Emerging Inter-Networking Technologies: From IPv6 to Host Identity Protocol and Beyond

Dr. Pekka Nikander

Ericsson Research Nomadic Lab
Hirsalantie 11
FI-02420 JORVAS, Finland

pekka.nikander@nomadiclab.com

Helsinki Institute for Information Technology
Metsänneidonkuja 4, P.O.BOX 9800
FI-02015 TKK, Finland

pekka.nikander@hiit.fi

ABSTRACT

This presentation leaves the projected longer-term trends in the background and focuses on more near-term developments. Looking from a practical, hands-on perspective, the potential role and significance of the following technologies are considered:

- *New technologies under IP: 3G, WiMax, LTE, etc.*
- *Mobility and Multi-homing: From Mobile IP, NEMO, and SHIM6 to a "CGA-world"*
- *Solving routing problems: Jacking up the protocol stack*
- *Full-fledged identifier / locator split: Host Identity Protocol (HIP)*
- Internet telephony: SIP, IMS, and beyond

1.0 INTRODUCTION

At the practical level, the current data networking environment has a large number of problems. Even larger is the number of attempts to provide short term remedy, to overcome some of these problems through new technologies or different clever tricks, varying greatly in their architectural insight.

In this presentation, we very briefly delve in the solutions space, starting from emerging wireless technologies. After that, we have a compact but still somewhat lengthy tour to the quite complex world of mobility and multi-homing support, continuing the tour with the so called "jack up" attempts to fix a number of scalability problems. We will spend a brief moment at the so called identifier / locator split approaches, since they seem to provide a slightly more architected solution to a number of problems than many of the other approaches, deferring any longer considerations to the third presentation [1] in this series. Finally, more as a requested scenery visit to the world of voice than a piece of solid application-independent solutions, we take a separate look at certain aspects of Internet telephony.

This presentation contains no conclusions, since we find it immensely difficult to conclude anything from this kind of a hodgepodge.

Nikander, P. (2007) Emerging Inter-Networking Technologies: From IPv6 to Host Identity Protocol and Beyond. In *Emerging Wireless Technologies* (pp. 2-1 – 2-20). Educational Notes RTO-EN-IST-070, Paper 2. Neuilly-sur-Seine, France: RTO. Available from: http://www.rto.nato.int.

2.0 FITTING IP TO NEW WIRELESS MEDIA

While media-independent in theory, in practise the Internet protocol family was designed primarily with fixed networks in mind. This became clear years ago when people started to notice various performance anomalies when trying to run TCP over radio networks that were not, in practise, designed with IP traffic in mind.

There are two main problem areas. First, the reliability, bandwidth, and delay characteristics of radio networks vary wildly. This causes problems to the TCP-compatible end-to-end error and congestion control techniques that the IP stack relies heavily on. Second, the IP architecture has been designed primarily with always-on connectivity in mind. In environments where always on is not possible or is impractical, for example, due to excessive pricing, the modern IP applications tend to behave erratically or not at all. For example, while it is technically possible to synchronise one's e-mail over a 9600 bps, 900 ms round trip time (RTT) GPRS connection, given the current volume of spam and other e-mail such a function may take several hours and cost up to hundreds of euros, depending on the rate plan.

While the new wireless technologies, some of which we consider in Section 2.1 below, will finally make it possible to practically run most typical IP applications over wide-range wireless networks, there remains a number of standardisation and interaction problems, considered in Section 2.2, and more fundamental problems, discussed in Section 2.3.

2.1 Examples of new wireless technologies

Today, Wireless Local Area Networks (WLAN) of the IEEE 802.11 variety remain the only readily-available wireless technology that provides good support for demanding real-time IP applications, such as multi-user games, streaming video, and IP telephony. With its 10–100 Mbps shared bandwidth and typically 1–2 millisecond round trip time (RTT), IP applications work without any problems at all. However, typically the coverage is quite limited and the available frequency bands are shared with other applications.

In the 3GPP world, High-Speed Packet Access (HSPA) in its current High-Speed Downlink Packet Access (HSPDA) incarnation also provides a reasonably well working solution, with typically around 100 ms RTT, about 1 Mbps downlink and 150 kbps uplink capacity in practise. However, especially the latency is still insufficient for some real-time applications. The next generation of HSPA, with the added enhanced uplink support, is expected to reach peak rates at 40 Mbps downlink and 10 Mbps uplink, with typically less than 50 ms RTT and perhaps 10% of the peak rates for typically available data bandwidth; those numbers should be reasonable for typical use. Hence, HSPA R8 is expected to be close in performance to LTE at 5 MHz. [2]

3GPP Future: Long Term Evolution (LTE)

The next generation 3GPP networks, based on Long Term Evolution (LTE) radio links and System Architecture Evolution (SAE) on the network side, is expected to reach latencies and data rates that are comparable to today's WLAN but available through the cellular system, thereby supporting real vehicular speed mobility and reasonably wide coverage (5 km cell size for high speed). At the same time, frequency efficiency is expected to be improved by 2–4 times compared to the current first-generation HSPA-systems.

The target latencies are less than 100 ms from idle to active and 5 ms in active state; in practise the system can be expected to reach 10–20 ms RTTs with realistic loads, given the more efficient SAE-based backbone network. The target peak rates are 100 Mbps downlink and 50 Mbps uplink. If the experience from the current HSDPA can be applied to LTE/SAE, realistic data rates with a stationary or slowly

moving terminal might be 10 Mbps downlink and 5 Mbps uplink, and somewhat slower for vehicular speeds. [3]

WiMAX and Mobile WiMAX

Today, WiMAX services are offered as a fixed, radio-based replacement for the "last mile" copper. In practical settings, fixed WiMAX can reach symmetrical speeds of maybe 10 Mbps at 10 km line or sight, or at 2 km when obscured by buildings. Typical latency is reported to be in the 200–500 ms range. The service is commercially available in several locations throughout the world, though frequency allocations and general availability vary a lot.

Worldwide, but mainly in North America, several companies have announced their plans to launch Mobile WiMAX services, based on the IEEE 802.16e-2005 standard. It should be noted that from the technical point of view, Mobile WiMAX is based on a different radio technology (SOFDMA vs. OFDM) than the currently deployed fixed WiMAX. Therefore, the base stations and the terminals will be quite different.

Since Mobile WiMAX has not any public deployment yet and the WiMAX forum has not publicly stated latency goals in the same way 3GPP has, it is hard to acquire reliable data about the expected latencies in the forthcoming WiMAX networks. However, based on the scattered pieces of information from here and there it seems plausible to expect about 30–80 ms RTT over the air interface (e.g. [4] indicates 30 ms; [5] indicates 80–160 ms design goal), yielding perhaps 80–100 ms RTT in a typical 3GPP2 mobile network, utilising Mobile IP, while working within the home continent. RTTs involving inter-continental roaming will depend heavily on the mobility solutions and may easily exceed 300 ms if several trans-oceanic links are needed, e.g., due to Mobile IP Home Agent dependencies; see Section 3.0, below.

