Names, Addresses and Identities in Ambient Networks

Bengt Ahlgren
Swedish Institute of Computer Science
Kista, Sweden
bengt.ahlgren@sics.se

Lars Eggert
NEC Network Laboratories
Heidelberg, Germany
lars.eggert@netlab.nec.de

Börje Ohlman
Ericsson Research
Kista, Sweden
borje.ohlman@ericsson.com

Jarno Rajahalme
Nokia Research Center
Helsinki, Finland
jarno.rajahalme@nokia.com

Andreas Schieder
Ericsson Research
Aachen, Germany
andreas.schieder@ericsson.com

ABSTRACT

Ambient Networks interconnect independent realms that may use different local network technologies and may belong to different administrative or legal entities. At the core of these advanced internetworking concepts is a flexible naming architecture based on dynamic indirections between names, addresses and identities. This paper gives an overview of the connectivity abstractions of *Ambient Networks* and then describes its naming architecture in detail, comparing and contrasting them to other related next-generation network architectures.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design: *Network Communications*. C.2.6 [**Computer-Communication Networks**]: Internetworking.

General Terms

Management, Performance, Design, Security.

Keywords

Ambient Networks, naming, addressing, identities, internetworking, bindings, indirection, resolution.

1. INTRODUCTION

Communication networks are tools that enable users to communicate and provide access to information and services. The networks and their intrinsic technologies are only facilitators; their operation and management should not concern the users. However, the situation today is different. Users must be familiar and distinguish between the technical and commercial specifics of networks of different types. In addition, access to these networks is often restricted due to security and business considerations. Usage requires pre-established, per-network subscriptions, although static, pre-established roaming agreements can extend the scope of these subscriptions to some other networks of the same technology. Furthermore, incompatibilities and inconsistencies between network functionality – especially in naming and addressing schemes – limit the potential usefulness of the available networks. The cur-

rent networking capabilities thus do not yet support a vision where services and information are the focus and access and networking details are transparent to the user. In essence, a more advanced internetworking mechanism than provided by the current usage of the Internet Protocol is needed.

The *Ambient Networks* project is addressing these challenges [6][7]. The project's main objective is to enable seamless interoperation between heterogeneous networks. *Ambient Networks* aim to establish this interoperation through a common control space distributed across the individual, heterogeneous networks and addressing domains. This new common control space functionality can be deployed both as an integral component of future network architectures and as an add-on to existing, legacy networks, enabling legacy interoperability.

This paper focuses on mechanisms of the *Ambient Networks* architecture that harmonize heterogeneous connectivity, bridge different addressing domains and provide a uniform internetworking environment for services and applications. *Ambient Networks* interconnect independent realms that may internally use different network technologies and may belong to different administrative or legal entities.

The project has developed a flexible and powerful framework of naming, addressing and identity mechanisms that enable dynamic bindings for supporting connectivity across heterogeneous network domains. These naming mechanisms can also provide advanced capabilities, such as delegation and indirection. They are the focus of this paper. Section 2 briefly outlines the connectivity abstractions that serve as a basis for integrating heterogeneous networks in one common control and naming framework. Section 3 briefly presents related research and Section 4 discusses the specifics of this framework in terms of the related research. Finally, Section 5 summarizes and concludes this paper.

2. NETWORK CONNECTIVITY ABSTRACTIONS

Ambient Networks enable interoperation of legacy networks by abstracting from the intricacies of legacy connectivity planes and by providing a number of common communication primitives to services and applications. *Ambient Networks* aim to establish this interoperation through a common control space distributed across the individual, heterogeneous networks and addressing domains. This common *Ambient Network* control space provides two "connectivity" abstractions that the naming framework tightly integrates. For a more detailed discussion of the connectivity abstractions, see [16].

