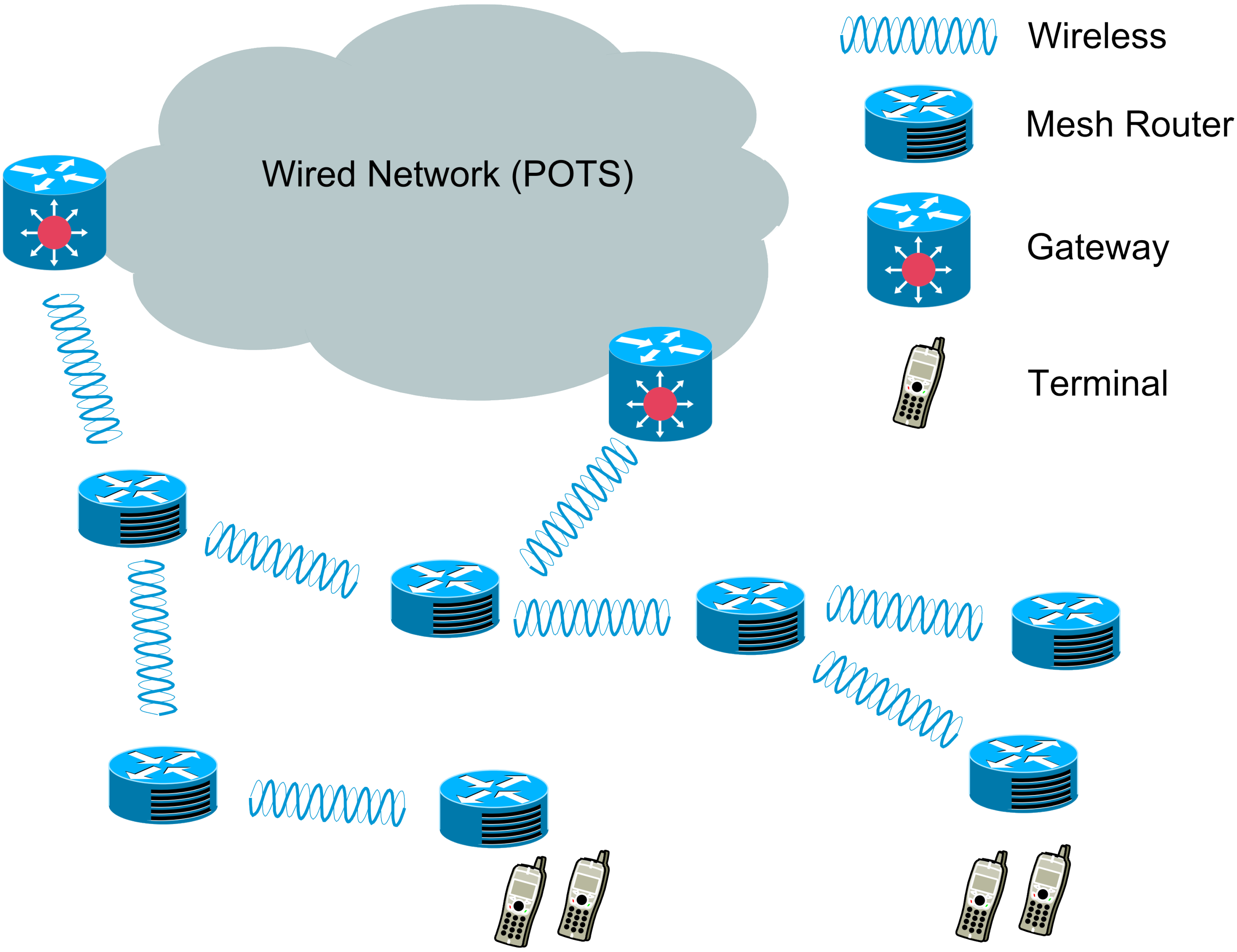
NDS Diplomarbeit

Distributed Terminal Location Discovery in a TOWN Network



Distributed Terminal Location Discovery in a TOWN Network

1. Abstract

The TOWN project aims to build a self organizing wireless metropolitan network. As a part of TOWN, we address how the network could keep track of individual terminal locations (association to a base station). We aim to define a set of evaluation criteria and metrics based on which we recommend location discovery schemes suitable for TOWN. We found, that due to the relatively small size of the considered network all discussed schemes may be implemented. It is recommended that either DUNDi or ENUM are chosen for the implementation in TOWN.

2. Table of Contents

3. Introduction

Imagine waking up in the middle of the night. After a short moment you find the reason for you unexpected awaking: The building is rocking, there is a light earthquake. As you can't get to sleep anymore you get up and check the Internet[1] to see where the epicenter is. You are alarmed to see that it actually was a > 5 Magnitude near Palm Springs where some of your relatives live. As a matter of course you get on the phone and call them. While you got a dial tone without problems, all numbers close to Palm Springs seem to be busy.
Our imaginary earthquake serves to show, that the urge for communication is especially strong in case of emergencies, or disasters. Now a common occurrence in emergencies is a break down in communication. Commonly available channels fail as they are overloaded or destroyed. Yet people want to know what is happening and tell their families and friends how they fared. Working communication channels are even more important to coordinate rescue mission and other disaster recovery efforts.

In a small scale event, rescue organizations may use existing infrastructure to connect the incident site with their command and control center. On site communication may be provided by walkie-talkies or even messengers. If however a bigger area is affected by, e.g. an earthquake, one cannot rely on any existing infrastructure. In addition communication by walkie-talkie will become more difficult in a lager scale event. These difficulties may arise form interference, multiple organizations using the same channels or simply insufficient radio-discipline. Also typical for a lager scale event is that multiple organizations that would normally work by themselves need to be coordinated.
Faced with existing infrastructure being unavailable and communication via two-way radio difficult, what would the ideal communication system for disaster recovery look like?

Lets build a "wish list":
For disaster recovery a voice communication network, covering a region such as a city or metropolitan area is desirable. If the network can be scaled to some extent that is a definite plus, but scaling to nationwide or even global operation is not desired. The network should probably charter for a few hundreds of users. It is essential that such a network can be deployed within short time prohibiting all but minimal radio- and network planning "on the ground". Not just the "building" of the network, but also the operation of the terminals needs to be possible with only minimal user training. The network may not rely on any existing infrastructure in the area of operation. While our network must not require a connection to any existing network, the possibility to connect to e.g. the public telephony system is desirable. The system should also be highly robust in the sense, that when parts of the network become non functional, the reminder should continue working.
How could such a system be built? The TOWN project (Telephony over Wireless Metropolitan Networks) will answer this question while developing such a network. The TOWN network will use IEEE 802.16 (WiMax) equipment as the "trunk"(network backbone). WiMax is a technology that offers both relatively high bandwidths and long range (expected 2 Mbit/s over 5km). The trunk of the TOWN network will be a mesh network. Such a mesh networks are characterized by the fact, that messages may be relied over multiple wireless hops. Not just the backbone will be wireless but also the terminals. The terminals themselves will not be "full members" of the mesh network, as they will not relay traffic from other terminals.

This thesis focuses only on one specific problem within the TOWN network: How should the network keep track of "in which part of the network" a terminal is to be found.

Why is this an interesting question? As users will physically move around, they will not always be connected to the same part of the network. They still have to be able to originate and terminate calls. In traditional telephony systems a number would always correspond to one specific "set of wires". Together with the fact, that numbers would be assigned geographically, it was easy to route a call correctly. But what do you do, when a given terminal (number you want to reach) could be anywhere in your network at a given time?

A mechanism is needed which either keep track of where all terminals are at all time, or allows us to inquiry any terminals location when needed. This mechanism will from now on be called terminal discovery scheme.

Our network is intended for disaster recovery and should conform to our "wish list". It is not desirable to simply keep one "directory" of terminal locations, as this could become unavailable when parts of the network become non-reachable. This is the reason we will focus on distributed terminal location schemes. To give an overview we will also discuss how existing systems such as GSM, SIP and ENUM solve terminal location.

4. Background

We consider the TOWN project's network architecture as depicted in Figure 1. Each mesh router offers services to the terminals. The area in which a service from a given mesh router is available is called its service domain. Service domains may overlap but only one domain may be used by a single terminal at a time. Additionally, there is no need to be able to switch between service domains (mesh routers) during active use of a service, such as during a call. However, the mesh network has to support mobility of terminals such that they are:

a) reachable form both within the mesh and the wired network
b) they can access resources within the mesh and the wired network

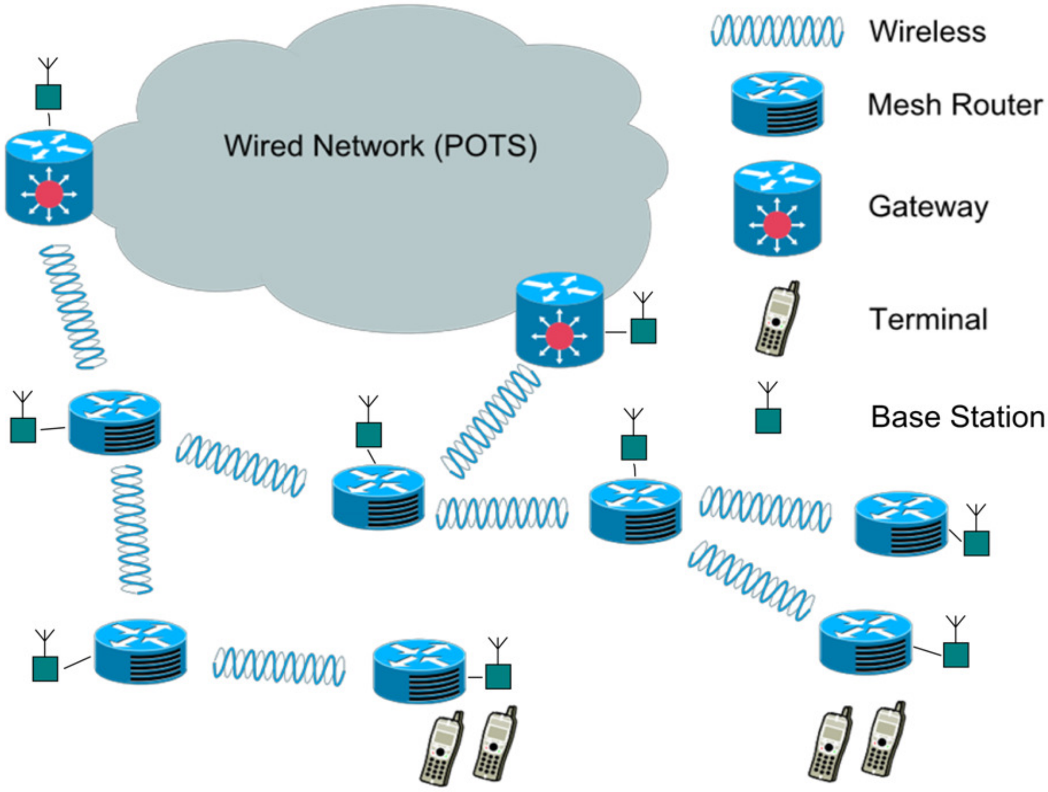


Figure 1: Network architecture of a TOWN Network as discussed in this work. The gateways are mesh routers that don't only connect to the wireless, but also the wired network. Each mesh router and gateway is connected to a base station. The base stations provide services to a number of terminals. At any time a terminal can only be associated with one base station – router pair.

