P1

Publication P1

Jukka Ylitalo, Tony Jokikyyny, Tero Kauppinen, Antti J. Tuominen, and Jaakko Laine, "Dynamic Network Interface Selection in Multihomed Mobile Hosts", In Proc. of the Thirty-Sixth Hawai'i International conference on System Sciences (HICSS-36), published on cd-rom, abstracts p.315, Big Island, Hawai'i, US, January 6-9, 2003, ISBN 0-7695-1874-5, Publisher: IEEE Computer Society (Ed: Ralph H. Jr. Sprague).

Dynamic Network Interface Selection in Multihomed Mobile Hosts

Jukka Ylitalo, Tony Jokikyyny, Tero Kauppinen, Antti J. Tuominen, Jaakko Laine

***Abstract*— Current mobile devices are often equipped with several network interfaces, which may be of different access technologies, both wireless and cellular. Different requirements of different applications can result in a different preference of the interface that should be used. Network connections should be placed in the best possible interface based on these requirements. During communication, changes in the availability or characteristics of an access network behind an interface may result in a situation where already established connections should to be moved from one interface to another. For this purpose, a variety of mobility management protocols supporting handoffs between interfaces have been proposed. Some of these protocols move all traffic from one interface to another at once, while some protocols allow simultaneous communication over different interfaces. However, the current solutions do not propose any means for the user or application to be able to dynamically influence the interface selection during the operation of a mobile device.**

In this paper, we will present an interface selection mechanism for multihomed mobile hosts. The mechanism allows for dynamic decision-making during the operation of a mobile device. In our solution, the local routing is controlled by user-defined rules defining which interface to be used for a certain traffic flow. The actual decision is based on the adaptation of these rules into availability and characteristics of the interfaces and access networks at any given time.

I. Introduction

As different wireless network technologies such as 3G cellular networks are being deployed at an increasing rate, interworking of these various technologies has become an important issue. Already, mobile Internet hosts are often equipped with several network interfaces or are at least able to connect to such interfaces. These interfaces may use different access technologies such as Bluetooth, WLAN and 3G cellular. For this purpose, a few mobile host *multihoming* protocols supporting handoffs between interfaces have been proposed. The most advanced protocols are able to move single traffic flows independently of each other.

However, the current solutions do not propose any means for the user to be able to dynamically influence the interface selection during operation. Different access technologies and access operators offer several types of price and quality. Therefore, a mobile user must be able to affect on the interface selection so that the most suitable of the available interfaces is used. Changes in the availability or characteristics of an access network may result in a situation, where the user wants to move already established traffic flows from one interface to another.

Multihoming, especially in a mobile environment, requires management and control of connections to become functional. Input for this control comes from several sources, e.g., user, application, access network and interface drivers. When a host has multiple paths to send packets to its peers, it must somehow make a decision of which path(s) to use for which connection(s). More specifically, the host needs a policy to select the source IP address and the outgoing interface.

When multiple access networks are available for a multihomed mobile host, it can make handoffs between these different access networks using methods described in other solutions, e.g., Mobile IPv4 [1] and Mobile IPv6 [2]. According to some proposals it is also possible to transfer single connections separately from each other thus communicating through several interfaces in parallel. We call this functionality *simultaneous multiaccess* [3]. In a multi-operator environment it is possible that the access networks are managed by different operators. It is also possible that several different bearers are available to a link layer interface at the same time.

It is anticipated that some applications and services will be able to adapt to changing access situations. Information on network characteristics should also be available to the applications, so that they can possibly adapt to the continuously changing environment. We consider it a natural requirement that the applications running on the user equipment should be network access independent in the sense that the applications themselves are unaware of the underlying access technology and have an interface toward the IP layer only through an IP based application programming interface (API).

In this paper, we present a policy-based handoff mechanism for mobile multihomed hosts. It is implemented within a publicly available protocol stack allowing simultaneous use of several network interfaces. The handoff decisions in our implementation are based on explicit user defined rules, i.e., policies. The mechanism allows for dynamic decision-making during the operation. Events that might require handoffs include change of the topological location of an interface, change in application requirements or change in the availability of access. In general, our mechanism allows implementation of applications adapting to the quality of available access. The policy entries are general enough to be bound to one certain data flow or a group of data flows.

We focus on interface selection policy and mechanism in multihomed mobile hosts (i.e. end-hosts), which is a relatively new area. We have based our implementation on Mobile IPv6

Mr. Ylitalo, Mr. Jokikyyny and Mr. Kauppinen are with Ericsson Research in Finland. E-mail: {jukka.ylitalo, tony.jokikyyny, tero.kauppinen}@ericsson.fi. Address: Oy L M Ericsson Ab, Telecom R&D, FIN-02420 Jorvas, Finland. Phone: +358 9 2991.

Mr. Tuominen and Mr. Laine are with the Telecommunications Software and Multimedia Laboratory at Helsinki University of Technology. E-mail: ajtuomin@tml.hut.fi, jola@niksula.hut.fi. Address: Helsinki University of Technology, P.O. Box 9201, FIN-02015 HUT, Finland. Phone: +358 9 4511.

(MIPv6) with multihoming support. The solution is independent of the underlying network technologies and the principles presented in this paper can also be used with other network layer multihoming protocols.

II. Multihomed mobility

A network that has an edge-router with several interfaces each of them connected to a different router, usually of different ISPs, implements *site multihoming* [4]. In a basic site multihoming architecture the network topology does not change very often. This makes it reasonably easy to implement static routing policies within a router compared to routing policies within a multihomed mobile host.

The interfaces in a multihomed mobile host dynamically change their point of attachment to the network. Furthermore, when a mobile host is moving the available access networks change frequently and might be offering different kinds of network characteristics for the connections. Therefore, multihomed mobile hosts require additional local routing (i.e., interface selection) mechanisms to manage their connections. The scope of this paper is in mobile hosts, whereas mobile routers are beyond it.

A. Definition

The basic requirement for a mobile host is the ability to change its point of attachment to a network without loosing its ability to communicate [1], [2]. If the mobile host is multihomed we can separate the local mobility management into two parts: *horizontal* and *vertical* mobility. A horizontal handoff is made between different access routers within the same link layer technology - typically due to the geographical movement of a host. A vertical handoff, in turn, means that we hand-off from one interface to another. Typically the access router and link layer technology changes at the same time.

