Publication I

Bengt Ahlgren, Jari Arkko, Lars Eggert, Jarno Rajahalme. A Node Identity Internetworking Architecture. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications Workshops, Global Internet Symposium*, Barcelona, Spain, April 2006 [7].

A Node Identity Internetworking Architecture

Bengt Ahlgren, Jari Arkko, Lars Eggert and Jarno Rajahalme

Abstract — **The Internet consists of independent networks that belong to different administrative domains and vary in scope from personal area networks, private home networks, corporate networks to ISP and global operator networks. These networks may employ different technologies, communications mediums, addressing realms and may have widely different capabilities. The coming years will add a significant level of dynamic behavior, such as mobile nodes and moving networks, which the Internet must support. At the same time, there is a need to address the increasing levels of harmful traffic and denial-of-service attacks. The existing Internet architecture does not support dynamic behavior or secure communication to a sufficient degree. This paper outlines a node-identity-based internetworking architecture that allows heterogeneous networks to work together without loss of functionality. Some of techniques employed in this architecture include reliance on cryptographic node identifiers, identity routers and localized addressing realms.**

I. Introduction

THE Internet consists of independent networks that belong to different administrative domains. The Internet architecture allows these individual networks to use different underlying communication technologies by imposing a uniform internetwork protocol that provides a transparent end-to-end network service. However, within the last ten years, the assumptions of the architecture are being challenged by new problems and attempts to solve them, exposing fundamental shortcomings which can not be solved by just adding new functionality.

The most obvious example is the introduction of *Network Address Translators* (NATs) [1] that have established isolated, private, partially overlapping addressing domains in the Internet. They have been extremely beneficial to alleviate the shortage in available IP address space and have allowed domains to independently manage their local operation. However, NATs have also severely restricted the types of end-to-end communication that the Internet can support, due to the need to inspect and modify high-layer protocol information at domain boundaries. Historically, NATs have significantly delayed the introduction of new services, protocols and applications, such as voice-over-IP (VoIP) solutions. NATs encode knowledge about higher-layer protocol internals that must be updated to enable traversal of new applications and protocols, which is not feasible on an Internet-wide scale.

Approaches to overcome the need for NATs have concentrated on deploying a larger global address space and on redesigning applications to work despite the barriers imposed by NATs. The first approach, deployment of IPv6, has been largely unsuccessful to date. One cause appears to be that the introduction of IPv6 causes some of the same types of problems it was designed to solve: it divides the global network into IPv4 and IPv6 parts that cannot easily communicate with each other. The second approach, engineering applications for NAT traversal, often reduces functionality, introduces security issues, decreases efficiency and requires duplicating the same complex, brittle techniques for every new application.

Even if a sufficiently large global address space became available, the Internet no longer provides a medium for completely unrestricted end-to-end communication. Firewalls and other "middleboxes" [2] limit arbitrary communication based on domain-specific and oftentimes conflicting policies, such communication directionality or inferred application use. As a result, the ability to communicate depends on specific end-to-end paths, which causes significant problems with in increasingly dynamic Internet.

A second example of a challenge to the current architecture is the emerging trend of dynamic behavior of nodes and networks which involves mobile nodes, mobile networks and domains that use multi-homing for load-balancing or fault-tolerance. A lot of effort has already gone into creating specific extensions for some of these uses. However, these extensions rarely work across multiple domains, particularly if domains use different network protocols or address spaces. Furthermore, they often do not cleanly integrate with one another, leading to a complex and brittle architecture. Dynamic behavior also imposes new requirements on an internetworking architecture, beyond the simple survivability of communication across movements. One example is the necessity to protect the privacy of a user's geographical and topological location as he or she moves around.

A third example is the steady increase in various forms of harmful traffic, which is hard to mitigate in the current architecture. From an internetworking perspective, denial-of-service attacks on individual nodes, networks or the distributed infrastructure are most interesting. A number of simple defenses are remarkably effective, such as protection mechanisms against TCP SYN flooding [3]. However, existing defenses are typically incapable of addressing attacks that simply overload links with network traffic. At the same time, malware that spreads through email and viruses is making large-scale distributed attacks possible.

