HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Electrical and Communications Engineering
Networking Laboratory

Nicklas Beijar

Telephony Routing with Support for Number Portability in Interconnected Circuit and Packet Switched Networks

Thesis submitted in partial fulfillment of the requirements for the degree of Licentiate of Science in Technology

Espoo, Finland, April 5, 2004

Supervisor Professor Raimo Kantola

Reader Professor Jorma Jormakka

HELSINKI UNIVERSITY OF TECHNOLOGY ABSTRACT OF THE LICENTIATE THESIS

Author:	Nicklas Beijar

Name of the thesis:	Telephony Routing with Support for Number Portability in Interconnected Circuit and Packet Switched Networks
Date:	5.4.2004 **Number of pages:** 117
Faculty:	Department of Electrical and Communications Engineering
Professorship:	S-38 Networking Technology
Supervisor:	Professor Raimo Kantola
Reader:	Professor Jorma Jormakka

Telephone networks are currently facing two major changes that affect routing. Firstly, number portability is gaining popularity. Number portability requires a mapping between the number dialed by the subscriber and the number used for routing. Secondly, Internet Protocol (IP) telephony is gradually replacing circuit switched telephony. As both technologies must co-exist for several years, a need arises to provide seamless routing between the technologies. This involves the process of locationing the best gateway for each call between the technologies.

This licentiate thesis examines the problems of routing, gateway location and number portability in interconnected circuit and packet switched network. The problems and their current solutions are first described and evaluated. We observe that the current solutions address each of these sub-problems separately and only in a single technology, while an efficient solution requires that all problems are observed simultaneously. We therefore establish a set of requirements for information distribution and usage of routing addresses. We then propose extensions to existing methods to enable better performance and integration with other methods in a hybrid scenario. Specifically, we examine solutions based on the Domain Name System (DNS), the Telephone Routing over IP (TRIP) protocol, and databases.

Because it is doubtful that a single method would be used in all networks and in both technologies, we examine combinations of methods. Further, we find that scalability can be improved by using two specialized consecutive mappings. We group the methods into schemes according to the involved types of identifiers and select a few feasible schemes. The schemes and the application of specific methods on the schemes are evaluated. Finally, different properties of the methods are analyzed and especially scalability is considered.

Keywords: routing, number portability, gateway location, VoIP

TEKNILLINEN KORKEAKOULU LISENSIAATIN TUTKIMUKSEN TIIVISTELMÄ

Tekijä: Nicklas Beijar

Työn nimi: Puhelun reititys ja numeron siirrettävyyden tuki yhdistetyissä piiri- ja pakettikytkentäisissä verkoissa

Päivämäärä: 5.4.2004 **Sivumäärä:** 117

Osasto: Sähkö- ja tietoliikennetekniikan osasto

Professuuri: S-38 Tietoverkkotekniikka

Työn valvoja: Professori Raimo Kantola

Työn lukija: Professori Jorma Jormakka

Puhelinverkossa on parhaillaan tapahtumassa kaksi merkittävää reititykseen vaikuttavaa muutosta. Ensinnäkin numeron siirrettävyys on yleistymässä. Numeron siirrettävyys edellyttää kuvausta tilaajan valitsemasta numerosta reitityksessä käytettävään numeroon. Toiseksi IP (Internet Protocol) -protokollaan perustuva puhelinjärjestelmä on vähitellen korvaamassa piirikytkentäistä puhelinverkkoa. Koska molempien teknologioiden täytyy toimia rinnan monen vuoden ajan, syntyy tarve reitittää puheluita saumattomasti näiden teknologioiden välillä. Jokaista verkkojen välistä puhelua varten täytyy paikantaa sopiva yhdyskäytävä.

Tämä lisensiaatin työ tutkii reititykseen, yhdyskäytävän paikantamiseen ja numeron siirrettävyyteen liittyviä ongelmia yhdistetyssä piiri- ja pakettikytkentäisessä verkossa. Aluksi näitä ongelmia kuvataan ja niiden nykyisiä ratkaisuja arvioidaan. Ilmenee, että nykyiset menetelmät ratkaisevat vain yksittäisiä osaongelmia ja vain yhdessä teknologiassa, mutta tehokas ratkaisu vaatii kaikkien ongelmien samanaikaista huomioon ottamista. Tämän takia muodostetaan joukko vaatimuksia tiedon levityksestä ja reititysosoitteen käytöstä. Sitten esitetään olemassa oleviin ratkaisuihin laajennuksia, jotka mahdollistavat paremman suorituskyvyn ja integroinnin muiden menetelmien kanssa hybridi-skenaariossa. Erityisesti tutkitaan nimipalvelinjärjestelmään (DNS, Domain Name System), TRIP (Telephone Routing over IP) -protokollaan ja tietokantoihin perustuvia ratkaisumenetelmiä.

Eri menetelmien yhdistelmiä tutkitaan, koska pidetään kyseenalaisena, että yksi menetelmä tulisi käyttöön kaikissa verkoissa ja kummassakin teknologiassa. Lisäksi todetaan, että skaalautuvuutta voidaan parantaa käyttämällä kahta peräkkäistä kuvausta. Menetelmät ryhmitellään tunnisteidensa mukaisesti skeemoihin ja muutamia toteutettavissa olevia skeemoja valitaan tarkasteltaviksi. Skeemat ja niihin sovelletut menetelmät arvioidaan. Lopuksi analysoidaan menetelmien ominaisuuksia ja erityisesti skaalautuvuutta tarkastellaan.

Avainsanat: reititys, numeron siirrettävyys, yhdyskäytävän paikantaminen, VoIP

Preface

This Licentiate Thesis has been written in the Networking Laboratory of Helsinki University of Technology within the INTERO project. INTERO is funded by Nokia Networks, Nokia Research Center, Elisa Communications and the Finnish Communications and Regulatory Authority. Because of the short duration of the project (1 year), the work has continued after the conclusion of the project. This thesis continues the work started in my Master's thesis.

I would like to thank Professor Raimo Kantola for his guidance, ideas and support during writing of the thesis. I also want to thank Professor Jorma Jormakka for reading the thesis and giving valuable comments.

I like to thank the colleagues at the lab who have contributed to my thesis with knowledge, ideas and an innovative atmosphere to work in. Especially the works of M.Sc. Antti Paju and M.Sc. Tuomo Rostela have contributed to this thesis.

Last, but most importantly, I would like to thank my friends and close relatives for their support during the studies. Special thanks go to Minna for all her loving support and encouragement.

April 5th, 2004 in Espoo, Finland

Nicklas Beijar

Table of contents

Index of figures

Index of tables

Abbreviations

ABNF	Augmented Backus-Naur Form
AVCC	Advertisement Validity Checking Client
BGP-4	Border Gateway Protocol version 4
CTAD	Circuit Telephony Administrative Domain
CTRIP	Circuit Telephony Routing Information Protocol
DB	Database
DDDS	Dynamic Delegation Discovery System
DN	Directory Number
DNS	Domain Name Service
DoS	Denial of Service
ENUM	Telephone Number Mapping
Ficora	Finnish Communications Regulatory Authority
GW	Gateway
GW-LOC	Gateway Location
GSM	Global System for Mobile communications
HLR	Home Location Register
HUT	Helsinki University of Technology
HTTP/S	Secure Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
IP_{gw}	IP address of a gateway
ISDN	Integrated Services Digital Network
ITAD	Internet Telephony Administrative Domain
ITU-T	International Telecommunications Union – Telestandardization Sector
LDAP	Lightweight Directory Access Protocol

LS	Location Server
MFVA	Mapping Function Validity Area
MSC	Mobile Switching Center
MSISDN	Mobile Station ISDN
MSRN	Mobile Station Roaming Number
NA	Numbering Agent
NAPTR	Naming Authority Pointer
NAT	Network Address Translator
NDB	Numbering Database
NP	Number Portability
NPDB	Number Portability Database
NPRA	Number Portability Routing Area
ODBC	Open Database Connectivity
PLMN	Public Land Mobile Network
PNAC	Ported Number Advertising Client
PNDB	Ported Number Database
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RFC	Request For Comments
RIS	Routing Information Server
RN	Routing Number
RN_{gw}	Routing number of a gateway
RAVA	Routing Number Validity Area
SDP	Service Data Point
SCN	Switched Circuit Network
SCSP	Server Cache Synchronization Protocol
SIP	Session Initiation Protocol
SLP	Service Location Protocol
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SS	Signaling Server

SSH	Secure Shell
TAD	Telephony Administrative Domain
TCP	Transmission Control Protocol
TRIB	Telephony Routing Information Database
TRIP	Telephony Routing over IP
UDB	Update Database
URI	Universal Resource Identifier
URL	Universal Resource Locator
VLR	Visiting Location Register
XML	Extensible Markup Language

Chapter 1
Introduction

1.1 Background

A popular vision is that nearly all telecommunications networks will be based on Internet technology in the future. With technology based on IP (Internet Protocol), it is possible to integrate voice, data and other types of communications into a single network, which saves the costs of maintaining multiple special-purpose networks. IP telephony has traditionally also been used to save costs in long-distance and international telephony. Nowadays, IP telephony is mainly seen as an enabler for new services. [Varshney 2002]

So far, the most common use of IP telephony is between computers in the public Internet, and for trunking of long-distance calls through the Internet. The first approach for a large-scale adoption of IP telephony is the Internet Multimedia Subsystem (IMS) in third generation networks as specified in release 5 of the 3rd Generation Partnership Project (3GPP) [3GPP TS23.228]. It is also envisioned that IP telephony will replace circuit switched transmission in the public switched telephone network (PSTN). [Nokia 2001]

Before a network solely based on IP technology can be realized, the IP telephony network has to coexist with the switched circuit network (SCN) for a long time. During this period, calls where the endpoints use different technologies are common. This type of calls require a gateway, which translates between the circuit switched and packet switched modes of transmission. The network technologies are interconnected with numerous gateways, and the set of networks based on different technologies form a *hybrid IP-SCN network*.

In the PSTN, the destination of a call is addressed with numbers adhering to the E.164 recommendation [ITU-T E.164] of the International Telecommunications Union (ITU). There is a common understanding that E.164 numbers should be used in IP telephony networks as well, as a complement to the Uniform Resource Identifiers (URIs) that are the native addressing method. The primary reason is that E.164 numbers can be entered with all existing telephones.

Number portability allows a subscriber to change service provider, geographical location or service type without changing his telephone number. Number portability has become a mandatory service in most countries, with the motivation that it encourages competition between

service providers. Number portability additionally lowers the subscriber's cost and discomfort in moving to another location, or switching to another service type. Number portability is currently available in the fixed switched circuit network, and is being adopted in cellular networks. With the introduction of IP telephony, number portability is considered necessary also within the IP telephony network, as well as between the IP-based and circuit switched technologies. The latter type of number portability simplifies the transition to IP-based technology.

1.2 Research problem

For calls where both endpoints are in the SCN, traditional routing methods for the SCN are used. IP telephony has also matured to a stage where most of the fundamental technology, protocols and standards are available. However, in a hybrid IP-SCN network, routing is more complex. Before a call between the technologies can be established, an appropriate gateway must be located. The selection of a gateway depends on several factors, including the location of the gateway, service provider agreements, load level, supported signaling protocols and supported media formats. This problem is usually called the *gateway location problem*.

Due to number portability, the destination may reside in another network than the directory number indicates. Therefore, the directory number must be mapped into a *routing address*, which indicates the actual location of the terminal. In the SCN, the routing address is a specially formatted *routing number*. The mapping method is specific to the implementation used in the country or network. Even the routing number may be valid only in a country or network. For calls between the two technologies, the optimal selection of gateway is dependent on the current location of the terminal. Since only the directory number is available to the originating network, an inappropriately located gateway may be selected for ported numbers. If number portability between technologies is allowed, the call might, in the worst case, be routed to the opposite technology only to be routed back immediately. Thus, gateway selection and number portability are interdependent problems. An efficient solution requires comprehensive routing information to be available in both networks. Consequently, number portability in a hybrid network requires more information than what is available in a single technology.

Solutions to the gateway location problem have been developed. These have still not been implemented in practice, partly due to their heaviness. Most countries also have developed a national implementation of number portability in the SCN. In the IP network, number portability has not gained the required amount of attention. Routing of calls in hybrid networks, where the IP and SCN networks are full peers, is still a relatively new research topic. Because the problems of gateway location, routing in hybrid networks and number portability are interrelated, the solutions to these different problems must interoperate. The currently proposed solutions only solve the contributing problems in separation. The aim of this work is to consider the whole scenario.

1.3 The goals and objectives of the thesis

In this thesis, the problems of gateway location and routing in hybrid networks are studied. Special attention is given to number portability, and it is required that routing to ported numbers should be efficient, and the scalability of the solution should be assured. Since the problems are interrelated, we study complete scenarios instead of only the solutions to the contributing problems.

The goals of this thesis are to

- Evaluate the suitability of existing and proposed methods to routing in hybrid networks in scenarios with number portability.
- Improve the existing and proposed methods to provide efficient routing and number portability support, and to provide better interoperation of multiple methods.
- Develop a theoretical framework for studying mapping of identifiers and combination of multiple mapping methods.

More specifically, the results include

- An evaluation of existing and proposed methods for number portability and gateway location.
- Extensions to the existing and proposed methods for improved scalability, performance and protocol harmonization.
- Theoretical analysis of mapping methods, combination of mapping methods, the use of identifiers and the information distribution.
- Based on the theoretical analysis, proposals to complete scenarios for solving the problems of both number portability and gateway location.
- An evaluation of the suitability of different methods as parts of the given scenarios.

1.4 The scope of the thesis

In this thesis, we assume that E.164 numbers are used for identifying subscribers. The reason is that E.164 is the only addressing scheme that is compatible with both new and existing terminals. Thus, the only assumption about the terminal is that it has a numeric keypad for entering the directory number of the destination. The thesis deals with subscriber numbers only. Numbers indicating services are excluded from the scope. Special purpose numbers, such as emergency numbers, are also excluded.

The packet switched networks under consideration are based on the IP protocol with SIP [RFC 2543] or H.323 [ITU-T H.323] signaling for telephony. The focus is on the SIP protocol. The considered circuit switched networks include the PSTN, ISDN and circuit switched mobile

networks (mainly GSM). Third generation mobile networks from the 3GPP Release 5 forward can be considered as a special case of IP networks. Only the core networks of mobile networks are considered; the radio network is out of the scope of this thesis.

1.5 The structure of the thesis

The thesis consists of three parts. Chapters 2 and 3 describe the background, the problems and the requirements. The following four chapters describe existing solutions and propose extensions and improvements to these. Chapter 8 and 9 analyze and evaluate whole scenarios consisting of combinations of the specific solutions. In more detail, the structure is the following:

Chapter 2 describes the problems of routing, gateway location and number portability in a hybrid network. Further, it presents the current solutions to these problems.

Chapter 3 specifies fundamental terminology and requirements.

Chapter 4 describes how number portability is implemented using a centralized database. In particular, the solutions developed by Ficora and Paju are presented.

Chapter 5 proposes solutions for using the ENUM scheme to implement number portability in a hybrid SCN/IP network.

Chapter 6 discusses solutions how the Domain Name System (DNS) could be used for gateway location. Three different approaches are discussed.

Chapter 7 examines whether number portability can be implemented with the TRIP (Telephony Routing over IP) and CTRIP (Circuit Telephony Routing Information Protocol) protocols. Especially the scalability issue is analyzed.

Chapter 8 provides a theoretical analysis on the requirements, the distribution of information, the combination of mapping methods, and the usage of an intermediate identifier. Additionally, five feasible schemes are extracted from the possible combinations of methods, and these are examined.

Chapter 9 contains an evaluation of mapping methods. The evaluation considers aspects related to administration, scalability, performance and functionality.

Chapter 10 summarizes the solutions for routing and number portability in hybrid SCN/IP networks, and suggests ideas for further work.

Chapter 2
Routing in hybrid networks

Switched circuit networks (SCN) and IP telephony networks differ in several fundamental aspects, including media transport, signaling, network architecture and routing. We say that they are based on different technology. A hybrid SCN-IP network consists of SCN and IP networks interconnected with gateways. This chapter explains the problems of routing, gateway location and number portability in a hybrid network. First, switched circuit networks are described from the perspective of routing and addressing. Then, IP telephony networks are presented, with the main focus on network architecture, signaling, terminal location and integration with switched circuit networks. Description of number portability is included for both network technologies. Further, the problem of gateway location is described, and two solutions for gateway location are briefly presented.

2.1 Switched circuit networks

In the switched circuit network (SCN), voice is transmitted as a stream of bits with constant bandwidth and constant delay. With the term SCN we refer to all switched circuit networks, including the public switched telephone network (PSTN), the integrated services digital network (ISDN), and the public land mobile network (PLMN).

2.1.1 Addressing and routing

In PSTN and ISDN, subscribers are identified with numbers conforming to the E.164 recommendation [ITU-T E.164] of ITU-T. An E.164 number has a hierarchical structure consisting of a country code, a trunk area code and a subscriber number. The first digits of the subscriber number traditionally identify the exchange and the last identify the subscriber within the exchange. For national calls, the country code can be omitted, and for calls within the trunk area, the trunk area code can be omitted. To establish a call, a subscriber dials the destination's E.164 number, which is also called the *directory number* since it is the number found in telephone directories. We use this term to separate them from *routing numbers*, which are numbers that are used internally in the network for routing to destinations that reside in a different topological location than the directory number indicates.

Numbers are allocated to operators in blocks. Every number block relates to a particular exchange, and therefore routing only needs to observe the first digits of the number. The number indicates the route to the destination topologically and geographically, i.e. each consecutive group of digits limits the location to a smaller area. An exchange only analyzes the number of digits required to correctly locate the route to the following exchange. Although the mapping between the digits and the route can be centrally generated and transferred to the exchange, it is generally *static* in the sense that it does not dynamically change according to network conditions.

In practice, also the PLMN uses E.164 numbering, although the trunk code is replaced by an operator identifier. A GSM call is routed with the directory number (called MSISDN) to the Gateway Mobile Switching Center (G-MSC) of the destination network. The Gateway MSC interrogates the Home Location Register (HLR) to obtain the Mobile Station Roaming Number (MSRN), which is a routing number temporarily allocated to the subscriber. The HLR has earlier obtained the MSRN with a location update, or alternatively it obtains it before the call by querying the current Visiting Location Register (VLR). The call is then set up using the MSRN. [Rahnema 1993]

In the SCN, operators are divided into local operators, long-distance carriers and international carriers. The caller can choose the long-distance and international carrier using a special *carrier selection code*. For example, in Finland the prefix 101 selects Sonera's network while the prefix 109 selects Kaukoverkko Ysi's network. [Ficora 2001]

2.1.2 Number portability

Number portability (NP) allows subscribers to switch services, service providers or locations without changing their subscriber number. These types of number portability are called *service portability*, *service provider portability* and *location portability*, respectively [Lin 1999]. A number subjected to number portability is called a *moved number* or a *ported number*. The ported number belongs to a number block, which was assigned to the *donor network*. The ported number is currently served by the *current serving network*. If the number has been ported several times, the previous network is the *old serving network*. [RFC 3482]

Number portability is possible within a set of networks constituting the *number portability routing area* (NPRA). In Finland, number portability in the fixed telephony network is possible within an area code. Country-wide number portability requires changing the number once to a special prefix reserved for national number portability. Later, all numbers may become portable between area codes. There are no plans for international number portability.

Most countries provide number portability in the PSTN [Lin 2003]. Number portability has also been available in the mobile networks in several countries (in Finland since June 2003 [Ficora 2003]).

The implementations differ in different countries. According to [RFC 3482], the implementations can be grouped into four main types:

- *All Call Query* (ACQ): The originating network queries a centrally administered number portability database to obtain a routing number, which is used to route the call to the serving network. Usually the originating network has a local copy of the database.

- *Query on Release* (QoR): The call is routed to the donor network, which detects that the destination has been ported. The call is released with an indication that the destination is ported. The originating network then queries a centrally administered number portability database, obtains a routing number and routes the call with the routing number to the current serving network.

- *Call Dropback*: The call is routed to the donor network, which detects that the destination has been ported. It queries an internal database to obtain the current routing number. The donor network performs a call release containing the routing number. The originating network routes the call using the obtained routing number.

- *Onward Routing*: The call is routed to the donor network, which detects that the number has been ported. It queries an internal database to obtain the current routing number. The donor network routes the call onward using the obtained routing number. Thus, the call is routed through the donor network, which wastes resources.

All implementations use a separate *routing number* for routing to the ported number. The routing number is obtained from a database, which may be centrally administered by a third party, or internal to the donor network. The database maps a directory number to the corresponding routing number. The routing number contains an identifier either for the current serving network or for the current serving switch. In the former case, a second query to an internal database is required to locate the current serving switch.

In Finland a variant of the All Call Query scheme is used. A database, called *Master database*, is maintained by a third party. The routing number has the following format:

`1D` + *operator-id* + *service-id* + *b-directory-number*

The first two digits ("`1D`") are used to separate routing numbers from other types of numbers. The *operator-id* is a one or two-digit number that uniquely identifies the current serving operator. Thus, a network specific method of locating the current serving switch is required. The *service-id* indicates the type of service implemented with the routing number, and has the value 01 for number portability. [THK 1996, RFC 3482]

2.2 IP telephony

In IP telephony, the media streams (e.g. voice or video) are sent in Internet Protocol (IP) packets instead of as a constant stream of bits. The media streams are sent directly between the endpoints, although signaling may take a longer route. In case of a call between the SCN and the IP network, a *media gateway* converts the stream between circuit-switched transmission and packet

transmission. The address and port number where the media streams are to be sent are signaled with a signaling protocol, which sets up and controls the call. [Varshney 2002]

2.2.1 Signaling and network elements

Currently there are two major signaling protocols for IP telephony: H.323 [ITU-T H.323] developed by ITU-T and the Session Initiation Protocol (SIP) [RFC 2543] developed by IETF. During recent years, SIP has gained popularity, partly because of its simplicity and extensibility. SIP is also the signaling protocol chosen by the 3rd Generation Partnership Project (3GPP) [3GPP] for use in third generation mobile networks starting from Release 5 [3GPP TS23.228]. Because of the high probability that SIP will dominate IP telephony, we only discuss SIP in this thesis. However, the major part is also applicable to H.323.

SIP uses the term *user agent* for the terminal. The user agent consists of the *user agent client*, which initiates calls, and the *user agent server*, which receives calls. The call is set up with the INVITE method. Responses to methods are identified by result codes. The simplest setup case, with only two terminals, is shown in Figure 2.1.

Figure 2.1. Simple call setup with SIP

Although calls can be established directly between terminals using SIP, most calls are established using one or several *signaling servers*. Signaling servers are either *proxy servers*, which relay signaling messages to the destination or to another signaling server, or *redirect servers*, which redirect signaling to another address without participating in further signaling. Moreover, proxy servers can be either stateless or stateful.

In carrier class networks, each operator has at least one signaling server. The signaling server locates the endpoints and performs charging functions. Therefore, all calls must pass through the signaling server. To prevent outgoing calls from bypassing charging, the signaling server may control firewalls and the provision of quality of service (QoS). The practice of using signaling servers in transit networks is still unclear since charging, QoS provision and access control in transit networks may also be controlled with dedicated mechanisms based on inter-operator agreements.

There are multiple possible configurations for using signaling servers. For further discussion, we define some terms based on a general configuration. When a terminal sets up a call, the INVITE message of SIP is sent to the *local signaling server*, which forwards the message to the next signaling server or to the destination directly. The address of this local signaling server may be

configured into the client or obtained with a service discovery protocol. If no local signaling server is used, the functions including address translation are performed by the terminal itself. The last signaling server before the call leaves the originating domain is the *outbound signaling server*, which in many cases is the same entity as the local signaling server. The signaling may pass through *intermediate signaling servers*, for example in the long-distance or international networks. In the destination network, the *inbound signaling server* receives the INVITE message, locates the destination terminal and forwards the INVITE.

In order to terminate calls, the inbound signaling server must know the current addresses of the terminals in its domain. A terminal informs about its current address by sending a REGISTER message to the *registrar* in the domain. This happens when the terminal starts or when its address changes. The registrar is an entity that receives register messages and updates the current location to the *location server*, which stores the address mappings of the terminals in the network of a service provider. A signaling server can query the location server to obtain the current location of a terminal, for example upon receiving an INVITE message. Although the registrar, location server and signaling server are separate logical entities, they may be implemented as a single device. The protocols used between them are not specified. Figure 2.2 illustrates the relation between different elements.

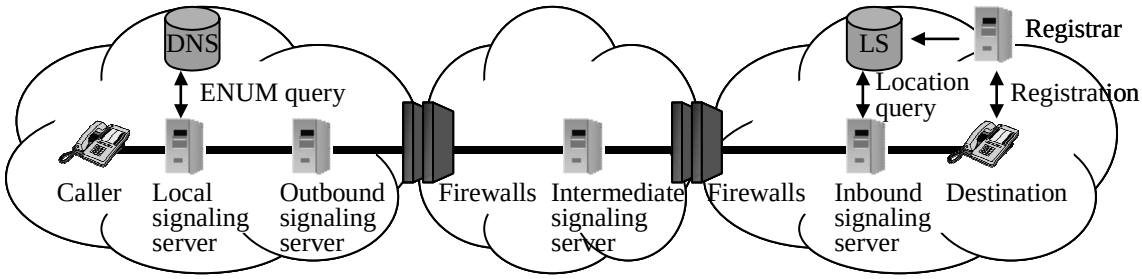

Figure 2.2: Usage of network elements in a call setup situation.

For calls between the SCN and IP network, a *signaling gateway* processes and converts signaling messages between the different signaling protocols. The signaling gateway controls the media gateway, which converts between circuit-switched and packet-switched transfer modes and between codecs. From the viewpoint of SIP, the signaling gateway appears like a normal terminal endpoint, terminating the IP-part of the call.

2.2.2 Terminal location

Calls to IP telephony terminals can be established using the IP address or host name of the called terminal as the destination address. However, to provide simple mobility and to provide services when the terminal is unavailable (e.g. switched off), the address of the inbound signaling server is usually used as the destination point address instead. The INVITE message is then sent to the inbound signaling server with an address in the format “sip:user@sip.provider.com”, and the server forwards the message to the address, to which the user has registered. This type of Unified Resource Identifiers (URI) [RFC 2396], called SIP URIs, consists of two parts: the user and the

domain name. The domain name is mapped to the IP address of the inbound signaling server using the Domain Name System [RFC 1034, RFC 1035], and the signaling server uses the user part as a key to retrieve the current location of the user. Other types of URIs are commonly used to identify email recipients and web pages.

The above solution works well while the calls are set up within the IP network. On the other hand, a terminal in the SCN has only a numeric keypad, which allows only numeric addresses to be used. Furthermore, only numeric addresses can be transported by SCN signaling and handled by the exchanges. Therefore, IP terminals can be assigned E.164 numbers as well. The number in E.164 format must be translated into a host name or IP address before the call can be set up. For this purpose, the ENUM working group of IETF has specified how the DNS is utilized to map E.164 numbers into URIs [RFC 2916]. This scheme is called ENUM, after the working group.

Before the DNS query of ENUM, the E.164 number is translated into a domain name using the following method [RFC 2916]. The input is the complete E.164 number including the country code. All non-digit characters are removed. Dots (".") are inserted between the digits. The order of the resulting string is reversed. The string ".e164.arpa"[1] is appended to the end. For example, the number "+3585415303" is converted to the domain name "3.0.3.5.1.4.5.8.5.3.e164.arpa". The DNS query returns the NAPTR (Naming Authority Pointer) records corresponding to the domain name. The NAPTR records contain the URIs, which represent different ways of contacting a host. If several URIs are returned, the URIs are examined in the order determined by the order and preference fields. For an IP telephony terminal, a valid URI is a SIP URI, but the query might also return for example an e-mail address.

Different levels of the DNS tree are utilized to give ENUM a hierarchical structure. The highest level of ENUM is named Tier 0, the following level is Tier 1, and the following is Tier 2. In some cases, a Tier 3 is used. In practice, Tier 0 includes the ENUM root servers representing the ".e164.arpa." part. Tier 1 is maintained by the national regulators. Tier 2 contains individual phone numbers. Direct inward dialing numbers can be maintained at Tier 2 or alternatively at Tier 3, if the subscriber maintains a separate ENUM server.

Nevertheless, the above structure can be implemented in several ways. In some implementations, the regulator (or an outsourced third party) maintains all the numbers in a country or in a part of a country as individual entries. Let us call this the *regulator-maintained model* for reference. In its simplest variant, all NAPTR records reside at Tier 1. Another variant uses two tiers, where the single ENUM directory is split into multiple regional directories implemented at Tier 2. Competition for the name service is, however, difficult using this model. In the *operator-maintained model*, the operator that originally was assigned the number block maintains the ENUM entries for the numbers in the block. In this model, the operator is responsible for its

[1] At the time of writing, the originally proposed top-level domain ".arpa" is being criticized by some countries. The final implementation may therefore use another top-level domain.

number blocks and maintains (or outsources the maintenance of) an ENUM server. This model has problems with number portability, as will be discussed later. [Rostela 2002]

A different approach is taken in the *U.S model*, where the individual numbers reside at both Tier 1 and 2. This allows competition at Tier 2, so that the customer can choose both the company providing the name service and e.g. the SIP service. [ENUM-Forum 2003]

2.2.3 Using ENUM in hybrid networks

ENUM provides the necessary tool for translating E.164 numbers into URIs. When both the caller and the destination reside in the IP network, the local signaling server or the terminal itself performs a DNS query to obtain the URI of the destination. The call is then established using SIP signaling to the obtained URI. When the call arrives from the SCN, the gateway performs the DNS query to obtain the address of the IP terminal, and establishes the IP telephony call-leg.

On the other hand, when the destination is a SCN number dialed by an IP terminal, the ENUM query does not return an IP address. In this case, it is impossible to say whether the destination is an SCN terminal or if the number does not exist. The current practise is to establish the call to a gateway, which tries to set up the call on the SCN side. If the number does not exist in the SCN, resources (e.g. gateway capacity) are wasted. This may also be a vulnerability to denial of service (DoS) attacks.

2.2.4 Relationship between IP address and URI

For the purpose of further discussion, it is important to clarify how the IP address and the URI relate to each other. In IP telephony, the URI indicates the destination of the call. It consists of two parts: the user part and the domain. The domain may be the host name of a SIP server, but it may also be a domain name, whereas DNS is used to retrieve the address of the SIP server responsible for the domain. In either case, the domain part indicates the SIP server, whose IP address is obtained with a DNS lookup.

Only in rare cases, the IP address uniquely indicates a specific terminal. This requires the terminal to have a globally valid static IP address, i.e. the terminal is not behind a network address translator (NAT). In the common case, the IP address only indicates the server, which locates the specific terminal using the user part. Consequently, the complete URI is required to uniquely identify the specific terminal.

In this work, we will often map some type of identifier to an IP address in order to locate a terminal. According to the above, the IP address is not enough to locate the terminal. However, in this type of usage, the E.164 number of the terminal is known. The combination of E.164 number and IP address uniquely specifies a terminal, since a URI in the format “number@ip-address” can be generated. Using local address translation, the server is then able to terminate the call to the correct terminal among the ones that it maintains.

2.2.5 Number portability

IP telephony networks are a rather new concept, and the focus has been on specifying the basic functionality. Number portability in IP telephony networks has gained minor attention. As IP telephony is adopted in public telephony networks, also number portability must be provided. IP telephony will most likely be regarded as an implementation technology from the legislation and service perspective, and the same regulations will be applied to IP telephony networks as well.

The mapping from an E.164 number to the address of the terminal is composed of two consecutive mappings: first ENUM maps the E.164 number to a URI, and then DNS maps the URI to an IP address. To implement number portability, either one of the mappings could be modified. Additionally, the IP address could be seen as a home address that is mapped to a care-of address using Mobile IP [RFC 3344]. However, Mobile IP involves inefficient triangle routing and duplicate address allocation, and is more suited for temporary mobility than for long-term portability. If number portability is implemented in the DNS mapping, the domain name will still belong to the donor operator. Therefore, the consensus is that number portability should be implemented in the ENUM mapping. The routing address is then the URI or the IP address. Number portability implemented with ENUM will be discussed in Chapter 5.