To enable this kind of mobility for the terminal, a means to discover the current location of each terminal must be available. We identify the terminal by a name and the terminal's location by an address[1].

This work discusses schemes that allow the network to find the location of any terminal given its name. These schemes are called location discovery schemes[2] from now on. The following are desired properties of a location discovery scheme: no single point of failure, scalable with respect to number of terminals, scalable with respect to number of routers, robust with regards to router failure, low overhead in terms of necessary number of interactions and more. For a more complete list refer to chapter 8.

[1] This convention is used throughout this document

[2] Other terms used in the literature are: location discovery mechanism, location discovery service or simply location service

5. Problem Statement

An important part in a mobile telephony system design is mobility management, i.e. to keep track of the association of mobile terminals to base stations. TOWN networks will be self-organized telephony systems for public safety application scenarios. Obviously, such system have constraints in call establishment delay and reliability. These constraints may be difficult to meet since the back haul mesh network is wireless.

We thus compare three generally different schemes for distributed terminal location discovery in such a network.

1a) Flooding incoming location discovery requests without caching at call setup time.
1b) Flooding incoming location discovery requests combined with caching.
2.) Updating terminal locations in caches (event driven approach).

We evaluate the suitability of these schemes for use in the TOWN project
with the following criteria.

- Max. call setup delay
- Requirements in terms of memory, CPU, network capacity
- Robustness with regards to (i) mobility of terminals, (ii) incoming call rate, (iii) failure of wireless mesh routers.
- Size of the network, number of gateways to the wired network.

5.1 Approach

We study how location discovery is done in existing systems (see chapter 7). The systems treated will be a selection covering mesh networks as well as mobile Telephony networks. Chapter 6 shows how location discovery schemes may quickly be characterized. A table of attributes will help the reader to quickly gain an overview of location discovery in the discussed system. Additionally a graphic representation of key ideas is defined.
To asses location discovery schemes a set of evaluation criteria (chapter 8) and performance metrics are defined (chapter 9). These criteria and metrics are applied to schemes proposed in the assignment for this thesis. The proposed schemes are substantiated and assessed in chapter 10. Following the discussion of the proposed schemes, is a comparison to the existing location discovery schemes covered in chapter 7.

The evaluation performed in chapter 12 leads to the conclusion that most location discovery schemes could be used for the considered network. The results further show that for this particular network, flooding with caching or updating terminal locations in caches are the most promising solutions. With an increasing number of mesh routers (or terminals) updating terminal location in caches will become less feasible. A very interesting question for further investigation is how well a given location discovery scheme scales with the number of mesh routers. While being very interesting, this question is outside the scope of this work, as it was agreed to only consider a network of up to 30 mesh routers.

6. Characterization of Location discovery schemes

To help characterizing the existing discovery mechanisms, we start the discussion with the following:

- A table listing the attributes of the discussed system which are relevant to location discovery.
- A figure showing to what extend the discussed mechanism makes use of three approaches to location discovery, any location discovery mechanism is built.

6.1 System Attributes

Skimming over the attributes allows to quickly identify how location discovery is handled in the discussed system. The attributes are not criteria for evaluation they simply allow a characterization. Below is a quick explanation for each attribute, as used in chapter 7.

Design Goal:
Which was the over all design goal of the system in question?

Maturity:
Where is the mechanism currently used? In **operational networks**? Or is there an existing **reference implementation** available?

Background / Current usage:
Where does the background of the proposed mechanism lie? In what kind of networks is the system currently used? In **wired telephony**, in **Internet technology** or in **P2P?**

Approach to Location Discovery
Which of the key ideas outlined in 6.2 are used for location discovery in this system?

Update Philosophy:
Either **proactive** or **reactive**. When using the proactive approach all nodes taking part in location discovery are informed as soon as a terminal joins the network. While proactive updates help reducing call setup times, they may cause unnecessary traffic.

Structure:
Does the approach use a **hierarchic** structure, or a **flat** structure? May help to asses if it is possible to build some nodes "lighter" e.g. with slower processors, less memory.

Redundancy:
Does the mechanism have redundancy **by design**? May redundancy be **introduced easily**? Or is any redundancy **missing**?

Caching:
Are the caches on **every node,** only on **select nodes** or are there **no caches** at all?

Single Central Authority:
Does the system use a single centralized authority to handle location discovery?

Relies on:
Does the mechanism rely on any other scheme or infrastructure, such as **TCP/IP,** Signaling System 7 **(SS7)** or the domain name system **(DNS)**?

Scalability (in Number of Terminals):
To which size (maximum number of terminals) will the system, as a whole, scale?
Max Call Setup Delay:
How many queries are needed before a connection can be established?

Name / Address
What are the system specific equivalents to our name and address?

6.2 Approaches to location discovery: Flooding, Mirroring, Centrally Registering

Any Location discovery mechanism needs to make use of either one or a combination of the following Approaches to location discovery.
One could **mirror** all available location information to any mesh router. On one hand mirroring alone is inefficient in terms of initial setup, on the other hand some mirroring is highly desired to allow for recovery if one router would fail.
Another approach would be to **flood** the whole network on any lookup. While this would lead to problems when a big amount of terminals initiates a lookup, it is highly robust and easy to implement.
Finally on can **register with a central authority** and have any lookup go directly to this authority. Obviously a central authority does also have some problems. When all terminals have the same authority, the load on that entity could get prohibitive. Also a failure of the authority would result in the terminals registered with that authority becoming unavailable.

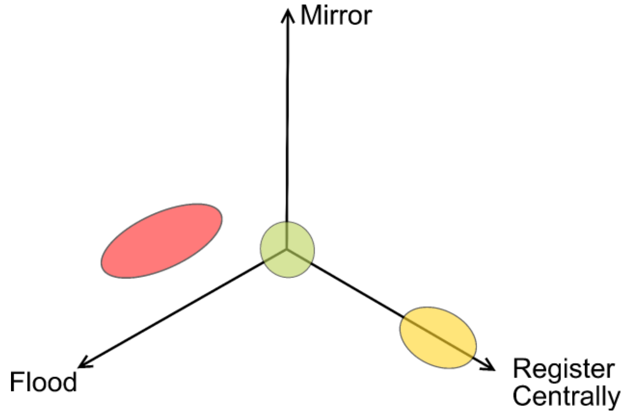


Figure 2: This example shows three different location discovery schemes. The position of the bubble indicates, to what extend a scheme is built on the three key ideas. The red bubble indicates a location discovery scheme that is for a large part based on flooding and for minor part on mirroring, but does not use a central register. The green system uses flooding, mirroring and a central register in equal parts. In contrast the discovery scheme represented by the yellow bulb only uses a central register.

6.3 Other approaches to Characterization

Another possibility to characterize location discovery schemes would be a taxonomy. Here all services are classified according to a set of properties. A very good taxonomy of location discovery schemes is introduced in [2]. We feel that our "characterization" is better suited to describe "Location Discovery in Existing Systems". Furthermore: the special architecture of the TOWN network warrants a different approach. TOWN is special because not all participants in the network are able to forward information. Terminals do not help with routing of messages. With this specific setup TOWN is neither a "full" mesh network, nor a standard mobile communications network.

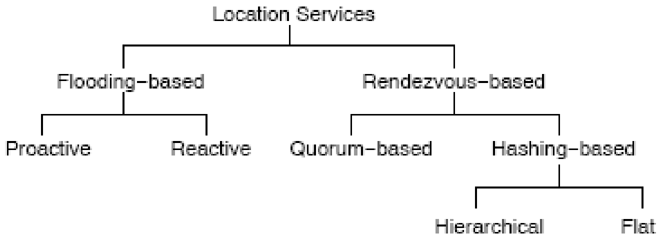


Figure 3 Taxonomy as shown in [2], we think our "characterization" is better suited help understand location discovery in existing systems.

7. Location discovery in existing systems

Next, five well known systems are discussed. Each of these systems makes use of location discovery, even if it is not called by this name. We will identify which parts of the system constitute the location discovery scheme and which key ideas for location discovery are used. Keep in mind, we are not interested in the destinations geographic location, but in the location within the network topology.
We start with one of the more widely used mobile telephone systems, then discuss one of the mayor VOIP Standards, explore two protocols for "call termination" discovery and finally cover a location discovery scheme that was tailored to ad hoc networks.

7.1 Global System for Mobile Communications (GSM)

The Global System for Mobile Communications is a standard with focus on voice communication. It is used by over 2 Billion people worldwide [3]

7.1.1 Characterization

Design Goal	Globally viable mobile communication
Maturity	Operational networks
Background / Current usage	Wired telephony / mobile telephony
Approach to location discovery	Register centrally
Update Philosophy	Proactive
Structure	Hierarchic, two levels: Home Location Register (HLR), Visitor Location Register (VLR)
Redundancy	Missing
Caching	Selected nodes (active VLR)
Single Central Authority	HLR (Per Operator)
Relies on	SS7 Network
Scalability (Number of Terminals)	Very large
Max call setup delay (Numbers of queries)	2 (Originating -> HLR ->VLR = MSRN)
Name	International Mobile Subscriber Identity (IMSI); to the user MSISDN
Address	Mobile Station Roaming Number (MSRN)

Table 1 Classification, characterization of the GSM location discovery mechanism. With its single central authority and reliance on a SS7 network, this approach can not be used in wireless mesh networks. But the system offers mature architecture that is in use world wide.

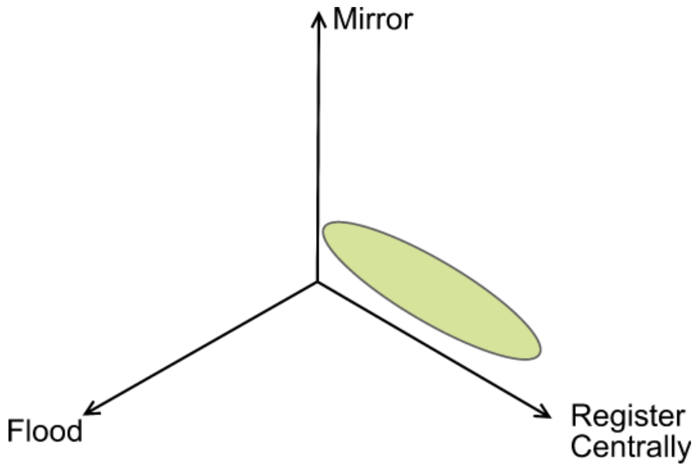


Figure 4: The approach to location discovery used in GSM is to register centrally. The central register is the HLR and the active VLR constitutes one mirror. If the GSM Location Area (LA) encompasses more than one base station flooding is also used for location detection purposes (paging). In this case the VLR broadcast the TMSI of the desired terminal in all base stations within the LA to find the current base station.

