Mobile Network and Applications manuscript No.
(will be inserted by the editor)

Multihoming Management for Future Networks

Bruno Sousa · Kostas Pentikousis · Marilia Curado

Received: 2010-11-15 / Accepted:

Abstract IP multihoming is a networking concept with a deceptively simple definition in theory. In practice, however, multihoming has proved difficult to implement and optimize for. Moreover, it is a concept, which, once adopted in the core Internet architecture, has a significant impact on operation and maintenance. A trivial definition of multihoming would state that an end-node or an end-site has multiple first-hop connections to the network. In this paper, we survey and summarize in a comprehensive manner recent developments in IP multihoming. After introducing the fundamentals, we present the architectural goals and system design principles for multihoming, and review different approaches. We survey multihoming support at the application, session, transport, and network layers, covering all recent proposals based on a locator/identifier split approach. We critically evaluate multihoming support in these proposals and detail recent developments with respect to multihoming and mobility management.

Keywords Computer networks · Computer network management · Reliability and performance · Multihoming · Multiaccess · Locator/identifier split · MPTCP · SCTP · SHIM6 · LISP · HIP · MIPv6 · PMIPv6 · MCoA

Bruno Sousa, Marilia Curado
CISUC, University of Coimbra
Polo II, Pinhal de Marrocos
3030-290, Coimbra, Portugal
E-mail: bmsousa,marilia@dei.uc.pt

Kostas Pentikousis
Huawei Technologies
European Research Centre
Carnotstrasse 4, 10587 Berlin, Germany,
E-mail: k.pentikousis@huawei.com

1 Introduction

Multihoming and multiaccess in IP networks have been lately fostered by the exponential growth in availability of devices with multiple built-in communication technologies. Paradigms where hosts have access to various networks are not new, of course. Multihoming has long been adopted to increase resilience, dependability, and performance in high-end servers. At the other end of the network node spectrum, mobile phone manufacturers have been integrating different cellular radio access technologies into "multi-band" cell phones to realize global reachability and ease migration. Nonetheless, multiaccess network selection is currently rudimentary and automation is not implemented. Today, efficient multihoming and multiaccess support in heterogeneous networks is still inhibited by mechanisms that rely mainly on presets and static policies, and require user input as well.

Nodes with multiple network interfaces have the potential of connecting to different networks and capitalizing on heterogeneous network resources and, in the process, enable their users to enjoy high-performing, ubiquitous communication. On the other hand, multiaccess and multihoming lead to more intricate application and protocol configurations in order to meet the challenging goals of reliability, ubiquity, load sharing, and flow distribution. These communication system properties are tightly coupled with the multihoming concept. For instance, the Stream Control Transport Protocol (SCTP) [6] natively uses a *primary-backup* model to deal with failures in active paths, over and above the path failure recovery mechanisms provided by the network layer. Still, multiaccess and multihoming are yet to become prevalent in network deployments despite years of research and development in the area. Indeed,

the corresponding support is often missing from state of the art protocols. For example, modern mobility management protocols, such as Mobile IPv6 (MIPv6) [24] are not capable of handling multihoming natively and must be combined with other protocols, such as Site Multihoming by IPv6 Intermediation (SHIM6) [4,8,16], to enable enhanced multihoming support.

Over the years, different solutions have been put forth, depending of whether they are designed for end-host or end-site multihoming. Proposals may also target multihoming support at different layers of the TCP/IP protocol stack, namely, at the application, transport, and network layers. Furthermore, in some proposals new layers are introduced, such as the Host Identity Protocol (HIP) [17] or SHIM6. The newly introduced layers perform specific functionalities and aim at reducing the ensuing complexity due to multihoming mechanisms in the original protocol stack.

From an end-site perspective, routing scalability is a concern that is driving research towards novel proposals such as, the Routing Architecture for the Next Generation Internet (RANGI). RANGI aims to be incremental or even to be employed by end-nodes or nodes with routing and forwarding functions, like Identifier Locator Network Protocol (ILNP). These proposals rely on a locator/ identifier split approach but differ on how identifiers are set.

This paper provides a comprehensive survey of protocols supporting end-host and/or end-site multihoming, as opposed to previous overviews on the matter [8, 13, 38] which focused only on a subset of multihoming protocols. Our evaluation of multihoming solutions is not restricted to a single criterion, such as cost [29], for example. Instead, we base our analysis on the degree of fulfillment of multihoming goals (i.e. resilience, ubiquity, load sharing, and flow distribution). We also adopt a simplified taxonomy – end-host and end-site multihoming – instead of distinguishing between routing, middle-box, core-edge and host-centric multihoming solutions, as the latter approach can lead to subjective interpretations and hence evaluation results. Finally, we overview multihoming support at all layers of the protocol stack, including network, transport, and application, considering all salient recent work in the area.

The remainder of this paper is organized as follows. Section 2 introduces multihoming definitions and related terminology, and presents design considerations for multihoming solutions. Multihoming support of mobility management protocols is discussed in Section 3. Section 4 discusses multihoming support in transport protocols. Section 5 overviews proposals aiming at end-host multihoming, while Section 6 discusses end-site multihoming approaches. Finally, Section 7 provides an outlook on future research directions in the area of multihoming management and concludes the paper.

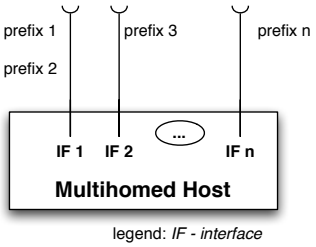

Fig. 1 Multihomed host

2 Multihoming Concepts and Design

This section presents end-host and end-site multihoming types, and clarifies terms related to multihoming, such as multiaddressing and multiaccess. Guidelines to enable multihoming goals are also discussed.

2.1 End-host and End-site Multihoming

A multihomed host, on which different interfaces (logical or physical) exist, is depicted in Fig. 1. In addition, each interface can have different network prefixes configured. For instance, interface *IF 1* has been assigned two prefixes, namely *prefix 1* and *prefix 2*. Moreover, the host can have multiple physical interfaces which have been associated with a single prefix, as is the case of *IF 2* and *IF n* with *prefix 3* and *prefix n*, respectively. Note that here we use the terms prefix and address interchangeably. From an *end-host* perspective, a multihomed host has multiple prefixes configured on the links it connects to, thus having the possibility to explore several paths to reach a peer, as each prefix is normally advertised by different access routers [10].

