

# Service Selection in the Access Network

Nick Marly, Dominique Chantrain, Stephane Focant, Koen Handekyn, Koen Daenen, Claudine Batsleer

Alcatel Research & Innovation, Francis Wellesplein 1, B-2018 Antwerp, Belgium

Email: [nick.marly@alcatel.be](mailto:nick.marly@alcatel.be), Voice: +32 3 240 7767

**Abstract** - As users become more and more mobile and broadband access is being deployed at an ever-increasing pace, new ways of deploying services can be examined. Next to ways for accessing global services (Portals), approaches are described to offer local services, based on the knowledge of the user's location in the nodes of the access network. Increasing the intelligence of the access nodes, localized, advanced and personalized services can be offered. Several methods are discussed to implement these local services, in order of increasing functionality.

## I. INTRODUCTION

The combination of broadband access, mobility and roaming are key enablers for the deployment of new services in the access and core network (Internet). We can classify services into two main categories, namely local and global services. Global services are accessible worldwide and are independent of the location of the user, typically these are portal services.

Local services depend on the access point of the user and are strongly connected to the NAS (Network Access Server) or PoP (Point of Presence) of the AP (Access Provider). Typical services are local information (restaurants, movie theaters, hotels, etc.), local government and local business services. To provide a mobile user with local services, the AP and/or SP (Service Provider) has to be aware of the whereabouts of the user.

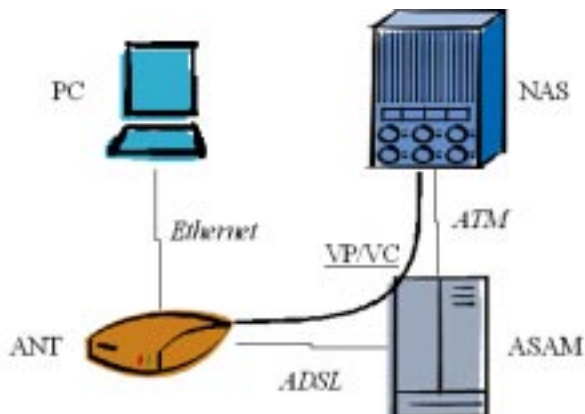


Fig. 1 Typical Access Architecture

In mobile telephony the HLR (Home Location Register) and VLR (Visitor Location Register) are aware of the location of the user. This information can be used to offer locations specific services to the user. The same principle can also be applied for roaming RAS users, who connect via public (wired) access-points, e.g. in hotels, airports, etc. In a typical broadband architecture (Fig. 1), the user will connect through an ADSL modem with the NAS. The NAS knows the VP/VC of the ATM connection, which is providing the IP connectivity to the user. This VP/VC is uniquely linked to the physical wire (twisted pair phone line) and thus with the location of the user.

In this paper some scenarios and architectures are described, giving users connectivity to a server through a NAS and providing local services. The features of the different architectures are discussed. The main example is based on DNS and HTTP, although it could be any server or service.

In the figures further on, a full line between boxes in the access architecture is an interaction based on a standard interface (e.g. HTTP); a dotted line means a non-standardized (proprietary) interaction. We also assume that the NAS does not contain user information, but is able to link an IP-address and physical port to a certain VP/VC and VPN, and thus to the location of the user.

The mechanisms discussed in this paper are complementary to service discovery and announcement as explored in the Service Location Protocol [1].

## II. POP-DEPENDENT SERVER SELECTION

### A. Concept

A service provider has agreements with various network providers all over the world, such that users can access his services via multiple PoPs. The starting point for the user is typically a homepage on the portal of the service provider. Clearly, one portal machine for the entire world is not really feasible in terms of scalability and efficient use of resources. Therefore the service provider has installed multiple portal servers, which are all more or less copies of a master portal. Nevertheless, these local portals may also provide area-specific information (e.g. different in Europe and US). In this way, for users connecting via a PoP in Belgium, the button 'weather forecast' will point to the weather forecast for Belgium. The user connects to this portal typically by

clicking the ‘home’ button in his browser.

We can provide a local service by making the server selection depending on which NAS (= PoP) the user connects to. According to this the (closest) portal server is selected. Multiple NASs in the PoP will typically all point to the same portal server. The required configuration information in the NASs could typically be distributed from a central point.