7.1.2 GSM Location Update

Components of a GSM Network:
The home location register (HLR) stores all subscriber data and is located in the users home network. An area has a corresponding Visitor Location Register (VLR) which mirrors some records form the HLR but also provides the current terminal address to the HLR. In most cases the VLR is implemented as part of the mobile switching center (MSC) a telephone exchange with special features for use with GSM. We will not separate VLR and MSC here.
Location discovery in GSM is actually not "discovery" but "management". Depending on current state of the terminal, the location information is updated by either a location update (LU) or IMSI attach procedure. The most complex case is a location update which involves two different VLR.[4] This process is shown below.

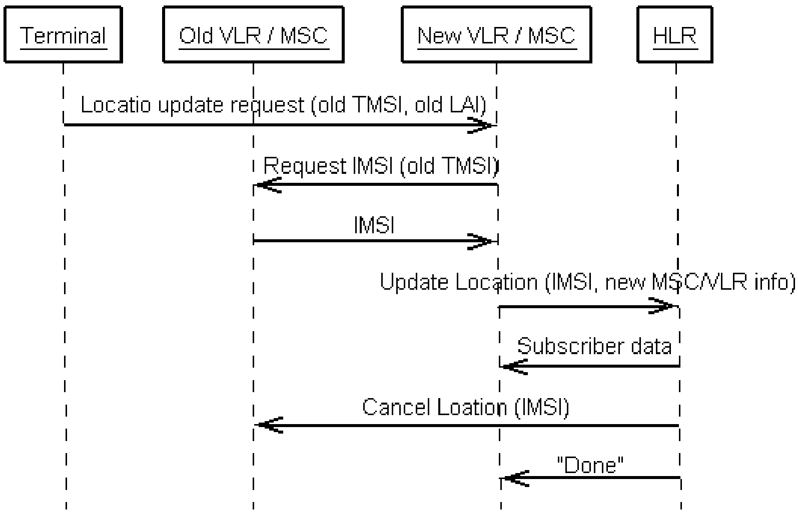


Figure 5: Sequence Diagram for a GSM location update. We assume the terminal would like to switch to a new cell due to it offering better signal strength. The case where the two cells are each associated to a different VLR is shown. This is the most complex location update in a GSM Network. Acknowledges, authentication and connection setup are not shown. The new VLR needs to get the Terminals International Mobile Subscriber Identifier (IMSI) from the old VLR. This is necessary as the Temporary Mobile Subscriber Identifier (TMSI) is local to a VLR. [4],[5]

7.2 Session Initiation Protocol (SIP)

7.2.1 Characterization:

Design Goal	Session Initiation in IP Networks
Maturity	Operational Networks
Background / Current usage	Internet technology (Mobile IP) / VOIP
Approach to location discovery	Register centrally
Update Philosophy	Proactive (whenever User Agent changes location it has to register with the registrar)
Structure	non hierarchic
Redundancy	Missing
Caching	Proxies and User Agents
Single Central Authority	Registrar (per Domain)
Relies on	IP, DNS, location service
Scalability (Number of Terminals)	Very many, as each domain provides its own location service
Max call setup delay (Numbers of queries)	one to the SIP location service, potentially many more to lower Level Network schemes (DNS)
Name	SIP URI
Address	User location

Table 2 The Session Initiation Protocol is an Internet protocol, designed to handle telephony and multimedia sessions. To allow for terminal mobility, the use of a registrar is mandatory. If the SIP location service is a simple database, local to the registrar, then SIP offers the same level of mobility as mobile IP.

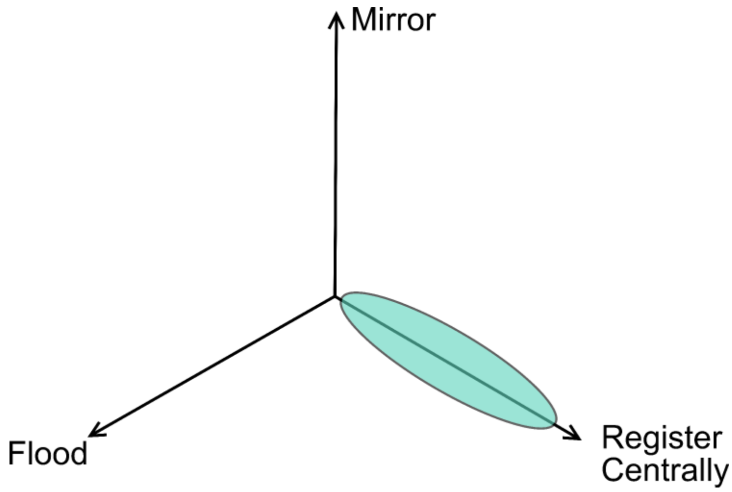


Figure 6 SIP is purely based on the approach to location discovery to have a central register (the SIP Registrar).

7.2.2 Locating an User in a SIP Network

The session initiation protocol or short SIP defines how session may be created, modified and terminated. These sessions may be used for Internet phone calls or multimedia conferences with one or more users. SIP relies on both the Internet's domain name scheme (DNS) and a "location service" to define an users current location. The location service comparable to the location discovery schemes we discuss. How the location service is implanted is not specified by the standard.

"The only requirement is that a registrar for some domain MUST be able to read and write data to the location service, and a proxy or a redirect server for that domain MUST be capable of reading that same data." [6]

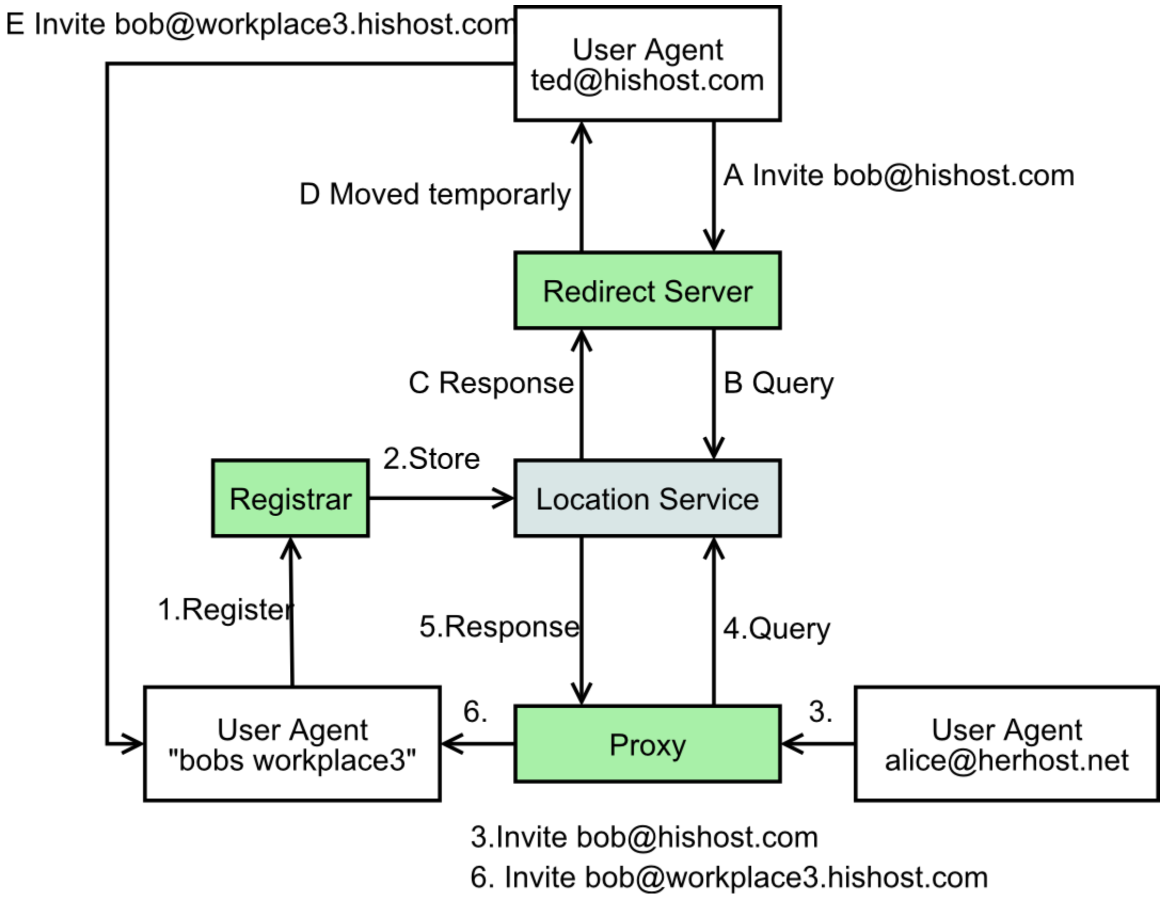


Figure 7: Message flow diagram for two session setups. Steps "1 to 6" shown the case where a Proxy is used. In "A to E" a redirect server is used. Not shown are the DNS system and DNS queries necessary to map host names to IP addresses.

As we have seen above, the location service is not specified in detail by SIP. Given this, SIP may very well be used to initiate a session in our network, but by itself it does not handle locating a user. Any off the evaluated location services could be combined with the session initiation protocol.

7.3 Telephone Number Matching (ENUM)

ENUM is used in VOIP networks to provide (multiple). Termination points for a call to a E.164 phone number.

7.3.1 Characterization

Design Goal	Allow the same Telephone Number to be used in PSTN and for VOIP
Maturity	Operational networks
Background / Current usage	Internet technology (DNS) / VOIP
Approach to location discovery	Register centrally
Update Philosophy	Proactive
Structure	hierarchic
Redundancy	Depending on Implementation
Caching	DNS caches
Single Central Authority	DNS Root server + Domain Servers
Relies on	IP, DNS
Scalability (Number of Terminals)	Good
Max call setup delay (Numbers of queries)	potentially up to 20 (for each digit in number + Root lookups)
Name	E.164 Phone Number
Address	one or more URI (unique resource identifier)

Table 3 Classification, characterization of the ENUM location discovery mechanism. In principle ENUM suffers form it's centralized architecture. But theses problems have already been solved for the DNS, to which ENUM is an extension. Much of the Performance depends on the actual Structure of used DNS configuration. A main Problem could be the update delays that the DNS commonly suffers.

