# A Review of IPv6 Multihoming Solutions

Habib Naderi
Department of Computer Science
University of Auckland
Auckland, New Zealand
hnad002@aucklanduni.ac.nz

Brian E. Carpenter
Department of Computer Science
University of Auckland
Auckland, New Zealand
brian@cs.auckland.ac.nz

**Abstract - Multihoming is simply defined as having connection to the Internet through more than one Internet service provider. Multihoming is a desired functionality with a growing demand because it provides fault tolerance and guarantees a continuous service for users. In the current Internet, which employs IPv4 as the network layer protocol, this functionality is achieved by announcing multihomed node prefixes through its all providers. But this solution, which employs Border Gateway Protocol, is not able to scale properly and adapt to the rapid growth of the Internet. IPv6 offers a larger address space compared to IPv4. Considering rapid growth of the Internet and demand for multihoming, the scalability issues of the current solution will turn into a disaster in the future Internet with IPv6 as the network layer protocol. A wide range of solutions have been proposed for multihoming in IPv6. In this paper, we briefly review active solutions in this area and perform an analysis, from deployability viewpoint, on them.**

***Keywords - IPv6, Multihoming***

## I. INTRODUCTION

The rapid growth of the Internet, during recent years, and known limitations in its native network protocol have raised some concerns among experts about the future. IPv4 addresses will run out in the near future. It is a big obstacle to the development of the Internet. One proposed solution, and the most promising one, is replacing IPv4 with a new protocol, which is able to resolve IPv4 issues. Early deployments and experiments have shown that IPv6 is stable and reliable enough to replace IPv4, but a practical and incremental deployment plan and also a reasonable solution for multihoming seem necessary. Multihoming has been an open problem for 35 years since the invention of the Internet [1] and no perfect solution has been proposed for that during these years.

Multihoming is simply defined as having connection to the Internet through more than one Internet Service Provider (ISP). Multihoming can be implemented at host or site level. A host with two or more independent connections to the Internet is called a multihomed host. A multihomed host is able to detect failures and move established communications from the failed path to one of the available working paths. A site with two or more independent connections to the Internet is called a multihomed site. A multihomed site provides multihoming functionality for its hosts. Hosts are usually unaware of the existence of multihoming in this case. Multihoming is a desired functionality because it provides fault tolerance and guarantees a reliable connectivity for users. So many users, all around the world, are interested to use and benefit from this functionality.

To achieve this functionality in the current Internet, Border Gateway Protocol (BGP) features are employed. A multihomed site acquires its Provider Independent (PI) or Provider Aggregatable (PA) prefix and then announces it through all its providers [2]. In case of PI addresses, the site's prefix appears in the Internet core routing system more than once. In other words, Internet core routers have to process more than one entry for this prefix in their routing tables. An observational study in 2009 [3] showed that employing techniques like CIDR, which make address aggregation possible, have been very helpful to keep the growth of BGP4 table size roughly proportional to the square root of the public Internet size during past years. According to another study [4], multihoming and load balancing have been two major sources of fragmentation and deaggregation of BGP4 announcements. A study in 2005 [5] showed that 20% of entries in the global routing table are associated solely with multihoming. So, as the number of multihomed sites grows rapidly, the routing table size will become a serious issue in the future. Although using PA addresses can avoid the routing table explosion problem, hosts need to be multiaddressed, which creates difficult new issues with ingress filtering, renumbering and session survivability [6].

One major concept, which is employed by most proposed solutions is the separation of identity and location. One of the assumptions in traditional IP design was static topology. So, an object's identity and location were combined into a single protocol element called *IP address*. In IP architecture, identity is the address, which also describes the location. But, new studies showed that we need to separate these two roles. *Identity* uniquely identifies a stack within an end-point, where *Location* identifies the current location of the identity element within the network. It makes it possible to define a multihomed end point with one identity and different locators. The upper layers of protocol stack will deal with identity whereas lower layers should struggle with set of locators. In other words, the upper layer does not need to be aware of multihoming and the service would be transparent to it.