Figure 1 illustrates how the *Ambient Network* control space incorporates the *bearer* and *flow* connectivity abstractions. The control space encompasses the middle two layers in the figure. The con-

trol space exposes the *bearer* abstraction to applications through the *Ambient Service Interface (ASI)*, its "upper" interface. It uses the *flow* abstraction to control the connectivity plane functions regardless of the underlying network technology; the "bottom" *Ambient Resource Interface (ARI)* provides the flow abstraction.
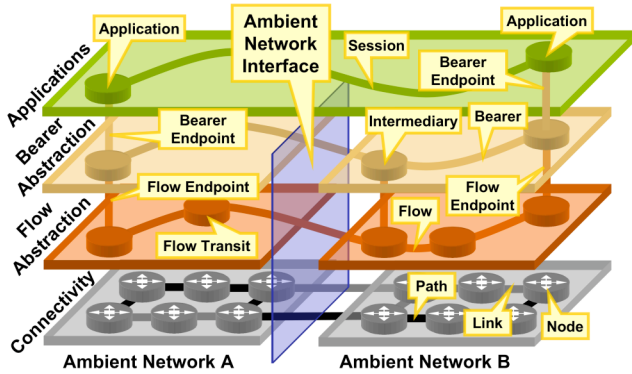

Figure 1: The *Ambient Networks* connectivity abstractions.

A flow is an abstract view of the connectivity provided by the underlying network technology. Flows are constrained to a single network technology and addressing domain; they terminate at these boundaries. A flow is a transfer of data between two instances of the ARI, where a technology dependent locator labels each *flow endpoint*. A flow may pass through intermediate resources not explicitly tied to the flow, but controlled through the ARI. These intermediate resources are *flow transits*. The set of intermediaries may change over the lifetime of a flow. The flow may also pass other nodes that are invisible and thus not controllable through the ARI.

The flow abstraction is comparable to the *communication substrate* of *FARA* [3]. However, important and deliberate differences exist. Flows are constrained to single network technology domains, whereas the communication substrate provides end-to-end connectivity across technology domains. The specific definition of a flow allows the internetworking functions to control the mapping at technology boundaries. The flow abstraction is not an interface – it is an abstraction of the connectivity provided by the underlying technology. In contrast, the communication substrate is an end-to-end communication primitive in its own right.

Whereas the flow abstraction provides an idealized, common connectivity plane for use by the control space functions, the *bearer* abstraction exposed through the ASI provides end-to-end connectivity primitives to applications and services, which may cross many *Ambient Networks*. The data transport functions in the control space construct bearers out of sequences of flows. Some flow transits provide bearer-level transport functionality and are visible as *intermediaries*.

Bearer endpoints, unlike flows, do not bind to locators but to higher-level entities in the naming framework. Bearers provide applications with the end-to-end customized transport service that supports all the functionality provided by the control space, such as mobility, address translation and media adaptation. For example, media delivery bearers may have properties that control content manipulation functions, such as transcoding. On top of the bearer abstraction, applications define *sessions* that may combine multiple bearers. Application-defined sessions are *not* part of the *Ambient Networks* connectivity framework and this paper mentions them for completeness only.

To summarize, flows abstract away from the underlying networking technology and provide a uniform view of the connectivity for the control space functions. Bearers combine sequences of flows into end-to-end transport primitives for application use. A dynamic mapping between flows and bearers maintained by the naming functions in the control space enables advanced internetworking functionality. The next section briefly discusses some related research in this area.

3. RELATED WORK

The Internet Protocol (IP) was originally designed to solve approximately the same problem as the *Ambient Networks* internetworking functions, i.e. bridging across heterogeneous network domains. The current Internet, however, does not meet several of the key design assumptions of IP anymore, causing it to fail to support end-to-end communication. IP addresses are no longer sufficient to establish end-to-end communication among all nodes, due to address space reuse enabled by network address translation. Widespread packet filtering has created a network that treats packets differently based on their content. In addition, the Internet has never supported advanced network capabilities, such as mobility or multihoming, and currently struggles to incorporate these functions in an architecturally clean way.

In mobile telephony networks, a large number of different identifiers are required to implement functions such as mobility support and subscriber handling, most of them having global significance. The *Mobile Station ISDN Number* (MSISDN) is part of the E.164 numbering plan [18]. It is the number dialed when attempting to establish a call. Telephone networks use the *International Mobile Subscription Identifier* (IMSI) or its local variant, the *Temporary Mobile Subscription Identity* (TMSI), together with the *Mobile Station Roaming Number* (MSRN) to route a call to the called terminal. Identifiers follow a hierarchical naming scheme that includes country codes, network codes and subscriber numbers. The rigid structure of these identifiers and the inherent locator properties of the MSISDN have proven suitable for basic call routing, but make the introduction of personal numbers and mobile number portability difficult. The concept of distinguishing between identifiers and locators in mobile telephony networks is thus regarded as effective and extensions to the Internet architecture also started to introduce a separation between locators and identities.

