


32603	Deliverable D4.5.3 Evaluation of Multihoming Solutions	
-------	--	---

Project Number:	<b>IST-2001-32603</b>
Project Title:	<b>6NET</b>
CEC Deliverable Number:	<b>32603/ULANC/DS/4.5.3/A1</b>
Contractual Date of Delivery to the CEC:	September 30 <sup>th</sup> 2004
Actual Date of Delivery to the CEC:	February 4 <sup>th</sup> 2005
Title of Deliverable:	Evaluation of Multihoming Solutions
Work package contributing to Deliverable:	WP4
Type of Deliverable*:	R
Deliverable Security Class**:	PU
Editors:	Martin Dunmore
Contributors:	Martin Dunmore, Njål T. Borch, Tim Chown, Oliver Krämer, Pekka Savola
Reviewers:	Brian Carpenter

\* Type: P - Prototype, R - Report, D - Demonstrator, O - Other

\*\* Security Class: PU- Public, PP – Restricted to other programme participants (including the Commission), RE – Restricted to a group defined by the consortium (including the Commission), CO – Confidential, only for members of the consortium (including the Commission)

**Abstract:**

This document is an evaluation of the many IPv6 multihoming solutions proposed in the IETF multi6 working group either as official working group items or as individual submissions. The various proposals are grouped into different categories according to the nature of their solution. The proposals are evaluated against all the goals for IPv6 site multihoming identified in RFC 3582.

**Keywords:**

Multihoming, evaluation, IPv6, host, site, shim, wedge, NOID, WIMP, MAST, HBA, HIP, LIN6

---

## Executive Summary

Multihoming is widely used in the IPv4 Internet today and is an essential service component for enterprises. With the introduction of IPv6 and the anticipated continued growth of the Internet, it is likely that multihoming will become increasingly common in the Internet.


However, the deployment of multihomed hosts and networks is not a straightforward task. Provider based addressing schemes (as used in CIDR and IPv6) mean that multihomed prefixes must currently be injected into the Internet's core routing tables in the 'default-free zone' (DFZ) for those networks to be reachable in all conditions. In addition, the concept of hosts changing their IP address during the lifetime of a connection is incompatible with the most popular transport protocols.

In recent years, there have been many proposals within the IETF to overcome the problems that are experienced when a site and/or host is multihomed. These proposals range from new transport protocols and sub-layers in the IP stack to new routing techniques and geographic addressing schemes.

Yet, so far, no single multihoming solution will cater for all the requirements needed for site multihoming in IPv6. It is most likely that several mechanisms will need to be used together to satisfy all the various requirements and parties concerned.

## Table of Contents

1	Introduction.....	5
2	Overview of Multihoming .....	6
2.1	Approaches to Multihoming .....	6
3	Goals for Site Multihoming .....	9
3.1	Basic Goals .....	9
3.2	Additional Goals .....	9
4	Routing Solutions.....	11
4.1	IPv6 Multihoming with Route Aggregation .....	11
4.2	Scalable Support for Multi-homed Multi-provider Connectivity .....	12
4.3	IPv6 Multihoming Support at Site Exit Routers .....	14
4.4	Multihomed Routing Domain Issues .....	15
4.4.1	Mutual Backup.....	15
4.4.2	Router Renumbering .....	16
4.5	Routing Support for IPv6 Multihoming.....	16
4.6	NAROS .....	18
4.7	Routing Header Usage .....	18
5	Proposals using Locators and Identifiers (Two Space).....	19
5.1	Layer 3 Shim Approach .....	20
5.1.1	Multiple Address Service for Transport (MAST).....	22
5.1.2	Multihoming without IP Identifiers (NOID).....	22
5.1.3	Weak Identifier Multihoming Protocol (WIMP / WIMP-F).....	23
5.2	Hashed Based Addresses .....	23
5.3	LIN6 .....	24
5.4	HIP .....	25
5.5	8+8, GSE and 16+16.....	26
6	Transport Solutions.....	27
6.1	Stream Control Transmission Protocol (SCTP).....	27
6.2	Preserving active TCP sessions.....	27
6.3	End to End Multihoming .....	28
7	Site Exit Router and Host Behaviour.....	29
7.1	Host-Centric Multihoming.....	29
8	Conclusions.....	31

32603	Deliverable D4.5.3 Evaluation of Multihoming Solutions	
-------	--	---

---

References.....	33
Abbreviations.....	36

---

## 1 Introduction

Multihoming is widely used in the IPv4 Internet today and is an essential component of service for enterprises. In recognition of this, the built-in features of IPv6 make it easier for end-hosts and networks to be multihomed than in IPv4. This, combined with the expected continued growth of the Internet, means that it is likely that multihoming will become an increasingly common phenomenon.

Unfortunately, the deployment of multihomed hosts and networks is not a straightforward task. The IPv6 aggregatable global unicast address format [13] is based on the aggregation of provider address blocks rather than geographical ones. This means that, currently, multihomed prefixes need to be injected into the routing tables of the Internet's default-free zone (DFZ). This has resulted in a substantial increase in the size of BGP tables in the core of the Internet [33]. In addition, the possibility of changing destination and/or source IP addresses during the lifetime of a connection, is incompatible with the most widely used transport protocols like TCP and UDP.

Recently, there have been numerous proposals made within the IETF to allow IPv6 multihoming to prosper without incurring the associated scalability and transport problems. The following section gives a brief overview of what multihoming is, the motivations for being multihomed and classifies the various approaches to multihoming. Section 3 describes the goals of IPv6 site multihoming upon which the various multihoming solutions are evaluated. The subsequent sections then describe the various solutions according to their classification: routing, two space, transport and site/host behaviour. Finally, the proposals are evaluated and conclusions drawn in section 8.

## 2 Overview of Multihoming

There are typically two types of multihoming: host multihoming and site multihoming<sup>1</sup>. A multihomed host is a host with more than one global IP address assigned to it. These addresses could be from the same Internet Service Provider (ISP) or from different ISPs. Furthermore, a host may be multihomed with a single interface (multiple IP addresses on one interface) or may have one or more global IP addresses (from different subnets) on several interfaces.

Site multihoming refers to a site that has more than one connection to the public Internet through either the same or different Internet Service Providers (ISPs)

There are several reasons why a site or an end host may wish to be multihomed:

- *Fault resilience/redundancy.* If the link to one ISP fails, a different link can be used to carry the traffic.
- *Load Balancing.* To achieve higher throughput a host/site could spread its traffic load over two ISPs.
- *Service Value/Policy.* With some particular services (e.g. VoIP or VoD) certain ISPs may offer a cheaper price than others. Thus, a site/host may wish to allocate traffic types to different ISPs to save on traffic costs.

IPv6 is more suited than IPv4 for deploying multihomed hosts since it is more able to dynamically detect and inherit multiple addresses. When an IPv6 host connects to a network it receives router advertisement messages from all IPv6 routers attached to the LAN. Using stateless address autoconfiguration [14], an IPv6 host can automatically assign a globally unique IPv6 address for each different network prefix it receives through router advertisements. These advertisements may be received on one interface (thus having multiple addresses on single interface) or on several interfaces (one or more addresses per interface). Thus, an IPv6 host can automatically configure itself as a multihomed host when it boots up.

To further understand how to implement site multihoming in IPv6, an IETF working group, ‘multi6’ [9], was established. This report concentrates on multihoming solutions that have been proposed through this working group, either as official working group drafts or as personal contributions.

### 2.1 Approaches to Multihoming

Without doubt the most simple solution for IPv6 multihoming would be to adopt the long preferred approach used for multihoming in IPv4. In the IPv4 multihoming approach, a site’s local prefixes are announced as distinct routing prefixes into the inter-domain routing system. These routing prefixes are propagated to the top level hierarchy of the routing system known as the default free zone (DFZ).

An important feature of this approach is that each site has a Provider Independent (PI) address space. That is, the site owns the address prefix so that the provision of transit connectivity by an ISP and the allocation of a suitable network prefix are decoupled. Thus, the site’s address prefix bears no relation to the addressing used by its transit providers and cannot be aggregated into a transit provider’s prefix.

Each transit provider connected to the multihomed site accepts a routing advertisement for the site’s network prefix and advertises this prefix into the inter-domain routing system via BGP. When a particular path via a transit provider becomes unavailable the BGP routing systems ensures that this routing advertisement will be withdrawn from the routing system thus allowing upstream routers to choose one of the alternative routes. How efficient this process is during a re-homing event depends on the convergence time of BGP.


---

<sup>1</sup> There is also a third type of multihoming: ISP/operator multihoming, although this is a trivial case since address prefixes are large enough so that there are no restrictions on route advertisements in the DFZ.

This IPv4 approach to multihoming satisfies most of the multihoming goals laid down in RFC 3582. However, this approach is completely non-scalable. As the number of multihomed sites in the Internet grows, the number of routing prefixes that are injected into the global routing system (i.e. the DFZ) increases linearly. The common opinion is that, even with modern router hardware, this will lead to an unacceptable number of routing prefixes to manage in the DFZ of the IPv6 Internet.

Therefore other solutions that will satisfy the goals of multihoming in IPv6 without having the scalability problems inherent in the IPv4 approach have been proposed in recent times. These solutions can be roughly classified as follows:

- **IPv6 Routing Solutions.** This class of IPv6 multihoming solutions uses the IPv6 routing system like the IPv4 approach but also adds mechanisms or procedures to alleviate the scalability problem of injecting prefixes into the DFZ. They also consider the use of provider aggregatable (PA) addressing rather than PI based addressing since PA addressing is the standard way in which sites will obtain their address blocks in the IPv6 Internet.
- **Identifier and Locator (Two Space) Solutions.** All multihoming scenarios have a common feature. That is, a single entity (a site or host) can be reached in a number of ways, i.e. it has >1 logical location. The general concept with two space solutions is that there is a separation between the identity of a node (denoted by an identifier) and its location in the Internet (denoted by a locator). This can be done in one of two ways. The structure of a node's IPv6 address can be manipulated to provide both identity and location information in the same 128 bits or alternatively, to provide a mapping at some point in a network (or even the end host) where the usual IPv6 address is associated with a locator. A common theme of these solutions is that at some point in the network or host stack there is a mapping function between an identifier and a set of locators. Transport layer protocols and applications will generally only be aware of the identifiers and not the locator sets.
- **Mobility Solutions.** One can view mobility as a special case of multihoming. When a host moves in the Internet and acquires a new address at its new location this is analogous to a re-homing event where the host's primary provider has become unreachable and must switch to using an address corresponding to one of its other providers. In Mobile IPv6 [2] the change in addresses as a host moves is hidden from the transport layer and ULPs. A mobile node (MN) always uses its home address (HoA) as far as transport protocols and applications are concerned. When a MN acquires a new address, known as a care-of address (CoA) because of movement, it uses binding update (BU) messages to inform its correspondent node (CN) about the new address. The CN holds a mapping between the MN's HoA and its new CoA and can pass up the original connection semantics to the upper layers. In a multihoming context, this procedure could theoretically be used to inform other hosts when a multihomed host experiences a re-homing event. However, the current authentication mechanism for the BU messages is incompatible for a multihoming solution. The authentication mechanism (known as return routability) involves verifying the new CoA via the MN's home agent. However, in a multihoming scenario the re-homing event means that the primary network is unavailable and therefore any home agent for this network would also be unreachable. Therefore for MIPv6 to be used as a multihoming solution either the return routability mechanism needs to be changed (or not used at all) or some other workaround involving multiple home agents across provider networks needs to be engineered.
- **Transport Solutions.** Traditional transport and other upper layer protocols (ULPs) do not support the notion of multihoming. Changing an IPv6 address during the lifetime of a connection will break the semantics of that connection in common transport protocols and other ULPs such as UDP, TCP, FTP, HTTP etc. If the IPv6 address change is not hidden from the transport layer as with mobility or two space solutions, then the support for address changes can be added to the transport protocols.
- **Site Exit Router and Host Behaviour.** There are some multihoming solutions that can be achieved by modifying the behaviour of a site's exit routers and/ or end hosts. The idea here is to be able to have some multihoming support without the need for new protocols to be defined nor new code to be installed on the site exit routers and/or end hosts. One example is to have some form of packet

32603	Deliverable D4.5.3 Evaluation of Multihoming Solutions	
-------	--	---

---

header manipulation in the site exit routers corresponding to the outgoing/incoming transit provider. In this way the end hosts will be unaware that they are multihomed. This would be accomplished by existing configuration options on the routers themselves rather than any new code implementations.