Copyright (c) IARIA, 2011 ISBN:978-1-61208-002-4 

A wide range of solutions have been proposed for multihoming in IPv6. These solutions can be categorized in five major categories [7]: Routing Approach, Mobility Approach, Identity Protocol Element, Modified Protocol Element and Modified Site Exit and Host Behaviors.

There are also other views for classifying the proposed solutions. They can be classified according to the location of required modification, i.e. hosts, routing system or both, or according to the network protocol stack element, i.e. network, transport or session, which is affected. In host based solutions, multihoming is implemented in hosts and the routing system is unaware of it. All required information is stored and managed by the host. In routing system based solutions, the routing system is responsible for providing multihoming functionality and storing and managing required information. Hosts are unaware of multihoming in this case. In mixed solutions, multihoming functionality is split across hosts and routers and each component should take care of its own functions and information.

The structure of this paper is as follows. Section II presents a brief overview of related works in the area. Section III presents proposed solutions, which are active and have a chance to be selected as the standard solution. Although some solutions discussed in this paper have not been proposed specifically for multihoming, in all of them multihoming is considered as an important feature. Section IV analyzes these active solutions from deployability view point. We conclude our work in section V.

## II. RELATED WORK

Pekka Savola et al. presented the result of their survey on site multihoming in IPv6 in [8]. They presented an overview of proposed solutions along with motivations and challenges in this area and tried to show that solutions for IPv4 are not well structured enough to be applied to IPv6. Cedric de Launois et al. [9] surveyed main solutions for IPv6 multihoming, which had been proposed to IETF over the period of 2000-2005. They also compared the solutions and presented their advantages and disadvantages. The results of a comparative analysis, by Shinta Sugimoto and et al., of two host-centric solutions, SHIM6 and SCTP, were presented in [10]. They specifically focused on architecture, failure detection and security. Jun Bi et al. [11] presented a summary of IPv4 multihoming solutions. They also reviewed and analyzed a number of IPv6 site multihoming approaches and chose SHIM6 as the most promising solution. Richard Clayton [12] analyzed multihoming from an economic viewpoint.

## III. ACTIVE SOLUTIONS IN THE AREA

Although a wide variety of solutions for IPv6 multihoming have been proposed during past years, there is no agreement in the research and technical community upon choosing one of them as the best solution. Scalability has been the main concern and avoiding huge routing tables has been one of the most important goals in this area. The identifier-locator separation technique is considered as fundamental for this problem and has been employed by a majority of the solutions.

Identifier-locator separation can be implemented in different ways. Deering [13], based on an earlier proposal [14], proposed dividing IP address space into two portions, one portion to be used as the set of end-system identifiers and the other portion as wide-area locators. Hosts put identifiers, as source and destination addresses, in packets, and border routers encapsulate these packets with an outer header, which contains locators. This scheme is generically called *map-n-encap*. Mapping identifiers to locators needs an infrastructure, which needs to be fast and reliable. Map-n-encap technique also increases the size of packets, which may cause packet fragmentation, if it exceeds MTU. Another way to implement identifier-locator separation is cutting the 16-byte IPv6 address in half and then assigning one half to identifier and another half to locator. The locator part can be rewritten by the routing system, while the identifier part is fixed and unique. Hosts ignore the locator part and just use the identifier part. This approach was initially proposed by O'Dell [15] and is referred as "8+8". The positive aspect of both approaches is that the delivered packet would be identical to the sent packet although the header is rewritten by exit routers. It avoids undesirable side-effects, which are caused by similar techniques like Network Address Translation (NAT) in IPv4 [16]. Because of perceived security issues, the 8+8 proposal was not updated, but the idea has been widely used in other proposals.

