Towards Autonomous Network Domains

Stefan Schmid Lars Eggert Marcus Brunner Jürgen Quittek

NEC Europe Ltd.
Network Laboratories
Heidelberg, Germany

{schmid | eggert | brunner | quittek}@netlab.nec.de

ABSTRACT

The Internet is currently evolving beyond what its architecture can support. Often, the mechanisms that allow the Internet to adapt to increasingly conflicting sets of new requirements break some of its basic design principles and can thus severely interfere with end-to-end communication. This paper recognizes that increased autonomy of network regions is a key requirement for future internetworking. It outlines a new internetworking architecture that enables interoperation among a set of autonomous, heterogeneous network domains. The architecture is based on a global identity space and does not require global addressing or a shared internetworking protocol. It integrates the new concept of dynamic network composition with other recent architectural concepts, such as decoupling locators from identifiers.

1. INTRODUCTION

The basic principles of the original Internet architecture include end-to-end addressing, global routeability and a single namespace of IP addresses that are locators and host identifiers at the same time. These principles are suitable for static and well-managed flat network hierarchies. As the Internet evolved from a small research network to a worldwide information exchange, a growing diversity of commercial, social, ethnic, and governmental interests led to increasingly conflicting requirements among the competing stakeholders. These conflicts create tensions that the original Internet architecture struggles to withstand. Clark *et al.* refer to this development as "tussles in cyberspace" [1]. This development has prompted research into different internetworking architectures, such as FARA [2], Plutarch [3], Triad [4] or IPNL [5].

Concurrently with this research into new internetworking architectures, a demand for private, autonomous networks is growing. Although still connected to the global Internet, these autonomous networks offer local features and capabilities that are independent from the public Internet. One important aspect of this autonomy is address space control. Because of the shortage of available IPv4 addresses in many countries, Network Address Translation (NAT) [6] is a popular method for reusing address space. A second perceived advantage of NATs is that they also provide autonomy. They decouple routing in the private network from routing in the public Internet. This enables private networks to attach and detach from the Internet as required, potentially using different access providers, or to multi-home by attaching to multiple service providers at the same time. Finally, NATs hide changes to the internal structure of private networks to the outside.

Although these capabilities of NATs mitigate many immediate problems, NATs are not a clean solution [19][20]. One result of the uncoordinated development of this *ad hoc* solution is significant interference with the operation of traditional internetworking protocols. NATs break the Internet's design principles of end-to-end addressing and global routeability. The private address spaces used internally are neither globally unique nor can the public Internet route them end-to-end. Moreover, the separation between private and public networks that traditional NATs provide is incomplete and therefore restricts the autonomy of private networks. Private networks use private "internal" addresses within their local domain but public "external" ones to address external hosts. Both, private and public network addresses need to be routed in the private network.

A second limitation arises from the use of NATs to overcome address space shortages. When used in this function, NATs map multiple internal private addresses into a few public external addresses (often just a single one). The IP address, however, overloads two separate functionalities onto the same bit string. One is its use as a *locator*, *i.e.*, as an address that denotes a location in the topology of the network. The second one is that of an *identifier* that describes the identity of a node. When NATs translate between internal and external addresses, they also implicitly translate between the associated identities. This causes, for example, applications and protocols that exchange IP addresses in their payloads, such as FTP, to break.

The current approach to deal with this is twofold: First, NAT implementations require constant updates to learn to parse and modify the data stream of new protocols. Besides introducing considerable overheads, this could also lead to instabilities due to frequent modifications to core network functionality. Furthermore, end-to-end encryption or compression can render this technique ineffective. Second, the presence of NATs hampers the development of new protocols, because "NAT interoperability" complicates the realization of many otherwise straightforward ideas and may even prevent adoption of

This work is a byproduct of the Ambient Networks project, partially funded by the European Commission – FP6 IST.

some of them, such as strong end-to-end packet authentication.

Braden [7] proposes the meta-architectural principle that individual regions of the network should be allowed to differ from each other: "minimize the degree of required global architectural consistency." This paper adopts this principle as a necessary enabler for diversity between domains. It describes a new internetworking architecture – called *TurfNet* – that supports end-to-end communication while providing autonomy to private networks [17].

