

# How to access home while on the road

Tapio Janasik  
Helsinki University of Technology  
tapio.janasik@hut.fi

## Abstract

Homes are increasingly equipped with a local network of one or more computers. The number of various services such networks provide for their users is already considerable, and will most certainly increase in the future; consider for instance file storing and sharing, programming a digital set-top box, gaming, and instant messaging. Some services can even be accessible while outside of the home network.

Unfortunately, very few homes have static IP addresses that remain constant across time. As computers and networks are identified by means of IP addresses, users travelling outside their homes thus face a problem when trying to access services available in their home network.

This study reviews various methods suggested to remedy this problem. A discussion of the classic DNS proves it insufficient for coping with the problem, but its dynamic extension provides a remedy. Other reviewed methods include rendezvous services, peer to peer networks and the IP multimedia subsystem. Among them, no definite hierarchy or preference order is found. However, this study proposes a set of simple criteria for choosing a suitable method.

**KEYWORDS:** home networking, host identification, DNS, Rendezvous server, IP multimedia subsystem (IMS), peer to peer networks

## 1 Introduction

### 1.1 The Problem

Let us consider the following situation. You are at work having a casual conversation over a cup of coffee with your colleagues. They mention a re-run of your favourite TV show that will be aired during the afternoon. At home, you have a digital set-top box connected to your computer, which, in turn, is connected to the Internet by means of an ADSL modem. The computer is thus assigned an IP address from the ISP's address space by means of DHCP [8].

Now, as you were completely unaware of the show being aired, you reach for your mobile phone to program the set-top box to start recording. Maybe you have an application on your mobile designed specifically for this task, perhaps you just open a terminal connection to your home computer. At any rate, if your home network had a designated, permanent IP address, such a task would not pose much of a problem.

Alas, very few homes have such addresses, simply because there are not enough of them around. Instead, the length of the period of time your network has a certain IP address ("the lease") is determined by the ISP's DHCP server.

While outside of the perimeters of the home network, you and other users thus need to be able to find the home network in some way; not only to be able to record your favourite TV shows, but to access any resources or services you might have available on your home computers, such as various files, home surveillance or even baby-sitting applications. Luckily, several such methods have been made available, some of which of fairly recent origin.

However, due to the scarcity of IPv4 addresses many networks use Network Address Translations (NATs). In addition, and introducing a wealth of new issues, any diligent home network administrator protects the home computers with a firewall. Issues pertaining to firewall or NAT traversal are thus also relevant, as well as more general security issues, such as authentication.

This paper reviews and compares some potential solutions to the aforementioned problem within the IP architecture. In addition to presenting their technical properties, this paper also comments their socio-economical impact as well as the security models they impose.

### 1.2 The structure of this paper

The next section will present several methods for finding home. First, in section 2.1, we will briefly discuss the Internet's Domain Name System, and describe how it, in its traditional form, is inadequate to handle the problem. As a potential remedy, we will review the dynamic extension to DNS.

Section 2.2 goes on to consider rendezvous techniques that tackle the problem by means of special servers keeping track of the home network's changing IP addresses, and mediating the connection between the home and the mobile user. As implementations of this method we briefly consider the Session Instantiation Protocol, and the Host Identification Protocol.

Then, in section 2.3, we will review a way of solving the problem without need for services from an operator or any other central point of service, but instead relying on a distributed network of peers.

Section 2.4, in turn, will review an operator-driven solution based on recent developments within 3GPP. This solution also heavily builds on the developments within the Session Initiation Protocol, but essentially enables providing a user with a wealth of services. Further, section 2.5 briefly considers the impact of IPv6 on the problem setting.

Finally, section 3 will conclude the discussion and present criteria for choosing between the alternatives.

## 2 Methods for finding home

### 2.1 The Domain Name System

#### 2.1.1 DNS operation

The Domain Name System (DNS, [12]) is a key component in the workings of the Internet, translating domain names to IP addresses, the former being easier for humans to handle than the latter.