Other approaches like using geographically based address prefixes [17], transport protocols with multihoming support like Stream Control Transmission Protocol (SCTP) [18] and introducing an additional level of identifier above the IP address, namely HIP [19] have also been proposed. From 2001 to 2003, more than 35 drafts related to IPv6 multihoming were produced in IETF to cover different classes of solutions [20]. After reviewing these proposals, SHIM6 [21] was selected as a standard solution. SHIM6 is a host centric solution, compatible with IPv6 and its routing architecture, which simulates identifier-locator separation. SHIM6 is not an attractive solution for service providers because it does not provide a powerful set of traffic engineering features. Using PI addresses were considered in some solutions when Regional Internet Registries removed restrictions for allocating PI prefixes. Some early IPv6 adopters used IPv4 style solutions, which raised the concern about routing table explosion problem. So, after an Internet Architecture Board workshop and report [22], new technical proposals were produced. Some of them are still active and under development [23]. LISP, ILNP, NAT66, MPTCP, continued work on HIP, name-based transport and SHIM6 are the main proposals, which are summarized and analyzed in this paper.

LISP (Locator/ID Separation Protocol [24]) is a map-n-encap solution, which is with an active IETF Working Group. LISP inserts a new network layer below the host stack network layer. The host network stack works with

Copyright (c) IARIA, 2011 ISBN:978-1-61208-002-4 

EIDs (End-point Identifiers) while the new layer works with RLOCs (Routing Locators). EID, which is a non-routable IP address, uniquely identifies a host while RLOCs are routable PA addresses, which should be easily aggregatable in the BGP4 system. LISP has two major components: data plane, which performs map-n-encap operation, and control plane, which is the EID-to-RLOC mapping system. The map-n-encap process is performed by LISP routers, ETR (Egress Tunnel Router) and ITR (Ingress Tunnel Router). ETRs perform decapsulation and ITRs are responsible for encapsulation. A fast and reliable mapping system should provide assistance for ITRs so that they can encapsulate outgoing packets in an outer header, which contains RLOCs. Incremental deployment, which needs interoperability with existing unmapped Internet, is a tricky issue [25]. One proposed solution for this problem is using proxy tunnel routers, which announce a large range of EIDs in an aggregated form. The communication between LISP and non-LISP hosts will then become possible through these proxies.

ILNP (Identifier Locator Network Protocol [26]), a direct descendant of 8+8 [15], is a network protocol, which has been designed based on identifier locator separation approach. To be incrementally deployable, designers propose building that upon IPv6. Packet headers for ILNP and IPv6 are nearly identical but, like 8+8, 64 bits of address is used as locator followed by a 64-bit identifier. The identifier names a node, not an interface, and is in IEEE EUI-64 format and is not used for forwarding. The identifier is not required to be globally unique, but a unique identifier would be very helpful. Hosts should be aware of ILNP to be able to detect failures and recover from them. ICMP protocol is used for locator updates and four new resource records should be supported by DNS.

NAT66 [27] is a stateless version of NAT44 (NAT for IPv4). Like NAT44, the source address is overwritten by NAT66 node before sending a packet out and the destination address is overwritten before sending a received packet in. NAT66 does not include port mapping, as there is an external address for every internal address. Employing NAT66 on the border router of a multihomed site enables address mapping from different external addresses to the same set of internal addresses. Switching between providers is done by changing external address in the NAT66 mapping process. Address mapping is algorithmic and checksum-neutral. Thus there is no need to maintain any per-node or per-connection state; address rewriting keeps the checksum in the transport layer unchanged. Thus there is no need to modify transport layer headers. NAT66 also allows internal nodes to be involved in peer-to-peer communications.