This paper presents the node identity internetworking architecture, which provides a solid foundation for a new Internet architecture addressing the shortcomings of the current. Others have proposed several of the ideas used in the architecture. The contribution of this paper lies in making the ideas more concrete, discussing assumptions and design tradeoffs and combining the ideas into a consistent whole.

The next two sections define the goals and describe the assumptions for the new architecture. Section IV describes the

Manuscript received March 29, 2006. Bengt Ahlgren is with the Swedish Institute of Computer Science (SICS) in Sweden (email: bengt.ahlgren@sics.se). Jari Arkko is with Ericsson in Finland (email: jari.arkko@ericsson.com). Lars Eggert is with NEC Network Laboratories in Germany (email: lars.eggert@netlab.nec.de). Jarno Rajahalme is with the Nokia Research Center in Finland (email: jarno.rajahalme@nokia.com).

node identity architecture, followed by discussion and related work in Section V and conclusions and future work in Section VI.

II. Goals for a New Architecture

Many of the existing attempts at improving the Internet architecture have tried to incrementally "patch" it through a series of relatively *ad hoc* techniques that each solve a specific issue. Unsurprisingly, the result has not been a unified, integrated, flexible and extensible platform for future internetworking. The approach presented in this paper, *the node ID architecture*, is different in that multiple technologies, address domains and various middleboxes are first-order components of the architecture, and not just nuisances that need to be worked around. The goal is to build an architecture that can work across multiple heterogeneous domains, can treat dynamic changes in a scalable manner and increase the level of protection against privacy disclosures or denial-of-service attacks. The node ID architecture employs cryptographic node identifiers, gateways, distributed hash tables and specific protocol constructs to address these issues.

Based on the deployment experience with IPv6, a new internetworking architecture needs to provide (1) very strong incentives for migration and deployment and (2) significant benefits for adopters even during partial deployment. Otherwise, widespread adoption of the new architecture is unlikely. Some of the new features that may create migration incentives include built-in security and mobility features, support for independent evolution of individual components and domains as well as more expressive communication primitives that enable new applications and services.

A new internetworking architecture should have stronger, integrated, security and privacy mechanisms. The need for secure bindings between names at various levels is only recently receiving attention in the Internet community, and many proposals are localized patches for specific functions that do not integrate into a coherent whole. However, the underlying research, such as cryptographic identifiers, can be readily adapted to a new internetworking architecture.

Another important feature is native support for mobility and moving networks. The current Internet supports them only with add-on mechanisms, which make them less efficient, more complex, and less secure compared to a built-in solution.

III. Assumptions

This section describes the main assumptions that underlie the node ID architecture. One key assumption is the existence of independent *locator domains*, or *domain* for short. A locator domain (LD) has a consistent internal addressing and routing system. Nodes within one LD can freely communicate, only relying on internal services of the LD. This definition is relatively flexible, *e.g.*, the global IPv4 network (excluding private, NAT'ed, subnets) can be seen as one LD, or each autonomous system can be seen as a separate LD, or an isolated *ad hoc* network may be seen as an LD. The key requirement is that within one LD, communication does not depend on the presence of outside internetworking mechanisms. In addition, different LDs may employ different networking technologies. LDs consequently may also establish technology boundaries.

An alternative to an integration of separate, heterogeneous, locator domains is the use of a common, global, locator space. The creation of a completely new locator space is problematic, due to the administrative requirements and political implications it introduces. Furthermore, the allocation of administratively assigned, topologically significant locator spaces can be problematic in private domains. A second choice is to reuse an existing locator space. The IPv4 address space is insufficient, due to the predicated shortage of available addresses within 5-10 years [13]. Consequently, a combination of the existing IPv4 and IPv6 locator spaces, including both their private and public subspaces, is most suitable. To enable transparent end-to-end communication, however, it is necessary to enable nodes to contact each other across different domains.