Fig. 2 illustrates a multihomed site, which has connections to two service providers. A multihomed network can have multiple routers, such as, for example, *MR 1* connecting to Internet Service Provider 1 and *MR 2* connecting to Internet Service Provider 2. Moreover, a single router can have several external interfaces that connect to the same or different service providers, as the example of *MR 1*. *End-site* multihoming, where a site uses multiple connections to the Internet to meet objectives such as increasing network reliability or improving performance [10,11], is a common network configuration.

Wang et al. [47] explain that multihoming support in a given protocol can follow different approaches. In

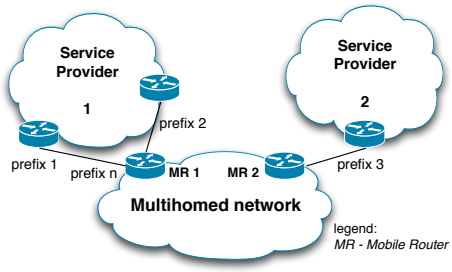

Fig. 2 Multihomed network

the ownership approach, the entity owning the Home Agent (HA) and mobile routers, and providing Internet access to multihomed network elements plays a key role. If these network elements are controlled by a single entity, this is called the Internet Service Provider (ISP) model, otherwise it is referred to as the Subscriber/Provider model. On the other hand, the configuration -oriented approach considers parameters such as the number of Home Agents or the number of prefixes advertised.

Multihoming is lately associated with other concepts, including multiaddressing, overlapping networks, multiple interfaces and overlay routing. Multiaddressing, for example, corresponds to a configuration in which multiple addresses are assigned to a given host based on prefixes advertised in different connections [5]. Overlapping networks correspond to networks that are configured in a way that there is a common area of coverage. Typically, mobile and wireless end-nodes connecting to these (overlapping) networks must have multiple interfaces, each one specific to the technology sustaining the respective network [46]. Finally, overlay routing is associated with inter-domain routing techniques that improve fault-tolerance, and is only applied in an end-site context.

2.2 Goals

Multihoming has gained attention over the last few years [11], mainly due to the potential benefits. In particular, multihoming solutions aim to achieve the following goals: **R**-Resilience, **U**-Ubiquity, **L**-Load balancing/sharing and **F**-Flow distribution.

The diversity of multiple interfaces/paths can improve *resilience* as upon a failure of one interface/path, another can be employed to provide connectivity. For instance, as mentioned above, a *primary-backup* model is adopted by SCTP [6]. That is, if the primary path fails, the backup path can be used seamlessly without causing any application-layer service interruption. Multiple network interfaces, in particular when used in a mobile and wireless network environment, enable *ubiquitous access* to the Internet over different media.

Load sharing goes one step further than the primary-backup model, as multiple interfaces/paths can be used simultaneously to improve throughput. For example, Iyengar et al. [20] describe how one can perform concurrent multiple transfers using base SCTP.

Flow distribution, or flow stripping, offers an even finer granularity than load sharing. For many, flow distribution is the ultimate goal to achieve, as it implicitly means that all previous goals are also attained. Flows are stripped, perhaps even dynamically, according to policies and preferences aiming to reduce cost, optimize bandwidth use, and minimize the effect of bottlenecks to delay-sensitive applications, among others. Such policies can be defined by users or service providers [35].

Multihoming support could potentially be added at any layer of the protocol stack. The designer's choice, of course, comes with certain pros and cons, and one needs to consider thoroughly the tradeoffs as well as the complexity of each solution. Deployment considerations need also to be addressed early on. There are two possible approaches for introducing multihoming. On the one hand, a multihoming proposal may be completely transparent to upper layers, in such a way that there is no disruption to ongoing sessions. On the other hand, the solution may not be transparent [36], but allows upper layers to participate in multihoming management and operation.

2.3 Multihoming Design Considerations

Architecture proposals for multihoming trying to address issues such as failure detection, security, path selection and default gateway choice [36, 46], should consider different design guidelines to meet one or more of the multihoming goals. Briefly, design considerations include adopting a locator/identifier split approach for end-host and end-site multihoming and the modification of site exit routers for end-site multihoming to support scalability and security.

The first guideline that should be considered relates to the **locator/identifier split**. Conventional IP architectures assume that the transport layer endpoints are the same entities as those used by the network layer. Thus, multihoming support based on a locator/identifier split requires that the transport layer identity is decoupled from the network layer locator in order to allow multiple forwarding paths to be used by a single transport session. Different approaches can be considered [8], either by modifying an existing protocol layer or by introducing a new layer. With the new layer approach,

upper layer protocols (e.g. applications) use endpoint identifiers to uniquely identify a session while the lower layer protocols (e.g. network) employ locators. If this approach is used, a mapping between an identifier and a locator is necessary. In a multihoming context, the identifier/locator mapping must be assured by a dynamic process so that a session can include different features, such as constant endpoint identifiers throughout the session lifetime, and modification of locators to maintain end-to-end reachability.

Another recommendation for end-site multihoming includes the **modification of a site exit-router**. End-site multihoming can be assured by a network element. For instance, an exit-router can perform packet rewriting for a given locator of a correspondent node. Nevertheless, this type of approach raises security concerns, which might be difficult to overcome. Redirection attacks are such an example, which may compromise routing, since packets for a destination can be redirected to any location [8, 13]. Thus, the host should always be able to perform the endpoint-to-locator mapping on its own.

Scalability is of essence in any network architecture and multihoming is not an exception. Multihoming architectures should be scalable and need to strive to minimize the impact on routers and end hosts. Basic connectivity must be always provided. If any modification is required it should be in the form of logically separating added functions from existing ones [11].

Security is also paramount for future architectures. Multihoming proposals should not introduce new security threats. For instance, multihoming solutions should be resilient to redirection attacks that compromise routing, new packet injection attacks (malicious senders can inject bogus packets into the packet stream between two communicating peers) and flooding attacks, which are normally associated with Denial of Service attacks [13].

3 Multihoming and Mobility Management

This section overviews multihoming support in IPv6-based protocols. IPv4-related protocols are left out of scope as their solutions for multihoming are less scalable and not forward-looking.

3.1 Mobile IPv6 and Proxy Mobile IPv6

MIPv6 [24] is to a large degree the archetypical mobility management protocol for IPv6 networks. Maintaining established communications while moving is similar to preserving established communications through outages in the multihoming context. MIPv6 maintains established communications while a mobile node moves across networks. However, current MIPv6 does not fully support multihoming, as it assumes that the home address does not change during the mobility management process. With such an assumption, whenever there is a change in the home address, e.g. a node with multiple prefixes in the home network, MIPv6 does not support new addresses acting as the home address. Even if binding update messages convey information in advance about alternative prefixes [8], this may not be enough to enable session survivability, as MIPv6 procedures fail, since they rely on a single address.