MPTCP (MultiPath TCP [28]) is an extension to traditional TCP, to enable it to use multiple simultaneous paths between multihomed/multiaddressed peers. The aim of MPTCP is to improve resource utilization and failure tolerance. MPTCP is a set of features on top of TCP, meant to be backward compatible, so as to work with middle boxes (e.g. NAT, firewall, proxy) and legacy applications and systems without affecting users. A MPTCP connection is started like a regular TCP connection. Then, if extra paths exist, additional TCP connections (subflows) will be created. MPTCP operates such that all these connections look like a single TCP connection to the application. There are two major differences between MPTCP and transport protocols like SCTP. First, MPTCP preserves the TCP socket interface, so it is fully compatible with existing TCP applications. Second, it uses all available address pairs between communication hosts simultaneously, and spreads the load between working paths using TCP-like mechanisms.

HIP [19] is a host-based solution for secure end-to-end mobility and multihoming, using an identity/locator split approach. In HIP, IP addresses are used as locators but host identifier is the public key component of a private-public key pair. Host identity is a long term identity so it can be used for looking up locators. Host identity is created by the host itself and can be stored in DNS to be searchable by other hosts. Each host has one host identity but can have more than one host identifier. [29] proposes a common socket API extension for HIP and SHIM6 since from upper layer's viewpoint, they look similar.

Name-based transport [30] is an evolution of the existing socket interface, which hides multihoming, mobility and renumbering from applications. Applications do not need to struggle with addresses. They can simply use domain names and leave the management of IP addresses in communication sessions to the operating system.

SHIM6 [21] is a host-centric solution, chosen by IETF as an engineering solution, for IPv6 multihoming. SHIM6 uses identity/locator scheme but does not define a new name space. IPv6 addresses are used as identifier and locator. Initial connection, similar to non-shim6 connections, uses one of the available host's IP addresses. This address will play the role of identifier, which is called ULID (Upper Layer ID), during the communication lifetime. ULID is associated with a list of the host's other IP addresses, referred to as locators. SHIM6 inserts a shim layer on top of the IP routing sub-layer and under IP endpoint sub-layer. This layer performs a mapping between ULID and locator(s). SHIM6 employs a separate protocol, called REAchability Protocol (REAP) [31], for failure detection and recovery. The recovery process is independent from and transparent to upper layer protocols. To benefit from the mentioned functionality, both ends of a communication should implement SHIM6. Also, hosts need to be multiaddressed.

## IV. ANALYSIS

Deployability is a key attribute for new Internet protocols. In this section we analyze the active solutions reviewed in section III from a deployability viewpoint. We have considered seven important aspects in our analysis: scalability, amount of required modifications, security, traffic engineering, deployment cost, ease of renumbering and code availability.

Copyright (c) IARIA, 2011 ISBN:978-1-61208-002-4 

**LISP:** RLOCs are assumed to be PA addresses, which are aggregatable in the BGP4 system. So, LISP is considered as a scalable solution. To deploy LISP, no change is required within sites or the Internet core routing system. Modifications are limited to border routers (xTRs). A mapping system, like LISP-ALT [32], is also required to maintain EID to RLOC mappings. Communications between xTRs are protected by using a 32-bit nonce. This technique only provides a basic protection. In fact, LISP and LISP-ALT are not more secure than BGP. The mapping system should support priorities and weights for each locator. Using this information, LISP is able to provide powerful facilities regarding traffic engineering and load sharing. Like other map-n-encap approaches, LISP suffers from encapsulation overhead. Encapsulation increases the packet size and probability of fragmentation, which may have negative impact on performance. IPv6 routers do not perform fragmentation and drop the packets larger than MTU. So, there is a possibility that large LISP encapsulated packets are dropped by IPv6 routers. LISP designers propose some solutions for this problem although, based on informal surveys, they believe that majority of Internet transit paths support a MTU of at least 4470 bytes and there is no need to be worried about this problem. Depending on the mapping system technology, a mapping process may impose an overhead on routing time and traffic. Employing a proper solution for interoperability, as mentioned in section III, LISP can be deployed incrementally. Renumbering only affects xTRs and mapping database. A fast mechanism is required for updating the mapping database, in case of renumbering, to avoid out of date responses to mapping requests. Two implementations are available for LISP: OpenLISP and LISP for IOS (from Cisco).