A second assumption is that connectivity between locator domains is dynamic. That is, the existence and characteristics of connectivity between individual locator domains may change dynamically on relatively short timescales, *e.g.*, due to routing changes, multi-homing or mobility events of nodes or networks. A related assumption is that the connectivity that binds individual nodes into one or more locator domains may similarly change. Nodes may move from some locator domains to others, while concurrently remaining part of other locator domains. The characteristics of the connectivity that binds a node into its locator domains may also dynamically change. Generally, for connectivity between locator domains and between nodes and locator domains, these events happen independently from each other.

A third assumption is that the distinction between hosts as the sources and sinks of traffic and routers as pure forwarding relays will cease to exist. Although core routers in future networks will continue to focus on data forwarding, hosts located at the edge of the network may temporarily or permanently relay traffic. This is due to the more dynamic future nature of the edge topology, where hosts may provide network access for other nodes and networks within their neighborhood. Examples include mobile phones that act as routers in personal area networks or servers that act as forwarding agents for mobility or multi-homing purposes.

With these assumptions, the "internetworking problem" can be defined as providing end-to-end connectivity between nodes across a dynamically changing federation of interconnected locator domains and nodes.

Traditionally, the Internet has solved this problem through the introduction of a single, global namespace together with uniform namespace management procedures and a complementary but mandatory routing mechanism, namely IPv4 with the *Border Gateway Protocol* (BGP). Both are strong invariants of the Internet architecture [22] and have limited its evolution for years.

IV. Node Identity Architecture

This section presents the node ID architecture, starting with an example and showing how routing, identification and name resolution occur. The section then continues to discuss advanced topics, including mobility, multi-homing, security and deployment. Due to space limitations, this paper only outlines the key components of the node ID architecture in each of these areas. A future revision will describe these mechanisms in detail.

Figure 1 shows three locator domains. One of these – LD1 – is the core IPv4 domain (= the existing Internet using global IPv4 addresses). Two communicating nodes A and B attach to the other domains (LD2 and LD3, respectively), which in turn connect to the core. All nodes have a cryptographic *node identity* (NID); a NID is the public key of a public/private key pair. The node ID architecture uses NIDs similarly with how HIP [8] uses host identities: decoupling node identities from their network locations and providing a firm foundation for security. In addition, NIDs are used to bridge across domains.

Figure 1 also shows two *NID routers* (NR2 and NR3) that connect their respective domains to the core. These NID routers map communications across the border by translating between different locator spaces and connectivity technologies. NID routers also serve as contact points for establishing communication with other nodes. In their most efficient form, NID routers only forward the initial signaling messages for establishing communication with a peer. However, NID routers may also need to remain involved in the forwarding of data, to provide features such as location privacy, denial-of-service protection or translation between domain protocols.

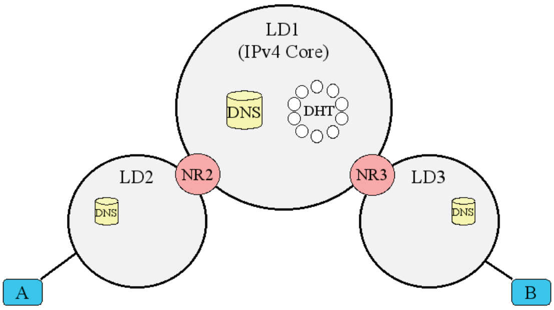

Figure 1. Example topology.

Nodes that join a domain obtain locators using technology-specific mechanisms, such as stateless address auto-configuration. The nodes also create records in the infrastructure to allow internetworking across domains to work, by registering their fully-qualified domain names (FQDNs), NIDs and locators with the local Domain Name Service (DNS) and NID routers. These registrations occur in much the same way that Internet hosts currently use DHCP to acquire addresses. The registrations may also be forwarded further up the domain hierarchy. Locators of NID routers in the local domain become the contact locators for local nodes when nodes in other domains initiate communication to them. This recursive process stops at the core. All core NID routers are part of a single distributed hash table (DHT) that enables them to discover each other's locators.