### 3 Goals for Site Multihoming


The goals for site multihoming, both basic goals and additional goals are specified in RFC 3582 [31].

#### 3.1 Basic Goals

- R1 - **Redundancy.** The multihoming architecture should allow for a site to remain connected to the Internet in the event of failure of one or more of its provider networks. Hosts within the site should be able to communicate with other hosts in the global internet provided at least one of the site's provider networks is operational.
- R2 - **Load Sharing.** A site should be able to distribute both its inbound and outbound traffic between multiple network providers in a concurrent manner.
- R3 - **Performance.** The multihoming architecture should allow a site to avoid performance problems that could occur on links directly *between* its provider networks.
- R4 - **Policy.** A site should be able to multihome and distribute its inbound and outbound traffic based on its own policies (e.g. use ISP A for all SMTP traffic or use ISP B between the hours 23:00-07:00).
- R5 - **Simplicity.** Any IPv6 multihoming architecture should not be substantially more complex to deploy and operate than the current multihoming practices for IPv4 as described in [ref].
- R6 - **Transport layer survivability.** Transport layer and other ULPs should be able to survive re-homing events. When a re-homing events occurs the applications on the communicating end hosts should experience no greater disruption than would be inherently associated with the event occurrence (i.e. packet delay and/or loss).
- R7 - **Impact on DNS.** The multihoming architecture must be compatible with the current DNS system or have little impact (i.e. be readily deployable) if modifications required to support the architecture are needed.
- R8 - **Packet filtering.** The multihoming architecture should not prevent the filtering of packets with forged source IP addresses or other inappropriate IP addresses at any administrative boundary in the Internet.

#### 3.2 Additional Goals

- R9 - **Scalability.** Unlike the current practice of multihoming in the IPv4 Internet, the multihoming architecture in the IPv6 Internet must allow for a greatly increased number of multihomed sites without imposing unreasonable and inefficient overheads on the routing system.
- R10 - **Impact on Routers.** Any changes needed to router implementations in order to deploy the multihoming architecture must be minimal or consist of logically separate functions to those that exist already. Furthermore, any changes must not affect normal single-homed operation and any modified routers must be interoperable with unmodified hosts and routers.
- R11 - **Impact on Hosts.** The multihoming architecture must not compromise a host being able to operate according to the core IPv6 specifications. Any modifications to the host stack must be minor or consist of logically separate functions to those that exist already.
- R12 - **Interaction between hosts and routing system.** Any interaction needed between multihomed hosts and the routing system should be simple, scalable and securable.

32603	Deliverable D4.5.3 Evaluation of Multihoming Solutions	
-------	--	---

---

R13 - **Operations and management.** Staff responsible for the operation of the site network should be able to monitor and configure the site's multihoming system.

R14 - **Cooperation between transit providers.** The multihoming solution should not require cooperation between the different network providers offering connectivity to the site.

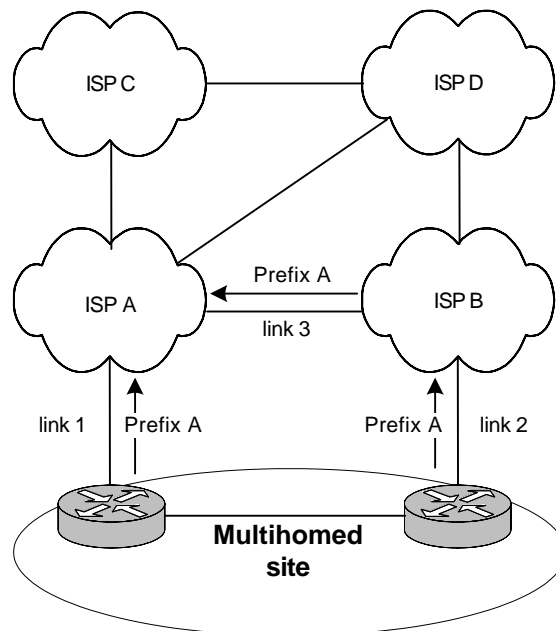
R15 - **Multiple solutions.** Multihoming solutions should attempt to cover as many multihoming scenarios as possible in order to minimise the need to deploy multiple solutions within one multihoming architecture.

R16 - **Security.** A site adopting an IPv6 multihoming solution should not become more vulnerable to security breaches than it would be as a IPv4 multihomed site. Also, any changes introduced by the multihoming architecture should not cause single-homed sites to become more vulnerable to security breaches.

## 4 Routing Solutions

### 4.1 IPv6 Multihoming with Route Aggregation

The Internet Draft “IPv6 Multihoming with Route Aggregation” [23], describes a routing scheme that supports IPv6 site multihoming. A site that is multihomed to two or more ISPs will designate one of its ISPs as its primary ISP and receive IPv6 address assignment from the primary ISP’s address block. In the example in Figure 1, ISP A is chosen as the primary ISP for the multihomed site and assigns prefix A from its address block to the multihomed site.



**Figure 1 IPv6 Multihoming with Route Aggregation**

The multihomed site will advertise prefix A to all of its connected ISPs, in this case ISP A and ISP B. ISP B will propagate the advertisement of prefix A to ISP A, but to no other ISP. ISP A will advertise its own aggregation into the Internet. Thus, traffic destined for prefix A will be routed to ISP A. ISP A will then forward the traffic either directly to the site (link 1) or to ISP B (link 3) according to the specific routing policy implemented. Load balancing may be achieved by having ISP A forward traffic for prefix A on both link 1 and link 3 (differentiated by, for example, traffic type). Similarly, traffic originating from the multihomed site may be sent on either link 1 or link 2 according to the site’s requirements.

If the site chooses to multihome for redundancy purposes, all of its traffic will be routed via its primary ISP, ISP A, should the connection to ISP B (link 2) fail. Conversely, should the connection to ISP A (link 1) be broken, all of the site’s traffic will be routed via ISP B. This can be achieved because ISP A will know from the route advertisements it receives from ISP B that traffic for prefix A is reachable via ISP B.

Note that with this scheme, the specific routing prefix associated with the multihomed site is only known by the directly connected ISPs (A and B in the example) and is never propagated to the rest of the Internet.

Thus, it has good scaling properties for the global Internet since the DFZ will not contain multihomed site prefixes. However, this scheme does rely on the primary ISP having links to the secondary ISP(s). If this is not the case, the site prefix would need to be advertised along the path between the primary and secondary ISPs, which could conceivably include the DFZ of the Internet.

A further drawback with this scheme is that the primary ISP is a single point of failure. Since the secondary ISPs do not inject the specific prefix of the multihomed site into the DFZ, the site will become unreachable should its primary ISP fail (that is, the primary ISP itself, rather than the link to the primary ISP). Also, the scheme does not cater for the multihomed site having multiple address prefixes assigned by different ISPs.

## 4.2 Scalable Support for Multi-homed Multi-provider Connectivity

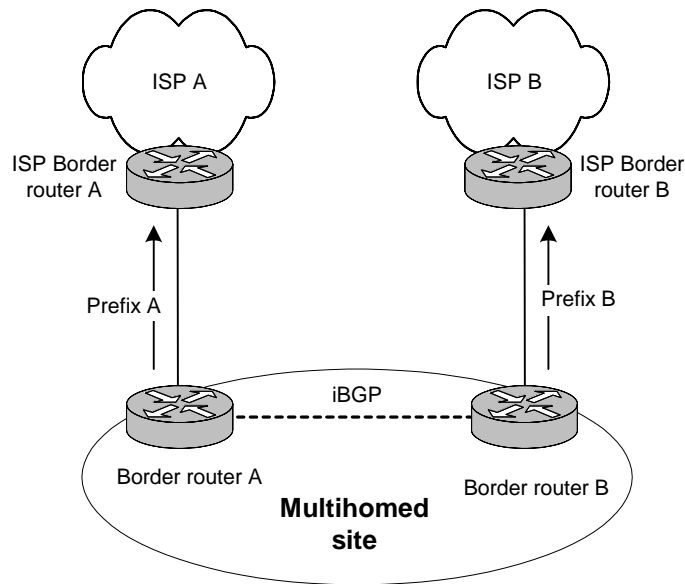
RFC 2260 [29], describes several strategies for achieving scalable site multihoming. Although the document is not specific to IPv6, the strategies it describes are applicable to IPv6 as well as IPv4. The multihoming strategies in RFC 2260 are aimed at removing (or minimising) the need for routes to be held in the DFZ of the Internet and to minimise the amount of coordination needed between ISPs.

The fundamentals of the RFC 2260 method are illustrated in Figure 2 and Figure 3. A multihomed site is connected to ISPs A and B via border routers A and B respectively. Under normal conditions, i.e. the connections to both ISPs are active, each border router advertises, to its connected ISP, the prefix that was allocated by that ISP. Thus, in Figure 2, border router A advertises prefix A to ISP A's border router. Similarly, border router B advertises prefix B to ISP B's border router. Since the prefixes advertised by each of the site's border routers are aggregated by the respective ISP, no additional routes are injected into the default-free zone of the Internet. However, if the link to one ISP fails, the site will not be reachable on that respective prefix.

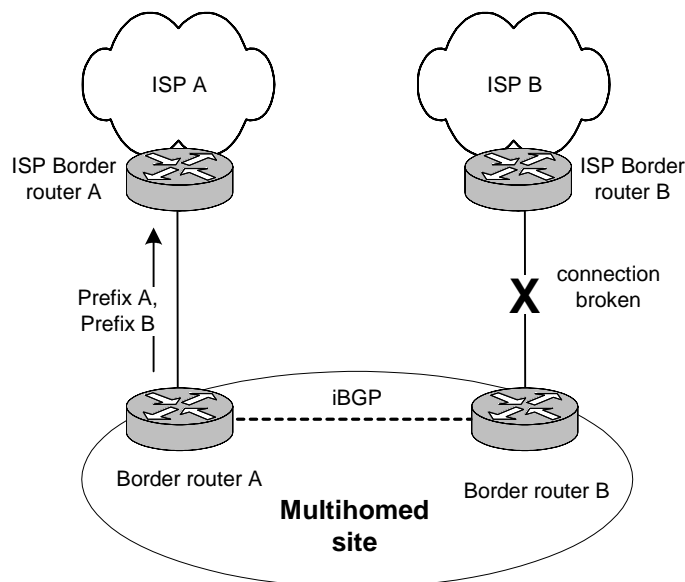
To get around this problem, RFC 2260 proposes that when a site's border router detects link failure between one or more of the site's other border routers and their respective ISPs, it should advertise the prefixes allocated by these ISPs. As an example, in Figure 3, when the link between the site's border router B and ISP B fails, border router A begins to advertise reachability of prefix B to ISP A. One way to achieve this would be to have an iBGP peering between the site's border routers. Thus, the absence of prefix B being advertised by border router B (due to the link failure) can trigger border router A to advertise prefix B to ISP A.

