Developing Delay-Tolerant Networking Applications for Arctic Communities

John Näslund

Luleå University of Technology
Department of Business Administration, Technology and Social Sciences
Division of Human Work Sciences

August 2013
Abstract

This thesis explores how usability in delay- and disruption tolerant network (DTN) applications might be developed and how DTN applications services might be put to practical use in remote regions of the Swedish Arctic. The tangible results presented are a set of end-user DTN application services that were developed for the remote sites Saltoluokta and Staloluokta.

Challenges that have created the need for new technology include topology, climate, lack of electrical infrastructure and lack of cellular coverage. DTN research aims to make data communication possible where communication is otherwise difficult, using state-of-the-art routing protocols and algorithms. Just as in a “legacy” Internet application, a DTN application is created to transfer information between nodes in a network: text, images, video, sound etc. Applications can be implemented in a network to perform a certain task, whether it be automated or user-generated data.

The main challenge with Arctic DTN applications is in the combination of technical and environmental contexts. The development of the applications presented in this thesis took place in the multi-disciplinary N4C project 2008 to 2011. The research team faced challenges that included how to make prototypes of the application services usable enough for the nomadic population in the Sami villages of Tuorpon and Sirges, in Padjelanta and Sarek national parks.

The thesis presents an assessment of the developed application services, with technical specifications and tries to answer what usability in DTN applications is, how it can be evaluated and when an application can be deemed usable.

The theories of Human-Computer Interaction (HCI) and a reference to this field’s history can explain the development of the DTN web applications as part of the history of technical development of information technology artifacts and their application in daily life. Since DTN “real life” applications are new, and the technology behind them is still in an early stage of development, it is not possible to point to any given evaluation method for usability assessment.

While an “only HCI” strategy would suffice to create good DTN services, some important aspects could be missed, for instance, the ties to democratic values and the overall socio-technical environment. In the work presented HCI theory meshes with Participatory Research and the Software Engineering process. The development of the prototype applications was an iterative process. The conclusion is that when participants see a benefit in using the services, this is acceptable usability.
Acknowledgments

Thanks to:

My supervisor Maria Udén, who assisted me in practical N4C project work and thesis writing. My colleague Samo Grasic, who I shared office with during the project at LTU as well as in the field. With the DTN implementation Samo made, I had a platform to test the DTN applications on, which was educational and rewarding. Malin Lindberg for giving me valuable feedback.

To Tannak and my field test manager Fritz-åke Kuoljok, who helped me with practical tasks in the Arctic field trials. Jan Lindblom, who worked on the first Web cache application. I also wish to acknowledge Elwyn Davies and Avri Doria, who both inspired me with their technical expertise on Internet technologies.

To all partners in N4C, including Ewa Gunnarsson who made the project multidisciplinary and interesting, during field work and project meetings. My gratitude goes to Sirges, Tuorpon Sami villages, Padjelanta tourist association for the accommodations, and to the helicopter pilots at Fiskflyg, who happily flew DTN bundles between villages.

To the EU Seventh Framework Programme (FP7) for funding my time in the N4C project.

Last but not least, I thank my family for the support and understanding they have given me over the years.
Preface

I remember my first introduction to the N4C project and my work at Luleå University of Technology. It was a day in May, 2008. I had just run the 10 km city marathon, when I got a call from Maria Udén, who asked me if I could meet her and some of her colleagues. There were senior Internet experts Avri Doria, Elwyn Davies and Maria at a table opposite me, asking questions. Suddenly, Elwyn asked me the most challenging question, which was something like this:

How do you think the problem with the lack of information technology in the Arctic region might be solved?

With this challenge was the start of a three year long process where my colleagues and I worked to create Delay Tolerant Networking applications for solving the communications problem. This included a lot of technical development and practical trials, in the office lab and around the Sami villages of the unique Arctic environment. During the first summer I worked in a small all-Swedish team as a field technician with both hardware and software preparation. As I am interested in Systems Engineering and Informatics perspectives, this suited me just fine. With my colleague Samo Grasic, I had the opportunity to learn how the DTN technology works, in practice. After a good start, I was accepted as a Ph.D student.

The project had many rewarding meetings in Slovenia, Norway, Portugal and Ireland. It was very interesting to be amongst so many researchers with various perspectives and ideas of things. The collaboration in the second year trial, when N4C partners joined us, proved to be even more rewarding than the first year’s experience. We also established a broader contact with the local Sami community. This gave a good base for a third trial. Especially helpful was our field test manager Fritz-Åke Kuoljok, who is a member of the community and a Sami reindeer herder. We worked hard to be able to provide pervasive DTN applications for people to test.

I also tried to keep up with the courses during the winter seasons. Two of the more influential were “Participatory Research” and “History of HCI” which provided me with a historical context and insights about topics of “usability”. I got to know Professors Ewa Gunnarsson and Yngve Sundblad, who had been part of the interesting history of creating usable IT artifacts. In 2010, we launched the last and largest N4C trial, where I was rewarded with spending even more time in the Arctic, working with four Sami families who tested our services during the summer. This time, services worked autonomously and the testers were able to use them without our assistance. After the N4C project had ended, I was ready to start assessing my experiences of developing and testing the DTN applications. It ultimately led to this work.

And so, I’m finally ready to come back to Elwyn’s challenging question - and this thesis is my contributing answer.
Contents

Abstract i
Acknowledgements ii
Preface iii

1 Introduction 1
1.1 Research Question .................................. 1
1.2 Delay Tolerant Networking ........................ 2
   1.2.1 Background of DTN ............................. 2
   1.2.2 Elements of a DTN application .................. 4
   1.2.3 DTN applications within the DTN architecture .. 4
1.3 DTN Application Development Projects ........... 5
   1.3.1 Scientific Monitoring DTN Systems ............. 5
   1.3.2 End-user applications for DTN ................ 7
   1.3.3 Previous work conclusions ..................... 9
1.4 N4C .................................................. 10
   1.4.1 N4C Characteristics ............................ 10
   1.4.2 N4C Project Organization ....................... 11
1.5 Setting the scene .................................. 13
   1.5.1 Description of the trial places .................. 13
   1.5.2 Staloluokta .................................... 16
   1.5.3 Saltoluokta .................................... 16
   1.5.4 Jokkmokk Municipality ......................... 17
   1.5.5 Reindeer Herding ................................ 17
   1.5.6 Cellular/Data Coverage .......................... 18

2 Theory 21
2.1 Usability and Applications: The history in short .... 21
   2.1.1 Augmented Reality and Ubiquitous Computing .... 21
   2.1.2 Human-Computer Interaction (HCI) Background .... 23
2.2 What is Good Design and Usability and how is it Developed .. 25
   2.2.1 Design and Developing process ................... 25
   2.2.2 Users and Interfaces ............................ 26
   2.2.3 Ethnography .................................... 28
2.3 Summary of Usability Theory ........................ 29
4.7 Evaluation of feedback .................................. 89
4.8 Locals and Hikers feedback ............................... 89
  4.8.1 Technical feedback ........................................ 90
  4.8.2 Individual user feedback ................................. 91
4.9 Focus group interviews ...................................... 91
  4.9.1 Feedback from the focus groups .......................... 91
  4.9.2 Technical feedback from the focus groups ............. 92
  4.9.3 Local politics feedback from the focus groups ....... 92
4.10 The Field/Environment feedback .......................... 92
  4.10.1 Weather feedback ......................................... 93
  4.10.2 Local area topography feedback ....................... 93
  4.10.3 Limitations ................................................ 94
4.11 Table of results from feedback ............................ 95

5 Realizing DTN Applications for Communications
  Challenged Arctic Communities ............................... 97
  5.1 An HCI and Participatory Design Approach .............. 97
  5.2 Usability in the DTN Applications ........................ 101
  5.3 Future Work Discussion .................................... 104

A Prototypes Technical Details
  and Examples .................................................. 106
  A.1 Components of Web Caching Application ................. 106
    A.1.1 Prototype example of Wwwoffle setup for the DTN gateway server 107
    A.1.2 Example of Web GUI client preparation ................ 108
  A.2 Components of Podcast Application ........................ 108
    A.2.1 Prototype examples of podcast request function on server ... 109
    A.2.2 Prototype examples of podcast on client ................ 110
  A.3 Components of NSIM Application .......................... 111
  A.4 Components of GUI ......................................... 111
  A.5 DTN over Virtual Private Network ....................... 112
    A.5.1 Configure the Server ................................... 112
    A.5.2 Security ................................................. 112
    A.5.3 Configure the Client .................................... 113
    A.5.4 Automatic Start ........................................ 114
1 | Introduction

This thesis was written at the department of Business Administration, Technology and Social Sciences of Luleå University of Technology (LTU), within the research topic of Human Work Science which falls within the faculty of Science and Technology. Research in work science, which includes human relations, socio-technical, democracy and critical perspectives, has been conducted at LTU since 1973. The curriculum for post-graduate studies in human work science states:

Work Sciences is concerned with the analysis, design, and development of products, production systems, and work organizations, based on people’s physiological, psychological, social and cultural conditions (LTU, 2007, p. 1).

1.1 Research Question

The aim of this thesis is to explore how usability in delay and disruption tolerant network (DTN) applications might be developed and how DTN applications might be put to practical use in remote regions of the Swedish arctic. It seeks to answer the following main questions:

- What is DTN application usability?
- How do we assess usability in DTN applications?
- How do we know if a DTN application has “acceptable” usability?

The thesis aims to contribute scientifically valid points to the research question, based on development work done during a three year period in the EU-funded Networking for Communications Challenged Communities (N4C) project. This implies that the points rely on the author’s assessment of his experience in developing and testing DTN applications in practice. A substantial part of the assessment is therefore based on knowledge produced in field trials with end-user participation and collaborative work in multi-disciplinary teams.

The tangible results presented in the thesis are Web, podcast and messaging DTN applications services that were developed for the remote sites, Saltoluokta and Staloluokta. In these sites, which are in Sarek and Padjelanta National Parks, tourists and local families have tested and experienced our DTN services. Challenges that have created the need for new technology include topology, climate, lack of electrical infrastructure and lack of cellular coverage. The hypothesis of the N4C project was that DTN can provide a
solution to these difficult conditions. However, DTN is a reasonably new technology and is therefore largely unexplored on a scale such as was the case in this project. It was not known how and to what extent the technology could, in practice, provide communication access to these remote sites.

If these questions are discussed theoretically together with results from practical work, it is reasonable that this thesis constructs a contributing answer to the question of how DTN applications with acceptable usability are best developed.

1.2 Delay Tolerant Networking

Delay Tolerant Network (DTN) (Cerf et al., 2002) is an information technology protocol. It is similar to the current TCP/IP driven Internet, but with the significant difference that it aims to be disruption and delay-tolerant. DTN is used in places were such technology is desired, needed, and appropriate, and where other protocol would fail. As an example, NASA conducts experiments involving DTN for space communications. Terrestrial applications include scientific monitoring systems, public services, military services, providing connectivity for rural populations, etc. In short, DTN research aims to make data communication possible where communication is otherwise difficult, using state-of-the-art routing protocols and algorithms.

1.2.1 Background of DTN

Data communication has evolved tremendously since the invention of Standard Internet suite protocols, TCP/IP. We now have fast backbones stretching between continents and between countries, enabling many to be a part of the evolving Internet, where the applications for users are seemingly endless if we allow ourselves some imagination. The speed with which electrons travel through our broadband cables is well suited to the Internet protocol that we use today - that is, if we live in a “connected” part of civilization.

The developing networking standard known as Delay Tolerant Networking (DTN) was originally envisioned as a tool for space communications purposes. NASA Jet Propulsion Lab (JPL) carries out DTN research, and has its own implementation of the protocol (Farrell and Cahill, 2006, p. 6). The DTN was thought of as an alternative to the Deep Space Network (established in 1963) that NASA uses for communications between spacecraft and Earth (JPL 2003). Delay tolerance was needed as the physical distances make normal Internet routines unfeasible. Gradually, the scope targeted widened, as it was realized that terrestrial scenarios were applicable as well. The DTN was proposed in an Internet draft in 2002 by some of the Internet pioneers (Cerf et al., 2002), where some of the core specifications and use scenarios are described, as “extreme environments”. The lack of reach of the “traditional” Internet in some areas might be the result of an “extreme” environment combined with excessive costs. For example, one cost that Farrell and Cerf mention is that traditional Internet connectivity uses the common TCP protocol, and hence relies on the speed of electrons and non-disruptive end-to-end transmissions.
By their nature, end user applications created for DTN scenarios typically aim to be as ubiquitous and pervasive as possible. For instance, peer-to-peer connections can be made between computers carried by people, by land/air vehicles, or DTN “stations”. If this is the foundation of the network, one of the many challenging tasks of developers and system engineers is to understand how people move about in a given area. Within such a local DTN, spontaneous ad-hoc WIFI connections are initiated. The simplest scenario is that people come and go within this area, and as they randomly meet, they share data between their devices. This sharing of data is a “store and forward” technique that is built in to the software. It basically means that all data passing through a DTN-enabled device is stored for a limited period and sent to some other DTN-enabled device, if the DTN algorithms in this case “think” it is a good idea. According to Farrell and Cahill (2006, p. 79), the details of how the decision is made are referred to the implementation of the particular DTN software and to the bundle protocol which specifies the rules of sending and receiving.

A comparison of the Internet and the DTN is illustrated in figure 1.1. What makes DTN disruptive and delay-tolerant is how the data, which is stored in bundles, can be transferred in regard to time by various methods. The versatility of the methods means that DTN can rely on TCP/IP or other protocols where suitable. Each DTN region can contain different delivery methods and implementations depending, on the unique situation. In those cases, DTN gateways between the Internet and other DTN regions might be used, as described by Fall (2003, p. 4).

Figure 1.1: This figure illustrates normal Internet layers versus the DTN layers. Source: Warthman (2003).
1.2.2 Elements of a DTN application

The imagined principles of DTN applications are often mentioned in relation to extreme environments and to developing regions in rural areas, where they are expected to help with information access, e-government services, health care, education, and citizen journalism (Kate et al., 2007, p. 2). Just as in a “legacy” Internet application, a DTN application is created to transfer information between nodes in a network: text, images, video, sound etc. Applications can be implemented in a network to perform a certain task, whether it be automated or user-generated data. The following paragraphs describe some basic DTN characteristics, as well as some applications that are reported in the literature.

A DTN architecture does not presuppose the normal advantages of legacy Internet, such as stable connectivity, low latency, low error rate, low congestion, high speed, and so on. However, that does not mean that applications can not use those favorable characteristics that Kate et al. (2007) mentioned:

...Yet, for optimum end-to-end performance we want to be able to take advantage of any of these favorable circumstances that are present, wherever they are present (Burleigh et al., 2003, p. 131).

This could mean that an application could use legacy Internet where provided, if it can support the transition technically. For instance, if an application user travels between an urban city and an extreme environment where DTN communication is available, the application can in principle be made smart enough to recognize whether it is in an DTN environment or not, and use that information to select the optimal strategy. In an extreme environment or any area where applications exist, they need to function for that situation - that is, in terms of data transfer cost and energy consumption. Therefore DTN applications can not be expected to provide end-users with exact details regarding delivery status or when data replies are expected:

Applications that are developed for use with DTN do not make assumptions about the timeliness of network transactions, so DTN messages can be delayed or batched to reduce cost or power (Brewer et al., 2005, p. 33).

Application development in circumstances cited can face extra complexity regarding the handling of the DTN service usability, since balancing usability and cost reduction may be a requirement. This is a characteristic issue in DTN application development involving people (end-users).

1.2.3 DTN applications within the DTN architecture

The main challenge with DTN has been in the routing component, which understandably is a huge and important component of DTN, and perhaps the biggest technical issue to resolve or better. Kevin Fall of Intel Research, who coined the term DTN, explained the routing problem in a conference paper, “Delay-Tolerant Networking for Challenged Internets” (Fall, 2003, p. 9). In the simple case, we have a sender and a receiver with properties and a message with properties. The problem here, according to Fall, is the routing problem: how to calculate the best way of sending the message from the sender to
the receiver. This task is up to the routing protocol - finding a way to connect the nodes in the network. DTN applications, whatever purpose they have, rely on these protocols (since they most likely will want to send user/application-generated information between DTN users and/or the users on the traditional Internet). In different areas, there are different protocols that might suit the particular situation, so DTN itself does not provide such a mechanism, as explained in the DTN Internet draft by Cerf et al. (2002):

> It is required that necessary routing information be able to be propagated throughout the DTN, but the mechanisms for doing that are neither defined nor constrained by the DTN Architecture.

Thus, a reasonable assumption is that DTN applications do not need to be created in the same way, after a certain standard or protocol. Therefore, DTN application implementations vary a lot in the current DTN research. The connection between the application and the protocol or DTN architecture may be described as an “application interface” that provides the architecture information necessary for the application to work. In many cases the application developer might need additional information about the associated bundles, and the application interface (API) might provide just that (Demmer et al., 2004, p. 10).

### 1.3 DTN Application Development Projects

This section looks at state-of-the-art applications available or under development for Delay Tolerant Networks (DTN). There are different categories of applications - some that are very technical and aimed mostly at developers and researchers, some that are meant for end-users, and applications that fit somewhere in between. The aim here is to identify DTN applications that have a connection to realistic settings, through development prototypes, implementations, trials or other. What kind of DTN applications have been created up until now, how were they used, and for what purposes?

It is quite clear that the focus of most existing DTN work concerns technical descriptions, books and reports. While traversing through DTN material it becomes evident that there are many researchers who work in lab-based environments. Typically they work with simulator engines trying to come up with results about the best way to transfer bundles through the network, or describing other technical problems and issues in theory. While these problems and issues are important, DTN based applications that have a potential impact on the realistic scenarios for end-users are less common in the general DTN material. Those that exist are therefore interesting and valuable for the future development of DTN based communication.

#### 1.3.1 Scientific Monitoring DTN Systems

*Low Earth orbit and interplanetary communication*

In 2008, “real world” practical tests by NASA and CISCO were done with satellite-earth communication (Ivancic et al., 2008). The aim was to have a DTN bundle-forwarding system installed on a satellite that would store images of the earth taken from the orbit,
which then would be probed by earth stations. The real life testing was done using the UKDMC satellite. Some of the goals were to show that DTN code can co-exist with the so called general SSTL code on satellites. Ivancic et al. (2008, p. 178) conclude that:

Delay-tolerant networking bundle transfers have been demonstrated from orbit. The DTN bundling shim onboard the UK-DMC and the ground station Saratoga client and bundle reconstitution mechanisms should continue to operate without affecting normal UK-DMC operations, giving NASA access to an operational DTN and bundle testbed in orbit.

Continued research by the same authors was also presented in a GLOBECOM paper from 2010: “File Transfers from Space Using Multiple Ground Terminals and Delay-Tolerant Networking”. Practical tests of large file transfers from satellites using DTN were demonstrated in 2009, to increase the efficiency of larger file transfers, and for better distribution among Earth-based stations Ivancic et al. (2010).

**Lake-water sensors**

In 2006, a sensor network using DTN was developed and tested at Trinity College, Dublin. The task was to monitor water quality in a lake with special sensor nodes. The project can be identified as a mission where DTN science makes progress from dealing with real world situations - experiences from “extreme” environments:

We are building a pilot DTN sensor network (http://down.dsg.cs.tcd.ie/sendt/) for lake-water-quality monitoring and have learned, and continue to learn, many lessons (some obvious, some less so) from this work. ...testing we performed on the sensor nodes prior to deployment, which involved ensuring that the enclosure was waterproof and that radio range wasn’t affected so close to the water’s surface (Farrell and Cahill, 2006, p. 76).

The project involved Dr. Stephen Farrell, one of leading DTN researchers and co-chair of the DTNRG (cited above).

**Vehicle and mobile ad hoc networks**

VANET stands for Vehicular ad-hoc network. Within VANET, researchers are trying to figure out the ways to best exchange information between vehicles (e.g. when they meet on the road). The research is based on ISO standards in the car industry, such as CAN (Controller-Area Network). In recent years, DTN software has been tried out as an extension of VANET, so that information from sensors on the car can be transmitted with disruption tolerance over a wireless link. In one trial DTN software was put on a modified OpenWRT image in a Linksys router. The prototype was connected to the OBD (message standards for in-vehicle data collected from the cars’ sensors) interface of the car and would then collect OBD messages for transmission to a base station (computer running DTN) while in wireless range. This demonstration was presented in a paper entitled “Extending Vehicular CAN Fieldbuses With Delay-Tolerant Networks” paper by Gil-Castineira et al. (2008):

The main contribution of this paper is the study of delay-tolerant networks (DTNs) for C2C CAN data propagation. We demonstrate that a DTN-enabled vehicular ad hoc network (VANET) is superior to a pure VANET.
As a proof of concept, we present a real implementation of a car node that accesses CAN data via the diagnostics interface.

The authors claim that their demonstration show that DTN communication is needed to overcome certain problems in “real life” situations regarding previous use of non-DTN communication between vehicles. Gil-Castineira et al. (2008) refer to well known DTN articles, thus relating their work in the DTN research context. The outcomes include prototypes of vehicular DTN applications that were tested in real settings. Related work involving vehicular applications of DTN has been done, although focusing perhaps on a different level. The new abbreviation “VDTN” stands for Vehicular Delay-Tolerant Network. VDTN research focuses on protocol design, mobility patterns, vehicle densities, and different movement models. The above DTN, for the VANET example, can be described as more practice-oriented and focusing on known issues, whereas VDTN research is on an overall level, and focuses more on the big picture of moving vehicles connected with DTN that could potentially be applied to many different scenarios.

Animal tracking
ZebraNet, which originated at Princeton University, focuses on developing and researching low-power tracking devices capable of sharing data between “nodes” using a store-and-forward method. The aim of the project was to place energy-efficient hardware on animals for tracking purposes. ZebraNet developers Liu et al. (2004) describe the intention as follows:

...ZebraNet Project explores these issues as we develop an energy-efficient mobile sensor network to help track zebra migrations in Africa...Individual nodes have been built in to collars and deployed on zebras near the Mpala Research Centre in Kenya.

The hardware prototype that was developed for ZebraNet includes a GPS module, a micro-controller (ultra low power from Texas Instruments) and a radio. The device battery was provided with power from solar chargers. The project focused heavily on the function of the hardware and software in the real world application, in respect of constraints such as memory, data storage, GPS, and energy. Other research includes protocols, routing schemes, and sensing mechanisms. For the purposes of engineering science, ZebraNet demonstrated the potential of a highly mobile, sparsely populated network.

1.3.2 End-user applications for DTN

Web applications
Jörg Ott and Dirk Kutscher of the Helsinki University of Technology published the paper “Bundling the Web: HTTP over DTN” (Ott and Kutscher, 2006). They claim that interactive protocols such as HTTP are not the most appropriate for disruptive networks. The authors points out that HTTP drawbacks in such networks can be avoided by using DTN technology:

In this paper, we address the former issue of enhancing HTTP to become workable in a delay-tolerant networking (DTN) environment. ...we focus on
the details for running HTTP in intermittently connected environments. In particular, we specify the HTTP-overDTN protocol operation and describe a system architecture suitable for incremental deployment in today’s Internet.

Ott and Kutscher (2006) further explain how HTTP services over DTN might be possible, by presenting lab experiments and measurements using a Web browser, DTN software, a proxy (squid) and Web servers. Although this particular paper does not include tests based in the real life scenario, it provides a good background to the theory behind cached Web access and DTN that has been exploited in the subsequent projects, SNC and N4C (section 1.4).

SNC
The SNC (Sapmi Network Connectivity) project seemingly started with a different focus than other typical DTN research, as it stemmed from an initiative of potential end-users living and working in the Swedish arctic. The project aimed to involve the whole context of state-of-the-art IT, extreme environments and the nomadic life style of the end-users. Discussions between reindeer herders in Jokkmokk municipality set the basic requirements for the web caching service. The first functioning Web cache prototype for DTN was created in the SNC project in collaboration with LTU. It was field-tested during the 2006 summer trial and consisted of one-way Web caching. SNC did a second summer test with Web caching available, and that prototype made use of scripts and the “Wwwoffle” application that was installed on “wrap” boards. Portable tablets were provided to let end-users and developers access the service through wireless LAN. Some of the Web cache knowledge learned in SNC became contributing input for the N4C.

Anders Lindgren first developed the influential PRoPHET protocol (Lindgren et al., 2012) for the purposes of SNC. He wrote about the “Experiences from Deploying a Real-Life DTN System” and setting up a Web cache service for real world testing in the field:

...web browsing on the other hand, tends to be a much more interactive task than reading and writing e-mails, and as such will also be more difficult to adapt for use within a DTN. During the test week, we deployed a web caching system that continuously fetched and distributed a predetermined set of web sites. Through this system, we could, for example, follow the events posted in the field test blog that was being updated during the test week (Lindgren and Doria, 2007, p. 2).