Establishing communication begins by resolving FQDNs. In Figure 1, node A attempts to contact node B that has registered the FQDN "B.EXAMPLE.COM" and receives NIDs for both node B itself and the globally reachable NID router, which stores node B's registration, *i.e.*, NR3 in this example. Because A has no knowledge of locator bindings for these NIDs, it sends its first packet to its own NID router, NR2. This packet, illustrated in Figure 2, includes the destination NID along with the NID router information returned from the DNS. In this example, NR2 does not know the NIDs contained in the packet and hence performs a lookup. It retrieves the locators for NR3 from the core DHT and forwards the packet to it. Upon receiving the packet, NR3 recognizes the destination NID as belonging to one of its locally registered nodes and forwards the packet to B.

A. Hybrid Routing

The NID routers are essentially creating an overlay network on existing infrastructure, that is, a new internetwork layer based on the node ID namespace. To provide connectivity, every namespace needs either its own routing system or a mapping function (name resolution) to a locator namespace that provides routing for it.

IPv4 Header	Node ID Header	ESP	Payload
Destination = NR2	Destination NID = B Destination NR = NR3	...	...

Figure 2. Logical packet format.

The obvious approach – routing solely on node IDs – has two drawbacks. First, it does not leverage any existing routing mechanisms that the individual locator domains may provide. Second, designing such a routing system is a challenging task, which will likely face scalability issues. However, assuming a loosely-coupled set of independent locator domains, it is not possible to solely rely on locator routing either.

For these reasons, the node ID architecture takes a hybrid approach. If communication stays within one domain and involves only static nodes, NIDs are resolved into locators on which the underlying domain performs routing. On top of this, the NID routers support mobility as well as cross-LD communication by implementing a NID-based routing mechanism.

The hybrid approach alone does not reduce the scalability problems of NID routing significantly. In the general case, *i.e.*, without core domains, the network topology consists of more or less arbitrarily connected locator domains, similar to the Internet's autonomous system topology. The requirement to support dynamic, moving domains within this system creates a routing problem that is far more difficult than the problem that the Internet's Border Gateway Protocol (BGP) solves today. The node ID architecture therefore assumes that a few core locator domains exist that remain fairly static. The non-core locator domains, which can be dynamic, attach in tree-like structures to the edges of the core domains, either directly or through other non-core domains. These "edge trees" can also dynamically change their topology. This design separates the NID routing problem into three distinct regions: the edge trees, the cores and routing between the cores.

In the first region, within the edge trees, all domains have well-defined default routes up the tree to a core. A corresponding reverse path is implemented when a node registers and its registration propagates recursively up to the core. This design eliminates the need for a global routing protocol and can be extended to support arbitrarily complex edge topologies, if a NID routing protocol can be designed to support it. Because edge topologies are assumed to cover limited number of nodes, NID-based forwarding is feasible even though state cannot be aggregated.

In the second region, within a core, routing is complex, because no default routes exist. Support for multiple cores makes it infeasible to use locators as global references for NID routers, so the core routing also utilizes NIDs, but covers only the nodes directly attached to the core LDs. The node ID architecture uses distributed hash table to organize core NID routers in a single system. In Figure 1, the initiating node A needs to provide the NIDs of both the destination node B and B's core NID router NR3 to initiate communication. These NIDs are in effect partial source routes. Without these NID router referrals, the core DHT would need to store bindings for potentially billions of individual destination NIDs.

Finally, in the third region, between multiple cores, routing utilizes the fact that the core DHT spans across all core domains and the bindings for NID routers indicate to which core they belong. Each core NID router needs to maintain routes towards NID routers that connect to other cores. Traffic destined for the other cores is forwarded along those routes.

The relation between the number of NID routes in the edge NID routers and the number of NID routers in the core DHTs will determine the overall scalability and performance of the internetworking system.

