Networking for Communications Challenged Communities:
Architecture, Test Beds and Innovative Alliances
Contract no: 223994

D3.2 Evaluation and progress report

Version 1.0

D3-2_Eval_progress_v1-0.doc

Norut/Luleå University of Technology/Folly Consulting/MEIS
Karl Johan Grøttum, Sigurd Sjursen/John Näslund/Elwyn Davies/Boštjan Grašič
www.norut.no/www.ltu.se/www.folly.org.uk/www.meis.sl

ABSTRACT

Starting in May 2008, N4C is a 36 month research project in the Seventh Framework Programme (www.cordis.lu/fp7). In cooperation between users in Swedish Lapland and Kočevje region in Slovenian mountain and partners, the project will design and experiment with an architecture, infrastructure and applications in field trials and build two test beds.

This document constitutes deliverable D3.2 Evaluation and Progress Report. It is an internal evaluation made by the WP3 partners at the mid-term of the project. Based on the feedback given by the evaluators in review meeting no 1 in Luleå June 2009, we have extended the scope of the deliverable to also include a description of progress and future work related to application development.

The report describes the status for development of the applications *Hiker's PDA*, *Meteorological and Environmental Data Capture, Web Caching* and *E-mail*.
The evaluation shows that WP3 has performed according to the original plans and given the answers to the questions that were raised. Based on the original plans in the DOW, experiences gathered during development and tests, and feedback from external reviewers, a strategy for iteration three in the spiral development model is outlined.

AUTHORS

John Näslund, Avri Doria, Martin Bonnevier, Luleå University of Technology,

Elwyn Davies, Folly Consulting,

Boštjan Grašič, MEIS,

Arne-Wilhelm Theodorsen, Tromsø,

Karl Johan Grøttum and Sigurd Sjursen, Norut Tromsø

Due date of deliverable: 31/12/2009 Actual submission date: 10/02/2010

Keywords: Delay tolerant networking, internet, applications, service, rural areas

Document history			
Version	**Status**	**Date**	**Author**
1.0	Submitted version with updated Abstract and Executive Summary	10/02/2010	Sigurd Sjursen
0.8	Changes from Folly	22/01/2010	Karl Johan Grøttum
0.7	Added Future work	15/01/2010	Karl Johan Grøttum
0.6	Added 'Product design for N4C' appendix	13/01/2010	Karl Johan Grøttum
0.5	Added abstract and executive summary	12/01/2010	Karl Johan Grøttum
0.4	Added 'Kindle in a DTN context' appendix	16/12/2009	Karl Johan Grøttum
0.3	Added chapter from Folly	15/12/2009	Karl Johan Grøttum
0.2	Added chapters from LTU and MEIS	04/12/2009	Karl Johan Grøttum
0.1	Created	5/11/2009	Karl Johan Grøttum

Dissemination level	
	Level
PU = Public	
PP = Restricted to other programme participants (including the Commission Services).	**X**
RE = Restricted to a group specified by the consortium (including the Commission Services).	
CO = Confidential, only for members of the consortium (including the Commission Services).	

EXECUTIVE SUMMARY

This document constitutes deliverable D3.2 "Evaluation and Progress Report". It is an internal evaluation made by the WP3 partners: Norut, LTU, Folly and MEIS. Submitted at the mid term of the project, the purpose of the deliverable is to:

- Report the progress of application development
- Evaluate the results achieved so far
- Adjust and confirm the plans for the remaining project period

Work package 3 Pervasive Applications covers the majority of application developments in N4C that are closely connected to software development. The project will develop a limited number of demonstrators or prototypes of applications for a DTN environment. The selected applications are:

- "Hiker's PDA" (Applications for Hikers, Hunters, Herders and Rangers)
- Meteorological and Environmental Data Capture
- Web Caching and E-mail

The evaluation concludes that WP3 has performed according to the original plans and given the answers to the questions that were raised. Based on the original plans in the DOW, experiences gathered during development and tests, and feedback from external reviewers, a strategy for the third iteration in the spiral development model is outlined.

Based on the feedback given by the evaluators in review meeting no 1 in Luleå June 2009, we have extended the scope of the deliverable to also include a description of progress and future work related to application development.

The document contains the following sections:

- Evaluation of methodology
- Evaluation of the software model
- A description of progress and evaluation of development and testing. There is one section for each category of applications, which are authored by the responsible partners plus an overall summary of the way forwards.
- Dissemination activities related to WP3

In addition, two appendices, which have some relevance to the application development, have been added to the document: An evaluation of the Kindle e-book reader in a DTN context, written by Avri Doria, LTU, and "Product design for N4C" – a paper submitted for a master thesis at Luleå University of Technology by Martin Bonnevier, master student at LTU.

CONTENT

FIGURES

1. INTRODUCTION

Work package 3 Pervasive Applications covers the majority of application developments in N4C that are closely connected to software development.

This document constitutes deliverable D3.2. It is an internal evaluation made by the WP3 partners.

In the DOW, D3.2 is described as follows:

Hiker's PDA: Some hardware and software platform possibilities will be evaluated with the needs of different user groups in mind. Results and advice from WP 5 will be incorporated and combined with general ICT considerations, such as choice of operating system and development platform. A few (1-3) hardware/software configurations will be chosen as platforms for applications development.