Mobile IP (MIP) [19] provides separation between a locator in the fixed Internet by introducing a *Home Address*, corresponding to the MSISDN, and the *Care-of Address*, a dynamically changing locator corresponding to the MSRN. However, Mobile IP does not clearly separate identifiers from locators, but allows transport protocols to continue using the locators as identifiers.

The *Host Identity Protocol (HIP)* architecture [5] defines a new global host identity namespace decoupling the name and locator roles currently filled by IP addresses. Host identities take over the naming role, while IP addresses become pure locators. With HIP, transport-layer mechanisms operate on host identities instead of using IP addresses as endpoint names. Network-layer mechanisms continue to use IP addresses as pure locators. Due to the introduction of a new global namespace, HIP also affects the Internet's name resolution services.

The *Host Identity Indirection Infrastructure (Hi³)* [8], a combination of *HIP* and the *Internet Indirection Infrastructure (I³)* [10] uses I³ for initial rendezvous and *HIP* for more direct, efficient and resilient communication afterwards. The basic naming scheme is the same as in *HIP*, but an improved resolution mechanism improves resistance against denial-of-service attacks. *Hi³* also enhances the *I³* concept of relaying all data through an indirection server by adding secure end-to-end communication. An added benefit of this addition is improved middlebox traversal.

TurfNet [11] is a novel internetworking architecture that enables communication among highly autonomous and heterogeneous network domains. The architecture uses a global identity namespace and does not require global addressing or a shared internetworking protocol. It integrates the concept of dynamic network composition with other recent architectural concepts, such as decoupling locators from identifiers. Addresses in *TurfNet* may not have end-to-end significance; they are merely transient routing tags for local routing. The architecture uses a name/identity registration and resolution process to find and set up the high-level routing path across a federation of composed networks. End-to-end communication across network boundaries is a combination of node registration, name resolution and packet relaying.

The *Split Naming/Forwarding* (*SNF*) architecture [4] separates node identities from locations. *SNF* is similar to other proposals in that it divides the network layer into a *forwarding* part and a *naming* (or identity) part. A unique property of *SNF* is that the naming layer can also *forward data*, not only resolve names. Because the naming layer spans different addressing domains, *SNF* bridges over them. To establish a path in the forwarding layer, the naming layer controls the translation points between the addressing domains. *SNF* uses three namespaces: fully-qualified domain names for node identities, IP addresses for identifying the location of nodes and ephemeral correspondent identifiers, which identify packet flows. The last namespace distinguishes *SNF* from other proposals – the packets do not contain node identities, but rather a connection identifier constructed at connection setup time.

The *Layered Naming Architecture* [1] borrows heavily from other proposals, such as *HIP*, and puts the pieces together into a more or less complete naming architecture. It defines four layers with three levels of name resolution in between: user level descriptors, service identifiers – used by applications to name services and data objects, endpoint identifiers – used to name end systems and finally IP addresses (or other locators) to name network locations. One unique principle of this architecture is that names only bind protocols to the relevant aspects of underlying structure. This facilitates independent re-mapping at all levels. The resolution of user level descriptors into service identifiers – persistent identifiers of a service that do not change when the hosting node moves – can be implemented in various ways. This architecture also supports the concept of *delegation*, where a network entity can direct resolutions of its name not only to its own location, but also to chosen delegates. This is a powerful principle that naturally accommodates intermediaries in the communication path without violating protocol layering.

The *NewArch* project describes the *FARA* addressing and routing architecture [3]. *FARA* embodies a new organization of the concepts of naming and binding that underlie the current Internet architecture. Communication in *FARA* occurs between pairs of *entities* (a generalization of applications) via logical linkages called *associations* (an abstraction of communication state related to a flow), using packet exchanges over a so-called *communication substrate*. *FARA's* main assumptions are that the entities are the units of mobility and that the introduction of a new global namespace should be avoided. *FARA* is a very general proposal and does not detail many of its functions.