It is possible that horizontal and vertical handoffs are addressed to single connections that move independently of each other - between different interfaces and access routers. Actually, when we bind a connection to a group of IP addresses which can be updated dynamically we provide simultaneous multiaccess (SIMA) functionality.

One of the main targets in SIMA is to take advantage of different concurrently available access networks when sending packets from one host to another. For example, high bandwidth WLAN networks can provide small “hot-spot” areas within a high coverage UMTS network as shown in Fig. 1. Because the applications, or even the different flows originating from the same application, running in a mobile host might have very different requirements for the network characteristics there might well be a need to use several access networks simultaneously.

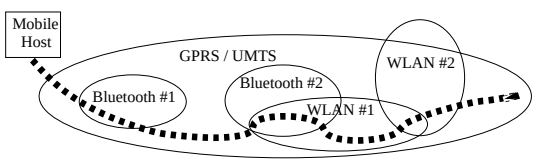

Fig. 1. Different access technology networks complementing each other.

It is apparent that all these networks should be accessible by the mobile host without breaking any active connections. The best possible network could be used in any situation based on the requirements of a connection. The mobile host bases its access network selection decision on the interface selection mechanism and policy. The rules for interface selection can be integrated into a local routing mechanism or they can be defined explicitly using policies. In the later sections we will show how a user can explicitly define policies controlling the local routing of connections.

B. Host mobility management

Mobile IP [1] in general, and more specifically Mobile IPv6 [2], solves the problem of mobility of a Mobile Node (MN). This is done by managing a mapping between the changing *Care-of Address* of the MN, and the *Home Address*, a permanently or semi-permanently assigned address of the MN in its home network. The applications and protocol layers above the IP layer in the MN and Correspondent Nodes (CNs), i.e., the peers which the MN is communicating with, use the Home Address when they need to use the services of the IP layer. Mobile IPv6 then provides the upper layer transparency concerning the current CoA, and thus the topological location of the MN.

A *Home Agent*, located in MNs home network maintains the mapping, or *binding*, between the (primary) CoA and the Home Address of each MN. When a valid binding for a MN exists, the HA will capture all the packets sent to the MN’s Home Address and forward them by tunneling to the CoA. MN notifies HA of binding changes with a *Binding Update* message.

MIPv6 additionally provides route optimization. When route optimization is utilized, CNs no longer need to send the IP packets destined to the MN through the HA. Instead, the MN will use a BU to inform CNs about its current CoA, so that CNs will be able to send packets directly to MN’s current CoA. Fig. 2 illustrates basic MIPv6 signaling (excluding Return Routability tests) and data packet flows with triangular and optimized routing.

C. Multihoming protocols

There are several proposals related to multihoming. In this section, we focus on proposals that are implemented at the network and transport layers because those are transparent to applications.

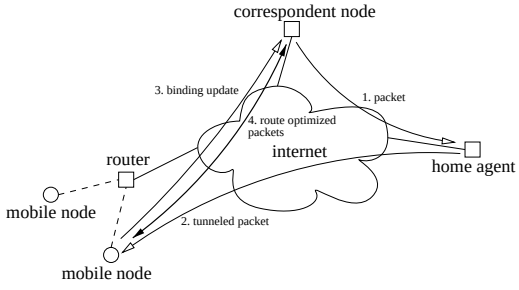

Fig. 2. Basic Mobile IPv6 Operation.

1) LIN6: LIN6 [5] provides a solution for mobility and multihoming using LIN6 *generalized IDs*. The basic idea is that each host has a 64 bit globally unique identifier, called LIN6 ID, which is presented in the IPv6 interface identifier portion of an IP address used by a host. In addition to this, LIN6 reserves a special IPv6 prefix, called LIN6 prefix, which is not routable. A host can be uniquely named by prefixing its LIN6 ID with the LIN6 prefix, resulting in what is called LIN6 Generalized ID (GI). These generalized IDs are then stored into the DNS, together with the address of a Mapping Agent. Since the GIs are globally unique and permanent, the communicating hosts use them as endpoint names. The Mapping Agent is queried for the mobile host's current addresses. The host then dynamically translates the prefixes on outgoing and incoming packets, making it possible to use GIs on sockets and real addresses in routing. LIN6 also supports multihoming through allowing a single GI to be associated with several real addresses.

LIN6 does not provide any means to explicitly select interfaces for different connection in a multihomed host.

2) MHTP: Multi Homing Transport Protocol (MHTP) [6] has been proposed for IPv6 network layer multihoming. The MHTP is targeted for site multihoming, being a feature of routers. It is strongly based on BGP4+ routing information. It can be described as a semi-symmetric, end-to-end, NAT protocol. The main idea behind MHTP is that multihomed traffic is transformed into single-homed traffic at a router close to the source and transformed back into multihomed traffic at the last router being the MHTP endpoint - a multihomed site that have been allocated an MHTP prefix. This prefix is a /48 block of multihomed addresses.

Implementing the MHTP in a mobile host would greatly increase the load of the MHTP endpoints because of the amount of signaling. Therefore it is not feasible for host multihoming. It seems that MHTP can only be a mid-term solution for site multihoming.

3) Multihomed TCP: Multihomed TCP [7], also called as Extended Transport Control Protocol (ETCP), extends the TCP protocol. The ETCP makes it possible to use a set of IP addresses in both endpoints. Addresses can be updated dynamically during communication, enabling multihomed mobility. The draft also presents an address selection mechanism based on time-stamps. Instead of IP addresses, the ETCP uses a separate 32-bit connection identifier, named as Protocol Control Block Identifier (PCBI) to identify connections. Hosts change their PCBIs during an ETCP handshake.

The ETCP does not solve fast movement or double jump problems but they are identified in the draft [7]. The protocol does not take stance on how to avoid PCBI collisions or how the current socket API would be affected. ECTP does not allow a user to define own local routing policies.

4) Stream Control Transmission Protocol: Stream Control Transmission Protocol (SCTP) [8] is a reliable transport protocol operating on top of the network layer where both of the endpoints may be presented by multiple IP addresses. The motivation in SCTP to use multihoming is the potentially better survivability of the connection in the presence of network failures. However, the connection management policy in SCTP is restricted, and multihoming is mainly used for redundancy purposes.