Web caching: The preferred path of implementation is to modify existing components such as web proxies that already have part of the functionality needed for the applications. In this task appropriate components, preferably with open source licensing conditions, will be evaluated as to functionality and design to determine whether and how they can be modified to support the required functionality of web caching.

Meteorological and data capture: Limited to detailed evaluations of components. The node will be based on an Intel 'Ultramobile device' development board based on the Low Power IA (Intel Architecture), running Linux and enabled for WiMax and 802.11n.

Based on the feedback given by the evaluators in review meeting no 1 in Luleå June 2009, we have extended the scope of the deliverable to also include a description of progress and future work related to application development.

1.1 THE PURPOSE OF THIS DOCUMENT

Deliverable D3.2 "Evaluation and Progress Report" is submitted at the mid term of the project. The purpose is to:

- Report the progress of application development
- Evaluate the results achieved so far
- Adjust and confirm the plans for the remaining project period

The N4C project will develop a limited number of demonstrators or prototypes of applications for a DTN environment. The selected applications are:

- "Hiker's PDA" (Applications for Hikers, Hunters, Herders and Rangers)
- Meteorological and Environmental Data Capture
- Web Caching and E-mail

This document contains the following sections:

- Evaluation of methodology
- Evaluation of the software model
- A description of progress and evaluation of development and testing. There is one section for each category of applications, which are authored by the responsible partners plus an overall summary of the way forwards.
- Dissemination activities related to WP3

2. METHODOLOGY

2.1 THE SPIRAL MODEL

At the technical kick-off meeting in Slovenia in project month 5 it was decided that a spiral development model shall be used to guide the routines for the interaction between work packages and the generations of solutions that are developed and exist in parallel during overlaps in development cycles.

The software development process has been described in the DOW as follows: "An iterative development process adapted to the overall progress of the project will be chosen. 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. The spiral model is a software development process combining elements of both design and prototyping-in-stages, in an effort to combine advantages of top-down and bottom-up concepts.

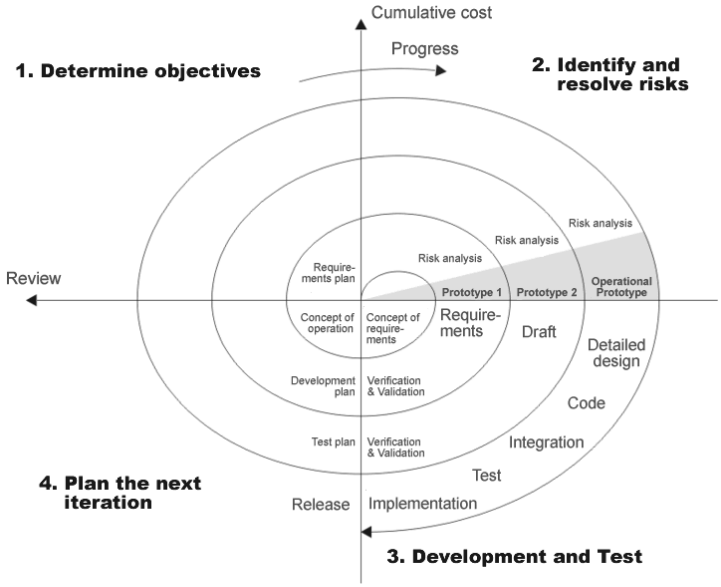

FIGURE 1: FROM WIKIPEDIA: SPIRAL MODEL (BOEHM, 1988).

The spiral model is intended for large, expensive and complicated projects; therefore we will not reach the final stages during the N4C project. The applications must be realistically possible to implement on the chosen platform, within the timeframe and resources available in N4C. A guiding principle will be to adopt existing open source applications, and with a minimum of changes, modify them to run over DTN.
According to the DOW, the requirements and functional specifications will be described in a "living document" and updated for internal use throughout the project, with internal delivery dates in month 18 and 24. The final version of this document and the rest of the WP3 documentation will be delivered in month 30 together with the final version of the prototypes (D3.3).

The first part of the "living document" was Deliverable D3.1 Functional Specification (Initial), due as deliverable D 3.1 in month 6. The second part was the resubmitted D3.1 Functional Specification, delivered in month 16. The third part is the present "living document" D3.2. All three documents are considered to be part of the spiral development cycles.

2.2 EVALUATION OF METHODOLOGY

Referring to the figure of the spiral model, we have reached the point of Verification & Validation of Prototype 2. This document is part of the validation, but the up-coming Winter Test 2010 and Summer Test 2010 are also important parts.

The model specifies that there is a risk analysis for each turn of the spiral.

A separate report on risk analysis for the project as a whole is documented in [N4C-RAR] Risk Analysis Mid Project Report. In our case some risk analysis has been dealt with in the D3.1 document and risk analysis also is implicit in the ongoing prototype software development. An explicit risk analysis for the summer test 2010 is planned in the project meeting in Tromsø in January 2010.

Our evaluation of using the spiral model for the software development process is that it has proved to be useful and productive. The model is, of course, quite general and it has the degrees of freedom that are required for a highly experimental project like N4C. We will therefore continue to follow the spiral model for the remaining project development. The spiral model is well suited to the project plan, where we have described three cycles of winter test/summer test during the project lifetime. An alternative development model could have been the Scrum model with agile software development that gives greater dynamics in the development. However, due to the fact that N4C is a rather complex project with many partner and interactions on many levels between work packages, it is vital to maintain control over the structure of the development process. The drawback of using the spiral model is that it can be rigid and complex when applied to this project. We therefore have made modifications to the spiral model and simplified parts of it.