2.3 Gateway location

2.3.1 Problem description

When a call is set up from the IP network to a destination in the SCN (we call this an *IP→SCN call*), a gateway is required. An IP telephony service provider has usually several gateways, which are connected to different regions or peer operators. In a global perspective, every gateway connected to the IP network may be a potential gateway for the call, even though it is owned and operated by another service provider. Service providers can have contracts for using each other's gateways against some annual or traffic-based fee. Consequently, the number of potential gateways may be high, and the selection of the most suitable gateway is nontrivial. This problem, which usually is called the *gateway location problem* [RFC 2871], is illustrated in Figure 2.3. The problem is further complicated by the existence of different signaling protocols, codecs, encryption mechanisms and capabilities. These problems were pointed out in [Rosenberg 1998].

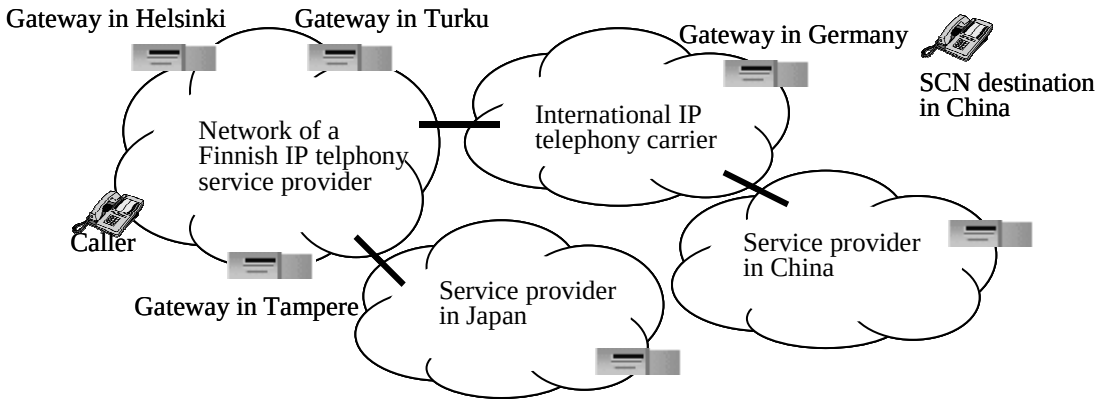

Figure 2.3: Illustration of the gateway location problem

We divide the gateway location problem into two sub-problems: *gateway discovery* and *gateway selection*. The problem in gateway discovery is to obtain the addresses of the available gateways that are capable of completing the call to the given destination and that support the used signaling protocol and codecs. The problem of gateway selection is to select one of these gateways, while respecting the policies of the originating operator and the intermediate operators, observing the location of the gateway and satisfying the quality-of-service requirements of the call. Generally, the discovery obtains all the serviceable gateways, and the selection chooses the preferred one from these.

The gateway discovery problem could be solved with a global directory of all the gateways in the world. In order to scale, the directory must be distributed. However, all operators are not willing to publish information about the gateways and their properties. The gateway discovery method should therefore preferably only show the gateways that a given network may use. This, on the other hand, implies gateway selection, since a reduced set of gateways are selected. Thus, the discovery and selection problems are related.

The gateway selection problem is driven by the policies of the operators. The originating operator has preferences concerning the gateway selection since the caller is a customer of the originating network, and the selection controls the cost and the perceived quality of service. Let us call a method where only the originating operator influences on the selection an *originator-determined policy model*. Further, the operator owning the gateway wants to control who can use it. We call such a policy a *destination-determined policy model*. The operators along the path of the call may have policies as well – especially the operators on both sides of the gateway. In general, it is difficult to determine who should perform the selection. Preferably, all operators along the path should be able to influence on the selection – as in the *path-determined policy model*. However, in the simplest case, the policies are concentrated to the originating network. The policies of the other networks are then incorporated into the contracts made by the originating network.

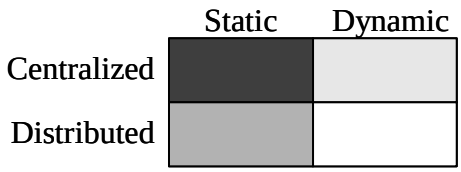

Figure 2.4. Classification of gateway location solutions

A classification of solutions for gateway location is presented in [Olsson 2002]. The problem space is divided into four parts by two criteria, as depicted in Figure 2.4. In a static solution, the information about gateways is not immediately updated when the gateway topology changes. Instead, it is manually and/or periodically updated. In a dynamic solution, the information is updated immediately. A solution is centralized or distributed, depending on whether a centrally administered resource or a peer-to-peer model is used.

2.3.2 TRIP

To solve the gateway location problem, the IPTEL working group of IETF has developed the *Telephony Routing over IP* (TRIP) protocol [RFC 2871, RFC 3219]. TRIP is an application layer routing protocol modeled after the Border Gateway Protocol version 4 (BGP-4) [RFC 1771]. According to the classification in [Olsson 2002], TRIP is a distributed dynamic solution.

In the TRIP framework, the resources of a single administrative authority form an Internet Telephony Administrative Domain (ITAD). In practice, an ITAD can be the network of an operator or service provider. The TRIP protocol distributes routing information between the location servers, which learn about routes to SCN destinations. There is at least one location server in each ITAD and advertisements are originated in ITADs with gateways. The protocol distributes these advertisements to location servers in neighboring ITADs. The advertisements are prioritized and filtered according to the policies of the receiving ITAD, which then forwards possibly modified and aggregated versions of the advertisements to its neighboring ITADs. The process repeats until all ITADs have a route to all available SCN destinations. The routes pass in the opposite direction through zero or more signaling servers. The final hop of a route is the gateway.

TRIP performs both gateway discovery and gateway selection. Gateways are discovered since each ITAD advertises its gateways, and the advertisements are propagated to the entire network. Separate routes are generated for each signaling protocol (called application protocol in the specification). Codecs are not considered. The selection is carried out by all ITADs along the route, thus TRIP implements the path-determined policy model. When an ITAD receives several routes to a destination, it chooses one of them according to its local policies.

2.3.3 CTRIP

For a call from the SCN to the IP network (an *SCN→IP call*), a gateway must be located as well. Currently, static routing is used. This works well as long as there are few gateways and few calls

requiring gateways. As IP telephony networks are taken into full-scale commercial use, and the number of IP telephony networks grows, this does not scale. A more scalable solution requires automatic gateway location. One solution to the problem of SCN→IP gateway location is the *Circuit Telephony Routing Information Protocol* (CTRIP) [Kantola 2001, Beijar 2002]. CTRIP is a version of TRIP that is adapted to circuit-switched networks. It has the same operational model and properties, but some attributes are modified to support routing in the SCN. TRIP and CTRIP will be presented more thoroughly in Chapter 7.

2.4 Network scenario

A hybrid SCN/IP network as discussed in this work consists of interconnected switched circuit networks and IP telephony networks. These networks are implemented with a given *network technology*, which can be SCN or IP. Networks are owned and operated by different operators or service providers. Let us call a network belonging to one organization a *domain*. Following the terminology of [RFC 2871], an IP telephony network controlled by a single administrative authority is called an *Internet Telephony Administrative Domain* (ITAD). A corresponding term for a SCN network controlled by a single administrative authority introduced in [Beijar 2002] is *Circuit Telephony Administrative Domain* (CTAD). To identify networks, ITADs and CTADs are assigned globally unique 4-octet unsigned integer identifiers [RFC 3219, Beijar 2002]. The ITAD and CTAD identifiers can be combined into a common *Telephony Administrative Domain* (TAD) identifier, assuming that they are non-overlapping.

Chapter 3
Definitions and requirements

For the purpose of further description and analysis, this chapter establishes a set of notations, definitions and requirements. We first present a notation for describing mappings in a short way. We then form a set of requirements for providing correct routing to ported destinations in the case where several mapping methods exist and routing numbers are valid in a limited subset of the networks. We examine how information about terminals on one technology can be used in the other technology.

3.1 Mappings

As we have seen, several addresses and interfaces are used for different purposes. In this work, we assume that E.164 directory numbers are used for identifying terminals in both SCN and IP networks. Calls to IP telephony destinations are routed with a URI, which indirectly specifies an IP address. Calls to SCN terminals are established with routing numbers. From the viewpoint of our observation, routing numbers are used even if the terminal has not been subject to number portability; if a number is not ported, the directory number and routing number are identical but still logically separate.

Both number portability and gateway location involve mappings from one identifier into another. Generally, number portability is the modification of the mapping from a directory number to a routing address, and gateway location is performed by mapping the directory number to the routing address of a gateway.

We define a *mapping* as a function

$$f_T : A \rightarrow B$$

that maps one type of identifier into another type. The input, which is an identifier of type *A*, is translated to another identifier of type *B*. The function f_T performs a mapping using the method *T*. Mapping methods can be simple table mappings (the value in A is used as a key to retrieve B from a table or database) or dynamic methods (e.g. calculating value B from the parameter A). For example, ENUM maps a directory number to a URI, and it can thus be described as a function

$$f_{ENUM} : DN \rightarrow URI$$

To be able to describe mappings more compactly, we introduce the following notation for identifying a mapping function:

$$A \xrightarrow{T} B$$

In this notation, the interpretation of *T*, *A*, and *B* is identical to the above function: the *input* of the mapping is an identifier of type *A*, and the *output* is an identifier of type *B*; the *mapping method* is marked with *T*.

Mappings can be chained:

$$A \xrightarrow{T1} B \xrightarrow{T2} C$$

In this case, the identifier of type *A* is translated to an identifier of type *B*, which is translated to an identifier of type *C*. A special case of chaining is a recursive mapping, which we denote with an asterisk:

$$A \xrightarrow{*T} B$$

Here the method *T* may generate another identifier of type *A*. When this occurs, the output is given as input to the same method *T*. The process is repeated until an identifier of type *B* is obtained.

The mapping methods discussed hitherto include *DNS*, *TRIP*, *CTRIP* and number portability database (*DB*) queries. Identifier types are directory numbers (*DN*), routing numbers (*RN*), IP-addresses (*IP*), URIs (*URI*) and TAD identifiers (*TAD*).

3.2 Validity areas

Some routing addresses are only valid within a given domain. For example, the Finnish routing numbers starting with "1D" are only recognized by exchanges in Finland. Other types of routing numbers, such as the Mobile Station Roaming Number (MSRN) used in GSM, are global. Routing addresses can only be used in the network that recognizes them and have a corresponding route. Therefore, we define the *routing address validity area* (*RAVA*) as the set of SCN networks where a given routing address can be used. The mapping from a directory number into a routing address must not be performed outside the validity area of the routing address. Once the mapping has been performed, the call setup must not leave the validity area.

Furthermore, different methods can be used to perform the mapping. Not every network supports every method, and the mapping information itself may not be accessible in all networks. Therefore, we define the *mapping function validity area* (*MFVA*) of a specific mapping function as the set of networks supporting the mapping function. With mapping function we mean the union of all mapping methods that share the same mapping information. Different mapping

methods can be used in different networks, but if two mapping methods share the same mapping, they can be considered as a single mapping function. For instance, if there is a continuous exchange of mapping information between TRIP and CTRIP through a numbering gateway, these methods are considered as a single mapping function.

3.2.1 Rules for validity areas

Given a directory number *DN*, and the routing address $RA = RA_{DN}$ corresponding to the directory number, let $D = D_{DN}$ denote the donor network and $S = S_{DN}$ the current serving network of *DN*. Let *M* denote the set of networks belonging to the MFVA, each of which has a mapping $DN \rightarrow RA$. Let *R* denote the set of networks belonging to the RAVA of *DN*. Further, for a given routing address *RA*, let $P(i, j) = P_{RA}(i, j)$ denote the ordered sequence of networks on the route from network *i* to network *j*.

In order to guarantee that the mapping is performed at some stage during the call setup, and in order to guarantee the validity of the routing address, we provide a set of rules:

Rule 1. $D \in M$

The mapping function validity area must contain the donor network. This rule guarantees that the directory number will be translated to a routing address at least in one network on the call path.

Rule 2. $S \in R$

$D \in R$

The routing address validity area must contain the current serving network and the donor network.

Rule 3. $\forall\, r \in R : \exists\, P(r, S)$

$\forall\, P(r, S) : P(r, S) = \{r, n_1, n_2, \ldots, n_N, S\} : n_1, n_2, \ldots, n_N \in R$

All networks in the routing address area must have a route to the current serving network. All routes to the current serving network contain only networks that are within the routing address validity area. This rule guarantees that a call established with a routing address will not leave the routing address validity area, and that every network routing the call has a route for the routing address.

Rule 4. The mapping from directory number to routing address must be performed in the first network *N*, for which both $N \in R$ and $N \in M$ are true.

In the following, we show that these rules guarantee that the directory number will eventually be translated into a routing address, and that the routing address will route the call to the current serving network. We assume that in every network, the route for the directory number leads to

the donor network. This assumption is justifiable since the directory number is included in the prefix assigned to the donor network.

Theorem 1: A call established with the directory number will reach a point where the subscriber number is translated into a routing address.

Proof: A call setup proceeds towards the donor network of the directory number. Since this network supports the mapping function (rule 1) and is within the validity area of the corresponding routing address (rule 2), the mapping will be performed at latest in the donor network (rule 4). Since another network on the path toward the destination network may support the mapping function and belong to the routing address validity area, the mapping may also take place before the destination network. □

Theorem 2: After the routing address has been taken into use, the call will be correctly routed to the receiver.

Proof: The translation to the routing address is performed in a network that is within the routing address validity area (rule 4). All networks within the routing address validity area are able to route to the current serving network (rule 3). Also the current serving network is within the routing address validity area (rule 2). The call will not leave the routing address validity area (rule 3). □

Formally, the path of a call can be described with the ordered sequence $\{n_1, n_2, \ldots, n_N\}$ of networks n_i divided into a first ($1 \le i \le L-1$) and second ($L \le i \le N$) segment, for which the following holds true:

$$
\begin{aligned}
&(\forall\, n_i : 1 \le i \le L - 1 : (n_i \notin R) \vee (n_i \notin M)) \wedge && \textit{(first segment)} \\
&(\forall\, n_i : L \le i \le N : n_i \in R) \wedge && \textit{(second segment)} \\
&(n_L \in M) && \textit{(mapping network)}
\end{aligned}
\qquad (3.1)
$$

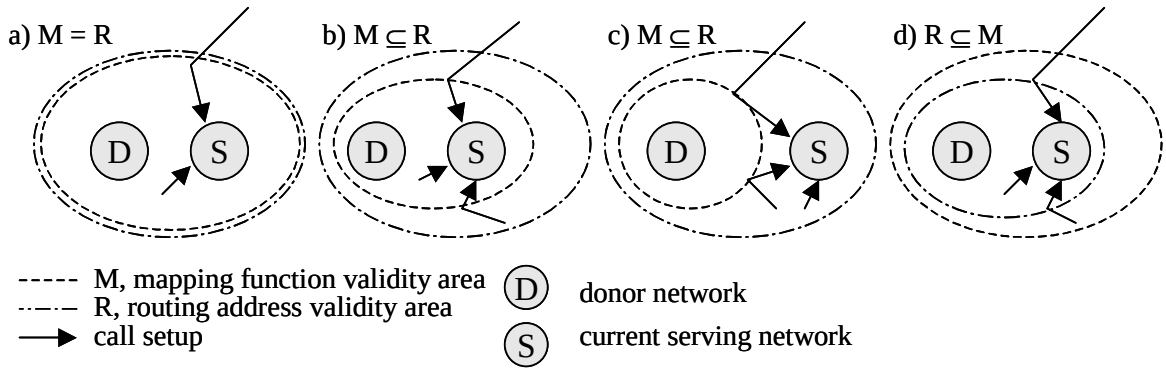

Figure 3.1: Possible configurations of validity areas and the call setup routes

From the rules we can make the following observations, which are illustrated in Figure 3.1.

- *Possible combinations:* The donor network must be within both the MFVA and the RAVA. The current serving network must be within the RAVA but does not need to be within the MFVA. Thus, the number is allowed to move within the routing address validity area. The mapping function validity area can be larger or smaller than the routing address validity area.

- *Connectivity:* The RAVA must be connected, since there must exist a route between each pair of networks within the RAVA. The MFVA is not required to be connected.

- *Efficiency:* The call is routed towards the donor network until it reaches the first network belonging to both the MFVA and the RAVA, where the mapping is performed. For efficiency, these areas should be as large as possible. The efficiency is limited by the smaller of these areas.

- *Control*: In a network *N*, which is in the MFVA but not in the RAVA ($N \notin R \wedge N \in M$), control is needed to prevent translation, since the routing address is not valid in this network. An implementation might perform a query that returns a routing address, which should not be used. The response to the query may include an indication that the routing address is not valid here, or the client may deduce this from the routing address format. This query is redundant and may cause additional overhead if it is repeated in several networks outside the RAVA. To prevent this overhead, a practical implementation may deduce from the directory number (e.g. the country code) that the current network is not within the RAVA for this number. Thus, this overhead is a rather theoretical problem.

3.2.2 Validity areas and hybrid networks

In the SCN, the routing address is a routing number, and in the IP telephony network, the routing address is a URI. It is obvious, that a URI is an invalid routing address in the SCN and vice versa. Consequently, a routing address validity area cannot contain both SCN and IP networks.

In a hybrid network, it is desirable to allow number portability between the technologies. In that case, we can talk about *donor technology* and *current serving technology* in addition to donor and current serving networks. The hybrid scenario necessitates some special considerations when the above rules are applied.

According to Rule 2, both the donor and current serving networks must be in the same routing address validity area. This implies that number portability between the technologies cannot be implemented with a single routing address. Instead, a separate routing address must be used in each technology, and each routing address has a related validity area. The gateway is the entity that connects the two technologies. It terminates a call in one technology and establishes a call in the opposite technology. Thus, from the viewpoint of the donor technology, the subscriber has been ported to the other technology, represented by a gateway. From the current serving technology, the subscriber has ported from the other technology, represented by a gateway. The

routing address in the donor technology represents a route from the donor network to the gateway. Similarly, the routing address in the current serving technology represents a route from the gateway to the current serving network.

The rules in section 3.2.1 are valid also in a hybrid IP-SCN scenario, but must be applied to each technology separately. In the donor technology, the donor network is still the network to which the number originally was assigned, but the current serving network is the opposite technology, in practice represented by a set of gateways. In the current serving technology, the donor network is represented by a set of gateways, while the current serving network is still the network where the subscriber currently resides.

Let us indicate the properties of the donor technology with the index "d" and the current serving technology with the index "s". Then a call uses a gateway, which appears as S_d in the donor technology and as D_s in the current serving technology, as illustrated in Figure 3.2.

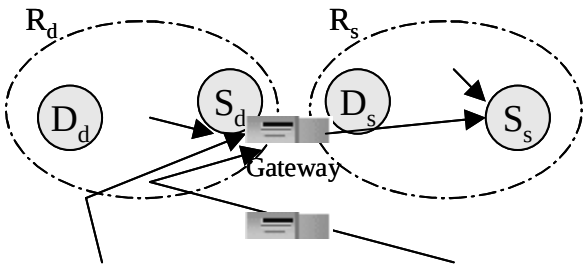

Figure 3.2: Donor and current serving networks in a hybrid scenario

The abstraction gives the requirements for valid gateway locations:

1. According to Rule 2, $S_d \in R_d$ and $D_d \in R_d$, thus the gateway (S_d) must be within the RAVA in the donor technology.
2. According to the same rule, $S_s \in R_s$ and $D_s \in R_s$, thus the gateway (D_s) must be within the RAVA in the current serving technology.
3. Furthermore, Rule 1 requires the gateway to be within the MFVA in the current serving technology, i.e. the network where the gateway is must be able to generate the routing address in the current serving technology.

We can also draw the conclusion, that also in this case the RAVA and MFVA should be as large as possible. A call originating from the current serving technology but outside the RAVA will be routed toward the donor technology through a gateway, since there is no valid routing address in the originating network. The call will be directed back trough a gateway in the first network that is in both the RAVA and the MFVA. In practical scenarios this is unlikely to happen, since the current serving technology is probably IP telephony, where the routing addresses (URIs) are globally valid. However, for a number moving from an IP telephony network to a SCN network, it may become a problem if the routing number in the SCN is not globally valid.

In practical scenarios, the borders of RAVA and MFVA are assumed to follow country or regional borders. Such an area, where the MFVA and RAVA of a technology coincide, and numbers are allowed to move between the RAVA of different technologies, constitutes a number portability routing area (NPRA). Numbers are allowed to move within the NPRA, and for calls between the technologies a gateway within the NPRA is selected.

3.3 Information distribution models

In a hybrid network, a different mapping is required for each technology. Since different types of routing addresses are used, these mappings are valid in a limited set of networks, which we identify as the validity areas $MFVA_{IP}$ and $MFVA_{SCN}$, respectively. Although a given routing address RA_{t1} can only be *used* in technology *t1*, it may be possible to *access* the mapping $DN \rightarrow RA_{t1}$ in another technology *t2*. It may be possible to use the information of the opposite technology to make routing more efficient. In particular, it is sometimes possible to transport RA_{t1} in the call setup for a call from technology *t1* to technology *t2*.

We can distinguish between four information models, depending on how the mapping information is shared between the SCN and IP networks. In the first model, depicted in Figure 3.3a, the information about SCN terminals stays in the SCN, and the information about IP terminals stays in the IP network. The mapping function validity areas are separate. Let us call it the *separate information model*. In the second model, depicted in Figure 3.3b, the information about IP terminals is available in both technologies, while the information about SCN terminals stays in the SCN. Thus, the $MFVA_{SCN}$ includes SCN networks but the $MFVA_{IP}$ includes both SCN and IP networks. We call it the *shared IP-information model*. The model in Figure 3.3c, the *shared SCN-information model*, is the opposite: the information about SCN terminals is available in both technologies, while the information about IP terminals stays in the IP network. Finally, the model in Figure 3.3d, distributes information about all terminals in both technologies. Let us call it the *shared information model*.

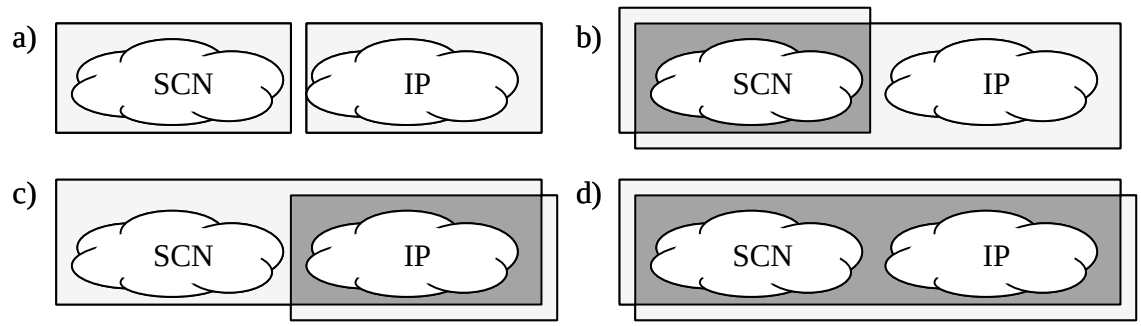

Figure 3.3: Information models

In the separate information model, the information in the SCN and IP networks differs. For destinations in the SCN, the databases in the SCN contain a mapping DN→RN and the databases in the IP network contain a mapping $DN \rightarrow IP_{gw}$. For destinations in the IP network, the databases in the SCN contain a mapping $DN \rightarrow RN_{gw}$ and the databases in the IP network contain a mapping

DN→IP. Figure 3.4a shows the setup scenario for SCN→SCN calls; Figure 3.4b shows the scenario for IP→SCN calls; Figure 3.5a shows the scenario for IP→IP calls, and Figure 3.5b shows the scenario for SCN→IP calls.

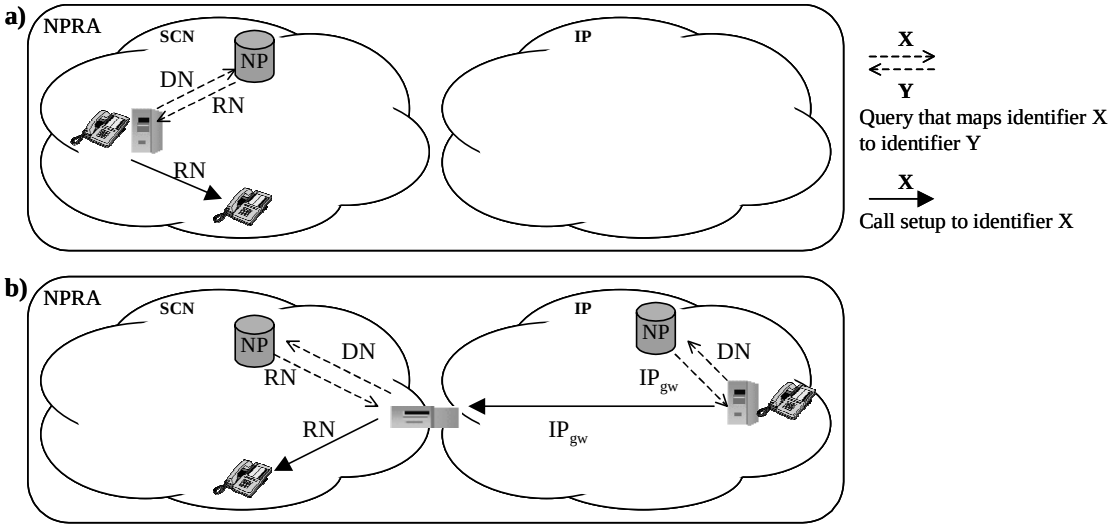

Figure 3.4: Separate information model scenario for destinations in the SCN

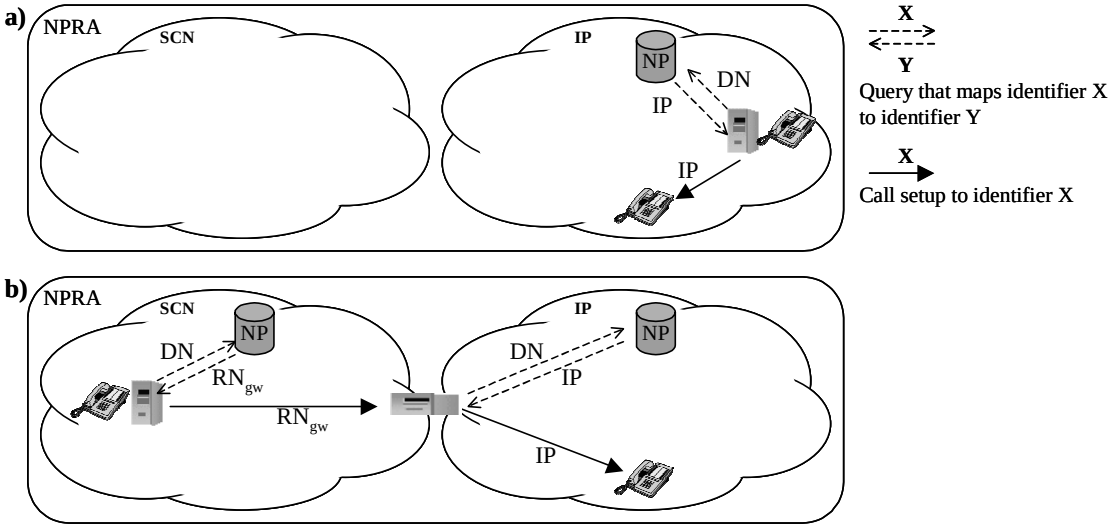

Figure 3.5: Separate information model scenario for destinations in the IP network

A SCN→IP or IP→SCN call can be understood as consisting of two parts: first routing to the gateway, and then routing from the gateway to the destination. Thus, a SCN→IP call requires two mappings:

1. $DN \longrightarrow RN_{gw}$
2. $DN \longrightarrow IP$

and a IP→SCN call requires the mappings:

1. $DN \longrightarrow IP_{gw}$
2. $DN \longrightarrow RN$

We now look at the shared information model, where the SCN and IP networks have identical information. Both the SCN and the IP network have a mapping DN→RN and DN→IP_{gw} for destinations in the SCN. For destinations in the IP network, both networks have a mapping DN→IP and DN→RN_{gw}.

In the shared information model, both queries can be combined to a single query. Thus, a new query is not required after crossing the technology border. Assuming a single query, Figure 3.6a shows the scenario for SCN→SCN calls; Figure 3.6b shows the scenario for IP→SCN calls; Figure 3.7a shows the scenario for IP→IP calls, and Figure 3.7b shows the scenario for SCN→IP calls. However, in order to be able to combine the two queries into a single query, the networks must be able to transport the identifier of the opposite technology. In the IP network this is possible using the "identifier@gateway" format, where the identifier is the routing number and the gateway is the name or address of the gateway. In the SCN, the URI or IP address cannot be transported, which limits the value of the model.

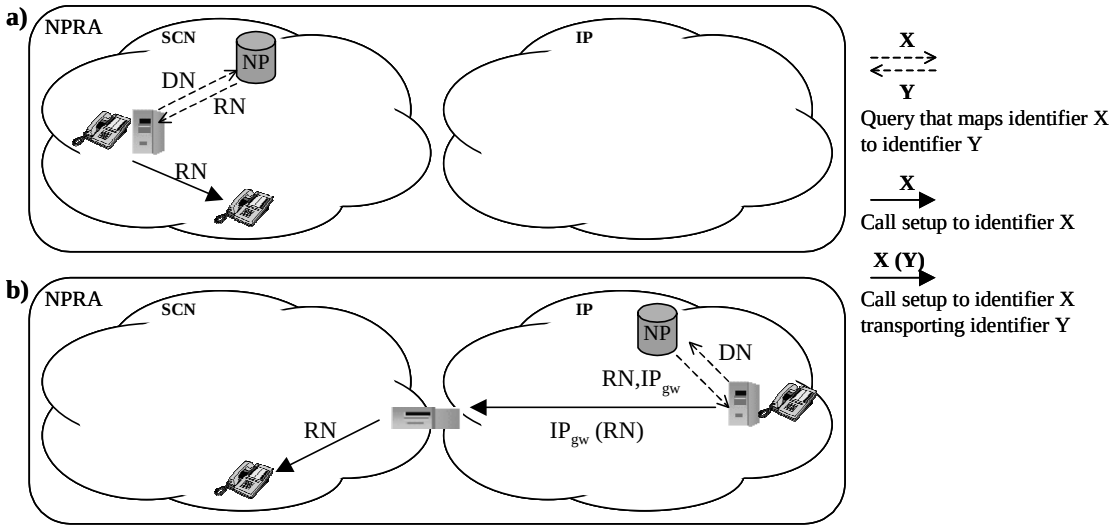

Figure 3.6: Shared information model scenario for destinations in the SCN

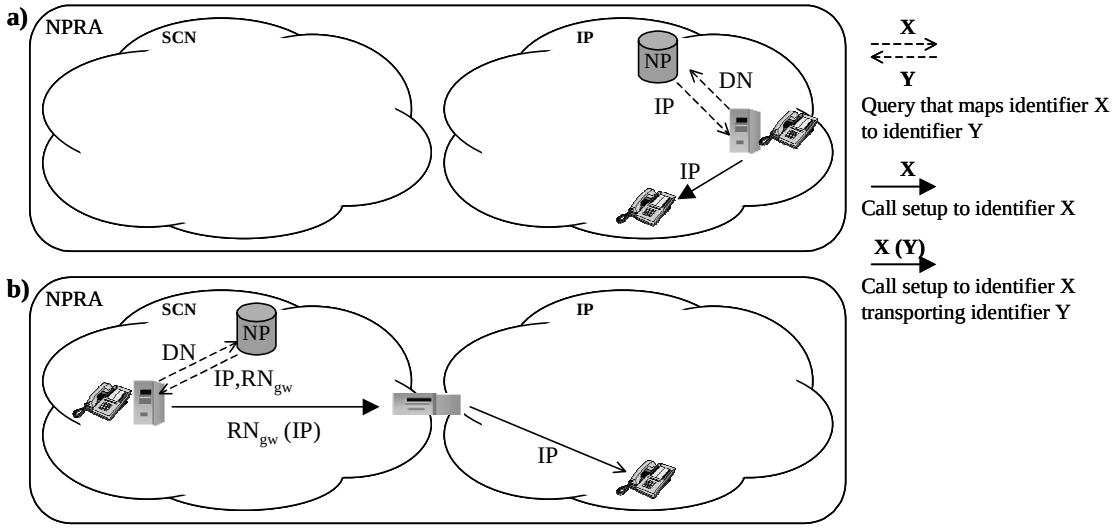

Figure 3.7: Shared information model scenario for destinations in the IP network

Like the shared information model, full utilization of the shared IP-information model requires that the SCN should be able to transport the IP addresses. Currently this is impossible. However, the shared SCN-information model is viable. The call setup scenario is represented by the combination of Figure 3.6 and Figure 3.5. For destinations in the SCN, the databases in the SCN contain a mapping DN→RN and the databases in the IP network contain the mappings DN→IP_{gw} and DN→RN. For destinations in the IP network, the databases in the SCN contain a mapping DN→RN_{gw} and the databases in the IP network contain a mapping DN→IP.