A host has one primary address and may have zero or more alternative addresses. The primary address is used by peer hosts as the destination address for all packets in normal data transmission. Alternate addresses are used for retransmitted packets to improve the probability of reaching the remote endpoint. When transmission to primary address fails several times packets are transmitted to alternative addresses until the protocol becomes confirmed that primary address is reachable.

SCTP endpoints exchange a list of addresses during the initiation of a connection association. Every endpoint must be able to send messages through any interfaces that are bound with local address set and receive messages from the address set associated with the remote endpoint.

SCTP specification describes a socket API that can be used to implement interface selection policy. However, it is quite inefficient to implement such kind of functionality totally on the application level.

5) Flow-based mobility: Per-flow movement in MIPv6 [9] proposes an extension to Mobile IPv6 signalling where the binding updates can also contain flow IDs, which are to identify the packets belonging to a specific traffic flow. Different flows can be assigned at the application level based on needs, thus making it possible to have different flows traverse through different interfaces.

The process of flow ID assignment is out of scope of the draft, but the interface selection mechanism presented in this paper could well be deployed for this purpose.

6) Homeless Mobile IPv6: In Homeless Mobile IPv6 [10] architecture the connections are not bound to interfaces represented by IP addresses, but to hosts themselves which are represented by sets of IP addresses. Every host has a cache consisting of local and foreign host cache entries. A local host cache entry contains a set of local IP addresses. Any of them can be used as a source address for outgoing IP packets. Correspondingly, there is a foreign host cache entry for each of the peer hosts.

Homeless Mobile IPv6 does not define how to implement an interface selection policy. Thus, it is difficult to explicitly define static rules for interface selection because the connection identifiers, i.e. IP addresses, change frequently when a host moves. This same problem exists in SCTP.

7) Host Identity Payload and Protocol: The Host Identity Payload and Protocol (HIP) [11] architecture describes a new name space called the Host Identity (HI), which completes the IP and DNS name spaces. The use of HI requires an own protocol layer called the Host Layer Protocol (HLP) located between the IP and transport layers. Cryptographically generated HI is used to identify a connection, while IP addresses are only used for routing information. This kind of namespace separation allows easier mobility and multihoming implementation and management because the IP addresses can be changed without affecting the connection identification.

The HIP architecture seems to be a good alternative for Mobile IPv6 to implement interface selection policy effectively for mobile multihomed hosts.

8) MIPL Project: Mobile IPv6 for Linux (MIPL) [12] is an open source implementation of Mobile IPv6 developed at the Helsinki University of Technology. In addition to standard

Mobile IPv6, MIPL also implements simultaneous multiaccess. Also the interface selection mechanism presented in this paper has been implemented in MIPL.

In addition to conventional Mobile IPv6 handoff (horizontal handoff), MIPL supports seamless IP traffic movement between different access network interfaces connected in a mobile node (vertical handoff). Available interfaces are found by using the standard IPv6 address auto-configuration [13]. In MIPL, no packets are lost in vertical handoffs, if both old and new interface are available during the hand-off. The implementation therefore supports session continuity and real-time performance while users are moving between access networks.

The MIPL solution does not require any modification to applications using IPv6 nor any additional nodes in the network. The solution is also independent of access technology. Fig. 3 presents a summary of multihoming protocols.

Protocol	End−point identifier	Home−agent functionality	Interface selection policy	IP version/ Layer
LIN6	GI	Mapping agent	implicit	IPv6 / L3
MHTP	IP address with MHTP prefix	MHTP router	implicit	IPv6 / L3
ETCP	PCBI	(none)	implicit	IPv4 / L4
SCTP	Set of IP addresses	(none)	implicit/ API	IPv4/v6/L4
Per−flow MIPV6	Haddr. + flow id	HA	not defined	IPv6 / L3
Homeless MIPv6	Set of IP addresses	(none)	implicit	IPv6 / L3
HIP	HI	Rendevouz server	not defined	IPv4/v6 L3−L4
MIPL	Haddr.	HA	Our solution	IPv6 / L3

Fig. 3. Summary table of multihoming protocols.

D. Connection Identification and Namespaces

In the current Internet architecture IP addresses have several meanings. They are used for location, routing, identity and identifier information. A host has always an identity that is identified by one or more identifiers. The concept is same as with a person (identity) and his passport (identifier). Currently Fully Qualified Domain Name (FQDN) and IP address are examples of host identifiers.

Originally IP addresses were good identifiers for hosts, because hosts were static and they could be identified on account of their topological location. However, the introduction of mobility and multihoming have changed the situation. Unfortunately, a transport layer connection is traditionally identified with IP addresses in both endpoints with the following association 5-tuple:

{protocol, local-addr, local-port, foreign-addr, foreign-port}

Implementation of multihomed mobility becomes easier if we separate the connection identification from routing information. In other words, there is a need for separate host identifier in addition to topologically defined IP addresses. A host identifier can be associated with one or several IP addresses. The connection properties can be expressed with the following three associations:

1) {protocol, {local-identifier}*, local-port, {foreign-identifier}*, foreign-port}
2) {local-identifier, {IP-address}*}
3) {foreign-identifier, {IP-address}*}

Association 1 identifies the connection while associations 2 and 3 bind the connection to a group of IP addresses. The connection identifiers are dynamically bound to a set of IP addresses. Local and foreign connection identifiers are typically IP addresses, like home addresses in the Mobile IPv6 architecture. Homeless Mobile IPv6 and SCTP use a set of IP addresses as connection identifiers. The Host identity Payload architecture uses a Host Identity Tag (HIT) instead of an IP address to identify a connection.

This short analysis presents the name space confusion that can be found in the current TCP/IP protocol stack. Chiappa [14], Bellowin [15] and Moskowitz [11] discuss the topic. The same name space confusion also exists in the interface selection concept. The problem is how to bind a set of routing rules to a pool of connections when both the rules and IP addresses are updated dynamically. We need an identifier which is dynamically bound to routing information and to policy rules that define the behavior of a pool of connections. An identifier does not only identify a host, but it also identifies a specific type of connections. The relationship between policies and IP addresses is discussed in detail in Section IV-B.