E-mail and messaging are among the most important DTN application services to implement. This is reasonable because these end-user services are the main communication tools in the DTN. Technically, e-mail and messaging is adaptable to the DTN, since the messaging basics are constructed in a “store and forward” manner. This is knowledge acquired from implementing an e-mail application:

...It is also very well suited to be deployed in a DTN network. While the protocols used for sending and receiving e-mail (such as SMTP, POP, and IMAP ...) are interactive and thus not well suited for end-to-end use over a DTN, the basic characteristics of the e-mail service makes it very easy to adapt it to DTN settings (Lindgren and Doria, 2007, p. 2).
E-mail is a common communication tool on the Internet and is therefore a crucial service even for potential DTN users. New DTN messaging applications prototypes have been created, such as “Not-So-Instant-Messaging” (Grasic, 2011), which means that simple text messages between DTN users can be sent directly peer-to-peer, without ever have to go through a proxy on the Internet.

The Background to LTU’s engagement in SNC
In 2001, a project started where researchers at Luleå University of Technology were engaged to support a local initiative where the current situation of, and new opportunities for women in, reindeer husbandry, were discussed. The main question was: “what social and technical circumstances have the potential to strengthen gender equality in reindeer herding?”. Ideas about interactive research methods and new ICT innovations can work as a democratic tool for development progress (Lindberg and Udén, 2010; Udén and Doria, 2007). The result was that the Gender and Technology Studies and the Department of Computer Science at LTU presented the project Sami Network Connectivity (SNC). The main project idea was that new and innovative DTN technology from NASA/JPL could be used to solve basic communication problems in a given Sapmi area. The ideas eventually became a reality when a two-year period of practical testing in 2006, initiated. Concepts of e-mail and Web cache applications were developed and tested in the field, as described above. This sparked interest among the reindeer herders, because of the potential economic value of the new communication technology (Udén, 2008), as well as in the IT research community, for its scientific value (Cerf et al., 2002).

1.3.3 Previous work conclusions
This section provided an overview of DTN application research tendencies. The subsections have presented some application examples. Evidently, researchers have varying focus, ranging from nomad populations to space crafts. Most research is very technically oriented, and only a few projects involves end-users, other than scientists or researchers. Therefore, no guidelines exist to assess usability in DTN applications against.
1.4 N4C

The N4C project\(^1\), successor of the SNC project, also used Sami reindeer husbandry as a scenario template. This EU-founded project was under way over three years, from 2008 to 2011 (Figure 1.2). One of its aims was to develop and implement an information delivery network in a real IT-challenged remote region; a place where conventional technology is not feasible or even possible (Udén, 2011). N4C continued partly along the same path as SNC but on a different level in terms of project management, number of partners and testing:

This project proposal envisions taking a major next research step in building converged networks that deliver delay and disruption tolerant Internet services for nomadic and other use. The challenge in that work is considerable: A full solution requires the development of networking infrastructures which are pervasive, ubiquitous and highly dynamic, supporting a wide variety of nomadic interoperable devices and services, a variety of content formats and a multiplicity of delivery modes (N4C, 2008, p. 5).

The project consortium consisted of 12 EU partners, and was organized in nine work packages (WP1 to WP9) that included everything from project management to the sharing of knowledge gained to the scientific community and the general public. Included on the N4C technical advisory board were experienced researchers who each had an advisory function for a particular aspect of the project. The advisors for technical aspects were: Vinton G. Cerf (co-founder of TCP/IP and DTN), Avri Doria (main architect of the SNC project), Patrik Fältström (Senior Consulting Engineer at Cisco), Anders Lindgren (SICS, co-author of PRoPHET) and Kevin Fall (Principal Engineer, Intel Corporation). The advisors for the usability aspect were: Jacqueline A. Morris (University of Trinidad and Tobago), Anita Gurumurthy (founding member and Executive Director of IT for Change, India) and Michael Gurstein (Editor-in-Chief of *The Journal of Community Informatics*) (N4C, 2008, 2011a).

With the expertise of the partners and the technical board concentrated in the N4C project, the challenge was to create novel technology that could function in a realistic scenario, using and adapting state-of-the-art hardware and software:

This development covered the whole spectrum from theory to applications and most importantly we have been experimenting with our developments in the field using “test beds” with real users. Our test beds were facilities in remote regions that were likely venues for “future DTN for remote regions” on a small scale, but operating under real conditions in scenarios that approximate ordinary life in such areas (N4C, 2011a).

1.4.1 N4C Characteristics

With N4C, as compared to most DTN research this far, there has been considerably more focus on the end-user participants and on usability. The project was highly multi-

---

\(^1\)N4C is an acronym for Networking for Communications Challenged Communities
disciplinary and the project’s partners provided various skills (N4C, 2011a). Important technical aspects in the N4C scenario are typically no power grids or mobile coverage in the places where DTN technology was deployed (Sapmi, Sweden). For instance, in a scenario in which an end-user participant wanted to send a message from a remote DTN location to the Internet, the round-trip might have taken a full day. If a user had sent an e-mail during the day, a reply e-mail might be received the next day. Among the social aspects is the fact that herders move about with their reindeer herd throughout the year (see section 1.5.5), making the task of achieving connectivity both more difficult and more important in a modern society in which contact with the surrounding world is essential.

1.4.2 N4C Project Organization

The Work Packages (WPs) within N4C were organized according to function, and the project partners had responsibilities for one or more packages. Each WP fills an important function, whose output is input for the next WP (Figure 1.3). The partners were: Luleå University of Technology (LTU), Sweden (Coordinator), Albentia Systems, Spain, Universidad Politécnica de Madrid (UPM), Spain, Intel, Ireland, Trinity College (TCD), Ireland, NORUT, Norway, ITTI, Poland, Instituto Pedro Nunes (IPN), Portugal, MEIS storitve za okolje d.o.o., Slovenia, Tannak AB, Sweden, Power Lake AB (PLAB) Sweden, and Folly Consulting Ltd., United Kingdom.

Tannak AB was a spin-off from SNC, started by two reindeer herders, and its purpose was to develop a reindeer tracking system. My work with WP3 and WP8 meant that I collaborated closely with partners NORUT, TCD, MEIS, INTEL, and FOLLY on the development of pervasive applications and their testing in the remote test bed.
Work Package 3
R&D pervasive applications WP originally contained the development of three different applications that were intended to be tested in field trials, and thus provide and demonstrate the capabilities of DTN: “The variety gives a broad view of the capability of DTN and sharpens the validation of the architecture and software development” (N4C, 2008, p. 28).

- Hiker’s Personal Digital Assistant (PDA). Involved partners: NORUT, MEIS.
- Web caching. Involved partners: FOLLY, LTU (John).
- Meteorological and Environmental Data Capture. Involved partners: TCD, INTEL, and MEIS.

Throughout the project, additional ideas were added and tried out. The WP3 subsequently led to the deliverables D3.1, D3.2, D3.3 for the EU Commission (see bibliography). The deliverables contains details about the functional Specifications, evaluation of development, and the final versions of the application prototypes.

Work Package 8
Tests and validation in two remote test beds meant the results gained in WP2-6 (the architecture and applications) were tested in the “real-life” environment by end-user participants such as members of reindeer herding families. Sub-objectives of WP8 included providing the end-users with basic Internet services such as e-mail and web caching and other features, i.e., the results of the R&D in WP3 (Näslund et al., 2010). Tests using DTN architecture to transfer meteorological data were carried out by MEIS, using sophisticated hardware and sensor systems in rural areas of Slovenia, the other remote test bed, besides the Swedish.
1.5 Setting the scene

This section contains information about the Swedish part of the N4C DTN summer and winter trials in year 2008, 2009 and 2010, extending into 2011. The aim of this information is to establish an understanding of the conditions in which researchers and developers from N4C worked during the trials. Norrbotten is the county that contains Jokkmokk Municipality and the Sami villages Sirges, Tuorpon, Visten and Sörkaitums. The main part of the N4C trials were within the Sirges and Tourpon Sami villages. Jokkmokk Municipality covers a large area in which there are a number of specific places where N4C conducted tests using DTN from 2008 to 2011. The tests have been conducted mainly in Staloluokta and Saltoluokta.

1.5.1 Description of the trial places

Saltoluokta, Staloluokta, Ritsem, Jokkmokk and Luleå were the key trial locations for the LTU team in N4C (Figure 1.4). Each represents unique characteristics that N4C researchers had to adjust for. Additionally, being inhabited by local Sami families and visited by hikers, Staloluokta and Saltoluokta were ideal villages to host the extended trials, due to N4C originating with a reindeer husbandry model.

Figure 1.4: N4C field test areas: Ritsem, Saltoluokta and Staloluokta. The blue area covers the main trial area, though the tests were conducted in areas immediately surrounding each of the three villages. Source: Google Maps editor, ©2013 Google.
Transportation by car, bus, or train from Luleå to Ritsem is possible, since there is a road leading to Ritsem. Along the road it is possible to stop by the side of the river in Kebnats and take the boat to the hiker and tourist-friendly Saltoluokta. If the traveller continues along the road instead, he or she will end up in Ritsem, which has a small local food shop/café connecting to a hostel. This accommodation is mainly used by hiker tourists as a start and end point for the Padjetanta walking path. In Ritsem there is a small harbour and a base station for Fiskflyg, a locally based air transport company, where helicopters land/start. They usually fly local people, supplies, and tourists to the mountain villages. Helicopters fly from Ritsem (farthest point from which land transportation departs to the test area) to Staloluokta daily at 2:00 p.m. A helicopter flies back again toward Ritsem at 3:00 p.m. The one-way flight takes about 30 minutes. If walking, it might take the traveller two days, walking along paths from Ritsem to Staloluoka.

Large parts of the county are protected by law: about 20% of the county area is protected as national parks and nature preserves. About 95% of all Sweden’s national parks are located in Norrbotten. The county reaches from the dramatic, mountainous alpine areas to the coast where, the unique archipelago exist. The national parks are Sarek, Padjetanta, Stora Sjöfallet and Muddus. Everyone has a right to visit this natural environment, but people are not allowed to stay or live there unless they belong to a Sami village, due to the protection by the law (Länsstyrelsen, 2013).

The roads in the mountains are sometimes subject to rock avalanche. The road to Ritsem (last outpost before one need to continue with boat or helicopter) is one of those roads, where if they become blocked, people might need to take helicopter in order to reach their destination. During the winter the road is not always passable, and bus traffic stops during this season.
Figure 1.5: Illustrates the closest train, flight and road connections for Ritsem (pin A). The hikers in the picture represent the Padjelanta area. Source: Google Maps editor, ©2013 Google.
1.5.2 Staloluokta

Staloluokta was the host of three Sweden based N4C summer trials. This Sami summer camp is in the Padjelanta national park and belongs to Tuorpon Sami village jurisdiction. It is located farther west, close to the Norway border, between snowy mountain tops. It is a semi-large camp where 15-20 local Sami families spend time during summers. The houses are scattered around the camp with up to several hundred meters between them, connected by walking paths. In the centre of the camp there is a tourist cottage with room for about 40 people. Tourist facilities are maintained by the local families and the cottage is usually open from July to September.

During the hiking-friendly summer season, Staloluokta is a popular stop for hiker tourists since it is located on the Padjelanta path. Helicopters from Fiskflyg lands twice per day, but Staloluokta is also served by the Lapplandsflyg airline that brings tourists mainly from the southern part of Norrbotten. Staloluokta is among the locations referred to by N4C researchers as “extreme” environments, not only because of the harsh weather, short summer and remote location. The phone communication is limited to satellite connections. Messages, such as shopping lists are often passed on to helicopter pilots, who act as mail-men. Short-range radio is used between locals, and to helicopter pilots sometimes. The GSM/NMT network is not reachable at all in or around this village. No electricity grid is built here, hence the electricity power is provided either from converters that runs on fuel, solar cells, or wind or from water power generators. This is something that local families in this village have been using since many years (as noted by N4C-LTU during trials). Solar power is among the most popular energy sources, as it is easy to install and use, and easy on the environment (no noise or heavy machinery is required). However, it is not necessarily always the most efficient energy source in Staloluokta, since it depends on the weather and on low power consumption. For best efficiency, a mixture of energy sources might be the best option.

1.5.3 Saltoluokta

Saltoluokta was host camp for N4C trials in summer 2008 and summer 2009. This camp in Sarek National Park has a unique setting for DTN researchers. It belongs to the Sirges Sami village jurisdiction and was built during 1912 when hiker tourism became popular (STF, 2013). The topography here involves more forest and smaller mountains than does Staloluokta. There is also less harsh weather in this village due to its lower altitude and the surrounding forest. Electricity grids exist in this village, but even if NMT coverage did exist according to the Net1 coverage map, in 2008 there was no usable coverage for NMT. To access the Internet, one had to travel back across the river by boat to the mainland. As in Staloluokta, hikers and tourists frequently stay there during summers. The amenities include a tourist hotel and a restaurant, so the village is a popular summer resort hosting concerts of folk music, etc. There are locals in Saltoluokta who stays for longer periods, meaning that people are moving around in the village along the paths. It is relatively easy to access Saltoluokta by car or bus, however, a short boat connection (Harbor of Kebnats-Saltoluokta) is necessary. The village was a good starting point for N4C researchers, because food and accommodations were easily available. In terms of
IT communication accessibility, the village was by default considered “disconnected” in 2008 and 2009 because there was no direct Internet connectivity.

1.5.4 Jokkmokk Municipality

Jokkmokk is one of the largest municipalities by area in Norrbotten. It is located on the polar circle and on the upper sections of the Luleå rivers. However, within its 18,144 sq. km, Jokkmokk only contains only about 5100 people. The population density in Jokkmokk is about 0.3 inhabitants per sq. km. Many have a connection to Sami reindeer herding occupation. Hence, the population is largely Sami, so naturally many speak both Sami and Swedish. There is a mix of traditional herding culture and modern society where both herders and non-herders can have jobs in teaching, nursing, administration, trade, IT, journalism etc. (Kuoljok, 2011; Länsstyrelsen, 2010).

Climate

The climate in the mountainous region can be extremely cold during the winter with down to –30°C or less. In the summer time weather can get warm, up to 30°C, but this is not the case in the mountain region. The further up the mountains one travels, the weather tend to get more harsh. The precipitation in the Arctic mountains is about 1000-1500 mm during the year, but can be as much as 2000 mm (SMHI, 2012).

1.5.5 Reindeer Herding

Reindeer herding is culturally and politically important to the Sami people. The traditional economy is closely tied to herding, implying that reindeer herding is an important business for many Sami (Lind et al., 2004; Kuoljok, 2011). The Jokkmokk municipality contains a number of Sami villages of two categories. There are three mountain villages, and there are two forest villages. The reindeer moves between and across villages (here, a “village” refers to an extensive reindeer grazing territory) during the year depending on the season. The migration of the nomadic Sami people takes place mostly during spring and autumn. During this time the herders can move their reindeer up to 150-200 km in about ten days. Among the noteworthy characteristics of the “semi-nomadic” Sami people of today is the fact that they work in various locations throughout the yearly cycle and the seasons. Their movements are dependent on certain key factors: reindeer behaviour, climate and pasture, and laws and regulations (Samer, 2010).

The herders themselves have a history of quickly adapting to new technologies. One example is the “snowmobile revolution”, in which reindeer herders were among the first to put snowmobiles into demanding, work-related use. This quickly made the herding business easier for the herders who adapted. Motorcycles, seaplanes, helicopters, and cars also have their place in modern reindeer-herding activities, simply because they are required in order to keep the Sami lifestyle and cultural heritage alive. Radio communication is another such example and is widely used for short-range communication between herders. The communication is a crucial aspect of the nomadic lifestyle.
Opportunistic use of existing infrastructure in the rims of the grazing lands, fits well into the traditional way of the reindeer herders, to add as little as possible to an environment (Udén and Doria, 2007, p. 9).

Internet Protocol (IP) communication could be a highly appreciated communication tool if it were applicable, due to the wide range of services that would then become available. Such a solution would help the Sami herders maintain both their lifestyle and their business. Recently, herders, well aware of this, have tried to adopt the various communication technologies that would give them access to the Internet while on the move. However, due to the natural movement in the cultural herding business, no stable, easily available technology has yet been found that provides IP communication for mobile use in remote rural areas. During N4C field trials, it was noted that mobile phones are popular among the Jokkmokk population, who used them especially frequently closer to the centre of the municipality, where the coverage is good.

1.5.6 Cellular/Data Coverage

In Jokkmokk Municipality there is access to mobile data communications to the Internet over 3G or 2G/Edge, like other municipalities in Sweden. However, the reindeer herders do not have constant access, due to the vast distances they can travel while herding. Other “over-the-air” systems include services which runs over a 450MHz frequency band (and is at the time of writing owned by NET1). This network works for telephones and fast Internet connections, and the plan was to aim for Scandinavian-wide coverage (Sweden, Norway and Denmark) where the competition is not able to provide the same quality of service, or none at all. The 450 is usable mainly in the low lands for broadband access, but it was not available in the remote mountain regions in most of the Sami villages where herder families have their summer residences. Satellite connections have often been thought of as a potential solution. However, the Sami discovered that the solution would be too expensive, and in many locations it is not a durable solutions, due to “radio shadow”. Another drawback with a satellite IP connection is that it is best used as a downlink, meaning that a land line uplink would still be necessary. Figures 1.6 and 1.7, following, show an approximation of mobile coverage over the mountainous N4C trial area.
Figure 1.6: GSM coverage in western Norrbotten. Current situation as of 2012 for the 2G/EDGE/3G GSM system. The white areas on the map do not have coverage for cellphone usage. Source: Telia (2012).
Figure 1.7: Map of NMT450 network coverage, showing no coverage for Padjelanta. Source: Net1 (2012).
Since DTN applications are relatively new and definitely unexplored in the realm of usability, manuals for developing end-user applications for DTN are basically non-existent. Thus, technical development experience from history might prove useful. This includes the history of IT development, innovations and modern development theories. These parts have led to the innovation of the DTN technology. However, since such theory is quite comprehensive, it should to be selective enough for relevance. The following three angles are appropriate to explore the usability of DTN applications:

First angle is DTN as result from history. understanding the history of human-computer interaction (HCI), i.e things that people have learned from IT-artifact inventions.

Second is the N4C project aim, challenges and the wishes of end-users.

Third angle is the generic IT usability field, which can be seen as helpful when starting from scratch with usability in DTN applications.

Earlier, the DTN scenario was explained as a communication access enabler for certain areas or realms, and the unique characteristics of DTN limitations were also highlighted. Existing knowledge produced over the years in fields like computer science and human-computer interaction (HCI) can be a providing factor or limitation for any DTN application, since they tell us how applications generally might be developed. Some theories could directly affect the usability of an DTN application, while other might not be so clearly defined. These theories, together with DTN knowledge and background, make up the total usability case for the applications prototypes (see page 67). The following sections will provide the historical view and the details that lay the groundwork for for applications.

2.1 Usability and Applications: The history in short

To understand the way usability is designed in applications, we need to explore the fundamental ideas, limitations, and progress that lie behind the technologies of the applications. The history of computer science helps to give some background as to how IT applications, such as those in DTN services, work in terms of usability and design.

2.1.1 Augmented Reality and Ubiquitous Computing

Dr. Vannevar Bush’s 1945 article, “As we may think” (Bush, 1945) suggested what scientists, mainly physicists, should focus on in the post-war period, after having collaborated
in wartime against a common enemy which would perhaps no longer exist. According

to Bush, scientists needed to look forward for a common cause. Bush gave examples of

what “good” science has given humanity, e.g. medicine and technology. With this, he

said, comes “a growing mountain of research”. Bush feared that present recording and

handling of research was insufficient. There was no time to grasp all the research pro-
duced, which could lead to the risk of important research being lost and forgotten. Bush

realized that something needed to be done for the sake of present research and future

knowledge inheritance.

Bush was looking ahead to the tools that might help solve this problem. He observed that

photocells, calculating machines, televisions, cars, and radios were by then being made

cheaply and worked reliably, even though they were of complex construction. Looking

at photography as one way of keeping science records, Bush envisioned how the future

camera might be strapped on a human head or built into eyeglasses. Limitations were

recognized at the time of writing (dry photography not as yet in existence), but Bush

also saw that these limitations would not always exist. He foresaw that photography with

connected text for database records of science’s progress would be developed. He did not

of course envision digital photography, but rather a physical microfilm format that could

store things like the Encyclopedia Britannica in a very small space.

Interestingly, Bush in essence described the modern-day application of computers, or

at least the concept of storing information as a photographic record. The difference is

that Bush investigated possibilities with then-existing examples of Human Computers

and punched cards or other electrical calculating machines. Bush also described the ba-
sics of a ordering system and a database. Another example is his “Memex” which could

view books and link two items to each other (an early vision of a hypertext system that

is now realized with the Internet). The system was, in theory, meant to be used by scien-
tists in their research for effective storing and recalling of information. Bush’s post-war

initiative investigated the future of computing and its importance for human advances in

science, and his research influenced other scientists in the field of computer science.

Evolving information technology

The idea of human communication through the use of computers networks has been

around, at least since the 1960s when mathematician, physicist and psychologist J.C.R

Licklider developed his “Galactic Network” (Licklider, 1960; Licklider and Clark, 1962)

concept. He suggested that computers would be interconnected to each other, allowing

network data to be extracted from remote data storage by users. Licklider later became

the head of DARPA’s computer research program where he developed his ideas about a

global computer network. Licklider’s goal was to combine the human intellect with the

power of the computer, to create something that would “empower” humans. For this to

happen, he pioneered the computer science research programs that would later create the

Internet.

In 1968, the specifications for ARPANET began to take shape. The theory of a packet

switching network, called Interface Message Processors, was invented by Leonard Klein-
rock in 1961. This discovery allowed the first Internet computers to be connected. One
host resided at UCLA’s Network Measurement Center and another host was provided by Doug Engelbart’s “Augmentation of Human Intellect” (Engelbart, 2001) project at the Stanford Research Institute. The project continued developing the ideas and scenarios that Vannevar Bush had foreseen with the Memex machine. Engelbart’s project used an early example of hypertext, called NLS (oNLine System). By the end of 1969, four Internet hosts were connected in a computer network called ARPANET, with the intention of facilitating resource sharing between computers over great distances.

**User interfaces**

Doug Engelbart’s project invented the mouse, first demonstrated in 1968. Star, a predecessor to the current computer desktop, was a graphical user interface (GUI) developed at Xerox PARC and presented in 1982 (Smith et al., 1982), after which it was put to commercial use. The desktop prototype was called “Star” and was developed at Xerox PARC. Susanne Bødker described Star’s process of innovation:

> The shifts away from Star’s design are worth paying attention to, because Star was not designed casually. Actually, Star was designed though an exemplary systematic design process, where state-of-the-art methods were applied (Erickson and McDonald, 2008, p. 68-72). The Star team put a lot of time into developing the user interface (UI), employing the principle of “User interface first”. That meant that software and hardware came second and were not in the forefront of the developers’ minds. The team used the iterative design (when the sequential waterfall model was the norm) and made changes and trials over an eight-year period. This long process led to the successful groundbreaking desktop design which still is used today on PCs, incorporating windows, drag-and-drop, what-you-see-is-what-you-get, etc. While “Star” offered simplicity and clean design, Bødker acknowledged that as desktops in general evolved and advanced they became more and more cluttered and the principle of simplicity was lost.

Weiser (1991) wrote in his article “The Computer for the 21st Century”, about ubiquitous computing:

> My colleagues and I at PARC believe that what we call ubiquitous computing will gradually emerge as the dominant mode of computer access over the next twenty years...Machines that fit the human environment, instead of forcing humans to enter theirs, will make using a computer as refreshing as taking a walk in the woods.

His belief was that technology will somehow improve people’s lives, but at the same time we must try to minimize the complexity that can happen when computers enter everyday life. Because of this, some usability engineering is going on, along with methods that support user centered design.

### 2.1.2 Human-Computer Interaction (HCI) Background

HCI is about the relationship between human and machine, how we adapt technology, and create or invent new technologies that change our behavior. In practical terms, it
is very much about user interfaces, information processing, and cognitive psychology. It is about user experience. The history of HCI consists of several movements, traditions, and programs. Kuutti (2009) give the background for HCI. Before the HCI field was created, there was some human-computer relations work from the 1960s on. One such early example is Doug Engelbart’s project mentioned above. HCI-related work occurred in the psychology of programming (information processing and cognitive psychology) and in ergonomics, such as cockpit designs.

Cognitive psychology was a leading contributor in the 1950s, and information processing was of enormous interest. Kuutti (2009) wrote that researchers viewed the human brain as a computer. Laboratory experiments consisted of “careful modelling and factorization of use situations” (Kuutti, 2009, p. 50), all based on the theories of information psychology. Some believed that it would be possible to map human cognitive processes into machine code, and consequently artificial intelligence (AI) became a much studied subject for about 20 years, until the HCI field began to shift the focus towards a more user-centered view. During these 20 years, before the PC arrived on the mass market, early HCI theory viewed humans as “rational information processors” and a functioning part of organization information systems.

