coverage areas are overlapped to provide simultaneous multiple wireless connections. The intelligent network selection algorithms (NSA) subsystem continuously monitors and analyses the conditions of the multiple networks so that it can determine in real time the policies for optimal distribution of diverse application flows over these networks. A decision is made according to a set of predefined algorithms that take into account the user's preferences, applications' requirements, operator's regulations as well as the network conditions. Detailed design of the NSA subsystem is beyond the scope of this paper as we focus on the subsequent operations once the policies are generated.

The mobile network is a PAN moving as a whole with the mobile user visiting the foreign access domain. The multihoming and mobility functionality of the PAN is managed at the PG, which is a NEMO-enabled Mobile Router (MR) enhanced with additional supporting functions for intelligent network selection. The PG also serves as a gateway to the fixed infrastructure for all the Mobile Network Nodes (MNNs) in the PAN. The PG (and thus the PAN as a whole) is multihomed in that it has multiple network interfaces (e.g., IF1, IF2), each of which has a globally routable IPv6 CoA associated with the corresponding access network.

The MNNs communicate with their CNs located in the fixed infrastructure partition for a number of tasks such as file downloading and real-time video streaming. To comply with the standardised NEMO basic support protocol [1], the bidirectional tunnelling mode is assumed and thus all communications are through the PG's HA, deployed in the home domain. Accordingly, the HA and the PG are responsible to enforce the policies generated by the NSA dynamically so that both the downlink and the uplink application flows can be distributed over the multiple access networks as expected.



- HA: Home Agent
- CN: Correspondent Node
- NSA: intelligent Network Selection Algorithm server
- AR: Access Router
- PG/MR: Personal Gateway/Mobile Router
- MNN: Mobile Network Node
- IF: Interface
- PAN: Personal Area Network

Figure 1 Multihoming support in MULTINET

Definition of Policies

The multihoming functionality is described by the MCoA draft [4], which introduces a binding unique identification (BID) to distinguish different (HoA, CoA) bindings. Each (HoA, CoA) binding is assigned a different BID as shown in Figure 2. Typically, each BID corresponds to an interface of the multihomed PG, and thus a certain BID can be used to represent a specific interface at both the HA and the PG. With the multiple CoAs enabled, the PG is able to utilise the corresponding interfaces simultaneously.

To facilitate QoS aware multihoming, policies are issued by the NSA so that the multiple interfaces can be exploited in an optimal way. In principle, an output policy from the NSA defines a binding of a specific interface (for a given multihomed PG) and a specific application flow, as shown in Figure 3. For instance, BindingN2 indicates that FlowN2 should be transmitted through the interface identified by BIDN. Clearly, a BID can be bound with more than one application flows as many applications can share one network interface.

Furthermore, an application flow can be identified by different parameters present in the application data's IPv6 or higher layer headers [4] such as the Flow Label, the Class of Service, and/or the Security Parameter Index. More commonly, a flow is specified by a flexible combination of a pentuple consisting of five objectives: source address, destination address, source port, destination port, and transport protocol. It is noted that many popular applications can be simply identified by their "well-known" port numbers, e.g., 80/8080 for HTTP and 20 for FTP.

In more generic scenarios, the port number (the source and/or the destination port number) needs to be combined with one or more of the five objectives in a pentuple to identify a fine-grained application flow. An example of a policy encoded in XML is shown in Figure 4. In this policy, the transport protocol is UDP and the destination port is 1234 (the source port is unspecified). This identifies a UDP-based video streaming application whose destination port is 1234. The source and destination addresses indicate the transmission direction as well as the source and the destination. Consequently, the combination of these four objectives specifies a unique UDP-based downlink video streaming flow from the server to the client (an MNN). The home address refers to the PG that manages the MNN and the BID identifies an interface of the multihomed PG. Thus, the policy binds the specified flow to the specified interface. Finally, a specific action should be associated with a policy for a policy enforcement entity to proceed. Essential actions include adding (as in the example) and deleting though additional actions can be defined.