The shared information model can be seen as a super-class of the shared IP-information model and the shared SCN-information model. The shared IP-information model and the shared SCN-information models can be seen as super-classes of the separate information model. All information required in the sub-class model is available in the super-class model. Therefore, some networks may use the separate information model if the architecture is based on the shared SCN-information model.

3.4 Functional models for shared information

In this section, we extend the information models described in the previous section by considering the function of the information in addition to the distribution. We saw that the two queries can be combined to a single query if the address of the opposite technology can be transported. Although this is not always possible due to technical or management reasons, the information of the opposite technology is still useful. This section examines the different functions that can be implemented based on information used outside the technology to which it relates.

For example, if the IP network has access to the information about the numbers in the SCN, it can use it

- to determine whether a specific number is in the SCN or not,
- to determine where in the SCN a specific number is, or
- to determine which routing address is used to route calls to a specific number.

We divide the functions into three levels of increasing functionality:

- *Existence function.* Given a number, determine whether it exists in the specific technology.
- *Network identity function*. Given a number, determine in which network in the specific technology it currently is. The network may be identified with for example a TAD identifier or by the prefix of the routing address.
- *Routing address function*. Given a number, determine the routing address of the given technology.

These can be seen as functions that map a directory number into a Boolean value, a network identifier or a routing address, respectively. Common to the functions is that the information origins from the opposite technology.

The value of the information is also determined by the scope of the included terminals. For example, the databases of a number portability implementation may include either all terminals or only the ported terminals. Let the *completeness condition* represent whether the information includes all terminals. When the completeness condition is true, non-ported numbers are included in addition to the ported numbers.

We now apply these functions to SCN and IP networks. When SCN information is accessible in the IP network, the following functions can support IP→SCN calls.

- *Number validation* implemented with the *existence function*: By default, an IP network routes calls to all unrecognized numbers (i.e. numbers without any ENUM entry) to the SCN. Knowledge whether a given number actually is in the SCN can be used to avoid unnecessary routing of calls to the SCN when the number is malformed. During call setup, the signaling server can check whether the given number is a valid SCN destination, and abort the setup before it enters the SCN if the number is invalid. This saves network capacity and gateway resources. However, it requires that information about all SCN numbers is available, thus the completeness condition must be true.
- *Inter-technology portability support* implemented with the *existence function*: The existence function can tell that a number has moved from the IP network to the SCN. Since this is also visible through the removed ENUM entry, the inter-technology portability support function is not useful for IP→SCN calls.

- *Gateway selection* implemented with the *network identity function:* Information about in which network a number resides can be used in gateway selection for IP→SCN calls. The IP network can then make the routing decision based on the real location of the destination, instead of using only the directory number. Especially for ported numbers, the directory number does not tell the actual location. The call can be routed directly to the current serving network instead of passing the donor network. A mapping between the network identifier (e.g. TAD or routing number prefix) and the gateway is required. Since the information is most useful for ported numbers, the completeness condition may be false.

- *Single query* implemented with the *routing address function:* The directory number can be translated to a routing number already in the IP network. The SIP URI then contains the routing number instead of the directory number. The advantage is that the query is performed only once, and a new query in the SCN is not required. This function provides optimization only, and works for partial information as well. Thus, the completeness condition may be false.

When information about IP terminals is accessible from the SCN, the following functions can support SCN→IP calls. Due to the different properties of IP and SCN technologies, these are not identical to the corresponding functions above.

- *Number validation* implemented with the *existence function*: The SCN does not by default route calls to unrecognized numbers to the IP network. However, it can be assumed that a network completely implemented with IP technology is assigned a number block, whereas SCN networks install fixed routes to a gateway for numbers in this block. Number validation can drop calls to nonexistent numbers within the block before they enter the IP network. For this, the information must include all terminals, i.e. the completeness condition must be true.

- *Inter-technology portability support* implemented with the *existence function*: A minimum requirement for inter-technology number portability is that information about which numbers are in the IP network is available. If a number moves to the IP network, there must be an indication about this in the SCN. One solution would be to map the directory number to a routing number leading to a gateway. Another solution is to use the existence function, which indicates that the number currently resides in the IP network and that the call shall be directed to a gateway. The specific gateway is not indicated. The indication is only required for ported numbers, i.e. the completeness condition may be false.

- *Gateway selection* implemented with the *network identity function:* Information about the network identifier, IP address or URI of the terminal can be used in gateway selection for SCN→IP calls. A mapping from the network identifier or IP address to a list of suitable gateways is required. The completeness condition may be false.

- *Single query* implemented with the *routing address function:* Theoretically the directory number could be translated to an IP address already in the SCN network, saving a second

query in the IP network. The problem with this function is to transport the IP address in the SCN. The completeness condition may be false.

Figure 3.8 illustrates routes that utilize number validation (marked 1), gateway selection (marked 2) and single query (marked 3).

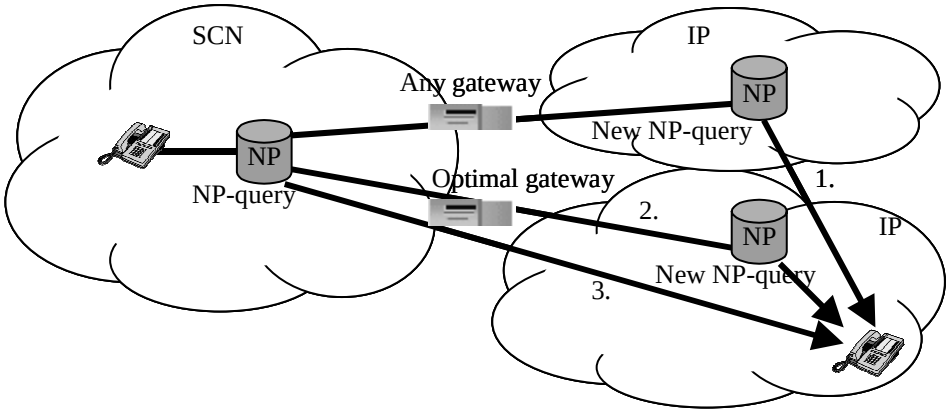

Figure 3.8: Routes based on different information function models.

3.5 Example scenario

To clarify the use of concepts, we summarize the most presumable scenario of current development in Finland. This scenario uses the Master database for number portability in the SCN. There is no automatic gateway location; instead, gateway location is implemented with a static mapping from the directory number to the routing number of the gateway. In the IP network, ENUM is used for terminal location and for implementing number portability. TRIP is used for gateway location for IP→SCN calls.

The scenario can be described with the following set of mappings:

$DN \xrightarrow{DB_{master}} RN$ (SCN→SCN calls)
$DN \xrightarrow{static} RN_{gw}$ (SCN→IP calls)
$DN \xrightarrow{DNS} IP$ (IP→IP calls)
$DN \xrightarrow{TRIP} IP_{gw}$ (IP→SCN calls)

In this scenario, number portability information is not distributed between the SCN and the IP network. Thus, the separate information model is used, and the information function models do not apply. Consequently, gateway selection is inefficient, number portability may cause inefficient routing, and two queries are required for calls between the two technologies.

3.6 Summary

In this chapter, we defined the terminology and requirements that will be used in the discussion in subsequent chapters. We introduced a notation for describing mappings between two identifiers with a given method. We defined the terms routing address validity area (RAVA),

mapping function validity area (MFVA), and number portability routing area (NPRA). We constructed rules that ensure correct routing to ported (and non-ported) destinations using routing addresses, and proposed how these rules can be used to select a valid gateway in hybrid networks.

In a hybrid network scenario, information may be distributed to or accessible from another technology than the one that it represents. We described various information models, divided into distribution models and function models. The distribution models represent the combinations of technologies to which mappings are distributed. The function models represent the functionality that is implemented with the information of the opposite technology. Furthermore, the functionality is determined by the scope of the information, represented by the completeness condition. The Completeness Condition is true if the information represents all terminals, and false if it represents only ported numbers.

Chapter 4
Shared database mappings

Among the mapping methods considered in this work, the most generic are the static mappings and the shared database mappings. Practically any type of mapping can be implemented with these methods, and there are various ways of implementing them. The chapter begins by defining the terms static mapping and database mapping as used in this work. The rest of the chapter discusses shared database mappings, where the database information is shared between operators. First we concentrate on number portability and then on gateway location. Because of the generality of database mappings, we study two implementations of number portability in particular: the Master system by Ficora and the database solution developed at HUT. After a short description, we analyze how they can be adapted to a hybrid SCN-IP scenario.

4.1 Terminology and objective

By a *static mapping*, we refer to a mapping that is implemented with a set of constant (source identifier, destination identifier) tuples. The tuples can be stored in a memory structure or in a database. In the viewpoint of this work, the property that distinguishes a static mapping from an actual database mapping is that the static mapping is local to the network element or the operator. There is no method of distributing and synchronizing information between operators. Configuration of static mappings is more or less manual, and therefore they can be characterized as demanding from maintenance perspective. The efficiency of static mappings depends on the application and implementation. For example, a routing number containing a TAD identifier can easily be generated from the TAD by simple concatenation and replacement methods.

We consider all mappings that are local to the operator as static, even if they may be implemented with a database. We also consider proprietary solutions that are specific to one operator as static mappings.

We use the term *shared database mapping* (DB mapping) to refer to a mapping implemented with a database that is shared between operators. Sharing can be implemented by replicating the database contents with e.g. SCSP (Server Cache Synchronization Protocol) or a database-specific protocol, or by accessing the contents remotely using a query protocol, e.g. SQL or LDAP. By using a replicated database, the load is shared between several databases, and the distances of queries can be reduced.

A shared database mapping can be maintained either centrally (e.g. by a third party or the regulator), or in a completely distributed fashion based on peer-to-peer relations. In a centrally maintained database implementation, the third party maintains a master database to which modifications are made by request. The database contents are accessed from the master database directly, or in order to scale, the contents are distributed in one direction to the operators. Alternatively, the third party acts as a coordinator, coordinating the distribution of updates made directly to the operators databases. In the completely distributed approach, every party can modify its own database and the changes are distributed to the other operators. Even other configurations are possible.

In this chapter, we will look at one existing and one proposed database implementation. However, because of the multitude of possible implementations of shared databases, these mainly represent example implementations, and when complete scenarios are discussed we consider databases at a more general level. We are mainly interested in the possibility to implement a mapping with a database, and the properties of such an implementation. We do not take into account the internal structure of an operator's network – instead we are interested in the case when the database is accessed or replicated between operators. With shared database mapping methods, we refer to any type of database that can be used between operators without specifying any specific implementation.

4.2 Database mappings for number portability

First, we examine number portability solutions based on a centralized number portability database (NPDB). The database contains the mapping between the subscriber number and the routing address, i.e. it performs the mapping $DN \xrightarrow{DB} RN$. The regulator is the formal administrator of the database, although the technical maintenance may be delegated to a third party. The database contains entries for all allocated numbers, or only for ported numbers.

We examine two specific implementations. Firstly, the Finnish telecommunications regulatory authority Ficora has developed a solution based on a centralized database named *Master database* [Ficora 2002b]. The database contents are replicated to databases in each operator's network. Another solution [Paju 2002] has been developed in the Networking Laboratory at Helsinki University of Technology (HUT). This solution also uses a centralized database system, with information replicated to databases of the operator. The difference is in the replication and update procedures.

In both solutions, information about number portability is exchanged trough a centralized database. No information is exchanged directly between the operators. No queries are performed to the central database. Instead, queries are performed to a copy of the database maintained by the operator. We first describe each solution, and then examine how they can be used in a hybrid scenario.

4.2.1 The Master system proposed by Ficora

The Master system [Ficora 2002b] can be divided into two functional parts. The Master database contains the information necessary for routing to ported numbers and for managing the process of moving a number from one operator to another. The server application provides the interface for replication, management, fault handling, government intervention, and for announcement service. The main functions of the system are to relay information about moving numbers between operators, and to store and distribute information about moved numbers. Since the system is implemented by a third party, Ficora has only specified the most central functions and requirements, and left the implementation details open.

The services provided by the Master system are:

1. Porting of a number from an operator to another for the first time.
2. Porting of an already ported number from an operator to another.
3. Release of a number back to the donor operator.
4. Termination of a ported number when the subscription is terminated.
5. Database replication.
6. Reports and fault handling.

A number is in one of the 12 states that are presented in Table 4.1. For the purpose of fault management, the system also stores the time that the number has been in a state. The system can alert if the number has been too long in a given state.

The Master system has interfaces to operators, authorities and external parties. The interfaces in the master system are based on secured TCP/IP connections. Information is transferred with three different methods: files for mass-transfers, events for smaller updates and a web-interface. The interfaces are depicted in Figure 4.1. The information consists of messages or events. The messages are presented in Table 4.2. The process of transferring a number from one operator to another is depicted in Figure 4.2.

Table 4.1: States of a ported number in the Master system. [Ficora 2002c]

Abbreviation	Description
TR	Transfer Request
TC	Transfer Confirmed
TOK	Transfer OK
AST	Approval of Switch Time
TRC	Transfer Request Cancel
SS	Subscription Switch
SUS	Subscription UnSwitch
TNP	Transfer Not Possible
NT	Number Termination
DT	Delay of Transfer
TOO	Transfer to Original Operator
RTC	Request of Transfer Change
TRCOK	Transfer Request Cancel OK

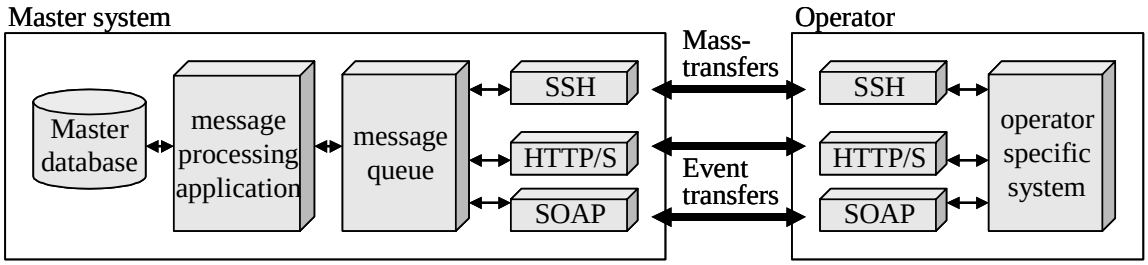

Figure 4.1: Interfaces of the Master system

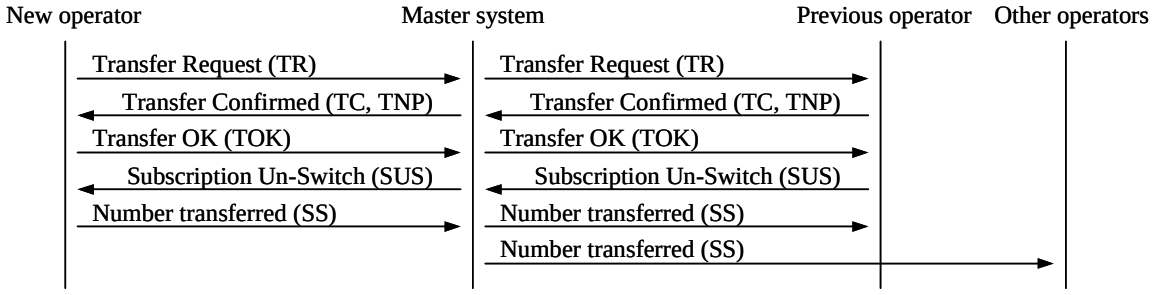

Figure 4.2: Transfer of a number between two operators

Table 4.2: Messages in the Master system.

Abbreviation	Description
CAN	Cancel. Cancellation of porting order.
DTR	Delay Transfer Request. The donor operator delays the porting.
NPC	Number Porting Confirmation. The receiving operator confirms the porting.
NPC-NOTICE	Notice about the confirmation.
NPO	Number Porting Order. Order to port a number.
NPO [Termination]	Number Porting Order. Termination of the number. Frees the number to the original operator.
NPOC	Number Porting Order Confirmation. The donor operator confirms that the number can be ported.
NPO-NOTICE	Notice about the porting order.
NPOR	Number Porting Order Rejection. The donor operator or the master system rejects the porting order.
SC	Subscription Connection. The subscription has been connected.
SC-NOTICE	Notice about the connection of the number.
SCO	Subscription Connection to Original. The number is connected back to the network of the original operator.
SD	Subscription Disconnect. The number is disconnected from the donor operator's network. The donor still manages the number.
SD-NOTICE	Notice about the disconnection.
SMS-NOTICE	Notice sent with the short message service.
TERMINATION-NOTICE	Notice about the termination of a subscription.

Files are used for transferring large amounts of data. The operator transfers the files into a directory on the Secure Shell (SSH) server of the Master system. The files are processed by a message queue in the order of transfer according to the date and time of the file. The process is started either when new files are received, or according to a schedule. The result is a set of files, which are put into the directories of every operator that should receive a reply from the operation. The operators then fetch the resulting files from the SSH server at scheduled times. All files are in XML format. A file consists of records, which correspond to events. In the beginning of the file is a start record and in the end a stop record.

Event-driven transfer provides real-time data transfer for smaller data amounts with protocols such as HTTP/S and SOAP. The events are transferred as files in XML format as well. Processing of single events is similar to processing of files: the same message queue processes both incoming events and incoming files in chronological order. As a result of the process, a set of files is generated. The operators that use event-driven transfers receive a message when the process is complete.

In addition, a web-interface is provided. The interface has functions for numbers transfers, report generation, maintenance, fault management and queries.

4.2.2 The SQL database solution proposed by HUT

In HUT's proposed database solution [Paju 2002], all information transfers are implemented as SQL (Structured Query Language) commands. To provide interoperability between different database system vendors, Open DataBase Connectivity (ODBC) is used. The connections are established on demand.

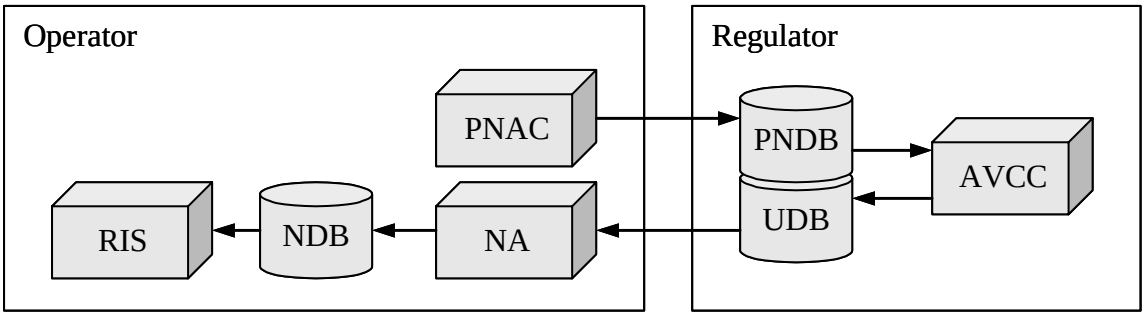

Figure 4.3: Architecture in the HUT solution

The architecture is depicted in Figure 4.3. The information flow related to a ported number is marked with arrows. The regulator (or another third party) maintains an *Update Database* (UDB) containing the most recent updates of numbering information. The *Numbering Agents* (NA) of the operators read the information in the UDB and update the operator's local *Numbering Database* (NDB) to contain the current information. Every operator has a NDB, which contains information about all ported numbers. When calls are set up, the *Routing Information Server* (RIS) in the network queries the NDB to obtain the routing number. The RIS may be part of the telephone exchange or implemented in a SDP. The regulator can maintain several independent UDB elements for redundancy. The UDB contains only the most recent updates. The older information, which already has been fetched by the operators, is periodically removed.

When a number moves from an operator to another, the *Ported Number Advertising Client* (PNAC) in the new operator's network updates the new routing number in the *Ported Number Database* (PNDB) of the regulator. The PNAC of the previous operator must confirm the moved number in the PNDB. When the *Advertisement Validity Checking Client* (AVCC) sees a confirmed and valid entry in the PNDB, it updates the information in the UDB. The PNAC is responsible for the success of the update, and an unsuccessful update is retried with exponentially increasing delay. There may be multiple PNDB elements for redundancy. In this case, the PNAC must update all of them.

The Numbering Database (NDB) is the most frequently accessed database, since it replies to several queries per call setup. For scalability, it is divided into several tables according to their prefix. The existing tables are marked in a database description.

The solution assumes that all numbers are portable. A query is performed for all calls. The Routing Information Server (RIS) is responsible for querying the NDB to obtain the routing number. RIS receives numbers from the user transported by the signaling protocol. The RIS contains a description of the database, which informs how many digits are required before the first query is performed. While digits are received, the RIS queries the NDB until a result is received.

4.2.3 Applicability to hybrid SCN/IP networks

Both the Master system and the HUT solution use TCP/IP as the transport protocol. Access to the database information can therefore also be provided to the IP network; thus, the shared SCN-information model can easily be implemented. If the information covers IP terminals as well, the shared information model can be implemented. Nevertheless, for the information to be useful on the IP network, it must include routing information suited for the IP environment. In this section, we analyze how the database solutions can be used in a hybrid scenario by applying the information function models described in section 3.4.

Two conditions on the database contents determine how the database solution can be used in a hybrid scenario:

- *IP-inclusion condition.* Does the database contain information about IP terminals in addition to the information about SCN terminals? If true, the shared information model can be implemented. If false, only the shared SCN-information model can be implemented.

- *Completeness condition.* Does the database contain information about non-ported numbers in addition to the information about ported numbers? If true, number validation can be implemented.

The Master system is designed to be used in the SCN only. The database contains a routing number corresponding to each ported directory number, and there is currently no possibility to store information about IP terminals. It means that the IP-inclusion condition is false. On the other hand, the HUT solution has taken both SCN and IP networks into consideration. A terminal in the IP network has an IP address as the routing address, and a terminal in the SCN has a routing number. Because of its IP support, the IP-inclusion condition is true.

However, database solutions are generally easy to extend, as the database implementation is independent of the specific technology. Therefore, it is conceivable that also information about moved IP terminals would be stored in the Master system as IP telephony becomes common. This is based on the assumption that the same regulation is applied to IP telephony. Information about IP terminals is especially important when numbers are allowed to move between the two technologies. An essential question is then what information should be stored instead of the routing number. One alternative would be only an indication that the number is in the IP network, enabling only the existence function. Another alternative is to store the current IP address.

The IP-inclusion condition becomes true also if the SCN network accesses information about IP terminals through some separate system, e.g. DNS. Then, a second query is required to obtain information about the IP network.

According to [THK 1996], only moved numbers are stored in the number portability database during the first phase, and the completeness condition is therefore false. In the second phase, all numbers are inserted into the database.

For IP→SCN calls the following functions can be implemented:

- The *number validation function* provides an advantage in reducing traffic to malformed numbers. The cost is an extra database lookup during call setup, which may not be worthwhile if the lookup does not provide any additional benefits.
- The *inter-technology portability support function* indicates that a number has moved from the IP network to the SCN. In practice, this function is redundant since the removed ENUM entry already indicates a ported number.
- The *gateway selection function* provides the information for gateway selection based on the routing number. The routing number is more suitable than the directory number for gateway selection since it can be aggregated. Gateway selection provides a major advantage in avoiding routing IP→SCN calls through the donor network, which is a burden that should be eliminated. This extra leg can be avoided if destination network is known already before the call enters the SCN. Otherwise, the call will most likely be routed to a gateway connected to the donor network or to the gateway nearest the originating network.
- The *single query function* does not provide any significant advantages. It saves a second database lookup in the SCN, which is worthwhile if the lookups are costly. As discussed in Chapter 3, a routing number is usually only valid within a specific area. Therefore, the valid gateways are reduced to the ones within the same number portability routing area (NPRA).

If the IP-inclusion condition is true, the following functions can be implemented for SCN→IP calls:

- The *number validation function* can avoid call setups to malformed numbers, provided that information about all numbers are available (the completeness condition is true). Further, the number portability database is the natural place to indicate that a number has moved to the IP network.
- The *inter-technology portability support function* indicates that a number has moved from the SCN to the IP network.
- The *gateway selection function* allows automatic selection of the most suitable gateway. A list of candidate gateways is given for each network identifier or IP address prefix (address,

mask combination). This requires an additional gateway location mechanism for the mapping.

- The *single query function* is impossible to implement with any solution, because the SCN cannot transport the IP address. A new query must therefore be performed by the gateway to obtain the IP address of the destination.

The thesis [Paju 2002] suggests an optional integration of TRIP and CTRIP into HUT's database solution. The TRIP and CTRIP protocols are in this case only used for gateway location, and the routing address is used as the key for locating the gateway. Thus, for an IP→SCN call, the routing number is used to locate the gateway with TRIP. Correspondingly, the IP address is used to locate the gateway using CTRIP for calls in the opposite direction. By using the routing address in gateway location, the scalability problems related to TRIP and CTRIP can be solved. The use of a number portability database together with TRIP is discussed in Chapter 7.

For inter-technology number portability, there must be a method to indicate that a number has been ported to the IP network. If the database contains information about IP terminals, the inter-technology portability function can be used. Otherwise, the entry in the database is either removed or replaced by a routing number to a gateway. In the former case, there must be another way to identify that the number is in the IP network, for example with a static route. In the latter case, the routing number is fixed, which does not allow for dynamical selection of gateway.

4.3 Gateway location with a database mapping

If a few gateways are available to a network, the addresses of these gateways can be stored in a database that the elements in the network can access. This type of operator-specific mapping is within our definitions for static mappings. However, if the mapping is shared between several operators, we can talk about using a shared database mapping for gateway location. The difference is visible in the case where an operator can use gateways belonging to other operators. In the static gateway mapping, gateway discovery is practically manual since the operator must obtain the information about the gateways and add the corresponding database entries. Only the originator-determined policy model is possible. In a shared database mapping, each operator can insert information about its gateways into the shared database, and the other operators discover these gateways indirectly. Thus gateway discovery can be made automatic. The originator-determined policy model is possible, but also the destination-determined policy model can be implemented.

The database information is only required to be shared between the networks using each other's gateways. In this way, groups of networks sharing gateways can be formed. Although a third party can maintain the database, this is not required since administration and authorization can be implemented using database mechanisms. There is no need for coordination in a similar way as for number portability.

The input of the database mapping is a directory number, a routing number, an IP address or a TAD. Directory numbers, routing numbers and IP addresses can be aggregated into prefixes, which reduces the number of entries. Due to number portability the aggregation of directory number is inefficient. Currently there are no plans for geographical allocation of TADs to networks. Therefore, these cannot be aggregated, and the number of entries is equal to the number of TADs. However, the total number of TADs is rather low if they are used in a limited area, e.g. in a NPRA.

Currently there are no known implementations of gateway location with shared databases. Most gateway location implementations are based on static mappings.

4.4 Summary

In this chapter we discussed the use of databases for number portability and gateway location. We divided database mappings into shared database mappings and static mappings. As examples of shared database mappings, we presented the Master system proposed by Ficora and the SQL-based database system of HUT. We analyzed how the number portability information can be used in a hybrid IP-SCN scenario. HUT's database has the capability to store information about IP terminals. The Master system is intended to be used in the SCN only, but simple extensions could allow information about IP terminals. Databases can also be used to distribute mappings for gateway location between operators sharing gateways.

Chapter 5
Number portability with ENUM

ENUM maps the directory number to a routing address for each call. Therefore, it can be considered as a natural place for implementing number portability in IP networks. We first discuss some implementation issues briefly. Then, in order to integrate the SCN and IP telephony networks, we examine how the ENUM information can be used in the SCN, and whether number portability can be implemented with ENUM in the SCN as well. We develop a routing number URI for specifying a routing number corresponding to a directory number. We define the its usage and describe some implementation scenarios. To reduce the amount of DNS queries in overlap sending, we finally present a method for storing the expected number of remaining digits in DNS.

5.1 ENUM-based number portability within the IP network

Number portability requires a mapping from a directory number to the corresponding routing address. In an IP telephony network, ENUM provides such a mapping. By changing the URI to which the number is mapped, the calls are directed to the new destination without passing through the donor network. Thus, number portability can easily be implemented by updating the URI in the NAPTR record. The solution works equally well for calls originating from the SCN, since the gateway performs the ENUM query in these calls.

Although the technical implementation is simple, the administration may cause some problems depending on the chosen implementation model (see section 2.2.2). Especially problematic is the operator-maintained model, where the donor operator is responsible for maintaining the NAPTR records of the numbers that have been ported out of his network. During the transition phase, only some operators use ENUM. Then, if a number moves from an operator not using ENUM to an operator using ENUM, there is no NAPTR record in the donor network to update. Consequently, the number cannot be transferred. A solution in this case would be to move the ported number up in the hierarchy to Tier 1. However, this is against the principles of DNS, since a server cannot delegate a number range to another server and simultaneously maintain some of the numbers in the range itself. [Rostela 2002, Albitz 1997]

A possible solution to this problem would require the regulator to maintain the ENUM entries on behalf of the operators that have not yet implemented it. Later when the operator has

implemented ENUM, the database can be transferred to the responsible network. However, in practice it can be difficult to pull down such a system later. Rostela [Rostela 2002] suggest a solution to use a separate server for ported numbers. This solution requires two queries: the normal ENUM query first returns a routing number, then the ported server is queried with the routing number, whereas the URI is returned.

The U.S. model and the regulator-maintained model do not have problems with numbers moving from operators without ENUM. In these, the ENUM entries are maintained by an independent body. However, since the U.S. model maintains the same entries at both Tier 1 and 2, it involves the load of updating number information at both tiers. On the other hand, the regulator-maintained model requires a third party to maintain the ENUM system, which may cause an increase in the annual number fee. [Rostela 2002]

5.2 Using ENUM in the SCN to determine endpoint type

ENUM is intended to be used in the IP network. However, in the rest of this chapter we examine whether ENUM can be used in the SCN as well. Many network elements in the SCN have access to the public Internet or to a private IP network. Above all, the Intelligent Network (IN) is a suitable place for implementing IP based services because it normally has IP connectivity. Let us further assume that IN has access to information in DNS and thereby access to numbering information in ENUM. This would allow us to use ENUM in the SCN as well.

In a normal call setup, the first exchange typically makes an IN query for numbers that are marked as portable. The IN query returns the routing number of the destination. As number portability becomes widespread, moved numbers are more of a rule than an exception, and a large part of the incoming call setups require mapping. In that situation, it is more scalable to perform an IN-query on every call setup than to maintain a separate list of all ported numbers that require special handling.

By performing an ENUM query at the call setup, the originating SCN network can determine the technology of the destination. If the ENUM query returns nothing, the destination is an SCN terminal and the call setup can proceed normally. If the ENUM query returns an SIP URI or any other URI used for IP telephony, the terminal is known to reside on the IP network and the call should be directed to a gateway. The gateway then performs a new ENUM query, which is used to establish the call on the IP side. The procedure for locating the gateway is left open at this point, as it will be discussed in Chapter 6. In this approach, the originating network knows in an early stage that the destination is an IP terminal, and the call can be directed to a gateway directly instead of routing it through the SCN. For international calls this can save significant costs. Using the terminology of section 3.4, we are able to implement the existence function for SCN→IP calls.

5.3 Using ENUM in the SCN for number portability

The purpose of an ENUM query is to obtain a list of possible methods to contact a host. The list should be processed in priority order, so that the method with the lowest value in the order and preference fields of the supported methods is tried first. Currently these methods correspond to IP based protocols. However, with simple extensions it is possible to include SCN based methods into ENUM.

5.3.1 Design of a routing number URI

In this section, we introduce a URI that specifies the routing number of destinations in the SCN. This "routing number URI" represents a method for contacting a host in the SCN. When an ENUM query is performed by an SCN element, the routing number URI is selected, since it is the only method supported by the SCN. If several routing number URIs are available, the one with the highest priority is selected. On the other hand, when an ENUM query is performed from the IP network, the routing number URI is ignored, since it is not supported in the IP network.