III. Interface Selection

Interface selection is a term that can be used for local routing of packets through local interfaces in a multihomed host. This routing can be based on a connection association (IP address, port number, protocol) and other (e.g. QoS) information. In addition, interface selection indirectly defines destination routers to be used for the outgoing packets.

In a multihomed host, it must be possible to configure and control the operation of multiple accesses according to the - dynamically changing - needs of applications and users. MIPL implements an interface selection mechanism capable of this.

The interface selection system is based on five basic components, which are very similar to those presented in the KeyNote Trust-Management System [16]:

- *Entities* define actions. An entity may be a user, peer node or 3rd party, e.g., operator.
- *Action* is an operation that is defined by an entity and is controlled by the system. Actions specify interfaces to be used for connections on account of entity's requirements. Actions can be presented as conditional statements.
- *Policy* governs the actions of one entity. Only one action can take place at a time in a policy. A policy set contains several policies possible defined by different entities. Policy definition language defines the priority between policies and actions.
- *Credentials* are used to authorize actions that are defined by different entities.
- *Mechanism* evaluates actions against connection related information and decides which interface is to be used with a specific connection.

A. Information

Normally routing decisions and interface selection are based entirely on IP/network layer information. In host multihoming this is inadequate, since we need to take multiple factors into account when selecting interfaces for outgoing traffic. As a rule, these factors lie outside the IP layer, thus forcing us to break the level hierarchy in order to provide the necessary interface selection functionality.

1) Link Layer Information: In wireless networks signal quality and related metrics play an important role when deciding which interface to use.

Though other universal factors must be taken into account, link quality is imperative, since it dictates what quality of service demands and policies can be fulfilled and how much of the theoretical bandwidth is actually available. Moreover, if the link quality is poor, a user with a PDA might not want to use it at all because of increased power consumption.

To be able to make as smooth and intelligent handoffs as possible, link quality must be constantly monitored and the information must be made available for the network layer and user applications in a form that suits them best. They may use it in combination with other information to make the best possible proactive routing decisions and interface selections. Most of the currently available wireless network drivers support information gathering and although the information and its presentation are far from uniform, the most important metrics are widely available in some form or another.

Interface selection algorithm should take into account all available information and at the same time minimize resource consumption and make decisions with as light computation as possible. Obviously all these requirements cannot be met at the same time and a compromise is needed. Experiments show that a simple algorithm based on a few snapshots of link layer metrics and their average works well enough and avoids oscillation between interfaces [17]. Algorithm can be adjusted by changing snapshot frequency and interval to meet different kind of requirements depending on traffic type and node's movement.

2) IP layer Information: Several attributes can be retrieved from the IPv6 header without looking into the data, e.g., source address and destination address [18]. Some attributes can also be retrieved from IPv6 extension headers. Because only transport protocols, like TCP and UDP, can be identified directly from the IP header, the higher level protocols, like HTTP, can be mapped to port numbers as these are visible in the IPv6 header. In the case of HTTP the port number could be, e.g., 80 and/or 8080, based on port number assignments by IANA [19].

3) Network Originated Information: A service provider may disseminate information about cost, bandwidth and availability of the Internet access. For example, an ISP might offer Internet access through WLAN and Bluetooth within an area. In addition to advertising the default gateway by means of Router Advertisements [20], access routers could also send cost and bandwidth information. The mobile user could then have preferences for connections, like *maximize bandwidth* or *minimize price*, and the host would select the appropriate interface satisfying these preferences.

Disseminating information from the network may be implemented using a new protocol (e.g., CAR [21]) just for this purpose, or Router Advertisements could be extended to carry the information. However, there is a security problem involved which is discussed in Section V

4) Information Originated From Users and Applications: Some applications may require certain characteristics from the connections. An application should be able to adapt into changing network environment and set it's own preferences for connections. This can achieved by extending the current socket API. The API must allow the delivery of user preferences to the interface selection mechanism. The preferences can be presented as policies which are presented in the next section.

B. Policies

The information described in the previous subsection includes connection type, availability of network interfaces and various characteristics of networks behind those interfaces. Mobile users, peer nodes, applications and 3rd parties may define preferences and requirements regarding the use of interfaces and access networks. The interface selection decisions are based on these preferences and requirements when they are evaluated against gathered information about transport characteristics.

The operation of the interface selection mechanism must be continuous as the information may change at any point of time, e.g., a network interface may become unavailable. Therefore, there have to be a policy database and mechanism to hold and maintain rules for interface selection. Policies provide different network entities a possibility to control the placement of mobile host's traffic flows into different network interfaces.

Interface selection policies describe the preference of different network interfaces in various situations. On account of the policies, the local routing mechanism routes outgoing IP packets into available interfaces. The most preferred available interface is always used, and if it becomes unavailable, e.g, when a user moves out of a wireless network coverage area, the connections will be moved to the next preferred interface.

1) Action in Policy: In this paper we analyze the interface selection management from a single policy and mobile user point of view. Distributed policy management is discussed in Section VII.

The conditional clause within an action consists of attribute-value pairs that are evaluated against connection association information. If there is a match then the interface candidates are searched in the preferred order.

Any action, except the default action, must include at least the following information: a) an unique ActionID, b) a parameter indicating whether or not the action must be forced, c) attribute-value pair(s) containing connection association data d) list of interface types or characteristics in preferred order.

If an action is forced, all traffic that is matching to the attribute-value pairs of the action, may not use any other interfaces than specified in that specific action. Likewise, if an action is not forced and all the interfaces listed in that action are unavailable, the next preferred action can be followed. An example of a pseudo action statement:

if (address=A and port=B) then use interface (case WLAN:if (price < P) then use WLAN; case Next action: force=FALSE;)

The presented action takes place if the address and port number match, after which a WLAN interface with price lower than P is selected if such is available. In other cases, the next action in the policy is examined. A policy description language should allow for complex action definitions.

There should always be a default action in the policy that is used if no other matching actions can be found. The default action defines a general preference for all interfaces attached to a mobile node.