3. PROGRESS AND EVALUATION OF THE SOFTWARE MODEL

The N4C application development started out in August 2008 with a Software model as shown in Figure 2. The original version had three tiers: Application layer, DTN layer and Link layer.

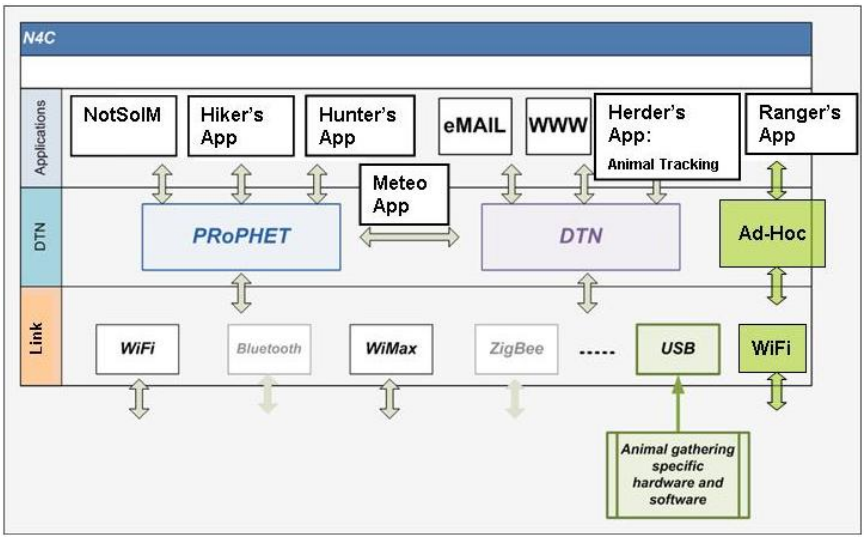

FIGURE 2: N4C SOFTWARE MODEL 2008

As development progressed during 2009, it became apparent that the model would be tidier if a middleware layer was introduced between the Application and DTN layers. Such a middleware layer would clean up some ambiguities concerning software entities appearing in two layers simultaneously, for instance, the code for handling web cache and the meteo application. As stated in D3.1 Functional Specification [N4C-D3.1], the N4C integrated networking architecture is intended to treat the conventional Internet infrastructure and DTN as peers rather than presenting DTN as an alternative link layer technology. This is appropriate because of the distinct communication paradigms used in the networks and is an important research issue because there are a number of other situations (such as mobile networks) where closely related problems are likely to arise in the near future:

- Routing across the boundary between the two types of infrastructure with multiple connection points
- Addressing for the DTN
- Mapping between the addressing schemes in the IP and DTN domains

Introduction of a middleware layer would simplify some of the diverse communication challenges, making it possible to handle a so-called Hybrid Synchronization Service plus the Auto Discovery.

The new Software architecture is shown in Figure 3. It contains four tiers:

- The Application layer, named Hiker's PDA, contains the end user applications.
- The Middleware layer provides the required bridging between applications on one side and DTN or local Ad-hoc networking on the other.
- The DTN layer provides communication to the Internet or to different CCRs.

- The Link layer manages the communication on the available physical technology, e.g. WiFi, WiMax and Bluetooth.

The implementation of the software platform is based on Python. The hybrid synchronization service in the middleware layer uses XML-RPC for synchronization of meta-data, and http for synchronization of user data. The auto-discovery function uses IPv6 and multicast.

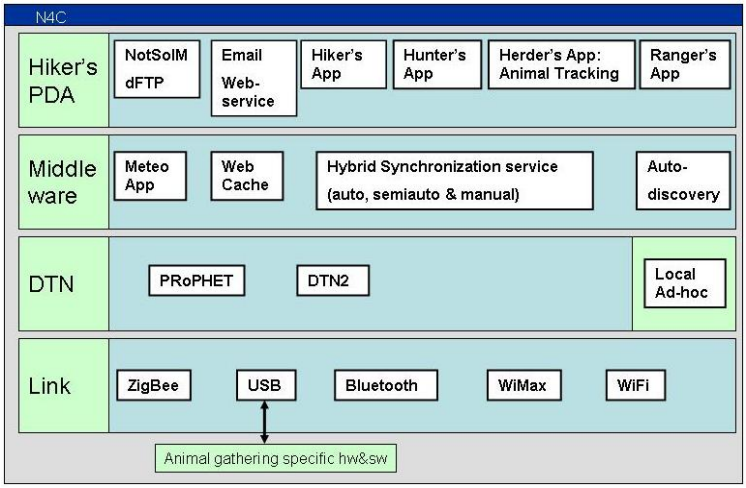

FIGURE 3: N4C SOFTWARE MODEL 2009

The justification for introducing a four-layer model came partly from writing a contribution to a book chapter on pervasive applications [N4C-PCA], and partly from the increased understanding of how the software components interacted. Related to the spiral model, we observe that the original three-layer model was used in the 'first turn' of the spiral and that the four-layer model came in the second turn.

We consider the present software model to be well balanced in the sense that it is sufficiently comprehensive and at the same time simple and easy to understand. In other words, the model is not too complex and it is easy for the developers to introduce new applications and functions in the system.