The basic functionality of DNS is rather straight-forward. Corresponding to the hierarchical name-space of the Internet, there is a hierarchy of Domain Name Servers, each of which is authoritative for one or more zones of the name-space. For a particular zone, then, a DNS server maintains a set of Resource Records (RR) pairing domain names to IP addresses. Some DNS servers merely cache this data, and do not store any RR database of their own.

The DNS thus amounts to a distributed storage and managing of the data. The organisations administrating the system range from governmental institutions running so called top-level domains (e.g. “.fi”, “.uk”, “.gov”) and ISPs to companies and educational institutions managing their own domains.

As further components of the DNS infrastructure, moreover, each host connected to the Internet has, as local components, a resolver and a cache.

Figure 1 sketches the basic DNS operation. When a user wishes to connect to a certain remote host, say, in the home network, the resolver checks if the host name is already cached locally. If not, it sends a query to the nearest name server (step 1 in figure 1). The name server then checks if it has the required data available, either cached or stored in its own RR database. If it does not, it sends the query onwards to other name servers, possibly also querying for the correct name server for the domain of the required remote host (steps 2 to 3, somewhat simplified in the figure, and steps 4 to 5). When it receives a reply, it caches it and sends it back to the original host (step 6), which then can commence communicating (step 7).

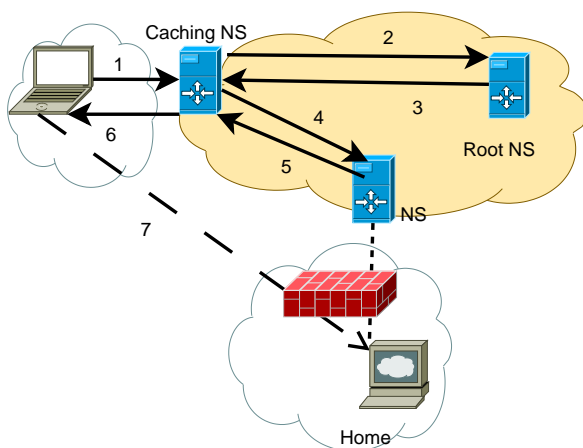


Figure 1: Basic DNS operation

Caches do not remain forever, however; each RR has a time-to-live property (TTL) specifying how long it can be cached. The TTL is assigned by the administrator of the

current zone. It can in principle be anything, even zero to prevent caching altogether. Two decades ago, interestingly, Mockapetris suggested employing TTLs in the order of days due to “the realities of Internet performance” [12, p. 12].

#### 2.1.2 DNS security

The Internet was initially a network of trustworthy, well-behaving users from the scientific community. Correspondingly, DNS was not intended to have to face malicious users. For instance, there is no way a user or a client software can ascertain that the reply originates from a legitimate DNS server.

Further, should the reply indeed come from a malicious server, it might amount to a name chaining attack, and include false data. This could have disastrous consequences, if for example queries for web-based banking services were redirected to sites pretending to be genuine but which instead gathered confidential information and abused it (for an analysis of various forms of attacks against DNS, see [3]; cf. also [6]).

To counter this kind of threats, DNS Security Extensions (DNSSEC, [2], originally in RFC 2065) was developed. DNSSEC is intended to provide for the integrity of DNS operation, both with respect to origin of the data as well as the transmitted data itself. In other words, the aim is to ensure that the data originates from a genuine source, and that the data has not been interfered with during transmission.

DNSSEC accomplishes this through digital signatures; a name server signs its zone data with the private key affiliated to that zone. Then, assuming it has acquired the public key in a reliable way, a resolver in the caching name server can verify that the data is authentic. In order not to leave any loopholes, further, the zone’s private key should be secured and stored offline.

Note that the signature is indeed verified in a caching name server, and not in the client originating the query — the latter is indeed known as a “security-oblivious stub resolver” [2]. Consequently, the last hop of the DNS reply is unaccounted for (for further discussion, see [6]).

From a functional point of view, nevertheless, as home network IP addresses change over time, the static resource records maintained by the name servers will constantly be obsolete. In other words, something else is required. Accordingly, we will next consider an extension to the basic DNS that may solve the problem.