For the purpose of the discussion, we define a new ENUM-service, which at the time of writing is experimental. Experimental ENUM-services are preceded with "X-" [Faltstrom 2003]. Therefore, we give this ENUM-service the identifier "X-rnum". The ENUM-service has no subtypes. Additionally, a new URI type must be defined. For this purpose, we use the following format given in ABNF [RFC 2234] notation:

```
routingnumber-uri    = "rnum:" routingnumber *parameter
routingnumber        = 1*HEXDIG
parameter            = ";" pname ["=" pvalue]
pname                = ALPHA / ALPHA *(alphanum / "-") alphanum
pvalue               = *paramchar
paramchar            = alphanum / escaped / unreserved
escaped              = "%" HEXDIG HEXDIG
unreserved           = "[" / "]" / "/" / ":" / "&" / "+" / "$"
```

The following is an example record for a terminal in the SCN:

```
$ORIGIN 3.0.3.5.1.5.4.9.8.5.3.e164.arpa.
 IN NAPTR 2 0 "u" "E2U+X-rnum" "!^.*$!rnum:1D8819578345!"      .
```

DNS is a global database, and a query returns the same result independently of where the query is performed. This poses two problems when number portability is implemented with ENUM: (1) the routing number must be globally unique and (2) it must be globally routable. The first of these problems is due to overlapping routing numbers in different countries. In the above example, the number "1D8819578345" may be a valid routing number in Finland, but some other country might use the same number for a different destination. To solve this problem, the country code can be added as a prefix in front of the number. Alternatively, it can be given as a parameter in the URI.

The second problem is harder to tackle. Firstly, the use of digits with number-base over ten in the routing number may cause problems in some networks. Secondly, combining a routing number with a country code might result in a number that is longer than the maximum of 15 digits allowed in ITU-T standards [ITU-T E.164]. The main issue, however, is that the solution should work even if it is used only in a few countries. It is not feasible to demand all networks to support a specific solution.

Because of these problems, we see that the best solution is to limit the validity of an entry to the set of networks supporting the specific format of routing numbers, i.e. to a specific routing address validity area (RAVA). Following the principles of section 3.2, the call is routed with the directory number to the correct RAVA, and the routing number is obtained from DNS in the first exchange supporting the described scheme. The directory number contains enough information to route the call to the RAVA, and numbers are not allowed to move outside their validity area. However, the entry point in the RAVA cannot be optimally selected.

The routing address validity area can be given with an additional parameter to the URI. The parameter identifies the country, within which the routing number can be used. If a number is valid in several countries, the parameter contains the list of countries. We propose to use the top-level domain codes [IANA TLD] for indicating the country. Another approach would be to use country codes [ITU-T CC]. In either case, the URI would be in the same format, with a list of country identifiers added to the base URI, for example “rnum:1D8819578345;valid=fi,se,no,dk”. In reality, validity areas are unlikely to include several countries, but rather to cover only part of a country. This type of validity areas can be implemented by adding an area identifier to the country identifier, for example separated by a dot. The more specific identifier precedes the less specific, in a similar way as in the domain name system. A routing number valid in only the Mid-Atlantic number portability area in the U.S. might be “rnum:1D8819578345;valid=ma.us”, for example. With the country code approach, a routing number valid in Helsinki area in Finland might be “rnum:1D745897;valid=+3589”. With the validity area information added, the URI-type format (based on the format in [Schulzrinne 2002]) is the following:

```
routingnumber-uri    = “rnum:” routingnumber *par
routingnumber        = 1*HEXDIG
par                  = validity / other-parameter
validity             = “;valid=” area *(“,” area)
area                 = domainname / global-number-digits
domainname           = 0*(subdomainname) toplabel
subdomainname        = toplabel “.”
toplabel             = ALPHA / ALPHA *(alphanum / “-”) alphanum
alphanum             = ALPHA / DIGIT
global-number-digits = “+” 1*phone-digit
phone-digit          = DIGIT / visual-separator
visual-separator     = “-“ / “.” / “(“ / “)”
other-parameter      = “;” pname [“=” pvalue]
pname                = ALPHA / ALPHA *(alphanum / “-”) alphanum
pvalue               = *paramchar
paramchar            = alphanum / escaped / unreserved
escaped              = “%” HEXDIG HEXDIG
unreserved           = “[“ / “]” / “/” / “:” / “&” / “+” / “$”
```

5.3.2 The routing number URI vs. the “tel” URI

The format of the routing number URI is very similar to the “tel” URI defined in [RFC 2806] and updated in [Schulzrinne 2002]. Also the “tel” URI allows the scope of a local number to be limited to a region. The “phone-context” parameter of the “tel” URI has the same function as the “valid” parameter in the “rnum” URI. Thus, the format of both the URI types are rather similar.

The reason for defining a new URI is to separate between the completely different functions of the two URI types. The “tel” URI is only an identifier and does not imply dialing semantics [Schulzrinne 2002]. On the other hand, the “rnum” URI defines the dialing semantics, and it should only be used for internal routing in its validity area in a SCN network.

According to [Schulzrinne 2002], “the ‘tel’ URI describes a service, reaching a telephone number, that is independent of the means of doing so, be it via a SIP-to-PSTN gateway, a direct SIP call via ENUM translation, some other signaling protocols such as H.323 or a traditional circuit-switched call initiated on the client side via, say, TAPI.” Thus, the “tel” URI can be used e.g. on web pages. If used in the NAPTR records, it generally means that another ENUM lookup should be performed with the given telephone number. On the other hand, the “rnum” URI should only be used in the NAPTR records. It is dependent of the means of reaching the specific destination.

5.3.3 Usage of the routing number URI

The usage of the routing number URI is largely dictated by the rules presented in Chapter 3. The networks supporting the “rnum” scheme constitute the mapping function validity area (MFVA).

The routing address validity area (RAVA) is specific to the routing number, and is expressed as a union of domain names (e.g. fi), E.164 prefixes (e.g. 358), or parts of these (e.g. 3589). The

validity areas are defined separately for every single number; thus, for prefixes the rules must be satisfied for every number included in the prefix. The validity area of an entry must only contain the countries or parts of countries where the number is valid[2]. If no validity area is given in the "valid" –field of an "rnum" URI, the routing number is globally valid. For this type of routing number, all networks must be able to route the call. For example, the Mobile Station Roaming Number (MSRN) used in GSM fulfils the requirement. The globally valid routing numbers are usually implemented by adding the country prefix to the routing number and reducing the effective routing area to a smaller region although the routing number itself is globally valid.

To specify the use of the routing number URI, we refine the rules defined in section 3.2.

The purpose of the first three rules is to guarantee that the mapping is performed at some stage during the call setup, and to guarantee the validity of the routing number. These rules are adapted to the "rnum" scheme:

Rule 1. The donor network must support the "rnum" scheme.

Rule 2. The validity area indicated for an "rnum" entry must contain the current serving network and the donor network

Rule 3. There must be a route from every network in the routing address validity area to the current serving network. A network must not route a call established with a routing number to a network outside the validity area of this routing number.

The fourth rule in section 3.2.1 defines where the mapping takes place. We adapt it to the "rnum" scheme by dividing it into five separate rules:

Rule 4. If the originating network supports the "rnum" scheme, it must perform an ENUM query for the originating call in the first exchange.

Rule 5. An exchange in a network supporting the "rnum" scheme must perform an ENUM query for all calls arriving from a neighboring network that is outside the validity area of *any* routing number valid inside the current network.

Rule 6. An exchange in a network supporting the "rnum" scheme must perform an ENUM query for all calls arriving from a neighboring network that does not support the "rnum" scheme.

Rule 7. A gateway in a network supporting the "rnum" scheme must perform an ENUM query for all calls arriving in the SCN network.

[2] In a strict sense, the validity area of an entry may exclude some area where the routing number is valid, with the consequence that this area is not considered in routing.

Rule 8. If an ENUM query returns an “rnum” entry with a routing number valid in the current (to which the call setup has proceeded) network, the call must be set up with this routing number. An “rnum” entry that is not valid in the current network must be ignored.

Although Rule 5 gives the impression to be difficult to implement, practical implementations are straightforward. Since the validity area of routing numbers coincide with regional or country borders, Rule 5 only affects calls arriving across these borders.

5.4 Implementation scenarios

Let us consider a hypothetical implementation of ENUM based number portability in Finland. The national routing number format is standardized [Ficora 1997, 1998, 1999, 2002]. Routing to ported numbers is implemented with a routing number starting with “1D” followed by an operator code and an operator-specific destination identifier. In the case of national number portability, where fixed subscribers may move within Finland, the number is valid in all networks in the country, i.e. support for this routing number format is mandatory in all networks. The RAVA therefore includes all networks in Finland. Another possibility would have been number portability within an area code only, but the principles are similar in that case.

We observe routing to a SCN destination in Finland. The destination is ported to another service provider, and calls are routed with a routing number built up by the identifier “1D”, the hypothetical operator identifier “88”, and an operator-specific code “19578345”. In compliance with Rule 3, all networks within the validity area have a route to “1D88”, i.e. to the current serving network. The corresponding entry in DNS is:

```
$ORIGIN 3.0.3.5.1.5.4.9.8.5.3.e164.arpa.
 IN NAPTR 1 0 “u” “E2U+X-rnum”
     “!^.*$!rnum:1D8819578345;valid=fi!”  .
```

In the considered scenarios, the donor operator has implemented number portability with the “rnum” scheme, so the mapping function validity area (MFVA) includes the networks supporting the “rnum” scheme. The MFVA must, at a minimum, include the donor network. It is important to notice that not all operators are required to implement number portability with the same scheme. Another operator may perform the mapping to the routing number using another method, provided that the rules for routing numbers are fulfilled. However, the efficiency of the “rnum” scheme grows when more networks implement it, since the mapping can be performed earlier before the call reaches the donor network.

For number portability within the SCN, we examine two different scenarios with different locations of the caller. Then we examine scenarios where the destination has been ported between the two technologies.

5.4.1 The caller is an SCN terminal

First, we examine SCN→SCN calls to terminals that have not been ported or only ported within the SCN. As defined in Equation 3.1, the call path consists of two path segments. The first (possibly empty) path segment consists of networks not belonging to the RAVA. The second path segment starts when the first network in both the RAVA and MFVA has been reached.

The first path segment consists of networks outside the RAVA; if the caller is inside the RAVA, the first segment is empty. The call is routed with the directory number. In a network supporting the "rnum" scheme, a DNS query is performed, but the obtained entry is ignored since the current network is outside the validity area. In a network not supporting the "rnum" scheme, no query is performed. The DNS queries can be suppressed if the current network can deduce from the directory number that the obtained entry will be invalid. For example, a network in Sweden does not perform a DNS query for directory numbers beginning with +358, since it is known that the two countries do not share routing numbers.

When the call reaches a network supporting the "rnum" and where the obtained entry is valid, the second path segment begins. From that point on the call is routed with the routing number to the current serving network. A call that has entered the RAVA does not leave it, and the second path segment consists of networks belonging to the RAVA. The mapping to the routing number will at last be performed in the donor network, which by definition supports the "rnum" scheme. Calls routed with a routing number do not trigger further DNS queries.

In Figure 5.1, the networks supporting the "rnum" scheme (i.e. inside MFVA) are marked with DNS servers. The call is routed with the directory number through networks 1, 2, 3 and 4 towards the donor network. Network 1 supports "rnum" but the obtained "rnum" entry is invalid since the network is outside the RAVA. Network 4 is both in the RAVA and supports the "rnum" scheme, so the mapping to the routing number is performed before the call reaches the donor network. The call is then routed with the routing number to network 6, which is the current serving network.

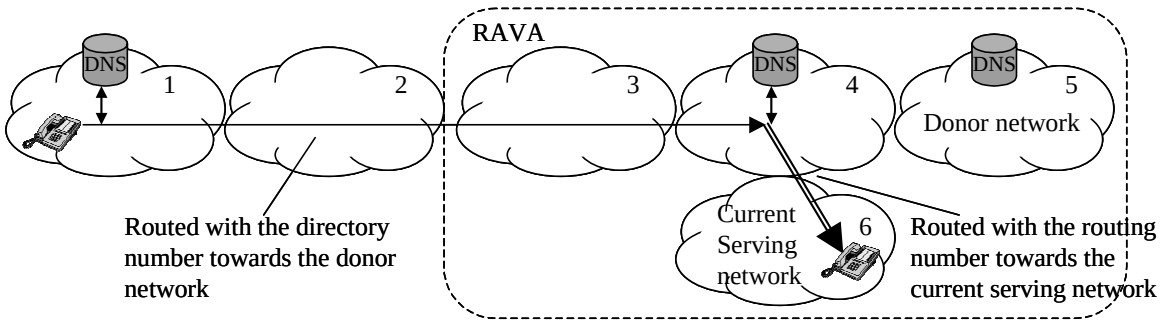

Figure 5.1: Call setup example with both caller and destination in the SCN

5.4.2 The caller is an IP terminal

Next we examine IP→SCN calls, where the number has not been ported or only ported within the SCN. The originating network performs a DNS query to obtain the ENUM entries. Since no recognized ENUM entry is obtained, the call is directed to a gateway. After that, the SCN portion of a IP→SCN call is similar to a corresponding SCN→SCN call originating from the gateway: the call is routed with the directory number towards the donor network, and the first network in the RAVA that supports the “rnum” scheme performs the mapping to a routing number.

However, since DNS is globally accessible, it is easy to implement the single query function (see section 3.4). An IP network may support the “rnum” URI format in addition to standard ENUM, and it may perform the mapping to a routing number already on the IP side. Then, instead of initiating the session to a “directorynumber@gateway” URI, it initiates the session to a “routingnumber@gateway” URI. It is important that the gateway connects to a network that is within the validity area of the routing number (requirement 2 in section 3.2.2). Otherwise, the call reaches a network that is unable to route the call. This requirement must be considered by the gateway location mechanism.

The IP network has the specific property, that the mapping to routing address (URI) is performed only once. When the E.164 number in the “sip” or “tel” URI has been translated to a URI containing a gateway or signaling server address, no further DNS queries are made. Therefore, if the originating network does not support the “rnum” scheme, the session is initiated with a “directorynumber@gateway” URI, and it is no longer possible to obtain the routing number in the same query. Any gateway is then valid, but the gateway selection is not necessarily optimal.

Even if a network that supports the “rnum” scheme does not perform a mapping, it may use the routing number information in selecting the best gateway.

5.4.3 Calls to a terminal ported between the technologies

For a number that has been ported from the SCN to the IP network, there is an ENUM entry (DN → URI) in DNS. Since the URI is a globally valid routing address, any call originating from the IP network will be optimally routed to the current location. However, a call originating from the SCN will be routed toward the donor network. The first network that supports ENUM (e.g. through the “rnum” scheme) will be able to determine that the terminal currently is in the IP network and the call will be routed to a gateway. At last in the donor network the call is routed to a gateway. From the gateway, the call is normally routed to the current location using the URI obtained from DNS.

The case where the number has been ported from the IP network to the SCN is more difficult. The previous mapping (DN → URI) in DNS has then been replaced with an “rnum” entry (DN → RN). The networks in the SCN still have routes for the prefix leading to a gateway.

We first examine the situation where the caller is an IP terminal (see Figure 5.2). The originating IP network is aware that the terminal is in the SCN, since the DNS query returns an “rnum” entry. Even if the IP network does not support the “rnum” scheme, it assumes that a terminal without a recognized ENUM entry resides in the SCN. Therefore, the call will be routed to a gateway with a “directorynumber@gateway” URI. The problem arises if the gateway leads to a SCN network that does not support the “rnum” scheme or that is outside the RAVA. This network tries to route the call back to the IP network through a gateway. A loop is formed if the gateway does not detect and correct this by dropping the call. In order to solve the problem, the selected gateway must be capable of routing the call correctly in the SCN. According to requirement 2 and 3 in section 3.2.2, the gateway must be in both the RAVA and MFVA of the current serving technology. A simple way to select a gateway fulfilling this requirement is to add a fixed “directorynumber@gateway” ENUM entry for a number ported between the technologies. However, more efficient gateway selection can be performed dynamically as discussed in subsequent chapters.

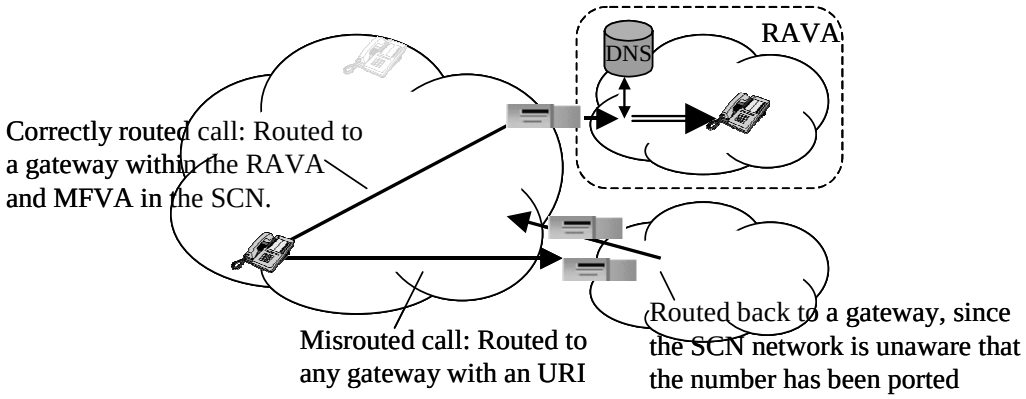

Figure 5.2: Routing to a number ported from IP to SCN.

A similar problem arises when the caller is an SCN terminal. If the originating network is outside the RAVA and/or the MFVA of the destination, the call is routed to the IP network though a gateway. If some network on the path between the originating network and the gateway is within the RAVA and MFVA, the mapping takes place and a more optimal route is selected; otherwise the call unnecessarily reaches the IP network. The call leg from the IP network to the destination is similar to the above, i.e. a valid gateway must be selected for the call to succeed. The unnecessary path through the IP network can only be avoided by enlarging the RAVA and the MFVA, or by some special “hacks” (e.g. a route for a single moved number).

5.5 Preference and multiple terminals

Hitherto, the order of processing entries has been determined only by the capabilities of the terminal or network. By using the order and priority fields of the NAPTR record, it is possible to have several terminals with the same number. For example, if a customer has both an IP telephone and a SCN telephone, the entry in DNS could be:

```
$ORIGIN 3.0.3.5.1.5.4.9.8.5.3.e164.arpa.
 IN NAPTR 1 0 “u” “E2U+sip”    “!^.*$!sip:nbeijar@sip.hut.fi!”     .
 IN NAPTR 2 0 “u” “E2U+X-rnum” “!^.*$!rnum:1D8819578345;valid=fi!” .
```

The entries are processed in the order given by the order field (lowest value first). For entries with the same value in the order field, the preference field determines the order (lowest preference value first). However, the order field *must* be followed, but the preference field *should* be followed [RFC 3403]. “Preference is used to give communicate[3] a higher quality of service to rules that are considered the same from an authority standpoint but not from a simple load balancing standpoint“ [RFC 3403].

If several “rnum” URIs are available, the client selects the first valid one according to the order field. If several entries are available for a given value in the order field, the client can try them in the order given by the preference field. However, if the client has tried call setup with an entry of a given order value, it must not try entries with other order values, even if the first fails.

What happens with routing if there are both IP and SCN destinations for the same number? In principle, the normal routing would be functional. The destination with better values in the order and preference fields would be used. In a successful call setup, this would be the same as if only one entry was given. However, if the preferred destination is unavailable, the next destination should be chosen. Without any signaling protocol assistance, there would be a risk of loops. This problem arises e.g. if both destinations are unavailable, and the networks of different technology start forwarding the call to each other. To solve the problem, the signaling protocol must transport the already traversed route or limit the number of hops. Currently, no such method is available in any common signaling protocol.

Because of this problem, we propose that the call should not be directed to any gateway if URIs for both technologies are available. If the call is set up from an IP network, only the IP URIs are considered and the “rnum” URIs are ignored. Correspondingly, an SCN exchange ignores the IP telephony URIs if any “rnum” URIs are present. The gateway is only used if it can be guaranteed that the call setup will not return to the same technology as the technology of the network from which it was set up.

5.6 Considerations

The solution essentially implements the functions of the number portability database with DNS. Thus, it performs the mapping:

$$DN \xrightarrow{DNS} RN$$

The main difference is that the DNS information is global. While it is possible to restrict the use of the database systems to a specific country or set of networks, all networks see the information

[3] Error present in [RFC 3403]

of DNS. Because of this property, it is necessary to limit the use of the entries to the validity area. Therefore, the validity parameter is crucial to the “rnum” URI.

The rules demand that both the donor network and the current serving network are within the routing number validity area. Number portability is thus only possible inside the validity area, which could coincide with country or region borders.

Only the donor network must support the “rnum” scheme. After the mapping has been performed, the networks on the path to the receiver do not make DNS queries, and therefore support of “rnum” is not required. Thus, the “rnum” scheme works even if it is used only in a few networks, but the donor operator must maintain entries for numbers moved out of the donor network.

It is an advantage that DNS can be used for number portability in both the IP network and in the SCN. Only one mapping needs to be maintained and there are no synchronization problems. Identical information about the moved number is available on both the IP and SCN.

Since the information is globally available, it is difficult to provide privacy. Routing numbers are visible to everybody. This problem is common to all DNS based systems.

5.7 Depth information for overlap sending

Signaling for most new technologies, such as mobile networks and IP telephony, transports the entire number of the called party in a single setup message (en bloc sending). The terminal first collects the digits composing the subscriber number, and when a particular “call” key is pressed, the setup message is sent. Nevertheless, overlap sending is still dominating in the PSTN. In overlap sending, the terminal sends the digits as the corresponding telephone keys are pressed, and there is no explicit signal for indicating that the entire number has been sent. Overlap sending allows exchanges to start routing before the number has been completely entered, but the network must determine when the dialed digits constitute a complete number.

In a solution based on DNS, overlap sending would require a separate DNS request after each received digit to examine whether the currently received digits match with an entry in the DNS server. This additional burden can be avoided by providing information about the expected number of digits to receive before the following query is needed. The number fundamentally indicates the length of the shortest branch of the number tree with the starting point indicated by the currently received digits. For instance, if the digits 12345 have been received, the number of expected digits is 2 if there exists an entry 1234567, although longer entries starting with 12345 may also exist.

The expected number of digits, i.e. the minimum remaining depth of the number tree, can be signaled together with the normal ENUM entries. Without specifying any specific format, the following is an example of one possible representation:

```
$ORIGIN 3.5.1.5.4.9.8.5.3.e164.arpa.
 IN NAPTR 1 0 “u” “E2U+rdepth”    “!^.*$!rdepth:2!”   .
```

The remaining depth is given in a new URI called “rdepth”. The motivation for using the E2U service to return a URI that does not specify any real location is to allow clients to obtain the remaining depth with the normal DNS query. The client only needs to query for NAPTR records using the E2U service. If no entries are obtained on this depth, the remaining depth URI indicates the number of digits required to be collected before the following query. It is obvious, that the “rdepth” URI is only returned when no matching entries have been found. An alternative to defining a new URI would be a new record type, but this alternative requires the client to request two types of records.

When the exchange has received an “rdepth“ entry from a DNS query, it collects the specified number of digits before performing the following query. If the caller does not enter enough digits, i.e. the time after the last received digit exceeds a timeout, the number is known to be invalid. Nevertheless, an implementation may try to perform an extra DNS query with the incomplete number.

The depth entries can be generated dynamically by a DNS server if that server knows all destinations beginning with the currently reached digits. This is true for the highest tier. The server examines its local database to find the remaining depth and replies with a dynamically generated “rdepth” record.

One should notice that the “rdepth” URI is an auxiliary feature, which reduces the number of unsuccessful DNS queries. A DNS server that does not support them will cause more load but will still be compatible. Similarly, a client can simply ignore “rdepth” URIs as an unrecognized URI format.

Depth information is not only useful in DNS applications on the SCN but also in standard ENUM. Normally, when a gateway receives a setup from the SCN using overlap sending, it must determine when the complete number is received. This is the same situation, and the number of queries can be reduced if the DNS server signals the remaining depth to the gateway with the “rdepth” URI.

5.8 Summary

In this chapter, we examined how DNS can be used for number portability both in IP networks and in the SCN. First, we saw that the effectiveness of number portability depends on the chosen implementation model for ENUM. We designed a URI for routing numbers on the SCN. With the URI, it is possible to obtain the routing number from DNS, which is an already functional

distributed global database. We defined the rules for using the URI. The existence of a separate routing number URI is motivated, since it performs a different function than the “tel” URI. We also elaborated on routing in the case when the ENUM information includes multiple terminals. Finally, we presented a method for signaling the minimum number of remaining digits in a dialed number.

Chapter 6
Gateway location and routing with DNS

As it currently seems, ENUM is being adopted in many countries. ENUM provides address mapping and some degree of portability for telephone numbers and services. As we have seen, it can easily be extended to support number portability on the SCN. Since it would be beneficial to provide as much functionality as possible with a single system, we examine whether it is feasible to use DNS for gateway location as well. Another reason is to search for alternatives to the rather heavyweight TRIP. In this chapter, we develop three different approaches that use ENUM for gateway location. The aim is to examine how additional information in DNS can be utilized for gateway location and routing – not to provide a final protocol definition. We compare the approaches and examine their applicability.

6.1 Approach 1: Number-specific gateway database

The idea of storing gateway addresses in DNS was proposed already in the early years of development of gateway location methods [Rosenberg 1998]. By that time neither ENUM nor TRIP had been developed. The NAPTR resource record had not been specified. Today, with the experience of TRIP and ENUM available, there may be more efficient methods of using DNS for gateway location. The first approach is based on the ideas in [Rosenberg 1998] but instead utilizing the NAPTR records of DNS.

With some additional definitions, it is possible to use DNS for gateway location in a way similar to ENUM. Information about several appropriate gateways can be attached to each destination in NAPTR records. The records contain the identifier of the applicable signaling protocol (sip, h323). In the simplest case, the information for a terminal in the SCN could look like:

```
$ORIGIN 3.0.3.5.1.5.4.9.8.5.3.e164.arpa.
 IN NAPTR 1 0 “u” “GW+sip”
      “!^.*$!sip:+35894515303@gateway1.provider.com!” .
 IN NAPTR 2 0 “u” “GW+sip”
      “!^.*$!sip:+35894515303@gateway2.provider.com!” .
 IN NAPTR 3 0 “u” “GW+h323”
      “!^.*$!h323:+35894515303@gateway3.provider.com!” .
```

These example entries give the URIs to two alternative gateways for SIP calls and one gateway for H.323 calls. The entries are returned when an ENUM query is performed to a destination that resides in the SCN.

A similar method can also be applied to gateway location in the SCN. Using the routing number URI described in Chapter 5, information for locating gateways to IP terminals can be given in the following manner:

```
$ORIGIN 3.0.3.5.1.5.4.9.8.5.3.e164.arpa.
 IN NAPTR 1 0 “u” “GW+tel” “!^.*$!X-rnum:1D1883457983;valid=fi!” .
 IN NAPTR 1 1 “u” “GW+tel” “!^.*$!X-rnum:3589923472834!”         .
```

These entries give the routing number URI to a number of alternative gateways when an SCN element performs an ENUM query to a number residing in the IP network. The same format and rules regarding the use of the URI apply. To provide global gateway location, one of the routing numbers should be globally valid. Routing numbers that are recognized in a smaller area, such as the numbers starting with “1D” used in Finland, can be included as a preferred alternative. However, the final decision about the use of routing numbers is determined by the regulators and operators.

6.1.1 Implementation

The gateway location application can be implemented using the Dynamic Delegation Discovery System (DDDS) [RFC 3401, RFC 3402, RFC 3403] as an application similar to ENUM. Each DDDS application must define the application unique string, the first well-known rule, the list of valid databases and the final expected output [RFC 3401]. The algorithm is illustrated in Figure 6.1.

For the gateway location application, these are similar to the corresponding ones of ENUM. The application unique string is the fully qualified E.164 number minus any non-digit characters except for the “+” character appearing in the beginning of the number [Faltstrom 2003]. The application unique string is the input to the first well-known rule [RFC 3402]. The first well-known rule is the identity rule, whose output is identical to the input. For the database lookup, the key is transformed to a domain name, using the same procedure that ENUM uses (reversing and inserting dots). The final expected output is a URI in its absolute form. The gateway location application (GW-LOC) uses the NAPTR records of DNS as the only database. The relationship between ENUM and the GW-LOC application is shown in Figure 6.2.

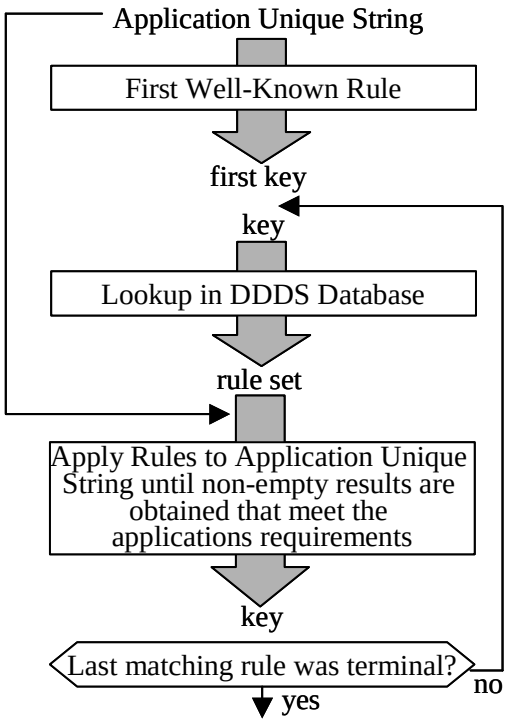

Figure 6.1: The DDDS algorithm [RFC 3402]

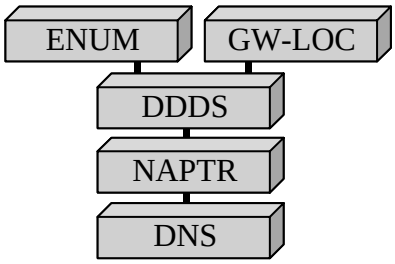

Figure 6.2: Relationship between ENUM and the gateway location application

For storing the information in DNS, a new service called "GW" is defined. The service parameters for the GW-LOC application take the following form, which is similarly constructed as the corresponding parameters of ENUM:

```
service_field = "GW" 1*(servicespec)
servicespec   = '+' gwloc-service
gwloc-service = type 0*(subtypespec)
subtypespec   = ":" subtype
type          = 1*32(ALPHA / DIGIT)
subtype       = 1*32(ALPHA / DIGIT)
```

The flag "u" is defined similarly to the "u"-flag in ENUM. This flag indicates that the rule is terminal and the result of the operation is a URI. The URI directs the call to the address of a gateway. Only gateway addresses are allowed in the URI. Multiple results are allowed and they

are examined in the order determined by the NAPTR order and preference fields. These indicate several available gateways, which have different preference. The gwloc-service identifier is identical to the enumservice identifier in ENUM. It identifies the application protocol. With this identifier it is possible to use different gateways for different application protocols, which may be necessary if IP telephony signaling does not converge to a single protocol. To enable gateway location for SCN→IP calls, there should also be a gwloc-service corresponding to circuit switched telephony. In the above example, the "tel" is an imaginary identifier for that purpose.

6.1.2 Adding hierarchy

So far, a list of gateways is specified separately for each destination. In practice, however, it is not feasible to manually specify a separate gateway address for every destination. The number of entries is the product of the number of gateways and the number of destinations. Furthermore, management of this data would be arduous. Nevertheless, we see it as a considerable alternative if the information is generated dynamically for each query. The information is then stored in a separate database and the ENUM entries are generated on demand. In this case, DNS is primarily used as a query interface.

The other problem is to define what happens if there are no records for a specific number. The paper [Rosenberg 1998] proposed that successive higher levels would be tried: first the exchange, then the area code and finally the country code. The assumption is that the gateway located in a particular area code is likely to provide the cheapest calls to the area code.

Whether or not the higher-level prefixes are successively tried, the solution can be improved by utilizing the hierarchy of DNS. Information about numbers is aggregated into information about prefixes. By defining the gateway for a prefix, several numbers use the same set of gateways and the solution scales considerably better.

In the hierarchical approach, we must be able to construct the URI by inserting the directory number into a pattern. This is easy to perform with the regular expression in NAPTR. In the following example, the directory number is inserted at the position of the "\1" marker:

```
$ORIGIN 1.5.4.9.8.5.3.e164.arpa.
 IN NAPTR 1 0 "u" "GW+sip" "!^(.*)$!sip:\1@gateway.provider.com!" .
```

The efficiency of the aggregated version decreases due to number portability. It is against the principles of DNS to delegate a prefix to another server while still maintaining information for some of the destinations in this prefix [Albitz 1997]. Consequently, if a number moves out from an aggregated prefix, each number in the prefix must be described independently. This is the same problem as described in Section 5.1, but with much more information about each number.