When the PC emerged in the 1980s, HCI’s focus shifted quickly due to interface design and commercial innovation. In 1979, Visicalc for the Apple II computer was invented by Dan Bricklin and Bob Frankston (Bricklin and Frankston, 1984). This application was ground-breaking: calculations could now be made using visual interface that was easy enough for use on desktop workstations. Before Visicalc one had to be a programmer to make something happen, which made the PC viable only for a core group of enthusiastic users, hobbyists, and hackers. IBM entered the PC market, which was the start of IBM’s commercial success. The problem in the early 1980s, according to Kuutti (2009), was that application developers were generally unskilled in creating software with high usability standards. Both the academic and business worlds realized this, and tried to solve the problem. The first CHI conference was held in 1983, and it is still active today with representatives from industry and academia, MIT, KTH, Palo Alto Research Center, Nokia Research Center, SICS, Aarhus University, Google Research, Microsoft, and others (ACM, 2012). Since that time, the industry and the academy have worked together in the HCI field.

It was apparent that it was not easy for academics to create useful HCI knowledge for the industry in the 1980s. The reason, according to Kuutti (2009), was that academe had a different time frame than the industry when it came to producing results. The pace was faster in the business world. Additionally, the theories of information processing psychology that dominated the research at that time needed to be conducted in controlled environments, distanced from real-world situations. It became an open question as to whether the knowledge produced in academic research had any meaningful effect in the real world. Towards the end of the 1980s, the HCI field demanded new and better understanding of real-world use situations and theory better suited for industry advances. The industry (i.e., IBM and Digital) started with Usability Testing where real users of software and workstations were observed in simulations. The testing was more practical
and less theoretical (disappointing for those in academic HCI work), but still produced useful results for the industry. Löwgren and Stolterman (1998) mention four general measurable usability terms:

- How the test-users manage the problem solving, how many errors are made and how long it takes.
- The flexibility of the system, measured from how many of a heterogeneous group succeed in the tasks.
- How easy the system is to learn, measured in time for solved tasks, errors over time, and how well test-users remember the system or how often they need to ask for help.
- What the test-users think of the system efficiency and helpfulness, based on their subjective opinions.

**Usability and HCI**

Usability is a central term of HCI, describing the actual ease with which computer systems can be used. The goal of usability research is to evolve and improve computer systems. In the 1990s, usability and its methods matured and improved. As new technology was developed and new real-world use situations appeared. During the 1990s, Tim Berners-Lee’s World Wide Web (Berners-Lee, 1996; Kuutti, 2009), a graphical user interface invention that came onto the mass market for PC users, and later to mobile users. The World Wide Web is now a necessary feature of the modern world. Such developments have kept the usability focus alive within the HCI field. This eventually led HCI in the last two decades to develop a closer relationship to the design community. Creating good usability can be seen as part of a larger design process.

### 2.2 What is Good Design and Usability and how is it Developed

I present here a selection of theories that represent the spectrum of design and usability. Theories are based on previously presented HCI discourse, which also contains system and informatics and links to Scandinavian interactive research.

#### 2.2.1 Design and Developing process

IT development that results in products or services has a wide breadth of aspects and complexity. There can be several dimensions of meanings, explanations, and understandings for an IT artifact. It is therefore not possible to have a standardized way of development for every unique situation. Löwgren and Stolterman (1998) emphasize that the design process has a form and structure with properties not ruled by laws of nature. Designing applications is decided largely by our deliberations and actions. The results are affected by development traditions and custom. Developers can in a sense be free of preconceived notions and locked-in concepts by learning about different design ideals and limitations.
However, according to Löwgren and Stolterman (1998), this is not enough to create a successful design in an IT artifact. More is needed to create meaning and order in a complex world. The meaning is created through structure, organization, categorization, or something that makes the complex world manageable. This is where the creation of theory becomes a helpful process that brings order and meaning out of the chaotic world.

**Vision to Specification**

Löwgren and Stolterman (1998) explain their view of the process the designer goes through in the beginning stages. Once given the task to create a solution (IT artifact) that is to solve a problem, the designer forms ideas of what should be done and what constitutes a good creative solution. But there is the reality of a situation that is both chaotic and concrete. The thinking process does not follow a rational, logical model; it is more natural and expected. The designer thinks about small details but also tries to visualize the overall solution. The different levels of abstractions are the vision, the operative picture, and the specification. They occur in a fully dynamic and complex process.

The **vision** could be a nascent idea of a technical solution, structure, or concept of basic function. The vision is only in the head of the designer and, depending on the experience of the designer, is created quickly. There can also be similar, earlier projects that help the designer create the vision. The vision’s character can be abstract, detailed, and concrete. There can be several ideas in the vision, which makes it a powerful tool. The designer can manage complex situations in which demands and wishes exist, wanting to be fulfilled. The overall vision is the first design suggestion from the designer.

The **operative** picture happens when the designer makes something concrete from the vision. The concrete operative picture takes form from simple sketches, metaphors, or other descriptions such as scenarios. What shape it finally takes is dependent on how the designer relates to and handles the vision, and the actual design situation which will limit and set boundaries on what is possible. Ongoing parallel changes - dynamic, not iterative - occur in the designer’s thoughts. When the operative picture has progressed to a certain point it can be called a product **specification**, and the development process can commence. For example, operative pictures can be built to specification level using screen shots, schematics, interaction models, and user interface prototypes.

### 2.2.2 Users and Interfaces

Considering the use of IT application **services**, we need to specify the user as a participant of a ongoing development process to successfully create good specifications. There are different types of users, but different disciplines also have views that do not exactly match. For example, the multidisciplinary research field of computer-supported cooperative work (CSCW, part of HCI's history) described by Grudin (1994), provides an example of words that can have different meanings:

In the field of human-computer interaction, “user” refers to a person sitting
at a display, entering information and commands and using the output. In the information systems field, “user” refers to a user of the output, a person who might not touch a keyboard. To deal with ambiguity, the latter field coined the term “end user” to describe a person (Grudin, 1994, p. 23).

The “end-user” is the term used when a person is using any IT application or service through any device: mobile phones, PDAs, or laptops. Interactive Research traditions such as Participatory Design include end-users as participants early in the development process. In this sense, the developers aim to work with end users rather than for them.

Although Xerox PARC had an early successful design development with the desktop PC GUI, Brereton (2009) share an additional view of the HCI and design relationship. It is only in recent years that design has been included in HCI, since much of HCI has been about usability, with the focus on functions. The software and IT applications industry has during the last decades put more focus on technology design for users, because the potential in well-designed artifacts has become more apparent. Brereton (2009) observe:

Design today faces the enduring challenge of how to bridge the development of technology with the expression of human ideals, not least such that people do not become enslaved to their technologies (Brereton, 2009, p. 100).

There is a relationship among designers and users that is now, in general, more acknowledged. Values, democracy, ethics, art, and aesthetic ideals are challenges for technological advances. At the same time, a creative developer needs to have a free hand to test, create, and explore in the development of an artifact, utilizing their own technical knowledge, “material practices”, and understanding and experience of the culture.

Binder et al. (2009) states that the design practices risk “oversimplification” since the industry still regards the user as a “recipient” of a developed IT artifact. This might explain why a designer-user relationship in its social context is not always brought to light. According to Suchman (2002), as explained in Binder et al. (2009, p. 102): “The relation between designer and user are sparse” when looking at the working relationship. Ehn (1998) (also explained in Brereton (2009, p. 101)) thinks that the oversimplification of “user” can happen, for example when the “personas” tool is used as a “proxy” for the “real” users if they are used in isolation as a static concept.

Weiser (1991) anticipated in his classic paper “The Computer for the 21st Century” that, in the near future, cheap non-personal ubiquitous computers would exist in certain places and be free for people to interact with. Following and realizing Weiser’s ideas on these user aspects, citetStreitz2008 offered the following reflection on Weiser’s paper:

Normal users are actually not very interested in interacting with computers. They are interested in interacting with information and with people in order to communicate and collaborate with them (Streitz, 2008, p. 56-57).

Streitz observed that modern technology, “ubiquitous computers”, are woven into everyday life. In essence, they become everyday life. He also noted that the Human-Computer Interaction might be somewhat misleading, since “normal users” want to “interact” with information and communicate with others. According to Streitz, there are two types of disappearances:
• **Physical disappearance** - when IT artifacts become so small that they can be integrated almost invisibly in clothes, etc.

• **Mental disappearance** - when the IT artifact is still visible or even large, but fulfills its task in such a way that the user does not experience the artifact and therefore can concentrate on the task. As Streitz puts it: “Technology disappears from our perception and moves mentally into the background” (Streitz, 2008, p. 57).

The big question, according to Streitz, is: How can we as designers of IT artifacts exploit our surroundings to help develop an augmentation of the activities that occur?

### 2.2.3 Ethnography

Part of understanding end user needs and developing better specifications comes from understanding complex “irrational” social behavior. Suchman (1987) explains “situated actions” in human social systems from an ethnographic point of view:

> Every course of action depends in essential ways upon its material and social circumstances. Rather than attempting to abstract action away from its circumstances and represent it as a rational plan, the approach is to study how people produce and find evidence for plans in the course of situated action (Suchman, 1987, p. 50).

Suchman (1987) partly describes this complexity from her standpoint of “situated action”:

> The stability of the social world, from this standpoint, is not due to an eternal structure, but to situated actions that create and sustain shared understanding on specific occasions of interaction (Suchman, 1987, p. 66).

As a problem for developing Human Computer Interaction, Suchman’s observations may be taken to mean that understanding social constraints affects any unique phenomenon:

> Social constraints on appropriate action are always identified relative to some unique and unreproducible set of circumstances (Suchman, 1987, p. 66).

These statements suggest that social actions are not always predictable, because people do not behave like robots which follow exact instructions. There is inherited knowledge from previous experiences that could make people think and act differently in their environment in response to any small change. Among Suchman’s conclusions are recommendations for researchers trying to understand social systems through social studies:

> ...instead of looking for a structure that is invariant across situations, we look for the processes whereby particular, uniquely constituted circumstances are systematically interpreted as to render meaning shared and action accountably rational (Suchman, 1987, p. 67).

In relation to the N4C project, the trials meant that researchers spent longer periods of time in the field, thus experiencing the unique situation in the environment, and how it affected the application services.
2.3 Summary of Usability Theory

A term for the development of IT applications can also be called “IT design”. This development is a process that, within the boundaries of time and economy, is created to develop an IT artifact with properties of structure, function, aesthetics, and ethics for one or more interested end users.

2.3.1 What is good design and usability

 Löwgren and Stolterman (1998) have a strong preference for writing in general terms and are influenced by the fields of HCI, information, and system design in a way that very much characterizes the “reflective designer” in the Scandinavian Interactive Research and/or Participatory Design traditions. From this theoretical point of view, they argue that the design process has a form and structure with properties which are not ruled by laws of nature. Designing artifacts is something that is decided largely by our deliberations and actions. The authors explain that IT design has a meaningful role in the development of IT artifacts, pointing out the complexity in the design process of the real world and to putting words to that complexity, for which each developer needs to figure out the meaning.

2.3.2 Defining the end user

In understanding usability there are aspects that directly focus on the user. One aspect is that usability is in some way understood from users’ actions and responses to an IT artifact. Löwgren and Stolterman (1998), for example, pointed out four measurable actions where usability is judged by observers and users themselves. Streitz (2008, p. 56) pointed out another aspect, that “most of the end users simply want to use the computer as a tool to communicate with other people or process information”. This is the case when technology is seamlessly integrated into our lives, that we take it for granted, and therefore do not want extra layers of obstructive technology if they can be avoided. If such a layer is unavoidable, then the designer needs to decide how best to develop usability, using available methods and tools. Involving users at an early stage and defining them as participants in the development project can make that leap toward multi-disciplinary research when HCI is joined to the participative research and design tradition. Brereton (2009, p. 102) means that the “relationship between designer and user is sparse” when looking at the “working relation” between technology development and its use. If this relationship is not handled correctly, there is a risk that the user may get an oversimplified meaning.

2.3.3 Developing design and usability

In the creation of applications and services, there are several factors involved in a ‘good’ design. An artifact needs to be looked at in a specific situation. It is in the specific situation that the most important properties are proven to work or fail. Even if the application itself is technically very good, that does not matter if it is not understood by the end users. In the same sense, a good interface design is useless if the artifact does not
provide end users with the functionality they need. The history of computer technology development and HCI shows how the relationship between technology, user, and interface has changed over time. IT design, general design, and graphical design have included the end user more frequently than in the time before the advent of the PC and microchip computers.

While Star interface offered simplicity and clean design, Bødker (2008) correctly acknowledged that as desktops evolved they became more and more cluttered and the principle of simplicity was lost. Simplicity is one of the critical factors when designing applications where functionality is the goal. If applications are too complex, failures in either the system or the end user’s activities are more likely to occur. Bødker (2008) suggested that the design of the interface of the artifact could be an important process in which the developers need to take care, since usability development is complex and contains several layers or dimensions, as Löwgren and Stolterman (1998) also stated.

That process - design of usability - can then be put into words using Löwgren and Stolterman’s (1998) philosophy about artifact development. There is the vision which involves abstract, detailed, and concrete ideas. There can also be several ideas in the vision that make it a powerful tool because the designer can manage complex situations in which demands and wishes exist, wanting to be fulfilled. When the designer wants to create something from those ideas, a concrete operative picture takes form in simple sketches, metaphors, or other descriptions such as scenarios.

In a complex developing situation, much of the quality depends on the designer’s ability to understand the unique situation. Suchman (1987) writes:

\[\ldots\] instead of looking for a structure that is invariant across situations, we look for the processes whereby particular, uniquely constituted circumstances are systematically interpreted as to render meaning shared and action accountably rational.

Suchman recommends that researchers should look for processes that create unique circumstances and which can be systematically interpreted for a rational understanding of the phenomenon. In practice, this normally takes place in the field, researching situated actions that present ‘reality’. Following that, researchers can make use of their experience in the next iterative development steps, to improve their research based on the new knowledge. The main conclusion for social network theories is that, by using an ethno- graphic, i.e., real-world, point of view, researchers can begin to explain how the real-life testing experience affects parts of the development process.
3 | Method and Iterative Development Process

In this section, I present the methods and work phases that have been used throughout my work in functional and field-tested DTN applications. First, part of my methodology comes from my own background in and knowledge of systems/informatics development. Second, the research subject and the faculty to which I belong have their own traditions in qualitative research, which have had some effect on how I do my research and development. Thirdly, the N4C project, which has brought together researchers from different research schools, has made its mark as a project which deals with cross-cutting issues while being multi-disciplinary. In this section, I focus on these three methodology inputs, which were used to collect data and provided support for developing and testing DTN application prototypes.

3.1 Informatics Technology and Design

One of the reasons I was chosen to work in the N4C was my educational background and interest in software engineering (SE), system informatics (SI), and SI methods. Material in the field typically contains many theories, methods, and explanations. One quickly learns that there are different ways to look at development, e.g. in a bigger project. In this section, however, an explanation of SI and software development is appropriate to give an idea of my origins and path towards the inclusion of Participatory Design (Action/Interactive Research) in my research.

A description of software development from an engineering standpoint is found in Software Engineering - A Practitioner’s Approach by Pressman (2000):

Computer software is the product that software engineers design and build. It encompasses programs that execute within a computer of any size and architecture, documents that encompass hardcopy and virtual forms, and that combines numbers and text, but also includes representation of pictorial, video and audio information (Pressman, 2000, p. 3).

This is the description of the technical details of development, not the methods. I brought this basic understanding of computer software development to my work as a software developer and researcher in N4C. Software engineering tradition nevertheless contains different kinds of models and methodologies that developers and researchers can choose to apply in their projects.
Generic Software Engineering

In a generic view of development, regardless of the project size, computer software development is part of an engineering process which can be, according to Pressman (2000), specified in three phases:

- **Definition Phase** – Focus is on *what*. The engineers need to specify functions and desired performance as well as functionality. This includes interface design, constraints, and any key requirements that might affect system behavior. Tasks include information engineering, software project planning, and requirement analysis.

- **Development Phase** – Focuses on *how*. The developer needs to figure out and design how the functions will be implemented in the architecture. This includes interface design and software design transition into programming code. Three major tasks occur here: software design, code generation, and software testing.

- **Support** – Here the focus is on *change*, the iterations of the software. It includes four tasks: correction, to correct discovered mistakes; adaptation, which keeps software up to date in a changing environment; enhancement, adding functions for the user as extensions beyond the original requirements; and prevention, re-engineering of the software as necessary to ensure it is more stable and tolerant over longer periods.

In daily work (i.e., office lab work, coding, meetings, and planning), the method of producing a prototype is like the Software Engineering *process layer* which, according to Pressman (2000, p. 19), is the foundation of software development. This process defines a framework with a set of key processes that establish how to use methods, quality assurance, milestones, and reports, etc. Then the tools, which is used to help develop software in the daily work, e.g., programming. The methods are the technical “how-tos” for the work. The overall strategy for putting everything together is called the model. The model for the N4C project is described in the following subsection.

### 3.1.1 N4C Main objective and model

A central objective in the N4C project was to build a field lab environment containing a prototype IT network, using advanced research and technology suitable for the task. To successfully build this advanced network, N4C needed to have a good testing platform. Such platforms were setup in the Sami area in northern Sweden and in rural areas of Slovenia¹. N4C took advantage of recent research in computer networking (DTN), which was shared among the partners in the project. These partners consisted of technology developers, business partners, system developers, local business users, and other end-users. In its work plan, N4C noted the importance of contact between the project, the developers, and the local community in the test areas. This understanding brought the partners together as a team, allowing us to focus our work on the most important tasks, and resulted in the choice of a software engineering model.

---

¹See test bed creation methodological report (Udén and Grasic, 2010).
The spiral model

In a project like N4C, a development model (Figure 3.1) can be a helpful guide for most of the partners. It helps explain the project development phases, and encompasses the methods, processes, and tools. The spiral model was chosen and accepted as best suited to represent the N4C development phases\(^2\). In the paper “A Spiral Model of Software Development and Enhancement” (Boehm, 1988), the primary function of a software process model is explained:

…to determine the order of the stages involved in software development and evolution and to establish the transition criteria for progressing from one stage to the next (Boehm, 1988, p. 61).

This process model aims to determine the next step and its preferred length. It provides guidance as to phases, increments, prototypes, and validation for the N4C project.

![Figure 3.1: The Spiral model (Boehm, 1988)](image)

---

\(^2\)Development phases as written in N4C (2008)
Pressman (2000) observes that the Spiral Model:

\[\text{\ldots is an evolutionary software process model that couples the iterative nature of prototyping with the controlled and systematic aspects of the linear sequential model. It provides the potential for rapid development of incremental versions of the software (Pressman, 2000, p. 35-36).} \]

The N4C project partners had such a focus on iterative development, testing application prototypes in several steps (three major summer tests and smaller winter tests). The model has a number of framework activities (usually three to six):

- Customer communication: For N4C researchers this meant maintaining good relationships between the participating end-users and the researchers.
- Planning: Defining resources, timelines, and required project data.
- Risk Analysis: Assessments of identifying and managing technical risks that might arise during the project.
- Engineering: Building and developing DTN application services.
- Construction and release: Tests, trials, installations, and documentation of the DTN and teaching end-users how to use it.
- Customer evaluation: Obtaining feedback from the participating end-users who worked with the applications created and implemented in trials.

\textit{Unified Modeling Language}

According to Pressman (2000), the Unified Modeling Language is a syntactic, semantic, and pragmatic way for a software engineer to express an analysis model. It is a useful tool for most software development processes because it employs defined symbols (syntax). Each symbol has a clear meaning, which makes the language comprehensible for almost anyone. Since UML has been widely used in software engineering for the last couple of decades, N4C-LTU and Folly Consulting (who initially created the first technical specifications) chose this modeling language to describe and demonstrate what the web application prototypes could do from the end user’s point of view.

\textit{UML user model view}

Models created at LTU are presented in use-case diagrams called user model views. These diagrams show how a system (i.e., DTN web scenarios) works from the end user’s perspective. The scenarios were first developed from ideas or previous work and formulated in a technical text description. These descriptions were then analyzed to identify the actors and objects. The actors can be people or machines, who have a certain role in the system as operators but may not necessarily be end-users. The diagram is then constructed using the UML syntax and is intended to be visually descriptive while avoiding unnecessary complexity. In the description of the N4C-LTU prototypes, the use-cases are built using a simple selection of UML symbols, which does not exploit all aspects and possibilities of use-case diagrams.
3.2 Qualitative Research Methods

Researchers in N4C had many questions and discussions regarding how to create applications that the end-users really needed and could easily use. It meant that we had to find out what counted as good usability and usefulness among the participating end-users. Identifying the different types of users was a significant part of this, as well as identifying what they would do with the new technology. There were uncertainties as to how to solve problems in the best possible way (which is, of course, what research is all about). Development at the beginning created applications based on the initial ideas identified in the predecessor SNC project. Within N4C, there are different participants: some have good knowledge of how the inhabitants (users) move within their physical environment (the Sami people, for instance). Others are skilled in the technical aspects, such as programming and hardware. Of course, there are different kinds of “views” on how an information system will be tested and where the focus will be, which should come as no surprise, given that N4C is a big project involving many partners and encompasses implementation of state-of-the-art technology and significant cross-cutting issues.

DTN application development - which was obviously an important part of the project - can be difficult at best, since it involves how the end-users look at, and use the DTN technology to assist them and enhance their lives. It was a question of gathering knowledge within the N4C consortium, obtaining external knowledge, and the successful execution of the field tests. This could also be a way of involving end-users, by inviting them to participate with the developers, and provide more opportunity to benefit from their knowledge and lifestyles. Other issues included learning how to assemble project members’ knowledge and learning how to communicate efficiently. In the following subsection I address how N4C-LTU resolved these issues.

3.2.1 Methods of Action Research and Interactive Research

Development of the N4C-LTU DTN applications employed the methods of Action Research (AR), a term coined by psychologist Kurt Lewin in 1946 (Aagaard Nielsen and Svensson, 2006). Lewin wanted Action Research to lead to a better society and better work life, and be viewed as promoting and defending democratization. It is meant to help the research practitioner in the study of concrete problems and situations:

Research is an activity and a knowledge base of the action researcher him/herself, running parallel with his/her interactive involvement in the help-seeking client

Presenting action and change in a planned manner is important. The researcher uses different phases in a spiral-like cycle. A general or initial idea is identified, followed up by fact-finding and planning. The first step of the action is completed and evaluated, followed by planning, secondary action steps, and so on. The end goal is to help a community change conditions for the better, by validating steps, communicating value judgments, focusing on group processes, and resolving conflicts.

The most important distinction between Action Research and other research is the researchers involvement in the whole action process. The Action Re-
searcher becomes the *change agent* (Aagaard Nielsen and Svensson, 2006, p. 50).

Action Research emphasizes the collaboration relationships both *between researchers* and the *researcher and client*, in which different goals and/or interests exist. Ethical considerations in these relationships need to be addressed, which may not be easy, but is nevertheless a vital aspect of Action Research theory and methodology.

The original method of Action Research has been improved and updated over the years into different branches, together with Scandinavian research tradition (i.e., Interactive Research and Participatory Design). Generally, in current Action Research theory, one differentiates between normal scientific research and Action Research as Mode 1 (traditional research) and Mode 2 (Action Research). Aagaard Nielsen and Svensson (2006) associate Mode 2 research with the Nordic context because of the growing demand since its debut in the 1970s.

Mode 1 research is often discussed in Action Research theories as the traditional, stricter “high science” of academic theories. It is not usually open to non-academics; the content and methods are decided within the academy and the research is often peer-reviewed. The aim is to create general knowledge which adds to existing knowledge.

This research is based on the scientific discipline with professional rules concerning how to undertake research. The relationship is hierarchical - both inside and outside the system (Aagaard Nielsen and Svensson, 2006, p. 21).

Mode 2 research is characterized by the focus on usefulness for participants, equality, problem-based multi-disciplinary research, an open approach, and variety so that more “robust” knowledge can be created. The ultimate responsibility is towards society rather than the scientific community, so the research approach is in an open form (as contrasted with “closed”, as in traditional university research). Mode 2 research is flexible and dynamic, and often makes use of new information and computer technology (ICT). The increased demand for Action Research can briefly be explained by the growing complexity in society, and new perspectives on innovation and change.

Interactive Research is the Swedish version of Action Research (Aagaard Nielsen and Svensson, 2006, p. 153). Interactive Research focuses on the *joint learning* between participant and researchers, throughout the whole research process. It is important that the knowledge produced be of very high quality, as is also expected in traditional research. The aim here is to effect change from and use the knowledge during the process, in which the participant has an active part.