While the advertised Mobile WiMAX peak rates (e.g. [4]) somewhat exceed the specified HSPA peak rates, most data seems to point that such speeds will be only reached under exceptional conditions near to the base stations. For first deployments, realistic rates might be in the 250 kbps – 2 Mbps range, depending on the distance from the base station.

Overall, it must be understood that all radio systems follow the same laws of physics and are basically similarly limited by computational power. Hence, under similar conditions and similar antenna systems they will necessarily reach roughly equalling performance. The biggest differences are basically related to parameters: how the available time-spectrum-capacity is divided between the uplink and downlink in one hand, and between the users in the other hand. These choices in turn create certain bandwidth, latency, an jitter characteristics, thereby creating the conditions within which the Internet protocols operate.

2.2 Standardisation and interaction problems

The Internet grew from a wireline network; indeed, many of the early links were modem connections running over telephone wires. Today, most of the core internet runs over fibers and a large fraction of last mile connections are either based on DSL, running on old telephone copper pairs, or cable-based, running on copper, too. Radio, on the other hand, is physically very different from copper or fiber.

Almost Internet technologies (with the natural exception of VoIP) rely on fairly large packet sizes (typically about 1500 bytes, mandated by the Ethernet standards), and can tolerate larger latencies and jitter than voice. Standardisation on cellular networks started in 1980s, solely focusing on voice. The aim was to use the bandwidth as efficiently as possibly, allowing hundreds of milliseconds for connection set-up and utilising small packets and deterministic scheduling to meet the voice latency and jitter requirements.

Another big difference lies in the assumptions about the socio-economic environment. Standardisation of the cellular networks has been largely led by telecom vendors and operators, focusing on balancing their

interests, more often than not on the end-user's cost. On the other hand, the Internet was originally built by its users for themselves. For a long time, the IETF was dominated by people who were as much independent Internet users as developers and designers; this still shows in the official attitude of the IETF members being individuals, not company representatives.

These two approaches — fixed+data vs. radio+voice and programmers-for-themselves vs. system designers-for-operators — have led to two very different cultures. Even though the large majority of IETF delegates are paid by their employers to attend the IETF, there are still a lot of people there who are primarily interested in social good, i.e., maximising the social utility though designing an open network that maximises the users' freedom. These people feel that they are primarily there to design the network for themselves and other private citizens. On the other hand, a larger fraction of the people there feel more like being company representatives, primarily interested in making sure that their company's interests go forward. Some of the people are able to successfully integrate these two roles, for another group of people IETF's official rules cause problems in how to follow them and advance company interests at the same time.

These differences in backgrounds and attitudes show in several ways in the current Internet standardisation. People coming from the wireless and cellular background claim that the IETFers don't understand radio. Many old time IETFers, on the other hand, blame radio link designers not understanding data. While neither of these claims are no longer strictly true, there still are clear problems in the way the Internet protocols run over cellular systems. Especially many of the earlier cellular data systems do not work well with TCP, for example. Full, seamless interoperation and optimal performance will most probably be achieved only in a few years, as the radio latencies become shorter and the cellular networks evolve towards optimal routing.

A major problem still today is the very loose interaction between the cellular link layers and the Internet protocols. In one hand, the Internet protocols have not been design with cellular radio characteristics in mind, thereby creating difficulties in trying to optimise the usage of radio capacity for different types of packet traffic. On the other hand, especially the past but also some of the current cellular technologies have been heavily optimised for voice, thereby creating unfavourable operating conditions for typical Internet packet data traffic.

2.3 Intermittent connectivity: will IP be the right model in the long run?

Taking a longer, more user-oriented perspective, radio resources are and will be an expensive resource. Sending and receiving data over radio consumes quite a lot of energy — a phenomenon known to anyone who has ever tried to use WLAN from a handheld device; the batteries just don't last very long. Even if the radio system, inter-connectivity, and applications get better integrated in the future through cross-layer interactions (see e.g. [6]), the resource consumption problem pertains.

The only fundamental remedy appears to require reconsideration of what the network is. With the notable exception of interactive media, such as telephony, most *applications* are relatively delay-insensitive, or the delay-sensitive portion is a small fraction of the total data. For example, for most e-mail messages it is not that important how many minutes, or even hours, their delivery takes. However, the current network reality does not reflect that insensitivity. Instead, TCP and most other transport protocols are relatively sensitive to latency, or, more precisely, to packet loss combined with a high bandwidth delay product (BDP) [7]. Hence, for example, in order to display a web page that has been created, say, a year ago, the network must provide relatively large bandwidth to get the data through when needed, and relatively low delay in order get TCP to transmit the data fast enough.

The only currently known, fundamental remedy to this phenomenon is to change the network's responsibilities, basically moving to data oriented network; cf. [8]. If the applications and the network had

a larger, semantically richer interface, the applications could inform the network better about their data needs, including both what information they are interested in and how delay-sensitive that information is. With such information, the network could provide the applications better with their needs, allowing many applications to reduce their near-real-time communication needs by orders of magnitude; sometimes even allowing the applications to work without any network connectivity.

3.0 MOBILITY AND MULTI-HOMING

The IP and original transport protocols, TCP and UDP, were designed with immobile or at most transportable hosts in mind; anyway, back at that time it would have been hard to imagine a computer that could be moved while being operated, if for no other reason then due to the power requirements. Consequently, the IP and transport layer naming systems are tightly bound together, creating a structure that makes node mobility unnecessarily hard; see Section 5.0 for a continuation of the discussion.

From the practical point of view, it is possible to discern mobility (and multi-connectivity) needs at several different granularities. The different styles of mobility are briefly described in Section 3.1, and a more comprehensive picture is formed in Section 3.2, as the granularities are represented as gaps in the stack. In Section 3.3 we very briefly characterise some specific approaches.

3.1 Mobility at different granularities

Host or node mobility is by far the best-studied type of mobility, with the possible exception of link mobility. However, since link mobility is always bound to a specific access technology, there are big differences between the level of mobility support between different link types. Beyond what we already said above about specific technologies, access aspects are left beyond the scope of this presentation.

A proper coverage of node mobility would require several pages. However, in the name of brevity, we only briefly outline some of the most important challenges and aspects here, leaving the details out. In node mobility, the moving entity is an IP-addressable node, typically a host. As the node changes its location, it also changes its IP address. A node mobility solution must make this change known to relevant other parties. Depending on the solution at hand, the relevant other parties may be active peer nodes, a fixed central node (such as Mobile IP Home Agent), a distributed infrastructure, or a combination of those.

All IP-based node nobility solutions IP addresses as location names. However, how the mobile node are named themselves depend heavily on the solution in hand. In some solutions, such as the Host Identity Protocol [9], the mobile nodes are named with identifiers from a new name space. In other solutions, such as Mobile IP and most tunnelling solutions, the mobile nodes are named with IP addresses that are made stable by providing a proxy service (the Home Agent). The most simple minded solutions do not fully support node mobility but partially solve the problem by using other identifiers (such as domain names) as mobile node identifiers, resolving them into IP addresses at the start of communication.