In order for traffic destined to prefix B to be able to reach its destination, ISP A would need to inject this prefix into the DFZ of the Internet. Although injecting site prefixes into the DFZ is undesirable, RFC 2260 argues that this method (known as *auto-route injection*), only results in site prefixes being injected into the DFZ temporarily whilst a connection between the multihomed site and one of its ISPs is down. Thus, the expected average number of site prefixes injected into the DFZ at any given time would be a small percentage of the total number of multihomed prefixes in the entire Internet.

This is true for sites choosing to multihome for reasons of redundancy. In the examples given, there is no reason for prefix A to be reachable via ISP B (or prefix B to be reachable via ISP A) unless there is a link failure. However, if the site chooses to multihome for other reasons (e.g. for traffic engineering purposes) there may well be a requirement for prefixes A and B to be reachable via both ISPs. In this case, both prefixes would need to be injected into the DFZ to be reachable via the non-aggregating ISP (i.e. the ISP not owning the prefix).

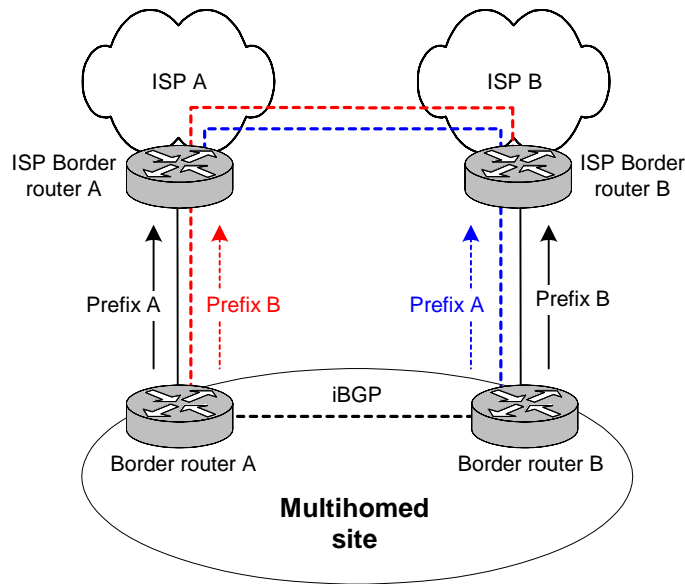


**Figure 2 Non-empty intersection**



**Figure 3 Empty intersection**

To achieve greater scalability, RFC 2260 suggests an improvement that eliminates the need for any multihomed site prefixes to be injected into the Internet's DFZ. Quite simply, each of a site's border routers maintains BGP peerings with all the ISPs to which the site is connected. Thus, as illustrated in Figure 4, in addition to direct BGP peerings with their directly-connected ISPs, border routers A and B maintain non-direct BGP peerings with ISPs B and A respectively.



**Figure 4 Non-direct peerings**

The site's border router A would advertise prefix A on its direct eBGP peering with ISP A, and would advertise prefix B on its non-direct eBGP peering with ISP B. Similarly, the site's border router B would advertise prefix B on its direct eBGP peering with ISP B, and would advertise prefix A on its non-direct eBGP peering with ISP A.

Within every border router (both at the site and at the ISPs), routes received over direct peerings have greater preference to routes received over non-direct peerings. Thus, in normal circumstances, traffic destined for prefix A is routed through ISP A and traffic destined for prefix B is routed via ISP B. If, for example, the link between the site's border router B and ISP B were to fail, ISP B would then use its less-preferred route, obtained via its non-direct peering. Forwarding along a route received from a non-direct peering is performed via tunnelling. Thus, ISP B's border router would encapsulate traffic destined for prefix B and tunnel it to the site's border router A, which would then decapsulate the traffic and send it to border router B.

Using this technique means that site prefixes are never injected into the DFZ of the Internet. It also has the advantage that all of the site's prefixes are reachable via any of its connected ISPs. However, this solution is still geared towards redundancy and does not facilitate traffic engineering requirements (non-direct routes are only used when the direct routes are not available). Furthermore, when non-direct routes are used, poor traffic performance may be experienced due to the sub-optimal nature of the tunnelled routes. Sub-optimal routing may also occur during normal circumstances. Imagine another site directly connected to ISP A, and sending traffic to a host containing prefix B of the multihomed site. Normal operation would see this traffic being routed to ISP B from ISP A (possibly traversing numerous hops) whereas the optimal route would be via border router A of the multihomed site. One way around this would be to allow ISP A to receive advertisements for prefix B from border router A, and distribute this to its customer sites (but not to its transit peers).

### 4.3 IPv6 Multihoming Support at Site Exit Routers

RFC 3178 [30], describes multihoming support for IPv6 site exit routers. This document is simply a more detailed description, for IPv6, of the method described in RFC 2260 that uses non-direct peerings. This method can be implemented for IPv6 by using BGP4+. The non-direct peerings themselves would be

implemented via IPv6-over-IPv6 tunnels. However, in some circumstances (such as when IPv6 connectivity is achieved via IPv6-over-IPv4 tunnels) it may be more appropriate to use IPv6-over-IPv4 tunnels.

Since IPv6 allows hosts to have multiple addresses, hosts within the multihomed site may be assigned addresses for each ISP the site is connected to. In this case, some sort of framework for source address selection (such as that described in [25]) needs to be in place. The problem of ingress filtering by ISP routers can be resolved using a variety of techniques as discussed in section 7.1.

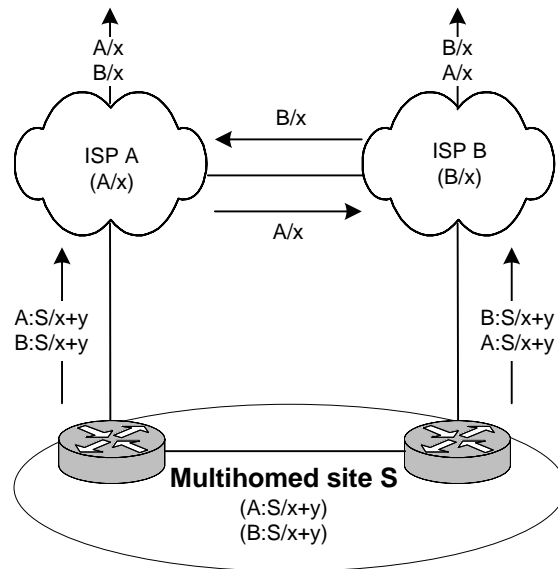
## 4.4 Multihomed Routing Domain Issues

This Internet Draft [12] discusses some of the issues for multihomed routing domains that use the aggregatable addressing and routing scheme [13], and proposes some solutions.

### 4.4.1 Mutual Backup

The ‘mutual backup’ scheme is proposed to address the transparency issue. The fact that its customer is multihomed to a different ISP should be transparent to the provider. However, in order for the site’s prefixes to be reachable through both providers, each provider must advertise reachability of both prefixes. Obviously, this breaks the transparency requirement and has poor scaling properties since the prefixes are injected into the DFZ.

The mutual backup scheme supports a scenario where two providers have an agreement to provide backup for each other.



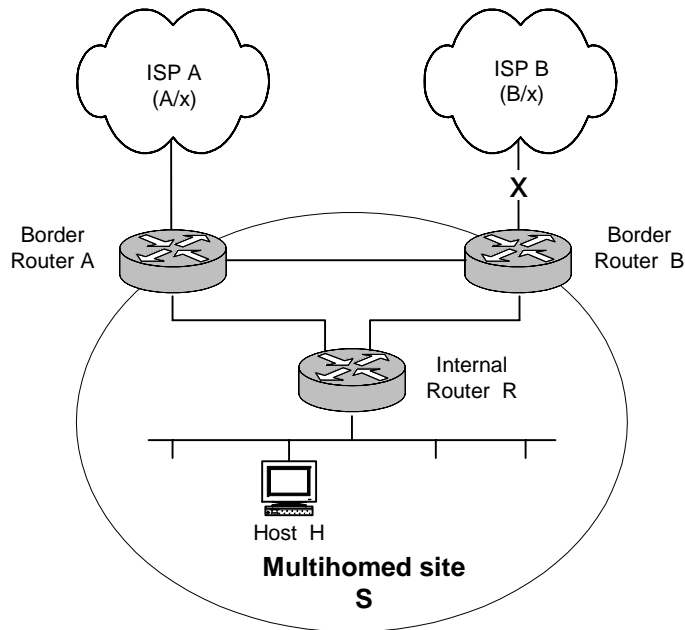
**Figure 5 Mutual Backup**

Traffic from the Internet destined for  $A:S/x+y$  will be routed to ISP A. At the level of ISP A and B (or above), the route advertisement for  $A/x$  from ISP A is preferred to that of ISP B since it is direct (one element in the path from A compared to two elements from B). Conversely, traffic destined for  $B:S/x+y$  will be routed to ISP B. Should the path through ISP A fail, all of the traffic for the multihomed site S ( $A:S/x+y$  and  $B:S/x+y$ ) will go through ISP B.

Although this scheme works, and does not inject long prefixes into the DFZ, it does rely on ISPs A and B having a direct connection to each other and consenting to the backup agreement.

#### 4.4.2 Router Renumbering

The Internet Draft proposes to use router renumbering [15] to force hosts inside the multihomed site to choose the correct source address when making new connections.



**Figure 6 Router Renumbering**

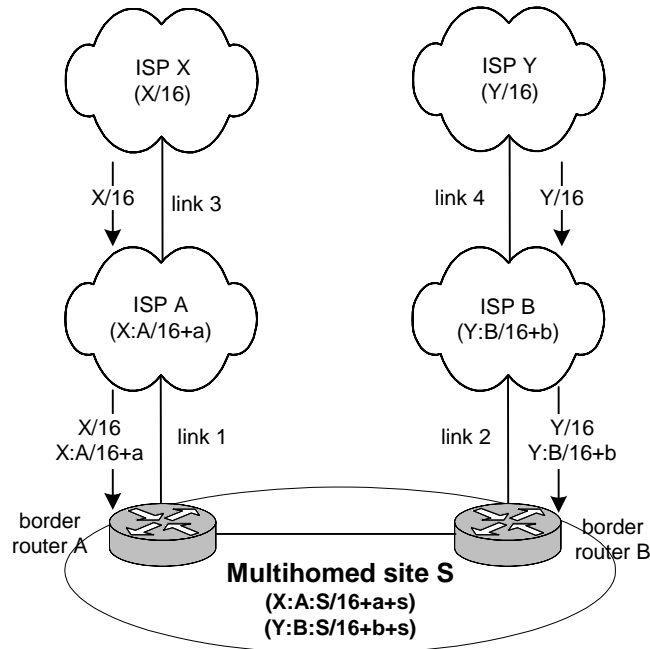
Referring to Figure 6, host H will have two global addresses, A:S:H and B:S:H, which it learns from the router advertisements sent by the site's internal router R. Host H is thus able to choose either A:S:H or B:S:H as its source address when making new connections. However, if the site's connection to ISP B were to fail (as shown in Figure 6), the host H should be made aware of this and thus form new connections using A:S:H as the source address. The proposal is for the site's border router (in this case border router B) to notify the site's internal routers of the broken path via the router renumbering protocol when it is detected. When the site's internal routers (in this case router R) issue their router advertisements, the 'preferred lifetime' of all the prefixes associated with the broken path are set to zero. The effect is to deprecate the associated source address so that the hosts do not use them to form new connections (and will thus choose alternative source addresses).