Binding1 (HoA, CoA1, BID1)
Binding2 (HoA, CoA2, BID2)
……
BindingN (HoA, CoAN, BIDN)

Figure 2 Bindings of addresses and interfaces

Binding11 (BID1, Flow11)
Binding12 (BID1, Flow12)
……
Binding21 (BID2, Flow21)
Binding22 (BID2, Flow22)

……
BindingN1 (BIDN, FlowN1)
BindingN2 (BIDN, FlowN2)
……

Figure 3 Bindings of interfaces and flows

```
<?xml version="1.0" encoding="UTF-8" ?>
<flowDistributionPolicy>
 <policy>
   <action>add</action>
   <homeAddr>2001:192:168:106::100</homeAddr>
   <protocol>UDP</protocol>
   <srcAddr>2001:192:168:106:10</srcAddr>
   <dstAddr>2001:192:168:3::100</dstAddr>
   <srcPort></srcPort>
   <dstPort>1234</dstPort>
   <BID>200</BID>
  </policy>
 </flowDistributionPolicy>
```

Figure 4 Policy example

The example shown in Figure 4 is a downlink policy for the HA to enforce. For symmetric flow distribution, a corresponding uplink policy is usually generated for the PG if the networked application is bidirectional. This normally happens in TCP-based applications such as HTTP-based video streaming and FTP file transfer. In these applications, a flow of TCP acknowledgement (ACK) messages is sent from the client to the server. To distribute this ACK flow to the same interface, a symmetric or reverse policy needs to be generated as well. Generally, in the symmetric policy, the destination and source addresses are swapped and so are the destination and source ports.

Policy-Based Flow Distribution

Once a policy is produced at the NSA, it needs to be packed into a message referred to as a trigger since this message is designed to provoke a policy-based flow (re)distribution (also referred to as a flow handoff/handover) over the active interfaces. If more than one policy is generated at the same time, these policies may share the same trigger message for signalling efficiency.

There are two major approaches to achieving the signalling of triggers and their acknowledgements between the policy transmission and reception entities. One approach is to extend the existing MIPv6/NEMO mobility messages so that the policy information can be piggybacked and signalled in both uplink and downlink directions. The other approach is to introduce new dedicated bidirectional messages, preferably standard based for implementation convenience and high interoperability. We have investigated the first approach in a previous work [6]; we explore the second approach in this work.

The Simple Object Access Protocol (SOAP) has been chosen as the bearer of the policy-related signalling for its high extensibility and readability. As a Recommendation of the World Wide Web Consortium (W3C), SOAP is a standard protocol for delivering structured information such as a complex policy based on XML in a decentralised, distributed environment. Typically, HTTP is coupled with SOAP for bidirectional request-response messaging over a TCP connection.