#### 2.1.3 Dynamic updating of DNS records

As it was originally designed, DNS is a system for maintaining static data for long periods of time. Should changes occur, the data must be edited manually.

In contrast, Dynamic DNS [18] is intended to enable real-time updating of resource records. When the ISP’s DHCP server assigns the home network a new IP address, the DHCP server, a local gateway representing the home, or a host in the home network can immediately send an update message to the relevant name server. From the viewpoint of the mobile user, thus, connecting home is easy, and accomplished by employing DNS queries in the usual way.

To this end, Dynamic DNS introduces the UPDATE message. With this message, it is possible to add or delete Resource Records from a specific zone. In order to facilitate swift propagation of the new IP address to other name servers, further, the time-to-live property of an RR is minimised.

From a security point of view, unless the update is authorised in some way, a malicious user might just update a web banking domain to have a new IP address of a fake site of the aforementioned kind. To counter this, Wellington extended DNSSEC to secure also dynamic updating by including a transaction signature to each transaction, by means of which requests are authenticated [19].

However, once a zone has been updated, the corresponding signature has to be updated as well. As such updates are expected to be frequent, the zone's private key must in effect be available on-line, thus potentially exposing it to attacks. Further, the transaction signature has to be computed separately for each transaction, thus also forcing the private key to be on-line. The problem becomes especially pertinent in root name servers hosting vast numbers of records.

This exposure could in principle be solved with cryptographic means, e.g.  $(m, n)$  threshold schemes, where the secret is divided into  $n$  shares such that  $m$  shares are sufficient to recover the secret. Such solutions would nevertheless introduce some further computational overhead.

However, in the current setting, the problem of ascertaining the identity of the other party may be solved by somewhat simpler means. For instance, the two parties may employ a shared secret on the application level. If the secrecy of the message itself is not particularly relevant, Message Authentication Codes (MAC) are a feasible alternative. Also, given that the mobile user visits home at least occasionally, and that the number of users is limited, key distribution is not an issue.

### 2.1.4 Summary

Let us briefly sum up the previous discussion. The DNS is an inherent part of the current Internet, and the administration of DNS servers is an integral part of managing the whole Internet. In its original form, however, DNS is unable to provide a mobile user with a way to connect to a home network with a dynamically changing IP address.

As a remedy, dynamic DNS introduces a method to update DNS records on the fly. As to administration, dynamic DNS also typically involves an ISP or some other organisation to manage updating of the DNS. Note that, in March 2007, a Google Directory search yields 67 providers of dynamic DNS services.

## 2.2 Rendezvous services

### 2.2.1 Establishing the initial contact

Let us now consider a method, which does not as much change the way DNS operates, but instead introduces a new service mediating between home and the mobile user. This new service employs the rendezvous mechanism.

Abstracting somewhat from implementation details, a Rendezvous Server (RVS) functions in effect as a directory

service, keeping track of the changing IP addresses of hosts it services. Once the home network has initially registered at the RVS, it subsequently updates its IP address whenever it changes. Then, the RVS provides the mobile user with the first point of contact when initiating the communication with the home network — hence the name of the mechanism. Importantly, the RVS is assumed to remain available at the same IP address at all times.

This basic mechanism is thus rather similar to dynamic DNS update. However, instead of mapping domain names to IP addresses, now the DNS server only keeps track of pairs of domain names and IP addresses of corresponding RVS servers. The latter then maintain the pairing of domain names and currently related host IP addresses. Consequently, by providing a single point of contact, the RVS mechanism reduces problems related to caching of DNS data.

The RVS operation is sketched in figure 2. As usual, the mobile user first queries the nearest name server with the home network domain name. However, instead of replying with the home network's IP address, the name server sends the IP address of the RVS next to the home network. Then, the mobile user contacts the RVS, which replies with the current IP address of the home network.