Node mobility is usually divided into two sub-problems:

1. The initial connectivity problem deals with the situation where a previously inactive node wants to open a session with the mobile node. To do that, the activating node must be able to send the first IP packet to the mobile node's current IP address, without having any a priori knowledge of the IP address. Almost all available solutions use some mediated mechanism (such as Mobile IP Home Agent) to solve this problem.
2. The session continuity problem deals with the situation where currently active nodes want to continue their current communications with a mobile node, even when the mobile node changes its IP address. There are both mediated and direct end-to-end approaches to solve this problem.

Other dimensions relevant to the node mobility problem include the following:

- Ability to deal with inter-addressing-family movements, or the mobile node using IPv4 and IPv6 alternately.
- Ability to support multiple simultaneous network accesses or other types of multi-homing.
- Integration or ability to integrate with security services, such as IPsec.

In today's practical networks, the NAT boxes add an additional problem by allowing connections to take place only in one direction.

Subnet mobility

The original Internet architecture supported sub-network mobility through routing. Each of the class A, B, and C networks had a separate routing table entry in the global routing table, making it possible (at least in theory) to move them around in the network. With Classless Inter-domain Routing (CIDR), this ability disappeared since instead of carrying separate global routing table entries for each network, entries were aggregated in order to save space in the routing tables. This, in turn, made it much harder to move specific network around. Additionally, due to the currently fairly slow convergence times in the global Internet, while routing-based sub-network mobility is still possible in a limited manner, it takes some time before the network reacts to movements.

In the traditional, routing-based sub-network mobility solutions, the mobile sub-networks are named with the IP prefixes. The locations are indirectly named via the connectivity information in the routing table. This can be compared with the NEMO-based sub-network mobility solutions, which are based on the Mobile IP architecture, using network prefixes both as names for the mobile sub-networks and their locations. The current NEMO solution uses tunnelling to pass traffic from the home location to the visited location.

HIP-based sub-network mobility solutions name sub-networks only indirectly by enumerating the HIP nodes in the sub-network, and using IP address prefixes (of the visited location) as the location name.

Mobility of application-level entities

There are three application-level entities that can be reasonably to be expected to be made mobile: users, services, and sessions. Here we focus only on session and service mobility. While being close relatives, from the architectural point of view session and service mobility represent slightly different phenomena and requirements. Since service mobility often involves session mobility, at least when services are being moved while active, we consider session mobility first.

In session mobility, the mobile entity is an end-point of an active session. It is important to understand that session mobility is distinct from node mobility, which may involve several sessions (all running at the same node) being moved at the same time. It is also distinct from user mobility, since a user may move an active session from one user to another.

For proper session mobility, it must be possible to name the end-points of the session[1]. In the IP architecture, this poses a challenge since active sessions (TCP connections or bound UDP sockets) are not independently named but bound to the underlying IP addresses, the IP addresses acting as node identifiers. SIP-based mobility solves this problem by providing an independent name space for sessions, and using a separate signaling channel for binding the session end-points to < IP address, port > pairs. However, it provides session mobility only for SIP-based services.

[1] Of course, the requirement of being able name the mobile entity is nothing new. However, in the case of session mobility where the mobile entity is the session end-point, this may not be completely obvious.

Service mobility, in turn, may involve moving both services and sessions. Moving inactive services from a host to another is relatively simple, involving data replication, service instantiating, and update of the DNS or other service directory information. However, due to the inherent slowness of DNS updates (caused by caching) and the fact that many older applications require a restart in order to perform the DNS resolution again, such service mobility is currently fairly slow. In general, moving global services from a host to another may require weeks or at least days of preparation time.

Moving active services, including their active sessions, poses a much larger challenge. In general, the current IP architecture does not support this type of mobility at all. While there are experimental research approaches that provide even this kind of mobility, they are usually limited to specific protocol environments, involve special application-level libraries, and/or require special versions of the underlying operating systems.

Information mobility

Information mobility refers to the system ability of providing information readily at need at different locations and at different times, independently of the underlying connectivity and mobility phenomena. As such, it gets close to information or data oriented networking, one of the main research approaches outlined in the previous presentation [10]. Typically, information mobility can be achieved through replication, caching, and forwarding, leading to cache consistency problems. The area of maintaining synchronised / coherent directory information is addressed to a limited extent in some distributed directory systems. However, the conclusions from that type of work is that there are limitations to what can be done using conventional approaches in due to the requirement of supporting scenarios with updates to data while the network is partitioned. The general aspects of information or content mobility are an ongoing area of research.

3.2 Mobility gaps

Given that the IP architecture was designed with a stable network in mind and that routing protocols cannot scale to the desired levels of host and sub-network mobility, the current IP architecture can be depicted as having a number of *mobility gaps.* That is, we can identify a number of structural “places” in the current IP architecture that can be considered to lack functionality that would be needed to implement certain types of mobility. These gaps are illustrated in Figure 1 — Mobility gaps in the IP architecture.

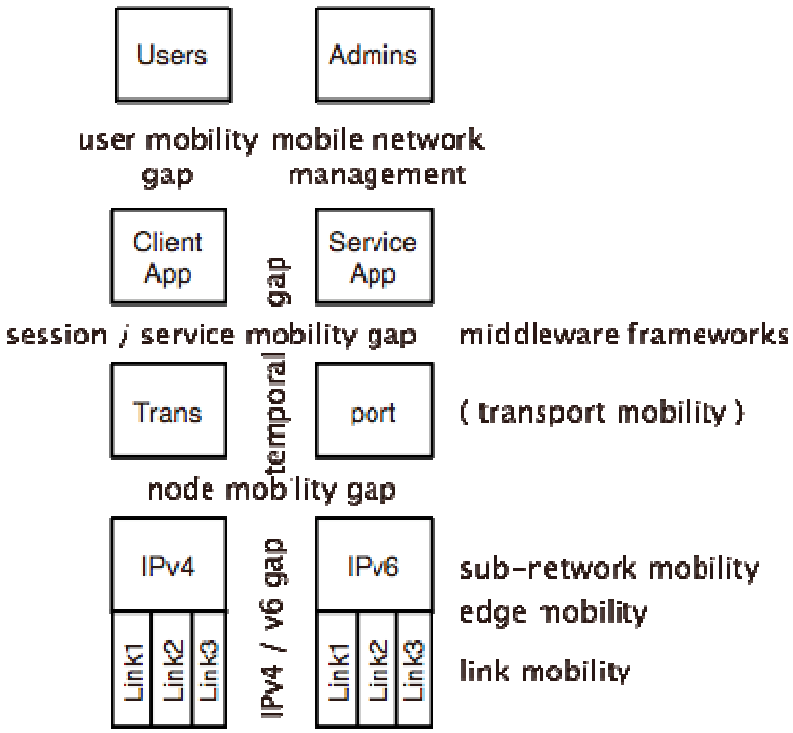

Figure 1 - Mobility gaps in the IP architecture

Starting from the bottom, there is a gap between the IPv4 and IPv6 versions of the IP protocols. While this gap is not directly mobility-related per se, it affects mobility solutions since any well-working global mobility solution must be able to deal with the fact that some parts of the network will still be IPv4-only while others might be IPv6-only, with gateways connecting these two disjoint parts of the network. Hence, the IP-version gap has its consequences to all the mobility solutions above, but especially to node and session / service mobility. Today, the IP version gap is explicitly addressed by many but not all node mobility protocols.