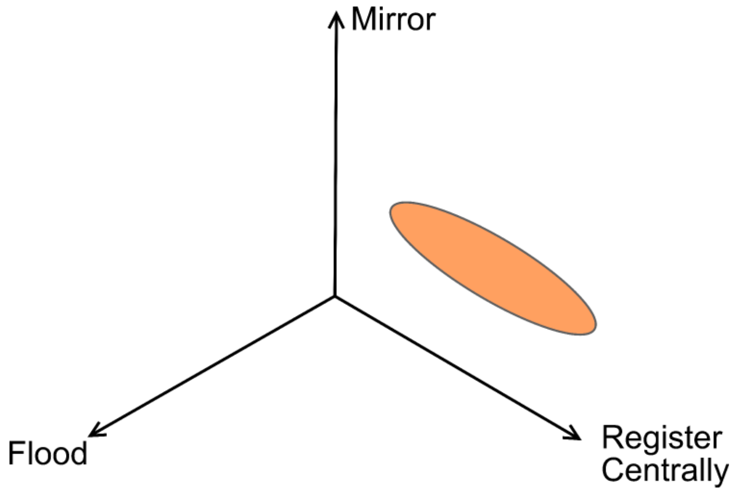


Figure 8 ENUM's approach to location discovery depends mostly on a central register. To some extend mirroring is used by the use of the DNS hierarchy and DNS Server redundancy.

7.3.2 Locating a user with ENUM

One "natural" way to have a telephone number translated into an Internet resource is the use of telephone number mapping short ENUM (as defined in [7]). While ENUM allows to translate a telephone address into a uniform resource identifier (URI) it is hardly a fully fledged location discovery scheme.
You "find" the URI of a user by transforming the users numbers according to this recipe:
Assume +4144/123 45 67; first any characters not representing a number are stripped. You get 41441234567. Then the number is "reversed" and dots are placed between the individual digits. Our example becomes: 7.6.5.4.3.2.1.4.4.1.4. This represents a domain name which may now be looked up in the domain name system. For the domain so called NATR (Net Address Translation Resources) are resolved. These are then converted in to the URI describing the ways the user may be reached. It is not necessarily given that any of the URI's correspond to Voice connections. they may also be voice mail boxes or email addresses.
ENUM is not really suited for real time changes as this would mean the DNS would have to be dynamic. The cache timeouts need to be set really low, such that for each connection a new lookup is initiated. Important is also that the root servers & domain name servers are implemented redundant. For the Internets root servers this is commonly done by using anycast routing. The nameservers themselves can be duplicated with help of "secondary" ore "slave" nameservers [8].
ENUM is also somewhat redundant as the URI that has been looked up with help of the DNS oftentimes contains domain names, that again need to be resolved to IP addresses with help of the DNS.

7.4 Distributed Universal Number Discovery (DUNDi)

According to its creators "DUNDi is a lightweight, low bandwidth, secure, binary encoded protocol for sharing and discovering routes to numbers in arbitrary dial plans." [9] "Dial plans" refers to how Asterisk will route incoming and outgoing calls. In Plain text you could say DUNDi allows for multiple Asterisk instances to act as one big Asterisk instance.

7.4.1 Characterization

Design Goal	ENUM alternative, connect multiple Asterisk PBX (Private Branch Exchange)
Maturity	few operational networks
Background / Current usage	peer to peer / Asterisk, Gizmo Project
Approach to location discovery	flooding
Update Philosophy	No
Structure	non hierarchic
Redundancy	Operational networks
Caching	Yes each DUNDi Node
Single Central Authority	None
Relies on	IP
Scalability (Number of Terminals)	Unknown
Max call setup delay (Numbers of queries)	2 for suggested overlay structure
Name	Number (Phone Number)
Address	SIP-URI, IAX-URI or H.323 Address

Table 4 DUNDi promises to be a "better ENUM". Without reliance on a centralized architecture it would lend itself well to use in mesh networks. However DUNDi has only been adopted by few operators. In addition no performance measurements or simulations could be found.

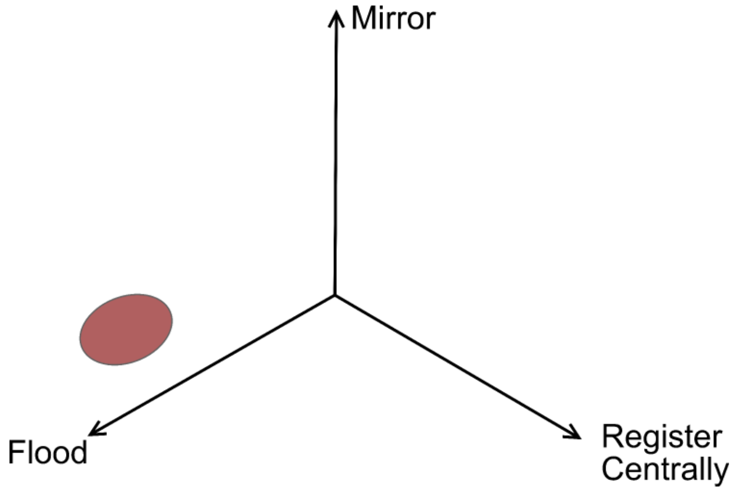


Figure 9 The key ideas behind DUNDi are flooding and mirroring. But flooding is used to a much bigger extent. A lookup will flood the peer to peer overlay network formed by the DUNDi nodes. If the lookup is successful, all nodes that receive the lookup response, will cache (mirror) this information.

7.4.2 Locating an User with DUNDi

The basic idea is that an peer to peer overlay network of DUNDi nodes is formed. The DUNDi Internet draft [9] however does not describe, or handle how the overlay network is constructed. In Asterisk the overlay network is built manually by listing the peers of each DUNDi node in a configuration file. When a lookup is performed, the network is flooded with a preset TTL (Time to Live). Any result is cached in all nodes that receive a response. The default cache timeout of one hour is most likely to long for a wireless mesh network. If a DUNDi node fails, terminals registered with it will be unreachable for parts of the network where a cache exists. Even if the Terminals re-registered with a new DUNDi node.

7.5 Simple location Service (SLS)

7.5.1 Characterization

Design Goal	Provide Location service for Geo-Routing
Maturity	Reference Implementation, no operational networks to our knowledge
Background / Current usage	Related to Internet standard "Routing Information Protocol" (RIP) as Routing tables are mirrored / used as standard example in Papers
Approach to location discovery	Mirroring / Flooding
Update Philosophy	Proactive
Structure	non hierarchic
Redundancy	by design
Caching	in neighbors
Single Central Authority	None
Relies on	no prerequisites
Scalability (Number of Terminals)	better than flooding, as part of lookups can be avoided through sharing of location tables
Max call setup delay (Numbers of queries)	same as flooding, as SLS resorts to flooding if target not found in own table
Name	Identification
Address	(geographical) Location

Table 5 SLS is a proactive location discovery scheme, that shares random parts of its "location table" (Table containing target identifications and locations) with its neighbors. When a location need to be looked up, the local table is checked. In case the table does not contain the desired information, SLS resorts to flooding the network.

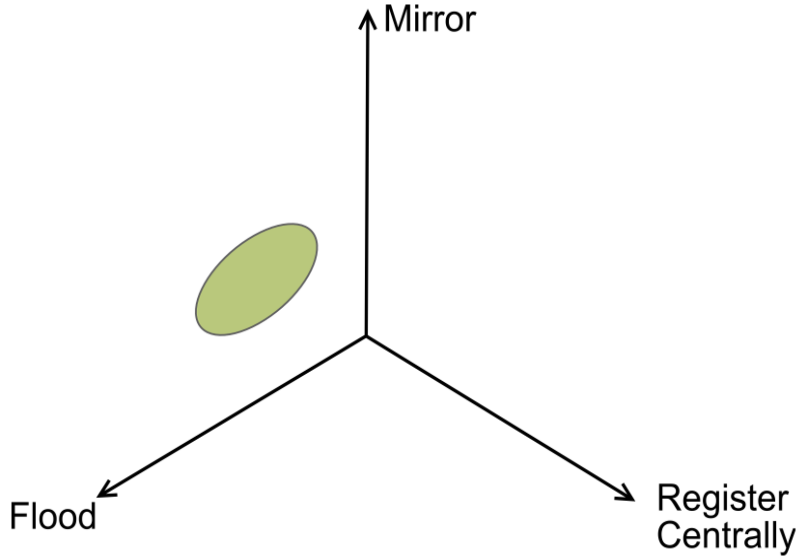


Figure 10 SLS does not use a central registry but relies to equal parts on mirroring and flooding. In case a location is not found within a mirrored table, the protocol resorts to flooding the network.

7.5.2 Locating an User with SLS

In SLS locating all nodes will transmit location packets to its neighbors at a rate depending on how fast it is moving. Each location packet contains a part of the location table, of the node from which the location package originates. Obviously a node will also receive such location packages from its neighbors. The received locations packages are used to update the location table of the receiving node.
When a location discovery lookup (called location request in SLS) takes place, the originating node will first check his own location table. If the desired target identification is not contained in the location table, it will flood the network with a location request.

7.6 Discussion of Location Discovery in existing Systems

In chapter 7 we explored, how location discovery is handled in a wide variety of Systems. Wherever a network needs to cope with parts of the system being mobile, location discovery is needed. Sometimes it may not be called location discovery (as in the case of GSM) but non the less, the network needs to know where (be it geographically or network-topographically) it may reach the destination. We chose the five Systems above mainly for their association with telephony. However SLS has been chosen to exemplarily represent the large number of location discovery schemes developed for sensor-node and other ad-hoc networks. Notable exceptions that have been omitted include GLS, DREAM, GHLS, LLS, SLALoM and mSLP. For an comprehensive overview of location discovery schemes see [10] or [12].

8. Finding Evaluation Criteria and Performance Metrics

To facilitate evaluating location discovery schemes we select set of evaluation criteria and performance metrics.
How do we find suitable criteria and metrics? Lets ask ourself again what the most important features of our desired network are.
The system should be fast to setup. One way to measure this is the **number of messages** that need to be **exchanged** for the network **to converge**.
While in use, the network shall allow the user to set up calls quickly. How fast this is possible can be measured with the **call setup time**. Finding and calling somebody should not only be quick, in addition it shouldn't be to expensive in terms of network traffic caused. We will measure this traffic for one lookup in **number of messages exchanged of a single lookup** (both for the **average**- and **worst case**).
Once set up and in use, the system should be highly robust. Now robust against which sort of influences? The over all network should not be affected when parts of the network become unavailable therefore **redundancy** is a very important property. One criteria should therefore be whether the location discovery scheme is equipped with any redundancy. To asses which impact the failure of one single mesh router has on the network we can count the **number of messages** that are **exchanged to replace** the **leaving mesh router**.
Our system should not be adversely affected by the fact that the terminals will move. This means there should not be a "chain reaction" when a lot of **terminal** are **moving** over a short period of time. The network should not suffer as a whole, when many parties want to establish a call in a short period of time. In short: it should be **robust against high volume of incoming calls**. A location discovery scheme that does not have a **single central authority** will be more robust than a scheme that would have to re-build the central authority in case it would become unavailable.