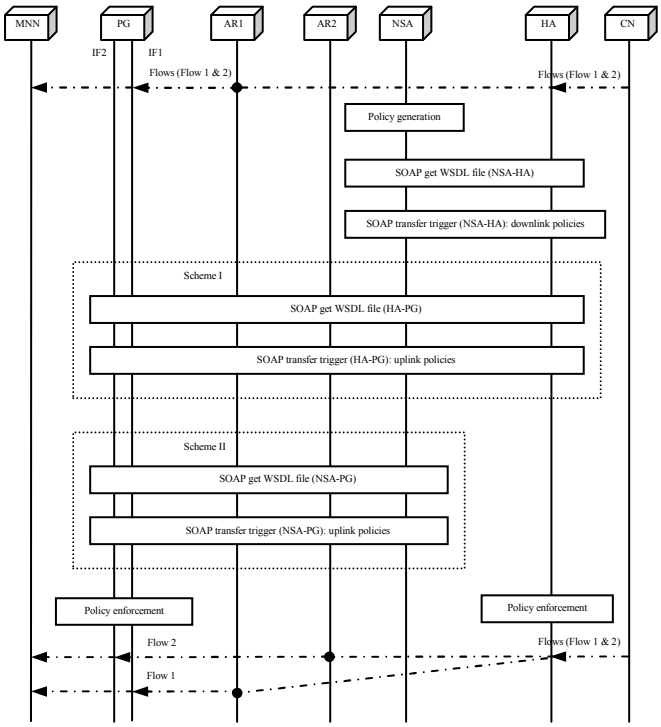


Figure 5 Policy signalling schemes for policy-based flow distribution

Two SOAP-based signalling schemes have been designed as depicted in Figure 5. In Scheme I, the NSA only generates the downlink policies and then sends the downlink trigger alone to the HA. The HA will then generate the symmetric policies and sends the resultant uplink trigger to the PG. In Scheme II, it is the NSA that generates both the downlink and the uplink policies and sends the downlink and uplink trigger to the HA and PG, respectively. One of the advantages in Scheme I is that the HA is given an opportunity to double check if the involved interfaces are still available according to the binding cache, which is usually exclusively available to the HA. If one of the involved interfaces becomes unavailable, the HA will inform the NSA instead of generating and sending the uplink trigger to the PG. This is especially useful in highly mobile scenarios. On the other hand, Scheme II is superior in that a single policy generation source is needed without inserting further overhead on the HA. The service provider may select a scheme from these two choices.

Web services description language (WSDL) files are located in the HA and the PG which define the web services such as the policy schema, messages and operations. The sender of a trigger may need to load the WSDL file at the receiver before it sends the trigger. To accelerate the process, WSDL files are cached locally. Thus, loading the WSDL file is only incurred at the first time before a trigger is ever transferred or when the local WSDL caches expire (after 24 hours by default). The detailed signalling sequences for retrieving a WSDL file and for sending a trigger between two entities, e.g., the NSA and the HA, are illustrated in Figure 6 and Figure 7, respectively. The TCP connection setup and teardown are shown in Steps (1) to (3) and Steps (8) to (10), respectively.

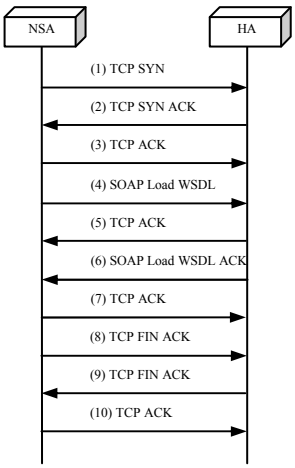


Figure 6 SOAP signalling to retrieve the WSDL file (NSA-HA)

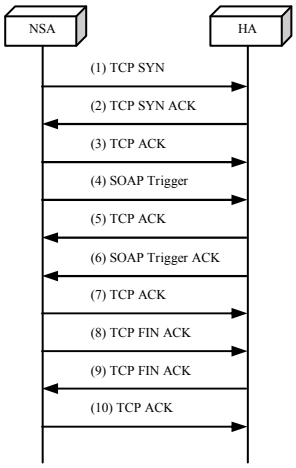


Figure 7 SOAP signalling to transfer a trigger/policy (NSA-HA)

Upon receiving and parsing the trigger (downlink or uplink respectively), the HA and the PG enforce the enclosed policies by manipulating their ip6tables; more details can be found in [7]. If the policies are successfully enforced, the flows are then (re)distributed as desired. In the case shown in Figure 5, Flow 1 and Flow 2 contained in an ongoing downlink data stream through IF1 alone are distributed in two access networks corresponding to IF1 and IF2, respectively.

Implementation and Validation

To verify the proposed policy-based multihoming architecture, we have implemented the proposed designs on a local IPv6 wireless testbed, which resembles Figure 1. A set of Linux PCs is configured to act as the NSA, the CN, the HA and the PG equipped with two Wi-Fi interfaces IF1 and IF2. The MNN is a Windows XP PC in the mobile network whose multihoming and mobility proxy is the PG. The NSA, simplified as a network trigger/policy generator, is collocated with the CN that is a video streaming server and a FTP server for the MNN. VLC and Proftpd applications are used for video streaming and FTP, representing typical real-time and non-real-time applications, respectively. A couple of 802.11b/g ARs provide the multihomed PG with two wireless connections (and thus two separate routes between the HA and the PG) whose data rates were set to be 11 Mbps.