B. NID Resolution

NID resolution begins with a FQDN, which nodes resolve via the DNS. The DNS query may be resolved locally or it may be forwarded towards the core. In the latter case, the NID routers perform the necessary address translations – communication with DNS servers in other domains is identical to any other communication. If a DNS lookup returns only a peer's locators, then communication uses domain-specific, locator-based protocols and mechanisms. This "legacy" communication requires both peers to reside either in the same domain, or in other domains connected by a default path, if address spaces do not overlap or domains use different network technologies along the path. However, if the DNS resolution returns NIDs (using record formats similar to those defined for HIP [10]), communication between arbitrary domains becomes possible, due to NID routing. If the DNS result contains multiple NIDs, they establish a loose source route towards the destination.

Within a locator domain, the DNS lookup results in a complete mapping from FQDNs to both locally registered NIDs and local locators. This extends up to the entire edge tree of locator domains, making all NIDs registered in the tree reachable within the edge tree. For other NIDs, a default path towards the cores is used. The NID-to-locator resolution in the cores uses the global NID router DHT.

In all cases, registration security utilizes the cryptographic properties of the NIDs. For end-to-end connectivity, the end systems can establish security associations and communication context based on the NIDs. This enables communication in the presence of route changes towards a node.

All NID routers between different locator or technology domains (such as IPv4 and IPv6 Internets) must remain in the communication path to relay future data traffic with address or protocol translation. NID routers within a domain may redirect the communication to another NID router, which eliminates the need for the former NID router to remain involved in the established communication.

C. Mobility and Multihoming

The node ID architecture uses three complementary techniques to support mobility and multihoming (see Figure 3). First, hierarchical registration through levels of domains hides the local addresses of a node from peers outside the local domain; peers only see the locators of NID routers. In addition to hiding locator changes when nodes move, this allows NID routers to provide network-based multihoming when nodes have multiple registrations in different branches of an LD tree. In contrast, Hierarchical Mobile IP [6] only supports movements and SHIM6 [12] only supports host-based multihoming.

Second, delegation of registrations and end-to-end updates enable support for moving networks, because nodes need no longer be aware of where their NID router attaches [11]. Finally, using their NIDs, nodes can update their locator information with end-to-end signaling (similarly with HIP [9]). This is useful when peers are within the same domain or when their core NID routers change. Such end-to-end changes also propagate faster than the updates of multiple routers in the infrastructure.

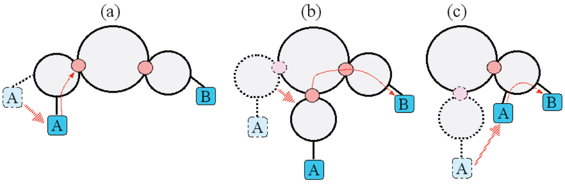

Figure 3. Local (a), network (b), or end-to-end mobility updates (c).

D. Security

The node ID architecture addresses security issues related to privacy, traffic redirection, denial of service, and supports an "always on" model for security. Security is largely based on the NIDs. After peers learn each other's identities, they bind their communication to these identities and can protect all communication by default. This allows simple authentication of signaling, for example, when a peer changes locators.

The initial exchange provides basic denial-of-service protection, by not requiring the contacted node to allocate state or commit resources until the connection attempt has been proven to be valid. NID routers add another layer of protection. Nodes can maintain multiple NIDs in such a way that they make some of them publicly known whereas they only communicate others to current, validated peers. In an attack or flash crowd situation, the NID routers can be asked to deregister NIDs under attack, rate-limit traffic sent to them or handle a part of the initial exchange to prevent bogus connection attempts from reaching their target at all.

Finally, the use of multiple and changing NIDs allows nodes to control privacy on a per-connection or per-peer basis. The use of NID routers provides location privacy.

E. Stateless and Stateful Modes

Gateways in past designs are either stateful (*TurfNet* [17]) or stateless (i^3 [15]). The node ID architecture recognizes this tradeoff between the *a priori* cost of setting up state and the ongoing cost of carrying sufficient information in some packets to operate statelessly. The architecture does not mandate a single approach but instead leaves it to individual nodes to choose either approach on a per-communication basis.