Of course, this will not cater for existing connections that are using the deprecated source address when link failure occurs.

#### 4.5 Routing Support for IPv6 Multihoming

Following on from the router renumbering technique just described, an alternative method for hosts to choose their source address, in light of network conditions, is presented in [16]. In this Internet Draft the author describes the possibility of a host within a multihomed site unwittingly selecting a source address that is unreachable on its return path due to link failure on the path associated with the source address. The solution fundamentally involves each ISP (more specifically, each AS) advertising its own prefix downwards, through lower-level aggregators towards the border router of the multihomed site. Furthermore,

each prefix received from upper-level aggregators are also propagated downwards towards the border router of the multihomed site. The example in Figure 7 illustrates this procedure.



**Figure 7 Routing Support for IPv6 Multihoming**

The site S is multihomed to ISPs A and B, who in turn connect to their upper level aggregators, ISPs X and Y respectively. The routing information is propagated as follows:

- ISP X advertises the prefix X/16 to ISP A
- ISP A propagates X/16 and advertises X:A/16+a to site S
- ISP Y advertises the prefix Y/16 to ISP B
- ISP B propagates Y/16 and advertises Y:B/16+b to site S.

Thus, a site's border router can detect failures in the upstream path by the absence of the relevant prefix. For example, a failure of link 3 would result in the absence of the X/16 prefix advertisement to the site's border router A. Normally, border router A would have no indication of this and hosts within the site could continue to choose source addresses derived from X. If the prefix advertisements are distributed within the site, internal routers will know that the Internet is unreachable through ISP A. This information can be disseminated to the hosts (e.g. by extending the prefix option semantics in router advertisements) and used to select the correct source address for outgoing connections. Furthermore, the hosts can still select source addresses based on a faulted prefix, if the destination address matches a prefix longer than the faulted prefix, but which is still active. As an example, if link 3 fails, a host in the site can still choose a source address based on X if the destination address matches the X:A/16+a prefix. In other words, the destination address is 'below' the link failure.

## 4.6 NAROS

NAROS meaning “Name, Address and ROute System” [38], is a host-centric multihoming solution for IPv6 that enables sites to engineer their incoming and outgoing inter-domain traffic without the need to manipulate BGP routing messages. It relies on using several IPv6 addresses per end host, one address for each upstream provider. The basic idea of the solution is that host must contact the NAROS service to determine the best IPv6 source address to use in order to reach a given destination address.

Since it is the NAROS service that essentially determines the site exit router to be used in a communication exchange, there is no need for the different ISPs of a multihomed site to cooperate with each other and exchange reachability information or for the site exit routers to tunnel to each ‘secondary’ ISP. For example, in a site that is multihomed to three ISPs, ISPA, ISPB and ISPC via site exit routers RA, RB and RC respectively, ISPA only receives outgoing traffic from RA using source addresses based on the prefix advertised by ISPA to RA. The same scenario exists for the other ISPs and routers.

When an end host wishes to communicate with another IPv6 host, it needs to determine the best source address to use from an available set of addresses that it owns. With NAROS a host will query a NAROS server with a request message. This message requests which source address a host should use for a given destination address. The NAROS server responds with a response message that contains the source address that the host should use in communication with the destination address.

Note that the NAROS service only gives a hint as to what source address should be chosen. It is not intended to preserve traffic flows across address changes. However, it could be combined with other mechanisms such as SCTP, HIP or MIPv6 in order to preserve flows when there is an address change.

The NAROS service is independent from any other service and the server does not maintain state about the internal hosts. It is possible to deploy several servers for redundancy or load balancing purposes or utilise anycast addresses. Although it is independent, integrating the NAROS service with the DNS service could promise a more efficient solution. Normally a host would choose destination addresses received from the DNS one at a time until communication was successful. By integrating both the services, a host could learn the best source address and destination address pair to use in one query exchange.

Since the source address chosen by a host equates to the outgoing ISP for the traffic, the NAROS service could be a useful tool for performing traffic engineering within a multihomed site. For example, the NAROS server(s) could implement load balancing in a simple round robin fashion amongst the available ISPs, or could even be combined with a other requirements such as Quality of Service, traffic cost etc.

## 4.7 Routing Header Usage

A technique whereby a multihomed site can force routing via a specific ISP is suggested in [34] and [11]. This technique consists of inserting a routing header containing an intermediate anycast address that identifies the routers of the intended ISP. Thus, the packets containing such a routing header will be explicitly routed to the selected ISP. However, this method has the drawback that the ISP must process the routing headers; a possible expensive overhead especially if many sites were to employ this technique.

An alternative to this would be to use site exit router selection. Instead of the routing header containing the anycast address for an ISP, it would contain an address specifying a site-exit router that connects to the desired ISP. If only one site exit router is connected to the desired ISP, the address in the routing header would be a site-local address of one of the router interfaces. If the desired ISP is served by multiple site exit routers, the address in the routing header would be an anycast address. This method has the advantage that the routing header is processed by the site exit routers and not by the ISP routers, thus demonstrating greater scaling properties. It also means that the ISP’s co-operation in implementing all routers anycast does not have to be sought by the customer site.

## 5 Proposals using Locators and Identifiers (Two Space)

In the current IP architecture, IP addresses have multiple personalities. Depending on your point of view an IP address can be viewed as

- an identifier for a specific interface
- an identifier for a specific service
- the locator for a specific interface

We know that a host may have multiple interfaces. In IPv4 this situation is somewhat simplified as the architecture permits only one IP address per interface. However, in IPv6 the situation is more complex as an interface may have multiple addresses, not just a mixture of different scoped addresses (e.g. global, link-local etc) but also multiple global addresses with (potentially) different network prefixes.

Thus if we imagine an interface located somewhere in the IPv4 Internet, there is little problem for the interface's IP address to have the dual role of both identifier and locator since there is a one-to-one relationship between the two. Thus an IPv4 interface with the address 192.12.13.14 will be identified as such and can only be located at that address.

The same interface in the IPv6 Internet may have several IPv6 addresses associated with it and since the current architecture allows each IP address to be both locator and identifier the interface can thus have multiple identities and multiple locations<sup>1</sup>. Unfortunately, the current IPv6 architecture doesn't handle this situation effectively. As an example, consider an IPv6 interface located on a site network that is triple-homed and receives three separate /64 prefixes, say 2001:1234::/64, 2001:5678::/64 and 2001:abcd::/64 thus giving three separate globally routable IPv6 addresses. When an application on a remote host wishes to contact the interface in question it will query the DNS to resolve its name (e.g. test-interface.foo.net) to one or more IPv6 addresses. Assuming all of the addresses of test-interface.foo.net have been entered in the DNS, the remote application will learn these from the DNS response to its query. Where the current architecture fails is that, as far as the remote application is concerned, the interface that it is trying to connect to has three separate identities and three possible locations. The application has no guarantee that:

- all these addresses actually refer to the same interface
- the interface is reachable (i.e. located) at all of these addresses

It may at first seem odd why the addresses returned by the DNS may not actually refer to the same interface. However, there is no restriction that multiple addresses associated with a FQDN entered in the DNS must refer to the same interface. The multiple addresses could just as easily refer to separate interfaces, perhaps even located on separate hosts in order to satisfy load balancing and/or redundancy requirements of popular servers (e.g. think of www.google.com).

It is a common misconception that the DNS performs the role of uniquely identifying endpoints in the Internet. The DNS simply resolves FQDNs to IP addresses as best it can. There is no guarantee that the FQDN is globally unique (different DNS servers may map different addresses for the same FQDN) nor any guarantee that the resolved addresses refer to the same entity in the network. Although 'entity' is in the eye of the beholder, the intention may be for it to mean a host, an interface or a service.

What is needed therefore, is a separation of identity and location. We need to be able to uniquely identify an endpoint in the Internet.

In August 2004 the multi6 working group formed a design team to refine the idea of a shim/wedge layer that would map between the context of location and identity for IPv6 addresses. The design team has proposed a

---

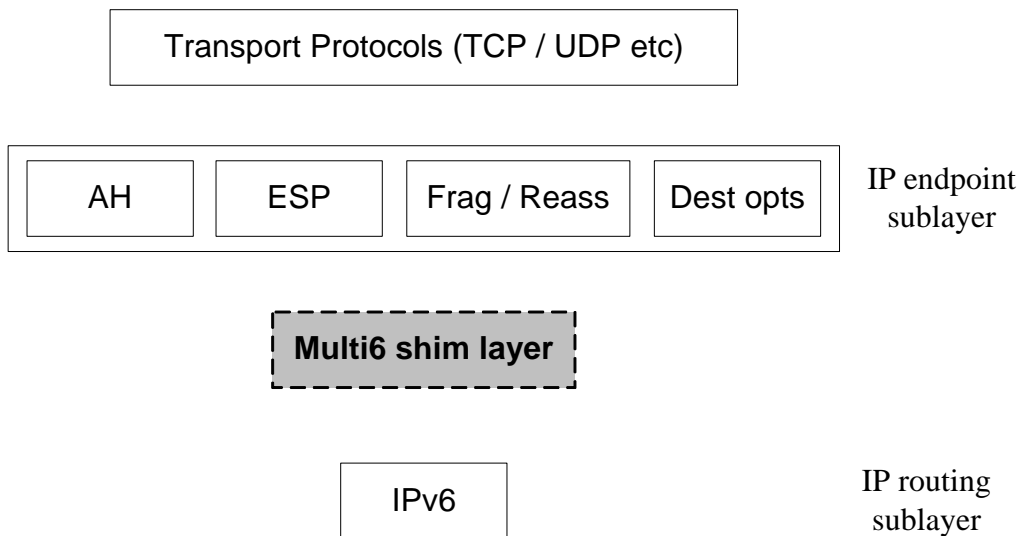
<sup>1</sup> The actual geographic location of the interface will of course be the same for each IP address but the topological locations will be different.

layer 3 shim approach described in the following section which adopts ideas from earlier proposals aimed at solving the multihoming problem. Other two-space proposals such as LIN6 and HIP which are not specifically intended for multihoming but nevertheless may be applied to the problem space, are also described.

## 5.1 Layer 3 Shim Approach

The layer 3 shim approach presented in [ref] is an abstraction of earlier proposals that employed a shim/wedge layer, notably MAST, NOID and WIMP (described in subsequent sections). It adopts the finer points of their architectures whilst assuming that details of specific mechanisms such as preventing redirection and DoS attacks are specified elsewhere.