Temporal gap

Each mobility event takes some time, having an effect in transport protocols and applications. Any communication break that is considerably larger than the average round-trip time must be considered as a potential issue for the transport protocols and applications. At minimum, the transport protocol may need to re-evaluate its delay and bandwidth estimates, possibly causing a larger disruption in communication than would strictly be needed. Any user-visible break longer than approximately two seconds may cause the user to wonder about the reason for the break and start to act.

The node mobility gap and the lower layer mobility types

The lowermost IP-technology-related mobility gap in the figure is the node mobility gap. This gap reflects the current dual role of IP addresses. As discussed in detail later in Section 5.0, from the routing point of view, IP addresses are considered as location names or locators, being assigned by the network based on the network topology. At the same time, from the upper layer protocols point of view, IP addresses work as node names, identifying the node a transport or other upper layer protocol communicates with. This dual-role nature of IP addresses works nicely as long as nodes do not change their locations but fails badly with mobile nodes. The node mobility gap is usually addressed by node mobility protocols, such as

Mobile IP, Host Identity Protocol (HIP), and the IPsec mobility protocol (MOBIKE). However, it can be addressed at some upper layer, too.

Below the node mobility gap, we can identify three other types of mobility, namely *sub-network mobility, edge mobility,* and *link mobility*. Of these, edge and link mobility do not really represent any gaps in the architecture — the architecture can be easily extended to support them. Indeed, there are tens of different link mobility solutions, and the IETF NetLMM working group appears to be on track to define an IP-based edge mobility solution.

Sub-network mobility is a slightly harder case to define and illustrate. In theory, the IP routing architecture does support sub-network mobility, and it trivially did it before the introduction of CIDR. Furthermore, even today it is possible to stretch the limits of the routing system to support sub-network mobility via routing, as amply demonstrated by Connexion by Boeing, a technically successful but commercially unsuccessful sub-network mobility service, offered 2004–2006. At the same time, there are scalability, security, and management arguments that indicate a desire for separate, routing-independent sub-network mobility solutions. While the amount of existing work in that area is much smaller than with node mobility, there seems to be two somewhat established proposals, one by the NEMO working group at the IETF based on Mobile IP ideas, and another one based on the Host Identity Protocol.

The session and service mobility gap

Climbing up the stack, the next visible gap can be identified between the transport and application protocols (or within the transport protocols, depending on the level of service provided by the transports). In the figure, this gap has been given the name of session and service mobility gap. This gap reflects two kinds of situations. The first one, corresponding to session mobility, is when a session (such as an active TCP connection) needs to move from one node or interface to another, independent of the other sessions at the same node or interface[2]. The second kind involves services, and reflects situations where a service (without any active sessions) needs to be moved to another node. Of course, the combined situation, where a service and its active sessions are moved at the same time, needs to be considered, too.

In this area, the range of available solutions is wide and varying. At the technically simplest end, Dynamic DNS allows modern applications to re-resolve a domain name to the IP address whenever the application detects problems with the current communication, thereby allowing the application to recover from service mobility. However, such a solution does not provide service mobility and requires explicit mechanisms from the upper layer protocols in order to recover from the potential state-loss caused by ripped down TCP connections[3].

The other end of the solution scale is represented by some complex middleware solutions, where the middleware maintains the mapping between service names and IP addresses and/or ports, making any changes in them invisible to the applications. The existing middleware platforms differ much on their support for mobility — some have no support while others provide extensive abilities to deal with different mobility situations. The basic drawback of middleware solutions is that they only deal with applications that use them.

From a longer-term point of view, there is also the possibility of changing the transport and session layers in the IP stack structure. However, at the time of writing there does not appear any credible proposals for implementing such changes; active researchers in the area do not appear to agree on what would be the characteristics of viable approaches.

[2] Note that moving a session end-point from one application to another application within the same host may be implemented in a way that is transparent to the communicating peers, basically as a local matter.

[3] Whenever a TCP connection is violently torn down (e.g. due to a mobility event), it is impossible to know how much of the sent TCP data actually reached the recipient. Consequently, the sender and recipient need explicit upper layer protocol mechanism in order to figure this out and be able to re-synchronise their communication state.

User and management gaps

The user and management mobility gaps, highest up in the figure, depict situations where a user changes the terminal equipment they use or where an administrator attempts to manage a network that employs highly mobile nodes and services. Both represent big challenges to the current systems. However, neither of these problems is pertinent to IP-based networks, and thereby outside the scope of this presentation. Consequently, we merely mention that user mobility is often tied to various single-sign-on solutions.

Gap summary

Table Table 1, next, attempts to summarise the different types of mobility from a networking point of view. The various entities include users, services, sessions, nodes, and sub-networks.

Table 1 - Entities, Entity Names, Locators, etc, when solving the different gaps

Moving entity	Entity name	Locator	Location(s) of mapping info	Protocols
User	User identity	Node identity	Directory	various
Service	Service identity	Node identity	Directory	various
Session	Session identity	Node identity	Peers	various, e.g. SIP
Node	Node identity[1]	IP address	Peers & Rendezvous points	HIP, MIP, etc.
Node	IP address	Edge identity[2]	Anchor points, Edge routers	NETLMM, PMIP
Node	MAC address	Link or access point identifier	Routing or switching tables	GPRS, 802.11r, 802.16e, …
Subnetwork	various	various	various	various, e.g. NEMO, BGP

[1] Different protocols use different items as their node identifiers.
[2] At the time of writing, there are no established methods for naming edges.

3.3 Specific technologies

In this section we have a brief look at some of the mobility-related IETF standards track technologies: Mobile IP, NEMO, and SHIM6. We also briefly envision how the SHIM6 approach could be generalised into what some people call the “CGA World”. The discussion later continues in Section 5.0, when considering a full-blown identifier / locator split approaches.

Mobile IP

Mobile IP is the IETF standard solution for bridging the node mobility gap. Currently there are separate solutions for IPv4 and IPv6 mobility [11][12], but there is ongoing work towards integrating the two approaches under a single framework. A number of other IETF solutions are based on it, often inheriting both the good and bad properties of Mobile IP.

The basic Mobile IP design stems from middle 1990's [13], and has remained essentially the same since then. The original design was strongly motivated by the desire to keep the corresponding hosts unchanged; i.e., any node communicating with a mobile node does not need to be aware of the node's mobility aspects. As a consequence of this fundamental design choice, Mobile IP uses IP addresses both as entity names and locators. When an IP address is used as an entity name, it is called Home Address. When an address (perhaps the same one) is used as a locator, it is called Care-of Address.