The similarities between Action Research and Interactive Research are many, but they are not interchangeable research terms, according to Lindberg (2010), who interpreted Professor Ewa Gunnarsson’s analysis of the differences. Gunnarsson observed that the Nordic tradition of interactive research contains a more mutual learning process between researcher and participant. While Action Research focuses on a development for the participant, the interactive method concentrates on development with the participant.
Svensson et al. (2002), in “Interaktiv forskning för utveckling av teori och praktik” (Interactive research for the development of theory and practice), state that joint learning is something that goes on between the researcher and participant during the entire research process.

I have no problem describing the work methods of N4C-LTU as Action Research oriented, but think that Interactive Research or Participatory Design might be more appropriate terms. Participatory Design describes the overall method inspired by Action Research theory. Participatory Design has been used by Scandinavian researchers since the early 1990s with roots in human-computer interactions (HCI) and earlier research that relied on local trades unions’ involvement (Sundblad, 2009).

### 3.2.2 Focus Groups

Focus group techniques have been used by researchers in various ways since the 1920s. Traditional focus groups have their roots in marketing surveys on the usefulness or otherwise of various commercial products (Wibeck, 2000). A focus group is used to gain knowledge about participants/users of the product, starting with a predefined subject of discussion and debate within the group. The method is simple. A group of people (i.e., end-user participants) is gathered, and the group interaction is started and led by a moderator using some kind of stimuli, such as interesting questions, that inspires the discussion.

From the 1980s onward, focus groups received more attention and interest from the social sciences, and are now employed in many disciplines. Wibeck describes focus groups as a type of group interview, but with the distinction that not all group interviews are focus groups. A definition by Wibeck states:

> Focus groups is a research technique where data is collected through group interaction about a subject that is decided by the researcher (Wibeck, 2000, p. 23).

In N4C during the summer trial in year three, focus groups were used as an additional qualitative, interactive method to collect data about the end-users of the application, their social situation, and DTN usefulness in general. Spontaneous discussions about the subject in the field between researchers and participants/end-users had also occurred, but I would not consider them as providing knowledge gained through focus group research since the subject was not moderated or predefined. As Wibeck points out, a focus group discussion needs to meet certain requirements:

- **Subject** – The group discusses and debates a given subject that is pre-decided by the researcher.
- **Moderation** – To ensure everyone in the group speaks and that the discussion is progressing in a balanced structure, the moderator presides. The more active the moderator, the more structured and controlled the discussion.
- **Participants** – Participants should have a meaningful relation to the product or service and should represent the actual users in gender, age, and social status.
The main reason for utilizing focus groups is that the qualitative data that arise from the discussions can contribute to the fact-finding process. It was also a time-saver for the LTU team, since individual interviews, paper surveys, or observations would have taken more time to produce the same quality of data.

No official environment is needed for a focus group so, in the N4C-LTU case, we held sessions in participants homes within the test area\(^3\). Two of those sessions were held with all the included families who were the end-users of our application prototypes. No recording devices were used; instead, notes were taken by the N4C-LTU moderators. In these sessions, we did not have an unlimited electrical energy supply so pen and paper were the best way to record key sentences, though it was not easy or perhaps possible to keep track of everything that was said.

---

\(^3\)Professor Ewa Gunnarsson assisted N4C-LTU team with the focus groups in field in the Padjelanta Sami village
3.2.3 Field Notes/Journal

Field journals or notes are often used in qualitative research. These methods are explained in Widerberg (2002). Journals and notes are simple data collection methods that researchers can use in the field. In field journals the researcher writes down impressions and memories of events that happen during the day, normally using pen and paper, or a computer. Journal writing is often raw and not very structured; it is basically a memory dump that has not yet gone through any real analysis. The important thing is to document recent events that are noteworthy. Input method is normally by computer or paper and pen.

Team Discussions

During the years 2008-2010 I had participated in the N4C project meetings held in Slovenia, Ireland, Portugal and Norway and in Sweden. The meeting types were both official and in-official, and was even held during the trials when the team had time to talk, during those times. In the official meeting I could get some feedback on the DTN services I was working on. In the field, when we met the feedback was more of a mix of experiences shared with other project partners. Most of my feedback that came with talking to Samo and Fritz-Åke in field or lab. The other, personal feedback came from taking notes of impressions and experiences that had happen during the day in field/lab.

The feedback from discussion in working sessions regarded my work in the work package 3 and 8. The early discussions in 2008 with Elwyn Davies of Folly Consulting led to the web services UML drawings. These set the base for what I was trying to achieve with the DTN application services in WP3 on a higher level. On a lower level, the actual implementation and coding, setup and testing, I discussed with Samo for most things. Because it was Samo and I working on the LTU part of N4C DTN, we had to be a close working team since my application services had to rely on Samo’s code parts.

Notes and journal from the field trials

As a researcher in the N4C field trials, my handwritten notes created a mine of data collected from impressions, observations, events, and interactions. It was the most reasonable choice for documentation, since field work was time-consuming. This meant that notes were written at the end of each day or sometimes during the day. Natural limitations decreased the efficiency of this method. By evening, battery power was often depleted, so I resorted to handwriting notes in total darkness. Another limitation was tiredness: long working days led to exhaustion and the urge to sleep rather than write notes in the dark. Even with these limitations, field notes were useful when later assessing data collected in the field, which eventually led to iteration in the N4C-LTU DTN applications and services.

The written personal notes and diary in writing that I did stand for a large part of feedback that cover most subjects, even things was not directly affecting the DTN services development. The journal notes covers, for example, people we met, and errors or results from lab work, field trials and such. The code examples (see appendix A on page 106) are something that was progressed from writing notes in the lab environment.
3.2.4 Tools for knowledge sharing

Development of the DTN application prototypes discussed in this dissertation was strongly influenced by how learning tools and communication were used among researchers in and around N4C. The project had many work phases proceeding simultaneously in different locations, such as project partners working on different milestones. N4C partners worked together both during workshops and in field tests and preparations, so there was a vital need to communicate and share existing and newly-created knowledge, both technical and non-technical, which needed to be discussed and made available for some or all partners in the project. The objectives of meetings and workshops was to ensure we attained a common ground of understanding for adequate preparation for the field tests. This synchronization was critical since we were setting up a field lab environment that would run for longer periods.

There have been trials using the new platform within the project, but these trials have usually included only two participants. Building a Wikipedia is another, albeit indirect, way to communicate knowledge, though such collaboration results in static web pages. The software to create a Wikipedia is free and it can be hosted anywhere on the Internet. N4C’s Wikipedia was hosted on a UK-based server to create a central meeting point for developers to share technical information.

The most common method for communicating person to person within the N4C project was through e-mail, because it is simple, ubiquitous, everyone is accustomed to it, and it does not rely on third-party assistance (i.e., Google) unless one specifically chooses this. Moving meetings, workshops, and work to online platforms does, however, present some problems. One is that face-to-face dialogue disappears and that can make it harder to understand everything that is said. Another aspect learned from the N4C experience is that physical interaction with DTN equipment and preparation among involved partners is difficult when it takes place at a distance. Online meeting was seen as a helpful and useful complement to physical meetings and hardware interaction, but not as a substitute for them.
3.2.5 Methodology Summary

Software Engineering
Software Engineering was a big part of N4C’s technical development phases. There are two reasons for including its description and some of its methodologies in this chapter. First, this is the tradition that explains different work models such as the “Spiral Model” (Boehm, 1988) employed as the core development model. Second, my educational background is in systems and informatics which directly involves the tradition, so the decision to apply Software Engineering concepts was for me straightforward.

Human-Computer Interaction
Human-Computer Interaction (HCI) with its strong usability and interface focus is an important field for me as an application developer. In general, the field explores the usability subject with elements of design, ethnographic studies and methods similar to Participatory and Interactive Research. Its history serve the theory for explaining the general development of DTN. Through HCI, the N4C project ideas and ideals can be better understood, by relating to the historical context.

Participatory Research
Participatory Research (as a type of Action Research) was the ideal method for gaining knowledge about the end-users and their environments. My field work included testing DTN service prototypes and figuring out reasonable specifications with respect to how the participating end-users live and communicate in their environment, rather closely resembling ethnographic studies. Further reasons for this qualitative approach were practical, incorporating the Work Science faculty research tradition, and the multidisciplinary nature of the N4C project. The technical experts chose the Spiral Model as the framework for the process as a whole, and for guiding the creation of milestones, reports, risk analyses, and technical application development.
As the figure above shows, the development of the application prototypes has been a mixture of applied Software Engineering tools and methods, and Participatory Research. It has been an ongoing process in which none of the research traditions or methods have been strictly adhered to. My work with the DTN applications and the following final prototypes should be viewed rather as layer upon layer of the different methods. In figure 3.2, it is Software Engineering, the middle part, in which N4C work models exist, and encompasses prototyping, milestones, work packages, reports, tests, etc., that contributed to all that work in N4C-LTU. HCI and Participatory Research gave support with qualitative “how-to” methods, and technical information that placed my work in historical context, and provided the “why” with respect to the prototype’s usability.
3.3 The Iterative Development Process

We have created and tested five DTN applications and services during 2008-2010: Web-caching, podcasting, Not-So-Instant-Messaging, localized applications and Web Graphical User Interface. The collection of empirical material for this work started in 2008 in my first year of employment at LTU as a field technician, when I participated in the first N4C summer trial. Here I present an account of my journey from 2008 to 2010, focusing on the research topic of usability in DTN end-user applications.

3.3.1 Year One

I consider system development and, in particular, communications technology very exciting, and have an interest in developing various IT systems and applications. My undergraduate background in systems sciences (also described as systems and informatics) includes theory with a focus on users, such as exploring close cooperation with all stakeholders and identifying what needs to be done by analyzing the problem. My education also covered practical things like application development and programming in various languages.

During my first months on the project, I familiarized myself with the technical part of N4C and its different work packages. This meant learning new technologies within a limited time. I was familiar with computers and networks at an amateur level but in my Ph.D. work I soon realized that the N4C DTN “system” would mean new challenges for me, for my co-workers, and the project itself. In 2008, I learned that the focus of my work would be web applications, which made sense in the context of my own interests and educational background.

However, I also needed to help build and set up the DTN, the system to support the applications in the WP3 (see section 1.4). Exactly how all this was to be done was not obvious in the autumn of 2008. Luckily, good work with my colleague Samo Grasic resulted in a good start. Grasic had been involved in the earlier SNC project (see section 1.3.2), in development and testing. The first summer trial saw us testing DTN in the mountains of northern Sweden. Our plan was almost overwhelming: we had to prepare for two weeks of setting up and maintaining DTN services for tourists in the Padjelanta area, without knowing much about the hardware, the applications, or the geographic area.

Technical preparation
The first summer trial started shortly after the N4C project began. The schedule was therefore tight. I had to learn about and set up the test equipment properly and quickly. In fact, much of the test preparation was purely technical and not user focused, since we had to quickly recover the hardware and software used in the previous SNC project so that the trials could be started. That is not to say that real-life usage of the system was not planned or considered; it was present as a set of goals, and to reach those we needed to focus on the technical side.

Someone had to take care of all this equipment and software - us. The amount of work
just to get the system up and running in the lab environment was unexpected. New hardware and software have a tendency to not cooperate during initial stages. Many hours were spent installing and investigating small errors or faulty settings in the software.

To give a sense of the technicalities we were dealing with, one example involved the main N4C hardware. I was introduced to this hardware consisting of so-called “wraps”, a hard shell containing the electronic components and memory for the operating systems, DTN software, and applications left over from the SNC project. The wraps were the size of an average home router, but with a stronger shell for protection from the harsh outdoor environment. The process of installing and configuring the wraps was especially time-consuming since they had no display output. The only way to see if they worked was to use another computer to “ping” the IP-number of the wrap or directly connect with an SSH login. Configuring the wraps with the new N4C software required learning new ways of extracting old data and copying new data to the wraps’ memory units. We worked out a way to use other computers to read and write to each wrap’s removable memory storage.

Preparation of end-user computers
Computers based on Windows and Ubuntu were prepared for field use. Most were Compaq tablets which had been used in the SNC project. They were lacking in both memory and performance, i.e., they were generally slow and did not respond very well but, more advantageously, possessed protection sleeves and were somewhat portable. They were basically useful for the task and readily available. Not all of the tablets were in a usable state. Some had to have the operating system reinstalled, while others were too damaged to use. The tablet chassises also had some wear and tear from the prior SNC field tests. This was an important warning to us that any equipment used in the field should be as insensitive as possible to minimize hardware faults.

Software Resurrection
The software had to be restored from what was left after the previous SNC project, and then tested with current and new hardware. DTN functionality consisted of an updated version of PRoPHET (Grasic et al., 2011) (Grasic’s implementation of the DTN routing protocol). The applications to be restored were e-mail and Web caching. These two applications had been left as they were in the SNC project. Grasic and I reverse-engineered the e-mail and web cache using some tutorials and descriptions of the software that were luckily available on an old SNC server in Jokkmokk municipality, as well as consulting Jan Lindblom, one of the SNC staff, who had developed and tested earlier versions of web caching. Not much additional work was done on the e-mail or web cache, since the main focus was getting it operational, ascertaining how it worked, and configuring a version of the application that would work with Grasic’s PRoPHET. In the field, we had to test:

- How to send and receive e-mail,
- How to browse cached websites,
- How to update our test blog, and
- How to log and analyze the traffic over the network.
The N4C-LTU network consisted of the wraps, a gateway server, and some handheld devices. The aim was to make it as simple and fault-tolerant as possible, so that it could be maintained in the field. For example, as the web cache had been untouched since the SNC project, it no longer worked. We examined and interpreted the scripts, removed some of the script code, and wrote some of our own. This resulted in a simple, yet effective version of the web cache application which would send a compressed file of downloaded web pages to the DTN PRoPHET software. The user would be welcomed by a simple web page with a few links, each of which would lead to a specific website. In the LTU lab, we tested the web cache functionality to ensure that the gateway downloaded and cached web content from the URL list as instructed. We would sometimes trigger this manually. The list of websites was kept short, comprising only those we thought the users would most like to access, such as the daily news.

We also ensured that the overall size of the data transferred would not overload the DTN and excluded some website objects that would otherwise have been downloaded. The web pages’ presentation in the end user’s web browser created a usability issue. Because of the complex nature of the gateway’s wwwofle (see page 71) application hosting the cached web pages, pages were sometimes stripped of video, audio, or other large embedded objects. Recursive fetching of the web pages presented another usability problem. If set to level one, the end-users would only be able to read the front page. Clicking a link would result in a “not found” page. If set to two levels, a user could click from the front page to a sub-page. However, the web page would expand so much that the files would be too large for DTN transfer. We had to choose with care what pages to include in the cache. In the end, we integrated simple one-page news reports readily available from various Internet news sources where all the articles could be viewed at once, as we now see in mobile phone web browsers.

**Practical preparation**

Apart from getting the DTN system and applications up and running, the first year presented another complexity: the test areas in Padjelanta and Sarek. I had little or no idea of what to expect from the location, topography, natural features, or weather, let alone the inhabitants and hikers. In 2008, most of my knowledge came from word of mouth and the *Description Of Work* (N4C, 2008). I had to rely on other people’s information. Most of my own field experiences are reported in the “Test bed creation methodological report” (Udén and Grasic, 2010), used for guidance for the last N4C summer trial. However, at the time, my lack of direct information resulted in time-consuming preparation of clothes, food, transportation/logistics, packing, checklists, personal items, and communicating with the local Sami community.

The first visit to the Padjelanta trial area required a full day of driving from Luleå to Ritsem and back. The purpose of that trip was to meet with Tannak, who gave us a quick overview of the conditions in these Arctic mountains. At Ritsem we continued by helicopter to the more remote areas. These field ventures became an important source of knowledge for me as a researcher and software developer. For instance, without such meetings I would have been far less prepared for the mountains’ climate and terrain. Even though the first summer trial was to be relatively limited, one of the more difficult challenges was deciding how to make the application services available to tourists and
residents in the test areas. Our two weeks of testing and service provision were split, the first week conducted in Saltoluokta and the second in Staloluokta. Together with LTU, TANNAK, and MEIS, the most closely-involved partners, we chose to test the DTN during tourist season, when the weather conditions are more amenable.

Week one – Saltoluokta
Saltoluokta provided an easier starting point for testing, since the environment is more pleasant, there are more tourists, and the accommodations were better, compared to Staloluokta. The first location provided better electrical, Internet, and telephone services to better support the testing of the new version of the PRoPHET software. We directly accessed the Internet with the NMT modem, though sometimes the signal was not uniformly good. Significantly, the NMT signal was in most cases non-existent in Saltoluokta so that the lack of an Internet connection there made it a real test of the technology.

There were more than software issues involved in this challenging test. Grasic and I drove a fully packed combi to Jokkmokk and on for another couple of hours to the Saltoluokta ferry, then collected by Tannak who guided us to our destination. The first day we settled in, finding our home for the week and meeting with the residents who provided accommodation and food. We set up a tepee that was used mostly for sleep and resting. A business owned by a Sami woman known for her traditional cooking style prepared all our meals. Tannak helped us set up a larger tepee as an “internet café”. The Tannak team, consisting of the two owners Susanne and Karin and their employee Birgitta, drew potential users to this café, preparing free waffles and coffee. Illustrated in figure 3.3 is the tepee and entrance to the “internet café”. Inside the tepee, handheld computers and tablets were used by both tourists and residents, connecting them via WiFi to the gateway (hotspot), that served the devices with daily-updated web data. Every visitor could try out the web and email services, using the N810 or one of the tablet PCs. Figure 3.4 shows a simple Web cache schematic.

Week two - Staloluokta
The three-day hiatus between week one and week two was useful for recovering from the week in Saltoluokta, though there was still much work to do. Travelling to the new location at Staloluokta village meant driving to Ritsem and flying in by helicopter. Staloluokta was a more demanding trial area because of its climate and the distance from infrastructure. As in Saltoluokta, however, this village was full of hikers during our trial period. We brought our own power sources to Staloluokta, two solar panels and three car batteries, which were all plugged together so that the panels charged the batteries during sunny conditions. We had to carry the heavy batteries and the non-ergonomic solar panels on rocky paths around the village. We had to bring our own food, along with more clothes. The Tannak team were with us during this week, too. Our accommodation comprised one room with six beds.

Helicopters transferred data between the Internet and the village. We planned the trial with the Fiskflyg company. The pilots were very accommodating and enthusiastic about helping with the tests, and being part of a project of this scale. An Internet
Figure 3.3: Network topology in Saltoluokta 2008 (I appear at the bottom right holding the battery-powered NMT modem, trying for an Internet connection). Source: Grasic and Grasic (2008).

Figure 3.4: Schematic of the web services offered in the 2008 tests in the tepee at Saltoluokta.
gateway was installed in a shed at the Ritsem Fiskflyg helicopter base and the DTN nodes mounted in the helicopters themselves proved to be very useful for sending and receiving data. However, the software in the nodes needed to be adjusted to suit the new network topology.

Figure 3.5: Network topology at Staloluokta. Nokia mobile smartphone, left bottom, connected to the local hotspot, which passed on the data to the helicopter hotspots once they landed in the village. The helicopter hotspots then exchanged data with the Internet gateway at the helicopter base at Ritsem. Source: Grasic and Grasic (2008).
Software/hardware updates
For a functional web caching system, the gateway scripts had to be updated to automatically download new website data on a daily basis. This was triggered well before a helicopter’s take-off at 14:00 pm every day during the tourist season. These updates were made during the weekend, building on experience in the first week of observing how the DTN behaved in the field. The conditions in Staloluokta (see page 16) meant that we had to bring our own power sources. For this we had two solar panels and three car batteries, which were plugged together so that the panels would charge the batteries during sunny conditions. This entailed a great deal of carrying the heavy batteries and the non-ergonomic solar panels on the rocky paths around the village.

Figure 3.6: Arrival at Staloluokta of the LTU/Tannak research team, unloading equipment. Photo by Samo Grasic.

Interacting and working conditions
While the first week gave us chances to interact with hikers and residents, the second week in Staloluokta proved to be even more fulfilling socially. Every day and evening, the whole team (Tannak and N4C-LTU) met at the local tourist station’s social/dining room, where between 20 and 30 people might congregate on busy days. We usually approached people after lunch and after dinner. Some people asked “why” and “how” questions about the project, others were interested in the technical details. We tried to answer all questions with as much as detail as possible, inviting them to try the Nokia 810 smartphone. Sometimes people came to our cabin to try sending email or browsing web pages. News and weather forecasts were of interest, as we had expected. The majority of users were hikers, but some residents, including cabin keepers and villagers, showed interest.
When not interacting with end-user participants, we kept busy maintaining the DTN. The computer batteries drained quickly, and constantly had to be in charging mode. However, the weather in the area is not always sunny, which meant we could not rely on the solar panels and thus use the devices. If something went wrong with the web caching, we would not likely know exactly what had happened if it occurred outside the village. Our only reliable source of information was the data transmitted daily from the helicopter nodes. If something was missing or no data was transmitted at all, it was almost impossible to know what was happening, since it was impossible to call or use the Internet directly (obviously). We could have asked the helicopter pilots to check the basic concerns, whether the computer was on, or if it responded to mouse movement. Simple hardware failures happened a few times.

Figure 3.7: The Web service scheme, showing the gateway to the left placed in Ritsem, and the mobile gateway (hotspot) to the right placed in Staloluokta near the landing place. The links or “mules” are the nodes carried by the helicopters. The Nokia N810 (and occasionally the PC tablet) devices operated by end-users were connected via WiFi to the hotspots.

3.3.2 Test outcomes year one

Analysis of the test outcomes by the LTU team for both weeks formed the basis for the next logical steps in the N4C project. We learned a great deal just from the Saltoluokta test. As all researchers know, field work is completely different from testing services in a laboratory environment. In perfect lab conditions, it was possible to get the equipment to work; despite the work involved, this was more or less straightforward. When we deployed the system in Staloluokta, things did not go exactly like clockwork. This issue is also discussed by Grasic and Lindgren (2012).

There were many external features that made the trial more difficult. The weather presented its own challenges: even during summer, the Arctic can be chilly, especially when sleeping in the tepee during week one, with no external heating system. Another problem is exhaustion, the result of working long days in an unfamiliar environment.
Weariness sometimes made me prone to error and forgetfulness. Most of these field experiences are reported in the “test bed creation Methodological report” (Udén and Grasic, 2010), which was used for guidance for the last N4C summer trial.

**Unexpected user/environment feedback**

Some people wanted to send emails from their own Internet mail accounts (e.g. Hotmail or similar), or they wanted to browse websites that were not cached or available in our DTN. We needed, therefore, to consider how we might expand the services for the next trial. For example, slow user interfaces suggested a need for suitable devices with faster, more reliable interfaces. In addition, the Nokia 810 and the tablet PC were poor outdoor equipment. The screens react slowly when used in wet or moist conditions, and are best for inside use, as most computer equipment is. We were even prompted to think about winter trials. One night, the temperature fell below freezing point, so that in the morning I noticed that my laptop, an aluminum MacBook had stopped working. It only displayed erratic pixels that made no sense. This phenomenon, which I labeled the “apple freeze”, made us aware that even slight drops in temperature can affect electronic devices quite negatively, to the point of loss of function. Battery power was another highlighted issue; our electrical power was significantly limited throughout the whole trial in Staloluokta. We realized that this would present a problem for the next trial. We experienced minor technical problems such as wireless connection drop outs and slow hardware when writing to CF-cards. Many problems that did not show up in the laboratory trials were manifested in the field, so the quest for stable and thorough technical solutions became extremely important to support the overall usefulness and usability of the DTN.

**Summary year one**

The trials taught us much more about the physical environment and what it is like working in the Arctic mountains 4. We confirmed the usability aspect in that we presented new technology to end-users, who were interested in being able to read news from reasonably “fresh” websites, and to contact relatives and friends via email, even if the services were only experimental. At this first trial stage, the use of these services was always under LTU and Tannak supervision, and was far from perfect as a pervasive DTN service. The trial established that it is at least technically possible to deploy and run a DTN in such an environment, and that services such as web caching and e-mail are attractive and desired by both tourists and local residents.

---

4A technical report for EU was submitted with our results (Grasic and Grasic, 2008)
3.3.3 Year two

The main goal of the second summer test was to expand on the first summer trial with updated software and hardware where needed. The preparations for the second trial were more “user” focused. Additionally, we had much to learn from the 2008 technical start-up meeting in Slovenia, where several topics were discussed. Another crucial factor was the inclusion of input from project partners who kept in contact and planned the progress of test preparations and prototypes.

In a project focused on user-driven development, I had to learn how, where, and when to collect the knowledge gained from the experience. This became more vital when we were developing new functionality, rebuilding the DTN services for the second trial. What did “usable” mean in a DTN application? It was necessary to separate terms like “user” and “usability” and define them in the context of N4C. Where in the N4C project does the user come in, and how do we know if our applications are indeed usable?