The actions in the policy database are dynamically evaluated against the constantly changing information. The user must be able to update policy on account of this information. This might happen either manually or automatically depending on the implementation. For example, if the price of some interface changes the user might get a notification on this and thereafter might want to reconfigure the policy database, e.g., change the order of preference of the interfaces within an action. This might also happen automatically if the user has defined some additional conditionals in the action, e.g.:

if price<5 use WLAN else use GPRS.

The user does not necessarily have to define the interfaces in actions directly. Instead, the user may prefer to define some simple rules which are further interpreted by the interface selection mechanism and used to automatically generate an order of the interfaces. Such a user defined rule could simply be, e.g., *Always use the cheapest interface.* In the presence of this rule alone, the mechanism would only create a default action listing with the available interfaces in an order of price.

2) Priority of Actions: The actions in a policy must have some priority defining the order in which the actions are searched and matched. Fig. 4 shows an example of a policy where the priority is implicitly included in the structure, the first action in the list having the highest priority. The default action must always have the lowest priority.

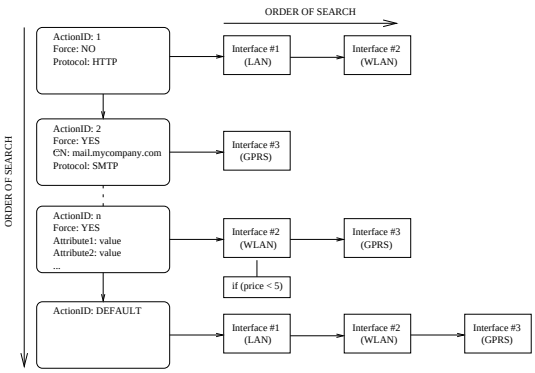

Fig. 4. An example of a policy.

Fig. 5 describes a set of priority classes for the actions which we have used in our implementation. Each action is put into one of these classes based on the connection association attribute-value pair(s). The ordering of the actions within a class can be random.

3) Distributed Policies: In future, it is very likely that 3rd parties, like the network operators, want to have some influence on how the different accesses they provide are used. In practice, a mobile multihomed host may obtain policies from several different operators. The policy mechanism must implement an algorithm to avoid conflicting local and remote policies. It is up to the policy definition language how the relationships between policies are handled. In most cases, access network operator's policies should override any conflicting user defined policies related to that specific access network. In addition, the owner of a mobile node must be able to define the priority between different operators' policies. It is obvious that operators do not want their policies to be tampered within a mobile host. Therefore, distributed policies should implement a strong cryptography and/or be stored in a tamper-free device, e.g., a smart card.

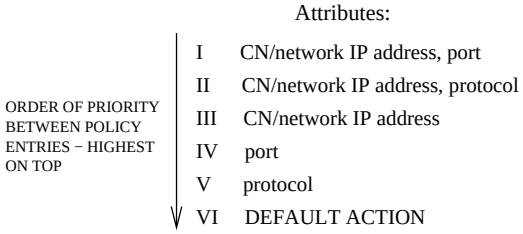

Fig. 5. Priority classes for policy entries based on attributes included.

A mobile node could also send policies to peer nodes or receive policies from them. This possibility would make destination interface selection possible. A policy received from a multihomed peer host would allow the receiver to determine correct the destination interface for a specific connection. This feature could be bound to personal firewalling. After this, a host might dynamically update its personal firewalling rules and send related policies to its peers. Peers could use the received policy to define a destination address for the specific multihomed host.

The presented remote use cases place additional requirements for the policy management system, most importantly: (1) secure policy exchange between entities, (2) usage of credentials to identify policies and (3) secure storage of remote policies. Therefore, it is necessary to use a decentralized trust management system [22], like the KeyNote[16] or SPKI [23].

C. Mechanism

The separation of policy and mechanism makes it possible to implement a dynamic interface selection system. The mechanism evaluates connection association and transport information against the actions in policies, using the following principles:

- The mechanism must allow dynamic management of policies and actions including add, update and remove operations.
- The evaluation of policies should always result in exactly one interface for any traffic flow or connection. This is reached by having a priority order for actions.
- All attribute-value pairs in an action must match for a traffic flow or connection for the action to take place.
- The mechanism selects an interface based on the priority order of interfaces in an action.
- If the condition clause of an action is a match and the action is forced then the mechanism does not further evaluate

the following actions. If none of the interfaces specified by the action is available, the flow is unroutable.
- The mechanism uses a default action which match to all flows and connections if no other matching action is found.
- The mechanism should support distributed policy management and allow explicit definition of priorities between policies.

The mechanism binds a connection to a specific action at the connection initialization phase. When an action is updated during the operation, it affects all active connections related to that specific action. Only the interface related part of an action can be updated without affecting the binding between the action and a connection. A general rule is that an action cannot be deleted before all related connections are closed.

The interface selection mechanism co-operates with the packet routing mechanism. Typically a routing mechanism uses destination IP address based routing and searches the routing table for an outgoing packet in the following order: (1) matching foreign host address, (2) matching network address and (3) default entry.

This is inadequate for a multihomed mobile host having a simultaneous multiaccess capability. The connections cannot be routed via different interfaces to the same peer host if routing is based on destination addresses. That is why our implementation is based on a source address routing mechanism. The next hop router and outgoing interface are determined on account of the source address. This allows traffic flows to be divided between interfaces, even when they are destined to the same peer host.

An overview of the proposal of an interface selection mechanism is shown in a flowchart in Fig. 6. It also shows the connection between Home Address and interface selection mechanism. After a correct interface is selected at the connection initialization phase it is mapped to an Home Address that is further bound to a Care-of-Address (CoA). If some event later causes a change in the used interface, the mapping is changed to a new CoA.

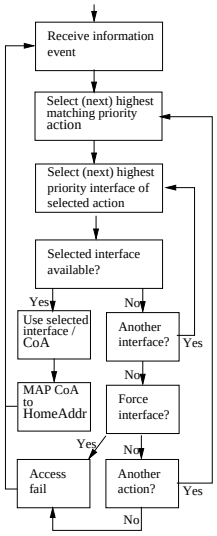

Fig. 6. Interface selection mechanism.

IV. Implementation in MIPL

The main design challenge was to incorporate the proposed multihoming and interface selection support into the HUT Mobile IPv6 stack, MIPL.

A. Architecture

The multihoming architecture in MIPL, depicted in Fig.7, consists of the following four logical components: core IP, MIPL with multihoming support, multihoming support API, and a control application.