State setup requires an explicit registration request from a node to the involved NID routers. The node proves its right to create state by demonstrating ownership of its NID. The result is soft state that the node needs to refresh periodically. Gateways may also inform nodes when their state has been lost.

Stateless communication requires carrying sufficient information in some packets to create the necessary mapping without an explicit set-up process. This information includes source and destination NIDs and, where needed, NIDs for the gateways serving those nodes. Stateless communication is useful for short-lived exchanges that involve too few packets to justify the delay incurred by an explicit registration.

V. Discussion and Related Work

The first part of section discusses implications of the node ID architecture. The second part contrasts the node ID architecture to a selected number of related proposals.

It is important to stress that the mechanisms of the node ID architecture can be implemented in different ways, depending on the requirements of a domain. For example, the mechanism provided by a NID router can be implemented in a single physical device, or it can be implemented as a distributed system of collaborating devices that are located at different places along the edge between two domains. In the latter case, the distributed NID router can automatically assign a device or set of devices to be responsible for a particular NID and can redirect communication accordingly. This distributed operation allows handling of larger traffic volumes and is also more resistant against denial-of-service attacks, because it introduces an additional, domain-internal indirection between two communicating peers and thus enables migration strategies similar to *Secure i^3* [16]. Nodes that have multiple NIDs can request the distributed NID router to filter or disable individual NIDs they detect to be under attack while still being reachable at different NIDs to other communication partners.

In “disconnected” operation mode, a single domain or a cluster of interconnected domains has no path to the core domains. Mobility events or other transient events may result in short disconnections. Other events can cause disconnections that can last days or are completely indeterminate. Likewise, the size of the cluster of disconnected domains varies from a single domain to large fractions of the overall topology. The node ID architecture must support efficient communication within disconnected clusters as well as within the remaining network that is in contact with the cores. The basic mechanism that the architecture uses to support this is the local routing within the domains as well as the hierarchical NID routing. A tree can operate without connectivity to the core, as long as the communications stay within the tree and the tree structure is static. Tree reorganization and *ad hoc* routing techniques would be able to enhance this capability, but their application to the architecture remains future work.

A third implication worth pointing out is that the basic form of communication in the node ID architecture is secure and authenticated unicast transmission of individual packets from one node to another. However, it can be extended to support other forms of communication. NID routers can provide the means for secure multicast and anycast communication by forwarding the same packet to multiple or alternative recipients. Security for these communication primitives uses can be based on reception delegation from the owner of a NID to a set of peers, authorized through signatures using the private part of the NID, similar in spirit to Anderson *et al* [23]. These mechanisms may also be able to take advantage of direct support for such communication primitives in traversed locator domains.

The remainder of this section contrasts the node ID architecture to other relevant proposals that aim to solve similar problems. The IETF is currently developing a number of specific extensions to the current Internet architecture, including *Mobile IP* [4][5], *MOBIKE* [7], *HIP* [8], *SHIM6* [12] and others. They all follow the same basic approach that provides host-based mobility solutions that avoid injecting movement information into the routing system. Because only the communication endpoints need to be aware of movements, they scale well. Gateway implementations can support many of these solutions, making it easy to communicate with hosts that do not have the necessary protocol extensions. However, these solutions still have two main limitations. First, not all of them can operate across different addressing domains, across domain borders or across different protocols. For instance, *SHIM6* solutions are limited to IPv6, *Mobile IP* is only now being extended to support communication between IPv4 and IPv6, all solutions have limitations with respect to passing through NATs and discovering connectivity beyond NATs, and typically provide only limited location privacy. Finally, none of the proposed solutions have inherent denial-of-service mitigation mechanisms.

Several related research efforts have also investigated new approaches to internetworking. They include the *Host Identity Indirection Infrastructure (Hi^3)* [14], a combination of *HIP* [8] and the *Internet Indirection Infrastructure (i^3)* [15][16], *TurfNet* [17], the *Split Naming/Forwarding* (*SNF)* architecture [18], the *FARA* architecture [20], the *Layered Naming Architecture* [19], *IPNL* [24], *8+8* [25], *TRIAD* [26] and *Plutarch* [21], among others. Due to space limitations, this paper can only briefly compare these systems to the node ID architecture.