The *TurfNet* architecture focuses on enabling interoperation between otherwise autonomous networks. These autonomous networks are modularized according to the inherent boundaries drawn by the different interests of the stakeholder involved. This paper uses the name *turf* to denote such an autonomous network. The term *turf* has an innate connotation to ownership and responsibility that the *TurfNet* architecture reflects. Other papers introduce different terms for similar concepts, such as *regions* [8] or *contexts* [3]. The concept is also related to the Internet's *autonomous systems* [18].

One key architectural feature of the *TurfNet* architecture is explicit separation of host identities and host locators, similar to HIP [9], multi6 [10], SNF [11], DOA [22] or other proposals. *TurfNet* introduces a new host identity space that enables the use of different addressing and routing mechanisms in each individual autonomous network.

The new host identity space lies between the host name and address spaces. Instead of mapping human-readable host names directly into network addresses, as in the Internet's Domain Name System (DNS), the *TurfNet* name space maps into logical host identities. A second mapping translates host identities into host addresses that are suitable for network-layer data forwarding. The *TurfNet* architecture manages the global name and identity spaces, whereas the address space is local to each individual autonomous network. This difference to the current Internet, which uses a global address space, allows using different addressing and routing mechanisms in individual autonomous networks.

The *TurfNet* architecture allows dynamic creation of forwarding paths across inter-domain gateways, which perform locator translation on packets that traverse between the different autonomous networks. Unlike traditional NATs that only translate addresses assigned to hosts in the private network, the inter-domain gateways in *TurfNet* act as twice-NATs [6] that translate both source and destination addresses. This unique translation operation enables the use of different address schemes in two adjacent but autonomous networks.

A second key feature of the *TurfNet* architecture is network composition. Isolated, autonomous networks can dynamically compose into new, larger autonomous internetworks that integrate the original networks. The process of dynamic network composition supports the interconnection of heterogeneous networks, such as mobile and *ad hoc* networks, IPv4 networks or IPv6 networks. Composed "super" networks manage this integration by abstracting potential isolation (*e.g.*, overlapping address spaces) or heterogeneity (*e.g.*, incompatible network protocols) issues among the constituent subnetworks.

The remainder of this paper is structured as follows: Section 2 outlines the *TurfNet* architecture and explains how it addresses today's changing networking requirements. Section 3 then describes basic communications across several layers of composed *TurfNets*. Section 4 provides a short discussion of the proposed architecture and Section 5 discusses related work. The final section of this paper concludes with an outlook on future work.

2. THE *TURFNET* ARCHITECTURE

A *TurfNet* is a completely autonomous network domain, also simply called *turf*. To achieve this autonomy, every turf encompasses its own independent network addressing mechanisms and all associated control plane functionality, such as routing protocols or address resolution mechanisms. Two common, shared name and identity spaces enable inter-turf communication. They are the only globally agreed state, apart from a common inter-turf control interface.

A second fundamental design choice that supports turf autonomy is the concept of encapsulation. Encapsulation allows turfs to hide their internal structures, characteristics and policies. Such a modular network architecture allows individual players with potentially competing interests to interoperate in a controlled and protected manner and thus suites the requirements of future network communication.

Figure 1 shows an abstract view of the *TurfNet* architecture. Its key components are:

Turf Control. The *turf control* is a logical, per-turf entity that consists of a turf's essential control functions and services. It encompasses all traditional control plane functionality, such as address allocation, routing and address resolution. It further includes the new *TurfNet* functionality to manage, for example, turf composition.

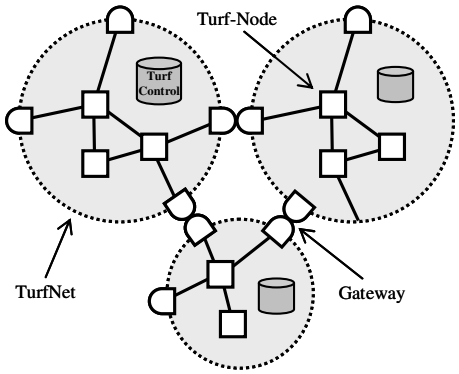

Figure 1. Overview of the TurfNet *architecture.*

Turf Node. A *turf node* is a network node in a specific turf. It interacts with the local turf control for all control plane operations, such as address allocation, routing or

address resolution. For turf-local communication, the turf node must support the local network protocols and addressing schemes. A physical node can participate as a full-fledged turf node in multiple turfs at the same time, allowing multi-homing. Each turf node possesses one or more global names that each map into one or more global host identities. Each of the host identities, in turn, map into locators, which are used for addressing and routing in the local turf.