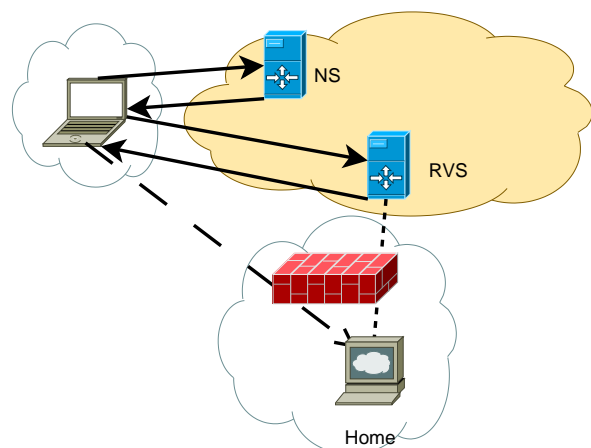


Figure 2: Basic RVS operation

So far, the RVS operation is rather similar to that of a home agent in Mobile IP (see [14]) in that the latter also maintains the current location of a mobile host, and serves as the initial point of contact. However, in addition, a home agent tunnels all traffic to the mobile host, thus in effect hiding its current address (so called care-of-address) from other hosts. In contrast, an RVS usually does not do this, but instead leaves the communication to the two parties after establishing the initial contact.

Well-known implementations employing the RVS mechanism include the Session Initiation Protocol (SIP, [15]) — an application level control (or signalling) protocol, used in establishing, modifying, and terminating sessions between user agents. SIP applications include Internet telephony and instant messaging, both relevant services from a home networking point of view. Furthermore, being an application level protocol, with focus on services bound to human users, SIP identifies users by means of identifiers similar to email addresses (e.g. `sip:alice@someisp.com`).

SIP utilises the rendezvous mechanism with so called registrar servers that accept registrations from user agents, containing information about their current location. This information is then used by proxy servers that help user agents to locate other user agents.

So, the RVS mechanism provides for a way to find home. However, as discussed in section 2.1 above, we also need a way to make sure we are communicating with the intended party. To this end, SIP does not define new mechanisms but relies on existing ones. For instance, such lower level security mechanisms include TLS and IPsec.

We will briefly return to SIP in section 2.4 below. Before that, let us consider another approach to ascertaining the identity of the other party.

### 2.2.2 Host identities (re)considered

In the current Internet architecture, IP addresses have a twofold function. Firstly, they **locate** hosts as destinations to which packets should be routed. Secondly, IP addresses **identify** hosts (or their network interfaces, to be exact). When a host gets a new IP address, it is usually due to change of location, not of identity.

Of course, as the current discussion is about finding the home network, the physical location of which is unlikely to change, we would probably not be very inclined to talk about change of location either. Nevertheless, IP addresses locate hosts in terms of routing, and thus location here pertains to logical network topology, not to the physical or geographical whereabouts of the home.

Now, the Host Identification Protocol (HIP) architecture [13] aims at distinguishing these two functions, and introduces a new name-space of Host Identifiers (HI). This new name-space in effect constitutes a new layer between those of networking and transport. Consequently, transport layer associations are now bound to Host Identities instead of IP addresses.

In this new architecture, hence, IP addresses still locate hosts, but HIs are used for identification and authentication — they are in effect the public keys of asymmetric key pairs, the private ones being possessed by the hosts. Importantly, a HI is (statistically) unique to a particular host. Further, HI is presented as a hashed value called the Host Identity Tag (HIT).

In the HIP model (see the work in progress [11]), then, HIP hosts register and update their current IP addresses at an RVS. Correspondingly, when the mobile user wants to connect home, it contacts the RVS with the home HIT. The RVS, as above, relays the message to the home network IP address. Once this initial contact is established, however, the two parties start communicating directly with a secured channel, without the RVS.

HIP also provides a way to manage the eventuality of either party changing its IP address during communication. When a host obtains a new IP address, namely, the HI remains the same. Still, during an ongoing connection, the other party has to be informed of the change, to prevent the connection from breaking. HIP provides a mechanism for this, in the form of a 3-way UPDATE packet exchange, where the packet includes the new IP address.

Now, HIP embodies fairly recent developments within the Internet architecture. As such, it is largely work in progress, and not universally supported by the current infrastructure. Among this work in progress are attempts to enable NAT traversal in HIP communication (see [17]).