Proxy Mobile IPv6 (PMIPv6) [24] is a network mobility management protocol designed to assist IPv6 mobile nodes that do not have functionality to support mobility management. PMIPv6 introduces two entities, namely the Local Mobility Anchor (LMA), which acts as the Home Agent of the MN; and the Mobile Access Gateway (MAG) which is an access router capable of managing the signaling for a mobile node attached to its link. PMIPv6 supports multihoming according to the following scenarios, detailed in [22]. In the **unique prefix per interface** scenario each interface of the mobile node is assigned a unique prefix. LMA maintains multiple binding cache entries and can sustain separate routes for each prefix. In the **unique address per interface** scenario, the mobile node has the same prefix across multiple interfaces but with a unique address per interface. For instance, the mobile node can connect to the same subnet via two interfaces. LMA maintains a separate binding cache entry per address of the mobile node and routing entries per address assigned to MN.

3.2 Multiple Care of Addresses and Flow Bindings

The Multiple Care of Address (MCoA) proposal [34] extends MIPv6 to allow the registration of multiple Care of Addresses. With several Care of Addresses the mobile node can maintain concurrent paths with its correspondent nodes [28]. The mobile node is always reachable at a unique permanent IPv6 address (employed as an identifier) while several temporary addresses (Care of Addresses) used as locators to reveal the current network location of the node. Since locators change over time, each path is identified with a Binding Unique Identification (BID) number. Moreover, multiple registrations can be conveyed in a single message to reduce overhead.

The enhanced multihoming support of MIPv6, empowered by MCoA registration, lacks a specification on how multiple registered addresses can be used. For instance, if the addresses can be used simultaneously, or if an address is chosen based on the link characteristics. Nevertheless, a non-standard mechanism may lead to a

situation where different MCoA implementations [39] become non-interoperable.

The specification of flow bindings [42] extends MIPv6 and MCoA specifications defining how multiple flows can be exchanged between two nodes, in a multihoming context. This enables to bind a particular flow to a Care of Address and use another address to receive information from other flows. The flow bindings specification permits the conveyance of policies between the mobile node and other mobility agents (e.g. home agents) [42]. Whilst the flow bindings specification deals with the transfer of policies, the way they can be generated or mapped to user preferences (e.g. link with higher bandwidth) is left out of scope.

3.3 Network Mobility

Network Mobility (NEMO) is a protocol [25] that manages the mobility of a network of nodes typically moving in tandem. NEMO Basic Support extends MIPv6 procedures, through the addition of the Mobile Router (MR) entity. Each Mobile Network Node is connected to MR, and all together they form the mobile network. A mobile network (NEMO) is considered multihomed when a MR has multiple egress interfaces connecting to the Internet, or when there are multiple MRs or multiple global prefixes on the network [47].

Each of the multihoming goals has different requirements for NEMO multihoming support [47]. In order to achieve permanent and ubiquitous access, at least one bi-directional tunnel must be available. For reliability, both inbound and outbound traffic must be transmitted over another bi-directional tunnel once the active one fails. Moreover, multiple simultaneous tunnels must be maintained to assure load sharing and load balancing. NEMO Extended Support (NEMO-ES) [9] enables route optimization and policy based routing. Multihoming support is improved, as care is taken with the choice of the router that will route packets in a nested mobile network.

3.4 Summary

The main restrictions of MIPv6 for multihoming include the assumption that the Home Address does not change during mobility and the use of a single binding between a Care of Address and the Home Address [24]. MCoA [34] and flow bindings [44] overcome such restrictions, but do not provide standard mechanisms to enable load sharing and local policies, respectively. PMIPv6 [24] addresses a key deployment issue by providing mobility management support to nodes which are not MIP-aware. Nonetheless, it requires support from the network and does not provide for achieving the load balancing or resilience multihoming goals. NEMO, in comparison to MIPv6, has ubiquity capabilities, as mobility is supported to a greater extent, but has limited multihoming features. As mentioned above, multihoming support of IPv6 mobility management protocols can be enhanced by employing other protocols, such as combining MIPv6 with SHIM6 [16, 30].

4 Multihoming and Transport Protocols

This section is devoted to an overview on the multihoming support at the transport layer.

4.1 MultiPath TCP

Multipath Transport Control Protocol (MPTCP) [15] allows the simultaneous use of diverse paths that can exist between two end hosts. The goals of MPTCP include throughput and resilience improvement by performing resource pooling, on which multiple addresses can be associated transparently with applications. Initially, MPTCP establishes a basic connection. When establishing a connection, peers exchange their MPTCP capabilities. If multiple addresses are available, additional subflows are added for these addresses to the already established connection. The Multipath TCP API [37] allows MPTCP-aware applications to control MPTCP operation. Through the API, applications can activate or deactivate MPTCP for certain data transfers, can query MPTCP regarding the used addresses on subflows, and obtain connection identifier.

4.2 Non-Standard TCP-based proposals

With the Multiple TCP Fairness proposal [43] an application may employ multiple TCP instances to stripe packets across different available paths. The issue with this approach resides on the independence of each data path. For instance, it is hard to guarantee that multiple TCP instances do not use more bandwidth than a single TCP instance over the path. In other words, a “fairness” issue arises, as greedy applications employing several TCP connections in parallel can grab a larger portion of what is their fair share of network resources. The Multiple TCP Fairness proposal allows multiple TCP instances but ensures that an application does not take a disproportionate share of the available bandwidth.

FAST TCP [48] is a TCP variant that employs a delay-based congestion control algorithm. Arshad and

Mian [1] propose an extension to FAST TCP to support multihoming and improve end-to-end throughput, by introducing mechanisms at the sender and receiver. A drawback with the FAST TCP multihoming mechanism is its susceptibility to throughput problems, namely, on network congestion situations.

4.3 Stream Control Transport Protocol

SCTP is a connection-oriented protocol designed to assure reliable transport [6] and support multihoming natively, through several mechanisms. First, via address management at association setup, during which a node informs its peers about its IP addresses (or host names). Second, *HEARTBEAT* chunks are employed to monitor peers and path status (active or inactive). SCTP uses a selective acknowledgements (SACKs) mechanism to enable accurate RTT measurements over each path. Finally, for path selection, as the association setup proceeds, an active path is chosen as the primary path. The SCTP API [12] allows applications to configure the behavior of SCTP, as for instance, to support connection-oriented features (e.g. as TCP) or connection-less features (e.g. as UDP).