The central theme of the layer 3 shim approach is that a new ‘shim’ sub-layer is inserted into the IP stack in end hosts that wish to take advantage of multihoming. This shim layer is responsible for performing the mapping function between the single identifier context presented to the transport and upper layer protocols (referred to as the upper layer ID or ‘ULID’) and the set of locators that can be used between the communicating hosts in a single communication session or exchange. Rather than introducing a new namespace to serve as identifiers, the layer 3 shim approach instead uses one of the locators from the locator set and designates this as the ULID. Thus, while the different locators used in a multihomed communication session (and thus appearing in the IPv6 header) may change over time, the locators used as the ULIDs will remain constant. In practice a multihomed host will assign the source ULID to one of its locators according to the default source address selection procedures specified in RFC 3484 [ref]. The destination ULID will be assigned to the first locator obtained from the DNS via a FQDN lookup of the destination host.



**Figure 8 The multi6 shim layer**

Using one of the locators as the ULID has the advantage that it supports callbacks and referrals in long lived communication sessions, since both the 128-bit locator address and the FQDN held in the DNS serve as handles for the upper layer protocols and applications.

The placement of the shim sub-layer within the IPv6 stack is illustrated in Figure 8. It is located between the IP layer and the transport/upper layers. Conceptually, the shim layer behaves as if it were an extension header in the IPv6 header where it would be placed before all the other extension headers including AH, ESP, fragmentation and destination options headers. By placing the shim layer beneath the AH and ESP

layers, IPsec can be made unaware of changes to the locators in the same way the transport and ULPs will be unaware. Thus, IPsec security associations will still be valid across locator changes. Note that this is a *conceptual* next header. The shim layer does not necessarily need to insert a next header (although that is one option it can choose) as will be explained below.

The transport and application layers located above the shim layer, only ever see the ULIDs. The shim layer maintains context state for each ULID pair (i.e. source and destination ULIDs) in order to perform the mapping between locators and their associated ULIDs. This mapping is performed at both the sending and receiving hosts in a consistent manner. Thus from an ULP or applications' point of view, packets are sent end-to-end using the ULIDs, when in reality the packets be sent using any of the locators associated with the ULIDs.

In order for the sender and receiver hosts to learn each others locator sets, a context establishment exchange needs to take place between the sender and receiver. In the layer 3 shim proposal this context establishment occurs asynchronously to the packet exchange between the two hosts. Until this context establishment has occurred, the hosts are unable to switch between their locators and packet exchange operates in the usual way for unmodified, single-homed IPv6 hosts. Precisely at which point a host should initiate the context establishment procedure is an open issue. It could be on a per application/protocol basis or use a heuristic such as when a certain threshold of transmitted packets has been reached.

The layer 3 shim proposal does not specify the details of the context establishment procedure. Suffice to say that it is only after context establishment has taken place when the communicating hosts can switch between locators. In theory a host should be able to switch to using a different locator pair without the need of any signalling or other form of notification. However, it may be prudent for the host to verify the new locator pair before using it.

The context state that maps between ULID pairs and their matching locator pairs is held at both sender and receiver hosts. This context state is consistent at both ends yet is accessed differently in the send and receive paths. On the sending side, when the ULP passes down a packet to be transmitted it is the ULID pair, i.e. the tuple <ULID(src), ULID(dest)>, that is used by the shim layer to uniquely identify the context state that contains the set of locators for the ULID pair. The shim layer can then choose a locator pair (which may or may not be the same as the ULID pair) to write into the packet header and transmit onto the network.. On the receive side, when the host receives a packet from the network the shim layer must discover the context state from the locator pair (i.e. src and dest address fields) contained in the packet header.

Note that ambiguities may exist between the mapping of locator pairs and ULID pairs. So we must either prevent the ambiguities from occurring (by preventing any locator pair possibly having more than one associated ULID pair), or place some other information in the IPv6 packets that will resolve the ambiguities. For the latter, the layer 3 shim approach proposes either using the flow label field or define a new extension header.

By writing a context identifier tag value into the flow label field, the context state at an end host can be uniquely identified without incurring any packet overhead. Thus, when receiving a packet the shim layer could identify the context state with the tuple <src locator, dest locator, flow label>. Using this tuple would mean that the sender and receiver hosts would have to know beforehand all of the possible locator pairings before agreeing on the flow label value for transmitted packets. While this can be achieved during the context establishment phase, any new locator added to the set of one host would have to be communicated to the other host before it could be used. Another option would be to only use the flow label as the context tag and not take into account the locators. However, this runs the risk of potential collisions (a receiver seeing the same flow label value for different sending hosts). Since the flow label field could potentially be used for other purposes some additional information is needed by a receiver to distinguish between flow labels being used as context tags and flow labels being used for other purposes. NOID (described below) proposes that judicious encoding of the next header field could provide this information without incurring any packet overhead.

The approach of using a new extension header to provide context state information is a more clean solution and does not run the risk of overloading existing IPv6 header fields. However, an extension header will of course incur packet overhead (8 bytes) and thus reduce the available MTU.

### 5.1.1 Multiple Address Service for Transport (MAST)

The Multiple Address Service for Transport (MAST) [ref] was one of the earliest proposals to suggest placing a layer between the IP and transport layers specifically as a solution for multihoming. This new layer only operates on the end hosts and affects only the hosts participating in a multihoming exchange. Since the layer is only at the end hosts, the Internet infrastructure is not affected.

MAST exploits similar functionality already defined in other proposals such as HIP and LIN6 (described below) but is more ‘streamlined’ as it is only concerned with multihoming functionality. MAST uses a simple request and response message exchange in order to create host-pair association at the participating hosts and to define the initial set of valid addresses (locators) between the hosts. The message exchange begins with each host sending an INIT message that includes the set of locators for each host and state is created when the associated acknowledgements are received. Once the association is created, a host may inform its peer of an updated list of locators by issuing a SET message. A MAST session is terminated by a 3-way handshake involving a SHUT message.

MAST is invoked after the start of a transport layer association. Thus the transport layer will already have an IPv6 address (learned from the DNS) that it is using as a handle for its peer. MAST will use this IPv6 address already known by the transport layer as the handle by which to map the set of locator addresses learned from the INIT and SET exchanges.

The MAST proposal also suggests the possibility of having a MAST agent located within a multihomed site as an alternative to having the MAST functionality in the end hosts. In this way MAST would operate in a similar manner to a NAT device, translating between locator addresses and the addresses known by the transport layer on the respective host.

### 5.1.2 Multihoming without IP Identifiers (NOID)

The ‘multihoming without identifiers’ (NOID) proposal relies on the DNS system to authenticate locator changes and prevent redirection attacks. The proposal makes use of a shim/wedge between the IP layer and ULPs called ‘M6’. The M6 layer behaves in the same way as the layer 3 shim layer described previously. Like the layer 3 shim approach the M6 layer behaves as if it were an extension header (placed before all the other extension headers including AH, ESP, fragmentation and destination options).

NOID proposes a new DNS record that would indicate whether a host is NOID-capable or not. When a communicating host (host A) uses the DNS to resolve a FQDN of another host (host B) the DNS reply will contain this record along with the set of locators associated with the FQDN of host B.

A context establishment phase consisting of a 4-way handshake can be initiated by host A in order to take advantage of the multihomed host B. In short, this context establishment phase allows host A to learn the set of locators that host B thinks it has and takes the intersection of this set with the locator set learned from the DNS. These locators are verified by performing forward and reverse DNS lookups for each FQDN-locator pair.

Once the context state has been established between the two hosts, they can communicate using any combination of valid locators. It is the M6 shim layer that performs the mapping function between the locator sets and the ‘application ID’ that is used by the ULPs. The application ID is usually chosen to be the first record returned by the DNS from the original FQDN lookup.

In order to avoid any packet overhead and reduce the MTU, NOID proposes to use the flow label field in the IPv6 header as ‘pointers’ to the context state held in each host. Thus by examining the flow label field in an incoming packet a receiving host can quickly identify and M6 context state that will map the locator used to an application ID.

Perhaps the most fundamental question to direct at the NOID proposal is ‘*How much trust can we place in the DNS?*’ If we cannot guarantee correct and authentic record entries in the DNS system the solution will not be very effective. It also means that every multihomed host will need both forward and reverse DNS entries. Thus, in reality only well run servers and other ‘important’ hosts will likely benefit from multihoming with NOID because most end hosts will not have both forward and reverse entries in the DNS.

There is also an issue with the use of the flow label field to identify context state on the end hosts. If the flow label field is being used for other purposes it may preclude the use of it for NOID (or vice versa).

### 5.1.3 Weak Identifier Multihoming Protocol (WIMP / WIMP-F)

The Weak Identifier Multihoming Protocol (WIMP) is very similar to NOID (and thus the layer 3 shim approach). Like NOID, WIMP employs a wedge layer between the IPv6 layer and the ULPs. The wedge layer takes the form of an IPv6 extension header which is before other extension headers. As with NOID, putting AH and ESP above the wedge layer enables IPsec to operate with locator changes. The first version of WIMP defined new application layer identifiers (AIDs) however the later version (designated WIMP-F) was designed to support any 128-bit AIDs and layer 3 context identifiers (CIDs).

One of the differences between WIMP and NOID is that WIMP uses AIDs that are non-routable and never appear in IPv6 headers. Unlike NOID, WIMP requires there to be a context establishment between two hosts before communication can occur. One way hash chains are generated by each end host and are used to authenticate messages exchanged by each other. WIMP uses an ‘opportunistic’ approach to context establishment. The WIMP philosophy is that a host does not care who the other host is provided that it remains the same host throughout the lifetime of the communication context. The host initiating the context establishment will receive a locator set from the corresponding host and will perform mapping between the locators used in the IPv6 header and the AID. Like NOID a ‘pointer’ or ‘tag’ is inserted into the IPv6 header of data packets to identify the WIMP context state to the receiving host. WIMP does not define where this tag is placed but notes that the flow label or SPI fields could be used.

Like NOID, WIMP allows locator sets to be updated after the initial context establishment. It claims to protect against redirection attacks based on the principle that the host receiving the redirection request can verify that the host making the redirection request holds the successor values of a hash chain and is able to compute a divided secret sent via parallel paths.

## 5.2 Hashed Based Addresses

The idea behind hashed based addresses (HBA) [51] is to provide a secure binding between the various addresses obtained from the multiple prefixes that are available to a host within a multihomed site. In brief, a host configures the interface ID part of its IPv6 address by computing a function that hashes all the available prefixes together with either a random number or a public key (if one is available). A multihomed host will configure all its interface IDs in the same manner. The interface IDs are then appended to the available prefixes to produce a set of hash based addresses. The hash function is designed so that information about all the multihomed prefixes is contained within the addresses themselves. Any peer host can verify if two addresses belong to the same set (and thus belong to the same multihomed host) by computing a single hash function containing the two addresses together with the parameters used in generating the HBAs.

HBAs are designed to be compatible with cryptographic generated addresses (CGA) [53] so that a multihomed host using HBAs will also be able to take advantage of secure neighbour discovery (SeND) [54] which relies on CGA addresses to prove address ownership. The compatibility is achieved by including the host’s public key in the HBA hash function and including a multi-prefix extension in the CGA data structure. However, multihomed host that does not need to use a public key or SeND can simply use HBA only addresses and use a random number in place of the public key in the HBA hash function.