My view after the first trial was that the term “user” has a broad application. For developers at LTU, the term “end-users” came to signify the Sami population and the hikers that tested our DTN services ‘on the fly’ from time to time, or during a longer period. We could describe ourselves as users, since we had to continuously use the system both in the field and the laboratory. Usability included the mixture of technical features that had to be implemented in the DTN applications from both existing knowledge and theoretical material. The usability would be affected by users’ input, and the available time for research and development. The work had to focus on the core functionality of the services, but also provide something of value for the end-users. This made for an iterative development process where end-users became a part of the “participatory design” in the overall work of N4C WP8, where we addressed everything that arose in the field trials.

Individual progress

As I became more familiar with my own goals within the project and my Ph.D. studies, I had help from my Ph.D. supervisor of the time, Avri Doria⁵, a researcher in both SNC and N4C, who co-invented the PRePHET DTN protocol. We created a course that introduced me to a deeper level of the architecture of the World Wide Web, and Web 2.0 communities, combining practical knowledge and technical theory. The material was chosen to provide me with a theoretical base when developing the applications prototypes for the DTN. This course supported me during the development phases, along with support from project meetings and discussions in which we exchanged ideas and suggestions.

Preparation

During the preparations for the second summer test we acquired a new LTU team member, Fritz-Åke Kuoljok, a reindeer owner, as the N4C field test manager in Jokkmokk. His presence meant more help with practical planning and execution as the trial period neared. It also meant that we N4C-LTU developers worked directly with someone who had detailed knowledge of the Padjelanta, Sarek and Jokkmokk areas. Kuoljok had ear-

⁵Avri Doria was one of the researchers in N4C and SNC, during which time she co-invented the PRePHET DTN protocol
lier been employed as a system administrator at SameNet, based in Jokkmokk, so his IT knowledge was well suited for his tasks as an N4C field test manager. It was almost immediately apparent that Kuoljok would be a crucial factor in the success of our DTN trials. He often acted as a translator between us, the N4C researchers, and the local community, because he had a basic knowledge of DTN and could explain this to the Sami end-users in a way that we could not. Kuoljok also grasped the potential of the DTN for the community, and was able to communicate this. His presence meant that we had more time to concentrate on the development and maintenance of the application services.

**Software and hardware iterations**

In September 2008, after the first two-week trial, we began collaborative work on applications for the DTN. The main focus for me was the development of the Web caching application beginning with the technical start-up meeting in Slovenia (Zlata Bodnar et al., 2008). Folly Consulting and LTU identified and worked on the theoretical scenarios that would be “nice to have”. Because of the results and the potential we saw in the first trial, these scenarios seemed reasonable and attainable. I presented scenarios for web caching during a conference proceeding, where we also had some Q&A about technical solutions for the system. In brief, these scenarios were as shown below, though greater detail is presented in the description of the DTN application prototypes chapter on page 67.

- Provider pushed regular
- Ad hoc search request
- Ad hoc site request
- User pulled regular
- Provider pushed event driven
- User pulled event driven

Not only would these improvements provide for a documentation structure for WP3 delivery reports from the project, they also split possibilities in different categories, so that WP3 developers like myself could envision what was possible in a web caching system in DTN and, in theory, how to accomplish it. The scenarios set the baseline for the final prototypes, presented in “D3.1 functional specification” report (Grøttum et al., 2009), of applications developed by LTU and Folly. LTU had primary responsibility for the practical development, configuration, planning, and testing of Web services. “Nice to have” in this case meant ‘everything technically possible’. There was not time, of course, to do everything: I was required to deliver a functioning service to a highly sensitive research platform. During the 2008 summer trial, if something went wrong at the other end of the network, we were stuck in the trial area with limited potential to analyze and fix errors. An overly complex system might put stress on or even break other links in the DTN chain. It could mean that the service itself failed, if quantity was valued over quality. Significantly, over-complexity might also affect how the end-users/participants viewed the service. Naturally, a balance between “nice to have” and a stable, functional web service was the goal. The input for the development iteration of the N4C-LTU
services came, as hinted above, from field trials 2008 experiences, technical meetings among project partners, and end-user feedback from locals and hikers.

I decided to design the web cache application to be more service oriented, to improve its usability for end-users. I reworked the design of the page where the user sees the cached web page links, so that it now offered a simple portal wherein most of our DTN services were collected. This portal would be the end user’s main entry point to the services and consisted of a main page with a welcoming text and background to the N4C project and the field trials, websites (in the cache), Podcasts, local temperatures, and a description of how to use the NSIM.

As we began to learn the true nature of “ubiquitous” in DTN applications, “radical” changes were made to the overall service. Having myself been a kind of user during the first trial, I had noticed the application limitations because of the “infrastructure” wireless mode used at N4C-LTU, where all the end user computers connected to a single point or node (hotspot). Grasic and I wanted to get away from this limitation so that the DTN would function more autonomously. We had brought this up in project meetings and begun working on this between the first and second weeks. Our solution was to move the services, the PRoPHET protocol, and the application to end user computers to create an ad hoc network. We also learned from the last test that we needed newer and more up-to-date hardware, that would be more power efficient, faster, smaller, cheaper, and lighter. During our lab preparation, Jan Lindblom brought a new type of laptop: an ASUS EEE netbook. We decided that this little machine was the logical way to proceed. Intel, one of the project partners, had at that time developed Atom, a low-powered, low-cost CPU. The decision to go with such computers was also influenced by our observations of the end-users’ interactions with the services. This better and easier hardware interface and its marriage with the DTN software and applications meant that the usability experience improved drastically. We had to be sure that the operating system would fulfill everything needed of the DTN, and at the same time make it usable and uncomplicated for the ordinary user. Anyone with experience of a PC should be able to use the service with little or no help from the developers.

As indicated in figure 3.8, our network had become larger and more ubiquitous. For the second year trial we placed a new node and client computer (netbook) in a village called Stadda. I had developed a new podcast application service to complement the existing web caching application. This podcast service was also installed to end-user computers and our Internet-DTN gateway at LTU. As with web caching, I chose a straightforward, not overly complex, service. However, the work on applications was not the only technical preparation. Installing the operating system (Ubuntu 9.04) on the netbook nodes, and assisting Grasic with the installation of the PRoPHET software were time-consuming tasks. We had, for the second trial, changed some wireless drivers for better network functionality, another example of an unexpected problem that might be quite common but difficult to predict without extended testing.
Figure 3.8: This illustration shows the network topology for the second summer trial. In pictures there are our new user computers Asus EEE. New wind turbine power source, new VPN connection and new border node. Illustration created by Samo Grasic (Grasic and Näslund, 2009).
The trial period

The second year’s trial occurred in three phases. The first was mostly technical. I helped Grasic prepare the border node (illustrated in figure 3.8), installed by him during the first phase beginning in July in Ritsem. I installed OpenVPN software for the virtual tunnel link between Ritsem and the LTU office where we had placed our gateway server (illustrated as “lime”) which communicated with the Ritsem gateway border node. Inexperienced in setting up virtual private networks, I had to learn more about the technology but, after some trial and error, the VPN functioned stably. This meant that we had a secure traffic route between Ritsem and LTU and, more importantly, we had extended the DTN to include the LTU Internet gateway, with an interface that belonged to DTN (same subnetwork address) so that the office gateway could be “seen” as a DTN node. Simultaneously, it had an Internet interface for retrieving and sending application data for e-mail, web cache, and podcast.

After Grasic came back from installing the gateway border in Ritsem, we successfully monitored the VPN link. At first we had good contact with the Ritsem gateway but a few days before the start of the trial phase the contact with the Ritsem gateway disappeared. At first I thought that something had gone wrong with the VPN I had installed, that its certificates might have inexplicably expired. I connected to the LTU gateway using another computer but could not find any errors. We had to physically go to Ritsem to discover what had happened. As described in the Milestone report:

After coming back to Ritsem we found out the reason why we lost the VPN connection with border node. First we were very glad that there was nothing wrong with the border node itself. Later on we found out that the Swedish IceNet provider turned off the NMT mast in Ritsem several days before we arrived without any notification. This fact brought us serious problems with basic Internet connectivity in Ritsem (Grasic and Näslund, 2009, p. 14).

This was typical of the disasters that one hoped would never happen, especially during the trials. We relied on the NMT connection for most of the test since it provided the only link to the LTU Internet gateway. Since we already were at Ritsem, we went ahead to the trial area to do some initial setup.

We flew to Staloluokta and worked for a couple of days installing the core functionality of the network. We mounted the power sources, the wind generator and solar panels, all hooked up to car batteries. This procedure is quite time-consuming and there is a lot of heavy lifting, especially when carrying the batteries (which were in the static node boxes). The solar panels had to be placed strategically, so that we maximized the energy the sun gave us during the day. We also walked the nine kilometers with some equipment to Stadda village to see Andreas, a cabin keeper who was helping us with some long-term testing. We knew him from LTU as a technically skilled person and long-term end user/participant. When we arrived, I set up the computer Andreas would use during the summer trial as Grasic set up the static node that would forward DTN traffic to the helicopters. Luckily, we got the helicopter to drop off the heavy static node before we arrived, so we did not have to carry it. (Such simple favours from the pilots meant a lot).
While trying out the podcast application, I noticed an interesting error while trying to play back an audio file: there were no codecs installed on the user node of the netbook provided to Andreas. An mp3 decoder was necessary to listen to the file. The error could not be solved because we were in a remote village with no way of obtaining the decoder. I was puzzled, as the player had worked flawlessly in the lab. We later discovered that the player automatically downloads the mp3 decoder in the background, when an mp3 file is being opened in the Ubuntu 9.04 operating system. This obviously requires an Internet connection. Luckily, I had a chance to solve this while at LTU after these first-trial setup days. Since we did not have an Internet connection available, there was no external traffic from and to the Internet. However, the NSIM application was useful because users could send DTN messages between the nodes we had brought. We also gave one netbook to the cabin keeper in Staloluokta to try this message application.

Once back at LTU we got a subscription to a 3G network with a USB modem, which we planned to install to replace the Ritsem NMT modem which no longer functioned. We were visited by representatives from TCD, Intel, and Folly Consulting, the N4C partners who were testing their own DTN implementations along with that of N4C. During our trial period, an ExtremeComm meeting took place in Staloluokta. This international meeting for advanced IT communication researchers was hosted by Anders Lindgren, and provided an excellent opportunity to meet developers and researchers from all over the world working with similar technology.

Figure 3.9: LTU office in Luleå, where we are preparing more equipment for the next trial phase, together with Folly Consulting, TCD and Intel. Photo by Samo Grasic.

After a few days of work at LTU, we went to Ritsem, staying a night at the hiker hostel to help our project partners with their equipment. Elwyn Davies of Folly Consulting and I did a last-minute configuration of several of his Nokia handheld (Figure 3.10) devices.
which were to be given to the ExtremeComm hikers walking from Ritsem to Staloluokta. This meant that they could test Elwyn’s implementation of DTN e-mail, as walking “data mules”. We had installed the 3G modem but only a 2G/Edge connection was established at the Ritsem helicopter base, due to limited coverage. This meant that data transfer from our LTU gateway was slower, but I ascertained that it was just fast enough for sending limited Web cache and podcast files.

Figure 3.10: Configuring the Nokia N810 smartphone devices with a small script. Photo by John Näslund.

At Staloluokta, data traffic worked as we had hoped. A full work day in the village starts in the morning and ends at night. Just taking care of food and personal hygiene is a time-consuming process. We usually followed a path down to a sauna during some evenings, where we also took a bath in the lake. There were no other ways to keep clean. We had brought our own food, and having three meals per day was of course key to our staying healthy and upbeat. Since we spent most of the days working outside, we had time to interact with the end-user participants. I was pleased to see the newer, more ubiquitous system worked well, though there always were bugs to sort out and little things to improve. Some of these were accomplished during the trials but access to electrical power was always an issue, imposing limitations to our work. For instance, it was often a bad idea to write notes on the laptop since the power could be used for more important tasks such as charging end-user computers or other devices.
3.3.4 Test outcomes year two

With respect to the application services, I noticed that the web caching was working more or less successfully. There were still glitches. Web pages did not want to be updated every day, and it was not clear if the problem was in the wwwwaffle cache engine at LTU or in the website itself. This was something I could not work on until after the trial. Another problem was the flawed podcasting: end-users could not play back audio, which was solved by my bringing the needed MP3 codecs to the trial area. I also noticed that podcasts were not presented for the end user in the best way. These design flaws were noted for future resolution. More than 120 podcasts were sent to the trial area, which can be considered a success. The use of instant messaging and email services became a new hobby for some end-users. From this we saw the potential for the NSIM software to be used for personal e-mail addresses. Grasic’s improved node design and better antennas also positively affected the use of services, providing better coverage and uptime. The wind power generator gave much-needed power for the end user netbooks. We had successfully taught the cabin keepers how to use these netbooks and how to charge them without much difficulty. When we were away between test phases the cabin keepers used the DTN services, as we had hoped. The capabilities had become more ubiquitous because of what we had learned from the first-year tests. While there were still many little design flaws in both software and hardware to address, it was now proved that the network’s technical system could work autonomously for a couple of weeks and the human-computer interaction aspect of DTN services was also functional⁶.

Figure 3.11: The last day of trial in Staloluokta, the meeting and dinner with ExtremeComm hikers who were happy to test and discuss our DTN services we provided.

---

⁶A technical milestone report for N4C was submitted with our results (Grasic and Näslund, 2009).
3.3.5 Year three

The preparations for the third year’s trial started soon after the second year trial ended. N4C project meetings were held in Cambria, Portugal; in Tromsø, Norway; and in Dublin, Ireland. All knowledge produced within the N4C project would be put to the test in the largest-scale DTN end user trial. We at LTU would combine our efforts with TCD, Folly Consulting, Norut, and Tannak regarding practical issues. We started the preparations knowing that the second year trial established that we had successfully set up and maintained improved DTN hardware and application services. We had also shown that extended end-user trials had worked with satisfactory results: users had found benefit in testing the services. We had left the DTN running autonomously, and left netbooks with local users. Our aim was to make all this better, and with closer involvement of end-user participants.

Our second-year experience provided the input for the third-year tests. Most aspects of the DTN project were increasing in scale and complexity. In addition, troublesome issues cropped up regarding malfunctioning hardware, loss of Internet connections, methodological errors in preparation (such as missing mp3 codecs), wireless problems, or bad weather conditions that meant limited power for equipment. Additionally, we cooperated with project partners, assisting each other on individual projects. Since we were to work as a team, it was necessary for us to work well together for the third trial. Lastly, we knew much more about limitations in equipment, and more about what we could expect from end-user participants.

Meetings prepared us for the start, deployment, and field work of the third-year N4C trial. The LTU office was a preparation and collection hub for some of the N4C partners during the preparation stages, and the large test was to be at Staloluokta. Grasic and I often had to handle the practical and logistical work since our project partners sent us equipment to be mounted and prepared at LTU. There was a measure of security in Staloluokta being the site of this trial: the place was known to us and we already had relationships with the local Sami villagers and cabin keepers. Most aspects of our DTN and equipment would reach the next-step iteration and would increase in scale, so naturally work on this started soon after second-year trials. Aside from planning the tests and developing iterations, we organized the practical matters of food, logistics, clothing, etc. To keep track of these practical issues, we produced a risk assessment in during a workshop in Tromsø, Norway with the other N4C partners, for present and future use by participating project partners.

End-user scale aim

The major change from the second-year trial was that the end-users were in greater focus. I wanted to make use of the previously acquired knowledge to make web services available and functional for a longer period for families living in Staloluokta. Hikers and tourists were a secondary focus in this trial, since the available resources could not cover all user scenarios. There was also a difference in response and user feedback: hikers and tourists often stayed only one night in Staloluokta, while a family stayed several weeks. Their needs and views about DTN services were different. I had also observed that hikers were
not as keen as residents on having Internet access in nature. Hikers seemed to think DTN was a “nice-to-have” technology for brief e-mails to relatives and friends, while families staying in the mountains for longer periods might have a stronger need for various communication with the outside world.

In this trial we planned to install upgraded hardware and software to DTN-service laptops with acceptable usability, and give these laptops to local families. We had contacted four families, a total of about 15 persons, with the help of our field test manager, Fritz-Åke. We prepared four netbooks, with the latest iterations. This might sound easy, since we had already done two major summer tests, but we had learned that even small changes had the potential for a seriously negative effect when actually implementing the changes. Each iteration had to be carefully tested, with previous field experience in mind.
Individual progress

I knew that I could not become an expert in every technical detail surrounding my work. Instead, I searched for a way that would guide me in my research. In the third year I combined all previous knowledge gained from the work in years one and two to set up a better, larger, more ubiquitous, and pervasive system. This was a focus for the project as a whole and for me as an individual developer and researcher.

To indulge my desire to become more conversant with the theoretical aspects of my work, before the third summer trial I took a course on Participatory Research, a branch of Interactive Research. This course was the foundation of the theoretical material directly connected to my dissertation. It was useful when describing the work I did in N4C and how and why I did it, because the N4C project was a user-focused project where I worked and interacted directly with users. In this course I did a presentation discussing...
my Interactive Research/design perspective, my work with end-users, and the practical issues and problems of the field.

Software and hardware iterations
I had some ideas as to how to improve the overall usability in the third-year application prototypes. I considered that anyone should be able to use the laptop with the DTN services. I had discussed the design of the end user node (the netbook) with Fritz-Åke during the development of a new web portal where all the services would be collected. The portal was presented in Swedish and English and users could choose which language they preferred. I optimized the web cache application, but did not change the software configuration drastically so as not to risk harming something that was working well. However, some errors, such as some websites not automatically updating to the cache and various issues with the gateway and wwwoffle scripts, needed attention after the second trial. The application needed to be robust in order to provide the overall Web cache usability in the third trial.

The Web cache “User Site Request”7 was a new scenario I had developed and implemented for the this third trial. It was not yet fully tested, but was launched as a prototype during the third-year field test, because it presented a common functionality request that local end-users have. So, I decided to test this use scenario as a side experimental Web cache function. Another important “ubiquitous” improvement I made was in the list of cached websites, where I implemented the Google news “function” in Swedish and English which brought in a selection of Swedish newspaper articles. This choice made it easier to cache more news in a smaller-sized file, thus providing end-users with various news even though our Internet connection to Ritsem was limited.

The podcast application service was updated to handle requests from users, but not ad hoc requests. I created a list of podcast subscriptions, and let end-users choose from that list. This service was mostly aimed at local families, and subscriptions were chosen according to their interests, providing that the size was small enough for the DTN to handle. Further improvement in human-computer interaction usability was made by ensuring automatic startup of the Ubuntu operating system with PRoPHET, the web browser, and the NSIM software. The end user could also see the date of the last updates to websites and podcasts, which I changed to be ordered and displayed by date.

---

7If a user enters the URL of any website that is not cached, the URL is forwarded to the DTN gateway for processing, and thus would be available in the next web cache update.
The trials in Staloluokta

From 3 July until 25 August, N4C-LTU participated in the 2010 summer trial. The first couple of days we installed power sources and nodes, and tested the DTN’s core functionality. This last was crucial, because we were installing the foundation for end-user services. If everything was not done correctly, those services would be at stake, because any loss of power or wireless connection would ultimately result in less time for actual application usage for the end-user participants.

Figure 3.13: Installation of the most advanced N4C-LTU node, with a Rutland 913 wind power generator, a new high-gain antenna, a solar panel, and a customized mounting pole. On the ground we placed Grasic’s latest iteration of a “static node” with a sturdy box and batteries. Photo by Samo Grasic.

I filled three different roles during this field trial:

- Engineer, helping keep the DTN up and running, moving around nodes when needed, maintaining and fixing small problems with the application services, logging in to nodes to read logs, ping nodes, etc.

- Interactive Researcher, interacting with local families and hikers, explaining the project and answering questions. This often happened ad hoc and at other times by appointment. People are scattered through the area and normally cannot be contacted other than by physical visits.

- Hiker, fetching fresh water, preparing food, eating, staying clean (sauna and bath), etc. These simple but important tasks took up a lot of time.
Throughout the trial period in Staloluokta, we often had to walk between the participating families’ houses to organize and meetings or update something in the software. Once in a while, participants came to our cabin to report a problem or ask questions about the DTN services. We asked them to use the netbooks as much as possible, as we were hoping to see traffic from all services, web, mail, NSIM, and podcasts, as well as to charge the computers when needed, so that the trialled services would really be put to the test. Some of the families had their own solar panels with enough capacity to charge the netbooks. Otherwise, the end user had only to press a button and the software would automatically load.

We held focus interviews with each family near the end of the trial period, after brief guidance and instructions from Prof. Ewa Gunnarsson\(^8\) who visited us during the trial. Many people had already seen us working in Staloluokta during the 2008 and 2009 trials, so they were not unfamiliar with the DTN concept. However, they had participated in the testing for a longer period and became critical for the assessment of the last trial.

\(^{8}\)Ewa was at the time professor of Work Sciences, gender and technology unit at LTU
3.3.6 Test outcomes year three

For the web cache application I obtained satisfactory results mostly because of the feedback given by the end-user participants. The local families had used their netbooks daily and browsed websites. Some of the more technically skilled users would alert me when they saw something not working as it should, for example when the zero byte web cache “tarball” arrived (see below). Overall, the participants saw real potential in the websites available to them, and discussed this during the focus group interviews and other interactions.

At least one end user tried the “ad-hoc website request”. They reported that it worked successfully, but that it also caused negative bi-effects, including the zero byte web cache “tarball error” (when the server sent an empty file due to an error in the web cache script or wwwaffle application). This was later fixed by shutting down this particular function. If I had known in advance that this error could occur, I would probably not have tested the request prototype at that stage. On the other hand, it was one of the more obvious things that end-users want, to be able to surf instantly to any site (even if the website would not be available in the cache until the next day). However, the web request function had a round-trip time of about two days, so it was not the perfect testing scenario when we had a limited trial schedule.

At the beginning of the trial, the end-user families were told that interviews would be conducted with them along the lines of a moderated discussion, letting them express their views on our work and DTN services in general. After the 2010 trial, they had a better understanding of DTN and had had time to think about the potential of the technology for them. A joint effort within the project occurred after the final summer trial to present the overall N4C results. The Delivery report 8.4 - Test Results (Romanowski et al., 2011) includes test result summaries from the N4C partners that participated in the trial.
4 | DTN Application Prototypes

This chapter contains information about the DTN application prototypes that our LTU team developed during the N4C project. It contains the following parts:

- Introduction to applications
- Description of the applications
- Evaluation of Feedback

4.1 Introduction to the DTN applications and GUI

The following subsections present the Delay Tolerant Network applications that my partners and I at LTU (N4C-LTU team), developed during 2008-2010. These prototypes are part of the final technical results of the N4C project. Romanowski et al. (2011); N4C (2011b) concludes some general facts about the usage of the services:

- The final applications prototypes were provided during 53 days.
- 6107 bundles were sent from the N4C-LTU DTN-connected computers.
- Bundle delivery rate was measured to 89%.
- Average delivery delay was two days.
- 1359 e-mails, 328 NSIM messages, and 167 SMS was sent from end-users.

Due to privacy issues, we did not monitor the usage of every individual application that the end-users tested during the 53 days. Also, such monitoring would demand more focus from us developers. Instead we used other usability evaluation methods. In this thesis the applications are also subjected to theory, methodology, analysis and discussions from a scientific perspective, taking the research question as the starting point. All of our team’s applications are included as they share a common ground, the DTN implementation, and as we in N4C-LTU often assisted each other in the development of the applications and testing them in field. The unique character of all these applications is that they are built to function where the normal Internet equivalents would fail due to delay or disruption, and this was therefore the primary focus of the work behind the thesis. The applications build upon the ideas and suggestions that went into the N4C - Description of Work document (N4C, 2008) to which foregoing SNC project contributed with practical examples and experience.
DTN Application | Responsible
---|---
Web cache | John Näslund
Podcast | John Näslund
Not so instant messenger (NSIM) | Samo Grasic
Localized applications | Samo Grasic
Graphical User Interface (GUI) | John Näslund

Table 4.1: DTN Applications developed by N4C-LTU for trials in 2008-2010

The applications (Table 4.1) went through three main development iterations. The first iteration was developed for the initial N4C trial in 2008, when we adapted left-over “source code” to suit our needs for that test. The second iteration occurred between the first and second trials, when we had evaluated the first trial. Finally the third, largest iteration started after the second evaluation, between second and third trial. It is important to remember that the applications rely on the DTN implementation in order to work, so not all of our time was spent on developing applications. Application services means that there is a whole chain of things that must function, i.e application software and computer hardware in a real environment. This, and the following section contains a basic introduction to as well as a more detailed overview of each application in its final prototype state. More in-depth technical information and setup examples for each application can be found in appendix A on page 106.