6.1.3 Considerations

The paper [Rosenberg 2002] pointed out that storing only the gateway address in the resource records is not enough. It would imply that clients must query all gateways separately to determine

its capabilities and cost, which does not scale. As a solution, it mentioned the “kitchen sink” resource record [Eastlake 1999], which can contain compact descriptions of protocol support, capabilities and cost structure.

In our solution, information about signaling protocols is already available. A SIP client will only receive addresses of SIP gateways. Negotiation of codecs and encryption mechanisms should be a function of the signaling protocol, not of routing. For that reason these functions are not part of TRIP, and we cannot see any reason to add it to the DNS-based solution. Cost has always been a problematic issue in gateway location. Currently TRIP leaves the issue open to be defined with proprietary attributes. As we see, cost information should not be included in DNS, but rather be obtained by other mechanisms.

Furthermore, the paper [Rosenberg 2002] described the problem of unbalanced trees. This arises, when no records exist under some exchange or area code, but the following level (e.g. country code) contains too many. It proposed an overlap solution, where a set of gateways are listed in several area codes, and a restriction solution, where the search continues in neighboring areas if the target area is empty. However, determining the neighboring areas requires additional information, such as a map. The overlap solution would lead to the undesirable situation where gateway providers list their gateways in as many as possible areas to increase business.

We rather see storing gateway information in DNS as an optional feature in addition to some traditional method. The gateways listed for a specific destination should be seen as a list of recommended gateways, and their use would be optional. Thus, an operator in an SCN network would be able to list the gateways directly connected to it, to let calls be transported in the IP network as long as possible. The maintenance of the DNS record for each number would be delegated to the operator serving the number.

6.2 Policy location

The previous approach leads to the situation, where the destination network selects the gateway. In this *destination-determined policy* model, the DNS entries corresponding to a SCN number are maintained by the operator of the number. This is the opposite situation compared to that when TRIP is used. Recall that in TRIP the operator only announces the gateways; the route is then selected by the location servers along the path, including the one in the originating network. In TRIP, the advertisement may be issued by any gateway able to complete calls to the destination.

Usually an *originator-determined policy* model is preferred. It allows the originating network to select the gateway – a method, which better complies with the policies of the operator and respects the user’s preferences.

It is difficult to implement originator-determined policy with the previously described approach. It would correspond to the situation where a network has different DNS information than its neighbor about specific destinations. ENUM is a global system, where the obtained information

should be independent of the location where it was obtained. Load sharing, e.g. for web servers, can be implemented with preference values in DNS. Preference can be given to nearest servers by rearranging the "A"-records according to measured round-trip time [Chao 1999]. These methods are not appropriate for gateway location. Load sharing would give equal preference independently of location, and the method of measuring round-trip times involves excess traffic overhead.

6.3 Approach 2: TAD information in DNS

Since it is beneficial to let the originating operator select the gateway, we take an approach where each operator can select the gateways to use independently of other operators. For this, we use administrative domain numbers, such as ITADs, instead of E.164 numbers. This implies that the DNS tree used for gateway location is no longer a part of the ENUM infrastructure. Instead, a new root is required. Let us use the imaginary root "tad.arpa" for describing the structure. We use the TAD identifier for identifying networks in a similar way as ITAD is used in the TRIP framework [RFC 2871], but with the extension that also SCN networks are included. A domain name is constructed by appending ".tad.arpa" to the string representation of the decimal TAD identifier. For instance, the TAD with number 1234 is mapped to the domain name "1234.tad.arpa".

This mapping allows us to enter information related to a specific administrative domain into DNS. The information may include the available gateways in the domain, and services provided by the domain. However, in this case we are interested in the gateways used by the domain for routing calls to SCN destinations. This information can be attached either directly under the TAD's domain name in NAPTR records for each destination, or under a subdomain indicating the destination network. The latter would use domain names in the format "destination.source.tad.arpa" as illustrated in Figure 6.3.

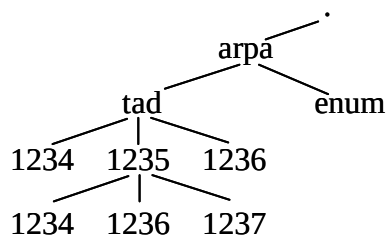

Figure 6.3: DNS hierarchy for TAD specific gateway location

The DNS entries represent the gateway addresses that a domain uses to reach all other domains. The size of the total database is therefore in the order $O(M\ N^2)$, where N is the number of domains and M is the average number of gateways per domain.

Contrary to the previous approach, this approach lets the originating network select the gateway for the call. It is easy to implement, and in the lack of an agreed root, such as "tad.arpa", the

information can be attached at any point in the DNS tree. However, it is doubtable if DNS is needed for this purpose. One can deem that the scope of DNS would be extended too far and that a global directory is not required for network specific functions. Further, the information is visible to everybody with DNS access, which is not usually desirable. This solution could equally well be implemented with any database, since no distribution between domains is needed. On the other hand, the solution allows for later addition of other information related to the network.

Because the total database size is too large and the DNS hierarchy is not actually used, we abandon this solution. The value of this approach is speculative and it leads us to the following approach.

6.4 Approach 3: Topology description with DNS

By redefining the semantics of the previous approach, DNS can be utilized for gateway location in a more efficient and general way. A structure similar to the one presented in Figure 6.3 can describe the domain topology of the hybrid network. However, contrary to the previous approach, the domain names have the format “source.destination.tad.arpa”. The difference is that the order of the source and destination parts is reversed. Given a destination TAD, it is thus possible to look up which TAD has a connection to it. Only directly connected domains are listed, which reduces the database size to a fraction of the above.

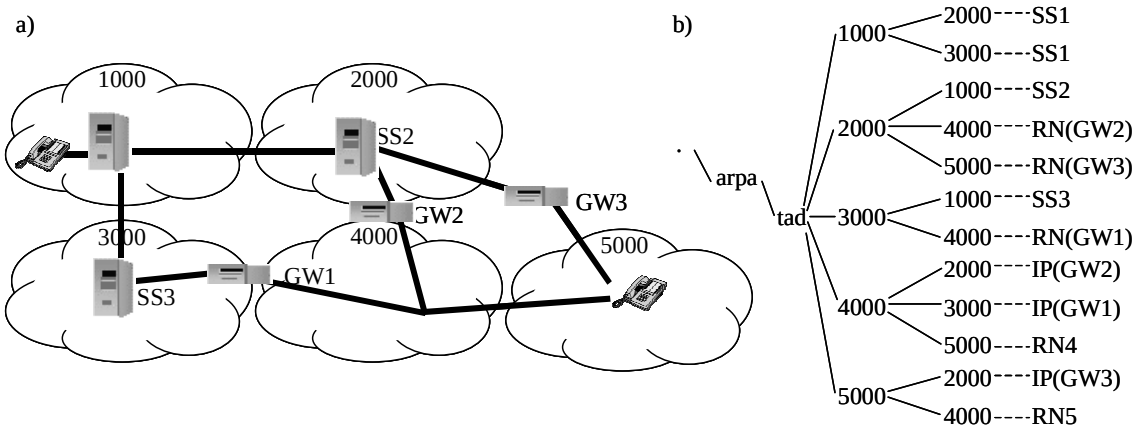

Figure 6.4: a) Example network topology b) The corresponding DNS hierarchy

In the example topology depicted in Figure 6.4, the domain with TAD identifier 4000 can be reached through the domains with TADs 2000, 3000 and 5000. For each destination domain, NAPTR records describe the available connections between the domains, and indicate if a gateway is used. The records for domain 4000 give the corresponding IP addresses of the gateways for calls received through domains 2000 and 3000. For instance, the domain TAD 2000 uses the gateway GW2 when a call is set up to TAD 4000. For calls received through domain 5000, a given routing number is used.

6.4.1 Routing algorithm

The setup procedure uses the information recursively. The input is the TAD identifier of the destination network and the output is a list of alternative next-hop addresses, from which the policy selects one. The procedure consists of a breadth-first search, where the list of alternative routes is built recursively.

Let us assume that a call is set up from a terminal in TAD 1000 to one in TAD 5000 in the above example network. The search starts by obtaining the entries of the destination network, which has TAD number 5000. One of the entries gives a gateway address, which is a usable next hop address. The other entry gives a path to a neighboring network (TAD 4000). When the recursion is continued, the records for TAD 4000 give two more gateways: the gateway GW2 connects TAD 2000 to TAD 4000 and the gateway GW1 connects TAD 3000 to TAD 4000.

For each recursion level, gateways farther from the destination and closer to the source are obtained. The originating network sets up the call to one of the gateway addresses. The gateway selection is thus performed according to the policies of the originating network. Caches are utilized in order to reduce the amount of ENUM queries. Cache entries can be long-lived since the topology is expected to be relatively static. In the recursion, paths to already visited TADs must be skipped to prevent loops. Loops are not possible if the calls are set up directly to a gateway and if the call does not return back to the IP network.

Going one step farther, a more precise path can be obtained. Let us say that the gateway GW3 was chosen in the previous example. The gateway connects domain TAD 2000 to TAD 5000. A continued recursion on TAD 2000 returns two gateways back to the SCN and the signaling server SS2, which should be used for calls arriving through TAD 1000. If the path was longer, the recursion could be continued. The originating network can set up the call either to the signaling server in TAD 2000 or directly to the gateway (skipping the signaling server). The choice is made by the originating network. If the call is set up through the signaling server, the signaling server performs the operation again, and continues call setup to another signaling server or to a gateway. In the example network, only the gateway is an alternative.

In this approach, special care has to be taken in order to prevent loops. No measures to prevent loops are provided by DNS. We propose two methods. The first method is that each signaling server on the path examines information provided by the signaling protocol about already visited networks. SIP provides this type of information. The second is by requiring the following network to be closer to the destination than the current network. The algorithm thus stops when is reaches the entry of its own network (in a breadth-first search). One should notice that loop prevention is only necessary when the next hop is a signaling server – not a gateway.

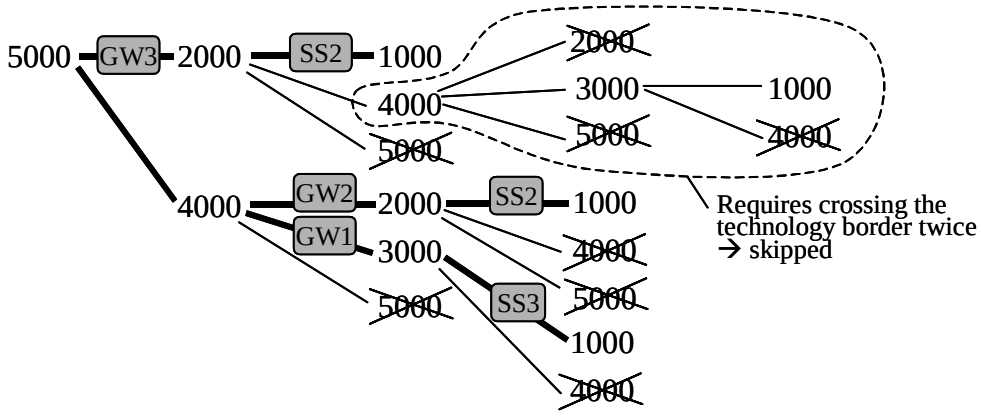

Figure 6.5: Tree view of the alternative routes between TAD 1000 and TAD 5000

The algorithm can be examined with a tree view as illustrated in Figure 6.5. The tree represents the memory structure used in the algorithm. The root is the destination network. The tree is built up by recursively obtaining the neighboring networks for each network in the tree. Networks that already exist on the path between the current node and the root are skipped. These are striked out with an X in the figure. The algorithm also skips networks that the setup signaling message has traversed. A branch that requires crossing the technology border twice is skipped or only used as a last alternative. Such a branch is marked with a dotted line in the figure. Recursion is not continued on branches that have reached the originating network. These represent valid paths between the originating and destination networks (thick lines in the figure). During the progress, the potential next hop addresses are stored. If the algorithm is performed by an IP telephony network, valid addresses are gateways and signaling servers but not routing numbers. The depth of the tree can be limited: when an adequate number of valid next hop addresses is found, the algorithm stops. Finally, one of the obtained next hop addresses is selected according to the preferences of the network performing the algorithm. The path length and the TAD numbers on the path can be used by the policy. The breadth-first search algorithm in pseudo-language is shown in Figure 6.6.

```
Find_next_hop_address(originating_tad, destination_tad):
  address_count = 0
  path = Get_traversed_networks_from_signaling_protocol()
  add (destination_tad, path, 0, none) to queue

  while queue is not empty
    get (current_tad, path, tech_count, serverlist) from queue
    if current_tad == originating_tad
      for each server in serverlist
        add server to found_addresses
        address_count = address_count + 1
    else
      add current_tad to path
      neigbors = DNS_request(current_tad)
      for each neighbor_tad in neighbors
        if not path contains neigbor_tad
          if address_count < MAX_ADDRESSES
            tech_change = Is_on_different_tech(neighbor, current)
            if tech_change
              tech_count = tech_count + 1
            if tech_count < ALLOWED_TECHNOLOGY_CROSSINGS
              server = DNS_address_between(current_tad, neighbor_tad)
              if address_is_a_valid_ip_address(server)
                add server to serverlist
              add (neighbor, path, tech_count, server) to queue

  return Policy_selects_one_of(found_addresses)
```

Figure 6.6: Pseudo-language algorithm for obtaining next-hop addresses

It is important to notice that any next hop address on the selected path can be used. In the example, the setup message can be sent to either GW3 or SS2 in the uppermost path alternative. However, because of firewalls or charging mechanisms (usually controlling firewalls), the call must in some cases pass the following next hop address (the inbound signaling server) on the path as seen from the current network. A network that forces calls to pass the inbound signaling server should flag this in the DNS information. Let us call them *mandatory servers*.

Gateway location is also possible for SCN→IP calls. A similar recursion is performed, and a gateway is selected according to the policies of the originating network. The routing number for setting up a call to the gateway is obtained. The routing number must be a globally routable number such as the MSRN in GSM, which does not define the path through intermediate networks in detail. Only one DNS query is required and a routing number in this format should not trigger new queries.

6.4.2 Implementation

The service is implemented using NAPTR [RFC 3403] resource records, which allow a regular expression to be used for generating a URI. The input to the rewrite operation is the E.164 number of the destination. We define a new service called “GW”, which is used as in the earlier described approaches. The service parameter has the following format in ABNF [RFC 2234] notation:

```
service_field = “GW” 1*(gwservice)
gwservice     = “+” type 0*(subtype)
type          = 1*32(ALPHA / DIGIT)
subtype       = “:” 1*32(ALPHA / DIGIT)
```

The format is similar to the one of ENUM. The gwservices are identical to the corresponding enumservices. Example types of gwservices are “sip”, “h323” and “message”. An example subtype is “voice”. The result of the operation is a URI, as indicated by the “u” flag.

The records for TAD 4000 in the previous example could look like the following:

```
$ORIGIN 4000.tad.arpa.
2000 IN NAPTR 1 0 “u” “GW+sip”   “!^(.*)$!sip:\1@sipgw.tel.com!”   .
2000 IN NAPTR 1 0 “u” “GW+h323”  “!^(.*)$!h323:\1@h323gw.tel.com!” .
3000 IN NAPTR 1 0 “u” “GW+sip”   “!^(.*)$!sip:\1@sip.tel.com!”     .
5000 IN NAPTR 1 0 “u” “GW+X-rnum” “!^+(.*)$!rnum:2D88\1!”          .
```

These records describe the addresses of the signaling gateways for setting up calls to destinations in TAD 4000. Two gateways connect TAD 4000 to TAD 2000: one using H.323 and one using SIP. Different routes are formed for calls established with different signaling protocols. Multiple gateways can be given for redundancy as well. The rewrite operation inserts the E.164 number of the destination into the URI, which identifies the next hop address. For call setups arriving from the neighboring SCN domain, a routing number is used. The routing number is generated in a similar way by adding a prefix to the directory number with a regular expression. An initial “+” sign can be removed by the regular expression.

6.4.3 Mandatory servers

As previously mentioned, a domain may require a signaling server to be used for calls routed through it. These domains must be able to flag this property. The flag field of NAPTR is suitable for this purpose, and we can define an additional NAPTR flag “m” for mandatory next-hop addresses. In the algorithm, all next-hop addresses that are on the path between the domain with the “m” flag and the destination must be ignored. Consequently, calls are not routed to a next hop address that is located after the domain that flagged a mandatory signaling server. If the specific path with an “m”-flagged server is chosen, the call can be set up to the mandatory server or to an address before the mandatory server on the path. However, a server behind a mandatory server can only be used if it is part of another path with no preceding mandatory servers. Mandatory servers must make a new DNS query to obtain the following address on the path.

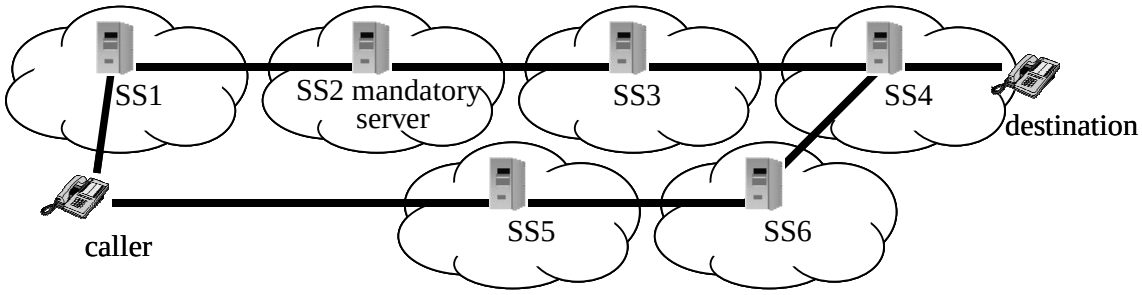

Figure 6.7. Scenario with a mandatory server

In the example in Figure 6.7, the call can be set up to SS1 and SS2. It cannot be set up to SS3 since it is between the mandatory SS2 and the destination. However, although SS4 is between SS2 and the destination, it can be used. The reason is that it is also a part of another path, which does not contain mandatory servers. All servers (SS5, SS6 and SS4) of the second path are valid next-hop addresses.

To detect all mandatory servers, the whole path between the caller and the destination must be examined. If the algorithm is stopped after a given number of found next hop addresses, some mandatory server may remain undetected. If the algorithm in the above example stops after finding 3 next hop addresses (SS4, SS3, SS6), the server SS3 may be chosen as the next hop without establishing the call through SS2. In practice, mandatory servers are not likely to be needed in transit networks, but only in the end network or in the network whose gateway is used. In that case, the limitation is not serious. In other cases, the algorithm is required to perform an exhaustive search to find a complete path between the originating and destination networks.

Mandatory servers are also possible for SCN networks, though not as useful. The corresponding flag is set and calls using the specific path are first set up to the mandatory routing number given in the entry. When the call setup reaches the intermediate network, a new DNS query is performed and the following routing number is obtained. Since there are no real benefits of this, the use can be expected to be minimal.

6.4.4 Scalability

The total database size is rather small since each operator only adds a few entries. If there are N operators and each operator has an average of M gateways, the database size is in the order of $O(M\,N)$ entries, which is much smaller than the $O(M\,N^2)$ of the previous approach. The database is small enough to be cached, in particular because only the frequently used entries need caching. The factor M is the product of the average number of neighboring domains (D) and the average number of gateways per neighbor relationship (G), thus $M = DG$.

The average number of DNS queries per call is $Q = D^{L-2}$, where L is the average number of domains on the path of an IP→SCN call. Thus, when the originating network is directly connected to the destination network (2 domains on the path), then $Q = D^0 = 1$ query. The

number of queries can be reduced by limiting the maximum allowed L to the lowest value $\hat{L}$ that guarantees that at least one gateway for every destination is found. In real networks, L is likely to be between 3 and 4 and $\hat{L}$ is likely to be about 5. For example, the values $D = 8$ and $L = 3.5$ give $Q \approx 23$ queries on average. It is important to notice that most of the queries are answered by the cache, which at a minimum contains the entries of the neighboring networks. For long paths, $Q < D^{L-2}$ since the algorithm skips networks that appear several times. If the queries are performed sequentially, the number of query rounds equals the number of queries, i.e. $Q_R = Q$. However, the queries can be performed in parallel, so that the entries for TADs at the same distance from the destination are obtained simultaneously. Then, the number of query rounds is proportional to the path length: $Q_R = L - 1$.

6.4.5 Considerations

The described approach routes calls in a way that resembles TRIP, and in particular the TRIP/CTRIP combination. The difference is that in this solution the information is global. While the policies in TRIP only select one path to each destination, this solution gives all the available paths. In TRIP, the policies of all intermediate networks contribute to the final routes. In the DNS-based solution, all paths are known, and the used path is completely determined by the originating network. Although it does not allow powerful policies in the intermediate networks, it has other considerable advantages: it is very lightweight and it generates alternative routes. The main disadvantage of TRIP has been its heaviness and convergence problems, and the presented solution solves both these problems. Additionally, the mandatory servers correspond to servers that modify the Next Hop attribute in TRIP.

Although we used DNS in this description, any other globally accessible database would be usable. Only two levels of the DNS hierarchy are used for the actual operation. The advantages of DNS are that the database can be distributed and that the infrastructure is available. Each branch of the information is maintained by the responsible operator, which simplifies maintenance. It also responds to failures: if a network is inaccessible, also the DNS information is inaccessible after the cache timeouts, and routes through the failing network are not detected.

6.5 Mapping directory numbers into TADs

Both approach 2 and approach 3 require that the originating network is aware of in which network the destination resides. Given a directory number, the TAD identifier is required. This mapping can be performed with a database, but it can also easily be performed with DNS, which allows a single widely adopted system to be used for both mappings. One possibility would be to add a record for each number. However, in the case where the number is attached to several terminals in different networks, a parameter is more suitable than a record.

Let us for illustration define a new parameter called “tad”. The format of the number is:

```
tad_parameter  = “tad=” tad_identifier
tad_identifier = 1*(DIGIT)
```

Adding the parameter to the previous example records gives:

```
$ORIGIN 3.0.3.5.1.5.4.9.8.5.3.e164.arpa.
IN NAPTR 1 0 “u” “E2U+sip”
     “!^.*$!sip:nbeijar@sip.hut.fi;tad=2000!”   .
IN NAPTR 2 0 “u” “E2U+X-rnum”
     “!^.*$!rnum:1D8819578345;tad=1000!”        .
```

The “tad” parameter is only necessary for calls requiring gateway location. Since the “rnum”-type is specified only in this work, it is still possible to modify it to make the “tad” parameter mandatory. Thus, we can use the above procedures to obtain a gateway for all destinations with “rnum”-type information in DNS. For destinations in the IP network, the location of the next hop server is optional. The above procedures can only be used for the destination, for which the current TAD identifier is available. This should not limit the applicability of the described solution. Standardization of an additional parameter to “sip” and “h323” URIs requires the corresponding URI schemes to be updated.

6.6 Comparison

The main difference between the above approaches is that in the first approach, the destination network selects the gateway, and in the following two approaches, the originating network selects the gateway. Usually the originating operator wants to control the gateway selection, which limits the usability of the first approach. However, the gateways listed using the first approach could be seen as recommended gateways. These gateways can be assumed to be closely located to the destination operator. Therefore, an IP→SCN call travels most of the distance in the IP network, which is desirable when the cost of IP calls is lower. In gateway location for SCN→IP calls, the call travels most of the part on the SCN, which is not usually as desirable, except for quality reasons. The approach can be described with the mapping:

$$DN \xrightarrow{DNS} IP_{gw}$$

The second approach is functionally similar to service location protocols, such as SLP. DNS does not add any noteworthy benefit and it is not motivated to use DNS for this purpose. Each operator can use whatever database for storing the gateway information and distributing it within its network. Therefore, we only described it for a conceptual purpose. The approach can be described with a chained mapping, where the first mapping returns the TAD identifier and the second one performs gateway location:

$$DN \xrightarrow{DNS} TAD \xrightarrow{DNS} IP_{gw}$$

The third approach solves the problems in the first and second approaches. It allows the originating network to choose the gateway, and simultaneously lets the other networks advertise the gateways in their networks. It scales well, since each operator only lists the connections to neighboring networks. One drawback is the high number of DNS queries required, but most of

these can be answered by the cache. Another drawback is that it requires the majority of the networks to use it to be efficient. However, groups of networks can share gateway information as envisioned in the TRIP framework [RFC 2871]. Furthermore, the mapping from a E.164 number to a TAD identifier requires one extra round-trip. The approach can be described with a chained and recursive mapping:

$$DN \xrightarrow{DNS} TAD \xrightarrow{*DNS} IP_{gw}$$

One should notice that the approaches are not mutually exclusive. An operator could participate in the third approach and additionally announce the recommended gateway using the first approach. Further, it is worth noticing that the mapping from directory number to TAD can be implemented with other methods equally well – using DNS is an attempt to accomplish as much as possible with a single method.

6.7 Summary

In this chapter, we developed three different approaches to using DNS for gateway location. The advantage of using DNS for gateway location is that it integrates well with ENUM, which is becoming the de facto standard for endpoint location in IP networks. The other main reason it that it allows a more lightweight approach to gateway location than TRIP provides. The approaches can be divided into originator-determined and destination-determined policy approaches. The destination-determined policy approach is simpler but we see them less useful from policy point of view. The originator-determined policy approaches are suitable for real networks, where the originating network can determine the path of the call.

Finally, we developed a method for mapping E.164 numbers to their corresponding network (TAD) identifiers, which is necessary in both the originator-determined policy approaches.

Chapter 7
Number portability with TRIP and CTRIP

The aim of the TRIP and CTRIP protocols is to locate a suitable gateway depending on the properties of the call and the gateway, while observing the policies of the operators. Because TRIP and CTRIP are routing protocols, they can also be used to route calls in the more general case where no gateway is involved. By integrating both the protocols, calls can be routed seamlessly across the technologies. However, number portability may cause a scalability problem, since each individual ported number may requires separate entry. In this chapter, we examine the scalability of these protocols in scenarios with number portability. We examine methods for solving the scalability problem: aggregation and combinations with another protocol.

7.1 The TRIP protocol

In order to automate the selection of gateways for calls from the IP network to the SCN, IETF has developed a protocol named Telephony Routing over IP (TRIP) [RFC 3219]. The protocol is essentially a routing protocol based on the Border Gateway Protocol version 4 (BGP-4) [RFC 1771]. Contrary to traditional routing protocols, TRIP operates on the application layer, and creates routes for IP telephony sessions. Routing information is distributed between Location Servers (LS), which maintain databases with routing information used in call setup. The information includes a next hop address, which points to the following signaling server on the route or alternatively to the gateway connecting the call to the SCN. In SIP, the signaling server is a proxy server. During call setup, the signaling server queries the location server to obtain the address of the next signaling server to forward the call setup to. In a call to the SCN, the last next hop address on the route is the address of a signaling gateway.

Similarly to BGP-4, TRIP uses policies in route selection. On receiving several advertisements for the same number prefix, TRIP chooses the one with the highest priority according to the administratively defined policy function. The chosen advertisement is forwarded to the neighbors, which perform a similar selection. Thus, the routes follow the policies of the intermediate domains and the path-determined policy model is thus used.

The priority is calculated from any of the attributes that TRIP distributes. The most central attribute is the list of intermediate domains, named Routed Path. Using this, an administrator may prefer routes through certain domains and avoid routes through other domains. The attribute also

gives the path length, which can be used to favor shorter paths. Other usable attributes include the next hop server address, which contains the next hop domain; the advertisement path; the multi exit disc, which specifies relative priority for several connections between two domains; and the converted route, which indicate that the application protocol is converted at some point on the path. A complete list of well-known[4] attributes is given in Table 7.1. Furthermore, it is easy to define new attributes, which for instance indicate the cost or capacity of the route. These optional attributes are equipped with flags, which specify how they are handled by an implementation that does not recognize them.

The structure of a TRIP node is shown in Figure 7.1 [RFC 3219]. The arrows show the information flow within the node. The node consists of four types of databases, called Telephony Routing Information Bases (TRIB). The *Adj-TRIBs-In* databases contain unprocessed routing information that has been received from other peers. These routes are available as input to the decision process. Routes from each external and internal peer location server are maintained independently in the database. The *Ext-TRIB* contains the preferred route for each destination, which has been selected by the route selection algorithm. The *Loc-TRIB* stores the local routing information that is selected by applying the local policies to routes from Adj-TRIBs-In and Ext-TRIB. The *Adj-TRIBs-Out* store routing information that is selected for advertisements to external peers. There is one Adj-TRIB-Out for each peer.

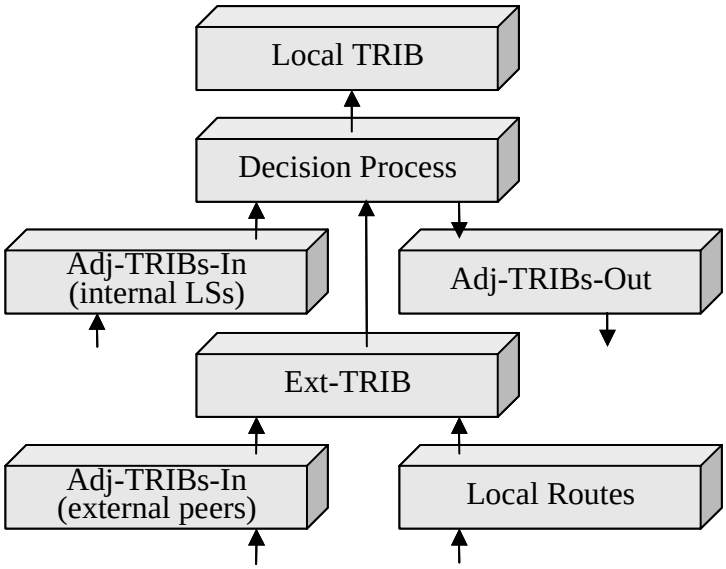

Figure 7.1: Structure of a TRIP node

[4] Well-known attributes are the attributes that must be supported by all nodes.

Table 7.1: Well-known attributes of TRIP

Attribute	Fields	Description
Withdrawn Routes	List of routes: - Address family - Application protocol - Address	Describes routes becoming unreachable. Used to withdraw previously advertised routes.
Reachable Routes	List of routes: - Address family - Application protocol - Address	Describes routes becoming reachable. The attribute contains prefixes reachable through the advertising domain. Works as a key for the rest of the update message.
Next Hop Server	Next hop ITAD Server	The ITAD identifier and server address of the next hop on the route.
Advertisement Path	List of path segments: - Type - List of hops: - ITAD identifier	Indicates the path that the advertisement has traversed. Mainly used to prevent loops.
Routed Path	List of path segments: - Type - List of hops: - ITAD identifier	Indicates the path that the signaling messages will follow. Routed Path is a subset of Advertisement Path, since it only includes the hops that have modified the Next Hop Server attribute.
Atomic Aggregate	(no fields)	Indicates that the path may contain ITADs not included in Routed Path due to aggregation.
Local Preference	Preference value	Used between internal peers to tell the preference of the route to other location servers in the ITAD.
Multi Exit Disc	Preference value	Used between external peers to specify the relative preference for routes received over one link compared to routes received over other links.
Communities	List of community values: - Community ITAD - Community ID	Used to group destinations sharing some common property so that the routing decision can be based on the identity of the group.
ITAD Topology	List of TRIP identifiers	Used within the ITAD to create the internal topology map of the ITAD.
Converted Route	(no fields)	Indicates that some location server on the path has changed the application protocol field in the Reachable Routes attribute.

At a general level, TRIP performs a mapping $X \xrightarrow{TRIP} IP$, where X can be any type of identifier defined as an address family. The currently defined address families are E.164 directory numbers (DN), and decimal and pentadecimal[5] routing numbers (RN).