### 2.2.3 Summary

The rendezvous mechanism utilises a service that helps the mobile user in establishing the contact home. As a general concept, the rendezvous mechanism can be common to a host of network architectures. Indeed, it is implemented for instance in HIP and SIP.

Even if the use of RVS does not in itself require dramatic changes in the way DNS currently operates, its implementations may do so. For example, introducing a new layer of Host Identifiers does impose some changes, e.g. to DNS resource records.

## 2.3 Peer to peer networks

### 2.3.1 De-centralised networking

So far, the solutions we have considered have relied on the availability of various backbone architecture and administration. Another way of approaching the problem, in contrast, is opened up by peer to peer (P2P) networks.

The basic idea in P2P networking is that various services or computing resources are distributed among several peers. No host is thus more important than others, in contrast to the methods discussed in previous sections. File sharing is a popular, and controversial, P2P application. Further, thanks to this distributed model, also risks of for instance loss of data are minimised.

As to finding the data, P2P networks employ various methods. Some P2P networks include a central database pairing e.g. file names to locations where the file is stored. Alternatively, a host searching for a particular piece of data could just broadcast the query to all members of the network. A more systematic, yet equally distributed, way is to employ Distributed Hash Tables (DHT, [4]).

A hash table in general is a data structure consisting of key-value pairs. Hashing the keys, further, provides the index for the table. The hash in effect yields a so called bucket, in which the corresponding value is stored. A lookup operation then retrieves the value for a given key, readily found by means of the hash. A distributed hash table, then, is a hash table that is distributed among several hosts, each playing the role of a bucket.

Consequently, in the P2P setting, distributed hash tables create an overlay network of peers, where keys can be stored and retrieved across hosts. Further, each host also maintains a routing table consisting of other nodes in the network. Should a new host join the network, or a member leave it, the routing table is adapted dynamically. The same applies to the data itself, which is in appropriate portions copied to the new hosts and from the leaving hosts as changes occur.

Figure 3 sketches the way peers could assist in finding home. Assuming an identifier for the home network is given (e.g. domain name or a public key, or a HIP Host Identifier), the hash table entry could consist of the identifier as

the key, and the current IP address as the value. Then, a hash of the identifier would provide the host in which the value will be stored. Whenever the mobile user wants to connect home, hence, he or she would check the P2P network for the home's current IP address. Correspondingly, the home network updates the IP address whenever it should change.

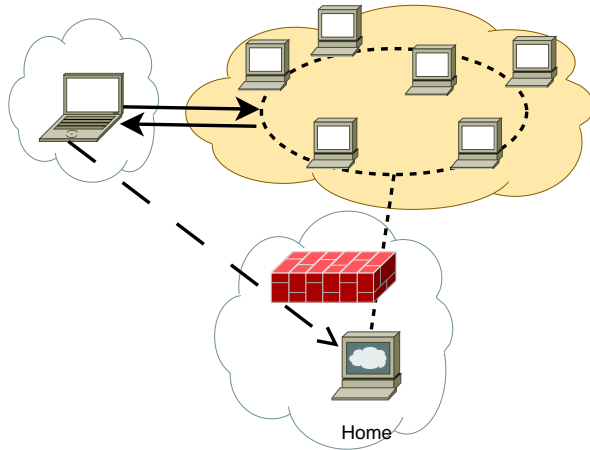


Figure 3: Finding home with help from the neighbours

However, this non-hierarchical architecture may introduce a considerably less desirable consequence. Namely, the risk of malicious users entering the network is considerable. Fortunately, effective cryptographic methods to counter this exist (see [4]).

Now, as mentioned above, not all P2P networks employ distributed hash tables. Instead, some such architectures provide for data lookup by other means, for instance flooding. In a large enough P2P network, flooding would clearly not be a very sustainable solution, but in a smaller scale it might.

Indeed, the Unmanaged Internet Architecture (UIA, [9]) is intended to provide P2P connectivity within a somewhat smaller social network of friends and family. This way, the UIA solution in effect restricts the network to users one personally adds, thus mitigating this risk. Instead of relying on DHTs to store the naming data, UIA then utilises scoped flooding within the social network to discover the current IP address of a host.