Plutarch [12] concentrates on naming and addressing issues for establishing connectivity between heterogeneous networks by bridging dissimilar networking contexts into a cohesive network. The architecture proposes no new protocols or modifications to existing ones. Instead, it focuses on mechanisms that allow interoperation of heterogeneous networks without mandating a one-size-fits-all protocol suite. Plutarch introduces two fundamental concepts: *contexts* and *interstitial functions*. Contexts are groups of homogeneous network elements bridged together through interstitial functions that map between them. Communication occurs between endpoints within the individual contexts when interstitial functions map between the different sets of addressing, naming, routing and transport functionalities.

Other such proposals include *TRIAD* [13], *IPNL* [14] and 4+4 [15]. *TRIAD* is an internetworking architecture that addresses the lack of end-to-end connectivity caused by NATs through an explicit content layer. It uses identifiers rather than addresses for node identification and routing. TRIAD uses source routing and requires IPv4 in all network domains. *IPNL* and *4+4* aim at isolating independent IP subnetworks through loose integration. They use NATs to integrate networks with potentially overlapping address spaces to avoid renumbering.

4. NAMES, ADDRESSES AND IDENTITIES

The *Ambient Networks* naming architecture adopts a layered entity model that borrows from several recent proposals [1][5] as well as from some previous ideas [9], for example, the concepts of indirection and delegation. The model deliberately focuses on the named entities and not the names used *for* the entities. With a migration perspective in mind, the architecture design permits the use of multiple namespaces at a particular layer. Dynamic bindings at different layers enable the basic mobility of nodes, bearers and applications. This section defines and describes the entities that form the naming layers of *Ambient Networks*, discusses their relations and contrasts them to related work discussed in Section 3. Figure 2 illustrates the relation of these entities to the connectivity abstraction described above.
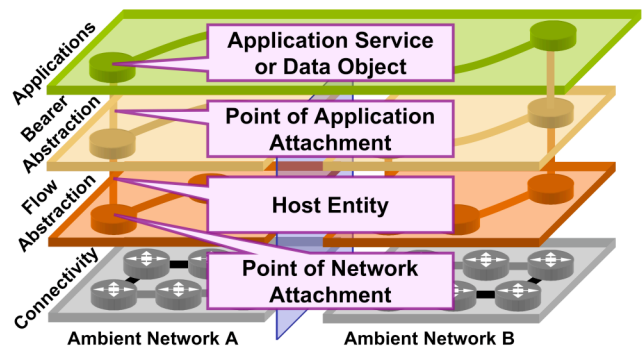

Figure 2: Entities in the naming architecture.

4.1 Applications and Points of Application Attachment

An *application service* or *data object* is an entity that is either an instance of a specific application service or a specific data object. The identity of the object persists over time and is not tied to the end system hosting the service or data. Examples are SIP services or web pages. An application is reachable to clients at its *points of application attachment*. They are located at the ASI and are comparable to standard TCP/IP sockets in the Berkeley API. A bearer, the upper-level connectivity abstraction described in Section 2, connects two points of application attachment with one another.

All related approaches from Section 3 avoid introducing changes at this level. *FARA* explicitly allows multiple user-level service namespaces. *Ambient Networks* is investigating this approach to allow application-specific service and data object names, including, for example, URLs, URIs or email addresses. The existing service namespaces can be seen as part of a single, bigger namespace that the *Domain Name System (DNS)* [17] bridges together through the use of DNS names as part of service names. Addition-

ally, other service namespaces support local significance of names. For reasons of interoperation with legacy networks, *Ambient Networks* avoids modifying the established relations between the current namespaces and the DNS.

4.2 Host Entities

A *host entity* resides in a node in a network. Hosts may be part of multiple networks at the same time. The identity of the host entity is independent of its current location(s) in the network. The host's locations are determined by the locators of the *points of network attachment* it attaches to with specific communication interfaces. The term host entity does not denote a physical machine – it identifies a logical entity that may share physical hardware with multiple other host entities. Chiappa's concept of an *endpoint* [2] and the *FARA entity* can be seen as similar to host entities. Depending on interpretation, they can correspond either to host entities or to points of application attachment.