**ILNP:** ILNP is mainly implemented in hosts. Hosts can be multiaddressed and by using PA addresses, address aggregation is completely possible. So, ILNP is considered as a scalable solution. To deploy ILNP, hosts should be modified. Also, support for new resource records (I, L, PTRI and PTRL) should be added to DNS. ILNP employs ICMP protocol for locator change notification. Support for a new message called *Locator Update* needs to be added to ICMP. Although ILNP encourage applications to use FQDNs instead of IP addresses, legacy applications would still be able to work with ILNP if required APIs for conversions between FQDN and IP addresses are provided. ILNP employs IPSec to improve the security of communications. It does not include locators in authentication header, so changing locators does not affect the security of communications. To provide proper traffic engineering facilities, ILNP authorizes edge routers to rewrite locators in packet headers and enforce TE policies. ILNP is compatible with pure IPv6 so an approach like dual stack seems possible for incremental deployment. To handle a renumbering, DNS records should be updated because Identifier-locator mappings are stored in DNS. Only a research demonstration implementation of ILNP, from the University of St Andrews, is available at the moment.

**NAT66:** With NAT66, sites are able to use PI addresses as internal addresses within the site and PA addresses, which are aggregatable in the Internet routing system, as external addresses. Although internal addresses are accessible from outside, but they don't need to appear in the Internet core routing tables, thanks to NAT66 two- way mapping algorithm. So, NAT66 can be considered as a scalable solution. To deploy NAT66, no modification is required in hosts and routers. Just a NAT66 device is required to be installed on the site's exit border. Two-way address mapping enables hosts behind a NAT66 device to be accessed from outside and involved in peer-to-peer communications. It makes NAT66 less secure than NAT44 but the result is not worse than regular IPv6 communications. NAT66 does not offer any specific feature for traffic engineering but NAT66 devices could be improved to enforce TE policies. Address translation imposes a processing overhead on packet forwarding. To use NAT66 address mapping algorithm, both internal and external prefixes should be /48 or shorter to have at least 16 bits available for subnet; otherwise checksum neutrality cannot be guaranteed. Renumbering is easy, only the NAT66 device should be modified to use new prefix(es). NAT66 is not able to preserve established communications in case of renumbering and failure. There is no implementation available for NAT66 at the moment.

**MPTCP:** MPTCP extends TCP capabilities and allows hosts to benefit from parallel flows to improve the performance and network utilization. MPTCP needs hosts to be multiaddressed and addresses are assumed to be PA addresses to take care of scalability. To deploy MPTCP, only hosts need to be modified. MPTCP is backward compatible with TCP, so TCP applications are able to use it easily without need to any change. MPTCP designers have tried to keep MPCTP as secure as TCP but multipath feature has opened some security concerns [33]. MPTCP allows hosts to enforce their preferences for spreading their traffic over different paths, but there is no way for receivers to change these preferences. Some solutions like ECN and fake congestion signals [34] have been proposed for this problem. MPTCP is backward compatible with traditional TCP, so incremental deployment is possible. But to benefit from multipath features, both end of communication should support MPTCP. One of the host's IP addresses, which is used for establishing connection, plays the role of identifier and also locator for one of subflows. In case of renumbering, such subflows can cause confusion and security problems. Two versions of MPTCP, based on LinShim6 code base, have been implemented in Université Catholique de Louvain. Both are still incomplete.