7.2 The CTRIP protocol

Gateway location is equally essential for calls from the SCN to the IP network. Currently, static routes are used to direct this type of calls to an administratively chosen gateway. However, the management load of maintaining such routes is high, and the set of routes may vary dynamically due to congestion and call-specific requirements. Therefore, a protocol similar to TRIP has been developed for the SCN. The protocol, named Circuit Telephony Routing Information Protocol (CTRIP) [Kantola 2001, Beijar 2002] is similar to TRIP but with a new set of attributes. The most fundamental difference is that the next hop address is given as a routing number, which is a necessary change for being able to use the protocol in the SCN. The routing number is generated from the directory number using a regular expression. An alternative method allows queries to obtain the number from an external database. This method can for example request the routing number to a mobile subscriber from a home location register (HLR). Another major difference compared to TRIP is that information about the network type of every domain on the route is given. Using this information, the policies can prefer routes without a gateway or with only one gateway on the path. The quality can then be improved.

Generally, CTRIP performs the mapping $X \xrightarrow{CTRIP} RN$, where X can be any address defined in an address family. CTRIP has the same address families as TRIP.

The power of CTRIP is in the symmetry with TRIP. By using similar protocols in both the SCN and the IP network, routing between the technologies is seamless. The routes are created on the application level, and the SCN and IP networks are only underlying technologies from the viewpoint of the application-layer.

The TRIP and CTRIP protocols exchange information through a numbering gateway. The numbering gateway is a logical entity that converts the information between the two protocols and inserts the address of the signaling gateway. In this way, routes to destinations on the other network technology are automatically generated. For each SCN destination, the CTRIP entry is converted to a TRIP entry for distribution in the IP network; and the path of the corresponding TRIP entry contains the gateway. Thus, by defining the prefixes corresponding to terminals in a specific domain as local routes, all other domains will receive advertisements describing routes to these prefixes. All domains are able to obtain at least one route to each prefix. The policy functions of the domains on the path contribute to selecting one of several available

[5] Separation between decimal and pentadecimal routing numbers is due to different aggregation characteristics.

advertisements. This solution modifies the rationale of network management from defining static routes for each prefix to defining the policy function of the domain.

The combination of the TRIP/CTRIP protocols can be viewed as a pair of mappings:

$$DN \xrightarrow{CTRIP} RN$$

$$DN \xrightarrow{TRIP} IP$$

7.3 Influence of number portability

TRIP was not designed with number portability in mind. The protocol is fundamentally a routing protocol, and it requires aggregation to reduce database sizes. A TRIP node requires several databases. For each peer, there is a database for incoming routes (AdjTRIBsIn) and a database for outgoing routes (AdjTRIBsOut). Additionally, the node contains a database with local routes for the domain (LocalTRIB), an intermediate database storing the selected external routes (ExtTRIB) and a database storing the selected routes used within the domain (LocTRIB). The databases contain overlapping entries, so the actual database size can be reduced by using pointers. Regardless of pointers, however, the total storage requirement grows extensively due to number portability. Generally, every moved number requires an additional entry. Aggregation is not effective for single moved numbers, as will be proved later in this chapter.

More important than the database size is the processing power. When a new route has been received in one AdjTRIBsIn, the routes need to be recomputed. The route computation is performed using a multiphase decision process. To reduce the computational load, the decision process is only started if a specified time interval has elapsed since the last run. The decision process first calculates a preference value for each entry, which is a lightweight operation since it only needs to be done once when a new route is received from a peer. The heavy load is due to the comparison of all entries in the AdjTRIBsIn for each prefix – an operation necessary to find the route with the highest preference. This is performed in two stages: first combining all external AdjTRIBsIn databases and the LocalTRIB to the ExtTRIB database, and then combining the internal AdjTRIBsIn databases and the ExtTRIB to the LocTRIB database. Finally, the AdjTRIBsOut databases are generated from the LocTRIB. Apparently, this is a time-consuming process, which needs to be performed periodically when an incoming route changes. In practice, incoming routes change regularly due to route flapping [Labovitz 1999], which further increases the load.

7.4 Scalability evaluation

7.4.1 Scalability without number portability

In a case without number portability, TRIP/CTRIP requires one entry for each route[6] to a prefix assigned to a network. For example, in Finland there are currently 6166 prefixes distributed to different operators in different areas [Rostela 2002]. Assuming no aggregation and a separate route to each prefix, there will be 6166 entries in each TRIP/CTRIP node. This is schematically illustrated in Figure 7.2a, with a lower number of prefixes.

However, within one area code, routes are only necessary to the prefixes within the area. For each other area code, the prefixes can be aggregated into a default prefix for that area. The number of entries is then the number of prefixes in the area plus the number of other areas. In the 09-area in Finland, there are 536 prefixes [Rostela 2002]. The number of other areas is 13 - 1 = 12 [Ficora 2001]. Consequently, the number of required entries is 536 + 12 = 548. This is schematically illustrated in Figure 7.2b. This type of geographical or topological aggregation can also be continued in areas smaller than the area codes.

The individual prefixes could further be statistically aggregated, so that 10 sequential prefixes belonging to one operator are aggregated into one prefix. In practice, the impact of this is negligible if prefixes are distributed to operators rather randomly.

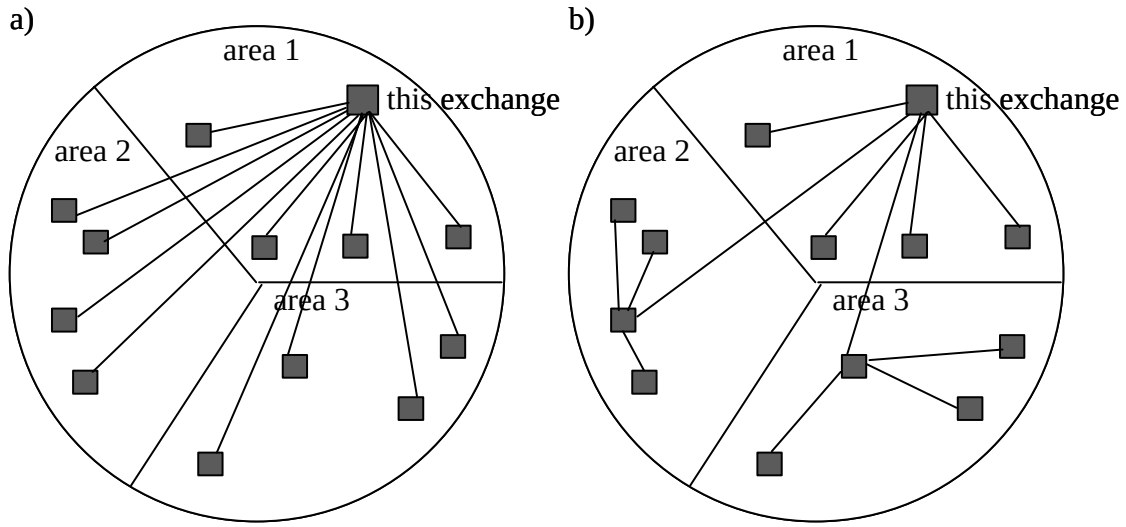

Figure 7.2: Routes to each prefix versus routes to prefixes in the same area plus routes to other areas

Routing to international numbers can be implemented with TRIP/CTRIP. Generally, this requires one additional entry for each country code. International routing is, however, likely to remain using static routes, and we will not consider these entries. The number of country codes is

[6] Generally, TRIP/CTRIP allows only one route for each prefix, but there may be a separate route for each application protocol.

relatively low and number portability is currently not possible across country borders. The impact of the international entries is therefore negligible.

The existence of two technologies does not influence significantly on the number of routes. A prefix can be assigned to an IP network in a similar manner as it is assigned to an SCN network. Since the number of subscribers does not increase due to IP telephony (except for the possibility to have both an IP and an SCN terminal), the number of prefixes in a stable hybrid SCN/IP–scenario is equal to the current number of prefixes. During the transition, the number may be higher; and in the worst case, each operator has both an IP and an SCN network, which effectively doubles the number of entries.

7.4.2 Scalability with number portability

When number portability is taken into consideration, the number of entries increases significantly. Each ported number adds one entry because a separate route is required for this number. If all numbers of a company move, there is a single entry for all numbers, providing that they have the same prefix. Because of the relatively small influence on the results, we assume that individual numbers move.

In Finland, there are 2.8 million subscribers to fixed telephony service [Liikenneministeriö 2000]. If 20% of the numbers move, there are $0.20 \cdot 2.8 \cdot 10^6 = 250\,000$ new entries. This number is two to three orders of magnitude higher than the number of fixed entries. It is obvious that the efficiency of the solution is purely dependent on number portability. Currently number portability in Finland is only possible within one area, which reduces the number of entries to a fraction of the above value. However, it is important to allow countrywide number portability in the future. One must also consider that number portability areas in other countries, such as in the U.S., may be several times larger than an area in Finland.

Aggregation is suggested as the means for reducing information amount. Aggregation of telephone numbers and routing numbers is much less efficient than aggregation of binary IP-addresses. In TRIP, aggregation of telephone numbers requires 10 sequential entries with similar properties to reduce to one entry, whereas binary aggregation only requires two entries. Routing numbers are usually pentadecimal, meaning that 15 sequential entries are required to produce one aggregated entry.

We can estimate the efficiency of number portability by examining the number space density. The number space density, δ, is the fraction of the allocated numbers of the total available numbers. In Finland, fixed telephone numbers are on average nine digits long, including the area code. Assuming fixed-size numbers, the number space size is $N_{tot} = 10^9$ numbers. With $N_{alloc} = 2.8 \cdot 10^6$ allocated subscribers, the density is $\delta = N_{alloc} / N_{tot} = 0.0028$. The density can be interpreted as the probability that a given number is allocated, assuming uniform distribution.

With 20% of the numbers moved[7], the density of the moved numbers is $\delta_{moved} = 0.20 \cdot N_{alloc} / N_{tot} = 0.00056$. Only moved numbers needs to be aggregated, since non-moved numbers are already included in the prefix assigned to the network. Therefore, 10 sequential moved numbers with the same attributes are required. Unassigned entries must be considered as belonging to the owner of the prefix (see the discussion below). The probability of 10 particular sequential entries is $P_{seq10} = (\delta_{moved})^{10}$. In the case with 20% of the numbers moved, this probability is $P_{seq10} = 3 \cdot 10^{-33}$. With several different operators, the probability that these have similar attributes is practically zero. Thus, this basic aggregation method does not help reducing information of non-topological numbers.

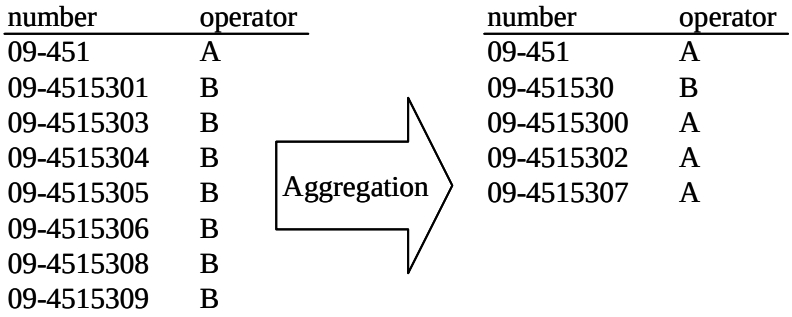

Figure 7.3: Example of longest-match aggregation

By using longest-match aggregation, better results are obtained. If more than half of the numbers in a given prefix have separate entries and similar attributes, these can be aggregated. In the example depicted in Figure 7.3, seven numbers have moved out of the prefix 09-451, which originally was assigned to operator A. Then, assuming normal telephone numbers, more than half of the numbers of the prefix 09-451530 have been ported. Thus, it is more efficient to have an entry for the prefix 09-451530 with separate entries for the remaining numbers than to have an entry for each of the seven numbers. With 20% numbers moved, the probability of more than five sequential entries is $P_{seq5+} = (1-\delta_{moved})^4(\delta_{moved})^6 + (1-\delta_{moved})^3(\delta_{moved})^7 + (1-\delta_{moved})^2(\delta_{moved})^8 + (1-\delta_{moved})(\delta_{moved})^9 + (\delta_{moved})^{10} \approx 3 \cdot 10^{-20}$. Considering that these must have similar attributes, neither does this provide any considerable improvement in aggregation efficiency.

The reason for obtaining low efficiency is due to the fact that TRIP requires that an entry is valid for all numbers within its range. Let us show the need for this requirement using an example (Figure 7.4) where we don't apply this requirement. Originally the prefix 451 has been assigned to Network A. Later, three numbers have been ported to Network C. Network C aggregates the entries (45161, 45167 and 45169) into a single entry (4516) and advertises it to its neighbors. Network B receives the advertisement for the aggregated entry from Network C and the original entry (for 451) from Network A. According to the longest-match rule, Network B sets up calls to

[7] Only numbers ported to another networks are considered. Numbers moved back to the donor network are not included.

the prefix 4516 through network C, and calls to other numbers within the prefix 451 through A. The problem occurs when a call to number 45162 is set up. This number still resides in network A but the call will be set up through network C. Thus, the problem must be solved by allowing aggregation of moved numbers only if all numbers within the aggregated prefix are in the network. This requirement is similar to the one of e.g. CIDR. In the example, aggregation is only possible if all numbers starting with 4516 really are in Network C.

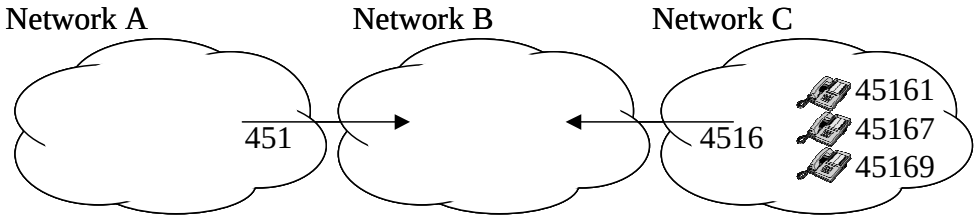

Figure 7.4: TRIP/CTRIP aggregation without exclusion of unknown numbers

The effect of unassigned entries is similar. We show this with a counter-argument: “unassigned numbers can be included into any aggregate”. If Network C in the above example knew that all other numbers within the prefix 4516 were unassigned, it could create an aggregated entry for the prefix 4516. The first problem is how Network C knows that all included numbers are unassigned, since the received advertisement for the prefix 451 does not tell anything about the individual assigned numbers. Anyhow, if this knowledge was available, the aggregated prefix 4516 could be created and routing to all assigned numbers would work correctly. However, if later Network A, who owns the prefix 451, assigned a number in the prefix 4516, this number would be unreachable. There are two equally bad ways to correct this problem. The first is to inform Network C to stop advertising the aggregated prefix, which would require some signaling mechanism. The second is to itself advertise the newly assigned number with a longer prefix, again hoping that this prefix does not include any assigned terminals of network C. Thus, the core problem is that a network cannot know which individual numbers in other networks are assigned and when assignments change. In conclusion, ported numbers can only be aggregated if all numbers in the aggregated prefix are included.

Regardless of the low aggregation efficiency, better results are obtained for longer paths. Numbers with the same next hop address can be aggregated if the other attributes are similar enough. Numbers that are reached through the same transit network have similar next hop address if a signaling server of the transit network is the next hop. Yet, information is always lost in aggregation, and especially the aggregated form of the path attributes may contain inaccurate information. An aggregated path attribute contains the path segments of all its elements, even though a particular number does not include a given path segment. When this occurs in the advertisement path, the advertisement does not reach the network of which some of the aggregated routes have traversed.

It is important to notice that the aggregation problems are serious only when TRIP/CTRIP is used as a routing protocol. The original intention, gateway location, has more relaxed requirements. All gateways are able to complete calls to all destinations, so an indefinite route does not lead to a misrouted call, but rather to an inefficient choice of gateway.

To sum up, aggregation may reduce information, but it should be avoided due to the resulting inaccuracies. Even if half of the information could be reduced with aggregation, which is an overoptimistic estimate, the number of entries would still be very high.

7.5 Routes to networks instead of to numbers

The scalability problems of the TRIP/CTRIP solution arise because a new entry is required for each moved number. The entry must be stored in all the databases, transported in the distribution messages and included in route computation. Paju [Paju 2002] proposed that a number portability database could return a routing number that is given as input to TRIP and CTRIP. Since properly chosen routing numbers can be aggregated, scalability is significantly improved. In this section, we examine combinations of TRIP/CTRIP with other methods and identifiers. Instead of routing to prefixes and to individual numbers, we propose routing to the destination network. The idea is to separate number portability from routing.

Two steps are required in this approach:

1. Given the E.164 number of the destination, obtain an intermediate identifier indicating the network where the receiving subscriber resides.
2. Given the intermediate identifier, obtain the next-hop address (in the IP network) or the routing number (in the SCN) for establishing the call.

The first step can be implemented using ENUM (see section 6.5) or with a replicated centralized database. The second step can be implemented with TRIP/CTRIP. The intermediate identifier identifies the network where the subscriber currently resides. The call is routed to this network using the next-hop address or routing number, and then the network may use any method for locating the subscriber within the destination network. One potential intermediate identifier is the TAD identifier (see section 6.3). Another alternative is to use the routing number directly as an identifier. Though other identifiers are possible, we only cover the use of these two in further discussion. With two types of intermediate identifiers and two alternatives for obtaining the identifier, we have four combinations, which are examined in the following subsections.

7.5.1 ENUM and TRIP/CTRIP with TAD identifiers

First, we examine how the TRIP/CTRIP solution can be combined with ENUM using a TAD identifier. When a call is set up, an ENUM query is performed. The query returns the URI of the destination; this may be an SIP URI for IP destinations and a routing number URI for SCN destinations. Already in section 6.5 we proposed to add a TAD parameter to various URI formats. With this parameter, the TAD is obtained simultaneously with the URI. The TAD is used as a

key to obtain the next hop address (in the IP network) or to obtain the routing number (in the SCN). Since TRIP/CTRIP currently only defines directory numbers and routing numbers as keys, a new type of address family must be added.

Definition of a new address family is straightforward. A new code needs to be reserved for the Address Family fields in TRIP. Because of the fixed length of the TAD (recall that the TAD is a superset of the ITAD identifier), the address can be given as a binary number. The TAD is always 4 octets long, so the length field in the generic route format contains the value 4.

Using this approach, TRIP and CTRIP become routing protocols for routes to networks instead of to specific numbers. The scalability problem is solved, since there needs to be a route to each TAD instead of to each number. The entries cannot be topologically aggregated, since the assignment of TADs does not respect the topology. Instead, reduction can be obtained by statistical aggregation, which combines entries that are sequential for probabilistic reasons. The efficiency is higher than for statistical aggregation of directory numbers, since the fixed length of the TAD allows for binary aggregation. Binary aggregation, however, requires the length of the TAD prefix to be known, so our new Address Family must be modified. The Address Family would then contain the 4-octet TAD and a 1-octet prefix-length field, and the length field in the generic route format would contain the value 5.

The mappings are:

$$DN \xrightarrow{DNS} TAD \xrightarrow{CTRIP} RN$$

$$DN \xrightarrow{DNS} TAD \xrightarrow{TRIP} IP$$

7.5.2 ENUM and TRIP/CTRIP with routing numbers

The intermediate TAD identifier can be skipped if routing numbers are obtained directly from the ENUM query. The ENUM query returns the routing number, for example using an “rnum” URI. Then the routing number is used directly (in the SCN) or as key to retrieve the next hop address with TRIP (in the IP network).

The TAD-extension to the URIs is not required, but there must be a routing number URI available. A new address format is not necessary: address families for decimal and pentadecimal routing numbers are already defined in TRIP. Further, routing numbers can be topologically aggregated, at least on a country-level. This approach seems superior to the previous one. The problem is rather administrative than technical. Each number in the IP network would require a routing number. The routing number is an SCN concept without any semantics in the IP network. It would be necessary to distribute routing numbers to IP telephony providers – something that would bring the legacy of the SCN to the IP telephony network. The routing number would be used directly in the SCN, and CTRIP would not be necessary. This, however, breaks the symmetry between the TRIP and CTRIP protocols. A major part of the power of the

TRIP/CTRIP model is due to this symmetry, which enables routes to be created seamlessly between the two technologies.

The mappings are:

$$DN \xrightarrow{DNS} RN$$

$$DN \xrightarrow{DNS} RN \xrightarrow{TRIP} IP$$

A variation of the approach can be described with the mappings:

$$DN \xrightarrow{DNS} RN_i \xrightarrow{CTRIP} RN$$

$$DN \xrightarrow{DNS} RN_i \xrightarrow{TRIP} IP$$

This variation uses an intermediate routing number to obtain the actual routing number in the SCN. The intermediate number would be a virtual routing number, which conceptually is similar to the TAD identifier. It would not be related to the real routing number in any way, and there are no corresponding routes. The main difference from TAD identifier is that the intermediate routing numbers follow a topological hierarchy, which allows them to be aggregated. Aggregation can reduce the number of entries, but would also let providers subdivide their networks into smaller areas and allows more precise routes to be formed. Nevertheless, the functionality is similar to TAD identifiers and we will regard them as a special case of TAD identifiers.

7.5.3 A database and TRIP/CTRIP with TAD identifiers

Likewise, the TAD identifier can be obtained from a database. The database solution could be one of the ones presented in Chapter 4. Instead of returning the routing number, the TAD identifier is obtained. Contrary to the corresponding ENUM-based solution, the TAD parameter of the URIs is not required. The TAD identifier is used as a key in TRIP/CTRIP, and therefore the new address family for TADs must be defined. The aggregation considerations are similar to the ones of the ENUM based solution in section 7.5.1.

The mappings are:

$$DN \xrightarrow{DB} TAD \xrightarrow{CTRIP} RN$$

$$DN \xrightarrow{DB} TAD \xrightarrow{TRIP} IP$$

7.5.4 A database and TRIP/CTRIP with routing numbers

The use of TRIP/CTRIP with routing numbers obtained from a database has the same advantages and disadvantages as the use with routing numbers obtained from ENUM. The difference is that the TAD parameter of the URI is not required.

The mappings are:

$$DN \xrightarrow{DB} RN$$

$$DN \xrightarrow{DB} RN \xrightarrow{TRIP} IP$$

7.5.5 Using IP addresses in the SCN

For symmetry, one could investigate the possibility of using IP addresses in the SCN. IP addresses can be topologically aggregated. A new address family is then required in CTRIP for IP addresses.

Using ENUM, the mapping is:

$$DN \xrightarrow{DNS} IP \xrightarrow{CTRIP} RN$$

$$DN \xrightarrow{DNS} IP$$

With a database, the mapping is:

$$DN \xrightarrow{DB} IP \xrightarrow{CTRIP} RN$$

$$DN \xrightarrow{DB} IP$$

This imaginary approach is however not feasible. It suffers from similar administrative problems as the approach using routing numbers as identifiers described in section 7.5.2. Each SCN terminal would require an IP address. Although it is expected that 3rd generation terminals are assigned IP addresses, it is not reasonable to allocate addresses to the vast number of fixed terminals that do not provide IP connectivity anyway. A static IP address would be needed for each terminal. As IP addresses are a scarce resource already, it is impossible to assign addresses to terminals that do not require them. With this motivation, we abandon the approach of using intermediate IP addresses.

7.6 Summary

After a brief presentation of TRIP and CTRIP, we examined the scalability of these protocols. Scalability problems arise due to number portability since entries corresponding to individual numbers are created. We showed that aggregation does not provide any significant reduction of information amount. A solution is to create routes to the destination network instead of routes to the individual numbers. This solution requires an additional intermediate identifier, e.g. a TAD identifier, an IP address or a routing number, and an additional mapping. We compared schemes based on these intermediate identifiers in combination with a mapping method (DNS or DB).

Chapter 8
Analysis of complete scenarios

We have studied various solutions to the problems of routing, gateway location and number portability in a hybrid network. The solutions aim to solve different problems, which partly overlap: some approaches solve the gateway location problem only in one direction (e.g. for IP→SCN calls), some approaches solve it in both directions and some solve number portability issues as well. Number portability is generally possible to implement in any mapping: modifying the mapping results in a moved number – the challenge of number portability is rather to do it efficiently and to utilize aggregation. In this chapter, we analyze complete scenarios covering all partial problems. After a brief introduction to ad hoc combinations of solutions, we present a more systematic approach based on schemes that combine mappings and identifiers. We describe the dependency between number portability and gateway location, the available identifiers and motivation for using intermediate identifiers. Then we select the most appropriate schemes and analyze their properties.

8.1 Problem separation

In order to use the discussed approaches to build an architecture supporting both number portability and gateway location, and in order to be able to compare the approaches, we need to divide the problem into smaller partial problems. The following partial problems can be identified:

1. *Number portability in SCN networks*. Definition how the SCN-specific NP-information is distributed and accessed, and how the information is replicated among the databases.
2. *Number portability in IP networks*. Definition how the IP-specific NP-information is distributed and accessed, and how the information is replicated among the databases.
3. *Gateway location for IP→SCN calls*. Selection of the most suitable gateway for a given IP→SCN call according to the parameters of the gateway and policies of the operator.
4. *Gateway location for SCN→IP calls*. Selection of the most suitable gateway for a given SCN→IP call according to the parameters of the gateway and policies of the operator.
5. *Network-path routing*. Selection of the most suitable path of intermediate networks interconnecting the originating network and the destination network in a hybrid SCN/IP

scenario, observing the policies of the operators. We do not consider network-path routing as an obligatory function.

8.2 Combinations

Since some of the solutions only address a particular set of partial problems, we need to combine several solutions in order to cover the complete functionality. Table 8.1 shows which partial problems are supported by the discussed solutions.

Table 8.1: Scope of the described approaches

Approach		Number portability SCN	Number portability IP	Gateway location IP→SCN	Gateway location SCN→IP	Network-path routing
1	Master DB	X				
2	HUT DB	X	X			
3	TRIP		(X) [1)]	X		X
4	CTRIP	(X) [1)]			X	X
5	ENUM		X			
6	ENUM with “rnum” URI	X				
7	DNS-based gateway location (approach 1)			X	X [2)]	
8	DNS-based gateway location (approach 2)			X	X [2)]	
9	DNS-based gateway location (approach 3)			X	X [2)]	X
10	TAD-parameter in URI	X [3)]	X [3)]			
11	DB mapping between DN and TAD	X [3)]	X [3)]			

[1)] Scalability problems
[2)] With the “rnum” URI
[3)] Requires additional mapping from TAD to an routing address

The rows in Table 8.1 can be combined so that all the required functionality is covered, forming a complete architecture for both number portability and the gateway location. For example, the combination of row 2 with row 9 covers all the functionality. Table 8.2 presents some example combinations that cover all aspects. The numbers in parentheses refer to the row number in Table 8.1.

Table 8.2: Some combinations of the described approaches

Combined approaches	Number portability SCN	Number portability IP	Gateway location IP→SCN	Gateway location SCN→IP	Network-level routing
TRIP/CTRIP	X (4)	X (3)	X (3)	X (4)	X (3, 4)
HUT-DB + TRIP/CTRIP	X (2)	X (2) [1)]	X (3) [1)]	X (4) [1)]	X (3, 4) [1)]
TRIP/CTRIP + TAD-parameter in URI	X (10)	X (10)	X (3)	X (4)	X (3, 4)
Database + TRIP/CTRIP with routing numbers	X (1, 2)	X (2)	X (3)	X (4)	X (3, 4)
Database + TRIP/CTRIP with TAD identifiers	X (11)	X (11)	X (3)	X (4)	X (3, 4)
DNS-based gateway location + Database	X (2) [3)]	X (2) [3)]	X (9)	X (9)	X (9) [2)]
DNS-based gateway location + TAD parameter in URI	X (10)	X (10)	X (9)	X (9)	X (9) [2)]
DNS-based gateway location + Database	X (11)	X (11)	X (9)	X (9)	X (2) [2)]

[1)] Also applies to the Master system, if information about IP terminals are added
[2)] In approach 3 only
[3)] Modified to return a TAD instead of RN/IP

Nevertheless, not all combinations form working and efficient architectures. The above way of combining solutions represents an ad-hoc approach to building the architecture. In order to examine which combinations are feasible, we need to consider the mappings in a systematical way. Furthermore, a systematical analysis of the combinations of different identifiers and mapping methods can reveal approaches not discussed before.

8.3 Schemes

To implement both number portability and gateway location the following set of mappings is required:

$DN \xrightarrow{A} RN$ (SCN→SCN calls)
$DN \xrightarrow{B} RN_{gw}$ (SCN→IP calls)
$DN \xrightarrow{C} IP$ (IP→IP calls)
$DN \xrightarrow{D} IP_{gw}$ (IP→SCN calls)

These mappings map the *name identifier* to the *address identifier*. In telephony the name identifier is the E.164 directory number (DN). The address identifier is the routing number (RN) in the SCN and the IP address in the IP network. The mappings that give the address identifier of the terminal perform *terminal location* (marked with A and C). The mappings that give the

address identifier of the gateway perform *gateway location* (marked with B and D). We call this type of scheme a *direct scheme*, since the address identifier is obtained with a single mapping.

Remembering that a mapping can be split into two mappings, we can separate between number portability and gateway/terminal location using an intermediate identifier. This forms an equivalent set of mappings:

$DN \xrightarrow{A} X_1 \xrightarrow{E} RN$ (SCN→SCN calls)
$DN \xrightarrow{B} X_2 \xrightarrow{F} RN_{gw}$ (SCN→IP calls)
$DN \xrightarrow{C} X_3 \xrightarrow{G} IP$ (IP→IP calls)
$DN \xrightarrow{D} X_4 \xrightarrow{H} IP_{gw}$ (IP→SCN calls)

The *intermediate identifiers* X_1, X_2, X_3 and X_4 can be of the same or of different type. Number portability can be implemented either before or after the intermediate identifier. In practice, it is best performed before the intermediate identifier, since the goal is that it should be possible to aggregate the intermediate identifier. The use of intermediate identifiers can be motivated with the improved efficiency in aggregation, as will be discussed in section 8.7. In the above scheme, the mappings A, B, C and D perform number portability. We call them the *primary mappings*. The mappings E and G perform terminal location, and the mappings F and H perform gateway location. These constitute *secondary mappings*. We use the term *intermediate identifier scheme* for schemes based on intermediate identifiers.

These basic types can be mixed so that an intermediate identifier scheme is used in the SCN and a direct scheme is used in the IP network, or vice versa. The first mapping is always the primary mapping. Thus, the direct scheme contains only primary mappings.

8.4 Dependency between number portability and gateway location

Before constructing and evaluating complete scenarios, we discuss the concept of dependency between number portability and gateway location (NP-GWloc dependency). Consider the following scenario, where different systems are used for number portability and gateway location in different technologies.

$DN \xrightarrow{DB_{master}} RN$ (SCN→SCN calls)
$DN \xrightarrow{static} RN_{gw}$ (SCN→IP calls)
$DN \xrightarrow{DNS} IP$ (IP→IP calls)
$DN \xrightarrow{TRIP} IP_{gw}$ (IP→SCN calls)

When a number is ported between two operators in the SCN, the information in the Master database is updated. Since the topological location of the destination has changed, another gateway may have to be selected, and the information in TRIP must be updated. Similarly, when a number in the IP network moves, the information in ENUM is updated. The new topological location may require the static mapping in several exchanges in the SCN to be updated for

maintaining routing efficiency. If a number moves from the SCN to the IP network, all mappings must be changed.

The problem arises because of the dependency between number portability and gateway location. Gateway location for IP→SCN calls depends on number portability in the SCN. Gateway location for SCN→IP calls depends on number portability in the IP network. Number portability between the SCN and IP networks, can be seen as number portability in both SCN and IP, and affects gateway location for both IP→SCN and SCN→IP calls. Because of this dependency, coordination is required between the methods used in different networks. When information in one mapping changes, one or all of the other mappings must be updated. In the given example, which corresponds to the current plans, there are dependencies between four mapping methods – this substantially complicates number portability and efficient routing.

One way to solve the dependency problem is to automate information transfer between the different methods, as in the TRIP/CTRIP architecture:

$DN \xrightarrow{CTRIP} RN$ (SCN→SCN calls)
$DN \xrightarrow{CTRIP} RN_{gw}$ (SCN→IP calls)
$DN \xrightarrow{TRIP} IP$ (IP→IP calls)
$DN \xrightarrow{TRIP} IP_{gw}$ (IP→SCN calls)

Here, the same mapping is used for both gateway location and number portability within one technology. This eliminates the need for information transfer between the methods within one technology. Information transfer between mappings used in different technologies (between TRIP and CTRIP) is automatically provided by the numbering gateway. Although this solves the dependency problem, it does not scale well for number portability as we have seen.

Another way to solve the dependency problem is to use technology-independent intermediate identifiers. In the following scenario, the intermediate identifier is common to both technologies. This allows the primary mapping to be identical for both technologies, and no information transfer or conversion is required between the mappings.