Further, UIA employs persistent personal names for various devices as a central component. For instance, Alice's mobile phone could be called `mobile.alice.someisp.com`. Each such personal name is then mapped to a unique and stable endpoint identifier. Similarly to host identifiers in HIP, an endpoint identifier is in effect a hashed public key, also acting as the endpoint of upper layer connections. UIA is thus also able to handle the change of IP address during communication. However, in contrast to HIP, these endpoint identifiers are connected to persons, thus allowing for shared use of one host by several users.

We will in section 2.4 consider a rather different perspective on personal names and identifying users instead of just hosts. Before that, however, let us note that we have so far assumed that the mobile user sends packets directly to the home network, once its current IP address is retrieved from the P2P network — thus basically mimicking the operation of a rendezvous server. Instead of such a two-way setup,

however, all traffic could be relayed through the network by means of the hash table entries. This, indeed, is the basic idea in the Internet Indirection Infrastructure (*i3*, [16]). A further development, *Hi3* [10], aims to combine *i3* with HIP, so that the former in effect functions as the transport mechanism for HIP packets. Also, employing the HI layer would amount to reduced traffic in comparison to *i3* as not all traffic needs to be flooded to the network.

### 2.3.2 Summary

Peer to peer networks definitely bring a new perspective to the current discussion in that all hosts of the network are treated on a par. No host, in other words, is irreplaceable. Consequently, lacking a single point of failure, a P2P network provides for significant fail-safety.

## 2.4 The IP Multimedia Subsystem

### 2.4.1 Towards a mobile Internet

Let us now turn to a more operator-driven framework which focuses on providing cellular access to various Internet services. This framework is the IP Multimedia Subsystem (IMS, [1, 5]), aiming at merging the two worlds of mobile phone networks and the Internet.

Of course, mobile phone users already have rather ubiquitous access to Internet services through 3G networks. However, the impetus for IMS stems from three central requirements presently not satisfactorily accounted for [5, p. 7]:

- **Quality of Service (QoS):** multimedia services are essential in the mobile world, and so is the ability to provide enjoyable multimedia sessions to the users.
- **Charging:** operators may want to impose varying business models on users depending on whether they are engaged in multimedia sessions or just exchanging instant messages.
- **Integrated services:** operators may want to be able to offer their customers different services, including those provided by third parties, in an integrated format.

Especially in order to facilitate appropriate charging, it is paramount that the user is identifiable in the role of a service subscriber. With roots within the mobile world, the IMS provides a natural solution. In addition to identifying the subscriber, essentially, the IMS also enables enforcing user-specific policies to control accessible services.

To this end, from the present perspective the most important components in the IMS architecture are the user databases and the SIP services.

The former include a Home Subscriber Server (HSS), corresponding to a Home Location Register employed in the GSM. The HSS thus registers and stores data about the subscribers. Specifically, the HSS stores mappings from the IMS subscribers to one or more private identities, not unlike the IMSI stored in the SIM of a GSM phone. The private identities need not be known by the users, as they are only used by the network for identifying and authenticating the subscribers. Moreover, a private identity of a subscriber is

further mapped to one or more public identities. The public identities are represented as SIP URIs or TEL URIs, the latter being in effect telephone numbers.

As the session control protocol IMS employs SIP. Correspondingly, the IMS architecture includes SIP registrars and proxies to facilitate creating connections, as described in section 2.2 above.

The IMS thus provides a natural framework for connecting home, particularly with a mobile phone. The HSS can contain several private identities for one subscriber, whereof the home network could be one and the mobile phone another, both readily available to each other. As the private identities can be publicly represented as telephone numbers, finding home would be a matter of simply making a call.

#### 2.4.2 Summary

The IMS comprises a rather extensive framework, the details of which are beyond the present scope. However, in the core of IMS are the rendezvous mechanism in its SIP implementation, together with various databases for identifying the user.

### 2.5 IPv6: IP addresses for everyone?

So far, we have been discussing various methods and mechanisms within the current IPv4 infrastructure. Indeed, the main initial cause of the problem of finding home rises from the insufficient address space of IPv4.