As we want to keep traffic caused by location discovery as small as possible, it would be useful, when successful lookups are **cached.** This will hold the number of lookups small and exploit the principle of locality. A location discovery scheme should also be simple to implement and low on computing resources. One important question is much **memory** is **required** for the desired ff the location discovery scheme.

Questions regarding availability of caches and redundancy in any location discovery scheme are easily answered. It is also easy to say if a discovery scheme relies on a single central authority while in normal operation or not. These criteria will not be further discussed and simply used during evaluation.
The performance metrics however are treated in more detail in chapter 9.

9. List of Performance Metrics

For each of the following metrics we define which aspects of a location discovery scheme should be assessed with help of the particular metric. How these aspects are measured and in which unit.

9.1.1 Memory requirements

As we are dealing with networks with a potentially large number of participants one has to know if the memory requirements of a particular location discovery scheme are small enough to be handled efficiently. We will first define what needs to be stored and then estimate the size of these records in bytes.

9.1.2 Bandwidth requirements for location updates

When the location of the terminals is updated periodically or on events, this metric helps to see how much bandwidth is consumed for these updates. To get the requirements, the size of update messages is studied and how many such messages will be sent. From message size and number we get the amount of data that needs to be transferred for the updates. Additionally the update rate (how many updates take place over what period of time) is specified. From the amount of data and the update rate cumulative data rates can be derived. These cumulative rates are then broken down for a single link. The bandwidth requirement is measured in bit/s.
In case no updates are needed for a location discovery scheme to work, the maximum number of lookups over a single link is estimated instead.

9.1.3 Number of messages to converge in initial state

The number of messages that are needed to make the location discovery scheme functional (e.g. all Terminals that are connected to a mesh router may be discovered by any other mesh router). Worst case conditions are assumed and the result is calculated both as number of messages and number of bytes exchanged.

9.1.4 Number of message exchanges for a single lookup

Whenever a call is initiated, that terminates within the mesh network, the destination needs to be looked up. There are two cases to consider here: the location of the desired terminal is known at the originating router, or the destination has to be looked up with the help of the location discovery scheme. For systems that use caches but do not update them proactively, these are both possible conditions. To get a feeling for how such location discovery schemes perform both the average case and the worst case is calculated. As in 9.1.3 both the number of messages and the number of bytes exchanged is calculated

9.1.5 Number of messages exchanged to replace a leaving mesh router

A mesh router may fail (get disconnected) at any time. The number of messages exchanged to replace a leaving mesh router quantifies the consequences of these failures. It is assumed that all terminals that were associated with the leaving router will join an adjacent mesh router as soon as they notice the failure. Now the number of messages each terminal generates are estimated and then multiplied with the maximum number of terminals any router can handle.

9.1.6 Call setup time

The TOWN system is a telephony network and call setup time is an important performance feature in any telephony network. We calculate the call setup time from the number of hops. And the assumption that for each hop the forwarding time and processing time are 1ms each.

9.1.7 Robustness

Different aspects of robustness are considered:

a) Robustness to router failure: mostly covered with the metric defined in 9.1.5
b) Robustness to terminal mobility
c) Robustness with regard to the incoming call rate

For all three metrics a qualitative assessment is used.

10. Assessing Proposes Solutions

This chapter discusses the different approaches outlined in the assignment for this NDS Diploma Thesis [11]. Specifically:
"Updating terminal locations in caches", "Flooding" and "Flooding with caching". Updating terminal locations in caches, which is also referred to as "Event driven approach", updates a location database based on join / leave events. We take the liberty and divide the event driven approach further. First we consider a case where the location database is centralized, and then we discuss the effect of distributing this database.
Through chapters 10 and 11 we assume that the scenario size is limited to

- Number of mesh routers in the network, N_m = 40
- Maximal number of terminals associated with mesh router M_t = 30
- Maximal number of terminals supported by the network $N_t = M_t * N_m$ = 1200 Terminals.

4-6 of the 40 routers are gateways to the PSTN (Public Switched Telephone Network). One of the gateways is the master gateway.

For the data rate calculations, we assume location updates employ UDP datagrams. Thus the location update message has the following structure:

Ethernet Header	IP Header	UDP Header	Terminal ID	Mesh Router IC
18 Bytes	20 Bytes	8 Bytes	6 Bytes	6 Bytes

Table 6: A location update message consists of 58 bytes in total if UDP is used.

If the update is successful, the recipient acknowledges this by sending an update message, containing only the terminal identity. The size of such an acknowledgment is 52 bytes. When no acknowledgment is received, the originating mesh router simply retransmits the update message, up to a total of 3 times.
Employing TCP significantly increased the overhead. In TCP the update messages size is 70 bytes. In addition, the TCP connection needs to be established with a three way handshake before, and torn down after use. Using TCP implies approximately 300 bytes are sent from the originating mesh router and 174 bytes will be returned for one update message.

10.1 Event driven: Keeping Updates in the Master Gateway Only

The current section covers the approach called "Updating terminal locations in caches (event driven approach)" in [11]. But this chapter is confined to keeping the updates in the master Gateway only. For the same approach, with updates kept in all mesh routers please refer to chapter 10.2.

10.1.1 Memory requirements for complete location table

We assume every terminal or router mesh is identified by a 6 bytes unique identifier (e.g. MAC address). Stored in the location table are pairs of terminal identifiers and current locations (as mesh router identifiers).
The maximum memory requirement follows directly form the number of terminals:

N_t*2*6 = 14.4 kB (kilo bytes, 1kB = 1000 bytes)

In addition to the location we could alternatively store the path to each terminal. When both the used location table and the target terminal are on different leaves of the network tree, we get a memory requirement of:

N_t*(MaxTreeDepth+1)*6 = 288.3 kB (for worst case MaxTreeDepth is N_m-1)

	Terminal location only	Route to Terminal location
Memory requirements	14.4 kB	288.3 kB

Table 7: Memory requirements for the location / routing table. With 14.1 kB respectively 288.3 kB, the memory requirements for the location and routing table are small enough to be easily handled by the mesh routers.

10.1.2 Bandwidth requirements for location updates

Please note that we calculate cumulative data rates, not the data rate on a given link unless noted.
We discuss the design alternative where all terminal location updates are tracked solely at the master gateway. An upper bound to the amount of network traffic induced by this alternative can be estimated as follows.

We assume that all 1200 terminals involved in the scenario update their location within 10 seconds. It is further assumed, that the updates occur uniformly over these 10 seconds and that the loss rate is 20%for the first two attempts and 0% for the third.
Total Numbers of bytes sent to the master gateway can be computed as follows:

$$N_t * 58 bytes * \sum_{i=0}^{retries-1} loss\ rate^i$$ = 86'304 bytes (for 3 retries and 20% loss rate)

If we do not take into account that the maximum Number of retires is 3 (defined at the start of Chapter 10) the upper bound is:

$$\lim_{n\to\infty} N_t * 58 bytes * \sum_{i=0}^{n} loss\ rate^i = N_t * 58 bytes * \lim_{n\to\infty} \frac{1 - loss\ rate^{n+1}}{1 - loss\ rate}$$

$$= N_t * 58\ bytes * \frac{1}{1 - loss\ rate} = 87000 bytes$$

It is strongly suggested that the maximum number of retries is set fairly low. In case there is no maximum number of retires, in the worst case the number of bytes sent would be infinite.

And the total Number of bytes for acknowledges sent back to the mesh routers is:
N_t*52 bytes = 62'400 bytes

This argument leads to the following data rates, to update a dedicated mesh router (e.g. the master gateway): 69.0 kbit/s (1 kbit = 1000 bits) for "upstream" (towards the gateway) and 49.9 kbit/s for "downstream".

10.1.3 Number of messages to converge in initial state

The maximum number of messages exchanged to converge (M_c) can be calculated as follows: Each Terminal sends its location to the master gateway which does not distribute it any further.
We thus estimate the number of updates messages required to initialize the location database in the master gateway as follows

$$M_c = N_t * \sum_{i=0}^{retries-1} loss\ rate^i = 1488\,(given\,3\,retries)$$

10.1.4 Number of message exchanges for a single lookup

Average Case
In case the location information is stored in the master gateway only, the average number of messages for a single lookup is two. First the master gateway is queried which then responds with the desired information.

Worst Case
For this type of system, where the location information is only distributed proactive, the worst case is equal to the average case. Two messages are exchanged.

10.1.5 Number of message exchanges to replace a leaving mesh router

We assume all terminals associated with the leaving mesh router are within the transmission range of another mesh router. Then the "current active" mesh router becomes disconnected from the network. The failure of the mesh router is noticed by the terminals, which will try to associate with the "new" mesh router. In this scenario the new location for each terminal has to be propagated to the master gateway. The number of messages exchanged is thus:

M_t = 30 which corresponds to 58 bytes * 30 = 1740 bytes

10.1.6 Call setup time

We assume processing time and forwarding time for any messages are 1ms each.
Further the Maximum tree depth is assumed to be 5.
When a call form a leave mesh router is initiated, the depth of the tree will be traversed twice (once towards the gateway and back to the leave). Thus the maximum call setup time for this setup can be calculated as:

(1ms+1ms)*5*2 = 20ms

10.1.7 Robustness to router failure, terminal mobility and incoming call rate

As can be seen in section 10.1.5, replacing a failing mesh router is relatively inexpensive. The low number of messages generated means that updating current terminal locations in caches is fairly robust to failure of single mesh routers. But as the system updates only the master gateway precautions have to be taken to be able to recover form a failure in the master. To ensure that a failure of the master gateway is not catastrophic to the whole network, the data in the master needs to be replicated on other mesh routers. We thus suggest to replicate the master gateway to some of the other 4-6 gateways over the PSTN. Additionally a mechanism that promotes one of the replicated slaves to the new master, in case of failure, would have to be implemented.
With regard to terminal mobility it is beneficial to keep the location database centralized. Only one (central) entity has to be updated when a terminal changes its location. It is easily seen that the number of updates cause by terminal mobility is only dependent on the actual mobility rate, not the network size.
The disadvantage of a central entity is that it may be overwhelmed by too many incoming location requests. Each call will trigger a lookup on master gateway. In case the master gateway to the network by a link (as in Figure 11.3) this link could be saturated with 3561 lookups per second (please refer to section 10.1.2).

10.2 Event driven: Keeping Updates in All Mesh Routers

The current section complements 10.1. Now we study the event driven approach with updates kept in all mesh routers.