The control application, which is shown as dashed lines, is used to manage actions in a policy. All the "intelligence" in the system is located on the user level in order to keep the kernel implementation as simple as possible. For minimizing the processing required in the kernel, the control application is also responsible for policy sanity checks.

The control application in our implementation further divides into two components: a daemon and a Java-based graphical user interface (GUI). Using the GUI, user-defined rules can be edited dynamically. The rules are used by the daemon to create the necessary policy actions, which are then sent to the kernel using the provided API. Using this API it is also possible to configure the rules using text-based configuration files, e.g., during the boot of the mobile node.

The next component, multihoming support API, enables user level applications to control the multihoming functionality. It also provides the control application information on the available interfaces and characteristics. An example of such information is a network interface status event, which is sent every time the availability of an interface changes.

MIPL with multihoming support (MIPL-MH), on top of the core IP layer, is a modified version of the original MIPL stack. MIPL-MH utilizes the core IP to propagate the changes caused by policy changes to the routing table and to the address selection database, which is used to assign a source address for the socket when it is created.

Each of the components defines a set of functions, which is used to access the component's services. The most interesting interface from the design point of view is the API, which exchanges the information between a user level application and a kernel level component. In our implementation a standard

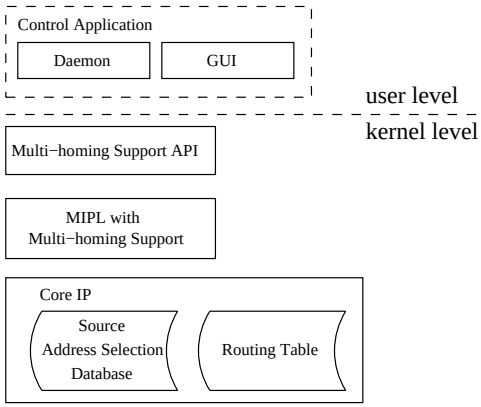

Fig. 7. MIPL multihoming architecture

Linux character device is used as the kernel/userspace interface, providing an easy programming interface.

B. Policy based Mobility Management

In our MIPL-MH implementation - *Simultaneous Multiaccess Mobile IPv6* - every single interface is mobile in the topological sense. Interfaces can move topologically from one access network to another, just like the host moves in a single-homed scenario. Similarly, when an interface changes its topological attachment to a network its new CoA is sent to the Home Agent (HA) and Correspondent Nodes (CN) using Binding Updates (BU). In addition to these *vertical handoffs* several interfaces may communicate simultaneously. Each interface has a configured CoA of it's own. To support this, the standard local routing of outgoing packet has been modified to fulfill the requirements of multihomed mobility and interface selection.

The multihomed Mobile Node (MN) has to configure one or more Home Addresses - one for each action, including the default action. This is a requirement of the interface selection mechanism and is not considered to be a scalability problem because of the large address space in IPv6. Fig. 8 shows the relations between actions, Home Addresses and CoAs. There is a one-to-one relationship between an action in the policy database and a specific Home Address. Packets are routed on account of the CoA bound to a Home Address. This is called *source address routing*. Based on the action the interface selection mechanism selects which Home Address is used as the source address for the outgoing packet. Using source address routing, we are able to support multihoming without modifying the standard Mobile IPv6 signaling, i.e, BUs. As a consequence, no modifications are needed in the CNs either.

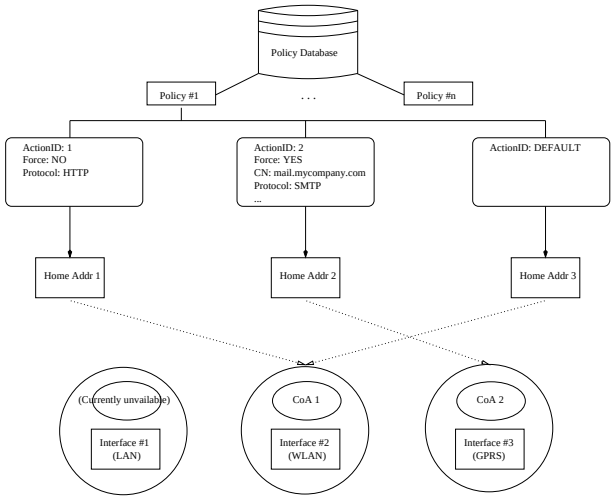

Fig. 8. Relations between policies, actions, Home Addresses, and Care-of Addresses.

If a MN has multiple simultaneous connections to a CN, the CN has one BU entry corresponding to each of the connected MN's Home Addresses. For the CN the situation looks identical as if there were several single-homed MNs communicating with it. During a *vertical handoff*, i.e., when connections are transferred from one interface to another, the pointer from the corresponding Home Address is redirected to the new interface. This process rebinds the Home Address to the CoA of the new interface. The new CoA has to be bind with the Home Address related to it and necessary BUs should be sent to the Home Agent and to all CNs communicating with the Home Address. This has a small drawback of increasing the amount of signalling during mobility if there are many different actions defined.

Our interface selection mechanism modifies the IPv6 routing table in correspondence to Fig. 8. The policy driver module within the kernel interprets the preferences stored by the user or operator and creates the necessary actions in the database, such as the default action. The available information on the interfaces is used as input in this process, both by the user and the policy driver. If some of the interfaces currently in use becomes unavailable or start violating some of the user-defined rules, regarding e.g. speed or price, the policy driver is able to look at the related action and select the "next best" interface thus modifying the routing table accordingly. At the same time, updated interface information is sent to the user. The different components of the policy implementation are shown in Fig. 9.

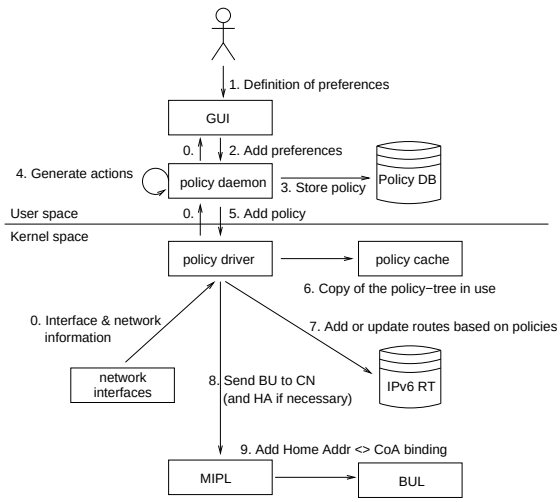

Fig. 9. An overview of the implemented mechanism.