Mobile SCTP (mSCTP) [14] extends SCTP to mobile environments. mSCTP allows dynamic address reconfiguration by modifying IP addresses that were negotiated during the SCTP association setup. Such support is specified with new message types that contain the IP address and parameters to indicate the operation to perform, namely add, remove or modify the primary address. mSCTP can be employed by fault-tolerant applications, which require fast recovery.

Concurrent Multipath Transfer (CMT) [20] adds simultaneous data transfer capabilities across multiple paths to SCTP. CMT addresses some performance issues of SCTP, such as unnecessary fast retransmission at the sender and increased ACK traffic due to fewer delayed ACKs. If the available paths have unequal delay or bandwidth, a standard SCTP receiver can experience packet reordering, which will consequently lead to fast retransmission at the sender. CMT mitigates these issues by introducing modifications in the SCTP specification, where a receiver delays the ACKs, instead of immediately acknowledging out-of-order packets. Further, the packet loss measurement mechanism takes into consideration historical information, in addition to the information conveyed by SACKs.

4.4 Summary

TCP, used by the vast majority of Internet applications, is being pushed forward in terms of multihoming support through the efforts around developing MPTCP. MPTCP can attain the resilience and load sharing multihoming goals using novel end-host congestion control mechanisms while following the original design rationale of TCP. The rest of the above mentioned non-standard TCP-based proposals are not fully compatible and offer limited multihoming support. For instance, TCP Fairness [43] allows load sharing when compared to FAST TCP [48] at the expense of additional overhead. SCTP, designed natively with multihoming capabilities has mSCTP and CMT extensions to enable ubiquity and load sharing, respectively. Nevertheless, SCTP is not as widely adopted as TCP in the Internet. Other protocols, such as DCCP and UDP, due to their unreliable nature, do not support multihoming efficiently or have limited support [8].

5 End-host Multihoming

This section overviews protocols and architectures tailored for end-host multihoming support.

5.1 Host Identify Protocol

HIP [17] is a protocol that adopts a locator/identifier split approach and supports multihoming natively. HIP introduces a new host identity namespace and a new host identity layer between the network and the transport layers. In addition, HIP decouples identifiers (used by transport layer protocols) from locators (used for routing purposes). In short, the transport layer sockets and the IP security associations are bound to host identifiers, which in the end are tied to IP addresses.

Multihoming support in HIP is based on two approaches: *LOCATOR* parameter and *RendezVous* service [18]. Using the *LOCATOR* parameter approach, a HIP host can notify a correspondent peer about alternate addresses through which it is reachable. With the HIP *RendezVous* service, each HIP host publishes its host identifier with a *RendezVous* Server. The *RendezVous* Server maintains the mapping between the host identifiers and the locators, with limited support for mobility. HIP may raise issues with firewalls and middleboxes that need to inspect packet contents. Also, multihoming support does not include traffic engineering or policy address selection schemes. With HIP API [23] applications can start communications with unknown peer identifiers or perform explicit mapping.

Pierrel et al. [35] introduced a policy system for simultaneous multiaccess based on HIP (HIP SIMA). The proposal extends HIP by allowing flows to use different paths independently of each other, since HIP does not support load sharing. To enable flow distribution, flows are identified by source and destination ports and by the Host Identification Tag. The *RendezVous* Server is also extended to be able to store flow policies. Whilst these policies define the usage rules of the available interfaces, the proposal does not detail the policy specification (e.g. rules actions, interface priority, and cost).

5.2 Site Multihoming by IPv6 Intermediation

SHIM6 [16] is a multihoming protocol that adds a shim layer in the IP stack of end hosts. SHIM6 brings the advantage of assuring transport layer communication survivability, as the identity and location functions are split. For instance, the switch between address pairs is transparent to applications, since the identifier is only used to identify endpoints, while the locator is used to perform routing. In this split, SHIM6 provides the mapping function between Upper Layer Identifier and locator at the receiver and sender end-hosts.

SHIM6 uses failure detection and recovery mechanisms described in the Reachability Protocol (REAP) [31], which work independently from upper layer protocols. Failure detection can be based on keep-alive mechanisms or using information from upper layers (e.g. TCP control features). Recovery mechanisms rely on the exploration of available addresses, so that in the end an operational pair can be found and used.

Despite providing fault tolerance, SHIM6 breaks the functionality of some protocols, such as Internet Control Message Protocol (ICMP), since routers on the path cannot see the host identifier. Notwithstanding, SHIM6, when compared to other multihoming solutions, for instance HIP, has the advantage of an easier deployment in the Internet [10], since SHIM6-compatible hosts can communicate with other nodes that are not SHIM6-aware. SHIM6 is accompanied by a socket API that allows applications to access information about failure detection and path exploration [12]. Through this API, applications can turn on/off the shim functionality, and get/set preferred source and destination locator(s).

5.3 Name Based Sockets

The Name Based Sockets (NBS) proposal [45] introduces a novelty that facilitates multihoming. Applications use domain names only, while IP addresses (e.g. selection, discovery) are managed by the operating system. Such functionality is proposed as an extension to the standard socket API. Nodes communicating with each other exchange names through an IP-Option/IPv6 extension header. The receiver, upon encountering such option, also adds its name on the reply packets. The name can be based on a Fully Qualified Domain Name (FQDN), on ip6.arpa (host interface address), or nonces that identify different sessions. The ports rely on IANA service keywords (e.g. http for port 80). The Name Based Sockets proposal can be combined with other protocols, such as MIPv6, to add mobility management support. Nevertheless, NBS is still work-in-progress and requires node modifications. Finally, NBS removes the possibility of applications to use multiple addresses according to their own requirements.

5.4 Practical End-host Multihoming

Practical End-host Multihoming (PERM) [41] enables flow scheduling in multihomed hosts. This framework extends the Linux socket API to allow a host to explore different paths on a flow-level basis. PERM also introduces the concept of collaborative multihoming in which users share their Internet connection with others. PERM includes different functions to allow this collaboration. For instance, besides the connection manager and the monitor, the incentive manager creates incentives to share Internet access, based on policies (e.g. user shares when the connection is idle). The hybrid flow scheduling algorithm in PERM considers the flow volume, the load of a link and the respective associated RTT. For instance, a flow with a light volume is scheduled on the connection with the smallest RTT, while others are scheduled based on the predicted flow volume and current load of each link. Nevertheless, optimal performance is obtained with prediction information which depends on particular scenarios.