**HIP:** HIP allows hosts to be multiaddressed and addresses are assumed to be PA addresses. So, address aggregation is possible without any change in routing system which makes HIP a scalable solution. HIP is a host-centric, solution and major modifications should be implemented in hosts. To maintain host identifiers, DNS or a PKI (Public Key Infrastructure) is required. To benefit from HIP features, applications should use an extended socket interface, which has been proposed for this purpose [29]. Another version of

Copyright (c) IARIA, 2011 ISBN:978-1-61208-002-4 

HIP, opportunistic HIP, has been proposed for situations where DNS or PKI is not available. HIP employs IPSec to provide a secure media for communications. There is no specific facility for traffic engineering in HIP. There are some issues regarding incremental deployment of HIP [35]. HIP is able to change locators without breaking communications. To handle renumbering, host identifiers should be updated in DNS/PKI. There are five implementations for HIP: OpenHIP, HIP for Linux, HIP for inter.net, InfraHIP and pyHIP.

**Name-based Sockets:** Name-based sockets implement an identifier-locator separation scheme by allowing applications to use domain names instead of IP addresses. IP addresses are assumed to be managed by the operating system. Using PA addresses, address aggregation is easily possible so, this solution can be considered as a scalable solution. To deploy Name-based sockets, hosts networking stack should be modified to support required features. No modification is required in routing system. Name-based sockets are vulnerable to domain name spoofing, redirection and flooding attacks. Solutions like using additional forward lookups in DNS for verifying domain names and exchanging random numbers, in case of redirection, have been proposed to protect Name-based sockets against mentioned attacks. Name-based sockets are not intended to improve IPv6 security; they just try to keep the level of security at the same level as today's Internet. This solution does not provide any specific facility for traffic engineering. Name-based sockets are backward compatible to traditional socket interface, so incremental deployment is possible. Name-based sockets provide required mechanisms for changing locators without breaking communication sessions. So, renumbering is easy and just needs an update to DNS. A prototype of name-based sockets has been implemented as a result of collaboration between Ericsson, Tsinghua University and Swedish Institute of Computer Science.

**SHIM6:** SHIM6 is a host-centric solution, which is able to provide multihoming functionality for multiaddressed hosts. If addresses are PA addresses, address aggregation would easily be possible. So, SHIM6 is considered as a scalable solution. SHIM6 is implemented in hosts and doesn't need any change in the routing system. SHIM6 is not intended to improve the security of the IPv6 communications. HBA/CGA, context tag and a 4-way handshake mechanism for context establishment have been employed to help SHIM6 not to downgrade the security. SHIM6 provides some simple mechanisms regarding traffic engineering. Hosts are able to notify the other end of communication about their preferences among available locators. It is a host level mechanism and site administrators need other mechanisms for enforcing traffic engineering policies in their sites. [36] proposes some improvements to SHIM6 for enhancing its traffic engineering capabilities. A SHIM6 capable host is able to communicate with non-SHIM6 hosts. Thus, incremental deployment is possible, although SHIM6 is unable to activate its capabilities in these cases. SHIM6 is able to handle locator changes on the fly, so handling renumbering is easy. If a renumbered prefix is in use, the corresponding context can still continue its work. But, such contexts are a source of confusion and security issues. Two implementations are available for SHIM6: LinShim6 and OpenHIP.

Figure 1 shows a table summarizing characteristics of the described solutions. Our analysis can be summarized as follows: SHIM6, HIP, MPTCP, ILNP and name-based sockets are, in fact, solutions for host multihoming while LISP and NAT66 are considered as site multihoming solutions. The amount of required modifications for deploying a solution is an important factor. Solutions, which need fewer modifications would be more desirable since they offer less deployment cost. LISP offers some precise features for traffic engineering, other solutions just propose some general guidelines and possibilities. Traffic Engineering is an important feature from administrator's viewpoint as it enables them to control site's incoming and outgoing traffic. LISP and HIP have some issues with incremental deployment. As the Internet is a widespread network, incrementally deployable solutions have a higher chance to be adopted. Only NAT66 is not able to preserve communications in case of failure and renumbering, although SHIM6 and MPTCP also have some issues with renumbering in special cases. Solutions, which make renumbering simple are more desirable from a site administrator's viewpoint because they offer more flexibility in changing service providers. From a technical viewpoint, it seems that ILNP and LISP offer a more complete set of features compare to other solutions. The co-chairs of the IRTF RRG have recommended the work on ILNP be pursued toward a routing architecture in which multihoming will be one of the main features [23].