Using IP addresses as entity names binds the names to topological locations; see Section 4.1. Architecturally, this can be considered as the main deficit of Mobile IP: the architecture binds the mobility anchor point (Home Agent) tightly to a specific location through the use of IP addresses as entity names. Attempts to alleviate this deficit necessarily lead to solutions where the mobility solution is partially delegated to the the routing system, e.g., through anycast.

NEMO

NEMO [14] is the IETF standard solution for bridging the subnet mobility gap. It is architecturally similar to Mobile IP, relying heavily on static home network prefix and a home router. The current versions do not support route optimisation, necessarily leading to tunnelling and triangular routing. The ongoing work to alleviate this problem is plagued with security problems; see [15]

SHIM6 as a bases for a mobility solution

SHIM6 is the IETF standards track solution [16] for providing node and, eventually, subnetwork multi-homing in IPv6 world. The main architectural feature is its ability to bind multiple IP addresses cryptographically together, providing the peer reasonably strong bases to believe that those addresses are possessed by a single host. The binding can be done in two different ways. In the first method, the multi-homed host selects a pre-defined set of IPv6 network prefixes and constructs an address for each of the subnets identified by these prefixes in such a way that the address is cryptographically bound to the prefix set. In the second method, the addresses are bound to a public key from a cryptographic public-private key pair. The latter method allows the prefix set to be dynamic, thereby making mobility support possible.

A "CGA World" approach

The SHIM6 approach can be generalised. The method of binding an IPv6 address to a public cryptographic key [17] is customarily called CGA, for Cryptographically Generated Addresses. The basic idea is to use a secure one-way function over the public key and other information as the low-order 64 interface identifier bits in the address. Due to the use of a one-way function, given any CGA address it is a computationally very expensive (though not impossible) problem to generate another public key that results into the same CGA address. In that way, given a CGA address and a public key (and the other related info used in the one-way function input), one can be reasonably sure that a single entity (node) has generated the public key and, consequently, the CGA address, and therefore wants to use the public key as a "higher layer" identifier for itself. In addition to SHIM6, CGA is also used in the IPv6 Secure Neighbor Discovery (SEND) protocol [18].

From a more abstract point of view, the CGA approach can be seen as a half-step towards identifying network entities with public keys instead of IP addresses. It allows a set of IP addresses to be identified

with a public key. This approach, in turn, can be considered as a "half-hearted" way of doing something similar to HIP (see Section 5.0), at a somewhat lower security level.

4.0 JACKING UP THE PROTOCOL STACK

From an architectural point of view, the inter-networking layer provides the important service of providing universal connectivity. With a given set of node identifiers (usually called IP addresses), the layer provides a facility to deliver datagrams to the identified recipient(s). The crucial question is the structure of the identifiers, and relatedly, their ability or inability to embed topological information; see Section 4.1. The conventions on how IP addresses have been used to embed such topological information has varied considerably over time. However, given the scalability and flexibility goals, the current practises do not seem sufficient, leading to a pressure of fundamentally changing the structure of the inter-networking layer; section 4.2 contains a very brief roundup of that.

It is also important to understand that IPv6, while it may solve many problems, does not solve the critical routing problems in the Internet today. In fact, in some sense, IPv6 exacerbates a number of them by adding a requirement for support of two Internet protocols and their respective addressing methods.

4.1 Network location vs. node identity

Currently, the IP addresses serve the dual semantics of acting both as (partial) names for network locations and as (full) names for nodes in the network. As long as nodes are immobile and logically only at one place in the network (i.e. not multi-homed), such a practise works fine. However, as soon as a nodes become mobile or a large fraction of nodes is reachable through a number of logically different connections, problems emerge.

There is currently a clear tendency to work towards a solution where the location naming and node naming properties of IP addresses are separated from each other. Customarily, the solutions where this distinction is complete are called identifier / locator split (or separation) approaches; see Section 5.0.

The history of how the IP addresses have gained their current dual role is pretty interesting. The big change in the way was the introduction of Classless Inter-domain Routing (CIDR), which took place in early 1990's. Additionally, both before that and well after it, the desires to perform policy-based routing arose and became gradually more and more complex. Let's delve a little deeper into this fascinating history, based on the ongoing work by Davies and Doria [19]. It contains a number of very interesting lessons to learn about large-scale inter-networking.

Pre-CIDR

During the early stages of inter-networking, the IP suite was still mainly in use on the ARPANET and the relatively small scale first phase NSFnet. This was a effectively a single administrative domain with a simple tree structured network in a three level hierarchy connected to a single logical exchange point (the NSFnet backbone). In the second half of the 1980s, the NSFNET was starting on the growth and transformation that would lead to today's Internet.

With the increasing complexity of the NSFnet and the linkage of the NSFnet network to other networks there was a desire for policy-based routing, which would allow administrators to manage the flow of packets between networks. The first version of the Border Gateway Protocol (BGP-1) was designed to work on a hierarchically structured network, such as the original NSFNET, but could also work on networks that were at least partially non-hierarchical. BGP-1 was the first real path-vector routing protocol and was intended to relieve some of the scaling problems as well as providing policy-based routing.

Routes were described as paths without any associated cost metric. This way of describing routes was explicitly intended to allow detection of routing loops. It was assumed that the intra-domain routing system was loop-free with the implication that the total routing system would be loop-free if there were no loops in the path, represented as Autonomous System (AS) numbers. Autonomous System (AS) numbers themselves were a 'fix' for the complexity that developed in the three tier structure of the NSFnet.

Meanwhile, the OSI architects, led by Lyman Chapin, were developing a much more general architecture for large scale networks. They had recognised that no one node, especially an end-system (host) could or should attempt to remember routes from "here" to "anywhere" — this sounds obvious today but was not so obvious 20 years ago. They were also considering hierarchical networks with independently administered domains. This led to a vision of a network with multiple independent administrative domains with an arbitrary interconnection graph and a hierarchy of routing functionality. Further refinement of the model occurred over the next couple of years.

Practical experience, IETF IAB discussion (centred in the Internet Architecture Task Force), and the OSI theoretical work were by now coming to the same conclusions:

- Networks were going to be composed out of multiple administrative domains (the federated network).
- The connections between these domains would be an arbitrary graph and certainly not a tree.
- The administrative domains would wish to establish distinctive, independent routing policies through the graph of Autonomous Systems.
- Administrative Domains would have a degree of distrust of each other, which would mean that policies would remain opaque.

Policies would primarily be interested in controlling which traffic should be allowed to transit a domain (to satisfy commercial constraints or acceptable use policies) thereby controlling which traffic uses the resources of the domain. The solution adopted by both the IETF and OSI was a form of distance vector hop-by-hop routing with explicit policy terms. The new protocols explicitly associated policy expressions with the route by including either a list of the source ASs that are permitted to use the route described in the routing update, and/or a list of all ASs traversed along the advertised route.