5.5 Strawman Architecture

Strawman [19] is an architecture performing flow striping at the session layer to improve the performance of applications in nodes with multiple interfaces. The Strawman architecture aims to allow striping over multiple connections, maximize throughput, and minimize delay, jitter and loss. Moreover, it also supports multimedia applications by allowing in-order delivery but without transport guarantees. To achieve such goals, different functionalities are included in the architecture. For instance, path evaluation mechanisms assess the

Table 1 Comparison of End-host Multihoming proposals

Protocol	Approach	Multihoming Goals				Pros	Cons
		R	U	L	F		
HIP	Loc/ID split	√	X	X	X	IP family agnostic; Security	Complicated implementation and deployment
HIP SIMA	Loc/ID split	√	√	√	√	Security	Limited policy specification
SHIM6	Loc/ID split	√	X	X	X	Easier deployment than HIP	Mobility and security issues
Name Based Sockets	Loc/ID split	√	X	√	√	Avoids addresses at application layer	Requires node changes
PERM	Flow Strip	√	X	√	√	Security	No LoC/ID split support; requires application modification
Strawman	Flow Strip	√	X	√	√	Security	No LoC/ID split support; requires application modification

service on a path based on network metrics. The Strawman architecture requires modifications to the sockets API to allow an efficient interaction with all transport protocols. This API must be available between the session layer and applications. In addition, the architecture does not follow a locator/identifier split approach, which in a sense limits its potential for mobility management and requires application modifications.

5.6 Summary

End-host multihoming proposals can follow different approaches, as summarized in Table 1. The Locator/Identifier (Loc/ID) split is one of the approaches aiming to break the dual role of IP addresses. SHIM6 is a locator/identifier multihoming approach that adds a shim layer between the network and transport layers. SHIM6 uses REAP to perform the detection of invalid locators and recover in an application-independent fashion. Nevertheless, SHIM6 must be combined with other protocols, such as MIPv6, to provide mobility support.

HIP [17] is an identity protocol that also decouples identifiers from locators. Its multihoming support relies on two approaches, one that resorts to the inclusion of new options in HIP messages, that is, the *LOCATOR* parameter, and another that employs a *RendezVous* Server that maintains the mapping between identifiers and locators. Extensions to HIP [35] introduce load sharing and flow distribution support. The *RendezVous* servers are updated to store flow policies and HIP messages are updated to convey policies.

Both the Strawman architecture [19] and PERM [41] introduce flow stripping mechanisms. Whilst such approaches have finer grain capability (e.g. support of flow distribution according to policies), they require modifications on applications.

6 End-site Multihoming

End-site multihoming has gained more attention than end-host multihoming, mainly due to the routing scalability problems that Internet is facing. This section presents end-site multihoming approaches.

6.1 Locator/Identifier Insights

In the context of a Loc/ID implementation, different approaches can be pursued, namely the so-called *map-and-encap* and *address rewriting*. The map-and-encap approach, as depicted in Fig. 3, is based on mapping and encapsulation processes as follows. A source host, on a domain sending a packet to a destination, inserts the source Endpoint Identifier (EID) and the destination EID in the packet header (Fig. 3:1). When the packet arrives at the border router of the same domain, the Ingress Tunnel Router (ITR) performs the mapping between the destination EID and the Routing Locator (RLOC) (Fig. 3:2-mapping phase). After the successful mapping, the ITR encapsulates the packet and sets the destination address to the RLOC retrieved in the mapping phase (Fig. 3:3-encapsulation phase). Finally, the packet arrives at the destination domain, on which a border router, the Egress Tunnel Router (ETR), performs the decapsulation and the delivery to the destination EID (Fig. 3:4). The advantages of this approach are the support of both IPv4 and IPv6, leaving end hosts unchanged, and minimizing the modifications in the routing system.

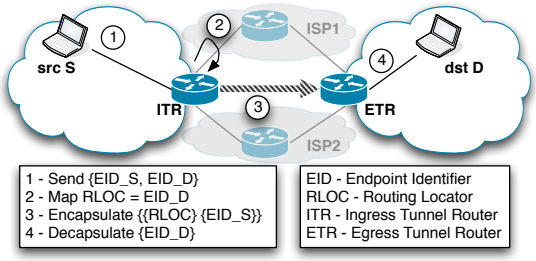

Fig. 3 *Map and encap* approach

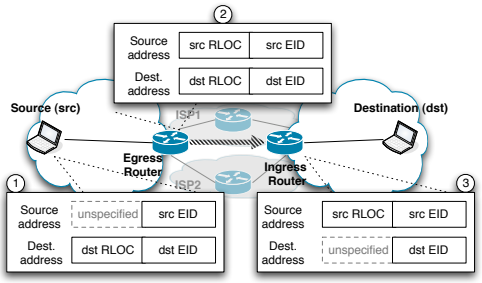

Fig. 4 *Address rewriting* approach

In the address rewriting approach, the 128 bits of an IPv6 address are split, where the 64 most significant bits are used as the routing locator and the 64 least significant bits are used as the endpoint identifier. Fig. 4 illustrates the process of address rewriting. The routing locator information is not known by the end nodes (source and destination). Whilst this approach only supports IPv6, it allows for consistency between prefix assignment and physical network topology.

6.2 Address Rewriting Approaches

The Global locator Local locator and Identifier Split (GLI-Split) [27] is a locator/identifier addressing and routing architecture. GLI-Split implements a global locator, local locator and identifier split, that is, it distinguishes locators for local routing (e.g. inside a domain) from those used for global routing. To allow compatibility with IPv6 protocols, locators and identifiers are coded as IPv6 addresses. GLI-split works by performing address rewriting carried out by the GLI-gateway with the assistance of the mapping systems. Moreover, GLI-split introduces two types of mapping systems. The local one is restricted to a domain, while the global mapping system is used for the global routing domain (e.g. Internet backbone). GLI-split supports mobility, but requires modifications to protocols like Dynamic Host Configuration Protocol (DHCP) to support multihoming.

ILNP [3] is another proposal that implements locator/identifier split by employing address rewritting. The locator is used to route traffic, while the identifier is employed as a node identifier without topological significance. Applications bind their sessions to the identifier and not to the locator. ILNP divides the IP address into a 64-bit identifier and 64-bit locator. If the identifier is globally unique, procedures like Duplicate Address Detection (DAD) are not necessary, which improve mobility support. In addition ILNP can be employed for end-host multihoming with IPv4 or IPv6 network stacks. Nevertheless, ILNP requires modifications to DNS in order to allow nodes to update their locator records.