10.2.1 Memory requirements for complete location table

We assume all mesh routers keep a full location table. Therefore, the memory requirement is calculated as shown in 10.1.1. Consequently, the numbers are again 14.4 kB if only terminal locations are stored and 288.3 kB if terminal locations and routes are stored.

10.2.2 Bandwidth requirements for location updates

We are tracking all terminal location updates in all routers. Assuming worst case, we don't take advantage of the tree topology of our network. To update each router with

the location data, each originating mesh router simply sends the update messages to all other mesh routers. As can be seen trivially, sending the location updates to all routers increases the demand for bandwidth linearly in number of routers in the network.
Thus, an upper bound to the total amount of update data generated (to all routers) can be estimated as follows.

$$(N_m - 1)N_t * 58\,bytes * \sum_{i=0}^{retries-1} loss\ rate^i = 3366\,kbytes$$

Total number of bytes returned to originating routers:

$(N_m - 1) * N_t * 52$ bytes = 2434 kbytes

From the total amount of data generated we again calculate the cumulated data rates as in 10.1.2. The bandwidth requirements to update each mesh router, are 2693 kbit/s for "upstream" and 1947 kbit/s for "downstream". It can clearly be seen that these requirements are significant. Please note again that we calculated cumulated data rates, not the actual rate on a given link. To calculate the rate on a given link, the network topology has to be known.

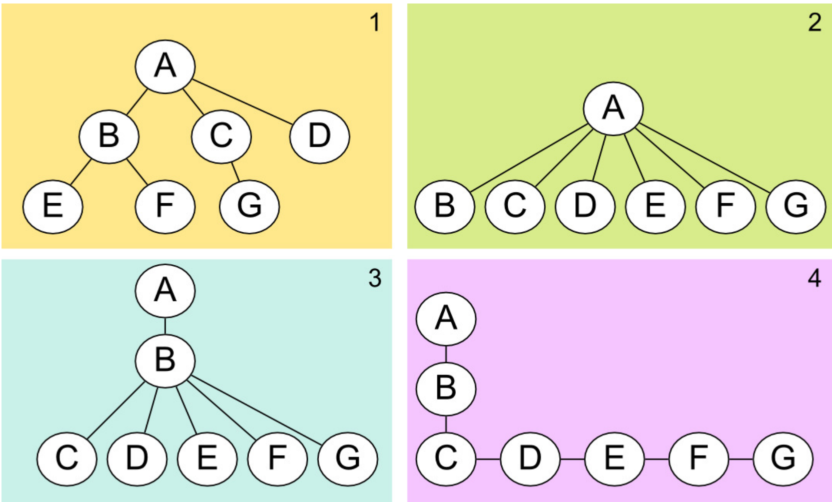


Figure 11: Example network topologies (trees). Node "A" is always the master gateway. Additionally node "A" is the root of the trees shown here. Top left (Figure 11.1) is a tree with a depth of two and three the master gateway has three children. Top right (Figure 11.2) is a tree that has maximal breadth. At the bottom right (Figure 11.4) a tree topology with a maximal depth (equal to N_m-1) is shown

The actual data rate on a given link is proportional to the number of children connected through that link. In Figure 11.4 above the data rate between "A" and "B" is equal to the cumulative data rate (e.g. for the example in Figure 11.1 the rate between "A" and "B" is ½ the total data rate. These examples clearly show that the network topology has to be chosen carefully to avoid overloading a particular link.
In the bottom left tree (Figure 11.3), the master gateway role should be assigned to node "B" which would make the tree identical to the maximal breath one. Constructing a network as shown in Figure 11.4 and Figure 11.3 makes no sense, as the link form node "B" to "A" becomes a bottleneck for all communication.
When the topology is similar to the one shown in Figure 11.1, updates from e.g. node "G" are implicitly known by "C" so the master gateway does not need to update "C". Additionally each parent node would only update its children (e.g. "A" only updates "B" and "C") further reducing the load the link "A-B". In essence: all routers that are on the path to the gateway are updated without generating any additional traffic.
This reduces traffic significantly as updates only need to got from the origin to the master gateway and form there to the leaves. In the case of binary trees, we get approximately ½ of the traffic. As the number of leafs (N_l) in a binary tree is:

$N_l = (N_m+1) / 2$ or $N_l = N_m / 2$ (depending on N_l being odd or even).

10.2.3 Number of messages to converge in initial state

In chapter 10.2 each terminal sends its location to all mesh routers. Again (as in 10.2.2) the number of messages scales linearly with the number of mesh routers in the network and is given by:

$$M_c = (N_m - 1)N_t * \sum_{i=0}^{retries-1} loss\,rate^i = 58'032\,(given\,3\,retries)$$

10.2.4 Number of message exchanges for a single lookup

Average & Worst Case
When a terminal initiates a call everything the mesh router has to do is look up the destination location in its local locations table. There are no messages exchanged at call setup, as the needed information has been shared with all mesh routes, when a change occurred. This is true for both the average and the worst case.

10.2.5 Number of message exchanges to replace a leaving mesh router

Again assuming all terminals of the leaving mesh router can reconnect to an alternative mesh router (as in 10.1.5). As now all mesh routers are keeping location information the maximum number of messages exchanged is:

M_t * (N_m-1) = 1'170, corresponding to a total of 67'876 bytes.

10.2.6 Call setup time

As we have seen in 10.2.4 only the mesh router local to the originating terminal is involved for a lookup. The Call setup time is therefore only the processing time for the local mesh router = 1ms.

10.2.7 Robustness to router failure, terminal mobility and incoming call rate

To keep updates in all mesh routers is very robust against failure of a single router. All information is available multiple times and therefore easily restored. As long as the information in each mesh router is identical restoration is trivial.
To keep the caches consistent reliable flooding should be used. This will ensue that even with mobile terminals, consistency between different caches will be upheld. Other than in section 10.1.7 not only one central location database but N_m databases have to be updated increasing traffic form relocating terminals.
Calls that originate inside the mesh network do not generate any additional traffic (see 10.2.4). Calls coming form the PSTN can be handled without any traffic too, as each gateway keeps a location table, not just the master gateway.

10.3 Flooding incoming location discovery requests at call setup time, without caching

How do we define flooding? Generally flooding means the originating node in a network propagates information to its neighbors which in turn forward it to their neighbors. We again assume that lookups and responses are done using UDP datagrams of approximately 55 Bytes each (arithmetic mean of 58 and 52 as derived at the beginning of chapter). The mesh routers will respond to all lookup requests disclosing whether the desired terminal is currently associated with them or not. In section 10.3 the network is flooded for every lookup, consequently no location information is cached in any mesh router.

10.3.1 Memory requirements in each mesh router

The network does not store terminal locations in this approach. The only memory requirements associated with locating are the ones for the currently occurring lookups and is therefore negligible.

10.3.2 Bandwidth requirements for location updates

As there are no location updates when flooding is used, we will calculate the maximum number of lookups per second over a single link.
Each lookup will flood the network, generating one lookup message to each other mesh router and equally many responses. A single lookup would therefore generate a total of:

N_m*55 bytes * 2 = 4400 bytes

accumulated traffic. Assuming the mesh router that originated the lookup is a leaf in the network tree, the most traffic on a single link will be the responses, on the link to the originating mesh router with 2200 bytes total.
How many lookups can be done within 1 second without saturating a single link? Assuming 2048 kbit/s links we get:

MaxLookups = 2048 kbits/s / (2200 bytes * 8 bits) = 116 Lookups / second

While not really bandwidth efficient flooding the whole network may be implemented for a network of limited size.

10.3.3 Number of messages to converge in initial state

As the lookups happen "as needed" and not in advance, there are no (lookup) messages exchanged when the network is being commissioned.

10.3.4 Number of message exchanges for a single lookup

Average Case
Each lookup requires a total of N_m-1 messages to be sent from the originating mesh router and one response from each other mesh router. Assuming a success rate of 80% in the first two tries and one of 100% in the third try we get for number of messages sent:

$$(N_m - 1) * \sum_{i=0}^{retries-1} loss\,rate^i = 49\,messages$$

We receive from each queried mesh router one answer (N_m-1) the total number of message exchanges is 49 + 39 = 88 messages. Equaling 4840 bytes.

Worst Case
The worst case number of messages exchanges is the same as stated above, as all mesh routers are always queried due to the nature of flooding.

10.3.5 Number of message exchanges to replace a leaving mesh router

As in 10.1.5 we assume all terminals of the leaving mesh router are able to reconnect with an alternative mesh router. The number of "location" messages needed to replace a leaving node is zero, as the knowledge is always local to the mesh router. Thus the only messages exchanged are the ones between terminals and their new mesh router.

10.3.6 Call setup time

Maximum call setup time is proportional to the maximum number of hops. For the worst case, the network would be a string of nodes, with the originating and the terminating mesh router on the respective ends (see 10.1.6). We get a maximum call setup time of:

(N_m-1) * (1ms+1ms) * 2 = 156ms

With a maximum tree depth of five one would get maximum call setup time of 20ms

10.3.7 Robustness to router failure, terminal mobility and incoming call rate

Flooding is completely distributed and no locations are cached in any mesh routers. As a result, the network recovers very well from router failure. A terminal can be located and connected to, as long as it is associated with any mesh router and the network

stays connected. The previous statement is true even if multiple other mesh routers fail.
With regards to terminal mobility flooding without caching is also very robust, as location changes do not trigger any sort of updates in the network. The incoming call rate however is a concern, due to the fact that each call leads to flooding the whole network.

10.4 Flooding incoming location discovery requests combined with caching.

Combining flooding and caching looks like the most promising of the proposed location discovery schemes. It would be highly desirable, if the positive effects of both flooding and keeping caches can be combined without the drawbacks. An ideal location discovery scheme should offer the high robustness (and hopefully simple implementation) of flooding combined with the low call setup times and relative low bandwidth demands of caching. We assume that caches are not actively invalidated when a terminal leaves a mesh router. The cached value is only overwritten when a new location for the terminal is reported. Regarding the question which mesh routers keep which kind of information, several caching strategies can be applied:

a) cache in **originating** router only
  only the mesh router originating a location request will keep the result
b) cache in **originating** mesh routers and mesh routers "**on path**" to the destination
  any mesh router that forwards a positive answer to a location request keeps this information in addition to information that was gathered form its "own" requests.
c) cache only in a **subset** of all mesh routers
  e.g. only the master gateway or all gateways (4-6 mesh routers at the top of their sub trees

If the cache is kept only in a subset of the mesh routers, all others will have to flood the network for each location request. However as the gateways would most likely have a high hit rate after, a relatively short time, and thus a flood would for the most part be limited to a sub tree of the network. From here on we assume that strategy b) is used, as it is only marginally more complex than a) while allowing the caches to be filled much faster.