4.1.1 Web Cache

A Web cache service for DTN provides a potential end-user with basic Web access. Although the technology behind it works differently, the user uses a Web browser just as on a traditional Internet-connected computer. The service was one of the priority DTN application services that our team at LTU worked with, because it is reasonable to implement and provide - simply because the World Wide Web is currently one of the most used and most important sources of information. Technically, it is reasonable because Web pages often consist of published static material that is possible to copy and transfer from the Internet to a local hard drive (cache), which is generally what Web-caching is about - that is, data storage. The Web pages are retrieved by a server with an Internet connection. From there they are transferred to the DTN-enabled computers where the person using the service can browse the selected websites. We selected the Web sites we wanted to provide based on user feedback and suggestions. Our solution was to have a static list of Web sites, each of which would be updated two times per day. That effectively meant that end-users normally got updated Web pages two times per day, such as weather and news from various Web sites.
4.1.2 Podcast

The podcast application (tested in 2009 and 2010) allows a person using our DTN services to receive and listen to broadcast audio (i.e., radio programs) from the Internet. It is an application that extends and complements our Web cache application, in the sense that it enables end-users to listen to news, weather, etc. instead of viewing it on their small laptop screen. It is also a complement to traditional radio, which is often seen an important source of information since it is ubiquitous and popular among older generations. In 2010, we developed and implemented an experimental “on-demand” function for the podcast service. Thus, in the final prototype trial the participants using the podcast service could “request” a podcast subscription from a list of selections, to be delivered in on-going daily updates.

4.1.3 Not So Instant Messenger (NSIM)

NSIM (Grasic, 2011) is an application that N4C-LTU made to provide the sending/receiving of SMS, local “Not so Instant” messages, and email. The reason for its development came from our field experience and our previous experience in the SNC project. This experience told us that such a service would be welcomed, especially among the local population. The NSIM application enabled local participants to login to an individual user account. If the user does not have an account, then the person can at least send email and SMS from a public account. Sending local messages within the region is intended particularly for local habitants who need to communicate directly, peer-to-peer, i.e., within the DTN “realm”. E-mail is written and sent in the same way as in a normal e-mail application, and the recipient may be any normal e-mail address on the Internet. It is also possible for a reply to be sent back if the sender has a registered NSIM account. The key factor in the NSIM application is that it incorporates important communication technologies, such as e-mail, into our DTN services.

4.1.4 Localized applications

In 2010, at the final prototype trials, N4C-LTU provided local weather forecasts from a portable weather station, for hikers and local habitants to view. This service was made possible by weather sensors and cameras that we used in Swedish N4C trials (also in Slovenian WP8 trials). Web cameras placed in the villages of Staloluokta and Ritsem automatically took pictures, which were stored on the DTN-enabled computers. These services were then accessible from our DTN Web portal (graphical user interface) on each end-user device. Having a picture and weather data including forecast can be very useful for hikers and locals, since such data sources are always limited in the Arctic mountains. This is of course unique to the area in which DTN exists, and the localized DTN services reflect the many ways in which DTN can be adapted to its operating environment.

4.1.5 GUI and HCI

The user experience in our DTN could be explained in these terms: it is everything the end-user sees and interacts with, using a device such as a laptop, a tablet, or a smartphone. The user interface is the custom interactive software, created by us in the
development team, that presents the non-hardware pieces to the end-user. The GUI is found on each end-user device. The hardware (device and peripherals) is what an end-user had to learn to use if our DTN service was to be thoroughly tested. This hardware was typically the laptop or the smaller PDA, including chargers.

Graphical User Interface (GUI)
Our GUI is a common Web interface, installed on each end-user device, that binds the above-mentioned DTN services together. The reason for having a Web GUI portal is to provide a key part of the overall usability of the DTN applications. N4C-LTU aimed to make sure that the user would be able to locate and use the services quickly and efficiently. The portal is installed locally on the DTN-enabled laptop and provides ubiquitous access to Web cache, podcast, NSIM and localized application services.

Human-Computer Interaction (HCI)
HCI refers here to how people interact with the totality of the services, not just the software itself. Our goal is to create something sustainable that a normal person could use on a daily basis, without it being too technical a task. There are a number of design steps that we have taken for the prototypes in order to improve our end-user participants’ experience. These iterative design improvements are explained in detail in the following section.

4.2 Web caching Application Prototype
Our Web-caching prototype is a service developed for the remote trial test bed of the N4C project. It provides access to “delayed” Web pages for remote inhabitants and tourists in the Arctic region. Following is a brief overview of what N4C-LTU’s Web cache concept means to us.

4.2.1 The concept of Web caching and the development process
Since Web pages are made up of fairly static content, even though many do include dynamic content, it is relatively easy to retrieve Web pages from the Internet, and save them to a hard drive. This is what Web-caching is really about: storing Web content for later use. In 2008, web caching development was started, using older SNC equipment and code. This was necessary to get a reasonably good start which made the service usable in the 2008 kick start summer trial. The trial included users (mostly tourists) in the Arctic region camps Staloluokta and Saltoluokta. Field experience and user involvement generated input to the upcoming work in relation to functional requirements and actual development of the service.

In the beginning of 2009 a limited winter trial took place in Jokkmokk. Although the test was focused mostly on hardware in cold environments, it gave a hint of how stably the software was running on the machines in these temperatures. Between the first and second N4C summer trials, the web caching development progressed. The trial focused
on the ubiquitous and pervasive aspect, so the upgraded service was implemented and
configured on all user computer DTN nodes, with a improved user interface and a selected
number of pages which depended on previous user requests, test experience, page content
and bandwidth limits.

The final prototype of web caching application, was tested in the last, longest and biggest
N4C summer trial. This trial focused more on end-user participants and valuable feedback
than previous N4C trials. Web caching was now accompanied with a new and updated
GUI with English and Swedish web sites, prototype of the web request functionality
(Grøttum et al., 2009), and finally improvements to the pervasiveness of the web cache
e.g. by including time stamps, and collected Google news. Extended testing was done
with locals in the CCR village Staloluokta, to whom we gave computers to use during
their several weeks stay. Although the focus in this trial were the local users, testing
was even done with tourists, with a computer installed in the tourist cabin. For several
weeks, these end-users put the web caching application prototype to extended real life
testing, which brought daily information and news without the need for us developers to
be present with the end-users. All that was needed was a quick how-to introduction at
the start, for each family. That included instructions on how to operate the software and
how to re-charge the computer.

4.2.2 Components of Web Caching

This section contains an overview of the Web-caching components that exist on either
the server or the end-user laptops. These are the software pieces that make up for the
complete Web cache application. For a more details, see appendix A on page 106.

On the server
For the server, our DTN development team at LTU installed an open source caching
application. In our solution, the wwwaffle application was installed (Bishop, 2013). This
application had one main task: to connect to various Web sites on the Internet and down-
load them to the hard drive in our server. The application was configured and tailored
to our needs by applying command line switches and variables during execution of the
application, and by editing its configuration file.

We also created a script for controlling the caching application. The script’s functions
is to start/stop the caching process, i.e., the downloading of Web content, and to send
that content to the DTN so that the client computers can receive the content at a later
stage in the network route. The second task of this script is to read and process incoming
variable URL requests from end-users. The script also contains the static list of Web
sites that was to be downloaded and cached.
On the client
The client laptop contains two pieces that help present cached Web content to the end-user. We implemented a script that regularly checks for new content coming through our DTN. The script notes down the time and date, and makes the content available to the Web GUI, which is the second piece and is what the end-user interacts with. There, the user can view available pages and see the latest update time for all Web content.

4.2.3 Web Caching Use-case Models
Unified Modeling Language (UML) activity diagrams are abstract methods illustrating how the scenarios for our Web-caching solution work in real usage. These diagrams are meant to show the different ways end-users are able to receive web content through the DTN. Based on the diagrams originally made for N4C project (Grøttum et al., 2009), the activity diagrams are updated to illustrate the real usage scenario of the N4C-LTU Web-caching application prototype, during the last trial in 2010.

The UML activity diagram “Network-pushed Regular” (Figure 4.1) is the most used implementation in the N4C-LTU Web-caching application. The diagram basically describes, on a general level, how data flows through the DTN. During the 2010 summer trial this type of Web-caching was used on a daily basis, which meant that selected Web sites were updated to the DTN client computers in the trial area. The gateway is the starting point of regular pushed material. A script (see section 4.2.2) is configured to download updates and then send them to the clients’ cache via DTN. Important variables in this script that we, the developers, could change and experiment with between trials were:

- The list of chosen Web sites can be edited.
- The time for pushing out content to DTN can be modified.
- The number of times it should send out content can be modified.

1The term “CCR” used in the diagrams means Communication Challenged Realm/Region
High level scenario: Network-pushed Regular

1. Gateway regularly reads a list of web sites and downloads web data to its cache (www.coffle)
2. Gateway replicates its cache after each update to a package and sends the package to DTN
3. Package arrives and gets automatically unpacked by user cache
4. Notification (in GUI) of updated web pages available to user when GUI is accessed

Figure 4.1: Web Cache Network-Pushed Regular
The UML diagram Ad Hoc Site Request (Figure 4.2) explains, on a high level, how the Web request function was used in the 2010 summer trial. The initiator in this case is the DTN end-user who tries to reach any Web page that does not exist in the local cache application on the computer. The Web page request is, in this case, intercepted by the cache application and sent to our DTN gateway for processing by the Internet-connected cache server script.

![High level scenario: Ad Hoc Site Request](image)

**Explanation:**

1. User tries to reach a URL that doesn't exist in the user's cache
2. The request handler catches the request and relays it to the gateway over DTN
3. Gateway processes the requested URL
4. Gateway creates a package of the returned data and sends it to CCR
5. The package finally reaches the user computer where it gets unpacked to the user cache
6. Web page is served to the user web browser when requested again

Figure 4.2: Web cache Ad Hoc Site Request

### 4.3 Podcasting Application Prototype

Podcasting is a prototype application developed by N4C-LTU for DTN use. The aim in the trials was to provide end-users of our DTN service with the “ubiquitous radio” experience. This application was tested in the trial test bed over two summers, and our hope was that audio could potentially broaden the utility of DTN from an end-user perspective.

For end-users living in a remote region, radio is often a key source of news and weather
information. Weather and local news are things that can have an impact on people’s lives and movement in the area. However, radio listeners are often subject to radio shadow in the “extreme” areas. In some scenarios radio is not even an option for the kind of media mentioned above. Generally, radio is an important source of information because of its property of being highly ubiquitous. Podcasting, a term that was invented with the increasing popularity of Apple iPods and other media players, is a way to push digital audio files (e.g MP3) to Internet-connected applications. It is also known as “personal on-demand broadcast”. Podcasts does not require tuning in to an broadcast, since the audio clip is instead downloaded onto the device with the help of an RSS feed.

After completing the first N4C trial in 2008, feedback and requests was extracted from impressions from field work and reports. This information led to ideas, where the Podcast application was chosen. The point of the podcast application is to expand the DTN service concept. It provides end-users with audio material an using podcast is relatively easy on the network, keeping in mind the slow 2G/EDGE connection that was the bottleneck in many cases the N4C trials.

The first prototype was used in the 2009 summer trial along with the other DTN applications. The prototype was based on the web cache one-way “provider pushed” scenario principle. The first prototype inner workings was comparable to the web cache application, in a technical sense. Providers (Swedish Radio in our case) would “push” out podcasts to our LTU server, through the gPodder application (Perl, 2013). Our DTN podcast script would then set a date stamp on the filename and move the file to a podcast application folder of Samo’s PRoPHET. The file would then be in the custody of PROPHET which would deliver the files to remote DTN nodes. The podcast client script then created a link for the web GUI, so that the end-user could access the audio file. One suitable podcast radio that we used was the Sapmi news (Samiska Nyheter) from Swedish radio, since the content itself was perfectly related to the environment of our trial base, and of natural interest to the local end-users. This podcast also had a reasonable file size that allowed podcast to be sent over DTN without too much stress on the network. Large file sizes could present a problem, especially in the slow Internet connection in Ritsem and wireless transfers to and from the helicopters. The trial with podcast generated interest from testers in the field, but also from the local community, so we decided to develop DTN podcasting further.

For the final N4C summer trial Podcast application was made more user friendly by improving the client Web GUI to handle incoming files better, by sorting them by date. The experimental request functionality was also implemented, giving users opportunity to request a podcast feed. For the prototype, we needed to limit the flow of podcasts coming into the network, mainly because of how podcast feeds are requested. In the client there is a static list of feeds that one can choose from. Another reason for that is that the DTN network only could handle smaller files, mainly due to syncing with the helicopter node at the Ritsem base (see Application Network Topology), so we had to set a limit to how many feeds that we were subscribing to. However, the usage of podcasting showed us that it was possible to send relatively large files on a daily basis. And while talking to the users, we understood that the service was tried out, including successfully

75
requesting and receiving podcasts. The following is a list of N4C-LTU’s aims with DTN podcasting:

- to expand the use of DTN to deliver services making not only text and/or images available, but also audio, in the packaging of podcasts easily available on end-user computers.

- to enable radio broadcasts (as podcasts) to be sent using DTN with push and request techniques to communication-challenged regions. In practice this would mean that a network administrator would add an RSS feed from a provider of a podcast. The audio file is transferred over DTN to end-users. It could also mean that an end-user initiates a request to subscribe to a podcast, and this will be found in a static or dynamic list available in the user’s Web GUI. Podcasting should be used as a complement to radio, but not as a single solution for everyone.

- to use open-source, free software - implemented similarly to the Web-caching application, in order to create room for ideas and for possibly expanding the application in the future. There are already many useful applications that handle podcasts in a way that is helpful to the server administrator. Thus, it should be relatively easy to implement such software to any DTN server/client environment.

4.3.1 Podcast Use-case Models

These activity diagrams give an overview of how our podcast application prototype worked in a real-life situation, during a 2010 trial. The two UML diagrams represent N4C-LTU’s real-life testing of the application in communication-challenged regions.

*Network-pushed Regular Podcast* (Figure 4.3) shows that the control point of the regular pushed podcast operation is the gateway, where podcast subscriptions can be added, removed or edited. Before the N4C trials, this was administered manually at the LTU gateway (2), so during the trial podcasts would periodically be sent to DTN computers in the field (the CCR). When DTN-enabled computers receive the podcast, Prophet stores them in the directory on the local hard drive (3). Afterwards, when the end-user refreshes or accesses the Web GUI, the podcast can be accessed through a link on the page (4).
4.3.2 Components of Podcast Application

The podcast application is made up of free software and scripts, similar in concept and function to those used in the Web cache application. This is an overview of the scripts and software pieces that our N4C-LTU team implemented on the gateway server or the DTN end-user computers. For a more detailed and in-depth overview, see A.

**On the server**

The heart of the podcast application is a program that looks for - and download audio files from the Internet. These files are called podcasts and they may be updated regularly by their provider. Internet links to podcast subscriptions are configured into the application and a download folder on the server is selected. When set, the application can automatically check for new podcasts, or be executed on command. For N4C’s purposes and for potential future development, our team selected the application according to the following criteria:

- Free and open source. It aligns with the N4C project’s general aim to be an “open”
Focus on usability, meaning that the application was easy to implement and use in our DTN podcast application or easy to use in other non-N4C DTN services in the future.

Supports various podcast feeds which potentially made it easy to add and manage end-user requests for N4C-LTU podcast application.

The second podcast implementation on the server that we did was a server script. The function of this script is to “manually” control the podcast downloader. The tasks the custom script performed were:

- Starting and stopping the podcast downloader.
- Adding dates to the filenames of the downloaded podcast audio files.
- Forwarding the files to the DTN.
- Checking for end-user requests.

We configured this to happen automatically once a day, during the trials.

**On the client**

A script was implemented on the end-user computers, with the task of presenting the podcast files to the user. Once the DTN had successfully transferred the files to the computer, the graphical user interface (GUI) would present them in a readable manner on the GUI prototype our team created (the Web portal). Once the user clicked on any of the podcasts, they would open in an audio player.

### 4.4 NSIM Application Prototype

The Not so instant messaging service (Grasic, 2011) was originally meant for developers to test the functionality of a setup between nodes in a DTN. A developer can choose a specific node from a list of known DTN nodes and send a text message, and the receiving end can read it when it arrives. During summer trial in 2008, NSIM was seen as a purely technically focused tool but nevertheless we knew it might be that it could be useful as a foundation for a messaging system even for other cases in the future as well. It is closely tied to the addressing scheme used for DTN nodes as compared to e-mail that uses a more general addressing scheme. NSIM was therefore restricted to communication within a DTN area.

In the 2009 summer test, the Not so Instant messaging service technically remained the same as in previous summer test. Users were able to send short messages from one node to another within the DTN domain. Testing the concept of broadcast feature on the DTN network allowed users to send messages to all the nodes within the domain at once. It turned out that this can be a really useful feature for sending announcements, but could potentially allow spam attacks. During the trial N4C-LTU recognized the potential in the NSIM application. Impressions and experience from then field tests with users gave a lot of ideas for a new version.

In the last summer test, the improved NSIM application allowed users to sign in with their own username and password. When a new user is created for NSIM, the administrator needs to assign one or more DTN nodes that the user will be reading messages on. This feature is important for the users that are mobile within a DTN region and do not use the same machine all the time. At the same time, any user can send out a message from any machine from their own user name account. Administration of NSIM users accounts is possible from any DTN node on the field or the gateway connected to the Internet. This was crucial in the last N4C summer test when most of the users were added to the NSIM system from the field location. A list of available NSIM recipients is always available to users and offered in a drop-down list when sending out a NSIM message. Features that commonly exist in regular e-mail services such as “reply” and “attachments” are also available in the latest version of the NSIM application. NSIM
messages can also be delivered peer-to-peer, without involving the LTU gateway computer. This makes the NSIM messaging service faster and reliable than e-mails for which the round-trip time was at least a day in our scenario. The NSIM service was widely used for in-village communication by our test crew for quick user support (as alternative to walking to the other side of the village camp). Instead of traditional use of PMR walkie talkies that does not provide any privacy, reindeer herders used NSIM during the last N4C summer test when they were organizing reindeer calf marking event.

A popular feature in the final version of the NSIM service was an option for sending out short text messages (SMS) to mobile phones outside the DTN area. This feature was widely used among hikers and tourist for letting their families know that they had arrived safely. The fact that most of the hikers are carrying their mobile phones while they are hiking (although there is no cellphone coverage on most of the test field area) helped people to retrieve recipients’ phone number (which was not the case with e-mail addresses). Another new feature was an option to send and receive e-mails. To simplify e-mail user management, all the e-mails were sent out and received in the DTN area through one e-mail address. When email was sent out, gateway application attached an id from the user to the subject, that was later used for retrieving the destination address when someone replied to an e-mail from the field. An unforeseen problem occurred when and if someone modified the “subject” of the reply e-mail. Such a case had to be solved manually by us developers.

In summary, The NSIM (Not So Instant Messaging) application was first developed as a very simple communication application that enabled people involved in SNC project deployment to communicate within the DTN test field. The drawback of this effective SNC application was that it only allowed addressing of network nodes and not of users. Hence, some users shared certain nodes, which was a drawback from a privacy point of view, but we made sure end-users knew this. However, the concept and need for an NSIM application in the field, was proven, and the service was further developed to its final stage in 2010. The final prototype allowed long term end-users to register an account in the NSIM application, which made it possible to send messages to other registered users.

4.4.1 Components of NSIM application

Within the N4C project, the NSIM application was redesigned and rewritten from scratch by Samo Grasic (N4C-LTU). To shorten the development time, part of the graphical interface and email libraries were “re-used” from open-sourced projects. To run the full NSIM service, two applications are needed:

- **NSIM** – A client application that is very similar to any email client and must be installed on every client machine. This application by itself is enough if the end-user only wants to send messages within the DTN region.

- **NSIMGW** – In order to allow the NSIM user to send and receive emails or send out SMS text messages to mobile phones outside the DTN region, the NSIMGW application provides the necessary connection on the DTN/Internet gateway at the N4C-LTU office.
4.4.2 NSIM Use-case Model

The following overview (Figure 4.5) illustrates how the NSIM components are connected. The end-user had the NSIM component pre-installed directly on the computer (Client 1, 2...N) and the gateway (NSIMGW) application was installed on the N4C-LTU gateway server by N4C-LTU developers.

Figure 4.5: Block diagram of the NSIM service (created by Samo Grasic)
4.5 Localized Application Prototype

Even if the Slovenian test bed was more advanced in this respect, both test beds used technical tools such as weather sensors to measure the environment, so this was a subject of collaboration between the N4C partners. The data generated by these sensors were usually sent over DTN for various reasons. While Slovenia had more of an environmental research intention, the Arctic test bed in Sweden made the data available directly to the end-users as a DTN service. In conjunction to the weather data we also put up Web cameras for the trials in strategic places, which allowed end-users to view the daily camera “snapshots” from the local area, Staloluokta (see example in figure 4.10), and the helicopter base in Ritsem.

The local services were extra useful for us in the test team, hikers and locals. The data produced by the services has been appreciated by people living in or visiting the trial region, since it has a “direct” connection to the DTN nodes they are using, meaning that users does not need to wait a long time to get the data in their own computers. The final prototype in 2010 provided the hikers and local participant families with advanced weather information from an weather station temporarily installed in Staloluokta.

4.5.1 Components of Local DTN Applications

To monitor the weather conditions in the field, N4C-LTU deployed two web-cameras in different locations, and one weather station for gathering weather data. Images from the web-cameras and meteorological data from the station were sent once each hour to the DTN border node gateway, were they were uploaded to our Web server at LTU. Having this kind of information available was crucial for analyzing problems in deployed systems when the test crew was not physically present in the field. It allowed us to estimate how much solar and wind power our outdoor nodes managed to harvest, and potentially detect damage from the water (e.g., oxidation of connectors) or wind (e.g., broken or twisted antennas).

4.5.2 Local DTN Applications Use-case Model

This abstract use-case scenario (Figure 4.11) illustrates how local applications are connected and used in the prototype that was used by N4C during the 2010 summer trial.
4.6 GUI and HCI Prototype

When the participating DTN user wants to use the Web services, he or she needs to be able to locate them quickly, be able to interact with the services properly and with no particular difficulties compared to using a computer connected to “normal” Internet. In order to ensure good usability, the GUI must provide a clear, straightforward design with high usability and bring satisfactory response to the person using the services. Our intention was that the end-users of our DTN services were to be greeted with a “homepage” after only a click of a button that turned the computer on. The GUI would then help them to complete their task, connecting the person to the different DTN services. N4C-LTU’s aim for the user interface was:

- to provide a “home” portal for DTN users. Equipment that is used in N4C’s final prototypes comes pre-loaded with DTN and services software, all accessible through a GUI. They can access DTN services like caching, podcast, e-mail and messaging.
that the GUI would be installable on a “village router” or a client computer, depending on DTN setup (i.e., infrastructure with server/router or ad-hoc with no central application service cache).

to gather services or access links to services in one place for easy access.

to provide a clear design that would make the device easy for a user to use and possible for a developer to expand as the network grows.

Since the N4C project start in 2008, a simple GUI existed that allowed local people, tourists and developers, to try the functionality of the DTN applications. In 2008 most activities were presented on a simple HTML web page, basically a list of links that forwarded the user to a web cache which served a copy of the web site that was chosen. Little or no development time was spent on improving the GUI, since the first N4C trial for the LTU team had a more technical focus. Getting hardware, network and the software to run together was considered a much more critical thing in the beginning of the trial. However, during the field test, many tourists used the simple interface and a PRoPHET application GUI (for messages). The test team was then present, guiding and encouraging users to try delayed communications. Although we did not have an explicit agenda to collect feedback about GUIs during the first summer test, impressions were created both from talking and interaction with end users which were mostly tourists in the camps of Saltoluokta and Staloluokta.

In preparation for the second N4C-trial, the GUI was completely re-designed, as compared to the first SNC style. We rebuilt the web page and continued to use a simple design with help of mainly XHTML and PHP scripts. The aim was to provide a more service oriented DTN solution, for tourists and local families. PRoPHET was an external application for messaging, thus it had its own GUI. The main changes to the overall user interface was how it was interconnected to rest of the DTN software. The most DTN software was moved to computers’ harddrive. Software such as Wwwoffle and PRoPHET was included to the “ad-hoc” DTN system we aimed for in the second trial. When a person used the DTN on the computer it worked against the “localhost”. This means that GUI and application content is always and instantly available. The PRoPHET application running in the background handles the communication to other devices through wireless connectivity. For debugging reasons the PRoPHET is always accessible for us developers, and even to the end-users if they wanted. Its GUI outputs status about connectivity to other DTN nodes and bundle data transfers in real time. This was necessary for debugging, even though it potentially could bring complexity or “clutter” to some users’ experience.

Heading towards the last N4C trial, our aims were to take a big step forward in terms of usability and user involvement. With this in mind, N4C-LTU made improvements to the HCI aspect. First of all we prepared all end-user computers to automatically boot up to the new GUI, which provided more information and interaction alternatives for the user. All the users had to do was to press the power button, and the system boots up with the PRoPHET application that works in the background. Users are greeted with the Firefox Web browser, displaying the links to various services. Charging adapters were included
in the hardware we provided, so the users easily could charge the batteries when they had access to electricity outlets in field. Otherwise, users could come to us, in the camp cabin, and exchange the battery for one that is charged.