Over the next three or four years successive versions of BGP were deployed to cope with the growing and by now commercialised Internet. From BGP-2 onwards, BGP made no assumptions about an overall structure of interconnections allowing it to cope with today's dense web of interconnections between ASs. BGP version 4 was developed to handle the change from classful to classless addressing. By the time the NSFnet backbone was decommissioned in 1995, BGP-4 was the inter- domain routing protocol of choice and OSI's star was already beginning to wane. [19]

Nimrod

Nimrod was a scalable routing architecture, first suggested by J. Noel Chiappa, designed to support a dynamic inter-networks of arbitrary size, to provide service-specific routing in the presence of multiple constraints, and to admit incremental deployment throughout the inter-network. The key features of Nimrod included representation of inter-connectivity and services in the form of maps at multiple levels of abstraction, source- and destination-controlled route generation and selection based on maps and traffic service requirements, and source- and destination-controlled message forwarding according to the routes selected.

The Nimrod architecture was capable of representing an inter-network as clusters of entities at multiple levels of abstraction. Clustering was meant to reduce the number of entities visible to routing. Abstraction was used to reduce the amount of information required to characterise an entity visible to routing. The

architecture supported restricted distribution of routing information, both to reduce resource consumption associated with such distribution and to permit information hiding. Each cluster determined the portions of its routing information to distribute and the set of entities to which to distribute this information. Moreover, recipients of routing information could be selective in which information they retained. Furthermore, the architecture encourage caching of acquired routing information in order to reduce the amount of resources consumed and delay incurred in obtaining the information in the future. [20]

At one point in time Nimrod, with its addressing and routing architectures, was seen as a candidate for IPng. History shows that it was not accepted as the IPng, having been ruled out on the grounds that it was felt to have too many open research question. Instead, IPv6 has been put forth as the IPng. However, there is another sense in which study of Nimrod and its architecture may be important to deriving a future domain routing architecture. Nimrod can be said to have two derivatives:

- MPLS in that it took the notion of forwarding along well known paths.
- Private Network-Node Interface (PNNI) in that it took the notion of abstracting topological information and using that information to create connections for traffic.

It is important to note that whilst MPLS and PNNI borrowed ideas from Nimrod, neither of them can be said to be an implementation of this architecture. [19]

CIDR

The Nimrod lesson going mostly ignored within the IETF, the IETF was soon forced to redesign the IP address space to meet the scalability requirements. Classless Inter-Domain Routing (CIDR), introduced in RFC 1519, was a major paradigm shift to establish a provider-based addressing and a routing hierarchy. It made it possible to create multiple hierarchical tiers; most tiers were envisioned to be internet service providers. Provider-based address space allocation was the new model, and BGP would evolve to BGP-4.

CIDR made it possible to have just one routing entry in a router for a whole block of networks. The goal of CIDR was to reduce routing entries in the backbone routers, which began to overflow due to the huge number of entries needed for class C networks (up to about 2 million). After implementing CIDR, that number decreased significantly, allowing some more time for developing longer term solutions. However, as we can observer today, this opportunity was mostly left unused and we are again entering a period of relatively high number of routing table entries.

The problem with CIDR becomes visible whenever a customer changes the provider but wants to keep the IP addresses. The old provider still announces the route to the entire block enclosing the customer's address while the new provider must announce a route to the customer-specific subnet. As a consequence, there will be two routes for that net, the CIDR route and the single route. By default, the more specific route will be used, with the disadvantage of needing a new entry in a backbone router routing and forwarding tables, a phenomenon which CIDR should have prevented.

The present quandary

Since the IAB Routing and Addressing Workshop [21] in October 2006, the current problems (and non-problems) with the routing and forwarding tables have surged to a the knowledge of a wider audience; basically, every IETFer and many other people know that we may be close to the limits of the current routing system. While there are wildly disagreeing opinions on how bad the problem is and what are its main causes, it seems to be a fact that the routing table is growing faster than high speed routing technology. More specifically, the static memory and ASIC technologies that the high speed backbone routers rely on have during the last few years developed somewhat slower than the generic processor architectures and dynamic memory, and clearly slower than the routing system. Therefore, the high speed

routers are becoming relatively more expensive compared to other computing and communications hardware. This, in turn, has led different communities to look for solutions from different directions; for example, there are many research projects aiming to use commodity hardware for high speed routing.

4.2 Jack up proposals

The IETF reaction to the looming routing system choke has mainly focused on the so-called jack up proposals, with some related interest towards identifier / locator split (see Section 5.0).

At the time of this writing (summer 2007) the discussion about the jack-up approaches is heated. At several IETF and IRTF mailing lists people have proposed something between half a dozen and a dozen different approaches to alleviate the routing and forwarding table scalability problems. Due to the rapid development and related flux, it appears not really feasible to describe any specific approaches, as they constantly borrow features from each other. Hence, instead, we try to concentrate to the fundamentals.

What is common to most of the proposals is that they attempt to "jack up" the protocol stack by introducing a new layer below the IP layer. In other words, the IP addresses as we currently use them are to be relegated from their major role as topology-related location identifiers, mainly to serve as node identifiers. However, in most of the cases the IP addresses continue to partially function as locational identifiers, at least within the scope of local networks. For example, even within the intranet of a large multi-national corporation spanning several continents the IP addresses would continue to be used as today. However, in the global Internet there would be a new layer below, taking care of "carrying" IP packets to a suitable egress point, such as a customer-premises edge router.

Hence, the basic idea is not to change the hosts or end-user equipment at all. Instead, the big ISPs would collaborate to upgrade their interconnected backbone networks so that IP addresses are no longer used as addresses but more like location-independent node (or subnetwork–node) identifiers. The approaches to accomplish this vary widely, from interconnected MPLS networks through IP tunnelling to proxied identifier / locator split approaches. Consequently, also the properties of the approaches differ widely, providing different scalability, manageability, and flexibility characteristics.

At the time of this writing (summer 2007), it remains to be seen if any of the currently proposed approaches manages to gain IETF consensus or practical deployment. It may well be that the practical approaches that the ISPs are deploying anyway may outpace the discussion at the IETF, leading to e.g. interconnected MPLS or GMPLS networks, while the user community may at the same time be working towards a full-blown identifier / locator split, thereby gaining a mobility and multi-homing solution that is relatively independent of the underlying connectivity providers.

5.0 IDENTIFIER / LOCATOR SPLIT

As briefly discussed several times above, an IP address serves two fairly different roles. Firstly, the application level connections, i.e., TCP sockets and UDP datagrams, and bound to and identified with IP addresses. Secondly, the IP addresses are used by the routers to forward the packets towards the right location. While this duality has served us well for a long time, the relatively recent developments in IP mobility and multi-connectivity are creating new requirements. These new requirements, together with the changed security environment, are driving the architecture towards some sort of identifier / locator split.

5.1 A practical approach: Host Identity Protocol

The Host Identity Protocol (HIP) is a new piece of technology that implements the identifier / locator split. HIP integrates IP-layer mobility, multi-homing and multi-access, security, NAT traversal, and IPv4/v6

interoperability in a novel and simple way. The result is much simpler than trying to implement these functions separately, using technologies such as Mobile IP, IPsec, ICE, and Teredo. In a way, HIP can be seen as restoring the lost end-to-end connectivity across various IP links and technologies, this time in a way that is secure and supports mobility and multi-homing. HIP also provides new tools and functions for future networking needs.