Gateways. Turf gateways are special, multi-homed turf nodes. Besides participating in multiple turfs at the same time, they can relay traffic between these different turfs. When turfs use different addressing or protocols suites, the gateways also perform the required address and protocol translations when relaying traffic. For example, a gateway between IPv4 and IPv6 turfs translates between the two network protocols and their respective address spaces. If two turfs use the same protocols and have compatible addressing, the gateway can simply forward data packets, acting similar to a traditional Internet router.

The new concept of network composition is central to the *TurfNet* architecture [21]. It provides the basis for individual, autonomous networks to connect to and integrate themselves with other networks in a way that allows them to remain locally autonomous. This means that individual networks in a composition can retain full control over their local addressing and routing mechanisms. The result is called a "composition" or "composed network."

Network composition is inherently different from the more common concept of "network merging." Here, individual networks give up all local control to seamlessly integrate into a uniform, merged network with a single control space. A typical example of network merging is the integration of networks that belong to the same or cooperative administrative domains. However, connecting a customer network to its provider network requires a different type of integration due to limited trust and the desire to preserve some degree of autonomy between the different parties.

Network composition offers this looser form of integration. It preserves the local autonomy of the individual networks, *i.e.*, a turf remains in control of its local facilities even after composition. This enables internetworking between independent, heterogeneous networks that may belong to different administrative domains or have different network architectures. Gateway nodes enable interoperation between the otherwise fully independent networks. The individual turf controls configure their local gateways during the composition process to perform the necessary translation or emulation, if required. The overhead associated with network composition is acceptable where network merging is not an option due to, for example, administrative concerns, lack of trust or desire for autonomy.

The *TurfNet* architecture distinguishes between two different variants of composition, namely *vertical composition* and *horizontal composition.*

When turfs compose *vertically*, one of the composing networks takes on a "provider" role for the other "customer" turfs in the composition. Vertical network composition conceals administrative, control and routing functionalities as well as network-internal structures of the composing turf. Figure 2 illustrates this composition variant. Here, turf B has composed with turf α and turf β, respectively, whereas turf C has composed with turf β only. Customer turfs B and C are encapsulated within the composed turfs α and β, and hence become selectively invisible to external networks. Note, however, that a turf can still compose with other turfs.

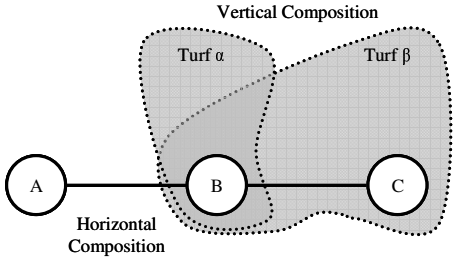

Figure 2. Horizontal and vertical composition.

Vertical composition also reduces the complexity of global control interactions. Due to the hierarchical relation between vertically composed turfs, new turfs can join locally without requiring global interaction, *i.e.*, parent turfs need not be informed when a turf composes locally. Examples of vertical composition are a home network that composes with a service provider network or a body-area network that composes with a mobile operator network.

Horizontal composition is an alternative way for networks to compose. It is the preferred composition variant when networks do not have an intrinsic customer-provider relationship. Horizontal composition is therefore also referred to as "peering" and would apply when, for example, two personal-area networks meet on the move or between service provider networks that establish a direct peering agreement. Figure 2 illustrates the peering relation between turfs A and B, and also between turfs B and C, respectively.

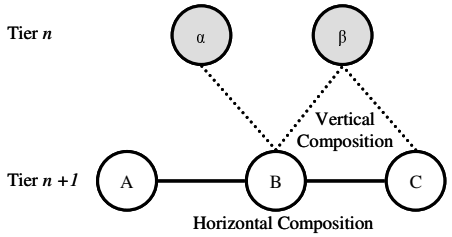

Figure 3. Horizontal and vertical composition (other view).

Figure 3 illustrates the two different variants of network composition in a different fashion that highlights the hierarchical relationship of the composing turfs. It also emphasizes that turfs can simultaneously compose with multiple higher-level turfs. Here, turf B composes with turf α and β at the same time.