4.6.1 Components of GUI

We endeavored to develop the GUI components according to design and Web scripting standards such as those proposed in the W3C organization² (W3C, 2013). The three main components of the GUI were installed on each end-user laptops and on the DTN gateway at our office. Participating end-users used a common Web browser, Firefox, to achieve most of the DTN tasks - something that could be recognized. The other inner workings of the GUI, such as the server and the various scripts, were hidden away from the end-user’s view. The technical components used to build the GUI were XHTML, PHP, Javascript and BASH. A web server on each computer, for hosting the GUI for each application, was installed. For a more detailed description of the components implementation and example setup, see appendix A on page 106.

4.6.2 GUI Screen Captures

The following (Figures 4.7, 4.8, 4.9, and 4.10) are screen captures taken from an actual end-user client laptop used in N4C-LTU’s 2010 summer test. These pictures represent examples of the end-user perspective on the DTN application services.

²World Wide Web Consortium (W3C) - www.w3.org
Figure 4.8: Podcast application screen capture (Radio news and request function on the right)
Figure 4.9: NSIM client user interface screen capture (NSIM message)

Figure 4.10: Webcam application screen capture (Camera snapshot from Staloluokta)
4.6.3 GUI Use-case Model

This use-case scenario illustrates how the graphical user interface connects to DTN applications and with the end-user. This was the case scenario during 2010 summer trial:

![Graphical User Interface use-case Model](image)

1. End-user can access the DTN applications through the GUI.
2. The applications exchange data over DTN with other nodes (users, mules, or stationary nodes) to keep updates.

Figure 4.11: Graphical User Interface use-case Model
4.7 Evaluation of feedback

This section evaluates results from various feedback types, that have come from experience of developing and testing DTN application services in the office lab and in the Arctic field. All feedback have contributed, more or less, to the final prototype applications specification and implementation. At the end of this section, a table relates feedback $x$ and its impact on prototype detail $y$. There is no particular order of importance, but the feedback structure is divided into four sections for an easier understanding for each feedback type’s contribution:

<table>
<thead>
<tr>
<th>Category of feedback</th>
<th>Feedback type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locals and Hikers</td>
<td>Technical feedback</td>
</tr>
<tr>
<td></td>
<td>Light individual user feedback</td>
</tr>
<tr>
<td>Focusgroup interviews</td>
<td>Technical feedback</td>
</tr>
<tr>
<td></td>
<td>Strong individual user wishes</td>
</tr>
<tr>
<td></td>
<td>Local politics</td>
</tr>
<tr>
<td>Environment</td>
<td>Weather</td>
</tr>
<tr>
<td></td>
<td>Local area topography</td>
</tr>
<tr>
<td></td>
<td>Limitations</td>
</tr>
<tr>
<td>Own/Team</td>
<td>Discussions</td>
</tr>
<tr>
<td></td>
<td>Diary/notes</td>
</tr>
<tr>
<td></td>
<td>Impressions</td>
</tr>
</tbody>
</table>

Table 4.2: Types of feedback gained from N4C-LTU Trials 2008-2010

4.8 Locals and Hikers feedback

During the field tests an indefinite number of spontaneous ad-hoc interactions with people took place. The people we met were both tourists, hikers and local people that stayed overnight or lived within or around the trial areas. Feedback was generally packaged as curiosity, questions and discussions. Also when locals were willing to help us (like small tips for mounting the nodes, or help with local logistics), this is a kind of feedback.

Most days during the field trials we met hikers and tourists in planned or unplanned events. In 2008, the first week in Saltoluokta and the second in Staloluokta, we had our first encounters with direct feedback from people that had tried our DTN services. End-user formal feedback consisted of sessions with hikers that came to our teepee tent where DTN services were provided by the combined N4C and Tannak team. The sessions were from lunch to evening. People came to the tent, talked a bit, had coffee and tried the services.

In Staloluokta, we often sat in the tourist dining room during afternoons between lunch
and dinner, and also in the evenings. We got response when hikers tried our services
during the time we were there (we brought with us the devices every day). The user-base
was not exactly the same in the two places. The difference between the teepee and the
tourist dining/social room was that we invited ourselves to the tourist “place”, but in-
vited tourists to “our” place in the teepee hut. If this changed the behavior of the users is
difficult to say, but in the teepee the users came because of the curiosity or the free coffee
and waffles. A second difference might be that the hikers in Staloluokta often were very
tired from long days of walking, even if their mood was good. Other, informal feedback
was based on spontaneous meetings around the trial areas, where discussions of DTN and
its services took place. But we were mostly worked in a formal way in the teepee tent
the first week, with demonstrations.

In 2009, N4C-LTU worked with the trial field first in Staloluokta where we meet with
locals and hikers in formal and informal ways. This was done much like in 2008, but with
the updated services in the public dining room. We also had formal meetings with the
 cabin keepers in Staloluokta and Stadda. Informal feedback was given during visits when
 hikers or locals came to our cabin and asked questions. At other times, it could be just
 someone we met on a path while we were working outside. We also participated in the
extremecom (Extremecom09, 2009) conference³.

In 2010 we had similar meeting in Staloluokta as previous years. This time the “for-
mal” meetings with the end-users occurred mostly with locals, since they were the real
focus for this trial, participating in the extended testing of our services. These meetings
were held in their homes or in “our” cabin. We also had informal meetings with both
locals and hikers, when working outside or when people came to our cabin asking ques-
tions about the DTN services. The feedback I got in 2010 did obviously did not affect
the DTN service prototypes, because the last iteration was already in use, and this was
the last N4C summer trial.

4.8.1 Technical feedback

We received negative feedback when people were using the “Nokia 810”, the smartphone
model that we provided for end-users in the field tests. The device display was hard
to read sometimes and it was not easy or even possible to browse the pages in the web
browser even in good weather. It was a small device and not very user friendly in the cir-
cumstances. The responsiveness of the device was not optimal (because of the hardware
inside). During cold Arctic summer days, the screen was less responsive. The battery
capacity was also lowered in the devices because of the cold. Most of this was also true
for the tablets we used in 2008.

I received questions regarding if one could browse other pages, besides those that were
not in cache. This was a quite common request, and natural, since we had chosen just
a few web pages to be available as a proof of concept Web cache application (the aim
in the year one summer trial - was to get the field lab environment started). Overall, a

³The Extremecom09 conference with IT researchers was held in conjunction with the N4C trials in
Staloluokta and Saltoluokta

90
larger choice of web sites and services was requested.

Confirmation about application data was strongly suggested. A typical example is to know whether one is reading the newest version of a web page. Or if not, why not?

4.8.2 Individual user feedback

People sometimes asked if they could use their own accounts (such as Hotmail or Facebook). This kind of extended service was impossible for us to implement during the N4C, but is an important and noteworthy request. There were quite a few non-Swedish speaking hikers around. They had less use of the service, during the first trial (although an English news web-page was available). A typical feedback that I received from people, mostly hikers, was that they were skeptical about the whole DTN technology. Some were just against any IT in the mountains when hiking (understandable stand-point). Some hikers were only a bit skeptical but definitely saw some potential for the local people. Then there was another kind of hiker that was very optimistic about sending emails to friends and relatives, or read updated news. However, hikers were not able to see the full potential in the DTN that we provided, because of the round trip time (RTT) of a day or even two. They stayed mostly for just one night, so they could not test the service in the same way as cabin keepers and the local families could.

4.9 Focus group interviews

During two focus group interviews in Staloluoka 2010, discussions about and around DTN technology took place. Three families responded to existing prototype behavior after having used the DTN application services. There were discussions that had much to do with local political issues and Sami culture. The focus group technique was not the only means of getting feedback, it was rather seen as a way of getting a qualitative feedback from long term testers in a formal way. Perfect focus group feedback was hard to achieve, simply because we were busy with the technology. If we had had more time to think about the local situation, we could perhaps have created better focus groups and feedback possibilities. Now, we had to “fit in” time for focus group sessions when the local families had time.

4.9.1 Feedback from the focus groups

Feedback showed what the individuals of the local families would have liked to see, or see more of. One family had a strong wish for a DTN network for their own, and the community’s benefit. For example, the kids could be helped if they had access to a service that connected them to friends and teachers. Those with jobs in the cities would like to be to work over DTN. Doing banking business (i.e online-bank) was high on the wish-list, even if it was understood that something like that probably was hard to implement with DTN at this point.

The time-delay is a critical point to make here. Many locals are used to waiting a long time for more or less everything, living in such remote place. Hence, they they find
it acceptable if the Round-Trip Time (RTT) is about a day or even two, for e-mail or requesting a podcast, web site etc.

4.9.2 Technical feedback from the focus groups

In regard to acceptable round trip time, a point was made that confirmation of everything one does is important (in-application feedback trough the GUI). For example, if one sends a mail, then it is hard to know whether is has gone trough the DTN or if it is stuck somewhere. The only way to know for sure is when the user gets a response on that mail. The requested feature is that GUI feedback is available for any application data (if it has been confirmed successfully sent or arrived).

One local tester made it clear that he prefers quality over quantity. This same person had technical knowledge and was curious to test our DTN services. Many persons with some technical skills were very helpful when they noticed odd behavior in the DTN connectivity or the applications.

The families that had tested our service, had used the computers mostly every day, as they reported to us afterwards. After our quick instructions they had almost no problem using the computers or the software. The biggest hardware problem was probably the power issues. In Staloluokta, electrical power is not always a given thing as one had to resort to solar and wind-generated electricity. Thus, that is something that eventually could affect the usefulness of the services. Luckily, since locals were well aware of this, some told us that they often provided power from their own power sources, i.e., solar panels. The power adapters we had supplied with the computers were compatible “car” chargers.

4.9.3 Local politics feedback from the focus groups

This feedback type was partly discussed in the focus group in Sami language. The subject of local politics was the IT development in general for the Sami people and lack of information about what is going on. It does not directly concern the development on the DTN applications, but it was something that the locals started discussing among themselves.

4.10 The Field/Environment feedback

The environment in which we tested our prototypes was among the really unique circumstances in the N4C project. The nature itself is important to the Sami culture, and national heritage. In that sense, the environment had to be recognized by us researchers as a factor of importance. I tested the prototype applications in iterations as inspired by the spiral model Boehm (1988), together with the colleagues and other project partners from N4C. I (we) noticed during these trials was that the environment’s influence was one of the more basic and given input for the application requirements. It also put limits how we could test our prototypes. The environment affected what we could do or when we could do something.
Both in Saltoluokta and Staloluokta, we had either the Tannak team or Fritz-Åke, our field test manager to accompany us. The project partners from Ireland had Tannak to accompany them in the field during their trial in year three. We were dependent on their knowledge of the environment: they introduced us to the nomadic herding culture and the living conditions. For us to function efficiently in the field lab environment, it was essential that Tannak and Fritz-Åke helped to initiate the researcher-user relationship with local families and events in the Sami villages.

4.10.1 Weather feedback

The harsh weather was something that we had to accept as being part of the environment in the Swedish Arctic mountains. During the summer it is of course warmer, but still quite cold (~0°C during the night even in Saltoluokta). This affected our testing behavior, for instance, it rained quite a lot during one trial period and thus we could not go outside for longer periods for maintenance, or for visiting local people. However, especially during trial two and three, a more pervasive system was in place, and this meant that all software needed to function even during the worst weather. The installations were more or less static. Samo’s static node design meant that DTN application services could be up and running in cold temperature, heavy rain and heavy winds. Nevertheless, the weather could sometimes make the connection fail, or make the batteries run out of power quickly as result of lacking sun or wind. This initiated a change in the DTN application: allowing them to exist in the client computer itself. In this mode the applications became more pervasive, at least if the computers could be charged once in a while. No wireless connection was needed to browse the cached web, for example (but the cache could not be updated in that state).

Based on our experience, hikers and local people were constantly asking about the weather. Even among ourselves the weather forecast was a nice to have information, obviously. A weather service named “StaloWeather” was added to the application software in the web portal we had created. The services offered in 2010 were the local Staloluokta weather with precipitation, and weather reports from various web sites in the cache. The local ‘weather-report’ was generated from a small weather station locally installed by the N4C-LTU team at the camp center.

4.10.2 Local area topography feedback

The topography in the area also helped to bring our application services to a more pervasive state. The place where the locals have their houses has some steep hills, and the line of sight is not good enough for any device to reach a hotspot (i.e., a central node near the helicopter landing place) at all times. Overall, the experience of being in Staloluokta is that quick travel is not very common. Mostly, one travels on foot on a rocky path. The topology meant that, when it comes to hardware, the lighter the better. Inexpensive, light-weight, replaceable and usable enough, the Asus EEE 10” netbook was chosen as a “standard” hardware platform for the end-users. The lack of trees made solar charging possible. The wind power generation was possible due to Staloluokta’s windy nature and open air.
4.10.3 Limitations

While doing DTN installations in the trial area, there is one thing in particular that always affects the routines. The National park laws prohibits any installation which is permanent or leave marks in the nature. This was something that we had to adapt to from the first trial. The distances to Internet connectivity was also a limiter. The time-delay for sending and receiving a response in this environment made me think of the technical “robustness” needed in the application software. For each iteration this became a core issue. Things must work, or it will be very hard to attend the problems during the trial. There was not much room for small errors in the software.
## 4.11 Table of results from feedback

In this table the most important and biggest development, changes, improvements etc., is presented with references to the corresponding sections in the development prototypes in the Result chapter. Feedback from focus group sessions is not included, simply because they happened after the last iteration of the prototype.

<table>
<thead>
<tr>
<th>Category and feedback</th>
<th>Cross-reference to prototype development</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>From locals and hikers</strong></td>
<td></td>
</tr>
<tr>
<td>End-users have hard time using small Nokia N810 device.</td>
<td>New Asus EEE netbooks chosen as end-user machines, see page 63 and 93.</td>
</tr>
<tr>
<td>Requests regarding cached web sites.</td>
<td>Expanded web sites choices, see page 63</td>
</tr>
<tr>
<td>Requests regarding ad-hoc browsing.</td>
<td>Created and tested a web-request experimental function in 2010, see page 63 and 66.</td>
</tr>
<tr>
<td>Issues from end-users about confirmation.</td>
<td>Created for prototype in 2010 was a small time stamp when latest web cache bundle arrived on the end-user computer, see page 106.</td>
</tr>
<tr>
<td>Requests of extended services such as “radio”.</td>
<td>Created for prototype in 2009 a Podcast service, see page 54 and 74.</td>
</tr>
<tr>
<td>Difficulty operating the machine and navigating the applications.</td>
<td>Rebuilt, made improvements in usability/ease of use for the GUI and hardware, see page 83.</td>
</tr>
</tbody>
</table>

Table 4.3: Table of results from feedback
<table>
<thead>
<tr>
<th>Category and feedback</th>
<th>Cross-reference to prototype development</th>
</tr>
</thead>
<tbody>
<tr>
<td>The environment</td>
<td></td>
</tr>
<tr>
<td>Weather and Topography affected use of application services</td>
<td>A pervasive system was built in 2009 and improved for 2010. see page 54 and 63.</td>
</tr>
<tr>
<td>Weather services was important for us and locals/hikers</td>
<td>Implemented local weather data in the Web GUI, see page 82.</td>
</tr>
<tr>
<td>Topography limited pervasive computing</td>
<td>Utilized ad-hoc networking and “localhost” applications, see page 54.</td>
</tr>
<tr>
<td>Own and Team discussions</td>
<td></td>
</tr>
<tr>
<td>Meetings with N4C partner Folly Consulting</td>
<td>A set of web cache scenarios, see page 72.</td>
</tr>
<tr>
<td>Discussions with co-workers (Le Samo)</td>
<td>Expanding services with Podcasts, see page 74.</td>
</tr>
<tr>
<td></td>
<td>Improving GUI, with better design and usability, see page 83.</td>
</tr>
<tr>
<td></td>
<td>Network integrity/VPN setup, see page 56 and page 112.</td>
</tr>
<tr>
<td></td>
<td>Integrating NSIM, Webcam and Weather service to new GUI, see page 83.</td>
</tr>
<tr>
<td>Documentations from notes and diary</td>
<td>Code and setup examples, see page 106.</td>
</tr>
<tr>
<td></td>
<td>Expandability for future functionality, see page 104.</td>
</tr>
</tbody>
</table>

Table 4.4: Table of results from feedback continued
5 | Realizing DTN Applications for Communications
Challenged Arctic Communities

Bush (1945) was one of the early pioneers that formed the concept of databases and data connections in the 1940s. In the 1960s, Licklider and Clark (1962) and Engelbart (2001) realized the ideas with the computer networks and augmented reality. In the end of 1960s, the first hypertext systems and online computer networks were available. Today, Delay-Tolerant Networking is a technology that, not unlike Bush’s ideas, tries to realize the sharing of information where previously not possible. Being a technology that is researched for space communication, and for earthly “extreme” environments, DTN represents the state of the art in networking.

5.1 An HCI and Participatory Design Approach

I have chosen Human-Computer Interaction (HCI) and its history as the theory base for explaining the development of the DTN web applications. HCI provides a history of technical development of information technology artifacts and how they can be applied in daily life. HCI provides historical context for the DTN applications in general, especially seen in the context of expanding web services.

Bricklin and Frankston (1984) furthered the HCI community with their innovative VisiCalc application for the PC. Smith et al. (1982) at Xerox PARC did the same with the “Star” GUI for the computer desktop, where they put the user interface first, before the hardware. In the same vein, DTN applications within N4C has developed from the “laboratory” research state, towards a HCI usability philosophy where a good user interface combined with the experimental technology of DTN. To realize the prototypes’ GUI during the development process, I used various developer tools, such as Bash, XHTML, Javascript, and PHP to be able to combine various segments to work with Samo Grasic’s implementation of PRoPHET.
While I think an “only HCI” strategy would suffice to create good DTN services, some important aspects could be missed. For instance, strong ties to democratic values, and the overall socio-technical environment. Human-Computer Interaction is also very multi-disciplinary, but the HCI tradition has roots in computer science, cognitive psychology and engineering, and mainly puts focus on **interface** design. In the N4C project, the environment was very different from the “normal” scenario. It was unknown for us researchers. The Participatory Design component of N4C was a three year long process, which took place in offices, laboratories and meetings, and in the field during trials. The method was not strictly adhered to, but some of its more commonly used methods were employed - observation, interaction, focus group interviews, and journal notes. Our work was intended to acknowledge key Participatory Design bases such as how end-user participants could benefit from the socio-technical perspective that DTN artifacts would bring. Openness in regard to multi-disciplinary approaches was another factor that I believe was foundational from the start of the project. HCI theory meshes with Participatory Research theory and methods in the Software Engineering process.

The development of the prototype applications was an iterative process. At the university in the winter, Samo Grasic and I worked in a protected lab environment with developing software, hardware and the DTN architecture. During successive summers, we tested the changes and improvements we had made to the system and applications, in accordance with feedback received. Overall, this process is illustrated in the software engineering realm by Boehm (1988); Pressman (2000), as the Spiral Model. How this model portrays the overall iterative work conducted within the N4C work package 3 is described in the deliverable D3.3 report (Grottum et al., 2011, p. 12):

> Each iteration will include an opportunity for user feedback either through foil presentations at project meetings (most important through early iterations), or through seasonal field test (most important through later iterations). Many partners are involved in the development, and have their specific software development environments. The software development process will be a simplified version of the spiral model, which is iterative.

If one look at the development process from the Boehm model point of view, it was its “Customer evaluation” method that was used to obtain feedback from the participating end-users who had tested the applications in the N4C field trials.

My understanding after the trials in Staloluokta in 2010, was that this environment and social context created many challenging requirements on the prototypes and the N4C trials in general. I learned during the 2008 and 2009 trials that improvements and changes made in the iterations, were due to the circumstances in which we as researchers placed ourselves. Being there, in this very place, was necessary in order for us to be able to appraise any feedback from the environment. Suchman (1987) describes the more abstract part of research (or design) and the task of identifying the processes that form human-computer interaction situations:

> ...we look for the processes whereby particular, uniquely constituted circumstances are systematically interpreted as to render meaning shared and action accountably rational (Suchman, 1987, p. 67).
As a researcher I could identify, or even understand some of the participants’ actions while they used our application services. The reason for this was that I lived in the end-user context, with the same limitations as the locals had (or even more limitations, since I was a newcomer). I became a small part of the social context in which I wanted to explore the usability of our DTN applications.

Since DTN “real life” applications are new, and the technology behind them is still in an early stage of development, it is not possible to point to any given evaluation method. At the start of the N4C project, it was not even clear what acceptable means for the end-users. Therefore, it was natural for us to do research in the same environment context as the DTN applications were to be implemented. We tested the technology in small steps, evaluating “usability” involved ethnographic considerations and a participatory design approach. Each step gave us more information about how we could adapt the system for better usability. After each application iteration, we tested the solutions in the field taking guidance from standard software engineering methods. The methods we used affected how we later improved the application prototypes in the iterative development process, and how we collected feedback from the end-users during the last trial.

- Observing and understanding the end-users’ interactions with the applications and their behavior in their own social and cultural context.
- Feedback from end-users through various planned and spontaneous interactions, during a longer visiting and testing period in their village.
- Our own and the team’s experience from the field trials. Since we were isolated in the environment and being “cut-off” from IT communications, except DTN. Thus, we experienced real effects from using our application services, including the things that worked and those that did not.

From the assessment methods, knowledge about the participants’ use of the final prototype application services was derived. Acceptable usability is *when the participants perceive a benefit from their use of the services, even with the limitations caused by the remote site’s natural environment, the technical limitations of the DTN, and the development state of the applications.*

- Usability in numerical terms tell that in 2010, we had an autonomously running DTN system for 53 days, with 1359 e-mails, 328 NSIM messages, and 167 SMS text messages were sent from end-users Romanowski et al. (2011); N4C (2011b).

In the field trial situation the feedback is often directly related to an information need. For example requests regarding web cache sites and “radio” service. Streitz (2008, p. 56) defined the typical end-user’s computer use:

> Normal users are actually not very interested in interacting with computers. They are interested in interacting with information and with people in order to communicate and collaborate with them.
This is partly true for much the feedback from the participating locals and some hikers in the Arctic. Löwgren and Stolterman (1998) portrays some measurable “general” usability cases. Although, due to the circumstances in the trial it was only possible to focus on one of those, which is: “What the test-users think of the system efficiency and helpfulness, based on their subjective opinions”. The feedback was not constructed as a technological feature; it was more of a wish to see a particular web site, to e-mail or message someone, or being able to “listen” to news or weather reports.

However, during interaction with participating end-users, I also received feedback about specific details. Bugs, errors or technical things in the application service design, were pointed out. I think the explanation to this is the way that we worked with the applications when in field. Our approach is probably best described as a “participatory design” approach (Lindberg, 2010; Aagaard Nielsen and Svensson, 2006). We got to know many of the participants really well, and as I regularly lived in their environment for longer periods, they could relate to me. Another supporting factor was the overall positive interest in testing the DTN services. Due to our long-term interaction with the participants, and the tests they did, we got examples of technological feedback, directly from the person(s) using the service, as follows:

- A wish that I would categorize as technical came when the participating families hinted that a web cache service would be even more useful if one could “request” various web sites to be included in the cache. The reason for such a function was that if a person spends much time in the camp, it usually does not matter if the web page will be available only one or two days later.

- Another wish from long-term testers was that it would be beneficial to have a sort of “confirmation” in the GUI when something happens, e.g. when an e-mail has been sent or something has been received.

- Some participants described difficulties operating the end-user devices. The early computers and Nokia smartphone N810 had slow user interfaces, which were also navigation ineffective due to the hardware design.

These are impediments to the end-user trying achieve a task. Bødker (2008) has referred to this when expressing her view of the “cluttered” desktop and Streitz (2008) discuss how end-users normally want to experience computers; i.e. they want no technical difficulties in the way of reaching their task.

A third role of the end-user participants had to do with the location. The unique Arctic Staloluokta camp belongs to an “extreme” environment type, which I was not used to live or even less to work in. The cultural and social context was almost completely new for me. Hence I was dependent on the support from the Sami who had a history there, i.e., our project partner Tannak for year one and our field test manager Fritz-Åke for year two and three, but also many others who helped me to adapt. They were fundamentally critical for me in order to learn how to live there and be able to test and do the necessary research. Also, the camp environment of Staloluokta in itself and the Sami way of living made certain requirements to the prototypes services:
Connections to “the outside world” depended on expensive helicopter rides or a two day hiking walk. How traffic best would travel to and from Staloluokta was unknown at the beginning of the N4C project.

Energy was always a main concern and there was always less electrical energy available than one would want, as the area does not have a power grid or any easily available electrical source. By using wind power harvesting and solar panels we were able to store energy in the static DTN nodes and car batteries. However, we still had to be very economical in terms of how we consumed electrical energy. Therefore, it led to the requirement of efficiency in the services, in both hardware and DTN applications.