The basic idea of HIP is to add a new layer to the TCP/IP stack. At this new layer, hosts (i.e. computers) are identified with new identifiers, Host Identifiers, which are public cryptographic keys. As applications open connections and send packets, they no longer refer to IP addresses but to these public keys. HIP has been designed to be fully backwards compatible to applications.

As HIP is a topic of the third presentation [1] in this series, we defer further discussion of HIP there.

6.0 INTERNET TELEPHONY

As we briefly alleged in the first presentation in this series, there seems to be two fundamentally different types of communication needs. In one hand, there is the need for relatively-non-real-time information delivery. In the other hand, there is the need for real-time delivery of interactive voice and other media, often indicated as interactive multi-media[4].

The basic difference between these two classes of communication is tolerated delay. In the world of interactive multi-media, the usually referred-to human factors constant is 200 ms: the end-to-end round-trip delay must not exceed 350 ms or usability will suffer. For many current multiplayer games, delay sensitivity is even larger. Given that it takes some 200 ms for the electrical signal just to circulate the 40000 km earth diameter, the time budget for any processing, such as packetisation and codecs, is minimal. On the other hand, with even slightly larger delay tolerance, such as a few seconds, the situation changes a lot. Instead of being forced to aim for optimising the network for minimal delay, the network has now enough of time to buffer the packets, though possibly only very briefly.

Against that background, it must be understood that the IP Multimedia Subsystem (IMS) [22] and, to some extend, its main constituent protocol, the Session Initiation Protocol (SIP) [23], are heavily geared towards providing the real-time characteristics needed by interactive voice and multi-media communications. When the delay budget is larger, these solutions may or may not be appropriate, depending on other requirements.

6.1 IMS and SIP: Bringing sessions where they are needed

The IP Multimedia Subsystem (IMS) is an architectural framework for delivering interactive multimedia over IP-based networks. It was originally designed by the wireless standards body 3rd Generation Partnership Project (3GPP), and is part of the vision for evolving mobile networks beyond GSM. Its original formulation (3GPP R5) represented an approach to delivering "Internet services" over GPRS. This vision was later updated by 3GPP, 3GPP2 and TISPAN by requiring support of networks other than GPRS, such as Wireless LAN, CDMA2000, and fixed line. In a word, from the business point of view, the IMS is a standardised way to enable communication between multiple networks run by multiple operators with multiple services and over multiple accesses. The IMS gives the operators the opportunity to co-operate to control and support all users with global service reachability, continuity, adaptability, and trustability; whether this leads to a socially desirable result or not depends on the micro-economic relation between the users and the operator.

[4] Of course, there are also other, more business related aspects in networking, such as coordinated service provisioning, control over the used applications, etc, which all are very relevant to IMS and part of the reasoning behind it. However, as we are ignoring those aspects elsewhere in these presentations, we will, to a large extent, ignore them also here.

To ease the integration with the Internet, IMS as far as possible uses IETF protocols. IMS is not intended to standardise applications itself but to aid the access of multimedia and voice applications across wireless and wireline terminals, i.e. aid a form of fixed mobile convergence (FMC). This is done by having a horizontal control layer that isolates the access network from the service layer. Services need not have have their own control functions, as the control layer is a common horizontal layer. A user could, for example, pay for and download a video clip to a chosen mobile or fixed device and subsequently use some of this material to create a multimedia message for delivery to friends on many different networks. A single IMS presence-and-availability engine could track a user's presence and availability across mobile, fixed, and broadband networks, or a user could maintain a single integrated contact list for all types of communications.

From our present discussion point of view, it must be understood that IMS is heavily geared towards providing commercial, pay-for services in operator-controlled networks in an open manner. From that point of view, it is a key technology to delivering dependable multimedia services with telecom-grade quality of service across fixed and mobile accesses. For operators, it creates new opportunities to deliver attractive, easy-to-use, reliable and profitable multimedia services with existing, often legacy-based, telecom services. Users benefit by being able to enjoy attractive converged multiple services regardless of access network and device. IMS is designed to allow the operators to climb up the value chain and take a more active part in service delivery.

An assumed key feature of IMS is that it allows the SIP or related signalling to be used to set up delay-guaranteed communication paths between multimedia terminal equipment, thereby providing the users the experience of real-time, delay-free communications. However, the IMS provides this in a fairly complex manner, at least if compared with the traditional Internet architecture, involving a layered architecture where the signalling path and media delivery path are logically completely separated from each other. While this is beneficial from a business point of view where the operator wants to have maximum control over the network use, it may or may not fit with the desired control model in other environments. The larger number of provided functions adds complexity and cost. For example, in geographically limited environments (where the propagation delay is not a big issue and where network-supported service combination is not demanded) it may be cheaper to simply provide over-provisioned network capacity, perhaps with a few classes of service to differentiated interactive traffic from non-iteractive one, instead of implementing the full control provided by the IMS control functions.

On the other hand, from the traditional telecom networks point of view, IMS allows many functions to be reused for fast service creation and delivery that can be accessed through standardised means. IMS services are hosted by Application Servers and various aspects of service control are defined. For example, IMS defines how service requests are routed, which protocols are supported, how charging is performed and how service composition is enabled.

In a non-IMS telecom network, services are specified and supported by a single logical node, or set of nodes, performing specialised tasks for each specific service. In such networks, each service is an island, with its own service-specific node(s). The only possible way to interface between services – for example, for service composition – is through protocols specific to each combination.

Of course, that is a complete opposite of the traditional Internet point of view, where each end-host can potentially support an unlimited number of services through simple software installation. Since the only "service" the network provides is best-effort unreliable packet delivery, introducing new applications is easy *as long as the delay and bandwidth provided by the underlying network are enough to provide satisfactory user experience and no service composition within the network is needed.* The key point is that while the relatively-uncontrolled Internet technologies are typically enough in the typical, over-provisioned local networks and in the fast majority of cases when the applications are delay-tolerant even in the global, Internet-wide scale, the plain best-effort Internet technologies may prove to be insufficient in

the case of providing world-wide interactive multi-media support. If the operators will take the opportunity to enable the services in a co-operative way, they will have a chance to offer the customers a wider community to communicate with with a better support for multiple services over multiple networks. However, only time will show what will be the case: will Skype-like opportunistic, plain-old-Internet-based multimedia provide adequate service also in the future, or will the IMS be a richer service enabler for also wireline-based Voice-over-IP.

From this point of view, it is important to understand that in IMS-compliant solutions, systems are designed to support multiple IMS Application Servers. This means the same infrastructure can be utilised for new delay-sensitive services, with the implementation effort focusing on the actual service and not on overcoming issues related to basic Internet connectivity or overcoming network congestion when it happens. [24][25][26]

6.2 Beyond IMS: Architectural considerations

Given the tight delay budget of interactive applications, such as telephony and games, discussed above, and the desire to move towards interconnecting information instead of interconnecting nodes, as discussed in the first presentation in this series, it looks likely that the networks will evolve towards providing two very different kinds of services, real-time and data-oriented, with some occasional hybrids.