Ambient Networks is considering to adopt an extension of *HIP* or a similar architecture as the host entity namespace solution. Note that the naming system does not attempt to provide support for frequent physical mobility, as exhibited in cellular systems today. Specific mobility management in access networks is the appropriate place to support such mobility events.

4.3 Points of Network Attachment

Host entities attach to the network at *points of network attachment*, which also define generic locations in the network topology. Locations are identified with some sort of *network address* or *locator*. These locators often depend on the network topology and technology used. Locators are exposed at the ARI. A flow, the lower-level connectivity abstraction, runs between two points of network attachment.

The related work in Section 3 has different approaches when dealing with locators. *FARA* proposes to consider the transport-layer port numbers as part of the forwarding directive. It adds an *Association ID* to multiplex between different communication instances existing between the entities at the same time. *SNF* proposes similar *ephemeral connection identifiers*. Other proposals favor keeping the existing port-based application point of attachment. For legacy interoperability, *Ambient Networks* attempts to maintain established use of port numbers, but relax the current usage model by allowing port numbers to change on the path between the communicating applications.

4.4 Dynamic Bindings

One purpose of defining a layered naming architecture is to provide dynamic bindings between entities at different levels. With dynamic bindings, names of entities become location independent. Furthermore, different types of mobility, such as for nodes and services, become possible without resorting to add-on mechanisms. *Indirection* and *delegation* mechanisms [1] can additionally support advanced mobility schemes, such as moving networks, and explicitly control middleboxes. Indirection means that the name of an entity does not bind "down" within the same node, but sideways to another location where an intermediary takes care of forwarding the communication to the entity's actual location. A simple application of this mechanism enables servers to operate behind a NAT without explicit configuration.

With delegation, one entity may represent another entity of the same kind at the same level. One example is a node in a moving network that binds its location to a designated gateway node. When the whole network moves, only the gateway needs to update its bindings to new network locations.

One open question is how many levels of indirection are required to support a desired functionality. *FARA* has one level of indirection; the *Layered Naming Architecture* has two. Multiple levels certainly add complexity and overhead and can result in lower performance. An attractive solution is an arbitrarily long chain of bindings that eventually terminate at a locator or set of locators. If the name resolution mechanism does not restrict the types of entities involved, arbitrarily complex bindings and indirections become possible, while still offering competitive performance in the simple case of a static service on a stationary node.

4.5 Bridging Across Different Addressing Realms

Two fundamental alternatives exist for bridging across different addressing (locator) realms or between realms that share an addressing realm but use overlapping regions of it. The methods are *translation* and use of a *common namespace*. These can be applied at different levels in the naming architecture.

With *translation*, gateways between networks translate the identifiers used at a particular level. The goal is to translate unique "foreign" locators of one domain into unique "native" ones of the other or to establish uniqueness between locators when both domains have overlapping locator spaces. NATs are a common example of bridging by translation.

Introducing a *common namespace* is another approach for bridging across different addressing realms. Here, identifiers used at a particular level belong to a common namespace shared by all networks. This method inherently establishes a uniform namespace for a new level of identities to which the local locators can be mapped and thus does not require translation. In the extreme, the common namespace is globally shared, for example, the Internet's global IPv4 address space. Note that a common namespace corresponds to the internetworking principle.

Independent of whether translation or a new common internetwork namespace provides internetworking, end-to-end communication requires the determination of end-to-end paths across the interconnected domains, *i.e.*, a routing mechanism. If the new namespace is flat, as in *HIP*, routing becomes challenging, because the routing information of nodes with flat names is inherently difficult to aggregate. The *Ambient Networks* project is considering two alternatives to approach this challenge.

In the first main alternative, the top level structure of the global network consists of arbitrarily connected network domains, much like the Internet's *Autonomous System* (AS) structure. The difference is that the network domains can use different internal addressing schemes. The top-level routing problem is thus similar to the one that *BGP* solves in the Internet, but with the difference that address prefixes are not used, just the equivalence of AS numbers and paths. Finally, the name resolution system needs to consult top-level routing information to be able to resolve names into addresses.