10.4.1 Memory requirements

Maximum memory requirements are again identical to the 14.4 kB for a complete location table. (See 10.1.1)

10.4.2 Bandwidth requirements for location updates

As in 10.3.2 there are no actual location updates. We will estimate how many lookups can be executed within one second instead. We assume that 5% of all lookups can be answered by the originating mesh routers, as the destination is already in the local cache. A further 10% of the lookups only flood half the network (maybe the local gateway has the destination cached).
We start form the maximum of 116 lookups per second estimated in 10.3.2. The number of additionally possible lookups is:

  5% (local caches) + 10%*0.5 (as for 10% only half the network is flooded)
  = 116 *(0.05 + 0.1/2) = 11 additional lookups / second

The result will obviously vary wildly with different cache hit rates. Also false cache hits have not been accounted for. For each wrong cache hit, the total network will have to be flooded.

10.4.3 Number of messages to converge in initial state

As the lookups happen "as needed" and not in advance, there are no (location discovery) messages exchanged when the network is set up. (As in 10.3.3)

10.4.4 Number of message exchanges for a single lookup

Best case
In the best case the destination location is already in the local cache and therefore no location request has to be sent out. Consequently no location messages are exchanged other than between the terminal and its local mesh router.
Average case
The average case depends only on the "hit rate". And is the sum of "Worst case" times "miss rate" and "½ worst case" times "half hit rate"

88*85% + 44 *5% = 87.5% = 77 messages

Worst case
In case no mesh router other then the one associated with the destination terminal knows the destination location, the whole network will be flooded. The number of messages exchanged is identical to the 88 calculated in 10.3.4.

10.4.5 Number of message exchanges to replace a leaving mesh router

As discussed in 10.3.5 no messages need to be exchanged to replace a leaving mesh router, as the network does not rely on any individual mesh router.

10.4.6 Call setup time

In the best case the call setup time is just 1ms. Specifically when the desired terminal is already in the local cache. The worst case on the Other hand is equal to the 156ms derived in 10.3.6.
For the average case we assume again 5% local "hit rate" and 10% inside the local half of the network. We get 85%*156ms + 5%*1ms + 10%*78ms = 140ms. With more optimistic hit rates (10% local and 50% in local half) we get 102 ms for the average case.

10.4.7 Robustness to router failure, terminal mobility and incoming call rate

The network resorts to flooding when no or wrong information is found in caches. Falling back to flooding makes the location discovery scheme equally robust to router failure as flooding alone. The only disadvantage will be the performance impact when a "full cache" becomes unavailable.
Terminals changing their location will not cause any traffic as we only take action on lookups (as in 10.3.7). Flooding combined with caching is also pretty robust with respect to high incoming call rates. The caches help to reduce the number times the network has to be flooded.

We assessed the location discovery schemes proposed in the assignment to this thesis with regard to the criteria defined in chapters 8 and 9. The following chapter assess existing solutions and chapter (12) will evaluate which of these solutions are best suited to TOWN.

11. Assessing Existing Solutions

In chapter 7 we showed how location discovery is done in existing systems. We now asses the systems treated before, to see how well they fit for TOWN.
SIP and GSM will not be covered in detail, as they are not suitable for town due to the following reasons. SIP will not be included, as it is not distributed, and RFC3216 does not detail how the SIP location service is to be implemented. GSM is also not distributed and will fail if the single central authority (the HLR) goes down. While GSM is not considered for TOWN a rough performance assessment is no the less don for reference purpose.

11.1 GSM

GSM is not included in the evaluation, as location discovery in GSM, however a rough performance estimate is provided as reference.

11.1.1 Memory requirements

We first estimate how much memory a single (bare minimum) home location register entry needs.
The HLR needs to contain at the very least (size in bytes): IMSI(15), MSISDN(15), current VLR Address(15), current MSC address(15), authentication triplets (each 28).
So one HLR entry is at the very least 88 bytes. The VLR will keep a copy of most HLR data so a VLR entry is probably also around 88 bytes (HLR address instead of VLR).
One Terminal well occupy at least 176 bytes. To store a complete HLR for 1200 terminals needs > 211 kBytes.

11.1.2 Bandwidth requirements for location updates

As can be seen in Figure 5 there are six messages exchanged between HLR and VLR's for a location update (without considering authentication). These messages will be relatively small expect for "subscriber data" with at least the aforementioned 176 bytes. Lets assume 100 bytes up and 200 bytes downstream. With these numbers we get for 1200 location updates within 10 seconds the following data rates:
96kBit/s upstream and 192kbit/s downstream.

11.1.3 Number of messages to converge in initial state

For each terminal that registers with the network a "Location Registration" is done. For registration slightly more messages are exchanged as for location updates, as the terminal is unknown to any VLR and a TIMSI needs to be generated. The number of messages exchanged without authentication and acknowledges is around eight.

11.1.4 Number of message exchanges for a single lookup

For one lookup six messages are exchanged in the worst case (Figure 5)

11.1.5 Number of messages exchanged to replace a leaving mesh router

In case the terminal can register with a new mesh router, the procedure is the same as a location registration (Number of messages to converge).

11.1.6 Call setup time

From personal experience: In the order of seconds.

11.1.7 Robustness

a) Robustness to router failure: not very robust in case the mesh router with the HLR fails.

b) Robustness to terminal mobility. Pretty robust, but massive “connection attempt” on one cell may overwhelm the common control channel (CCCH)
c) Robustness with regard to the incoming call rate. Very robust, as updates are done proactively.

11.2 ENUM

11.2.1 Memory requirements

Total memory requirements for ENUM hard to qualify exactly. The authoritative name server for our example number +41441234567 will store a record that looks like this:
$ORIGIN 7.6.5.4.3.3.2.1.4.4.1.4.e164.arpa.
NAPTR 10 100 "u" "E2U+sip" "!^.*$!sip:terminalName@mesh-router.net!"
A size of around 100 bytes is realistic. A complete table would be 1200x100 = 120kBytes.

11.2.2 Bandwidth requirements for location updates

For ENUM to work, a terminal will have to update it's own “Naming Authority Pointer” (NAPTR) in the DNS whenever it associates with a new mesh router. A update may will contain the new record plus some overhead, lets assume 120 bytes transferred to the DNS server and 20 bytes returned as acknowledgments.

11.2.3 Number of messages to converge in initial state

As discussed, each terminal needs to set it's NAPTR record before it can be called. The number of messages will be one update request (and it's response) for each terminal. Resulting in a total of 2400 messages.

11.2.4 Number of message exchanges for a single lookup

The number of messages needed for a lookup may vary widely. It is possible, that a lookup can be answered without exchanging any messages between mesh routers. This may be the case, when a mesh router contains a ENUM resolver & DNS server with a current NAPTR record of the desired terminal. For the **worst case** there may be up to 15 queries to DNS servers with 14 deferring answers and one answer containing the record. In praxis there will most likely not be a DNS server for each sub-domain hierarchy. So number of messages for a lookup will probably be around 3 queries and answers in the **average case**.

11.2.5 Number of messages exchanged to replace a leaving mesh router

For each of the terminals associated to the failing mesh router the NAPTR record has to be updated resulting twice as many messages as terminals -> maximum 60 messages.

11.2.6 Call setup time

Due to the relatively large number of systems involved (ENUM Resolver, DNS servers) call setup time is hard to estimate. If we take the average case (three queries) and assume again 1ms processing time for each system we would get:
ENUM resolving in the originating mesh router (1ms) plus transfer and processing time for three DNS message exchanges (3x (1ms+1ms+1ms)) call setup time would be 10ms.

11.2.7 Robustness

a) Robustness to router failure: ENUM is not highly robust against router failure, as the network needs to exchange tow messages for each terminal, that can be transferred to a “new” mesh router. For lookups it is important, that the DNS system is implemented redundantly, otherwise the network could fail.
b) Terminal mobility is not critical to ENUM, as updating NAPTR records is cheap.

c) As there is only a relative cheap lookup for each call to set up, ENUM is pretty robust against high incoming call rates.

11.3 DUNDi

11.3.1 Memory requirements

In DUNDi the records for each terminal are nod shared proactively. It is likely, that no single DUNDi-Peer (mesh router in our case) holds records of all terminal locations.
We estimate Memory requirement to be small enough to be easily handled.

11.3.2 Bandwidth requirements for location updates

A location update may be accomplished by a "REGREQ". This contains the following date: 2 bytes command size, "VERSION" (4 bytes), "EID" (8 bytes) and "EXPIRATION" (4 bytes). The total size of "REGREQ" is 18 bytes. The response, containing the command(2) an optional "CAUSE" (0) and EXPIRATION(4), totally 6 bytes.
Protocol overhead (sequence numbers etc.) is an additional 6 bytes for each message.
To update 1200 locations in 10 seconds we get:
1200x24x8/10=23.04 kbit/s accumulated upstream data rate and half that (11.52 kbit/s) downstream.

11.3.3 Number of messages to converge in initial state

Each terminal will register with the closest mesh router initially. Resulting in each 1200 "REGREQ" and "REGRESP" totaling 2400 messages.

11.3.4 Number of message exchanges for a single lookup

In the **worst case** a lookup will result in all mesh routers being flooded with the discovery request Resulting in at least N_m x 2 = 80 messages. In the **average case** flooding will only occur if the terminal is not in the originating mesh routers cache. Assuming a hit rate of 10% the average number of messages exchanged is 72.

11.3.5 Number of messages exchanged to replace a leaving mesh router

Terminals that where associated with a failing mesh router will have to re-register with another mesh router in range. This will generate a total of 60 REGREQ and REGRESP messages.

11.3.6 Call setup time

In the best case call setup time will be only 1ms, when the location is cached locally. In case the desired terminal is located at the "opposite end" of the network, call setup may take 81ms.

11.3.7 Robustness

a) Robustness to router failure: DUNDi is pretty robust, as only few (and small) messages need to be exchanged to replace a failing router. The overall network is not affected in it's ability to route calls.
b) Robustness to terminal mobility: Again only two small messages need to be exchanged for each terminal and these changes are not propagated in the network.
c) Robustness with regard to the incoming call rate. Here the robustness depends very much on: how many requests can be answered by consulting the local cache.

11.4 SLS

11.4.1 Memory requirements

A full location table for SLS could be contained in the 14.4 kB derived in 10.1.1

11.4.2 Bandwidth requirements for location updates

In the average case SLS will transmit a "location packet" containing multiple entries of the nodes location table in an interval predetermined interval. Or in a time proportional to the transmit range divided by the nodes velocity. We assume our nodes (mesh routers) are stationary, so lets assume each mesh router transmits a location package every second. This way mesh routers that are in radio range will know all locations of terminals associated to their neighbors. In the general case however the network will be flooded.

11.4.3 Number of messages to converge in initial state

A TOWN network will never fully converge with standard SLS, as mesh routers are stationary. As the network is however functional from the start by means of flooding, no messages need to be exchanged.