HBA is complementary to the layer 3 shim approach described previously. Although a multihomed host will generate its addresses according to the HBA hash algorithm, any communication with a peer will begin as with normal single homed IPv6 exchanges. At an arbitrary point in the communication exchange if one of the hosts wishes to take advantage of multihoming it will invoke the ‘capabilities detection protocol’ to check if both peers support the multihoming protocol. This protocol has not been specified, although a companion internet draft [52] notes that the realistic possibilities are to either use a new RR record in the DNS or some direct signalling between the hosts during the early stages of initial communication. Regardless of how the hosts discover that each other supports multihoming, they can then participate in a context establishment phase that is described for the layer 3 shim approach (and other earlier ‘shim’ proposals). With HBA, this context establishment phase will include the transfer of the HBA data structure parameters as well as the HBA set in order for a peer host to be able to perform the hash function to verify the addresses. The context establishment will also include the communication of any context tag (as described previously) needed to identify packets belonging to a specific context.

Once context state has been established, the hosts can then take advantage of the shim layer to perform the mapping between the HBA set of locators and the identifiers known by the transport and upper layers (these will be the original IPv6 addresses used during the initial communication exchange).

### 5.3 LIN6

LIN6 [6], [7] is a protocol to support host mobility and multihoming in IPv6. LIN6 introduces two new types of IPv6 addresses, the LIN6 generalized ID and the LIN6 address.

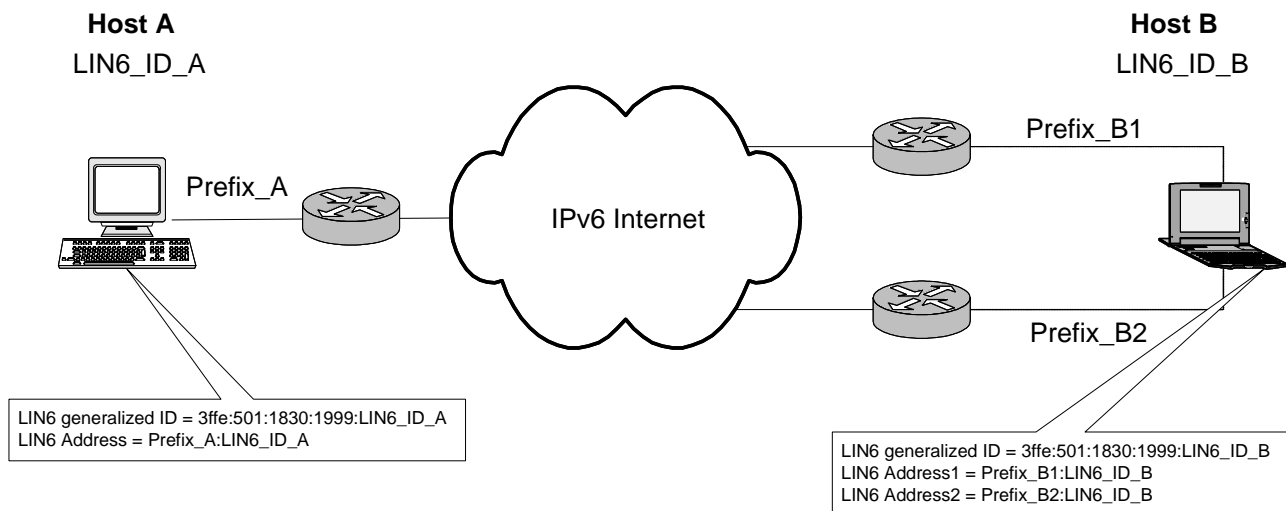
The LIN6 generalized ID is 128 bits and consists of a 64-bit LIN6 prefix (3ffe:501:1830:1999) and a 64-bit LIN6-ID. The LIN6-ID is the global unique identifier of a mobile node and uses the EUI-64 format. The LIN6 generalized ID is only used in the transport and upper layers and is assigned on a per-node basis. This address will not change even when a mobile node moves between networks.

The LIN6 address is 128 bits and consists of a 64-bit network prefix and a 64-bit LIN6-ID (the same LIN6-ID as for the LIN6 generalized ID). The 64-bit network prefix is the actual network prefix that a mobile node is currently located at. The LIN6 address is used in the network and appears in the IPv6 header. It is not passed to the transport layer. This address is assigned on a per-interface basis and will change value when a mobile node changes to another network (but only the network prefix will change).

Within the LIN6 protocol, the LIN6 generalized ID is used by the transport and application layers to designate the identity of a mobile node since this address remains constant. The LIN6 address is used by the network layer to designate both the location and the identity of a mobile node as it moves between networks. The correlation between the LIN6 generalized ID and the LIN6 address is held by one or more mapping agents. For more details on the LIN6 protocol, the reader should refer to [7].

Figure 9 illustrates multihoming support using LIN6. Host B is a multihomed host, having 2 network interfaces. Hosts A and B have the LIN6 generalized IDs ‘3ffe:501:1830:1999:LIN6\_ID\_A’ and ‘3ffe:501:1830:1999:LIN6\_ID\_B’ respectively. Host A has the LIN6 address ‘Prefix\_A:LIN6\_ID\_A’ and host B has two LIN6 addresses: ‘Prefix\_B1:LIN6\_ID\_B’ and ‘Prefix\_B2:LIN6\_ID\_B’.

If host A wishes to establish a TCP connection to host B it uses the LIN6 generalized IDs such that the TCP association is between 3ffe:501:1830:1999:LIN6\_ID\_A and 3ffe:501:1830:1999:LIN6\_ID\_B. In LIN6, the mapping agents will resolve these addresses to the current location of the nodes (i.e. their LIN6 addresses) for transmission at the network layer.



**Figure 9 Multihoming in LIN6**

Assume that host A learns both ‘Prefix\_B1’ and ‘Prefix\_B2’ as the network prefixes of host B and chooses ‘Prefix\_B1’. The src/dest addresses in the IPv6 header will now be Prefix\_A:LIN6\_ID\_A and Prefix\_B1:LIN6\_ID\_B respectively. Supposing host B’s connection to network prefix ‘Prefix\_B1’ breaks. Host A will receive ICMP unreachable errors and can then decide to use ‘Prefix\_B2’ as the destination address. However, the original TCP connection is not broken because the TCP association is between the LIN6 generalized IDs and not the LIN6 addresses.

One can imagine how this multihoming solution could be extended so that host A could use both host B’s prefixes simultaneously for different flows.

## 5.4 HIP

The Host Identity Protocol (HIP) [39] is a mechanism which decouples the transport layer from the network layer so that sockets are not bound to IP addresses as they traditionally are. HIP introduces a new host namespace between the network and transport layers. For a host using HIP, the transport layer sockets and IP security associations (SA) are not bound to IP addresses. Instead, they are bound to host identifiers which, in turn are dynamically bound to IP addresses. Applications do not see IP address, instead they see the HIP identifiers. The mapping between host identifiers and IP addresses can be one to many, thus allowing for multihomed hosts. Because the mappings are dynamic, hosts can change their addresses during established communication sessions with other hosts.

In the base HIP specification [39] hosts use the same IP addresses for their peers that were agreed during the initial HIP exchange. In order to facilitate multihoming (and mobility) an extension to the specification [40] describes how IP addresses can be changed during communication between hosts.

When a host wishes to change its IP address, e.g. by selecting a different outgoing interface or outgoing ISP, it sends a HIP Readdress packet to its peer. Due to the danger of flooding attacks from bogus hosts, the peer must check reachability of the new address before it is used in the socket context. This reachability check is achieved via HIP Address Check and Address Check Reply packets.

The HIP specification also includes the notion of a Forwarding Agent that can provide reachability to HIP nodes that are located behind the FA but do not support the same protocol version (IPv4 or IPv6) as the other HIP peer.

HIP facilitates host multihoming and multi-addressing and preserves connectivity across address changes. However, HIP has no solutions for address selection or traffic engineering. Also, since applications will see a new namespace, it is not clear how HIP hosts can interoperate with legacy hosts (perhaps via Forwarding

Agents) and how legacy applications will behave when faced with not being able to manipulate IP address at the socket API.

## 5.5 8+8, GSE and 16+16

The original 8 + 8 addressing model [43] split the IPv6 address into two 8-byte parts. The least significant 8 bytes formed the *End System Designator* (ESD) which uniquely identifies an interface on an end host. The most significant 8 bytes forms the *Routing Goop* (RG), which encodes information pertaining to the location of the interface in the global Internet.

The structure of the ESD is based on an *Identity Token* and a *Private Topology Partition*. The Identity Token is essentially the IEEE 48-bit MAC address although the 8+8 proposal allows for different ‘modes’ so that other addressing possibilities (such as numbering point to point links where no identifiable MAC address is available) can be catered for. The top 15 bits of the ESD is Private Topology Partition, which provides for 32768 distinct partitions in a site’s topology. The ESD is globally unique thanks to the Identity Token but the Private Topology Partition also allows for a wide variety of organisational subnet hierarchies within a site.

The RG part consists of a 3-bit prefix to identify the type of IPv6 address followed by 13 bits of a Large Structures Identifier (LSID) and 48 bits available for further routing information. The ‘Large Structures’ are considered to be those that offer significant topological aggregation possibilities such as large transit networks and exchange points.

Within a site using the 8+8 methodology, the source address of packets sourced from within the site will generally contain the site local prefix. The site’s border router will insert the correct RG in place of the site local prefix corresponding to the attachment path that site border router represents. Thus, a site that is multihomed can set local policies to determine which site border router a packet leaves by.

Discovering the correct RG for the destination address of packets sourced from within the site is not as straightforward. The 8+8 proposal is to provide two new records in the DNS: An “AA” record which returns the 8-byte ESD and an “RG” record which returns the 8-byte RG. It also mentions the possibility of having an “RG Name” (which is a full DNS name) inside the AA record so that it can be recursively resolved upstream. However, there is no concrete explanation of how all the RG information pertaining to each ESD can be distributed throughout the global DNS and whether or not this would scale or be at all feasible.

The 8+8 model was later revised and called GSE (Global, Site, End-system) [44]. In GSE the 8+8 structure of the IPv6 address was changed to 6+2+8, with the lower 2-bytes of the RG being replaced with a *Site Topology Partition* (STP). The RG still represents the location of the site in the global Internet but now site specific topology structure is encoded in the STP. This leaves the 8-byte ESD to be entirely used as an Identity Token (i.e. the Private Topology Partition is removed from the ESD). In DNS queries the “AA” record now contains the 8-byte ESD + the 2-byte STP and the RG record contains the 6-byte routing information.

A similar solution to 8+8 is the so-called 16+16 proposal. In 16+16, the full 16-byte IPv6 address is used as the locator. The 16-byte identifier part is included in an appropriate IPv6 extension header.

Whichever method of locator/identifier mapping is considered it is unlikely to be a short term site multihoming solution for a number of reasons. Firstly, because the identifier needs to be mapped to an arbitrary number of locators the mapping state must be distributed throughout the global Internet. Doing this securely and in a scalable manner is a non-trivial task. If DNS is used as the mapping agent it will have a dependency on the distribution of routing information and successful routing information will depend on DNS lookups thus creating a cyclic dependency which contravenes the architectural principles of the Internet (see [46]). There is also a question of privacy. Where globally unique identifiers are visible there is potential for them to be spoofed in addition to unwelcome identity tracking.

Note that both the Host Identity Protocol and Location Independent Networking described in the previous sections are general two space solutions, albeit applied to the problem of host multihoming.