### B. Architecture & Mechanism

This mechanism can be implemented in multiple ways. The most straightforward one would be to include the DNS [2] server and the HTTP server in the NAS. The NAS itself acts as DNS server, and recognizes the URL as pointing to the local portal. The local portal itself is also running on the NAS.

However, as it is not always desirable to run a full-blown HTTP server on the NAS, the issue arises what would be the impact of moving some functionality to separate servers.

As a first step the HTTP server can move to a separate machine. Indeed, the DNS server just needs to resolve the URL to a different IP address, and the HTTP request will be sent there. So there is no need to have the HTTP server running on the NAS.

Secondly the DNS resolution has to be adapted. The typical scenario is that the user terminal starts a PPP session (or other protocol) with the NAS, and receives its IP parameters, including the default DNS server. In other words, the default DNS server does not necessarily has to run on the NAS itself. But of course the NAS has to know which DNS server to point to.

So the requirements for the NAS to support this feature are limited to just this one: point all user terminals to a DNS server, that is associated with the PoP, and that can resolve the URL of the portal to a specific portal associated with the PoP. This requires no new DNS mechanism or other protocol, just a correct configuration of the NAS. Therefore the DNS server and the HTTP server are both drawn as separate servers (Fig. 2).

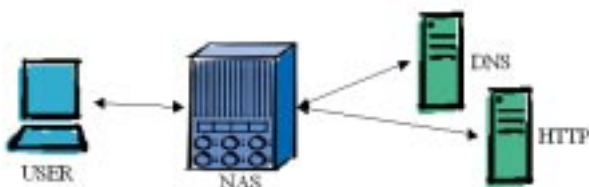


Fig. 2 PoP-dependent server selection

The mechanism requires that there is no caching of DNS information at the user terminal side, at least not in between different sessions. Indeed, if the user has a portable PC and is connecting subsequently through different NASs, each session may resolve the URL of the portal differently. So it must be assured that no old DNS information is cached in the terminal.

### C. Extensions

The whole mechanism can also be applied to other types of servers (e.g. SIP server, mail server, etc.) that are contacted via a DNS operation.

The mechanism has been described for a single URL, a single homepage pointing to the portal. Clearly, the same can be applied for multiple URLs. For example, the user may have a set of bookmarks pointing to homepages of VPN-specific portals. Also for these, separate portal-instances may be introduced for each PoP, and the DNS server can be configured such that the same universal URL (for each of these VPNs) guides the user to the PoP-specific portal.

## III. LOCATION-DEPENDENT SERVER SELECTION

### A. Concept

The PoP-dependent server selection described in the previous section is situated entirely at the level of the service provider. All users connecting to the same PoP will receive the same information. In terms of location-specific information this would typically be used to adapt the weather-forecast pages as mentioned before.

The granularity of this location-dependent server selection can be increased, in the sense that the portal could be different, depending on the exact location of the user. For example, if the user connects from some kiosk-like facility in a gas station, he would be guided to a portal of the gas station; and similar for connecting from a specific hotel, train-station, library, etc.

This cannot be achieved by solely configuring the NAS, to which the user connects, as this would imply that each hotel, gas station, etc. has its own NAS. Typically this is not the case, the NAS is owned by the AP, and each hotel (let's stick to the hotel example for the remainder of the explanation) has a number of ATM VP/VCS to this NAS (via ADSL lines terminated in an ASAM).

The NAS knows the user's location, based on the logical channel through which it connects (this applies for a BB-NAS, for a NB-NAS the same thing can be achieved via calling line id). This information can be used to provide location-dependent services to the user. The user would have a bookmark (homepage) pointing to the portal of the service provider, and a bookmark (button) ‘Local Services’ which guides the user to a portal associated with the specific hotel, train station, etc. from where he is currently connecting. In

other words, a single universal button leads to a different server depending on the location of the user.

The situation is actually quite similar to what is implemented already today in the context of mobile WAP services, where the location information, available from the intelligence in the mobile network, is used to adapt the services.

### B. Architecture & Mechanism

We start from the architecture for the PoP-dependent server selection described earlier.