11.4.4 Number of message exchanges for a single lookup

In the best case no messages are exchanged, as the desired location is in the local "table" already. In the worst case however, the network will be flooded. Both average and worst case are identical to 10.3.4.

11.4.5 Number of messages exchanged to replace a leaving mesh router

Again as in the normal flooding scenario no messages need to be exchanged to replace leaving mesh routers.

11.4.6 Call setup time

Best case call setup time would be equal to the 1ms in the "flooding with caches scenario" (10.4.6) and 102ms for the average case.

12. Evaluation and Results

First we briefly discuss which evaluation criteria where chosen and how these criteria are weighted in the evaluation. Then an evaluation matrix is shown summarizing performance of the considered location discovery schemes, with respect to the evaluation criteria. Finally we suggest which location discovery schemes fit the needs of the TOWN project best and should therefore be further investigated.

12.1 Criteria used and their weight

We do not include memory requirements in our evaluation as a complete location table is small (14.4 kb, see 10.1.1). Therefore memory requirements are not a constraint for any of the discussed location discovery mechanisms. The weights are as follows: The more important a criterion is to the TOWN Project the higher the weight. As the maximum weigh we use 3. For criteria that are not crucial but still important a weight of 2 is used and criteria that don't matter too much the weight is one. The identifiers for the criteria have been shortened to fit the table below (e.g. "Average call setup time" becomes "Call setup t").

Redundancy: in case there is a single point of failure and therefore no redundancy for any part of the network the criterion is not fulfilled. To reach a good or very good most if not any parts of the location discovery should be redundant. As redundancy is a fundamental requirement we assign a weight of 3 to redundancy.

Caching: does the discovery mechanism keep track of lookups that where successful, or are lookups only used once and then discarded? If all mesh routers keep results this is considered desirable as it minimizes network traffic and call setup time but increases system complexity. We will use a weight of 2 due to the discussed reasons .

Scalability: How many lookups or location updates can occur simultaneous? For the event driven approaches, the lookups are rather "lightweight" in comparison to flooding the network. So not only the lookups are considered but the updates too. We try to define how well the location discovery scales with the numbers of terminals and mesh routers. As the network size is known and rather small, scalability is not too important. Consequently we set the weight to 1.

Convergence: We consider convergence as the effort, needed to get to a working location discovery scheme. Is there a need to exchange messages to get the network into a "working state" or will the scheme deliver results straight away? Similar to scalability, convergence is not too much of an issue for our network, again the weight is set to 1.

Lookup cost: How much network traffic does a single lookup induce? As the network is relatively small this aspect is not as important as in lager networks. We set the weight to two.

Robustness: Combines three aspects:

a) Robustness to router failure: how much network traffic is generated when a mesh router fails and other mesh routers take its place? (10.1.5, 10.2.5, 10.3.5 and 10.4.5) The better the scheme is "distributed" (in the sense that no routers are special) the more robust, to router failure, is the scheme. As router failure is expected to be quite common, due to routers becoming unavailable to the network, a weight of 3 is assigned to this aspect.

b) Robustness to terminal mobility: How well does the design of the location discovery scheme handle mobile terminals? Will the location discovery scheme have to perform (extensive) update procedures when a terminal changes to a different mesh router? Event driven approaches are generally less robust with regards to terminal mobility, as they have to update the location database. We know that our terminals are mobile but most likely not highly mobile. Therefore a weight of 2 is chosen.

c) Robustness to high incoming call rate: How well does the location discovery scheme handle a high rate of incoming calls?. Location discovery schemes making use of distributed location databases (caches) will be more robust to high call rates than location discovery schemes without caches. We set the weight to 1 as the gateways can regulate the rate of calls coming from outside the wireless network.
Call setup time: How long does it take on average to setup a call? As the TOWN Project aims to build a telephony network call setup times are important justifying a weight of 3.

12.2 Evaluation Matrix

The degree to which one specific criterion is fulfilled is assessed with a scale form 0 to 3 with the following denotations: not fulfilled = 0, ok = 1, good = 2, very good =3.
The Evaluation criteria's names have been shortened to fit the table. The names in the tables relate to
We will not include location discovery schemes of GSM and SIP into our Evaluation matrix. The GSM location discovery scheme is not really suited for mesh networks and therefore excluded. For SIP the location discovery scheme is not specified in the standard and therefore subject to implementation differences.

	Event Driven no cache	**Event Driven cache**	**Flood no cache**	**Flood cache**	**ENUM**	**DUNDi**	**SLS**
Redundancy	0	3	3	3	2[1]	3	3
Caching	0	3	0	2[2]	1[3]	3	3
Scalability	2	1	2	2-3	3	unknown	2-3
Convergence	2	1	3	3	2	3	2
Lookup cost	3	3	1	2	2	1	2
Robustness to router failure	1	1	3	3	2	2[4]	3
Robustness to terminal mobility	2	3	2	3	1	2	2
Robustness to high call rate	1	3	1	3	2	2	2
Call setup time	2	3	1	1-2	1	1	1-2
Overall	27	34	25	33	27	27	34

Table 8 The evaluation matrix. The short hands for the location discovery scheme names correspond to the schemes discussed in the following chapters of this work: "Event Driven no cache" corresponds to 10.1, "Event Driven cache" to 10.2, "Flood no cache" to 10.3 and "Flood cache" corresponds to 10.4. ENUM, DUNDi and SLS are covered in the sections 7.3, 7.4 and 7.5 respectively.

12.3 Location discovery schemes suited for TOWN

It is evident from the overall scores (last line of Table 8) that the following are the location discovery schemes best suited for TOWN: Event driven with caches (EventD2), flooding with caches (Flood2) and SLS. It is not astonishing that SLS and Flood2 score similar results, as SLS is an implementation of a Flooding scheme that uses caches. Unfortunately SLS relies on movement of the nodes to effectively fill the caches we therefore believe that SLS is not well suited for TOWN. If a standard solution is

[1] Only as redundant as underlying DNS infrastructure
[2] Anywhere form '1' (gateway only) to '3' (all nodes)
[3] Only DNS caches
[4] A failing router can lead to a terminal being unreachable for one hour, if the default caching strategy for DUNDi is used

preferred ENUM and DUNDi are the contenders. As we do not know how well DUNDi scales, a score of 0 was inserted. If it would scale "ok" (a score of 1) DUNDi would have an edge over ENUM already. On the other hand, ENUM is adopted widely while DUNDi is not. It can also be seen that even a simple event driven location discovery scheme will work pretty well on a network this small.

13. Conclusion

The TOWN Project and its network define the setting for this work. Special about TOWN is the special "hybrid" design, between wireless mesh network and mobile telephone (communication) system.
Five existing systems, covering both “mobile communications” and wireless mesh networks, have been studied with regard to their location discovery schemes. To be able to compare and evaluate the different location discovery schemes:

- A characterization was defined;
- A set of evaluation criteria was proposed
- Performance Metrics were defined.

With based on characterization evaluation criteria and performance metrics, both existing and proposed location discovery schemes where assessed. It was found, that for a network as proposed in the TOWN project all discussed location discovery schemes would be suitable. While nearly any approach would work with TOWN, best suited are location discovery schemes that use flooding and cache their results. When choosing one of existing systems (as discussed here) either ENUM or DUNDi should be selected. As they are best suited to distributed terminal discovery.

Sadly we could not evaluate the location discovery schemes used in the Swiss Militaries "IMFS" and the Swiss National Rails distributed PBX system, as no information could be gathered. Not covered was the question, how well the location discovery schemes scale with the number of mesh routers in a network. We dis not investigate scalability with number of mesh routers, as a TOWN network will contai 40 mesh routers at the most.

Acknowledgments

Many thanks to my supervisors Ulrich Fiedler and Rainer Bauman for their most valuable help and support. Further thanks go to my employer Swissphone Telecom AG for allowing me to pursue this NDS.

14. Appendix

14.1 Glossary / Acronyms

DNS	Domain Name Scheme
DUNDi	Distributed universal Number Discovery
E.164	Definition of International Telecommunications numbering plan
ENUM	Telephone Number Matching
Geo Routing	In WMNs: Packages are routed towards the geographical location of the destination
GSM	Global System for Mobile Communications
H.323	Protocol Family for Audio / Video Sessions (similar to SIP)
HLR	Home Location Register
IAX	Inter Asterisk Exchange (VOIP Protocol)
IMSI	International Mobile Subscriber Identity
LA	Location Areas
LU	Location Update
MSC	Mobile Switching Center
MSRN	Mobile Station Roaming Number
Node	In WMNs a member of the network that is both provider & consumer of network resources
P2P	Peer to Peer
PBX	Private Branch Exchange
PSTN	Public Switched Telephone Network (sometimes POTS Plain Old Telephone System)
SIP	Session Initiation Protocol
SLS	Simple Location Service
TMSI	Temporary Mobile Subscriber Identity
TOWN	Telephony over Wireless Mesh Networks
URI	Uniform Resource Identifier
VLR	Visitor Location Register
VOIP	Voice Over IP
WMN	Wireless Mesh Network

15. References

[1] Southern Californian Earthquake Data Center, Index Map of Recent Earthquakes in California-Nevada, http://www.data.scec.org/recenteqs.html as of 2007-6-20

[2] Saumitra M. Das, Himabindu Pucha and Y. Charlie Hu, **Performance Comparison of Scalable Location Services for Geographic Ad Hoc Routing,** IEEE INFOCOM March 2005 http://web.ics.purdue.edu/~smdas/INFOCOM05-GHLS.pdf

[3] The GSM Association, **About GSM Association**, http://www.gsmworld.com/about/index.shtml as of 2007-02-22

[4] Heinz Ochsner, **Mobilkommunikation**, Lecture Notes ETHM.2005

[5] Event Helix, GSM Location Update Sequence Diagram, Telecom Call Flows (GSM, ISUP, VoIP) http://www.eventhelix.com/RealtimeMantra/Telecom/GSM_Location_Update_Sequence_Diagram.pdf as of 2007-01-31

[6] Rosenberg et al, **SIP: Session Initiation Protocol**, RFC3261, June 2002

[7] P. Faltstrom and M. Mealling, **The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)** RFC3761, April 2004

[8] P. Mockapetris, **Domain Names - Concepts and Facilities**, RFC1034, November 1987

[9] M. Spencer, **Distributed Universal Number Discovery (DUNDi)**, October 2004

[10] R. Friedman and G. Kliot, **Location Services in Wireless Ad Hoc and Hybrid Networks: A Survey**, April 2006

[11] R. Baumann and U. Fiedler, **NDS Diploma Thesis Assignment: Distributed Terminal Location Discovery in a Wireless Mesh Network**, November 2006

[12] M. Mauve, J. Widmer, H. Hartenstein, **A survey on position-based routing in mobile ad hoc networks**, IEEE Network Magazine, November 2001