4. HIKER'S PDA

The Hiker's PDA is a portable handheld computer (PDA) in a Communications Challenged Region (CCR) equipped with a suite of rather simple and intuitive applications that aims to be useful tools for hikers, tourists and people that are doing any type of professional job in a CCR. The applications are based on user requirements collected from potential users over the last 3-4 years, which should guarantee that the applications are based on user needs.

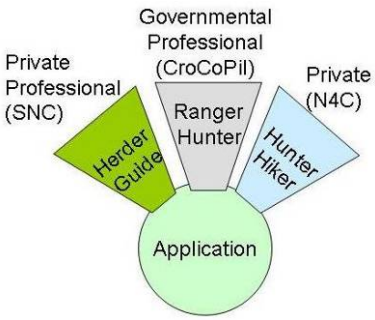

FIGURE 4: N4C USER GROUPS

The Hiker's Apps were designed with three kinds of user groups in mind:

1. Private professional users (business): Reindeer herders living and working in the Arctic area of Scandinavia and tourist guides who are working (and living?) in the Arctic area of Scandinavia.

2. Governmental professional users: Nature Park Rangers and the police working in the vast wilderness areas in the Arctic area of Scandinavia.

3. Private users: Tourists who are hiking in the Arctic area of Scandinavia.

We see that applying the spiral model in the development process has been fruitful for structuring the description of the user groups. In the first round, the users were perceived as a general group, based on three different user groups from three different projects. In the second round we have been able to structure the users according to the context each group operates within.

The applications and use cases that have been developed and tested are:

- **DTN based E-mail and Web caching** (described in separate sections)

- **Auto Discovery:** Auto Discovery using ad hoc Wi-Fi communication when any other Hiker's PDAs are within communication range.

- **Geoblog:** A simple blog application where the user can enter text, photos, GPS location and timestamp. There will be an automatic actualization of the Geoblog when there is a connection opportunity with other Hiker's PDAs.

- **Send Message with own location:** Multicasting a message with one's own location to other Hiker's PDAs in the CCR area. The first version is simple, without any interception criteria.

- **Maps for own location:** Downloading maps of the area around one's own location.

- **POI:** Download of Point Of Interest (POI). Extraction of GPS maps with location of interest points, from the web cache/internet. For instance, the location of nearest medical services (medical, physical or heart starter) can be one type of POI.

- **RSS feed in DTN:** (Optional) Application for RSS feed in DTN environment in the field.

The developers have also taken into consideration and have been working with issues of location privacy [EFF], security, battery life time, wireless range and ad hoc connections, all very important for the functionality and usefulness of the applications.

4.1 PROGRESS AND EVALUATION

N4C is an ongoing project with 3 development iterations, 2 of them completed. The second field tests have been completed during summer 2009. During the first two iterations automatic and manual synchronization have been implemented, none of these approaches are suitable to all the applications. Therefore a new improved version is being designed to support hybrid synchronization which can be customized to the specific needs of the different applications. Additionally, the next version will include improvements in the user interfaces and the integration of new DTN technologies (e.g. DTN2 and/or PRoPHET) [N4C-FIRE].

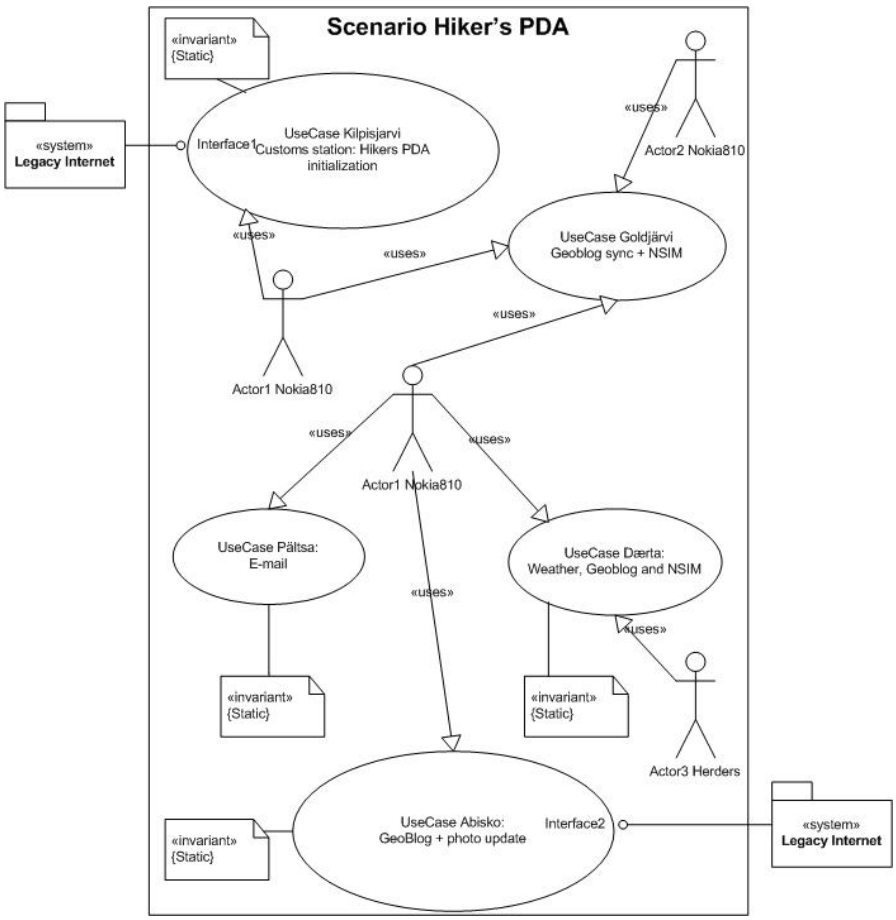

FIGURE 5: HIKER'S PDA SCENARIO

The use cases for the Hiker's applications scenario shown in Figure 5 are described in [N4C-HPU]. Use Cases Goldjärvi, Dærta and Abisko are all elaborated during the Norut Summer test 2009, see [N4C-VST].