The SOAP-based policy signalling schemes were implemented with PHP5 [8], which has built-in C-based SOAP support. The flow-distribution execution functionality was built upon the NEMO implementation NEPL [9] with integrated MCoA support. IPv6 stateless host auto-configuration was achieved through the radvd module. The multihomed PG automatically configures a CoA for each of its interfaces and registers the CoAs with the HA via the MCoA support.

We have performed numerous experiments to verify the effectiveness of the proposed policy-based multihoming support. In the following, we present a case study of flow redistributions (handoffs) of two different applications: FTP file downloading and UDP-based video streaming. As illustrated in Figure 8, the X Axis represents the time sequence of the experiment whilst the Y Axis shows the traffic volume of the application flows.

At time T0, a RTP/UDP-based video streaming has been established and is being transmitted from the CN towards the MNN via the PG. By default, all traffic travels through the IF2 interface of the PG and the corresponding interface of the HA.

At time T1, the NSA issues Trigger 1, which contains one policy to be enforced at the HA. The policy indicates that the ongoing video streaming should be handed off from IF2 to IF1. The symmetric policy for the PG is optional as the streaming is a one-way traffic (no uplink traffic for acknowledgement or reverse streaming). After the policy is enforced at the HA, the streaming flow is handed over from IF2 to IF1.

This flow redistribution decision could be made according to the output of the intelligent network selection algorithm for different reasons such as load sharing, fault tolerance or user/application preferences. For instance, the NSA had detected that the route via IF1 was underutilised or the route via IF2 was overloaded (or temporarily going down). It could also result form an establishment request of a new application session (e.g., the subsequent FTP downloading), which has the priority to use the IF2 route and demands other applications to use alternative routes if possible.

At time T2, a FTP file transfer was initiated by the MNN to download a large file from the CN. Again, by default the FTP flow (from the CN to the MNN) and the corresponding TCP ACK flow (from the MNN to the CN) were transmitted through IF2.

At time T3, the NSA issues Trigger 2, comprising two policies. One policy is to switch the FTP flow (downlink traffic) from IF2 to IF1. A symmetric policy was also generated at the HA for the PG to redirect the TCP ACKs (uplink traffic) together. Meanwhile, the other policy demands that the video streaming be handed off from IF1 to IF2. The symmetric policy to the latter one is optional.

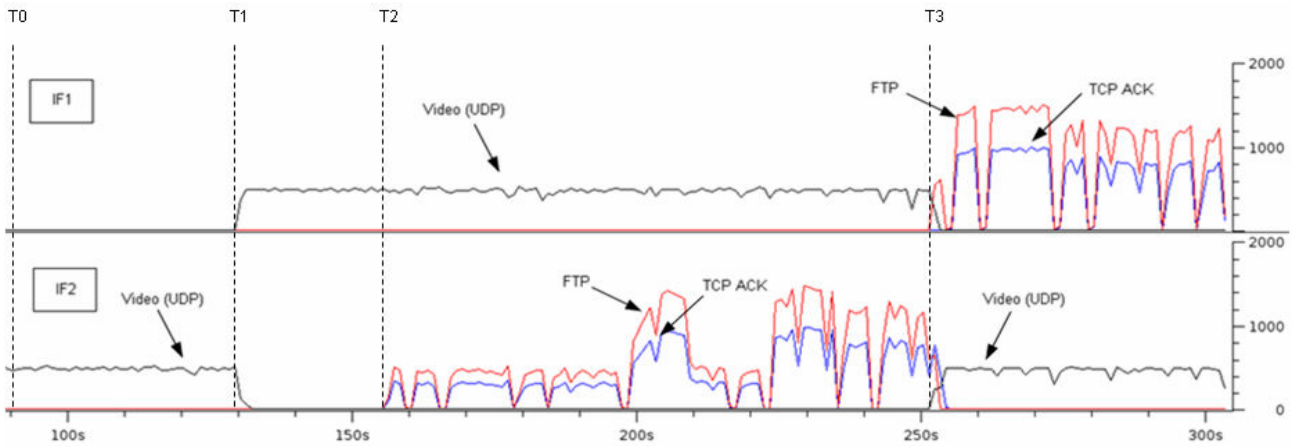


Figure 8 Policy-based flow redistribution

Consequently, the three flows were handed over to the targeted interfaces, respectively.