5.2 Usability in the DTN Applications

The aim of this thesis has been to investigate the usability aspect of Delay- and disruption Tolerant Networks applications in Arctic settings.

What is DTN application usability?

How do we assess usability in DTN applications?

How do we know if a DTN application has “acceptable” usability?

The assessment in this thesis aspired to portray the nature of the N4C and applications development for DTN; focus on field trials, gathering of feedback from field trials, end-user participation, and the collaboration of the work teams. The research and development carried out by the N4C was aimed at creating and testing DTN application services in real life “extreme” environments, and produced useful experience regarding the usability aspect.

Constructing usable DTN services relies in successful team work and communication within the project. The knowledge sharing between researchers and with the Sami community depended on our cooperation with project partner and herders Tannak and Fritz-åke, who initiated the necessary steps that introduced us to the nomadic environment context. Affecting the usability, is also how we adapt various technologies with DTN technology, from the laboratory to this environment.

Many ideas formed already in the field, during writing field notes and journals as ethnographic field tools (Widerberg, 2002). The process where ideas starts to form is depicted by Löwgren and Stolterman (1998) as the vision, when we think of various kinds of solutions already in the field environment. However, it was in the office environment that feedback was later developed to specifications that were then fed into prototype development, in accordance with the software engineering tasks and the Spiral Model. To produce an application service that would have improved usability N4C-LTU team had two main engineering considerations:

That the most important usability aim was to produce a service, meaning the whole chain of hardware and software in conjunction with service providers (e.g. Telia 2G/EDGE) needed to work seamlessly, as one integrated service. The initiation
point of an application service was normally in our office at LTU. From there, application data was transmitted over the Internet to the DTN border gateway, to helicopters, to a “static” node near the camp center, and finally to the end-users’ computers over wireless LAN.

- To realize some requests and wishes from the end-users, where “some” meant those that were feasible in the work situation we were in. Our experience from previous trials of the topology, climate, technical equipment, and working situation provided knowledge of what was realistic or not.

As discussed by Löwgren and Stolterman (1998), the thinking process of the development did not follow a rational, logical design model. I had thoughts about how to realize the prototypes within the iteration process, meaning that I was thinking about detail such as how I would construct and improve application scripts, and other technical details such as stability and fault tolerance in the operating system environment. I also had my mind on things like the overall implementation process and the execution of field tests, and the integration of the applications in the rest of the system. In essence, my thoughts existed on different abstract levels. The start of the operative picture construction took place when I started to make UML-models (Pressman, 2000). They were created after trial experiences in 2008, from feedback gathered by the test team members from LTU and Tunnak, as “nice to have” features for the end-users in that particular environment.

Streitz (2008) defined two types of disappearances, which depict how IT artifacts integrates in the end-users every day life. The first is the Physical disappearance, which means that the artifact is so small that the user does even have to physically interact with it. However, such small devices are not available, with the hardware specifications we need. The other disappearance is the one that we aimed for - Mental disappearance, which mean that the end-users could take advantage of the artifact that we developed, since we minimized their trouble using it, i.e. we tried to make hardware disappear from their perception as much as possible. The nomadic participants’ context were suitable for a ubiquitous service, which we augmented in form of i.e. lightweight Asus EEE laptops with larger battery capacity and better portability, ad hoc networking, and DTN messaging applications. This way, the herder families could test the prototype applications in their own context, without being too restricted by the technology.

The spiral model of Boehm (1988) includes a risk management process, which we adapted to our needs. In the Arctic, if the service has not been tested thoroughly, technical sensitivity increases and things are more likely to fail. Thus, as a developer, one needs to be sufficiently certain that the functionality upgrade is justified in terms of usability versus risk. Practical experience of the DTN application field trials suggests that the complex nature of the DTN introduces variables, such as time and distances. In the Arctic settings where we worked, application data transmitted over the DTN can take a full day or even more. The possibility of traveling around to obtain a larger picture of the network is in reality very small. Nor is it possible to use telephones. Hence, the DTN is already in a high-risk state if, as a developer, one cannot quickly go to attend or investigate a possible error.
The usability of the developed DTN application prototypes is the *synergy between the various components that forms the DTN applications*. Effectively, the building blocks of *usability* in the DTN application services result from iterations of the *configuration, software* and/or *hardware*, which are designed to function in the particular context.

- Applications that adapts to the end-users’ need by including them in the development process. We provided the nomadic end-users with a ubiquitous DTN service implemented with ad-hoc wireless data transmission.

- Including basic application services that are wanted by the end-users, but not possible without DTN.

- Hardware that is adapted to the environment, but still manageable and understood by the end-users. We used the daily helicopter route for data “mules”. We adapted to the weather by implementing solar panels and a wind power generator to our DTN.

- A DTN system that can function autonomously and is stable, providing a platform for the applications, i.e. Samo Grasics’s PRePHET implementation and various other configurations in the hardware and software that we made.

- Making the connection between the user interface and DTN as seamless as possible, thus we provided a user interface that strive to provide the users with a *service*, not technology.

- The web cache and podcast applications were evaluated continuously during the field trials with the end-user. Even though we had very limited resources to provide a wide selection of web sites and podcasts, both applications were relevant, because the data that was there provided wanted information that otherwise was non-existing in the village.

- The end-user participants exhibited interest in the services by making their own suggestions, talking about functional and non-functional things. They were eager to test more, in order to get more connected to the outside world.
5.3 Future Work Discussion

The Web caching service developed by N4C-LTU allows for increasing functionality. As web caching is merely applications using the DTN as a transport medium, should be possible to develop new models for more advanced web caching functionality, as a service for end-users as the network matures and expands. For example, that could mean implementing new functionality and scenarios, handling certain types of web sites, and supporting a growing number of users. The application would be implemented in accordance with their own particular needs, but within the limits of what is technically possible.

Web services development is potentially a major challenge in remote areas. It has been discussed within the N4C project that developers for DTN web caching could develop “plug-ins” that tells the DTN application software when time-delay is an issue, so the application can behave differently depending on its location. This could enhance the transparency for the end-users traveling between Internet-connected areas and Communication Challenged Regions. Expandability is important if the service is to be useful in the long term. Web-caching applications should be made up of free applications such as web proxies and operating systems that can be modified and controlled using script languages, etc. Making the software prototypes, ideas and knowledge gained from the N4C available for any user/developer community that wants to build upon the existing solution can be important foundation work for future DTN web caching and services.

The podcast service is one of many typical services that could be used within DTN. Our prototype exists with the “provider pushed” model plus the basic request function. This means that predetermined podcast subscriptions are being sent to the DTN a number of times per day, and that end-users can request a new podcast subscription whenever they want. This service, could potentially be expanded to include even other scenarios like “ad-hoc” use, by letting users add their own wishes for podcast shows, by sending a request to an administrator of the DTN network or by searching a cached database of podcast subscriptions within the DTN. This could mean that a user in a DTN realm could connect to the web GUI with a web browser and point to whatever he/she wanted to listen to. Much like the web cache application, podcasting can be very demanding of the DTN due to file sizes, especially if the delays are long and files are stacking up at a location in the network (this did occur in our project, at the border node). The problem presents itself if the data “mules”, such as the helicopters, do not have enough time to transfer all the data over the wireless link during landing and take-off. Applications such as podcasting need to be managed in such way that they grow with the DTN as it expands. That means if the network accelerates, more traffic functions can be allowed for usability and ubiquity, without sacrificing the stability of the network.

The NSIM application that Samo Grasic developed for DTN has gone from being just a development support tool in the field, to a useful messaging client for end-users. It has found its place in the “state of the art” DTN real life applications. There are some features that would be “nice to have”, such as multi-platform messaging integration with Web GUI, and administration tools. NSIM can be installed quite easily to DTN ad-hoc networks such as it was done by N4C-LTU, in the Arctic area. That NSIM became a
useful messaging tool among the Sami during the 2010 trial, indicates that it has the possibility to be developed further with participant end-users included from start, as our trials have shown that messaging is a useful tool among the users. While e-mail, instant messages, and SMS functionality has been tested successfully; one could imagine that other popular communication methods might be integrated in the application, such as Google Hangout, Facebook or whatever is the commonly used messaging tool on Internet at the time.

The localized applications were developed with researchers and end-users in mind. In general, DTN such as in N4C test bed will always have some localized applications because certain environments involve unique cultural and social contexts. In our trials images of landscape or DTN hardware equipment and visual weather information proved to be useful. Services beyond these are much up to the end-users and their wishes for the particular region they live in, if it would be feasible and possible with the DTN system. I believe that no two DTN solution will look entirely the same; instead unique area data and local traditions will be exploited to maximize the potential of the developing DTN.

Graphical User Interfaces and Human-Computer Interaction has tremendous possibilities to improve with new technology, as new devices are getting smaller, more energy efficient and lighter. It is possible that a future DTN device could be a bendable and disposable screen using oled or other display technology with touch capabilities. However, there are some principles that I think is important for the usability aspect:

- When new services are developed for DTN, they should be easily available on a GUI. The less the users have to do in order to achieve their task, the better. The platform-independent XHTML and PHP languages purpose in our case were used to serve and display services such as podcasting and web caching for end-users.

- Scripts and mark-up languages can be used to control the kind of functions the user can access, or to hide unnecessary complexity. This can of course enhance usability since it will become more obvious to the end users what they can or cannot do.

- For users with mobile devices (that are compatible with DTN applications) Web GUI can be built with added support for smaller screens with less screen resolution, preferably following mobile web best practices and developing guidelines from W3C.

- An aim might be to provide a seamless transition from a Internet-enabled area to a non Internet-enabled area (i.e CCR). The big challenge here is to make the computer to be able to know where it is, what state it is in, and if it should switch to “DTN mode” or not. For the user, the advantage would be that no special attendance would be necessary other than on regular hardware maintenance.
The DTN applications made by N4C-LTU are presented here with details that were not needed or reasonable to include in the main result chapter. Naturally, far from everything ever produced in N4C-LTU are presented here. I have chosen to include a few selected examples that I think are basic software engineering solutions, but nevertheless had an impact of the overall DTN application service. Note that these examples not meant to be a manual. The aim is to portray the general logic behind some of our configuration of the DTN applications. The scripts examples uses mostly “bash” scripting language, which was available in the Ubuntu operating system.

A.1 Components of Web Caching Application

The components presented in this section make up for the last prototype functionality of the DTN web caching used in the latest N4C-LTU trials, WP8 test and demonstrations (Romanowski et al., 2011).

**Wwwwoffle server/cache engine**

This application was installed on LTU Internet connected gateway (Ubuntu Linux 9.04) that N4C-LTU researchers installed and maintained. It fetched data from web sites and stored it in a special folder location in the hard drive. Wwwwoffle (Bishop, 2013) is configured by its editing its config file, where we can choose which kind of file types it will be allowed to download from web sites, limitations in the file/cache and how many threads it will use.

**Web cache update script (server)**

This script activate and deactivates the Wwwwoffle engine regularly. It allows us to specify how much of a web page we want to get (the depth of possible link clicks). When activated, Wwwwoffle updates the web sites we have specified and downloads any existing user web requests, makes a package of the Wwwwoffle cache and sends the package to a folder in a location so that PROPHET application can process, and send it to the DTN end-user nodes.

**Wwwwoffle client/cache engine**

Same third party open source software as above, but configured for use in a client computer existing in the DTN realm (meaning it is always in “offline” mode). It serves the
cached web pages to the users web browser whenever the user clicks on links through the N4C Web GUI. The setup is almost the same, however less complicated than the server installation. N4C-LTU used Asus EEE 901, 1000H and 1001HA with OS Ubuntu 9.04.

**Web unpack script (client)**
A simple script checks for the incoming package (tarball) every few minutes and, if found, unpacks it to the wwwoffle cache storage location on the computer. It notes down the time for the event.

**Web GUI component (client)**
This is a web page portal component that exist on the users’ DTN-enabled computer. From this portal updated pages can be accessed with the Wwwoffle cache. New web pages can be requested through this portal, if the participant end-user wishes (requires the request function to be enabled). A time stamp, that is generated in the “Web unpack script” is viewable in the GUI. This way, the user can easily check whether the service works as intended or not.

### A.1.1 Prototype example of Wwwoffle setup for the DTN gateway server

- Search for wwwoffle in Synaptic Package Manager and install the latest build, or type:
  ```
  sudo apt-get install wwwoffle
  ```
- To see status, in command window, type:
  ```
  sudo wwwoffle -status
  ```
- The built in web server index page can be located at:
  ```
  /etc/wwwoffle/html/default/local/ and:
  http://localhost:8080/local/index.html
  ```
- The Wwwoffle config file needs to be configured to the specific machine. IP, password, custom cache-directory etc. It is located at:
  ```
  /etc/wwwoffle/wwwoffle.conf
  ```
- The default directory for cached webpages is located at:
  ```
  /var/cache/wwwoffle/http/
  ```
- Requesting a page from computer where wwwoffle is installed:
  ```
  sudo wwwoffle www.google.se will fetch example URL and store it locally when wwwoffle has internet connection. If offline, it will save the request.
  ```
- The following example:
  ```
  sudo wwwoffle -F -r1 -g www.dn.se caught all pages from dn.se from page and depth -r1 (one click deep). The -F means to force even if it is already in cache. The -g means no images, scripts, stylesheets, frames, objects.
  ```
- Type `wwwoffle -h` for more help.
A.1.2 Example of Web GUI client preparation

These following configuration options were set for Asus EEE 901, 1000H and 1001HA with OS Ubuntu 9.04.

- Make the browser talk to the wwwoffle and apache installation, using the menus in Firefox (Example using version 3.x). Navigate trough Edit -> preferences -> Advanced -> Settings -> Manual proxy configuration -> and in “http proxy”, enter:
  localhost and for Port: 8080
- In Edit -> preferences -> Main -> home page, enter:
  http://localhost/n4c
- Make the browser less sensitive to errors while in the offline realm. In the url-field type:
  about:config
- Set app.update.auto to false
- set app.update.enabled to false
- Set browser.sessionstore.enabled to false
- Set browser.sessionstore.resume_from_crash to false

A.2 Components of Podcast Application

The podcast application is made up of free software and scripts, similar in concept to those used in the web cache application. Component overview:

**gPodder**

gPodder’s (Perl, 2013) main task is to download audio and video content (podcasts) from the Internet and store it on a hard drive. It is available for most Linux distributions, FreeBSD, for Windows, on Nokia Internet Tablets (N800 and N810) and on the Nokia N900 mobile computer. The main usage for gPodder is to automate the download procedure of podcasts (audio files) whenever new appear on the Web. The executable application exists on a stable internet-connection. The gPodder application is used by N4C-LTU as a base for DTN podcasting. It is installed on the LTU gateway which is marked in the use-case models on previous pages. It has many features but there are a few core advantages that are important for our purposes:

- Free and open source (GPL), it aligns with N4C project’s general aim to be an “open” project and support future development for DTN podcast applications.
- Focus on usability, meaning that the application is easy to implement and use in N4C-LTU DTN podcast application. Supports RSS, Atom, YouTube and Soundcloud feeds which potentially make it easy to add and manage end-user requests for N4C-LTU podcast application.
**DTN podcast server script**

This LTU-N4C script interacts with the gPodder application and forwards new podcasts to the DTN (via the PRoPHET implementation application). This script is installed to the same server, LTU gateway, as gPodder. It is set to execute once per day. When it runs it will check for new podcast requests, add podcast requests and download new episodes from the providers (if there are any). The script will then append a date on each filename and move the files to PRoPHET folders, which means that they will be sent to the DTN network.

### A.2.1 Prototype examples of podcast request function on server

The following code is set to execute on the server a few times per day:

```bash
for pcreqdir in 'ls -d $podreqdir/*'; do
cd "$pcreqdir"
for pcreq in 'ls -l | grep -e "podcastreq"'; do
  if [ -f $pcreq ]
    then
      echo "found a request file: $pcreq"
      # how many lines does reqfile contain
      lines='`cat $pcreq | wc -l`
      t="0";
      exec 0<$pcreq
      while read line
        do
          # extract the url and name per each line
          requrl='echo $line | awk ' print $1''
          reqname='echo $line | awk ' print $2''
          echo "adding url: $requrl name: $reqname"
          t=$(( $t + 1 ))
          # now add it to gpodder
          /usr/bin/gpo subscribe $requrl $reqname;
          # remove the reqfile
          $rm $pcreq
        done
    fi
  done
done
```

The above code example looks for the incoming podcast request file, and extracts the information regarding the podcast subscription (URL and name). It then proceeds to add the subscription to gPodder application through a command line. Other parts of the podcast application takes care of sending podcasts out to the DTN client computers.

**DTN podcast client script**

On the client side (in a CCR), podcasts is shown to the users in the Apache hosted web GUI whenever the DTN has transferred them to the local host computer. Playback is then possible through the GUI by clicking on the generated links (which is done by
the client script). The link opens a window where the sound starts playing. N4C-LTU are using a free software MP3 decoder that has been installed as a part of the podcast application on each DTN end-user computer.

A.2.2 Prototype examples of podcast on client

Our solution for offline playback mode was to install an external MP3 player, from files on USB memory sticks (while in the “disconnected” field environment). However, any MP3 player can be installed easily if the client computer is connected to the Internet during setup and configuration.

Edit Apache2 config file to so the web GUI can access podcasts:

```
sudo nano /etc/apache2/apache2.conf
```

...and in the end of the file we entered:

```
Alias /podcasts/ "/home/n4c/n4c/apps/dftp/podcast/incoming/gw-lime.dtn.n4c.eu/"
```

```
<<Directory "/home/n4c/n4c/apps/dftp/podcast/incoming/gw-lime.dtn.n4c.eu">

Order allow,deny
Allow from all

</Directory>>
```

Apache2 web server must then be restarted for the changes to take effect:

```
sudo /etc/init.d/apache2 restart
```

If the user of the Web GUI decided to create a podcast request (i.e., from the static drop-down list), we create a text file with the podcast subscription URL to the N4C-LTU Gateway destination folder in PROPHET. This connection between the Web GUI and PROPHET application is handled by PHP, by executing a shell script, that makes so that Apache/PHP has the rights to create the file in the operating system and so that PROPHET can process them:

```
$newfile = shell_exec('sudo /var/www/n4c/podcast-scripts/makereqfile.sh');
```

The shell script makereqfile.sh is executable from PHP after we have run “visudo” as “root” and changed a line so that Apache’s “user” (www-data) can run this script:

```
www-data ALL=(ALL) NOPASSWD: /var/www/scripts/makereqfile.sh
```

With these prototype examples implemented, the podcast client on the Asus EEE computers could make and send requests to the gateway server at LTU.
A.3 Components of NSIM Application

Within the N4C project, the NSIM application was redesigned and rewritten from scratch by Samo Grasic, at LTU. To shorten the development time, part of the graphical interface and email libraries were “re-used” from open source projects. To run the full NSIM service, two different applications are needed:

- NSIM – A client application that is very similar to any email client and should be installed on every client machine. A client application itself is enough if users wants to send messages only within the DTN region.

- NSIMGW – In order to allow NSIM user to send and receive emails or send out SMS text messages to a mobile phones, the NSIMGW application provides the necessary connection on the Internet-connected gateway at the N4C-LTU office.

A.4 Components of GUI

The web GUI that N4C-LTU have developed basically consist a web page build with standard web languages, such as XHTML and PHP scripts. The and can exist on any N4C DTN user computer, or be served from a central host. This prototype is built so that user computer serve their own GUI. Each end-user DTN computer comes configured with Apache Web server, Wwwoffle and PHP engine for accessing the GUI. DTN software (PRoPHET) is installed on the side, with no interference with the GUI.

Apache Web server
This open source software is installed to all user computers. It is easily available to the Ubuntu operating systems that we are using on Asus EEE netbooks. N4C-LTU have configured Apache to run at startup, and its mission is to serve the web GUI to the web browser. It is served from localhost with use of a virtual directory called “n4c”. This means that web GUI was accessible at http://localhost/n4c on each DTN user computer.

Firefox web browser
This is another open source software that we use on the end-user computer. It has the task of displaying the files of the web GUI to the end-user. It is configured to automatically start as the computer operating system boots up, and when started it accesses the web GUI so that the different services are immediately visible to the user.

Web GUI
These are the files that make up the visible presentation for the end-user. It is based on simple, straight forward design with XHTML, PHP scripting. It links together the DTN applications available to the user: Web caching, Podcasting, NSIM, Weather and Webcam.
A.5 DTN over Virtual Private Network

The open source software OpenVPN can act as a bridge between two computers over the Internet. That means they will have the same local network. This technique is called “Ethernet Bridging”. To setup bridging you need to specify which network IP you want to use, 192.168.x.x etc. A client computer will receive the servers range of network IPs and take the first free IP. You set what range to give out in the server.conf file. In N4C-LTU, we used the VPN link mainly over two computers, our DTN/Internet gateway in LTU office and the border node in Ritsem.

Begin with installing on the DTN gateway server:

```bash
sudo apt-get install bridge-utils
sudo apt-get install openvpn
sudo apt-get install openssl
```

Then install OpenVPN on the client:

```bash
sudo apt-get install openvpn
```

A.5.1 Configure the Server

Setting up the server is the trickiest. Here is our N4C-LTU example how we setup the VPN service during field trials two and three. When help or more examples are needed; there is a nice guide to follow at openvpn.net.

**Bridging**

The purpose of bridging is so that our N4C-LTU DTN gateway can be connected to the internet and the DTN subnet. It maintains two IP (two logical Ethernet cards) and forms a link between those. We edited the `bridge-start.sh` from the bridge-utils, to bridge with `br0`, with the name of `eth1`. Then we started the bridge with:

```bash
sudo bridge-start.sh
```

A.5.2 Security

To have a secure link, between LTU gateway and Ritsem border node, we generated keys and certificates on the server:

```bash
sudo su
./vars
./clean-all
./build-ca
./build-dh
./build-key-server serve
./build-key client
```

112
Then, we put the generated keys in the `/home/n4c/n4c/scripts/vpn/keys/` folder on the same server. We edit OpenVPN’s `server.conf` to include this:

```
port 1194
mode server
dev tap0
;dev-node MyTap
proto udp
persist-key
persist-tun
key /home/n4c/n4c/scripts/vpn/keys/dtn.n4c.eu.key
dh /home/n4c/n4c/scripts/vpn/keys/dh1024.pem
cert /home/n4c/n4c/scripts/vpn/keys/ca.crt
cert /home/n4c/n4c/scripts/vpn/keys/dtn.n4c.eu.crt
server-bridge 192.168.3.10 255.255.255.0 192.168.3.250 192.168.3.255
client-to-client
duplicate-cn
verb 3
status openvpn-status.log
log openvpn.log
```

As you see we have our 192.168.3.10 IP above, and our range of IPs to “give out”. To make sure that the connected router has an open “port forward” to the OpenVPN server, we have set our office router to forward VPN port 1194 to our DTN-GW (192.168.0.10).

```
VPN can now be executed:
sudo openvpn server.conf
```

### A.5.3 Configure the Client

First we installed OpenVPN on the client:

```
sudo apt-get install openvpn
```

After, we copied the needed key files that we generated to the client computer in previous steps (we used the same folder structure as the server). Then edited the `client.conf` so that it is almost like the server.conf, but with a few changes for the client mode:

```
client
remote gw-lime.dtn.n4c.eu 1194
dev tap
proto udp
key user.key
cert user.crt
cert ca.crt
resolv-retry infinite
nobind
persist-key
persist-tun
```

A.5.3 Configure the Client

First we installed OpenVPN on the client:

```
sudo apt-get install openvpn
```

After, we copied the needed key files that we generated to the client computer in previous steps (we used the same folder structure as the server). Then edited the `client.conf` so that it is almost like the server.conf, but with a few changes for the client mode:

```
client
remote gw-lime.dtn.n4c.eu 1194
dev tap
proto udp
key user.key
cert user.crt
cert ca.crt
resolv-retry infinite
nobind
persist-key
persist-tun
```
A.5.4 Automatic Start

Starting VPN is important if the computer, for some reason reboots. Then it should reconnect automatically, and the system should resume as normal. This procedure should be installed at both ends, the gateway server and the DTN border node. We created an `vpnstart.sh` file and edited it:

```bash
#!/bin/bash
cd /home/n4c/n4c/scripts/vpn/
openvpn client.conf &
```

The small startup-script is placed in `/etc/init.d/` and from that directory, we typed:

```bash
sudo chmod +x vpnstart.sh
sudo update-rc.d vpnstart.sh start 51 S
```

The dot at the end is crucial for the script to work. If the server is rebooted now, the openvpn starts automatically.
Bibliography


Davies, E. (2008). N4c technical work walk-through. Folly Consulting Ltd. a Powerpoint presentation for the N4C Kick-off meeting held in Luleå, Jokkmokk, Sweden.