3. BASIC OPERATION

End-to-end communication across turf boundaries is not trivial due to the autonomy and potential heterogeneity of the individual networks. The *TurfNet* architecture thus decouples names and locators, similar to FARA [2]. *TurfNet* also uses globally unique node identities as identifiers for turf nodes. These identifiers are different from the node addresses (locators) used for traffic forwarding.

Addresses of turf nodes have typically no end-to-end significance; they are merely transient, local forwarding tags. To establish relaying state, turfs use new node registration and address lookup processes. These new mechanisms configure paths across composed turfs and enable node lookup. End-to-end communication across turf boundaries is thus a product of the following processes: *node registration*, *node lookup* and *packet relaying*. Successful registration and lookup operations pin the data path through the hierarchy.

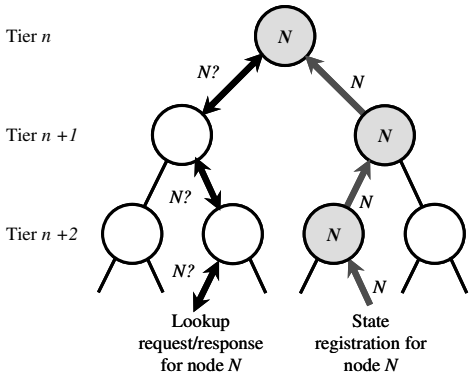

Figure 4. Node registration and lookup request/response (simplified; shown without horizontal compositions).

Node Registration. The lack of a global address space across all turfs may prevent turf nodes that belong to different turfs from communicating without prior registration. The *TurfNet* architecture exacerbates this problem as different turfs may not only use overlapping but also completely different address spaces or network protocols (*e.g.,* IPv4, IPv6, or other internetworking protocols).

A turf node becomes reachable to other nodes by registering its local address with the host-identity-based lookup service of its local turf control. This registration propagates through the hierarchy of composed turfs to achieve turf-external reachability. Turfs always forward non-local registration messages to their vertically composed parents, resulting in a system where subsequent lookups are guaranteed to terminate at the root (see Figure 4). They may also forward them to peer turfs as an optimization, as described below.

Node Lookup. When a turf node initiates communication, it attempts to looks up a local network address for the desired peer via the turf-local host identity resolution service. If the peer node is part of the same turf, this local lookup succeeds and communication remains a local operation supported by the turf-local protocols. However, if the peer node is not part of the same turf, the node resolution request propagates to the vertically composed parent turfs, which then try to resolve the host identity within their respective domains. As an optimization, turfs may also forward a lookup request to horizontally composed peers.

For successful node resolutions, the turfs along the lookup path configure their gateways to allocate proxy addresses install the necessary translation or emulation state between the different address spaces and/or network protocols. (A companion paper describes further details of inter-turf communication [17]).

Packet Relaying. End-to-end communication among turf nodes can begin as soon as the address lookup completes. Data communication follows the path established by the prior registration and lookup operations. The specifics of inter-turf communication depend on the involved turfs. If they use the same address space and communication protocols, the gateway nodes can simply act similar to traditional routers and forward traffic. Otherwise, turf gateways must also perform the necessary address and protocol translations.

The remainder of this section will illustrate the basic operation of the *TurfNet* architecture with a few examples. Figure 4 illustrates a simplified registration/lookup process in a turf hierarchy without horizontal compositions. Here, a node registration for node with identifier *N* propagates up the turf hierarchy. Intermediate turfs register this node in their local turf control. Later, a lookup request for the node with identifier *N* appears (*"N?"*) and propagates up the hierarchy as well. When the request reaches a turf that can resolve the request – here, the topmost turf – it sends a response to the lookup request along the reverse path back to the original requester.

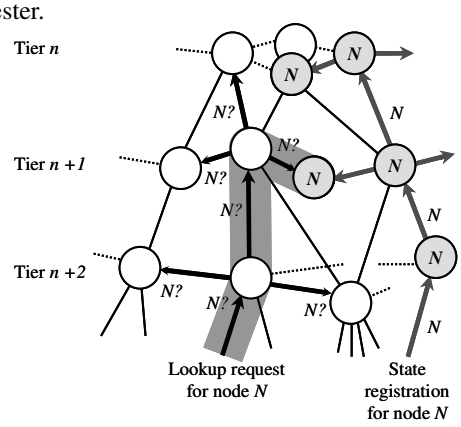

Figure 5. Node registration and lookup request with succeeding lookup across horizontal composition.