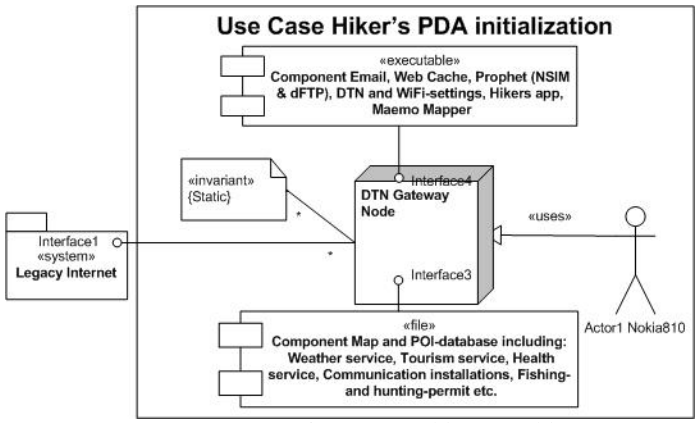

FIGURE 6: HIKER'S PDA INITIALIZATION

The use case for the initialization of the Hiker's PDA is shown in Figure 6.

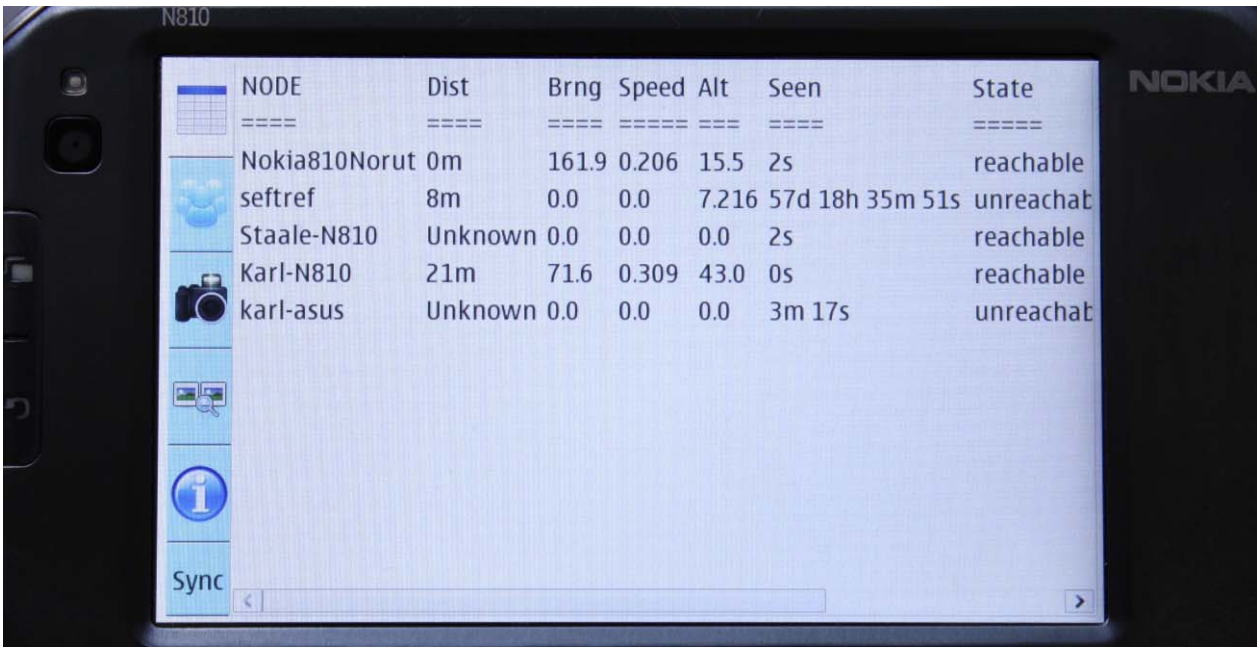

FIGURE 7: NOKIA N810 WITH HIKER'S APP [N4C-FIA]

Figure 7 shows the user interfaces for the N4C Hiker's PDA (Auto Discovery and Photo Sync) application[1] developed for the summer and winter test 2009 [N4C-VST, N4C-M8.2]. The user interfaces is written in HTML, and the application is written in python. The application was tested on Nokia N810 and Asus Eee PC, according to the N4C Plan for winter test 2009 [N4C-WTP].

Applications have been developed on the following **platforms**:

Nokia N810 tablet computers with Maemo, which is a computer architecture platform, built on desktop open source components. The platform is based on the GNU/Linux operating system and the Hildon desktop.