RANGI [26] introduces a host identifier layer between network and transport layers. The host identifier has an organizational structure to allow easier mappings between locators and identifiers. The locators are based on IPv4 addresses embedded in IPv6 addresses, in such a way that the domain identifier is a 96-bit prefix (assigned by the provider) and the remaining 32 bits correspond to a private or public IPv4 address. In the address rewriting approach of RANGI, the mapping between domain name and host identifiers is done via DNS, while the mapping between identifiers and locators is performed on a distributed mapping system. RANGI allows incremental deployment and facilitates the migration from IPv4 to IPv6 networks.

6.3 Hierarchical Approaches

Hierarchical IPv4 (hiPv4) [26] is a framework that splits the core address space (ALOC) from the edge address space (ELOC). ALOC is globally unique, while ELOC is only used for routing and forwarding purposes inside the local domains. With ALOC and ELOC split, there is a hierarchical organization of addresses, in the sense that the ALOC can correspond to the AS. hiPv4 introduces a Locator Swap router to perform the change between the prefixes and the introduced locator header that includes information about the ELOC and ALOC elements. Additionally a host identifier scheme is introduced to avoid locator renumbering at security nodes (e.g. firewalls). hiPv4 requires modifications to DNS, nodes, routers and security elements (e.g. firewalls) that do not facilitate its implementation. In addition, hiPv4 may break the functionality of other protocols, such as Mobile IP, since the IPv4 header is changed.

Aggregation with Increasing Scopes (AIS) or evolution [21], is a locator identifier split approach on which prefixes are aggregated in different steps and according to their scope. The first step aggregates prefixes with the same next hop. A second step configures a router

Table 2 Comparison of End-Site Multihoming proposals

Protocol	Multihoming Goals				Pros	Cons
	R	U	L	F		
GLI-Split	√	√	√	X	Security	Requires nodes changes
ILNP	√	√	√	√	Supports end-host multihoming	Requires changes to DNS
RANGI	√	√	X	X	Facilitates IPv4 to IPv6 migration.	Requires changes to hosts
hiPv4	√	√	√	√	Hierarchical organization	Impacts other protocols
AIS	X	X	X	X	Address aggregation done by scope	Unclear Multihoming support
IRON-RANGER	√	√	√	√	Follows a business model.	Relies on an overlay network
IvIP	√	X	√	√	Mobility supported with extensions	Scalability issues
HIP MR	√	√	X	X	Security	For HIP-aware nodes only
LISP	√	X	√	X	Flexible mapping	Encapsulation overhead

as an Aggregation Point Router (APR) that aggregates prefixes as a virtual prefix. Other routers, not acting as APRs, store only routes announced on the virtual prefixes. Aggregation leads to reduction in the mapping sizes, nevertheless may also lead to route traffic through non-optimal paths since they must traverse the APR.

IRON-RANGER [40] implements an overlay network, on which specific routers manage virtual prefixes, from which provider independent prefixes are leased to end-nodes (e.g. customer sites). This proposal introduces serving routers, clients in end-user networks, and relay routers. The serving routers perform forwarding and mapping services, while the clients connect end user networks to the overlay network, via tunnels. The relay routers connect the IRON network to the rest of the Internet, and also perform the function of advertising virtual prefixes. The hierarchical organization of IRON-RANGER makes it scalable and facilitates deployment.

Mobility and Multihoming support Identifier Locator Split Architecture (MILSA) [33] is a Loc/ID-based proposal that introduces different hierarchies in the network, namely the Real-Zone Bridging Server (RZBS) hierarchy and the Realm Hierarchy. The Realm Hierarchy corresponds to a logical concept, in which the trust relationships between different groups of objects are maintained. The RZBS Hierarchy contains an overlay network of RZBS servers which map identifiers to locators. MILSA does not affect DNS and includes support for mobility. The Enhanced MILSA (EMILSA) [32] avoids global routing and improves MILSA with respect to mobility and multihoming. EMILSA does not affect DNS as the Loc/ID-based proposal introduces different hierarchies in the network. In addition, a specific sublayer is added in the network layer to perform the separation between identifiers and locators. Nevertheless, the (E)MILSA architecture is still at an early stage of development and neither simulation nor actual code is available to the research community.

6.4 Map and Encapsulation Approaches

The Internet Vastly Improved Plumbing (IvIP) Architecture [50] is a core-edge split proposal implementing a map-and-encap approach. IvIP uses a fast-push mapping scheme, where all mapping information is kept on query database servers. Ingress tunnel routers query database servers to determine the correct egress tunnel router, to which traffic must be routed. IvIP works for IPv4 and IPv6 and supports mobility through extensions. Nevertheless, the mapping requires real-time monitoring of the reachability of egress routers, and in addition, it has scalability issues.

The Locator Identifier Separation Protocol (LISP) is a map-and-encap protocol [7] aiming to improve site multihoming, decouple site addressing from provider addressing, and reduce the overhead associated with routing tables (e.g. size and latency lookup operations). To implement such goals, LISP specifies the data plane on which the mapping and encapsulation processes take place, and the control plane to manage the EID-RLOC mapping system. Since LISP only defines the messages for querying data and receiving information from the mapping system, it adopts a flexible design that allows different solutions for a mapping system. The proposals to perform EID-RLOC mapping under standardization include LISP Alternative Topology (LISP-ALT) [7] and LISP Map Server (LISP-MS). LISP-ALT uses existing protocols to build an alternative topology in order to manage the mapping. LISP-MS includes MAP-Servers

that accept Map-requests from ITRs and resolve the EID-to-RLOC mapping using a database, which is filled with the authoritative EID-to-RLOC mappings provided by ETRs.

6.5 HIP Mobile Router

The HIP Mobile Router (HIP-MR) [49] is a proposal to enable network mobility for HIP-based hosts. The network comprises mobile nodes and mobile routers which perform mobility management on behalf of the mobile nodes. The delegation includes a registration in the mobile router, via a HIP extension. The MR maintains the binding state and verifies session activity between the mobile nodes and their peers. On mobility events, MR sends update messages to the peers of the mobile node, and optionally may inform the mobile node of the address change. Of course, the HIP-MR proposal focuses on HIP-aware nodes, which limits the scenarios in which it can be used.