An example of this approach is:

$DN \xrightarrow{DB} TAD \xrightarrow{CTRIP} RN$ (SCN→SCN calls)
$DN \xrightarrow{DB} TAD \xrightarrow{CTRIP} RN_{gw}$ (SCN→IP calls)
$DN \xrightarrow{DB} TAD \xrightarrow{TRIP} IP$ (IP→IP calls)
$DN \xrightarrow{DB} TAD \xrightarrow{TRIP} IP_{gw}$ (IP→SCN calls)

8.5 Identifiers

The identifiers discussed in this work are summarized in Table 8.3. Aggregation base refers to the number of consecutive entries needed to form one aggregated entry. Because of their variable length, directory numbers and routing numbers must be aggregated on digit-by-digit basis. TADs and IP addresses have a fixed length, and therefore binary aggregation can be used. URIs are character strings, which cannot be aggregated, but theoretically their size could be reduced with compression algorithms. Even though aggregation is possible, it may not be feasible in practice: number portability makes aggregation of directory numbers inefficient and network identifiers will probably not be allocated with any hierarchical structure.

Table 8.3: Summary of identifiers

	Directory numbers (DN)	Routing numbers (RN)	Network identifiers (TAD)	IP-addresses (IP) [1)]	Uniform Resource Identifiers (URI)
Size	1…15 digits	Variable	32 bits	32 bits	Variable
Aggregation possible	Yes	Yes	Yes	Yes	No
Aggregation base	10	16 (also 10 or 15)	2	2	-
Aggregation practically feasible	No (due to number portability)	Yes	No (due to non-hierarchical allocation)	Yes	No
Availability	Moderate	Good	Good	Poor	Good
Home technology	SCN	SCN	IP and SCN	IP	IP
Administrative body	ITU-T, National regulator	National regulator	IANA [2)]	IANA, (APNIC, ARIN, LACNIC, RIPE NNC)	Based on domain names or IP addresses

[1)] Only version 4 considered in this work.

[2)] The TAD identifier is the proposed superset of the ITAD and CTAD identifiers. ITADs are administered by IANA. CTAD is a proposed counterpart to ITAD identifiers for the SCN.

The availability of new directory numbers and routing numbers depends on the national number plan. Since, routing numbers can be relatively long, and are not directly accessed by the user, the availability is good. Availability of TAD identifiers is currently good, but the limited size can decrease availability later. IP addresses are already a scarce resource, and this shortage has led to using techniques such as network address translation (NAT). The length of URIs is unlimited and

availability is therefore good. Home technology means the technology for which the identifier was originally designed.

Any type of identifier assigned to networks can be used as network identifier instead of the TAD defined as the generalization of the ITAD identifier. In this and the following chapter, we use TADs as examples of network identifiers, but practical implementation could also use some other type of identifier.

We will not consider IP addresses as potential intermediate identifiers because of the limited availability. URIs are not directly suitable as intermediate identifiers because of their length. However, URIs in the "directorynumber@ip-address" format can be constructed from the directory number and the IP address.

8.6 Mapping methods

Table 8.4 presents the mapping methods described in this work. The column indicates the source identifier and the row indicates the destination identifier. The mapping from IP addresses to other identifiers (IP→) is omitted because this mapping is only required when IP addresses are used as intermediate identifiers, and we do not consider IP addresses as appropriate intermediate identifiers.

Table 8.4: Described mapping methods

	DN→	TAD→	RN→
→RN	CTRIP [1)] ENUM with "rnum" URI Master DB HUT DB	CTRIP [2)]	–
→IP	TRIP [1)] ENUM HUT DB	TRIP [2)]	TRIP
→TAD	TAD-parameter in URI	–	TAD-parameter in "rnum" URI
$\rightarrow RN_{gw}$	CTRIP DNS-gw (approach 1)	CTRIP with new TAD address family DNS-gw (approach 2) DNS-gw (approach 3)	CTRIP
$\rightarrow IP_{gw}$	TRIP DNS-gw (approach 1)	TRIP with new TAD address family DNS-gw (approach 2) DNS-gw (approach 3)	TRIP

[1)] Scalability problems

[2)] Requires the definition of a new address family for TADs as described in section 7.5.

In addition to the methods in Table 8.4, mappings between any types of identifiers can be implemented with a static mapping or using a database. These are omitted from the table, since they are applicable between any pair of identifiers.

8.7 Motivation for using intermediate identifiers

The methods that perform network-path routing (i.e. TRIP, CTRIP and *DNS) require the input number space to be small and stable. These methods create a separate route for each element in the input number space and the routing databases are replicated in several networks. Large databases slow down the decision process and a frequently changing input space causes instability and route flapping in TRIP and CTRIP.

A small number space is accomplished by performing the number portability mapping before network-path routing. The intermediate identifier connects the number portability mapping to network-path routing. The TAD identifier has both a small and stable number space: the size of the number space is proportional to the number of operators, and the assignment of identifiers to operators is invariable. On the other hand, a hierarchical number space functions as a small number space when aggregation is used. Routing numbers and IP addresses have a topological hierarchy, and aggregation allows specifying routes for a small number of prefixes.

The primary reason for using an intermediate identifier is to be able to utilize a small intermediate number space. The other main reason is to solve the dependency problem discussed in section 8.4.

Additionally, the use of intermediate identifiers separates between the functions of number portability and gateway/terminal location. There is no need for the location methods to be similar on both technologies, and even different operators may use different methods. The operators can form clusters, where the networks within the cluster share location information. The operators within a cluster can share gateways if their policies allow.

8.8 Choice of intermediate identifier

We consider scenarios both with and without an intermediate identifier. The intermediate identifiers discussed in this work are the network identifier (TAD), the routing number (RN) and the IP-address (IP). With four different “types” of intermediate identifiers (3 identifier types + no identifier), used in gateway location and number portability in two technologies there are $4^{2\cdot2} = 256$ combinations. The number of combinations is even higher if more than one intermediate identifier is used.

In order to reduce the possible combinations, we abandon the most inappropriate candidates. Firstly, we only consider schemes containing at most two consecutive mappings (only one intermediate identifier). The reason is that a third mapping does not provide any additional benefit, but rather slows down call setup and adds overhead.

Secondly, we always perform the number portability mapping in the primary mapping. The reason is that only then the number space for the second identifier is small and/or aggregateable.

Thirdly, we abandon the solutions where a different intermediate identifier is used for number portability and gateway location within a technology. For example, we will not consider a scheme where the mapping DN→RN is used for number portability and DN→TAD→RN_{gw} for gateway location. The reason is that such a solution requires maintenance of two different systems for the same technology. It would create dependencies within one technology.

After these eliminations, the remaining combinations can be described with the following set of mappings:

$DN \longrightarrow A \longrightarrow RN$ (SCN→SCN calls)
$DN \longrightarrow A \longrightarrow RN_{gw}$ (SCN→IP calls)
$DN \longrightarrow B \longrightarrow IP$ (IP→IP calls)
$DN \longrightarrow B \longrightarrow IP_{gw}$ (IP→SCN calls)

In these mappings, A indicates the intermediate identifier used in the SCN, and B indicates the intermediate identifier used in the IP network. The intermediate identifiers can be omitted, whereas we mark the type of A or B as "None". The TAD can be considered as a generalization of any identifier representing the network of one operator. Table 8.5 shows the possible combinations of intermediate identifiers with the combinations numbered as references for further discussion.

Table 8.5: Combinations of intermediate identifiers

		Intermediate identifier in SCN (A)			
Intermediate identifier in IP network (B)		None	TAD	RN	IP
	None	1	2	3	4
	TAD	5	6	7	8
	RN	9	10	11	12
	IP	13	14	15	16

The shaded cells in Table 8.5 represent combinations that are not appropriate: Since IP addresses are a scarce resource, they are not suitable as intermediate identifiers, as discussed in section 7.5.5. Therefore, we eliminate combinations number 4, 8, 12, 13, 14, 15 and 16. In combinations number 3, 7, 11 and 15, the intermediate identifier in the SCN is the same type as the address identifier. Thus, the last mapping is either an identity mapping (RN→RN) or a mapping between two different identifiers of the same type (RN_a→RN_b). Both situations are inappropriate, and we abandon the corresponding combinations. However, we can regard the use of intermediate routing numbers as a special case of aggregatable TAD identifiers, and combination 11 is thus discussed as a special case of combination 6.

Combination number 10 is described by the mappings:

$DN \longrightarrow TAD \longrightarrow RN$ (SCN→SCN calls)
$DN \longrightarrow TAD \longrightarrow RN_{gw}$ (SCN→IP calls)
$DN \longrightarrow RN \longrightarrow IP$ (IP→IP calls)
$DN \longrightarrow RN \longrightarrow IP_{gw}$ (IP→SCN calls)

Although this combination is functional, it represents a scheme where intermediate identifiers are used in places where they do not provide any benefit. There are no benefits in using routing numbers in the IP network if a different intermediate identifier is used in the SCN.

Now the 256 combinations have been reduced to five appropriate schemes (numbered 1, 2, 5, 6 and 9 in Table 8.5). We discuss these schemes in the following sections.

8.9 Direct scheme

The direct scheme is the simplest scheme, as the name identifier is mapped to an address identifier in a single mapping. Both number portability and gateway location are performed with the same mapping.

We consider the methods given in Table 8.4, as well as static mappings and shared database mappings. With these methods, we can describe the scheme with the following mappings (the slash "/" marks alternative methods):

$DN \xrightarrow{DB/DNS/CTRIP/static} RN$ (SCN→SCN calls)
$DN \xrightarrow{DB/DNS/CTRIP/static} RN_{gw}$ (SCN→IP calls)
$DN \xrightarrow{DB/DNS/TRIP/static} IP$ (IP→IP calls)
$DN \xrightarrow{DB/DNS/TRIP/static} IP_{gw}$ (IP→SCN calls)

The NP-GWloc dependencies are between the DN→RN mapping and the DN→IP_{gw} mapping, and between the DN→IP mapping and the DN→RN_{gw} mapping. This implies the requirement that it should be easy to transfer information between the corresponding methods, or the methods should be the same. If number portability between technologies is considered, all mappings are dependent.

Information transfer between different types of static mappings, or between a static mapping and another mapping method is difficult, so static mappings should be avoided. Information between DNS and any other method is difficult as well, which is a result of the distribution model of DNS. Therefore, if DNS is used then all mappings should be performed by DNS. Information transfer between TRIP, CTRIP and distributed databases is rather straightforward, so any combination of these methods is good. However, TRIP and CTRIP suffer from scalability problems when used in the primary mapping. The Master system, which is a database mapping, can be used for the SCN part of the direct scheme.

The current development in Finland is pointing towards a direct scheme:

$DN \xrightarrow{DB_{master}} RN$ (SCN→SCN calls)
$DN \xrightarrow{static} RN_{gw}$ (SCN→IP calls)
$DN \xrightarrow{DNS} IP$ (IP→IP calls)
$DN \xrightarrow{static/TRIP} IP_{gw}$ (IP→SCN calls)

This scheme has different methods for every mapping, and involves dependencies between widely dissimilar methods. The information transfer between different mappings is difficult to automate. The result is that the management cost for number portability is high, which may lead to resistance in deploying number portability between technologies.

8.10 Intermediate TAD scheme

In the intermediate TAD scheme, the functions of number portability and gateway location are separated. The primary mapping (between directory number and TAD) performs number portability. Since this mapping is technology independent, the same method and the same information can be used in both technologies. Thus, only one technology independent system for number portability needs to be maintained.

The secondary mapping provides gateway and terminal location. It gives the route to the specific TAD. The mapping is technology dependent. In fact, the last mapping is always technology dependent, since it must return either a routing number or a URI depending on the technology. The secondary mapping is independent of the primary mapping: if a number is ported, only the primary mapping is updated, and the update does not proceed to the secondary mapping.

Applying the methods from Table 8.4, the scheme can be described with the following mappings:

$DN \xrightarrow{DB/DNS/static} TAD \xrightarrow{CTRIP/*DNS/DB/static} RN$ (SCN→SCN calls)
$DN \xrightarrow{DB/DNS/static} TAD \xrightarrow{CTRIP/*DNS/DB/static} RN_{gw}$ (SCN→IP calls)
$DN \xrightarrow{DB/DNS/static} TAD \xrightarrow{TRIP/*DNS/DB/static} IP$ (IP→IP calls)
$DN \xrightarrow{DB/DNS/static} TAD \xrightarrow{TRIP/*DNS/DB/static} IP_{gw}$ (IP→SCN calls)

All NP-GWloc dependencies are between the DN→TAD mappings and therefore information exchange between the primary mappings should be easy. Information transfer between static mappings is difficult, so these are not considered. Combinations of database mappings are applicable because information transfer between databases is relatively easy. Also DNS is a good solution if used in both technologies. Since information transfer between databases and DNS is difficult, the DB and DNS mappings should not be combined. The Master system cannot without modification be a part of this scheme, since it does not provide a mapping to a TAD.

Network-path routing is supported if TRIP/CTRIP or recursive DNS (marked *DNS) is used in the secondary mapping. Recursive DNS was described in Section 6.4. Thanks to the technology-independent intermediate identifier, the method for the secondary mapping can be different in the

SCN and the IP network. It can even be different for gateway location and for terminal location. Every operator can use a separate method. However, using the same method has the advantage that routes can be seamlessly formed across network and technology borders.

The TAD identifiers defined as the generalization of the ITAD identifiers are flat and every provider has a single identifier. If the concept of TAD is extended to long numbers that have a hierarchical structure, these could be aggregated. Further, a provider could extend the hierarchy by subdividing the network into smaller areas, and control routing more accurately. This type of network identifier basically works as a virtual routing number. If TRIP or CTRIP are used for the secondary mapping, an address family must be defined for the aggregatable network identifier. Furthermore, all other TAD fields in the protocol must be adapted for this new network identifier, or alternatively two parallel types of network identifiers must be used (the TAD and the aggregatable network identifier). DNS mapping from aggregatable network identifiers is easier to modify since it has not yet been standardized.

8.11 Intermediate RN scheme

The number portability part of the intermediate RN scheme resembles the intermediate TAD scheme. Number portability is provided by the primary mapping, which maps the directory number into a routing number. The routing number can be seen as a "virtual" routing number in the IP network, since it only functions as an identifier. The mapping is technology independent. Thus, the same method and the same information can be used in both technologies, and only one system for number portability needs to be maintained. One suitable candidate for the number portability mapping is the Master system, since it provides the necessary DN→RN mapping.

In the SCN, the number portability mapping and the gateway location mappings are combined. The mapping method returns the routing number to a gateway if the destination is in the IP network. In the IP network, a secondary mapping (between routing number and URI) provides terminal and gateway location. The secondary mapping is independent of the primary mapping.

Applying the methods from Table 8.4, the scheme can be described with the following mappings:

$$DN \xrightarrow{DB/CTRIP/DNS/static} RN \qquad \text{(SCN→SCN calls)}$$

$$DN \xrightarrow{DB/CTRIP/DNS/static} RN_{gw} \qquad \text{(SCN→IP calls)}$$

$$DN \xrightarrow{DB/CTRIP/DNS/static} RN \xrightarrow{TRIP/DB/static} IP \qquad \text{(IP→IP calls)}$$

$$DN \xrightarrow{DB/CTRIP/DNS/static} RN \xrightarrow{TRIP/DB/static} IP_{gw} \qquad \text{(IP→SCN calls)}$$

The NP-GWloc dependencies are between the DN→RN mappings, so if the same method is in both the SCN and in the IP network, the dependencies are eliminated. The applicability of different methods is similar to the intermediate TAD scheme. However, the intermediate RN scheme additionally contains the CTRIP method, which can be combined with database methods, although it does not provide the required level of scalability.

The drawback of this scheme is that a routing number must be allocated to IP terminals. However, the size of the numbering space is not limiting, since routing numbers can be made longer and they can contain pentadecimal digits. Rather the administration of routing numbers is a burden.

8.12 Intermediate TAD in SCN only scheme

The "intermediate TAD in SCN only" scheme is a combination of the direct and the intermediate TAD schemes. The TAD identifier is used in the SCN only. The number portability mapping is technology dependent. In the SCN, number portability and gateway location are performed with separate mappings. In the IP network, a single mapping performs number portability and gateway location.

Applying the methods from Table 8.4, the scheme can be described with the following mappings:

$DN \xrightarrow{DB/DNS/static} TAD \xrightarrow{CTRIP/*DNS/DB/static} RN$ (SCN→SCN calls)

$DN \xrightarrow{DB/DNS/static} TAD \xrightarrow{CTRIP/*DNS/DB/static} RN_{gw}$ (SCN→IP calls)

$DN \xrightarrow{DB/DNS/TRIP/static} IP$ (IP→IP calls)

$DN \xrightarrow{DB/DNS/TRIP/static} IP_{gw}$ (IP→SCN calls)

The NP-GWloc dependencies are between the DN→TAD mapping and the DN→IP mapping, and between the DN→TAD mapping and the DN→IP_{gw} mapping. The dependencies are similar to the ones in the direct scheme, which implies that the same combinations of methods are applicable. The difference is the missing CTRIP method in the DN→TAD mapping. Thus, database mappings can be combined with TRIP and other database mappings only. If DNS is used then all mappings should preferably be performed by DNS. The Master system cannot be included in this scheme, since it does not provide a mapping to a TAD.

8.13 Intermediate TAD in IP network only scheme

The "intermediate TAD in IP only" scheme is a combination of the direct and the intermediate TAD schemes as well. The TAD identifier is used in the IP network only, and the number portability mapping is technology dependent. In the IP network, separate mappings are used for number portability and gateway location, while a single mapping performs both functions in the SCN.

Applying the methods from Table 8.4, the scheme can be described with the following mappings:

$DN \xrightarrow{DB/DNS/CTRIP/static} RN$ (SCN→SCN calls)

$DN \xrightarrow{DB/DNS/CTRIP/static} RN_{gw}$ (SCN→IP calls)

$DN \xrightarrow{DB/DNS/static} TAD \xrightarrow{TRIP/*DNS/DB/static} IP$ (IP→IP calls)

$DN \xrightarrow{DB/DNS/static} TAD \xrightarrow{TRIP/*DNS/DB/static} IP_{gw}$ (IP→SCN calls)

The NP-GWloc dependencies are between the DN→RN mapping and the DN→TAD mapping, and between the DN→RN_{gw} mapping and the DN→TAD mapping. The dependencies are similar to the ones in the previous scheme, so the preferred combinations of mapping methods are similar, with CTRIP instead of TRIP. The scheme can be based on the use of the Master system in the SCN, which would imply using a database mapping also in the IP network.

8.14 Summary

In this chapter, we analyzed scenarios solving both number portability and gateway location in a hybrid network. We identified the partial problems that the solutions in previous chapters solve. We specified what mappings are required for the number portability and the gateway location problems, and how a mapping can be split into two mappings using an intermediate identifier. We analyzed which methods can be used for providing specific mappings, and which dependencies there are between mappings. Finally, we selected five appropriate schemes and analyzed the applicability of various identifiers and methods in them.

Table 8.6 summarizes the properties of the five discussed schemes. The table can be used for comparing the schemes and selection of appropriate methods for number portability and gateway location.

The scheme together with the information model determines the structure and performance of the architecture on a higher level. The information model determines how information is shared between the technologies. On the other hand, the scheme and the used mapping methods determine how information is transferred between different mappings. Information transfer between some mappings is difficult (e.g. DNS and DB) while transfer between some mappings is easy (TRIP and CTRIP). The scheme also determines whether the same method can be used in both technologies.

Table 8.6: Properties of schemes with or without intermediate identifiers

Property \ Scheme	Direct	Intermediate TAD	Intermediate RN	Intermediate TAD in SCN only	Intermediate TAD in IP network only
Primary mappings	DN→RN DN→IP	DN→TAD	DN→RN	DN→TAD DN→IP	DN→RN DN→TAD
Secondary mappings	(none)	TAD→RN TAD→IP	RN→IP	TAD→RN	TAD→IP
Primary and secondary mappings combined in SCN	Yes	No	Yes	No	Yes
Primary and secondary mappings combined in IP	Yes	No	No	Yes	No
Can include the Master system	Yes	No	Yes	No	Yes
Requires new TAD identifier	No	Yes	No	Yes	Yes
Requires routing numbers for IP terminals	No	No	Yes	No	No
Recommended combinations of NP mapping methods [1)]	DNS or DB + TRIP + CTRIP	DNS or DB	DNS or DB + CTRIP	DNS or DB + TRIP	DNS or DB + CTRIP

[1)] A + B indicates any combination of A and B.

Chapter 9
Evaluation of mapping methods

After selecting a few feasible schemes and examining combinations of mapping methods and identifiers, we proceed to evaluating the mapping methods. The desired properties of a mapping method depend on whether the method is used in the primary or secondary mapping. Therefore, we evaluate the methods separately for these two cases. The focus of the evaluation is on scalability, size and performance issues. Also administration and security aspects, the porting procedure and the gateway selection mechanism are considered.

9.1 Purpose

In the five schemes developed in Chapter 8, some mappings were used for number portability (NP), some for gateway/terminal location and some for both. Table 9.1 classifies the mappings according to their purpose.

Table 9.1: Purpose of mappings

Mapping	Purpose	Methods	Scheme
DN→RN	NP and location	DB, DNS, CTRIP, static	Direct, Intermediate RN, Intermediate TAD in IP
DN→IP	NP and location	DB, DNS, TRIP, static	Direct, Intermediate TAD in SCN
DN→TAD	NP	DB, DNS, static	Intermediate TAD, Intermediate TAD in SCN
TAD→RN	Location	CTRIP, DB, *DNS, static	Intermediate TAD, Intermediate TAD in SCN, Intermediate TAD in IP
TAD→IP	Location	TRIP, DB, *DNS, static	Intermediate TAD, Intermediate TAD in IP
RN→IP	Location	TRIP, DB, static	Intermediate RN

Number portability is always performed in the primary mapping, which maps a large number space into a small one. Therefore, the scalability requirement for the primary mapping is high.

Each ported number requires an entry in the number portability database. In addition, some schemes involve a secondary mapping. These map a small or aggregated number space into a relatively small number of routing addresses. For each identifier, there may be several possible routing addresses. The input of the primary mapping is always a directory number. The input of secondary mappings is either a TAD identifier or a routing address.

We will not consider static mappings appropriate for the primary mapping because of the high expense of manual work in the frequently changing mapping. However, static secondary mappings are a viable option since secondary mappings are infrequently changed. Shared database mappings can be implemented in numerous ways. Therefore, we only discuss databases adhering to the Master system or the HUT database solution for the primary mapping, although other databases could be used as well. We regard both database solutions in a similar way, which is motivated by the ease to extend the Master system with information about IP terminals. Since no existing databases are applicable to the secondary mappings, we discuss database solutions at a more general level for the secondary mapping. We assume the operator-maintained model for DNS implementations.

9.2 Evaluation of primary mapping methods

Let X denote the intermediate identifier or the address identifier. This identifier can be a routing number, IP address or TAD. The mapping methods covered by the evaluation are:

- Database mappings, $DN \xrightarrow{DB} X$
- DNS mappings, $DN \xrightarrow{DNS} X$
- TRIP/CTRIP mappings, $DN \xrightarrow{TRIP/CTRIP} X$

9.2.1 Administration and security

In the database solutions, the data is centrally located. The distribution is controlled by a third party, who coordinates the distribution and authenticates the information. An incidental or malicious change in an operator's local database copy does not propagate to the other operators' databases. Changes are authenticated, and therefore operators can only change information about numbers assigned to them. Consequently, the database contents are very error resilient. However, the central database is a possible point of failure and a bottleneck. A problem with the central database inhibits distribution of updates but does not invalidate the information in the operator's database copies.

In DNS, the hierarchy leads to distinctive properties in data administration. The operator who initially was assigned a number block (the donor operator) maintains the information about all numbers belonging to the block, even about numbers that have been ported out of the block. When a number is ported, the old and the new operator must communicate with the donor operator to update the DNS information. Further, a number cannot move out of a block if the

donor operator does not use DNS. In that case, the information about the specific number must be inserted at a higher hierarchical level, i.e. at the regulator level. Since the information is in the network of the donor operator, the donor operators name server constitutes a single point of failure and a bottleneck. The information becomes inaccessible if the server is down longer than the lifetime of the entries. All operators must trust the donor operator. Anyone can access information in the public DNS, so privacy is not provided.

In the TRIP/CTRIP architecture, the operators are full peers. Each operator maintains the local information about its current numbers. The information distribution is not centrally controlled, so without authentication it is even possible to distribute malicious information about numbers belonging to other operators. In such an environment, the information must be authenticated and the operators must trust each other. On the other hand, the distribution implies that there is no single point of failure or bottleneck. Instead, error conditions in one network are reflected in the information distribution so that routes through an inaccessible network become unavailable. This is an expected and favorable consequence, since TRIP and CTRIP are fundamentally routing protocols.

9.2.2 Porting procedure

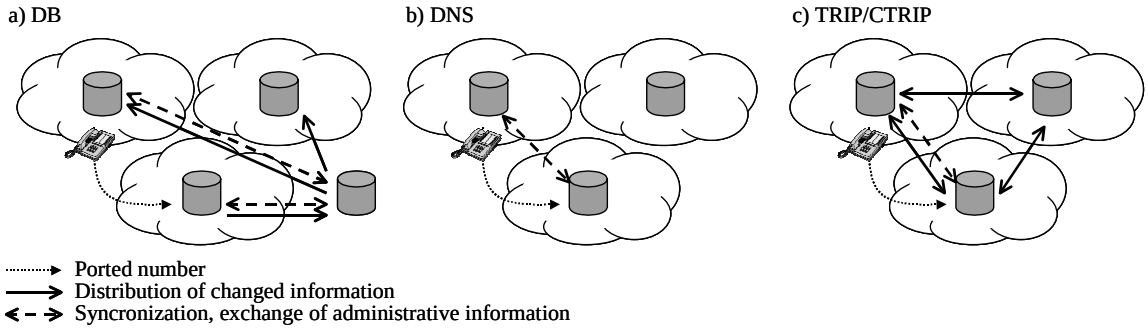

Figure 9.1: Information exchange for a ported number in the DB, DNS and TRIP/CTRIP methods

Then, we compare the messaging involved between networks when a number is ported between two operators. We ignore the messaging related to internal replication inside the operators' network. Figure 9.1 illustrates the information exchange for different mapping methods. The information includes two types of information exchange: the information used for (1) signaling and synchronizing the movement of the number, and for (2) distribution of updated information.

Both the HUT database approach and the Master system involves communication through a third party when a number is ported. In the Master system, the new operator informs that the number has been ported and a new routing number is installed; then the old operator confirms the move. The information about the ported number is distributed to all operators through the third party. According to Figure 4.2, this yields five messages between the previous operator and the third party, five messages between the new operator and the third party, and one message for each other operator. All messages are acknowledged, which doubles the message count. Marking the

number of operators with N_{op}, the total message count is $M = 20+2 \cdot N_{op}$. In HUT's approach, the new operator creates an entry in the ported numbers database (PNDB) of the third party. The entry indicates that the given number has been ported. The previous operator regularly reads the PNDB contents. It notices the ported number and confirms the move. The third party transfers the confirmed entry into the update database (UDB). The updated database contents are then fetched by all operators from the UDB. Considering only messages between networks, and assuming two messages (request, reply) are required per transaction, the total message count is $M = 4+2 \cdot N_{op}$.

The DNS approach involves minimal communication during the porting procedure. An entry with the new routing address is simply inserted into the database of the donor operator. The information is not replicated to other operator's DNS databases. The communication between the new and donor operator must be arranged with some out-of-band method. Depending on how the DNS hierarchy is utilized, information transfer to the higher-level DNS server may be required. Assuming that a five-message sequence similar to the one in Figure 4.2 is used for the out-of-band method between the new and the previous operator, the number of messages is $M = 5$.

In TRIP/CTRIP, the previous operator first withdraws its advertisement for the number[8]. The new operator advertises the route to the new number. Both the withdrawn and the new routes are propagated to the whole network. However, due to policies and aggregation, the distribution is usually limited to an area around the new and old network. There are at most two messages per peer connection per node. Oscillation may increase the number of messages before stability is reached. The communication and synchronization between the new and previous operator must be arranged with some out-of-band method, or using the Number Portability attribute proposed in [Beijar 2002].

In the case of HUT's database solution, entries are transferred in text format as SQL commands. The Master system transfers entries in text format as well, but in XML format, which involves higher overhead. TRIP/CTRIP transfers the entries in binary format. The amount of traffic during the porting procedure has a negligible impact on scalability, since porting is a relatively infrequent event.

9.2.3 Database size

The inputs of the primary mapping are directory numbers, which represent the individual numbers of ported terminals and (depending on implementation) the prefixes describing assigned number blocks. The amount of ported numbers is dominating, and because of negligible aggregation efficiency, the number of entries is practically equal to the number of ported numbers. Thus, the database size has approximately $S = N_{ported} = Q_{ported} \cdot N_{alloc}$ entries, where Q_{ported} is the fraction of ported numbers. Non-ported numbers are topological and they can be assumed to be represented as aggregated prefixes. However, in the case where non-ported numbers are

[8] If the number has been advertised as part of a less specific prefix, this prefix must not be withdrawn. In that case, no changes in the information advertised by the previous operator are made.

stored as individual entries in the database and no aggregation is used, the worst-case size is $S = N_{alloc}$.

All the discussed methods support mapping from aggregated directory number prefixes.

In the database solutions, each operator has a copy of the database containing all entries, and the size is $S_{operator,DB} = S$. Let N_{op} denote the number of operators. The third party also has a copy of the database, so the total database size of all operators is

$$S_{total,DB} = N_{op} \cdot S + S \tag{9.2}$$

In DNS, operators only have entries for the numbers originally assigned to them. Each operator only has a portion of the number space and the total database size of all operators is

$$S_{total,DNS} = S \tag{9.3}$$

Each TRIP/CTRIP node contains several databases containing slightly different versions of the same entry. Considering only external peers, each node has one Adj-TRIBs-In and one Adj-TRIBs-Out for each peer. Additionally there is one Ext-TRIB, one Local-TRIB and one Loc-TRIB per node. Denoting the number of peers of a operator i with $N_{peers,i}$ the database size of one operator is

$$S_{operator,TRIP} = \left(2N_{peers,i} + 3\right)S \tag{9.4}$$

and the total database size of all operators is

$$S_{total,TRIP} = \sum_{i=1}^{N_{op}} \left(2N_{peers,i} + 3\right)S \tag{9.5}$$

The information in the Loc-TRIB, and the Adj-TRIBs-Out have only minor differences, which depend on route selection. The differences between these databases and the Ext-TRIB are also small. An advanced implementation can therefore reduce the total size by using pointers instead of storing each entry. However, since each peer advertises different routes, the information in the Adj-TRIBs-In cannot be reduced significantly. With an estimated optimal reduction, the information in the Loc-TRIB, the Adj-TRIBs-Out and the Ext-TRIB can be reduced to 1.7 times the size of one complete database. The motivation is that the estimated difference between the Loc-TRIB and different Adj-TRIBs-Out is about 10...30% and between the Loc-TRIB and Ext-TRIB about 50%. Each Adj-TRIB-In is added to the sum, resulting in an estimated $N_{peers} \cdot S + 1.7 \cdot S$ entries in a node, where N_{peers} is the number of peers of the node. The database size of one operator is then approximately

$$S'_{operator,TRIP} = \left(N_{peers,i} + 1.7\right)S \tag{9.6}$$

and the total database size is approximately

$$S'_{total,TRIP} = \sum_{i=1}^{N_{op}} \left(N_{peers,i} + 1.7\right)S \tag{9.7}$$

9.2.4 Query performance

In the database solutions and the TRIP/CTRIP solution, the query is made to the local copy of the database. Internal synchronization allows for multiple identical copies of the database, which is useful for load distribution. The query involves only one round-trip time.

Due to the property that information is stored in the donor network only, DNS has a higher query overhead. Directory numbers are the key in ENUM. As depicted in Figure 9.2, the resolver queries the local name server, which recursively queries other name servers at decreasing hierarchical levels.