Figure 5 illustrates similar registration and lookup operations in a more realistic turf hierarchy that contains both vertical and horizontal compositions. The key difference to Figure 4 is that, as an optimization, turfs not only forward the registration and lookup request "up" the hierarchy along vertical compositions but also "sideways" to their horizontally composed peer turfs.

This optimization has advantages when communication exhibits locality, because it is more likely that lower-level peers can resolve lookup requests, reducing the load on higher-level turfs. Additionally, this optimization can also improve data communication, because in the *TurfNet* architecture, the registration and lookup process pins down the data path.

4. DISCUSSION

Because the *TurfNet* architecture is still in an early development stage, only a brief qualitative evaluation of the overall architectural concepts can be provided here.

Scalability. The main factors that limit the scalability of the *TurfNet* architecture are the storage of address bindings and translation state within top-level *TurfNets*. Ongoing analytical work tries to estimate realistic numbers for, *e.g.*, the size of top-level registration databases or the number of registration/lookup requests. However, it is already clear that a distributed registration and resolution service is required for high-level turfs. Distributed hash tables such as Chord [14] or Koorde [15] could provide this service.

Resilience. Inter-turf communication relies on gateway nodes to relay traffic between adjacent turfs. Failure of these gateways interrupts communication. One way to address fault tolerance in the *TurfNet* architecture is through configuration of redundant gateway paths during the initial address registration. A future paper will investigate this mechanism in more detail.

Performance. Per-packet processing overhead is one important factor that affects overall performance. The relaying method – forwarding, address translation or protocol translation – can significantly affect performance for large numbers of communicating hosts and/or data flows. However, NAT devices for large corporate networks are already able to perform similar operations for fast links. A second factor is performance of registration and lookup operations. Pushing registration information of popular nodes down the hierarchy, similar to techniques proposed for web caches [23], can help speed up these operations.

Mobility. Mobility support is an important criterion for next-generation network architectures. Correspondent nodes typically address a mobile *TurfNode* by its external proxy address. If the mobile node moves only within its local *Turf*, its external proxy address does not change. Mobility management remains local. The hierarchical structure of composed *TurfNets* allows such local handoffs at any level in the hierarchy. Inter-turf mobility is currently under investigation.

5. RELATED WORK

This section discusses related work that also addresses the limitations of today's Internet architecture.

TRIAD [4] is an internetworking architecture that addresses the lack of end-to-end connectivity caused by NATs through an explicit content layer. Similar to the *TurfNet* architecture, TRIAD uses identifiers rather than addresses for node identification and routing. The main difference between TRIAD and *TurfNet* lies in data forwarding. TRIAD uses source routing where *TurfNet* uses a node registration and lookup mechanism to configure paths. Another major difference is that TRIAD requires IPv4 in all network domains, whereas *TurfNet* can operate across heterogeneous domains.

Similar to *TurfNet*, Plutarch's goal is explicit support of heterogeneity [3]. It introduces the concept of *interstitial functions* to translate communication among heterogeneous networks. Plutarch differs from *TurfNet* with respect to naming and routing. Plutarch assumes that namespaces differ in every domain and that forwarding is based on sender selection of a context chain, together with configuration of the required interstitial functions.

IPNL [5] and 4+4 [16] aim at isolating independent IP subnetworks through loose integration. They also use NATs to integrate networks with potentially overlapping address spaces to avoid renumbering. Two fundamental differences to the *TurfNet* architecture exist. First, *TurfNet* does not limit the number of hierarchical composition steps, whereas IPNL supports only two levels: NAT'ed *private realms* and global *middle realms*. Second, the *TurfNet* architecture does not depend on a common addressing scheme or network protocol.

6. CONCLUSION AND FUTURE WORK

This paper has motivated the concept of autonomous network domains that overcome key limitations of the current Internet architecture and presents an architecture that supports this autonomy. Although existing mechanisms, such as NATs, provide some degree of autonomy, they also break several of the Internet's key design principles and consequently interfere with end-to-end communication.

The *TurfNet* architecture enables global, packet-switched internetworking across autonomous network domains. *TurfNet's* support for dynamic network composition allows individual networks to maintain a high degree of autonomy. *TurfNet* can integrate individual networks that use different network protocols and addressing schemes into a shared whole. *TurfNet* uses a common control interface across the individual networks to register and look up hosts, and to establish relaying state in of border gateways. These gateways perform the required protocol and address translations and facilitate communication across turf boundaries.