The entries in the routing table are based on source addresses and are created dynamically in line with the policy database and information on the interfaces.

V. Security considerations

By allowing information to be updated dynamically through, e.g., router advertisements there is a significant security risk involved. When policies can be updated remotely, the door is open for malicious users as well. The attacks on interface selection can be divided into attacks against the policy information and into attacks against the technologies and protocols used (e.g., Mobile IPv6). The former can further be divided into false information propagation and Denial of Service (DoS) attacks. Threat scenarios against various technologies used are discussed, e.g., in [24] and are beyond the scope of this paper.

A. Threat scenarios

False information propagation is a straightforward way to attack against any mechanism, which relies on information not derived directly from node's own observations. As we rely on ISPs to provide us with information on network characteristics and operator policies, a malicious user may advertise cut-rate prices, higher bandwidth or other such information, which directly affects our interface selections. This will effectively redirect our traffic the way the attacker wants, if not taken care of. Attacker may also rapidly change the values, causing the network connections to oscillate between different access networks.

The aim of false information propagation can be either to interfere with target's communication capabilities or to redirect its traffic as a part of more sophisticated attack. If the purpose is solely to prevent the target from communicating properly, false information propagation can be seen as one form of a DoS attack. The attacker only needs to have some basic knowledge about the network to be able to advertise information, which will cause an unaware node to stop using the correct network and switch to the (non-existent) network advertised by the attacker. The attacker may also flood the target with policy information, which may result in discarding valid policy entries from the policy database.

Man-in-the-middle attacks require the attacker to lie in between the target and the entity with whom it is communicating. To achieve this, it might be necessary to redirect target's traffic. Advantageous policies can be used for this purpose. In wireless networks a malicious user may also insert an access point of his own with higher signal strength, thus causing nearby nodes to select it as default router. This form of attack does not apply to all sorts of wireless networks, but in WLAN networks this is quite easy to implement.

B. Protecting against threats

Dynamic interface selection necessarily needs some external information updates originated from the network to be able to work in a meaningful way. This in turn implies, that there must be a secure way to propagate information to nodes in the network. The information, as well as its source must be authenticated so that we know (with reasonable certainty) that we are only using legitimate information when making interface selection decisions. One possibility to propagate information is to extend Router Advertisements. A proposal to make them secure is presented in [25].

Security requirements vary greatly between different networks, so it must be possible to specify what level of security is needed in different situations. In corporate intranets or military networks the only acceptable level of security is "perfect" security, while in some situations within public networks authentication is not necessary. Strong authentication necessarily needs some prior arrangements between communicating nodes or a global Public Key Infrastructure - something that is currently not available globally. If the node wants to communicate with previously unknown peers, weaker authentication must suffice. Simultaneously, strong authentication can be used in smaller scale, e.g., in intranets or within ISPs.

To provide extra security and flexibility, it should be possible to set limits for certain values, e.g. "Never use prices higher than X" or "Never change interface more often than X times in Y minutes".

VI. Related work

There are few implementation proposals for interface selection in host multihoming that would allow explicit policy definition. Several Internet drafts [26], [27], [28], [29] identify the interface selection problem and present that the selection should be based on some policies. However, the drafts do not present any concrete solutions.

Existing host multihoming architectures (see Section II-C) typically combine a policy and mechanism together. Implicit routing rules do not allow end-users to affect dynamically on the interface and access router selection. SCTP defines an API that allows interface selection implementation on the application level. The main problem in transport layer multihoming protocols is that other existing transport layer protocols, like TCP and UDP, cannot take advantage of them. However, it is possible to extend many of the existing multihoming protocols to support user-defined interface selection policies.

If we allow a host to use several LIN6 IDs, we can use the interface selection mechanism presented in this paper also together with LIN6. In pure multihomed TCP, the selection of interfaces cannot be affected by the mobile user. If the PCBIs would not be generated randomly but rather based on the input given by our interface selection mechanism it would be possible to use the multihomed TCP together with our solution.

Further, Draves [30] defines a framework and algorithms for the address selection problem. The address selection rules are based on IPv6 address scopes and the length of prefixes. However, Draves does not take stance on how to select an interface for specific connections.

Ylianttila et. al present in [17] a handoff case study between GPRS and WLAN based on mobile IP. The handoff information is gathered at link layer (e.g., signal strength) and transmitted to a daemon program on the application level for decision making. Handoffs are made on account of implicit rules utilizing fuzzy logic. It would be possible to use this solution in parallel with our implementation.

VII. Future work

Our future work will focus on distributed interface selection policy management. This will require a common PKI architecture that allows nodes and operators to exchange policies. The architecture has to support some common policy description language. In addition, the policy language should allow an elegant way to define priorities between different operators' policies.

Our existing work is an initial step toward a distributed connection management architecture, where operators can affect on mobile node's roaming between different access networks. Further, a multihomed node may send preferences to its peer nodes, which help peer nodes to select a suitable destination interface. Such an architecture allows dynamic control of outgoing and incoming connections.

One possibility would be to bind a cryptographically generated home address (CGA) [31] to an action in a policy. This might open an interesting research area for operators granting policies to mobile nodes. The policies might control access to the access networks on account of CGAs.

Some effort will be put on investigating different methods that allow QoS information delivery from access routers to mobile nodes. CAR [21] specifies the requirements for an access router capability discovery protocol. RSVP [32] protocol, in turn, defines how QoS information can be presented in IP packets.

We will also analyze the current communication interface between policy daemon and policy driver to define a new API for *adaptive applications*. We will keep the current socket API untouched and define a new dedicated policy API. The policy API must allow applications to send preferences to and receive events from the policy driver.

VIII. Conclusion

Host multihoming protocols have to support both horizontal and vertical handoffs. The vertical dimension makes handoff algorithms and decision making more complex, but it enables robust communication. It is obvious that there must be a common mechanism for managing both kind of handoffs. Different access technologies and access operators offer different types of service. Therefore, a mobile user must be able to affect on the interface selection for connections.