6.6 Summary

Although there are several proposals based on the locator/identifier split idea, implementation requirements may determine, in part, the respective success of each proposal (see also Table 2). For instance, GLI-Split [27] maintains compatibility with IPv6 but requires changes to protocols like DHCP. The same applies to hiPv4 [26], which brings the benefit of the hierarchical organization of addresses. ILNP [2] is a proposal that has the merit of being applied as an end-host or end-site solution and of avoiding procedures that induce high delays in the address configuration process. Others, such as, IvIP [50] do not address mobility natively. Concerning deployment feasibility, approaches like RANGI [26], which allow incremental deployment seem to be promising. Nevertheless, care should be taken to avoid non-optimal paths, like in AIS [21], when routing scalability is one of the main concerns. Also the specification of proposals that require parallel networks to introduce benefits on a first one, such as IRON-RANGER [40] may represent a cost difficult to justify. LISP [7] is expected to decrease the size of routing tables in the core network when deployed due to the core-edge separation and the flexibility to implement the mapping system following different guidelines. Implementations for LISP are already available, such as OpenLisp (see http://gforge.info.ucl.ac.be/projects/openlisp). On the other hand, MILSA [32] requires the deployment of a network infrastructure but no implementation is available to confirm its potential benefits.

7 Outlook and Conclusions

As we have seen, in the present Internet protocol stack, multihoming may require support from all layers including applications. Of course, the decision to place the bulk of multihoming support at any particular layer comes with its own advantages and drawbacks. Typically, one resorts to the utilization of different paths according to preference sets, for instance, based on bandwidth and delay estimates. An application which supports multihoming may be better suited to control its flows with much finer granularity than what is possible, say, for example, with HIP and a set of static policies. On the other hand, in the absence of scalable source routing mechanisms, applications cannot be assured that their preferences will always be attended to, with the current crop of transport protocols. Furthermore, presently there is no standard mechanism for sharing network path information with the applications. As such, advanced applications usually employ active and passive measurement mechanisms and/or participate in overlay networks in order to obtain a better view of network performance across different paths.

From an end-host perspective we find that we lack a standard mechanism for address selections, taking into consideration upper-layer requirements, such as that real-time applications require faster paths while data applications require paths with more bandwidth. We argue that an efficient multihoming protocol cannot be coupled with a single layer, but instead it must be the result of cooperation between multiple layers, which act in a concerted manner to meet the same goals. Applications can share information, in a cross-layer fashion and enforce decisions according to their requirements via protocol APIs. Care should be taken, so that the functions belonging to a layer do not overlap with others, or that applications do not take decisions that break the functionality of layers below.

From an end-site perspective multihoming proposals should not focus only on routing scalability. Instead they should incorporate support for the diverse multihoming goals natively, rather than relying on extensions. For instance, improved resilience support should not come at the expense of mobility support. Going a step even further, the possibility of employing protocols, both as end-host and end-site solution is an approach that deserves attention. The advantages of having protocols at the end-host cooperating to achieve a goal can be extended to the network level, where different hosts in an end-site cooperate to achieve efficient multihoming support in future networks.

Acknowledgements Bruno Sousa would like to acknowledge the support of the PhD grant SFRH/BD/61256/2009 from Ministério da Ciência, Tecnologia e Ensino Superior, FCT, Portugal. This work has also been supported by CoFIMOM project PTDC/EIA-EIA/116173/2009 and TRONE project CMU-PT /RNQ/0015/2009.