## 6 Transport Solutions

Although there are suitable host multihoming solutions at the IP layer, changing an IPv6 address during the lifetime of a connection has implications for transport layer protocols such as UDP and TCP. Quite simply, the transport layer will not recognise a received packet with a different source/destination address as belonging to the same transport connection before the address was changed. Therefore, either the IPv6 address change must be hidden from the transport layer (as with two space or mobility solutions), or support for address changes must be added to the typical transport protocols.

### 6.1 Stream Control Transmission Protocol (SCTP).

The Stream Control Transmission Protocol (SCTP) [7], is a reliable transport protocol operating on top of IPv4/IPv6 that provides network-level fault tolerance by supporting host multihoming at either end of the connection.

If a client host is multihomed it informs the corresponding server host about all of its IPv6 addresses in the INIT chunk's address parameters. The corresponding server host will then list all of its IPv6 address in the INIT-ACK chunk. An instance of SCTP regards each IPv6 address of its peer as a 'transmission path' towards it. Each SCTP instance chooses one of these addresses as the primary transmission path, upon which data exchange will normally occur.

Each end of the SCTP connection monitors all the transmission paths to its peer by sending HEARTBEAT chunks on every path that is not being used to exchange data chunks. The peer SCTP hosts acknowledge each HEARTBEAT chunk with a HEARTBEAT-ACK chunk.

Each transmission path is assigned a state: either active or inactive. A path is active if it has recently been used and has received the corresponding ACK for the SCTP datagram that was transmitted on it. If these ACKS repeatedly fail, the path is marked as inactive and a notification is sent to the application layer of the host.

If the primary path becomes inactive, the sending host may automatically choose a new primary path or the user may instruct the local SCTP instance to use a new primary path.

### 6.2 Preserving active TCP sessions.

A way of allowing hosts to change their IP addresses during the lifetime of a TCP connection is presented in "Preserving Active TCP Sessions on Multihomed IPv6 Networks" [35]. This solution recognises that each participant in a TCP session may potentially have several global IPv6 addresses. Furthermore, this set of valid addresses remains relatively stable over a period of time. Therefore, it seems logical that, prior to the establishment of a TCP connection, the valid set of IPv6 addresses that could be used during the lifetime of the connection, should be exchanged by the participants. Thus, assuming that the *set* of addresses valid for a host does not change during the TCP connection, the connection should be able to accept any IPv6 address change by either participating host, provided that address appears in the agreed set..

The specific proposal in [35] is to include a 'PREFIXES' IPv6 option with the SYN and SYN/ACK packets during the 3-way handshake of a TCP connection establishment. Each participant acknowledges the PREFIXES with a PREFIXES\_ACK containing the prefixes that it received.

---

### 6.3 End to End Multihoming

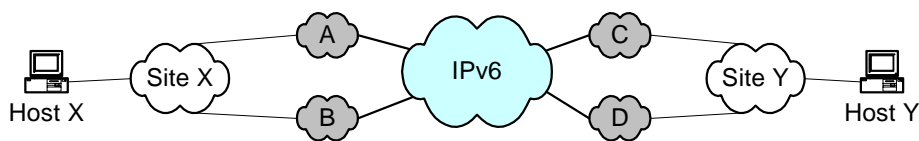
The notion of end to end multihoming as described in [8] is supported by transport (TCP or UDP) or application layers. Since multihoming support is implemented in the end systems no routing protocol changes are needed in the network. Instead end to end multihoming requires modifications to APIs and applications on the end systems.

End to end multihoming is a simple approach whereby multiple addresses are assigned to an interface and the choice of addresses to use is delegated to the application or transport layer. The authors of [8] propose to change TCP so that applications can pass multiple addresses to the transport layer and that all possible addresses are transmitted to the TCP peer via a TCP option. It is proposed to use DNS lookups for applications to learn of all the possible addresses for a destination. It is not clear how a host learns of all the addresses it may use as a source address (perhaps by router advertisements) nor how to best select a source address based on the destination (or vice versa) Changes must also be made so that the transport layer or application can detect and react to a loss of connection. Although details of how this may be achieved are not presented in [8].

## 7 Site Exit Router and Host Behaviour

### 7.1 Host-Centric Multihoming

The “Host-Centric IPv6 Multihoming” Internet Draft [40] discusses problems and possible solutions of hosts receiving site prefixes in a multihomed network. If a site is connected via two ISPs, their border routers will advertise both prefixes to the site.



**Figure 10 Site multihoming setup**

Figure 10 shows an example multihoming scenario. Site X is connected via ISPs A and B to the IPv6 Internet, and site Y via ISPs C and D. Host X receives prefix advertisements from ISPs A and B, and generates the corresponding IPv6 addresses (say A:X and B:X) with A and B corresponding to the site prefixes and X being the host identifier. The same holds true for host Y and ISPs C and D: the addresses generated are C:Y and D:Y respectively.

If host X tries to communicate with host Y, it selects a destination address from the 2 available global unicast addresses of host Y (assuming that both are made available via DNS). Subsequently, host X selects a source address from the addresses associated with the outgoing interface. However, the selection of the source address is not as simple an issue as seems at first glance. Since most site exit routers now perform ingress filtering [36] as a way of preventing security attacks using spoofed source addresses, if host X selects the wrong source address its packets will be dropped by the site exit router. For example, if the address based on prefix A (A:X) is used as the source address, the interior routing of site X (which routes on the destination address) may route the packet through site X’s exit router connected to ISP B. If the router connected to ISP B performs ingress filtering, packets with a source address based on prefix A will likely be dropped since they do not match the expected prefix, which would be ISP B’s prefix. This problem is known as the ‘site exit issue’.

The following alternatives are proposed and to solve the site exit issue:

1. *Relaxing source address checks*

Either the ISP could switch off the source address checks altogether, or the site exit routers may be configured in such a way that they accept all prefixes that are valid in the site, i.e. the exit router for ISP B accepts A’s prefixes. The former approach requires a high level of trust between the ISP and the site, but has the advantage of not requiring any changes in the hosts or routers. If a list of authorised prefixes is used, the site exit routers need to be configured somehow to be aware of these address prefixes.

2. *Source address dependent routing*

Packets originating in site X are passed through a source routing domain, to which all site exit routers are connected. These routers would choose a route based on the destination and source addresses of a packet, selecting the appropriate exit router for the given source address. Alternatively, the site exit routers could be connected via a mesh of tunnels. If the source address check on a packet reveals that the address

prefix is not a valid one for the ISP the exit router is connected to, it forwards the packet through a tunnel to a more appropriate site exit router<sup>1</sup>.

As this approach would require significant changes in the router implementations, it would be only a long-term solution.

### 3. *Source address selection by the host*

This possible solution relies on the host to determine the appropriate source address that is compatible with the site exit chosen by the routing protocol. Alternatively, the host could tunnel the packet to the correct site exit router. The former approach would require a form of “source address discovery”, the latter an “exit router discovery” (for a given source address).

Both approaches require significant changes in the IPv6 implementation in hosts, and additionally minor changes in routers (for generating the new ICMP packets for the “source address discovery” or “exit router discovery”).

### 4. *Packet rewriting at the exit router*

A site exit router receiving a packet with a “wrong” source address could take action to replace the address instead of dropping the packet, provided it recognises the address as a valid address from its site. If it would do so on any packet, it would prevent all ingress filtering. The packet rewriting could either mean adding a home address option, with the original source address in the option, and a newly generated “correct” address as the source address of the packet, or the site exit router could use IPv6-in-IPv6 encapsulation with the original destination address and a new conforming source address in the encapsulating header. Simply replacing the source address in the IPv6 header (without adding a home address option) is mentioned too, but is probably not a good solution due to problems with IPsec, for example. Besides, the router would need to know which alternative source address could be used for the host originating the packet.

Option 1 requires no changes to existing systems (if ingress filtering is switched off completely), or only minimal configuration at the site exit routers. All other options require more or less significant changes to existing host and/or router implementations. As a long-term solution, option 2 is favoured by the authors of [40].

---

<sup>1</sup> It is not detailed how this “more appropriate” router should be determined.

## 8 Conclusions

Requirements from RFC 3582																
Solution	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16
IPv4 approach	Y	Y	Y	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y
Routing																
Route aggregation	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	?	Y
RFC 2260	Y	N	N	Y	Y	Y	Y	Y	Y <sup>1</sup>	Y	Y	Y	Y	N	?	Y
RFC 3178	Y	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	?	Y
Mutual Backup	Y	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	?	Y
Router renumbering	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	?	Y	Y	?	?
Routing support	Y	Y	Y	Y	?	N	Y	Y	?	Y	Y	?	Y	Y	?	Y
NAROS	Y	Y	Y	Y	?	N	Y	Y	Y	Y	?	?	Y	Y	Y	?
Routing header	Y	Y	Y	Y	Y	N	Y	Y	?	Y	Y	?	Y	Y	Y	?
2 Space																
L 3 Shim <sup>2</sup>	Y	Y	Y	?	N	Y	Y	Y	Y	Y	N	Y	?	Y	Y	Y
LIN6	Y	Y	Y	?	N	Y	Y	Y	Y	Y	N	Y	?	Y	Y	Y
HIP	Y	Y	Y	?	N	Y	Y	Y	Y	Y	N	Y	?	Y	Y	Y
8+8	Y	Y	Y	Y	N	N	N	Y	Y	Y	N	Y	?	Y	Y	?
Transport																
SCTP	Y	Y	Y	?	N	Y <sup>3</sup>	Y	Y	Y	Y	?	Y	?	Y	Y	Y
Preserving TCP	Y	Y	Y	Y	N	Y <sup>4</sup>	Y	Y	Y	Y	N	Y	?	Y	Y	Y
End to end	Y	Y	Y	?	N	Y	?	Y	Y	Y	N	Y	?	Y	Y	?
Other																
Host centric 1	Y	Y	Y	Y	Y	N	Y	N	Y	Y	Y	Y	Y	Y	Y	N
Host centric 2	Y	Y	Y	Y	N	N	Y	Y	Y	N	Y	Y	Y	Y	Y	Y
Host centric 3	Y	Y	Y	Y	N	N	Y	Y	Y	Y	N	?	Y	Y	Y	Y
Host centric 4	Y	Y	Y	Y	N	N	Y	Y	Y	N	N	N	Y	Y	?	?

**Table 1 Evaluation of proposals**

<sup>1</sup> Assuming optimisation that there are BGP peerings between providers so no prefixes are injected into DFZ

<sup>2</sup> Including MAST, NOID, WIMP, HBA

<sup>3</sup> Only for SCTP. Any connections not using SCTP on the host will not survive across address changes

<sup>4</sup> Only for TCP applications

**Key:**

Y – requirement satisfied

N – requirement not satisfied

? – debatable or further study needed to determine if requirement is satisfied

Table 1 provides an overview of how well the various proposed solutions satisfy the goals for IPv6 site multihoming as described in RFC 3582. Something that is obvious from glancing at the table is that non of the proposed solutions satisfy all the IPv6 multihoming goals.

The various routing proposals, whilst satisfying the scalability requirements that plague the IPv4 approach, fall short in other areas. For example they may require co-operation between ISPs in order to deploy the multihoming solution. This may also lead to performance degradation should the link between the providers suffer congestion. Other routing proposals do not offer transport layer survivability during rehomeing events because only new connections can take advantage of their mechanisms. Existing connections have no way of switching their addresses without breaking their transport layer semantics.