## V. CONCLUSION

This paper presents a review of active multihoming solutions for IPv6. Although a large number of solutions have been proposed for this problem, few of them satisfy necessary technical requirements and therefore have a chance to be chosen, by the technical community, as the standard solution. We summarized and analyzed seven important solutions, which are active in this area. Results of our analysis show that each solution has its own drawbacks and weak points so that it is difficult to choose one of them as "the perfect solution". On the other hand, some characteristics, which are positive from technical viewpoint, do not seem to be easily deployable in the Internet. For example, considering number of modifications as a deployability parameter, host-based solutions need modifications only in one component: hosts. Technically, it might be possible to consider this class of solutions as "simply deployable" but such changes cannot be made without close cooperation of OS and networking software vendors. Also, end users should be convinced to pay the cost of such updates to their hosts. It seems that more research and effort is still needed for achieving a scalable, deployable, manageable and secure solution for IPv6 multihoming.

Copyright (c) IARIA, 2011 ISBN:978-1-61208-002-4 

| Solution / Characteristic | LISP | ILNP | NAT66 | MPTCP | HIP | NBS | SHIM6 |
|---|---|---|---|---|---|---|---|
| Product Modifications | ER | H, SP, A*, ER* | None | H | H, A, SP* | H, A | H, A* |
| Security(Compare to BGP4) | Similar | Stronger | Similar | Similar | Stronger | Similar | Similar |
| TE (Compare to BGP4) | Stronger | Similar | Weaker | Weaker | Weaker | Weaker | Weaker |
| Incremental Deployment | Possible with Conditions | Possible | Possible | Possible | Possible with Conditions | Possible | possible |
| Renumbering without breaking established communications | Possible | Possible | Impossible | Possible with Conditions | Possible | Possible | Possible with Conditions |
| New Component | Mapping System | None | NAT Device | None | PKI* | None | None |

*: optional A: Application ER: Edge Router H:Host SP: Services and Protocols

Figure 1. Summary of characteristics of the discussed solutions

## REFERENCES


[1] L. Pouzin, Interconnection of packet switching networks, 7th Hawaii International Conference on System Sciences, Supplement, 1974.

[2] J. Abley, K. Lindqvist, E. Davies, B. Black, and V. Gill, IPv4 Multihoming Practices and Limitations, Internet RFC 4116, July 2005.

[3] B. E. Carpenter, Observed Relationships between Size Measures of the Internet, ACM SIGCOMM CCR, 39(2) (April 2009).

[4] T. Bu, L. Gao, and D. Towsley. On characterizing BGP routing table growth. Computer Networks, 45(1):45–54, 2004.

[5] X. Meng, Z. Xu, B. Zhang, G. Huston, S. Lu, and L. Zhang, IPv4 Address Allocation and the BGP Routing Table Evolution, ACM SIGCOMM Computer Communication Review, 35(1), 2005.

[6] J. Abley, B. Black, and V. Gill, Goals for IPv6 Site-Multihoming Architectures, Internet RFC 3582, August 2003.

[7] G. Huston, Architectural Approaches to Multi-homing for IPv6, Internet RFC 4177, September 2005.

[8] P. Savola and T. Chown, A Survey of IPv6 Site Multihoming Proposals, 8th International Conference on Telecommunications (conTEL), 2005.

[9] C. de Launois and M. Bagnulo, The Paths Toward IPv6 Multihoming, IEEE Communications Survey 8 (2006) 38-50.

[10] S. Sugimoto, R. Kato, and T. Oda, A Comparative Analysis of Multihoming Solutions, IPSJ SIG Technical Report, 2006.