In the real-time services, the focus will be on delivering bits, typically untouched, in the fastest possible manner between end-user equipment. In the typical case, the communication will be two-way, consisting of a user interactions flowing in both directions. A limited multi-user case, where a group of users are participating to a teleconference or a gaming world, will also be increasingly common. Along this path but in the shorter term perspective, it seems likely that delivering interactive IPTV, in a large scale, will push the load in the core and in many medium band accesses causing the delay to cross the tolerable threshold for real-time services, requiring more control than what is available in today's Internet.

In the data-oriented case, the network will work more as a world-wide data storage than a fast bit pipe. Applications create and consume data, often represented as self-authenticating, integrity protected pieces of information, asking the network to store it and provide it anywhere when requested. The lifetime of the data will vary widely, typically from a few seconds to years. In the typical case, there is no real hurry in getting the data where it is requested: the usual human factor in play is around two seconds.

Occasionally, there may also emerge a need for hybrid services where the network both stores the data and distributes it in a delay-sensitive, fast manner. However, given our current understanding of networking it looks like that the delay-sensitive real-time services in one hand and delay-tolerant storage-sensitive data-oriented services in the other hand will dominate the applications space.

It will be necessary to provide clear incentives for the users and applications to mark their traffic accordingly or to make the services can be easily classified by some other means. However, in the present network architecture there are no incentives for peer-to-peer data storage or Skype-type interactive services to announce their difference. On the contrary, as a result of the current competitive situation, all such new services are very similar and evasive for classification, leading to a socially undesirable equilibrium; i.e., non-optimal use of the resources.

How the networks will evolve to take into account this dual-pronged nature of communication is currently an open question.

7.0 REFERENCES

[1] Pekka Nikander, "In-depth Look at the Host Identity Protocol (HIP): Providing Agile Mobility, Multi-homing, and Security," NATO IST-070 Lecture Series on "Emerging Wireless Technologies. Oct 2007.

[2] Erik Dahlman, et.al, "The 3G Long-term Evolution- Radio Interface Concepts and Performance Evolution", 2006 IEEE 63rd Vehicular Technology Conference, Melbourne, Australia, 7-10 May, 2006.

[3] 3GPP, "Technical Specification Group Radio Access Network; Requirements for Evolved UTRA (E-UTRA) and Evolved UTRAN (E-UTRAN) (Release 7)," 3GPP Technical Report TR 25.913 V7.3.0, 2006.

[4] Carsten Ball, "LTE and WiMax — Technology and Performance Comparison", EW2007 Panel, Tuesday, 3rd April, 2007, http://www.ew2007.org/Documents/Comparison_LTE_WiMax_BALL_EW2007.pdf

[5] WiMAX, A New Broadband Wireless Technology, WiMAX Forum, http://www.wimaxforum.org/news/downloads/supercomm_2005/WF_Day_in_a_Life_with_WiMAX_Final.pdf

[6] Luis Alonso and Ramón Agustí, "Optimization of wireless communication systems using cross-layer information," Signal Process 86:8 (Aug. 2006), 1755–1772. DOI= http://dx.doi.org/10.1016/j.sigpro.2005.09.029

[7] T. V. Lakshman and U. Madhow, "The performance of TCP/IP for networks with high bandwidth-delay products and random loss," IEEE/ACM Transactions on Networking 5:3 (Jun. 1997), 336–350. DOI= http://dx.doi.org/10.1109/90.611099

[8] Teemu Koponen, Mohit Chawla, Byung-Gon Chun, Andrey Ermolinskiy, Kye Hyun Kim, Scott Shenker, and Ion Stoica, "A Data-Oriented (and Beyond) Network Architecture", in Proceedings of SIGCOMM'07, Kyoto, Japan, August 27–31.

[9] Robert Moskowitz and Pekka Nikander, "Host Identity Protocol (HIP) Architecture," RFC 4423, Internet Engineering Task Force, May 2006. http://www.ietf.org/rfc/rfc4423.txt

[10] Pekka Nikander, "Evolution of Networking: Megatrends, Current Problems, and Future Directions," NATO IST-070 Lecture Series on "Emerging Wireless Technologies. Oct 2007.

[11] Charles E. Perkins (ed.), "IP Mobility Support for IPv4," RFC 3220, Aug 2002.

[12] David B. Johnson and Charles E. Perkins, "Mobility Support in IPv6," RFC 3775, Jun 2004.

[13] Charles E. Perkins, "Mobile IP," IEEE Communications Manazine 35:5, May 1997, pages 84–99. DOI=10.1109/35.592101

[14] Vijay Devarapalli, Ryuji Wakikawa, Alexandru Petrescu, and Pascal Thubert, "Network Mobility (NEMO) Basic Support Protocol," RFC 3963, Internet Engineering Task Force, Jan 2005.

[15] Chan-Wah Ng, Fan Zhao, Masafumi Watari, and Pascal Thubert, "Network Mobility Route Optimization Solution Space Analysis," work in progress, Internet draft draft-ietf-nemo-ro-space-analysis-03, Sep 2006.

[16] Erik Nordmark and Marcelo Bagnulo-Braun, " Shim6: Level 3 Multihoming Shim Protocol for IPv6," work in progress, Internet draft draft-ietf-shim6-proto-08.txt, May 2007.

[17] Tuomas Aura, "Cryptographically Generated Addresses (CGA)," RFC 3972, Mar 2005.

[18] Jari Arkko (ed.), James Kempf, Brian Zill, and Pekka Nikander, " SEcure Neighbor Discovery (SEND)," RFC 3971, Internet Engineering Task Force, Mar 2005.

[19] Elwyn Davies and Avri Doria, "Analysis of IDR requirements and History," work in progress, Internet draft draft-irtf-routing-history-05.txt, Feb 2007.

[20] Isidro Castineyra, J. Noel Chiappa, and Martha Steenstrup, "The Nimrod Routing Architecture," RFC 1992, Aug 1996.

[21] David Mayer, Lixia Zhang, Kevin Fall, "Report from the IAB Workshop on Routing and Addressing," work in progress, Internet draft draft-iab-raws-report-02.txt, April 2007.

[22] Gonzalo Camarillo and Miguel-Angel Garcia-Martin, "The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds," John Wiley & Sons, August 4, 2004. 406 pages.

[23] Gonzalo Camarillo, "SIP Demystified," McGraw-Hill Professional, August 28, 2001. 320 pages.

[24] "IP Multimedia Subsystem", a Wikipedia article, referenced on 19 July 2007.
http://en.wikipedia.org/wiki/IP_Multimedia_Subsystem

[25] Tim Hills, "IMS Guide", Light Reading Reports, March 24, 2005.
http://www.lightreading.com/document.asp?doc_id=70728

[26] White Paper, "Introduction to IMS", Ericsson AB, March 2007. 284 23-8123 Uen Rev A.