[1] http://trac.itek.norut.no/n4c/wiki

Asus Eee PC with GNU Linux Operating system (Xandros distribution). In particular the fully solid state versions of the Asus Eee PC are well suited because of light weight and low battery consumption.

Other platform candidates are:

HTC Hero, Magic (Google 1 Phone) with Android™ operating system. Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack. http://www.htc.com/www/product/g1/specification.html

The reason for choosing these platforms is that they satisfy the requirements listed in [N4C-D3.1] Functional Specification.

The machines are used as mobile, nomadic hosts that will typically accompany human users moving about in the CCR at large, rather than limited to the village areas where they could have Wi-Fi access to a village gateway.

The machines are compact, lightweight and battery powered with Wi-Fi communications capability and with a keyboard and visual display to provide a human user interface. Also, the platforms are open source with readily available developer's tools and the DTN platform is available on Linux.

From an operating point of view, a practical reason for choosing the platform was that both hardware and development environment are well known by the partners in N4C.

The hardware and software platforms and the development results are evaluated in section 2.x. below.

4.2 TESTS

4.2.1 Hiker's PDA Winter test 2009

Preparation

A detailed test plan was developed before the test was executed [N4C-WTP]. The plan document covered testing objectives, testing strategies, description of experiments and conclusions on what expected outcome of the test should be. The plan had great impact both on the winter test itself and later work, since it represented a systematic and comprehensive approach to what should be done. The test itself was therefore executed with efficiency and the results were of good quality.

Test in Jokkmokk

Part one of the Hiker's PDA tests was executed in Jokkmokk, in parallel with other N4C-tests. The weather was quite good; sunny, no wind and temperatures between –4 and –10 °C. This made the test execution quite simple seen from an operational view.

Part one of the test was focused upon testing the Auto Discovery functionality. This was done by using three different Nokia N810 and one Asus Eee PC, so that they were moved apart each other in a strictly controlled way (10 meters, 20 meters, 30 meters ...). Shooting a picture with each of the PDAs produced the actual sync files. The photo files were subsequently automatically synchronized between the devices.

The test was successfully executed and the results are documented in [N4C-M8.2]. During the tests, data logs were stored on each PDA in order to be analyzed after returning to the office. The conclusions are given in the test report. Due to limitations of available time and test facilities, the tests had a quite "static" approach. When the devices were continuously moved with some speed it was difficult to test the Auto Discovery functions

One conclusion from the test was therefore that the Auto Discovery had to be tested with two or more PDAs that were moving relative to each other with some speed.

Test in Kroken, Tromsø

Part two of the test was executed at the ski-slope Kroken, just outside Tromsø, also under nice winter conditions

One ASUS Eee was placed at a fixed location, while a skier who passed the fixed-point in quite high speed (30-40 km/h) carried one PDA. In this way it was possible to test Auto Discovery between moving devices. The test result was not completely successful. Due to some malfunctioning data logging software, it was not possible to get a complete picture of how a very limited time window for communication functioned together with initialisation of data transmission. Still, it was possible to synchronize devices and to transfer data. The results are documented in [N4C-WTS].

A conclusion from the winter tests can be given as follows:

Synchronization between ASUS and Nokia N810 works quite well within a distance of up to 150 - 200 meter between devices. With several devices placed "in a row" it will therefore be possible to transmit data between handheld devices over quite a long distance.

Data transfer rate is depending on the fact that big files make TCP more efficient, obtaining better transfer rates with an average of 900 Kbps. As commented in the winter test 09 report [N4C-M8.2], we observed certain inefficiency of the TCP protocol when transmitting small size files, probably because the “overhead” is significant in comparison with the size total size transmitted. In our context the smallest files had a size <= 200 Kbytes and the biggest files had a size around 14 Mbytes.