Interface selection in existing multihoming protocols is mostly based on static rules. Typically the vertical handoff decisions applies to all connections routed through an interface. Unfortunately, the user cannot dynamically affect much to local routing decisions.

We have introduced an architecture which allows a user to dynamically create and modify interface selection policies and thus control how the network interfaces are used in a multihoming environment. Our architecture makes it possible to define policies for different connections on account of user preferences. Each connection is bound to a profile that contains local routing rules. Therefore, it is possible to make a vertical handoff to a single connection or to a group of connections without affecting any other connections that are using the same interface. The implementation is based on the Mobile IPv6.

Policy based local routing within multihomed mobile hosts seem to be a very interesting area of research. It is becoming more important as heterogeneous access networks are being deployed at an increasing rate. In the future, also network operators may be able to have an effect on the handoff decision within a mobile node. The authors believe that the enforcement point of such decisions will ideally be within the mobile node. However, some rules and preferences for this decision making may well originate from the operators in addition to those of the mobile user.

References

[1] Charles E. Perkins, "RFC3220: IP Mobility Support for IPv4," Jan. 2002.

[2] David B. Johnson and Charles E. Perkins, "Mobility Support in IPv6," Internet-Draft, Nov. 2001, Work in progress.

[3] Petri Jokela, Teemu Rinta-aho, and Jorma Wall, "Multiaddress Bindings in IPv6," in *Finnish Wireless Communication Workshop, Tampere, Finland*, Nov. 2001.

[4] B. Black, V. Gill, and J. Abley, "Requirements for IPv6 Site-Multihoming Architectures," Internet-Draft, Nov. 2001, Work in progress.

[5] Fumio Teraoka, Masahiro Ishiyama, Mitsunobu Kunishi, and Atsushi Shionozaki, "LIN6: A Solution to Mobility and Multi-Homing in IPv6," Internet-Draft, Aug. 2001, Work in progress.

[6] M. Py, "Multi Homing Translation Protocol (MHTP)," Internet-Draft, Nov. 2001, Work in progress.

[7] Christian Huitema, "Multi-homed TCP," Internet-Draft, May 1995, expired.

[8] R. R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. J. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, "RFC2960: Stream Control Transmission Protocol," Oct. 2000.

[9] Karim El-Malki Hesham Soliman and Claude Castelluccia, "Per-flow movement in MIPv6," Internet-Draft, Nov. 2001, Work in progress.

[10] Pekka Nikander, Janne Lundberg, Catharina Candolin, and Tuomas Aura, "Homeless Mobile IPv6," Internet-Draft, Feb. 2001, Work in progress.

[11] R. Moskowitz, "Host Identity Payload and Protocol," Internet-Draft, Nov. 2001, Work in progress.

[12] Sami Kivisaari, Niklas Kämpe, Juha Mynttinen, Toni Nykänen, Henrik Petander, and Antti Juhani Tuominen, "MIPL Mobile IPv6 for Linux," Software, http://www.mipl.mediapoli.com/, May 2000.

[13] Susan Thomson and Thomas Narten, "RFC2462: IPv6 Stateless Address Autoconfiguration," Dec. 1998.

[14] J. Noel Chiappa, "Endpoints and Endpoint Names: A Proposed Enchancement to the Internet Architecture," text, http://users.exis.net/ jnc/tech/endpoints.txt, 1999.

[15] Steven Bellowin, "EIDs, IPsec and HostNAT," Mar. 1998, Presentation in IETF.

[16] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis, "RFC2704: The KeyNote Trust-Management System Version 2," Sept. 1999.

[17] M. Ylianttila, R. Pichna, J. Vallström, J. Mäkelä, A. Zahedi, P. Krishnamurthy, and K. Pahlavan, "Handoff Procedure for Heterogeneous Wireless Networks," in *Global Telecommunications Conference - Globecom'99*. Dec. 1999, pp. 2793–2787, IEEE.

[18] Stephen Deering and Robert Hinden, "RFC 2460: Internet Protocol, Version 6 (IPv6) Specification," Dec. 1998.

[19] J. Reynolds and J. Postel, "RFC 1700: Assigned Numbers," Oct. 1994.

[20] Thomas Narten, Erik Nordmark, and William Allen Simpson, "RFC 2461: Neigbor Discovery for IP Version 6 (IPv6)," Dec. 1998.

[21] G. Krishnamurthi, "Requirements for CAR Discovery Protocols," Internet-Draft, May 2002, Work in progress.

[22] Joan Feigembaum Matt Blaze and Jack Lacy, "Decentralized Trust Management," 1999.

[23] C. Ellison, B. Frantz, and et.al B. Lampson, "RFC 2693: SPKI Certificate Theory," Sept. 1999.

[24] Allison Mankin, Basavaraj Patil, Dan Harkins, Erik Nordmark, Pekka Nikander, Phil Roberts, and Thomas Narten, "Threat Models introduced by Mobile IPv6 and Requirements for Security in Mobile IPv6," Nov. 2001.

[25] James Kempf, Craig Gentry, and Alice Silverberg, "Securing IPv6 Neighbor Discovery Using Address Based Keys (ABKs)," Aug. 2002.

[26] H. Berkowitz and D. Krioukov, "To Be Multihomed: Requirements and Definitions," Internet-Draft, July 2001, Work in progress.

[27] C. Huitema and R. Draves, "Host-Centric IPv6 Multihoming," Internet-Draft, July 2001, Work in progress.

[28] J. Abley, B. Black, and V. Gill, "IPv4 Multihoming Motivation, Practices and Limitations," Internet-Draft, June 2001, Work in progress.

[29] B. Black, V. Gill, and J. Abley, "Requirements for IPv6 Site-Multihoming Architectures," Internet-Draft, Nov. 2001, Work in progress.

[30] Richard Draves, "Default Address Selection for IPv6," Internet-Draft, Sept. 2001, Work in progress.

[31] Greg O'Shea and Michael Roe, "Child-proof Authentication for MIPv6 (CAM)," Apr. 2001.

[32] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "RFC2209: Resource ReSerVation Protocol (RSVP)," Sept. 1997.