References

1. Arshad, M.J., Mian, M.S.: Issues of Multihoming Implementation Using FAST TCP: A Simulation Based Analysis. Proc. IJCSNS **8**(9), 104–114 (2008)
2. Atkinson, R., Bhatti, S., Hailes, S.: ILNP: Mobility, Multihoming, Localised Addressing and Security Through Naming. Telecommunication Systems **42**, 273–291 (2009)
3. Atkinson, R., Bhatti, S., Hailes, S.: Evolving the Internet Architecture Through Naming. IEEE J. Sel. Areas Commun **28**(8), 1319–1325 (2010)
4. Bagnulo, M., Garcia-Martinez, A., Azcorra, A.: IPv6 Multihoming Support in the Mobile Internet. IEEE Wireless Commun. **14**(5), 92–98 (2007)
5. Bagnulo, M., Martinez, A.G., Azcorra, A., de Launois, C.: An Incremental Approach to IPv6 Multihoming. Computer Commun. **29**(5), 582–592 (2006)
6. Budzisz, L., Ferrús, R., Brunstrom, A., Grinnemo, K.J., Fracchia, R., Galante, G., Casadevall, F.: Towards Transport-Layer Mobility: Evolution of SCTP multihoming. Computer Commun. **31**(5), 980–998 (2008)
7. Dave, M.: The Locator Identifier Separation Protocol (LISP). The Internet Protocol Journal **11**(1), 23–36 (2008)
8. De Launois, C., Bagnulo, M.: The Paths Toward IPv6 Multihoming. Commun. Surveys Tuts **8**(2), 38–51 (2006)
9. Deleplace, A., Ernst, T., Noel, T.: Multihoming in Nested Mobile Networks with Route Optimization. In: Proc. SAINT, p. 49 (2007)
10. Dhraief, A. and Montavont, N.: Toward Mobility and Multihoming Unification-The Shim6 Protocol: A Case Study. In: Proc. WCNC, pp. 2840–2845 (2008)
11. Espi, J., Atkinson, R., Andonovic, I., Dunlop, J.: Proactive Route Optimization for Fast Mobile IPv6. In: Proc. VTC-Fall, vol. 6, pp. 1–5 (2009)
12. Fekete, G.: Network Interface Management in Mobile and Multihomed Nodes. Ph.D. thesis, University of Jyväskyla, Faculty of Information Technology (2010)
13. Fekete, G., Hämäläinen, T.: State of Host-Centric Multihoming in IP Networks. In: Proc. NTMS, pp. 1–5 (2009)
14. Fitzpatrick, J., Murphy, S., Atiquzzaman, M., Murphy, J.: Using Cross-Layer Metrics to Improve the Performance of End-to-End Handover Mechanisms. Computer Commun. **32**(15), 1600–1612 (2009)
15. Ford, A., Raiciu, C., Handley, M., Barre, S., Iyengar, J.: Architectural Guidelines for Multipath TCP Development. IETF Request For Comments: 6182 (2011)
16. Garcia-Martinez, A. and Bagnulo, Marcelo and Van Beijnum, I.: The Shim6 Architecture for IPv6 Multihoming. IEEE Commun. Mag. **48**(9), 152–157 (2010)
17. Gurtov, A.: Host Identity Protocol (HIP): Towards the Secure Mobile Internet. Wiley Series (2008)
18. Gurtov, A., Komu, M., Moskowitz, R.: Host Identity Protocol: Identifier/Locator Split for Host Mobility and Multihoming. The Internet Protocol Journal **12**(1), 27–32 (2009)
19. Habib, A., Christin, N., Chuang, J.: Taking Advantage of Multihoming with Session Layer Striping. In: Proc. INFOCOM, pp. 1–6 (2007)
20. Iyengar, J., Amer, P., Stewart, R.: Concurrent Multipath Transfer Using SCTP Multihoming Over Independent End-to-End Paths. IEEE/ACM Trans. Netw. **14**(5), 951–964 (2006)
21. Khare, V., Jen, D., Zhao, X., Liu, Y., Massey, D., Wang, L., Zhang, B., Zhang, L.: Evolution Towards Global Routing Scalability. IEEE J. Sel. Areas Commun **28**(8), 1363 –1375 (2010)
22. Kim, H., Choi, S.: A Method to Support Multiple Interfaces a Mobile Node in Next Generation Wireless Network. In: Proc. NCM, pp. 276–281 (2010)
23. Komu, M., Henderson, T.: Basic Socket Interface Extensions for Host Identity Protocol (HIP). IETF Draft: draft-ietf-hip-native-api (work in progress) (2010)
24. Kong, K.S., Lee, W., Han, Y.H., Shin, M.K., You, H.: Mobility Management for All-IP Mobile Networks: Mobile IPv6 vs. Proxy Mobile IPv6. Wireless Commun. **15**(2), 36–45 (2008)
25. Kuntz, R.: Deploying Reliable IPv6 Temporary Networks Thanks to NEMO Basic Support and Multiple Care-of Addresses Registration. In: Proc. SAINT, p. 46 (2007)
26. Li, T.: Recommendation for a Routing Architecture. IETF Request for Comments: 6115 (2011)
27. Menth, M., Hartmann, M., Klein, D.: Global Locator, Local Locator, and Identifier Split (GLI-split). Tech. Rep. 470, University of Würzburg, Institute of Computer Science (2010)
28. Mitsuya, K., Kuntz, R., Sugimoto, S., Wakikawa, R., Murai, J.: A Policy Management Framework for Flow Distribution on Multihomed End Nodes. In: Proc. MobiArch, pp. 1–7 (2007)
29. Moore, T., Pym, D., Ioannidis, C.: Internet Multi-Homing Problems: Explanations from Economics, 1 edn., chap. 5, pp. 67–78. Springer (2010)
30. Nováczki, S., Bokor, L., Jeney, G., Imre, S.: Design and Evaluation of a Novel HIP-Based Network Mobility Protocol. Journal of Networks **3**(1), 10–24 (2008)
31. de la Oliva, A., Soto, I., García-Martínez, A., Bagnulo, M., Azcorra, A.: Analytical Characterization of Failure Recovery in REAP. Computer Commun. **33**(4), 485–499 (2010)
32. Pan, J., Jain, R., Paul, S., Bowman, M., Chen, S.: Enhanced MILSA Architecture for Naming, Addressing, Routing and Security Issues in the Next Generation Internet. In: Proc. ICC, pp. 1–6 (2009)
33. Pan, J., Paul, S., Jain, R., Bowman, M.: MILSA: A Mobility and Multihoming Supporting Identifier Locator Split Architecture for Naming in the Next Generation Internet. In: Proc. GLOBECOM, pp. 1–6 (2008)
34. Pan, J.Y., Lin, J.L., Pan, K.F.: Multiple Care-of Addresses Registration and Capacity-Aware Preference on Multi-Rate Wireless Links. In: Proc. AINA, pp. 768–773 (2008)
35. Pierrel, S., Jokela, P., Melen, J., Slavov, K.: A Policy System for Simultaneous Multiaccess with Host Identity Protocol. In: Proc. ACNM, pp. 71–77 (2007)
36. Rathnayake, U., Petander, H., Ott, M., Seneviratne, A.: Protocol Support for Bulk Transfer Architecture. In: Proc. WCNIS, pp. 598–602 (2010)
37. Scharf, M., Ford, A.: MPTCP Application Interface Considerations. IETF Draft: draft-ietf-mptcp-api (work in progress) (2011)
38. Shinta, S., RyoJi, K., ToshiKane, O.: A Comparative Analysis of Multihoming Solutions. Information Processing Society of Japan (IPSJ) pp. 209–216 (2006)
39. Sousa, B., Pentikousis, K., Curado, M.: A Multiple Care of Addresses Model. In: to appear in Proc. ISCC (2011)
40. Templin, F.: The Internet Routing Overlay Network (IRON). IETF Request For Comments: 6179 (2011)

41. Thompson, N., He, G., Luo, H.: Flow Scheduling for End-Host Multihoming. In: Proc. INFOCOM, pp. 1–12 (2006)
42. Toseef, U., Udugama, A., Goerg, C., Fan, C., Pittmann, F.: Realization of Multiple Access Interface Management and Flow Mobility in IPv6. In: Proc. MOBILWARE, pp. 1–8 (2008)
43. Tse, R.: TCP Fairness in Multipath Transport Protocols. Bachelor Thesis, Brown University, Department of Computer Science (2006)
44. Tsirtsis, G., Soliman, H., Montavont, N., Giaretta, G., Kuladinithi, K.: Flow Bindings in Mobile IPv6 and Nemo Basic Support . IETF Request For Comments: 6089 (2011)
45. Ubillos, J., Xu, M., Ming, Z., Vogt, C.: Name-Based Sockets Architecture. IETF Draft: draft-ubillos-name-based-sockets-03 (work in progress) (2010)
46. Viagenie, M.B., Seite, P.: Multiple Interfaces and Provisioning Domains Problem Statement. IETF Draft: draft-ietf-mif-problem-statement (work in progress) (2011)
47. Wang, Q., Atkinson, R., Dunlop, J.: Design and Evaluation of Flow Handoff Signalling for Multihomed Mobile Nodes in Wireless Overlay Networks. Computer Networks **52**(8), 1647–1674 (2008)
48. Wei, D.X., Jin, C., Low, S.H., Hegde, S.: FAST TCP: Motivation, Architecture, Algorithms, Performance. IEEE/ACM Trans. Netw. **14**(6), 1246–1259 (2006)
49. Ylitalo, J., Melén, J., Salmela, P., Petander, H.: An Experimental Evaluation of a HIP Based Network Mobility Scheme. In: Proc. WWIC, pp. 139–151 (2008)
50. Zhang, W., Yin, X., Wu, J., Zhang, W., Huang, S.: Real Aggregation for Reducing Routing Information Base Size. JCIT **5**(6), 1–7 (2010)