Again, two alternatives exist. Name resolution either results in a sequence of addresses, a partial *source route* that defines the path the packets take over possibly heterogeneous addressing domains or the resolution results in just the first hop in the address sequence – the process is then repeated at each address domain boundary in a *late binding* fashion.

In the second main alternative, a small and well-defined set of top-level routing structures is assumed, such as the current global IPv4 and IPv6 Internets. Networks that move or use other – *e.g.*, private – addressing schemes attach in tree-like structures to the top-level structures. These attaching networks always have well defined paths to the top-level, *i.e.*, they have a *default route*.

Note that even if a network attaches to a top-level structure of a certain kind, the network itself can use a different technology. Just as in the first main alternative described above, name resolution needs to know about the top-level routing topology. However, because that topology is very constrained, the top-level routing problem can be handled by three simple steps: (1) a default path can always be used to reach a top-level structure, (2) name resolution needs to know which top-level structure a node is attached to and (3) the top levels need to know how they are interconnected to each other. Name resolution here also can result in a partial source route or occur incrementally with late binding.

5. CONCLUSION

This paper presented the naming and addressing functions of the *Ambient Networks* architecture, which are critically important to provide end-to-end communication across heterogeneous internetworking environments. These functions integrate with other components of the architecture – the connectivity abstractions – to enable bridging across different internetworking technologies and hiding of network complexity. Connectivity abstractions hide the differences of heterogeneous internetworking technologies and enable applications to operate across them. A common naming framework enables end-to-end communication across otherwise independent internetworks and supports advanced networking capabilities, such as indirection or delegation, through dynamic bindings between named entities.

This paper described the current state of these mechanisms in the *Ambient Networks* architecture, focusing on the naming functionalities, including dynamic binding, indirection, delegation and bridging capabilities. The project is currently refining and detailing these concepts in conjunction with the production of detailed architecture specification.

REFERENCES

[1] 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.

[2] J. Noel Chiappa. Endpoints and Endpoint Names: A Proposed Enhancement to the Internet Architecture. Work in Progress (*Unpublished Internet Draft* http://users.exis.net/~jnc/tech/endpoints.txt), 1999.

[3] 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.

[4] 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.

[5] Robert Moskowitz and Pekka Nikander. Host Identity Protocol Architecture. *Work in Progress* (Internet-Draft draft-ietf-hip-arch-02), January 2005.

[6] Norbert Niebert, Andreas Schieder, Henrik Abramowicz, Göran Malmgren, Joachim Sachs, Uwe Horn, Christian Prehofer and Holger Karl. Ambient Networks – An Architecture for Communication Networks Beyond 3G. *IEEE Wireless Communications*, Vol. 11, No. 2, April 2004, pp. 14-21.

[7] Norbert Niebert, Hannu Flinck, Robert Hancock, Holger Karl and Christian Prehofer. Ambient Networks – Research for Communication Networks Beyond 3G. Proc. *IST Mobile Summit*, June 2004.

[8] 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.

[9] Jerome Saltzer. On the Naming and Binding of Network Destinations. In *P. Ravasio et al. (ed.), Local Computer Networks*, North-Holland Publishing Company, Amsterdam, 1982, pp. 311-317. (Reprinted as *RFC 1498*, August 1993.)

[10] 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.

[11] 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.

[12] 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.

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

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

[15] Zoltan Turanyi, Andas Valko and Andrew T. Campbell. 4+4: An Architecture for Evolving the Internet Address Space Back Towards Transparency. *ACM SIGCOMM Computer Communication Review*, Vol. 33, October 2003, pp 43-54.

[16] Bengt Ahlgren, Lars Eggert, Börje Ohlman and Andreas Schieder. Ambient Networks: Bridging Heterogeneous Network Domains. To appear: Proc. *16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, Berlin, Germany, September 11-14, 2005.

[17] Paul Mockapetris. Domain names - concepts and facilities. *STD 13, RFC 1034*, November 1987.

[18] ITU-T Recommendation E164. The international public telecommunication numbering plan. February 2005

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