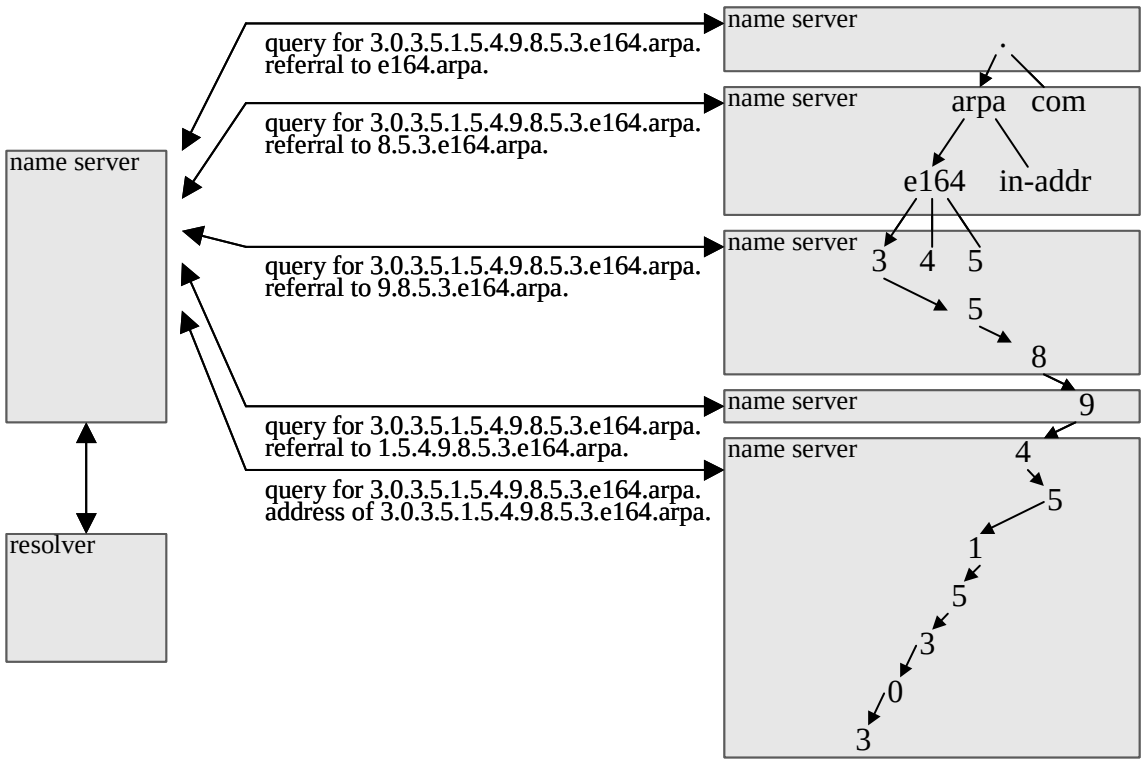

Figure 9.2: ENUM query procedure

The number of queries required for an ENUM lookup depends on the organization into tiers. Let R denote the number of round-trip times (queries) needed for one lookup. In Figure 9.2, five recursive queries are required, which can be considered as a worst case. In this case, there are six round-trip times in total (R=6) because of the additional query from the resolver to the local name server.

The performance is improved by using a cache. For an entry located in the cache of the local name server, only one round-trip time is needed. The average number of round-trip times is thus

$$R_{cached} = P_{hit} + (1 - P_{hit})R, \tag{9.8}$$

where P_{hit} is the probability that the requested entry exists in the cache, and R is the number of round-trip times needed if it is not in the cache (we assume R=6). A formula for calculating P_{hit} can be found in [Lin 1999]:

$$P_{hit} = \frac{p_{h1}\lambda_1 N_1 + p_{h2}\lambda_2 N_2}{\lambda_1 N_1 + \lambda_2 N_2} \tag{9.9}$$

Here, the numbers are grouped into two classes: frequently accessed and less frequently accessed numbers. The call arrivals to class i form a Poisson process with rate λ_i. N_i is the number of entries in class i. p_{hit} is the cache hit probability of class i. According to [Lin 1999], P_{hit} is between 10% and 20% if the cache size is 20% of the number of frequently accessed ported numbers. The efficiency of caching also depends on the lifetime of the entries. Since number portability events are infrequent, the lifetime value can be high. If the move is known to the donor operator a few days before it comes into effect, the lifetime of the DNS entry can be set to match the actual remaining time exactly.

We can draw the conclusion that DNS relocates the load of distribution to the query: there is no distribution but the query is heavy. With a larger portion of the entries cached, DNS can be seen as having an on-demand type of distribution mechanism. Only required entries are transferred to the caches, and they are removed when their lifetime is exceeded. With an appropriate choice of lifetime and a sufficiently large cache, the DNS solution can be very efficient.

9.2.5 Summary of properties

Table 9.2 shows a summary of the properties of primary mapping methods. Since a separate entry is required for each ported number, database size and distribution efficiency are important properties for the primary mapping. Therefore, DNS and database mappings are preferred to TRIP/CTRIP. In the cases where no secondary mapping is used, the primary mapping must also be able to perform gateway selection. Network-path routing is supported only by TRIP/CTRIP, which unfortunately scales poorly. Consequently, network-path routing requires a secondary mapping. Additionally, the political question of who maintains the number portability information is important.

Table 9.2: Properties of primary mapping methods

	DB	DNS	TRIP/CTRIP
Database maintainer	Third party	Donor operator (or third party)	Distributed
Critical point	Third party's network	Donor operator's network	None (any network)
Consequence of failure of the critical point	Updates delayed	Inaccessible information	Routes through failed network become unavailable
Regulator database size [1)]	S	0 [2)]	0
Operator database size [1)]	S	S_{own}	$S \cdot (N_{peers} + 1.7)$
Total database size [1)]	$\Sigma S + S$	S	$\Sigma\, S \cdot (N_{peers} + 1.7)$
Round-trips for one query	1	$P_{hit}+(1-P_{hit})R$ $R \approx 6$	1
Entry representation	Master: XML HUT: SQL	-	Binary

[1)] Number of entries

[2)] The regulator maintains higher-level tiers of comparatively small size

9.3 Evaluation of secondary mapping methods

Secondary mappings are used in schemes with intermediate identifiers. Let X denote the intermediate identifier. This identifier can be a TAD or routing number. Let A denote the identifier type of the routing address. The routing address is an IP address in the IP network and a routing number in the SCN.

The mapping methods covered by this evaluation are:

- Database mappings, $X \xrightarrow{DB} A$
- Static mappings, $X \xrightarrow{static} A$
- TRIP/CTRIP mappings, $X \xrightarrow{TRIP/CTRIP} A$
- Recursive DNS mappings, $X \xrightarrow{*DNS} A$

9.3.1 Administration and security

Secondary mappings have the distinct property that they can be implemented in different ways in different networks. However, routes crossing networks with a network-path routing method are only possible for networks sharing information.

The secondary mapping can be implemented in numerous ways with databases and static mappings. In the simplest implementation, each operator maintains its own mapping. Thus, when a route is required to a specific network, the operator queries an internal database, which returns the routing number or URI. We refer to this as static mapping, since there is no controlled distribution of information between operators. With static mappings, only the originator-determined policy model can be implemented.

On the other hand, if the information is shared between operators and identical information is available for all operators, we call it a shared database mapping. The information can be contained in a central database, which is queried by all networks; contained in a central database, which is cached in all operators' networks; or completely distributed without a central database. Common to these approaches is that the information available to all operators is identical.

Static mappings are simple to implement, but require manual exchange of routing information. Since changes in the routing information are relatively infrequent (on the application layer), a static secondary mapping causes much less inconvenience than a static primary mapping. Only when a peer operator changes its signaling server address or routing number, the mapping must be updated. The mappings are internal to the network, which makes them secure from external modifications. In a database mapping, the update is automated at the expense of higher complexity. The operator whose routing address changes can update the information in the database and the new information is available to other operators immediately. Security of databases depends extensively on the implementation.

The TRIP and CTRIP protocols have the same properties when used for the secondary mapping as when used in the primary mapping. The only difference is that routes point to networks instead of to individual directory numbers. The destination networks are identified either directly through TAD identifiers or indirectly through aggregated routing numbers.

9.3.2 Selection mechanism

Gateway location is a mapping that returns the address of a gateway. The method may include gateway selection, whereas the method includes the policies that select the most suitable gateway. The method may also rely on external selection, whereas the method returns a list of addresses and the client performs the selection according to its policies.

TRIP and CTRIP include gateway selection as part of the protocol. On the other hand, the recursive DNS mapping is purely a gateway discovery mechanism, and the selection can be vendor specific (though, respecting the rules about mandatory servers).

9.3.3 Network-path routing support

TRIP, CTRIP and recursive DNS support network-path routing in the IP network. The result is a route consisting of the networks on the path between the caller and the destination. Each network is represented by a signaling server. Due to policies, some networks, however, may omit their

address from the path. In contrast, the database solutions and static mappings do not support network-path routing.

9.3.4 Database size

The estimation of database size for the secondary mapping is not as straightforward as for the primary mapping. The size of the input space of the secondary mapping depends on the type of intermediate identifier. If TADs are used, the number of entries equals the number of TADs in the number portability routing area. It is also possible to have routes to TADs in larger areas, and even to all networks in the world. Depending on how identifiers are allocated, it may be possible to aggregate TAD identifiers. The number of entries of an intermediate routing number depends on the format of the routing number. In principle, it is possible to reduce the number of entries through aggregation to a single entry for each network. In that case, the number of entries equals the number of networks. Even in this case, it is possible to incorporate a larger area, whereas the number of entries equals the number of networks in that area. Global routing numbers have the additional benefit, that they include the country identity in the beginning of the number, which allows representation of each country with a single entry.

In any case, the formulas on database size made for the primary mapping are valid for the secondary mapping as well. Let the number of entries in the input space be S. For TRIP/CTRIP mappings, the size of one operator's databases is given with equations 9.4 and 9.6 for different implementations. The corresponding sizes for the total database sizes are given in equations 9.5 and 9.7. For database solutions, the total size is given by equation 9.2. The recursive DNS has the same database size as the ordinary DNS, given in equation 9.3.

9.3.5 Query performance

In the IP network, the queries are performed by the signaling server, the location server or the client itself. In the SCN, the query is performed by the exchange or by an IN element. The same calculations on the required number of round-trip times also apply when a database, TRIP/CTRIP or DNS is used for the secondary mapping. Static mappings are also considered as requiring one round-trip time for obtaining the routing address.

However, the recursive DNS method has a multiple times higher number of round-trip times than normal DNS. In section 6.4.4, we observed that an upper value for the number of lookups per call is $Q = D^{L-2}$, where D is the average number of neighbors per domain and L is the average number of domains on the path. We estimated a practical upper limit of $Q \approx 23$. Observing caching, we estimated that $P_{hit}+(1-P_{hit})R$ DNS-queries are required for a lookup. Thus, for a recursive DNS mapping, the total number of queries is

$$Q_{*DNS} = D^{L-2}\left(P_{hit} + \left[1 - P_{hit}\right]R\right) \tag{9.10}$$

Translation from directory numbers (R=6) stored according to ENUM gives approximately $Q_{*DNS} \approx 23 \cdot 6 = 138$ queries, which is too many. However, translation from TAD identifiers (R=4) is

more feasible because of the lower number of hierarchical levels, as shown in Figure 9.3. One lookup requires one query from the resolver to the local name server and three recursive queries; in total R=4 round-trip times. An approximate upper value for the number of queries for a lookup is $Q_{*DNS} \approx 23 \cdot 4 = 92$, which still is high. Since the number space of TADs is smaller than the E.164 number space, the cache hit probability increases and cache sizes are smaller. Because of aggregation, also routing addresses perform better as identifiers than directory numbers.

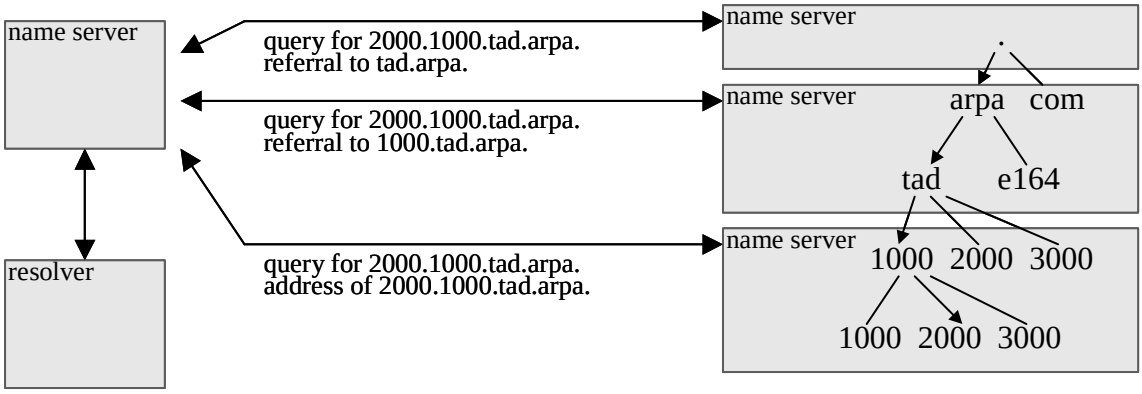

Figure 9.3: TAD query procedure

The performance is improved by using parallel queries. Then $Q_R = L$-1, and the total number of query rounds is:

$$Q_{*DNS,R} = (L-1)(P_{hit} + [1-P_{hit}]R) \tag{9.11}$$

Assuming a path length L = 3.5 as in section 6.4.4, mapping from a directory number requires approximately $Q_{*DNS,R} \approx 2.5 \cdot 6 = 15$ query rounds, and mapping from TAD identifiers requires approximately $Q_{*DNS,R} \approx 2.5 \cdot 4 = 10$ query rounds.

9.3.6 Summary of properties

Table 9.3 shows a summary of the properties of secondary mapping methods. In the secondary mapping, scalability and database size is less important than in the primary mapping. Instead, the emphasis is on gateway selection, route selection and administrative features. Schemes with a secondary mapping can benefit from network-path routing, e.g. with recursive DNS or TRIP/CTRIP.

Table 9.3: Properties of secondary mapping methods

	Static	DB	*DNS	TRIP/CTRIP
Database maintainer	Separate DB for each operator	Specific to implementation	Distributed	Distributed
Critical point	Specific to implementation	Specific to implementation	None (any network)	None (any network)
Consequence of failure of the critical point	Specific to implementation	Specific to implementation	Routes through failed network become unavailable	Routes through failed network become unavailable
Security	High	Specific to implementation	High	Low to medium
Operator database size [1)]	S	S	S_{own}	$S \cdot (N_{peers} + 1.7)$
Total database size [1)]	ΣS	ΣS	S	$\Sigma S \cdot (N_{peers} + 1.7)$
Round-trips for one query (TAD)	1	1	$(L-1)[P_{hit}+(1-P_{hit})R]$ $R \approx 4$	1
Round-trips for one query (RN or DN)	1	1	$(L-1)[P_{hit}+(1-P_{hit})R]$ $R \approx 6$	1

[1)] Number of entries

[2)] The regulator maintains higher-level tiers of comparatively small size

9.4 Summary

In this chapter, we provided an evaluation of mapping methods used in the primary and secondary mappings. For both types of mappings, administration and security aspects, database size and query performance were compared. As the primary mapping implements number portability, the porting procedures of the different methods were compared. Some methods support network-path routing and gateway selection in the secondary mapping.

Chapter 10
Conclusions and further work

In this work, we have examined the problems of telephony routing, gateway location and number portability in hybrid SCN/IP networks. Since the problems are interrelated, the focus has been on developing complete scenarios instead of only solving the contributing partial problems.

We started by studying the state-of-art of routing in the SCN and IP networks, and solutions to number portability to SCN. We saw that these solutions aim to solve a single partial problem, while in hybrid scenarios they lead to inefficient routing, especially for ported numbers. We observed that gateway location and number portability can be described as mappings, which can be performed with various methods. We defined the requirements for the use of routing addresses. Then, we provided a classification for different scopes of information distribution and described the functions that can be implemented with the additional information.

In the following chapters, we presented different mapping methods and discussed their applicability to hybrid SCN/IP scenarios. We provided extensions to existing methods that allow implementation of mappings beyond the original purpose of the method. The proposed extensions include the use of DNS (ENUM) for number portability, DNS for gateway location and DNS for TAD mappings. To this category also belongs the CTRIP protocol, which extends the TRIP protocol to the SCN. CTRIP has been thoroughly presented in the author's earlier work. The aim of these extensions is to distribute information across technology borders and to provide more functionality with a single protocol.

By using an intermediate identifier, the number portability and terminal/gateway location mappings can be separated. This improves aggregation efficiency and reduces dependency. Based on the use of intermediate identifiers, five feasible schemes were selected and examined. Finally, the applicability of different methods for the primary and secondary mapping was evaluated. The considered issues included administration, database size, performance, porting procedure and gateway selection mechanism.

10.1 Conclusions and discussion

After reading this work, one question remains unanswered: which scheme and which method is the best? For each scheme, a number of different methods can be applied in different

combinations. We have rather been broadening the selection of alternatives than reducing it. In the first part of this work, we extended the existing mapping methods by modifying them to be used with other identifiers and in another technology than the original intention was.

Partly, the aim has been to show that number portability and gateway location are fundamentally mappings, which can be performed with practically any method. More important than the choice of method is how the mappings are combined and how they share information. By performing number portability and gateway/terminal location in separate mappings with an intermediate identifier, aggregation can significantly reduce database contents. The reduced number of entries is a prerequisite for network-path routing with TRIP/CTRIP or recursive DNS. The primary mapping is thus preferably a quick and scalable mapping from the directory number to an aggregated routing number or network identifier, and the secondary mapping performs more heavyweight operations, such as gateway location or routing based on various parameters. Ideally, information is exchanged between all networks independently of the network technology. Thus, all networks within a number portability routing area have access to similar information. The gateway location mapping allows groups of operators to share information, but routing efficiency increases for larger groups.

The mapping methods can be classed as *remote access methods*, *replication methods* or *routing protocols*. DNS is an example of a remote access method, since all information is stored in remote servers and the requesting network has no own copies of it (except for the caches). Replication methods maintain updated copies of the information in each network, and the information is accessed locally. A database mapping can be a remote access or replication method depending on the implementation. Both the studied database methods, the Master system and the HUT database solutions, are replication methods. In contrast to the other two classes, routing protocols may store different views of the information in different locations. In TRIP and CTRIP, the view is determined by the policy of each network.

We see that the selection of method is largely determined by the existence of standards and implementations. According to Metcalfe's Law, the usefulness of a network equals the square of the number of users [Gilder 1993]. Therefore, we have favored solutions based on DNS since it is standardized, widely adopted, easily extended and it is already part of telephony routing due to ENUM. However, DNS has the distinctive property that it is a remote access method – not a replication method. Because of the hierarchy, the number of queries related to a lookup can be high, and caches must be used to speed up lookups and reduce server load. Caches again increase the risk of stale information. While caches are efficient for single hosts accessing a few destinations frequently, an exchange accesses specific destinations proportionally seldom.

On the other hand, shared databases are easy to replicate. The queries are made to local copies in a lightweight manner, but the replication process may be heavy. Therefore, databases are suitable for applications where information is frequently accessed but seldom modified, e.g. for number portability. However, databases are very general and in order to adopt a database solution

globally, standardization is necessary to determine the used fields, the replication procedure, the security implementation and the access interface. Because of the lack of standards, no global databases for telephony are available in addition to the rather unsuccessful X.500 directory. On a national level, shared databases can be implemented, as has been shown by the Master system.

The threshold to adopt the TRIP and CTRIP protocols is high because of their complexity. Since they are routing protocols, adoption in several networks is required for them to be useful, and this critical mass is difficult to build up in global scenarios. We also described the scalability problems that number portability causes. Considering the scalability issue, these protocols are not suited for the primary mapping. Instead, the protocols are best utilized in the secondary mapping for groups of operators sharing gateways, or within a large network divided into partitions. The strength is in the fact that they are routing protocols, which are able to generate efficient routes and adapt to the network topology. TRIP and CTRIP are also the only considered methods that support the path-determined policy model.

We expect that number portability will be a national issue, which can be solved in different ways in different countries. An implication of this is that routing numbers are valid nationally, and the NP-mapping is performed for all incoming international calls. Independently of the used number portability mapping methods, a destination residing in the IP network must have an ENUM entry. A telephone number for which there is no ENUM entry is assumed to be an SCN destination (even if it were an IP destination) and calls to this numbers are routed through the SCN.

A good comprehensive solution provides scalable number portability across both the network technologies in a secure, cost-efficient and manageable way. Security and coordination is best provided by a closed system, such as the Master system. In these, the regulator has an important role as a mediator. TRIP and CTRIP fail as number portability methods due to scalability problems. DNS is a considerable alternative for the number portability mapping if security and coordination between operators can be provided. It integrates with ENUM and the infrastructure already exists. However, security of DNS is still weak, and e.g. a DoS attack could have a severe effect on a DNS-dependent telephone system. The porting procedure is also undefined – we assumed that the current operator directly communicates with the donor operator to indicate that a number has moved, but an alternative approach could include the regulator as a mediator. Thus, currently database mappings are a more mature and feasible solution than a DNS-based system. However, on long term, if IP telephony replaces the SCN and ENUM becomes the method for locating IP destinations, it seems natural to store information about all destinations in ENUM instead of using separate national number portability databases for SCN destinations.

Further, a comprehensive solution must enable policy-based gateway location and future seamless routing between different technologies. Instead of proposing a single secondary method, we prefer to leave this decision to the operators – thus, we propose to separate the number portability mapping from the routing and gateway location mapping. This allows different secondary mappings to be used in different networks. It also allows operators to use static

mappings and proprietary methods. Our vision is to enable sharing of gateways and routing information within a group of networks. Each of these groups may utilize a different method, for example TRIP/CTRIP. For such separate secondary mappings to be possible, the interface between the mappings, i.e. the intermediate identifier, must be well defined. We suggest that the format of the intermediate identifier should be technology independent. Routing numbers and IP addresses are both related to a specific technology. The TAD identifier is independent of the technology but it only identifies the network – it does not contain the identity of the subscriber.

Although not considered in this work, a more complex intermediate identifier could include information about the country, the technology, the network and the subscriber. This type of identifier could be implemented as an extended routing number composed of the normal country code, a digit indicating the technology, some digits indicating the network/operator and a subscriber identifier. The subscriber part would be the normal routing number in the SCN or a coded IP address (e.g. A.B.C.D represented as AAABBBCCCDDD) in the IP network. The benefit of this type of intermediate identifier is that it can be used as an actual routing number in the SCN, and easily translated into an IP address in the IP network. It is also globally valid.

10.2 Future research

In this work, we have considered a few selected types of identifiers. For example, the 128-bit addresses of IP version 6 have been intentionally omitted, since they are still not in widespread use. It would be possible to use IPv6 addresses as intermediate identifiers, and the larger address space significantly increases the availability of new addresses. There are still administrative problems in using IP addresses in the SCN. Furthermore, this work has mainly considered TAD identifiers as network identifiers. Generally any type of identifier, aggregatable or not, could be used to indicate the network. We also briefly discussed about extending routing numbers with additional information.

Gateway location and number portability with ENUM have been discussed from a theoretical point of view. The aim has been to show the possibilities of DNS beyond current applications. No practical implementations or measurements have been performed as part of this work. These should be verified and tested with prototypes.

The deployment of ENUM is still ongoing, and especially the administrative and management issues are unclear. Several countries are currently testing ENUM with pilot projects. IP telephony is still relatively new and only a few operators offer IP telephony commercially. The amount of IP calls is low, and no advanced gateway location has been needed. It is difficult to predict the long-term commercial and technical development of IP telephony. Furthermore, the amount of ported numbers is still low although number portability is becoming more popular. Therefore, the value of efficient routing in hybrid network is so far low. However, the requirements of routing in hybrid networks should be observed in an early stage, since it might be difficult to change established implementations later. The importance of efficient routing is expected to increase as both IP telephony and number portability become more common in a few years.

References

[3GPP] The 3rd Generation Partnership Project, www.3gpp.org

[3GPP TS23.228] 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects, "IP Multimedia Subsystem (IMS); Stage 2 (Release 5)", TS 23.228 V6.0.0, January 2003

[Albitz 1997] Paul Albitz, Cricket Liu, "DNS and BIND", Second Edition, January 1997,O'Reilly & Associates Inc, Sebastopol, ISBN 1-56592-236-0

[Beijar 2002] Nicklas Beijar, "Distribution of Numbering Information in Interconnected Circuit and Packet Switched Networks", Master's Thesis, Espoo, January 30, 2002

[Bos 2001] Lieve Bos, Suresh Leroy, "Toward an All-IP-Based UMTS System Architecture", IEEE Network, January/February 2001

[Chao 1999] Han-Chieh Chao, Tin Yu Wn, Chang, S.W., Reen-Cheng Wang, "The network topology based domain name service", 1999 International Workshops on Parallel Processing, Proceedings, 1999, pp. 528 –533

[Eastlake 1999] Donald E. Eastlake 3rd, "The Kitchen Sink DNS Resource Record", IETF Internet draft, September 1999, expired in March 2000, http://www.ietf.org/proceedings/99nov/I-D/draft-ietf-dnsind-kitchen-sink-02.txt

[ENUM-forum 2003] ENUM Forum, "ENUM Forum Working Document: ENUM Forum Specifications for US Implementation of ENUM", 14.3.2003, Work in progress, http://www.enum-forum.org/documents/6000_1_0.pdf

[Faltstrom 2003] Faltstrom, P., Mealling, M., "The E.164 to URI DDDS Application (ENUM)", draft-ietf-enum-rfc2916bis-03.txt, 22 January 2003, Internet draft, Work in progress

[Ficora 1997] Viestintävirasto, “Siirrettävyyden toteutusohje 1”, Viestintäviraston työryhmäraportti 6/1997

[Ficora 1998] Viestintävirasto, “Heksadesimaalinumerot televerkoissa”, Viestintäviraston työryhmäraportti 2/1998

[Ficora 1999] Viestintävirasto, “Siirrettävyyden toteutusohje 4 (palvelunumeroiden siirrettävyys)”, Viestintäviraston työryhmäraportti 5/1999

[Ficora 2001] Viestintävirasto, ”Yleisen valintaisen televerkon numeroinnista”, THK 32 E/2001 M, February 2001

[Ficora 2002a] Viestintävirasto, “Matkapuhelinnumeron siirrettävyys, tekninen verkkototeutus”, Viestintäviraston työryhmäraportti 10/2002

[Ficora 2002b] Viestintävirasto, ”Puhelinnumeron siirrettävyyden master-järjestelmän määrittely”, Viestintäviraston työryhmäraportti 11/2002

[Ficora 2002c] Viestintävirasto, ”Matkapuhelinnumeron siirrettävyyden proseduurit”, Viestintäviraston työryhmäraportti 14/2002

[Ficora 2003] Viestintävirasto, ”Määräys puhelinnumeron siirrettävyydestä”, Viestintävirasto 46 A/2003 M, 28.5.2003

[Gilder 1993] George Gilder, “Metcalf’s law and legacy”, Forbes ASAP, September 1993

[IANA TLD] Internet Assigned Numbers Authority, “Root-Zone Whois Information”, http://www.iana.org/cctld/cctld-whois.htm

[ITU-T CC] International Telecommunication Union, Telecommunication Standardization Sector, “List of ITU-T Recommendation E.164 Assigned Country Codes”, 1 May 2002, Geneva

[ITU-T E.164] International Telecommunications Union Telecommunication Standardization Sector, “The international public telecommunication numbering plan”, ITU-T Recommendation E.164, Geneva, May 1997

[ITU-T H.323] International Telecommunications Union Telecommunication Standardization Sector, Study group 16, “Packet-based multimedia communications systems”, ITU-T Recommendation H.323, February 1998

[Kantola 2001] Raimo Kantola, Jose Costa Requena, Nicklas Beijar, “Interoperable routing for IN and IP telephony”, Computer Networks, Volume 35, Issue 5, pp. 597-609, April 2001

[Labovitz 1999] Craig Labovitz, Robert Malan, Farnam Jahanian, “Origins of Internet Routing Instability”, Proceedings of INFOCOM, IEEE, June 1999, www.ieee-infocom.org/1999/papers/02b_02.pdf

[Liikenneministeriö 2000]

Liikenneministeriö, “Televiestintätilasto 2000”, 2000, Helsinki

[Lin 1999] Yi-Bing Lin, Herman Chung-Hwa Rao, “Number Portability for Telecommunication Networks”, IEEE Network, January/February 1999

[Lin 2003] Yi-Bing Lin, Chlamtac I, Hsiao-Cheng Yu, “Mobile number portability”, IEEE Network, Volume 17, Issue 5, September-October 2003, pp 8-16

[Nokia 2001] Nokia Corporation, “MITA – Mobile Internet Technical Architecture”, 2001

[Olsson 2002] Lars Olsson, Per Bergman Lidebrandt, “Routing av IP telefoni med TRIP”, Master’s Thesis, Stockholm, January 2002,

[Paju 2002] Antti Paju, “Tilaajanumeron siirrettävyys yhdistetyssä piiri- ja pakettikytkentäisessä verkossa”, Master’s Thesis, Espoo, December 3, 2002, ISBN 951-22-6289-4

[Rahnema 1993] Joe Rahnema, ”Overview Of The GSM System and Protocol Architecture”, IEEE Communications Magazine, April 1993

[RFC 1034] Mockapetris, P., ”Domain Names – Concepts and Facilities”, RFC 1034, November 1987

[RFC 1035] Mockapetris, P., ”Domain Names – Implementation and Specification”, RFC 1035, November 1987

[RFC 1771] Y. Rekhter, T. Li, “A Border Gateway Protocol4 (BGP-4)”, RFC 1771, March 1995

[RFC 2234] Crocker, D., ”Augmented BNF for Syntax Specifications: ABNF”, RFC 2234, November 1997

[RFC 2396] T. Berners-Lee, R.T. Fielding, L. Masinter, “Uniform Resource Identifiers (URI): Generic Syntax”, RFC 2396, August 1998

[RFC 2543] M. Handley, H. Schultzrinne, E. Schooler, J. Rosenberg, “SIP: Session Initiation Protocol”, RFC 2543, 1999

[RFC 2806] Vaha-Sipila, A., ”URLs for Telephone Calls”, RFC 2806, April 2000

[RFC 2871] J. Rosenberg, H. Schulzrinne, “A Framework for Telephony Routing over IP”, RFC 2871, June 2000

[RFC 2916] Faltstrom, P., “E.164 number and DNS”, RFC 2916, September 2000

[RFC 3219] Rosenberg, J., Salama, H., Squire, M., ”Telephony Routing over IP (TRIP)”, RFC 3219, January 2002

[RFC 3344] Perkins, C., “IP Mobility Support for IPv4”, RFC 3344, August 2002

[RFC 3401] Mealling, M., “Dynamic Delegation Discovery System (DDDS) – Part One: The Comprehensive DDDS”, RFC 3401, October 2002

[RFC 3402] Mealling, M., “Dynamic Delegation Discovery System (DDDS) – Part Two: The Algorithm”, RFC 3402, October 2002

[RFC 3403] Mealling, M., “Dynamic Delegation Discovery System (DDDS) – Part Three: The Domain Name System (DNS) Database”, RFC 3403, October 2002

[RFC 3404] Mealling, M., “Dynamic Delegation Discovery System (DDDS) – Part Four: The Uniform Resource Identifiers (URI) Resolution Application”, RFC 3404, October 2002

[RFC 3405] Mealling, M., “Dynamic Delegation Discovery System (DDDS) – Part Five: URLARPA Assignment Procedure”, RFC 3405, BCP 0065, October 2002

[RFC 3482] Foster, M., McGarry, T., Yu, J., “Number Portability in the Global Switched Telephone Network (GSTN): An Overview”, RFC 3482, February 2003

[Rosenberg 1998] Jonathan Rosenberg, Henning Schulzrinne, “Internet Telephony Gateway Location”, INFOCOM '98, Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings, IEEE, Volume 2, March 29 – April 2, 1998, pp. 488-496

[Rostela 2002] Tuomo Rostela, “Numerointi ja reititys operaattoritasoisessa hybriidiverkossa”, Master’s Thesis, Helsinki, November 11, 2002

[Schulzrinne 2002] Schulzrinne, H., Vaha-Sipila, A., ”The tel URI for Telephone Calls”, draft-antti-rfc2806bis-07.txt, 5 December 2003, Internet draft, Work in progress

[THK 1996] Telehallintokeskus, “IN-tekniikkaan perustuva puhelinnumeron siirrettävyys”, Telehallintokeskuksen julkaisuja 5/1996, Helsinki, 25.3.1996

[Varshney] Upkar Varshney, Andy Snow, Matt McGivern, Christi Howard, ”Voice Over IP”, Communications of the ACM, January 2002, Volume 45, Issue, pp. 89-96 1