Of the proposed two space solutions, most of them fulfil similar requirements. The scalability problem of injecting prefixes into the DFZ is not an issue and transport layer survivability is assured by the consistent mapping of locators and identifiers. However, these solutions are not likely deployable in the short term since they involve significant changes to host systems. From a site customer point of view, it would also mean updating all hosts within the site that wish to take advantage of that site being multihomed.

The transport solutions have similar advantages and disadvantages to the two space solutions. Most of the multihoming goals are satisfied but they have the same kind of deployment problems as the two space solutions. Furthermore, the transport solutions can only provide transport layer survivability for applications using that use that specific transport protocol. For example, SCTP will not provide survivability on rehomeing events to applications using TCP or UDP. They do not have the same transport transparency advantage that the two space solutions operating between the IP and transport layer have. Thus there is an additional barrier to deployment with transport solutions in that applications need to be modified to use the new or transport protocol.

None of the four options for host centric multihoming offer transport layer survivability. Of those options the simplest approach is to relax ingress filtering checks, although this involves a large amount of trust between the provider and the multihomed site and may also introduce a potential security vulnerability.

For a site wishing to exploit multihoming in IPv6 now there is no clear solution or combination of solutions which can be recommended. Certainly the choice of mechanisms to deploy depends greatly on the specific goals and circumstances of the site in question. If the site in question can get its ISPs to co-operate it may adopt one of the routing solutions such as using route aggregation, RFC 2260 or RFC 3178. Although these do offer transport layer survivability of some sort (existing connections are re-routed and not broken) there is no way for hosts to discover which of their multiple addresses should be chosen for new outgoing connections. However, combining one of the above solutions with one that informs hosts of outages (e.g. in the renumbering approach) would seem like a promising strategy.

## References

- [1] N. Montavont, T. Noel, M. Kassi-Lahlou, "MIPv6 for Multiple Interfaces", IETF Internet-Draft draft-montavont-mobileip-mmi-00.txt, July 2002 (work in progress).
- [2] D. B. Johnson, C. E. Perkins, J. Arkko, "Mobility Support in IPv6", IETF RFC 3775, June 2004.
- [3] H. Soliman, K. El-Malki, C. Castelluccia, "Per-flow movement in MIPv6", IETF Internet-Draft draft-soliman-mobileip-flow-move-01.txt, November 2001 (work in progress).
- [4] IST 1999-10054 Project BRAIN, Deliverable D2.2, March 2001, <http://www.ist-brain.org>
- [5] LIN6 homepage, <http://www.lin6.net/>
- [6] F. Teraoka, M. Ishiyama, M. Kunshi, A. Shionozaki, "LIN6: A Solution to Mobility and Multi-Homing in IPv6", IETF Internet Draft draft-teraoka-ipng-lin6-01.txt, August 2001.
- [7] R. Stewart, et al, "Stream Control Transmission Protocol", IETF RC 2960, October 2000.
- [8] M. Ohta, "The Architecture of End to End Multihoming", IETF Internet Draft draft-ohta-e2e-multihoming-02.txt, July 2001.
- [9] Site Multihoming in IPv6 (multi6), <http://www.ietf.org/html.charters/multi6-charter.html>
- [10] IPv6 Multihoming Solutions (ipv6mh), <http://arneill-py.sacramento.ca.us/ipv6mh/>
- [11] M. Bagnulo, A. Garcia-Martinez, A. Azcorra, D. Larrabeiti, "Survey on Proposed IPv6 Multi-Homing Network Level Mechanisms", IETF Internet Draft draft-bagnulo-multi6-survey6-00.txt, July 2001.
- [12] F. Dupont, "Multihomed Routing Domain Issues for IPv6 Aggregatable Scheme", IETF Internet Draft draft-ietf-ipngwg-multi-isp-00.txt, September 1999.
- [13] R. Hinden, M. O'Dell, S. Deering. "An IPv6 Aggregatable Global Unicast Address Format", IETF RFC 2374, July 1998.
- [14] S. Thomson, T. Narten, "Stateless Address Autoconfiguration", IETF RFC 2462, December 1998.
- [15] M. Crawford, "Router Renumbering for IPv6", IETF RFC 2894, August 2000.
- [16] N. Bragg. "Routnig Support for IPv6 Multi-homing", IETF Internet Draft draft-bragg-ipv6-multihoming-00.txt, November 2000.
- [17] M. Py, "Multi Homing Aliasing Protocol (MHAP)", IETF Internet Draft draft-py-mhap-01a.txt, April 2002.
- [18] I. Van Beijnum, "Provider-Internal Aggregation based on Geography to Support Multihoming in IPv6", IETF Internet Draft draft--van-beijnum-multi6-isp-int-aggr-01.txt, June 2003.
- [19] I. Van Beijnum, M. Py, "A Geographically Aggregatable Provider Independent Address Space to Support Multihoming in IPv6", Work in Progress.
- [20] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, V. Paxson, "Stream Control Transmission Protocol", IETF RFC 2960, October 2000.
- [21] M. O'Dell, "GSE – An Alternative Addressing Architecture for IPv6", IETF Internet Draft draft-ipng-gseaddr-00.txt, February 1997.
- [22] T. Hain, "Application and Use of the IPv6 Provider Independent Global Unicast Address Format", IETF Internet Draft draft-hain-ipv6-pi-addr-use-02.txt, March 2002.
- [23] J. Yu, "IPv6 Multihoming with Route Aggregation", IETF Internet Draft draft-ietf-ipngwg-ipv6multihome-with-aggr-01.txt, August 2000.

- 
- [24] M. Blanchet, “A Flexible Method for Managing the Assignment of Bits of an IPv6 Address Block”, IETF Internet Draft draft-ietf-ipv6-ipaddressassign-03.txt, August 2002.
  - [25] R. Draves, “Default Address Selection for IPv6”, IETF Internet Draft draft-ietf-ipv6-default-addr-select-09.txt, August 2002.
  - [26] L. Coene, “Multihoming Issues in the Stream Control Transmission Protocol”, IETF Internet Draft draft-coene-sctp-multihome-03.txt, February 2002.
  - [27] L. Coene, “Multirouting”, IETF Internet Draft draft-coene-multi-00.txt, February 2002.
  - [28] F. Dupont, “The Host Identity Payload Protocol: toward a secure solution to mobility and multihoming”, May 2002.
  - [29] T. Bates, Y. Rekhter, “Scalable Support for Multihomed Multi-provider Connectivity”, IETF RFC 2260, January 1998.
  - [30] J. Hagino, H. Snyder, “IPv6 Multihoming Support at Site Exit Routers”, IETF RFC 3178, October 2001.
  - [31] B. Black, V. Gill, J. Abley, “Goals for IPv6 Site Multihoming Architectures”, RFC 3582, August 2003.
  - [32] J. Abley, B. Black, V. Gill, “IPv4 Multihoming Motivation, Practices and Limitations”, IETF Internet Draft draft-ietf-multi6-v4-multihoming-00.txt, June 2001.
  - [33] G. Huston, “Analyzing the Internet’s BGP Routing Table”, Internet Protocol Journal, Volume 4 Number 1, March 2001.
  - [34] D. Johnson, S. Deering, “Reserved IPv6 Subnet Anycast Addresses”, IETF RFC 2526, March 1999.
  - [35] P. Tattam, “Preserving Active TCP Sessions on Multihomed IPv6 Networks”, September 1999.
  - [36] P. Ferguson, D. Senie, “Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing”, IETF RFC 2827, May 2000.
  - [37] C. de Launois, O. Bonaventure, “NAROS: Host-Centric IPv6 Multihoming with Traffic Engineering”, IETF Internet Draft draft-de-launois-multi6-naros-00.txt, May 2003.
  - [38] P. Nikander, R. Moskowitz, “Host Identity Protocol”. IETF Internet Draft draft-moskowitz-hip-07.txt, June 2003.
  - [39] P. Nikander, J. Arkko, P. Jokela, “End-Host Mobility and Multi-Homing with Host Identity Protocol”, IETF Internet Draft draft-nikander-hip-mm-00.txt, June 2003.
  - [40] C. Huitema, R. Draves, “Host-Centric IPv6 Multihoming”, IETF Internet-Draft draft-huitema-multi6-hosts-01.txt, June 2002 (work in progress).
  - [41] C. Ng, J. Charbon, “Multi-homing Issues in Bi-directional Tunneling”, IETF Internet Draft draft-ng-nemo-multihoming-issues-01.txt, May 2003.
  - [42] J. Charbon, C. Ng, K. Mitsuya, T. Ernst, “Evaluating Multi-homing Support in NEMO Basic Solution”, IETF Internet Draft draft-charbon-nemo-multihoming-evaluation-00.txt, July 2003.
  - [43] M. O’Dell, “8+8 – An Alternate Addressing Architecture for IPv6”, IETF Internet Draft draft-odell-8+8-00.txt, October 1996 (expired)
  - [44] M. O’Dell, “GSE – An Alternate Addressing Architecture for IPv6”, IETF Internet Draft draft-ipng-gseaddr-00.txt, February 1997 (expired)
  - [45] A. Durand, “GSE+ - An Alternate Addressing Architecture to GSE”, IETF Internet Draft draft-durand-gse+-00.txt, February 1997 (expired)
  - [46] B. Carpenter, “Architectural Principles of the Internet”, IETF RFC 1958, June 1996
  - [47] E. Nordmark, “Multihoming without IP Identifiers”, IETF Internet Draft draft-nordmark-multi6-noid-02.txt, July 2004.
  - [48] D. Crocker. “Multiple Address Service for Transport (MAST)”, IETF Internet Draft draft-crocker-mast-proposal-01.txt, September 2003 (expired).
  - [49] J. Ylitalo, V. Torvinen, E. Nordmark, “Weak Identifier Multihoming Protocol Framework (WIMP-F)”, IETF Internet Draft draft-ylitalo-multi6-wimp-01.txt, June 2004 (expired).

- 
- [50] R. Draves, “Default Address Selection for Internet Protocol version 6 (IPv6)”, IETF RFC 3484, February 2003.
  - [51] M. Bagnulo, “Hashed Based Addresses (HBA)”, IETF Internet Draft draft-ietf-multi6-hba-00.txt, December 2004.
  - [52] M. Bagnulo, J. Arkko, “Functional Decomposition of the M6 Protocol”, IETF Internet Draft draft-ietf-multi6-functional-dec-00.txt, December 2004.
  - [53] T. Aura, “Cryptographically Generated Addresses (CGA)”, IETF Internet Draft draft-ietf-send-cga-6.txt, April 2004.
  - [54] J. Arkko et al, “Secure Neighbour Discovery (SEND)”, IETF Internet Draft draft-ietf-send-ndopt-06.txt, July 2004.

---

## Abbreviations

AS	Autonomous System
BGP	Border Gateway Protocol
CIDR	Classless Inter-Domain Routing
DFZ	Default-Free Zone
eBGP	Exterior Border Gateway Protocol
FQDN	Fully Qualified Domain Name
IGP	Interior Gateway Protocol
iBGP	Interior Border Gateway Protocol
IPSec	IP Security
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISP	Internet Service Provider
PA	Provider Aggregatable
PDA	Personal Digital Assistant
PI	Provider Independent
SCTP	Stream Control Transmission Protocol
TCP	Transport Control Protocol
UDP	User Datagram Protocol
VoD	Video on Demand
VoIP	Voice over IP