Although these architectures support some of the same mechanisms and scenarios as the node ID architecture, significant differences exist. Some of them do not support internetworking across heterogeneous locator domains (*Hi^3*, *i^3*), use a large number of proxy locators to forward data instead of node identities (*TurfNet*), require significant communication state (*TurfNet*), or are high-level framework architectures with no defined realization (*Plutarch, FARA, Layered Naming*).

VI. Conclusion and Future Work

This paper has presented the node ID architecture, a new internetworking architecture for independent networks that may employ different technologies, communications mediums and addressing realms and may have widely different capabilities. The Internet Protocol was originally designed to solve approximately the same problem as the node ID architecture, *i.e.* bridging across heterogeneous network domains, but has since failed to provide this service due to the divergent evolution of parts of the architecture. This has resulted in a loss of end-to-end transparency. Other proposed solutions, such as the move to a larger global address space or *ad hoc* extensions to mitigate specific isolated issues are ineffective in creating a unified, extensible platform for future internetworking.

The key design elements of the node ID architecture include independent locator domains, reliance on cryptographic self-managed NIDs, routing based on both locators within domains and NIDs or default routes between domains, router referrals to avoid a single administration for the whole network, and end-to-end security based on the NIDs.

An extended version of this paper is currently being prepared to describe the mechanisms of the architecture in detail. In parallel, the EU-supported *Ambient Networks* project is prototyping and evaluating an implementation of the architecture. One focus of this effort is an evaluation of the performance of existing transport protocols over a new internetworking layer. The results of this analysis may guide the development of transport enhancements for dynamic internetworking.

Acknowledgment

The authors would like to thank their colleagues in the architecture group of the *Ambient Networks* project, especially Robert Hancock, Börje Ohlman and Anders Eriksson, for their comments and feedback on this effort.

This paper is a product of *Ambient Networks*, a research project partly supported by the European Commission under its *Sixth Framework Program*. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the *Ambient Networks* project or the European Commission.

Bengt Ahlgren is also partly supported by the *Winternet* research program funded by the Swedish Foundation for Strategic Research.

References

[1] Pyda Srisuresh and Matt Holdrege. IP Network Address Translator (NAT) Terminology and Considerations. *RFC 2663*, August 1999.

[2] Brian Carpenter and Scott Brim. Middleboxes: Taxonomy and Issues. *RFC 3234*, February 2002.

[3] Jonathan Lemon. Resisting SYN flood DoS attacks with a SYN cache. Proc. *BSDCon*, San Francisco, CA, USA, February 2002, pp. 89-98.

[4] Charles Perkins (*ed.*) IP Mobility Support for IPv4. *RFC 3344*, August 2002.

[5] David B. Johnson, Charles E. Perkins and Jari Arkko. Mobility Support in IPv6. *RFC 3775*, June 2004.

[6] Hesham Soliman, Claude Castellucia, Karim El-Malki, and Ludovic Bellier. Hierarchical Mobile IPv6 Mobility Management Protocol (HMIP). *RFC 4140*, August 2005.

[7] Pasi Eronen. IKEv2 Mobility and Multihoming Protocol (MOBIKE). *Internet Draft* draft-ietf-mobike-protocol-07 (Work in Progress), December 2006.

[8] Robert Moskowitz and Pekka Nikander. Host Identity Protocol Architecture. *Internet Draft* draft-ietf-hip-arch-03 (Work in Progress), August 2005.

[9] Thomas Henderson. End-Host Mobility and Multihoming with the Host Identity Protocol. *Internet Draft* draft-ietf-hip-mm-02 (Work in Progress), July 2005.

[10] Pekka Nikander and Julien Laganier. Host Identity Protocol (HIP) Domain Name System (DNS) Extensions. *Internet Draft* draft-ietf-hip-dns-04 (Work in Progress), December 2005.