Hence, extending the IP address size from 32 bits to 128 bits, and thus amounting to a dramatic increase in the number of available addresses, IPv6 [7] would definitely seem to provide great relief for the foreseeable future and longer. To be exact, IPv6 increases the number of addresses from  $2^{32} \approx 4 \times 10^9$  to  $2^{128} \approx 3.4 \times 10^{38}$ .

Consequently, instead of deploying some more or less complicated mechanism for the mobile user to call home, in the IPv6 world, the home network could just have a plain, static IP address. In the same vein, so could mobile phones, cars, boats, set-top boxes, saunas, toasters, and other household devices. Home networks would not, in other words, even need NATs anymore.

However, IPv6 is still not very widely supported in the Internet infrastructure. Further, it is still too early to say whether IPv6 addresses in fact will be static. Other solutions are thus required for the time being.

## 3 Analysis and comparison

We have above discussed several alternative methods for finding home. Generalising somewhat, they fall into two categories. Firstly, we could just have static IP addresses. In the current IPv4 world this is not feasible, but in IPv6 it may be possible. Secondly, we can utilise a database of some kind for providing the mobile user with the home's current IP address. The database solution can then take various forms; we have discussed dynamic DNS, rendezvous services, P2P networks, and the IP Multimedia Subsystem.

As criteria for choosing a suitable solution, let us consider the following:

- Management: Who manages the solution, and how hard is it to manage?
- Security: What potential attacks are there?
- Resources: What resources does the solution require?
- Maturity: How mature is the solution, and are there well-known implementations? That is, is it a "safe" and reliable solution?

Table 1 summarises the analysis based on these criteria. A subject for future work is to further specify the comparison, and e.g. to take into account various details of implementations of the methods presented rather generally in this paper, namely rendezvous services and peer to peer networks. Equally, economic aspects are also omitted here as they depend heavily on the particular implementation.

As a final note, let us speculate that as things stand currently, it is not very likely that new innovations requiring new infrastructure or significant changes in the way the Internet works will be widely adapted very soon. Correspondingly, solutions commonly perceived as safe and reliable will comprise the most natural alternatives.

## 4 Summary and conclusions

Let us finally sum up the discussion. Assuming the current IPv4 address space is insufficient for home networks to be able to possess static IP addresses, we have discussed various solutions for finding home. First we discussed DNS, which was not originally even intended to facilitate frequent change of IP address. The dynamic update mechanism of DNS records aims to remedy this.

The rendezvous mechanism aims to solve the problem by introducing an additional component in the DNS infrastructure. This additional service keeps track of the home network's changing IP address, and is the first point of contact for the mobile user. DNS thus only keeps track of the (static) IP address of the rendezvous server. The rendezvous mechanism is implemented by several recent developments.

In peer to peer networks, in contrast, the mediating service providing the mobile user the way home is distributed in the network. Or, better, all traffic could be routed within the network, instead of sent directly to the home network.

Finally, the IP multimedia subsystem employs SIP-like mechanisms to facilitate a mobile user connecting home. The rendezvous service is thus essential also here. This mechanism is primarily intended for mobile phone users.

Of course, IPv6 explodes the IP address space, thus at least in principle enabling all homes and such to have a static IP address. On the other hand, IPv6 is still not deployed widely enough for us to draw such conclusions.

As to choosing a suitable solution, the home network administrator has to take several considerations into account. Some solutions require configuration effort from the administrator, but then also enable the user e.g. to exchange services with his or her social network. However, if the additional effort should prove to be too demanding, an ISP or an operator driven approach would be a more natural choice. Given the current emphasis on mobile presence and mobile