[11] J. Bi, P. Hu, and L. Xie, Site Multihoming: Practices, Mechanisms and Perspective, Future Generation Communication and Networking (FGCN) 1 (2007) 535-540.

[12] R. Clayton, Internet Multi-Homing Problems: Explanations from Economics, Eighth Annual Workshop on Economics and Information Security (WEIS09), London, UK, June 24-25, 2009.

[13] S. Deering, The Map & Encap Scheme for scalable IPv4 routing with portable site prefixes, presentation at IETF35, Los Angeles, March 4-8, 1996.

[14] R. Hinden, New Scheme for Internet Routing and Addressing (ENCAPS) for IPNG, Internet RFC 1955, June 1996.

[15] M. O'Dell, 8+8 - An Alternate Addressing Architecture for IPv6, Internet Draft (work in progress), 1996.

[16] K. Egevang and P. Francis, The IP Network Address Translator (NAT), Internet RFC 1631, May 1994.

[17] F. Baker, A Business Model For Metro Addressing, Internet Draft (work in progress), 2001.

[18] R. Stewart, Stream Control Transmission Protocol, Internet RFC 4960, September 2007.

[19] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson, Host Identity Protocol, Internet RFC 5201, April 2008.

[20] List of Internet-Drafts relevant to the Multi6-WG, http://ops.ietf.org/multi6/draft-list.html (last visited 2010-02-24)

[21] E. Nordmark and M. Bagnulo, Shim6: Level 3 Multihoming Shim Protocol for IPv6, Internet RFC 5533, June 2009.

[22] D. Meyer, L. Zhang, and K. Fall, Report from the IAB Workshop on Routing and Addressing, Internet RFC 4984, September 2007.

[23] Li, T. (ed.), Recommendation for a Routing Architecture, Internet Draft (work in progress), 2010.

[24] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, Locator/ID Separation Protocol (LISP), Internet Draft (work in progress), 2010.

[25] D. Lewis, D. Meyer, D. Farinacci, and V. Fuller, Interworking LISP with IPv4 and IPv6, Internet Draft (work in progress), 2009.

[26] R. Atkinson, ILNP Concept of Operations, Internet Draft (work in progress), 2008.

[27] M. Wasserman and F. Baker, IPv6-to-IPv6 Network Address Translation (NAT66), Internet Draft (work in progress), 2010.

[28] A. Ford, C. Raiciu, and M. Handley, TCP Extensions for Multipath Operation with Multiple Addresses, Internet Draft (work in progress), 2009.

[29] M. Komu, M. Bagnulo, K. Slavov, and S. Sugimoto, Socket Application Program Interface (API) for Multihoming Shim, Internet Draft (work in progress), 2009.

[30] J. Ubillos, M. Xu, Z. Ming, and C. Vogt, Name-Based Sockets Architecture, Internet Draft(work in progress), 2010.

[31] J. Arrko and I. Van Beijnum, Failure Detection and Locator Pair Exploration Protocol for IPv6 Multihoming, Internet RFC 5534, June 2009.

[32] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, LISP Alternative Topology (LISP+ALT), Internet Draft(work in progress), 2010.

[33] M. Bagnulo, Threat analysis for Multi-addresses/Multi-path TCP, Internet Draft (work in progress), 2009.

[34] K. Ramakrishnan, S. Floyd, and D. Black, The Addition of Explicit Congestion Notification (ECN) to IP, Internet RFC 3168, September 2001.

[35] T. Henderson, P. Nikander, and M. Komu, Using the Host Identity Protocol with Legacy Applications. Internet RFC 5338, September 2008.

[36] M. Bagnulo, A. Garcia-Martinez, and A. Azcorra, BGP-like TE Capabilities for SHIM6, Proceedings of the 32nd EUROMICRO-SEAA'06, 2006.


Copyright (c) IARIA, 2011 ISBN:978-1-61208-002-4 