This paper gave an overview of the *TurfNet* architecture and its basic mechanisms. One important area of future work is investigation of specific approaches for inter-turf routing. Other areas include approaches for managing dynamic composition of potentially mobile networks. Finally, evaluating the scalability and performance characteristics of *TurfNet* through simulations and measurements of a prototype implementation is another future work item.

REFERENCES

[1] D. Clark, J. Wroclawski, K. R. Sollins and R. Braden. Tussle in Cyberspace: Defining Tomorrow's Internet. Proc. *ACM SIGCOMM*, Pittsburgh, PA, USA, August 19-23, 2002, pp. 347-356.

[2] D. Clark, R. Braden, A. Falk and V. Pingali. FARA: Reorganizing the Addressing Architecture. Proc. *ACM SIGCOMM Workshop on Future Directions in Network Architecture (FDNA)*, Karlsruhe, Germany, August 2003, pp. 313-321.

[3] J. Crowcroft, S. Hand, R. Mortier, T. Roscoe and A. 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.

[4] D. R. Cheriton and M. Gritter. TRIAD: A Scalable Deployable NAT-based Internet Architecture. Stanford Computer Science Technical Report, January 2000.

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

[6] P. Srisuresh and M. Holdrege. IP Network Address Translator (NAT) Terminology and Considerations. RFC 2663, August 1999.

[7] R. Braden, D. Clark, S. Shenker and J. Wroclawski. Developing a Next-Generation Internet Architecture, Whitepaper, available at http://www.isi.edu/newarch/DOCUMENTS/WhitePaper.ps, July 2000.

[8] K. R. Sollins. Designing for Scale and Differentiation. Proc. *ACM SIGCOMM Workshop on Future Directions in Network Architecture (FDNA)*, Karlsruhe, Germany, August 2003, pp. 267-276.

[9] R. Moskowitz and P. Nikander. Host Identity Protocol Architecture. Work in Progress (draft-ietf-hip-arch-00.txt), October 2004.

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

[11] 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.

[12] R. Droms. Dynamic Host Configuration Protocol. RFC 2131, March 1997.

[13] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460, December 1998.

[14] I. Stoica, R. Morris, D. Karger, M. Frans Kaashoek and H. Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. *IEEE/ACM Transactions on Networking*, Vol. 11, No. 1, February 2003, pp. 17-32.

[15] M. F. Kaashoek and D. R. Karger. Koorde: A simple degree-optimal distributed hash table. Proc. *2nd International Workshop on Peer-to-Peer Systems*, Berkeley, CA, USA, February 2003, pp. 98-107.

[16] Z. Turanyi, A. Valko and A. 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.

[17] S. Schmid, L. Eggert, M. Brunner and J. Quittek. TurfNet: An Architecture for Dynamically Composable Networks. Proc. *First IFIP TC6 WG6.6 International Workshop on Autonomic Communication (WAC 2004),* Berlin, Germany, October 18-19, 2004.

[18] J. Hawkinson and T. Bates. Guidelines for creation, selection, and registration of an Autonomous System (AS). RFC 1930, March 1996.

[19] J. Touch. Those Pesky NATs. *IEEE Internet Computing*, July/August 2002, p. 96.

[20] M. Holdrege and P. Srisuresh. Protocol Complications with the IP Network Address Translator. RFC3027, January 2001.

[21] C. Kappler, P. Mendes, C. Prehofer, P. Pöyhönen and D. Zhou. A Framework for Self-Organized Network Composition. Proc. *First IFIP TC6 WG6.6 International Workshop on Autonomic Communication (WAC 2004),* Berlin, Germany, October 18-19, 2004.

[22] M. Walfish, J. Stribling, M. Krohn, H. Balakrishnan, R. Morris and S. Shenker. Middleboxes No Longer Considered Harmful. Proc. *USENIX Symposium on Operating Systems Design & Implementation (OSDI)*, San Francisco, CA, USA, December 2004, pp. 215-230.

[23] J. Touch and A. S. Hughes. The LSAM Proxy Cache - a Multicast Distributed Virtual Cache. *Computer Networks and ISDN Systems*, Vol. 30, No. 22-23, November 25, 1998, pp. 2245-2252.