Such experiments were repeated with random flow handoffs of the applications between the interfaces. When the flow handoffs occurred in these experiments, no noticeable disruptions were perceived by a number of observers viewing the video streaming at the MNN. Meanwhile, the file downloading was also successful. Therefore, it seems that the policy-based flow redistributions do not significantly degrade the subjective QoS of real-time applications or affect the continuous delivery of non-real-time applications.

Conclusion

We have presented the design, implementation and validation of policy-based multihoming support in the MULTINET architecture.

The MULTINET architecture is established upon the IETF NEMO paradigm, and it exploits the multiple care-of addresses extension to enable basic multihoming for nomadic users who have a personal area network. For policy-based multihoming support, policies are dynamically generated by intelligent network selection algorithms (NSA). A policy defines a distribution of a selected application flow over a specific network interface and the corresponding access network. With a set of dynamic policies enforced at the HA and the multihomed PG for downlink and uplink traffic respectively, flows of diverse applications can be distributed in an optimised way over the multiple access networks. Policy signalling is a key enabler to the policy-based multihoming support and SOAP has been proposed to signal the policies, which are encoded in XML for high readability and extensibility. Two schemes have been designed depending on which entity (the NSA or the HA) is responsible to generate and transfer the symmetric policy (or policies) to the PG. Each scheme has its own pros and cons. The choice is the service provider's.

The proposed architecture has been implemented and verified on a testbed. The experimental results show that the desired policy-based multihoming support appears effective for both typical real-time and non-real-time applications such as video streaming and FTP downloading.

Acknowledgments

This work has been funded by the EU IST FP6 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.

References

[1] Devarapalli, V., Wakikawa, R., Petrescu, A., and Thubert, P., "Network mobility (NEMO) Basic Support Protocol," *IETF RFC 3963*, Jan. 2005.

[2] Johnson, D. B., Perkins, C., and Arkko, J., "Mobility Support in IPv6," *IETF RFC 3775*, Jun. 2004.

[3] Wakikawa, R., Ernst, T., and Nagami, K., "Multiple Care-of Addresses Registration," *IETF Internet Draft*, <draft-ietf-monami6-multiplecoa-03.txt>, work in progress, Jul. 2007.

[4] Soliman, H., Montavont, N., Fikouras, N. and Kuladinithi, K., "Flow Bindings in Mobile IPv6 and NEMO Basic Support," *IETF Internet Draft*, <draft-soliman-monami6-flow-binding-04.txt>, work in progress, Feb. 2007.

[5] Mitsuya, K., Tasaka, K. and Wakikawa, R., "A Policy Data Set for Flow Distribution," *IETF Internet Draft*, <draft-mitsuya-monami6-flow-distribution-04.txt>, work in progress, Aug. 2007.

[6] Wang, Q., Atkinson, R., Cromar, C., and Dunlop, J., "Hybrid User- and Network-Initiated Flow Handoff Support for Multihomed Mobile Hosts," *Proc. 65th IEEE Vehicular Technology Conf. (IEEE VTC2007-Spring)*, Dublin, Ireland, Apr. 2007, pp. 748-752.

[7] Wang, Q., Hof, T., Filali, F., Atkinson, R., Dunlop, J., Robert, E., and Aginako, L., "QoS-Aware Network-Controlled Architecture to Distribute Application Flows over Multiple Network Interfaces," *(Springer) Wireless Personal Communications*, accepted for publication (in press).

[8] Hypertext Preprocessor (PHP) SOAP functions, http://uk2.php.net/soap.

[9] NEMO Implementation for Linux (NEPL), http://www.nautilus6.org/nemo/.