Method	Management	Security	Resources	Maturity
Dynamic DNS	An operator or service provider can manage, some setup at home is needed.	Susceptible to various DNS attacks. DNSSEC has problems, but simpler measures may be sufficient.	Not necessarily very much required at home. A domain name is needed, unless the service provider provides one.	Well-known method with several implementations.
RVS	Some service provider is required.	The RVS method itself does not provide any particular security model, but depends on implementation.	The RVS server is required, most details depend on the preferred implementation.	Rather new technique. Implemented in SIP and HIP, of which the latter is still somewhat experimental.
P2P	Managed by the participating peers. Requires know-how from the user.	May be susceptible to malicious users, requires cryptography to counter them.	Access to a network of peers is required. Not very likely to emerge in large scale.	Experimental implementations so far.
IMS	An operator manages and provides as service.	Employs existing security solutions.	IMS capable mobile phone and operator required.	Employs known methods, such as SIP. Gaining leverage.
IPv6	Managed locally at the host. Set-up requires some know-how.	Includes security mechanisms such as IPsec.	IPv6 capable host and network is required.	Quite mature, but not widely implemented yet despite pressure from the Internet organisations.

Table 1: Comparison of the methods with four criteria

applications, IMS will most likely be an important framework in the future.

## References

- [1] IP multimedia subsystem (IMS), stage 2, December 2006. 3GPP technical specification. Online: <http://www.3gpp.org/ftp/Specs/html-info/23228.htm>.
- [2] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS security introduction and requirements. RFC 4033 (Proposed Standard), 2005.
- [3] D. Atkins and R. Austein. Threat analysis of the Domain Name System (DNS). RFC 3833 (Informational), 2004.
- [4] H. Balakrishnan, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica. Looking up data in P2P systems. *Communications of the ACM*, 46(2):43–48, February 2003.
- [5] G. Camarillo and M. A. Garcia-Martin. *The 3G IP multimedia subsystem (IMS): merging the internet and the cellular worlds*. Wiley, 2nd edition, 2006.
- [6] R. Chandramouli and S. Rose. Challenges in securing the domain name system. *Security & Privacy Magazine, IEEE*, 4(1):84–87, Jan-Feb 2006.
- [7] S. Deering and R. Hinden. Internet protocol, version 6 (IPv6) specification. RFC 2460 (Draft Standard), 1998.
- [8] R. Droms. Dynamic host configuration protocol. RFC 2131 (Draft Standard), 1997.
- [9] B. Ford, J. Strauss, C. Lesniewski-Laas, S. Rhea, F. Kaashoek, and R. Morris. Persistent personal names for globally connected mobile devices. In *7th USENIX Symposium on Operating Systems Design and Implementation*, 2006. Online: <http://www.brynosaurus.com/pub/net/uia-osdi.pdf>.
- [10] A. Gurtov, D. Korzun, and P. Nikander. Hi3: An efficient and secure networking architecture for mobile hosts. Technical Report 2005-2, Helsinki Institute for Information Technology, June 2005.
- [11] J. Laganier and L. Eggert. Host identity protocol (HIP) rendezvous extension. Internet-Draft (expired Dec 9, 2006), June 2006. Online: <http://www.ietf.org/internet-drafts/draft-ietf-hip-rvs-05.txt>.
- [12] P. Mockapetris. Domain names - concepts and facilities. RFC 1034 (Standard), 1987.
- [13] R. Moskowitz and P. Nikander. HIP architecture. RFC 4423 (Informational), 2006.
- [14] C. Perkins. IP mobility support for IPv4. RFC 3344 (Proposed Standard), 2002.
- [15] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP session initiation protocol. RFC 3261 (Proposed Standard), 2002.
- [16] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. Internet indirection infrastructure. In *Proceedings of ACM SIGCOMM*, August 2002.
- [17] H. Tschofenig and M. Shanmugam. Traversing HIP-aware NATs and firewalls: Problem statement and requirements. Internet-Draft (expires Apr 26, 2007), October 2006. Online: <http://www.ietf.org/internet-drafts/draft-tschofenig-hiprg-hip-natfw-traversal-05.txt>.
- [18] P. Vixie, Editor, S. Thomson, Y. Rekhter, and J. Bound. Dynamic updates in the domain name system (DNS UPDATE). RFC 2136 (Proposed Standard), 1997.
- [19] B. Wellington. Secure domain name system (DNS) dynamic update. RFC 3007 (Proposed Standard), 2000.