[11] Jukka Ylitalo. Re-Thinking Security in Network Mobility. Proc. *NDSS Workshop of Wireless and Mobile Security*, San Diego, CA, 2005.

[12] Geoff Huston. Architectural Commentary on Site Multi-homing using a Level 3 Shim. *Internet Draft* draft-ietf-shim6-arch-00 (Work in Progress), July 2005.

[13] Tony Hain and Geoff Huston. A Pragrmatic Report on IPv4 Address Space Consumption. *The Internet Protocol Journal,* Vol. 8, No. 3, 2005.

[14] Pekka Nikander, Jari Arkko and Börje Ohlman. Host Identity Indirection Infrastructure (Hi3). Proc. *Second Swedish National Computer Networking Workshop (SNCNW)*, Karlstad, Sweden, November 23-24, 2004.

[15] Ion Stoica, Daniel Adkins, Shelley Zhuang, Scott Shenker and Sonesh Surana. Internet Indirection Infrastructure. Proc. *ACM SIGCOMM*, Pittsburgh, PA, USA, August 2002, pp. 73-88.

[16] Daniel Adkins, Karthik Lakshminarayanan, Adrian Perrig and Ion Stoica. Towards a More Functional and Secure Network Infrastructure. *Technical Report No. UCB/CSD-03-1242*, EECS Department, University of California, Berkeley, CA, USA, 2003.

[17] Stefan Schmid, Lars Eggert, Marcus Brunner and Jürgen Quittek. Towards Autonomous Network Domains. Proc. *8th IEEE Global Internet Symposium*, Miami, FL, USA, March 17-18, 2005.

[18] Andreas Jonsson, Mats Folke and Bengt Ahlgren. The Split Naming/Forwarding Network Architecture. Proc. *First Swedish National Computer Networking Workshop (SNCNW)*, Arlandastad, Sweden, September 8-10, 2003.

[19] Hari Balakrishnan, Karthik Lakshminarayanan, Sylvia Ratnasamy, Scott Shenker, Ion Stoica and Michael Walfish. A Layered Naming Architecture for the Internet. Proc. *ACM SIGCOMM*, Portland, Oregon, USA, August 30 - September 3, 2004, pp. 343-352.

[20] Dave Clark, Robert Braden, Aaron Falk and Venkata Pingali. FARA: Reorganizing the Addressing Architecture. Proc. *ACM SIGCOMM Workshop on Future Directions in Network Architecture (FDNA)*, Karlsruhe, Germany, August 2003, pp. 313-321.

[21] Jon Crowcroft, Steven Hand, Richard Mortier, Timothy Roscoe and Andrew Warfield. Plutarch: An Argument for Network Pluralism. Proc. *ACM SIGCOMM Workshop on Future Directions in Network Architecture (FDNA)*, Karlsruhe, Germany, August 2003, pp. 258-266.

[22] Bengt Ahlgren, Marcus Brunner, Lars Eggert, Robert Hancock and Stefan Schmid. Invariants - A New Design Methodology for Network Architectures. Proc. *ACM SIGCOMM Workshop on Future Directions in Network Architecture (FDNA 2004)*, Portland, OR, USA, August 30 – September 3, 2004, pp. 65-70.

[23] Tom Anderson, Timothy Roscoe and David Wetherall. Preventing Internet denial-of-service with capabilities. *ACM Computer Communication Review* (CCR), Vol. 34, No. 1, January 2004, pp.39-44.

[24] Paul Francis and Ramakrishna Gummadi. IPNL: A NAT-Extended Internet Architecture. Proc. *ACM SIGCOMM*, San Diego, CA, USA, August 2001, pp. 69-80.

[25] Mike O'Dell. 8+8 - An Alternate Addressing Architecture for IPv6. *Internet Draft* draft-odell-8+8-00 (Work in Progress), October 1996.

[26] David Cheriton and Mark Gritter. TRIAD: A Scalable Deployable NAT-based Internet Architecture. *Stanford Computer Science Technical Report*, January 2000.