Whereas for the PoP-dependent server selection the DNS server didn't need any specific parameters (it was assumed that one DNS server served only one PoP), things are different now. To achieve location-dependent server selection, the DNS server now needs to take into account the logical channel (or any other parameter that uniquely identifies the location of the user). This implies that additional information has to be exchanged between the NAS and the DNS server. This can be realized by running the DNS server on the NAS: the interface then becomes a dedicated, proprietary interface between processes inside one box. Alternatively the DNS server may still be considered as a separate module, running on a separate machine. Nevertheless it clearly needs a closer link to the NAS, therefore a second NAS interface is drawn with a dotted line (Fig. 3, full line is standard interface, dotted line is dedicated interface).

The basic mechanism is as follows. The user starts a PPP session with the NAS, and the NAS indicates the default DNS server. When the user terminal sends a request to the DNS server, the DNS server interrogates the NAS to find out the location of this user (or rather, the location of the user terminal corresponding to the source IP address of the request).

Instead of a table 'domain name – IP-address', the DNS server returns an IP address based on both the requested domain name and the current location. The IP address points to an HTTP server, typically external to the NAS and further communication to this server is transparent for the NAS.

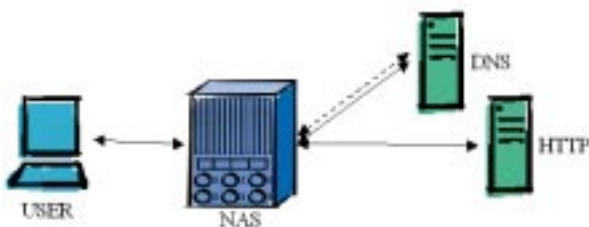


Fig. 3 Location-dependent server selection

### C. Extension

This scenario is not only applicable for HTTP servers. In some cases the hotel (as an example of a specific location) will want to use dedicated servers for certain applications. This could be the case for a PABX-type of SIP proxy server (Internet Telephony) allowing a private numbering scheme for hotel calls and providing an interface to the public telephony network. It's also possible that the hotel wants to use the server of one specific provider, so all call signaling should be routed through that SIP server. In these cases, the DNS server has to decide which server to select, based on the location of the user.

## IV. LOCATION-DEPENDENT SERVICE SELECTION

### A. Concept

The mechanism described for location-dependent server selection enables quite interesting services (e.g. hotel-dependent homepages) and can be relatively easily implemented. However, as the name indicates, the basis of the mechanism is server selection. This implies that each hotel would have a separate server. Typically this is not the case: the service provider will offer a web-hosting service, such that each hotel can offer a set of web pages on this server. In other words, multiple hotels will run their services on the same server.

What we need is not server selection, but service selection. In terms of implementation, this means one step further. As the DNS server can only return IP addresses, now also the HTTP server itself becomes involved.

A user resides in hotel A, connects to the NAS, and requests the hotel page (button on browser). The HTTP server (resolved by DNS server of NAS) checks with the NAS where this request originated (IP-address – VP/VC map of the NAS) and based on this information returns the web page specific for that location. The same user, going to hotel B, will receive another page, even when pressing the same button, and even when hotel A and hotel B share the same HTTP server.

### B. Architecture & Mechanism

Starting from the PoP or location-dependent server selection architecture, a specific server can be selected, based on the URL resolution of the DNS server. However, the DNS server can only resolve a URL into an IP-address, and cannot convey any service specific information, meaning one server can only offer one service.

To solve this problem, the server resolved by DNS has to be able to determine which response (service) to return to requests originating from certain locations (typically which web page to return to a request originating from a certain range of IP-addresses).

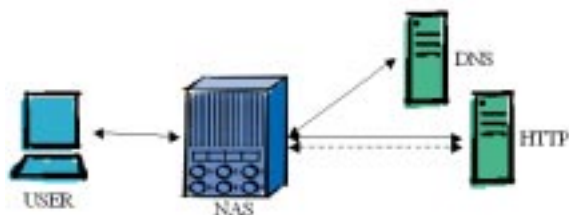


Fig. 4 Location dependent service selection

The server only knows the IP-address, so it has to have a means to link this IP-address with a location (typically VP/VC). This information is maintained at the NAS (location – IP-address map), so the server interrogating the NAS is the most obvious way to retrieve the location information. This can be realized by running the server on the NAS: the interface then becomes a dedicated, proprietary interface between processes inside one box. Alternatively the server may still be considered as a separate module, running on a separate machine. Nevertheless, it clearly needs a closer link to the NAS; therefore a dotted line is drawn from HTTP server to NAS (Fig. 4).