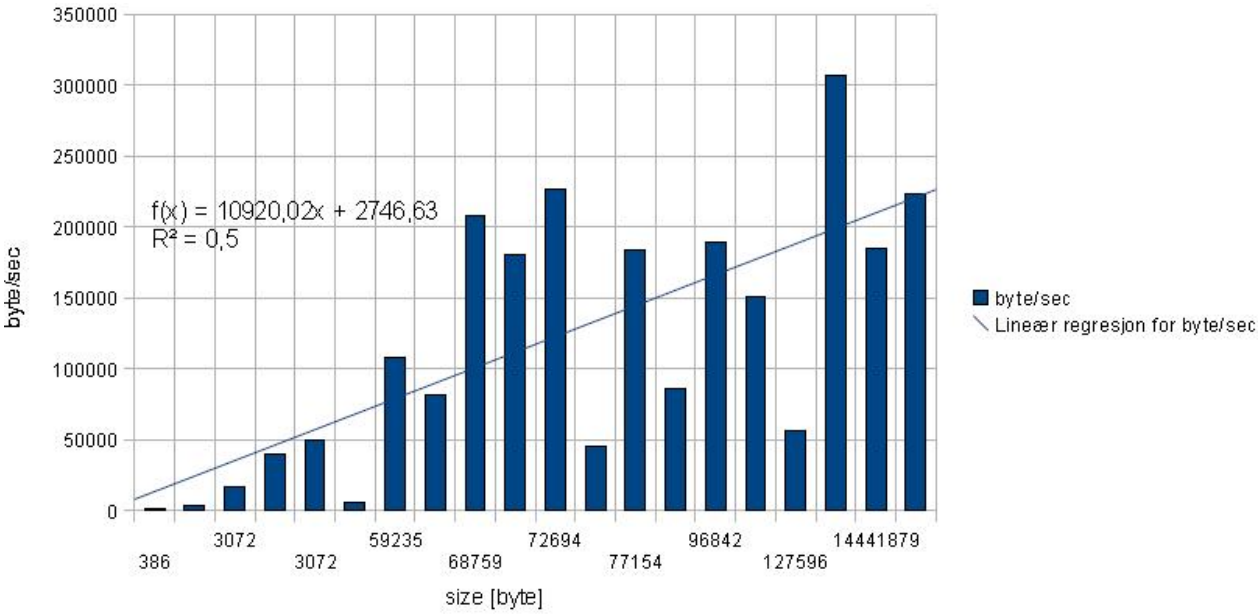

FIGURE 8: LOG OF FILE SYNC WITH BYTE PER SECOND VS FILE SIZE IN BYTES

The technology for ad-hoc wireless transmission seems to be very promising. However, at this point of time it will be difficult to establish a fully operational network based on this concept since the distance between devices is relatively low. To cover a CCR of any reasonable size would require quite a number of devices out in the field, in our case you would need one for each 200 meters.

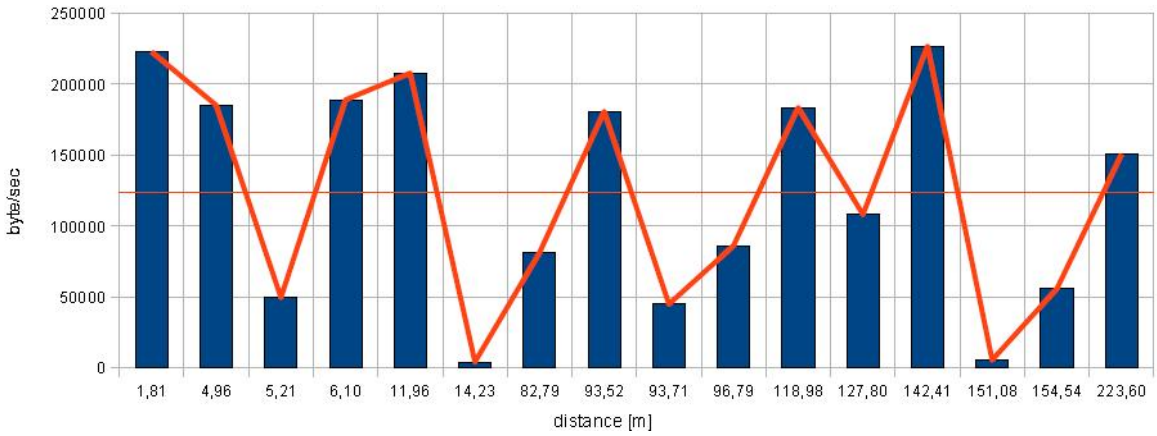

FIGURE 9: LOG OF FILE SYNC WITH BYTE PER SECOND VS DISTANCE BETWEEN NOKIA N810

4.2.2 Hiker's PDA Summer test 2009

Preparation

A detailed test plan was developed before the test was executed [N4C-STP]. The plan document covered testing objectives, testing strategies, description of experiments and conclusions on what expected outcome of the test should be. As we have stated above for the winter test, having a well structured test plan is instrumental for achieving good results, and the test itself was therefore executed with quality and efficiency.

Test background and overview

The Hiker's PDA summer test 2009 took place in Ytre Fiskelausvatn, Norway, from 20th to 22nd of July. This particular location was selected because

- There was no mobile phone or wireless Internet connectivity
- There was access to a power generator for charging batteries
- It was accessible by car

By the mid-July 2009 the development of the Hiker's PDA was in its second stage. In this period we concentrated on improving the synchronizing capabilities of PDAs and gaining experience with the possibilities and limitations when using office type computers under field conditions.

The test plan is described in detail in [N4C-STP], and an overview is shown in Figure 10.