The basic mechanism is as follows: A user is connected to a NAS. His browser has a button set to a general URL. When the user clicks the button, the DNS server resolves the URL to the HTTP server having a logic interface with the NAS, and a page is downloaded specific for that location and NAS.

Instead of a new interface between the NAS and the DNS or HTTP server, one could also use an alternative architecture whereby the NAS updates a database with the current location information of the user, and all requests from the DNS would be sent to this database.

## V. PERSONALIZED & LOCATION-DEPENDENT SERVICE SELECTION

### A. Concept

As described in the previous sections, the NAS can provide specific information (mainly location information) allowing value added service delivery to end-users. Extending this information base is no longer the responsibility of the NAS, but should be taken care of by an external database server. This DB can receive triggers (with information) from several instances (e.g. NAS) to populate, update and synchronize its DB tables. The main tables in this relational DB will be user, network, terminal, and service profiles, possibly split over multiple DB instances.

A user connects to a specific VPN through a certain NAS. The NAS is aware of specific user information: its login name, VPN, location and IP-address. The NAS can send an update trigger with this specific information to a DB maintaining a user profile (identified by e.g. the user's login name). Several NASs can share one DB.

Suppose the user clicks the same button ('Local Services') to receive location specific information, as described in the previous sections. The URL is resolved by a DNS, as configured during bootstrapping. If the DNS is able to link the IP-address of the originating request to a certain location by inquiring the user profile DB, it can return the IP-address of an HTTP server based on very generic user information. If also the HTTP server is able to relate the IP-address of an incoming request to a user, it can color the returned web pages with user and location specific information, and adapt it to the user terminal and access network, providing a very generic service model.

### B. Architecture & Mechanism

The profile DB and servers can all be seen as external to the NAS (Fig. 5). When the user sets up a connection, profile information (user on line, IP-address, etc.) is logged to a profile database containing personalized user, terminal and service profiles, etc. Any server can use, inquire or update this information through logic (API) or network interface (e.g. using LDAP, RMI, CORBA, etc.). If the user wants to resolve a certain URL, the DNS can personalize this URL according to the user profile. Any server (e.g. HTTP server) can adjust the offered service (e.g. HTML pages) based on this personalized profile. Of course, the profile DB must be able to link an IP-address to a certain user; a link that is made when the user connects to the NAS and is assigned an IP-address. A NAS trigger (e.g. SNMP, RADIUS, etc.) forces the update of the DB.

### C. Issues

With the introduction of a user profile database, privacy issues emerge. Which personal information can be stored, which can be shared or should be filtered, should the user be able to control this process (e.g. with P3P [3]), and what about non-repudiation and authentication?

The location of the DB, and where and which information should be stored are interesting business issues. Profile information will probably be maintained in several locations (VPNs, access provider network, corporate networks, etc.) and a policy framework will have to be provided to allow information sharing.

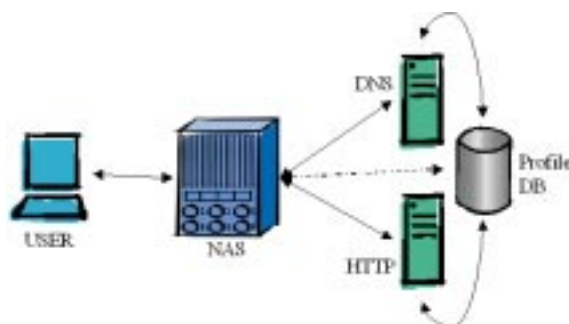


Fig. 5 Personalized and location dependent service selection.

## VI. CONCLUSION

Adding functionality in the access network and using the available user location information, value can be added to the services and a new range of local services can be deployed. Several methods have been described with increasing functionality to implement these services, varying from local server selection to full-blown localized and personalized service selection. Typical applications are local hotel information, access to local administrations and information, local businesses, etc.

## REFERENCES

- [1] J. Veizades, E. Guttman, C. Perkins, and S. Kaplan, "SLP: Service Location Protocol," IETF-RFC 2165, June 1997.
- [2] P. Mockapetris, "Domain Names - Implementation and Specification," IETF-RFC 2165, November 1987.
- [3] Platform for Privacy Preferences (P3P) Project (<http://www.w3.org/P3P/>)