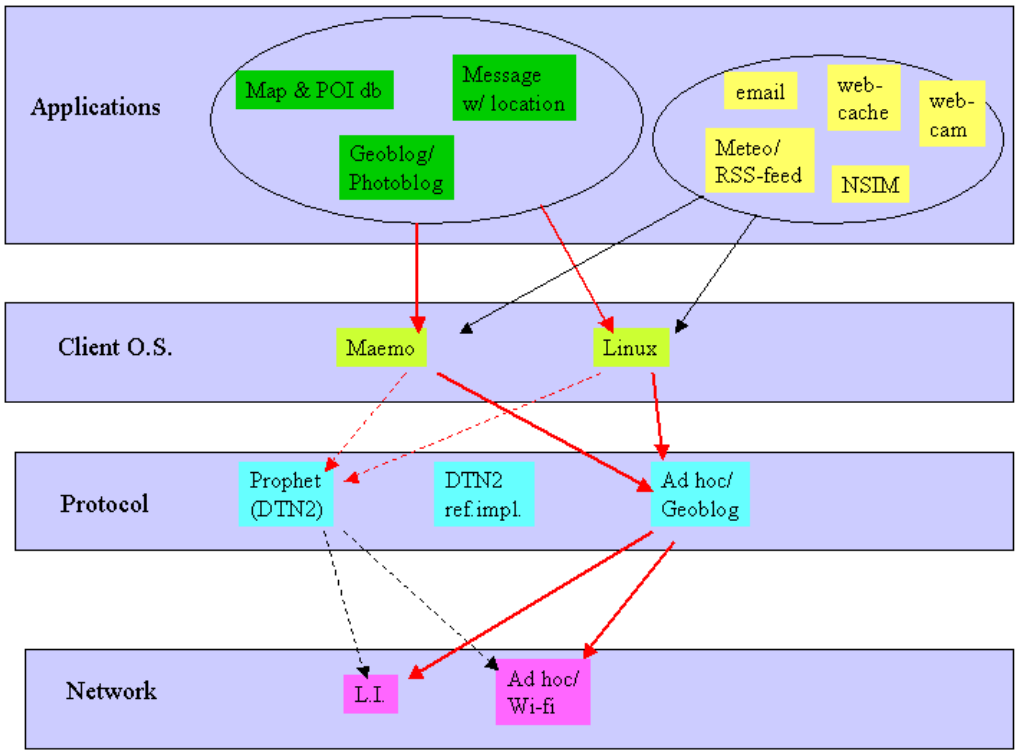

FIGURE 10: HIKER'S PDA SUMMER TEST 2009

The hardware used for the tests was unmodified off-the-shelf units designed for an office environment, and we wished to test how the components would behave in a realistic field environment. The tests were done with Nokia 810, Asus Eee PC 900 and iPhone. The type of questions we investigated were:

- how does the application components behave under field conditions,
- what is the maximum distance of communication with standard Wi-Fi cards,
- what is the performance of the ad-hoc p2p software for communication between units.

Although a bit premature, we also tried to make a separate test with parts of the DTN2 software. This was done by using Telenor Mobile EDGE (Enhanced Data rates for GSM Evolution, 100-200 Kbit/sec) near the location Vesterli, Ytre Fiskelausvatn.

Software Components in the Test

Auto Discovery: Test Auto Discovery using ad hoc Wi-Fi communication in the field

Send Message with own location: Test the process of multicasting a message with one's own location to other hiker's PDAs in the CCR area. The present version will be very simple, without any interception criteria.

Maps for own location: Test the process of downloading maps of the area around one's own location.

POI: Test the download of Point Of Interest (POI). This test case tests the application in terms of extraction of GPS maps with location of interest points, from the web cache/internet. For instance, the location of nearest medical services (medical, physical or heart starter) can be one type of POI.

Geoblog – Photo blog: Test the automatic actualization of geoblog and photo blog when there is a connection opportunity.

RSS feed in DTN: (Optional) Test of software for RSS feed in DTN environment in the field. The test assumes that the RSS DTN functionality is operational.

NSIM, DFTP: Test NSIM and DFTP with a public Wi-Fi Internet access point at the Petrol station in Nordkjosbotn or using Telenor Mobile EDGE. In particular, we tested transfer of Geoblog/Photoblog messages to Legacy Internet by using DFTP.

The test results are documented in [N4C-VST].

4.3 CONCLUSIONS

The evaluation of the methodology and software model and components are described in sections 1.2.1 and 1.2.2.

The tests are a systematic way of evaluating the results of the work performed in the application development in WP3. The detailed test results are given in the test reports, [N4C-M8.2] and [N4C-VST]. In this section we sum up the conclusions from these reports.

4.3.1 Evaluation of the test plans

The winter test plan [N4C-WTP] described the test procedures in detail. At this stage of development, this level of detail was appropriate, and the plan was complete, in the sense that it covered all aspects that were possible to test at the time.

The summer test plan [N4C-STP] was partly a repetition of the winter test plan regarding the testing of Wi-Fi range. Summer conditions are different from winter conditions, but the summer test was too detailed. The consequence was that there was not enough focus for testing of random movements and encounters.

4.3.2 Evaluation of winter and summer tests

The tests were performed according to the test plans [N4C-WTP] and [N4C-STP].

Due to the high level of details in both test plans, the time left for in-field bug fixing and adjustments, was somewhat restricted in both cases. However, the tests resulted in valuable data including comprehensive log files, and gave answers to most of the questions we had raised in the plans.

The main conclusions from the tests are:

- The maximum distance between two devices is still relatively low. The winter test showed up to 180 m, the summer test showed well above 200 m. The winter test was done in rather dense spruce forest, while the summer test was done in birch forest. In both cases we tried to maintain the line-of-sight, and maximum distance was measured under such conditions.

- Battery capacity for both the Nokia N810 and the old version of the Asus Eee (without Atom processor) is a serious concern. Batteries need to be recharged after 3-4 hours operation. We assume that technology development in general will result in lower power consumption in the handheld devices and increase battery capacity in the near future. The possibility for recharging under field conditions is a separate concern that is addressed throughout the N4C project. Lightweight (or, at least, transportable) technology for recharging is becoming commercially available, for instance solar panels, hand powered mechanical chargers, thermo-electrical chargers.

- Both technically and functionally PDAs can be used for communication in field, but the present technology has clear limitations and must be improved. After the start of the N4C project in May 2008 we have experienced a rapid development of new generations of mobile phones that may replace the present generation of PDAs. The new class of mobile phones satisfies most of the requirements we have stated for a Hiker's PDA: high quality screen that works well in sunshine, long battery lifetime, high computing power, very large storage

capacity, integrated GPS and Wi-Fi. In addition we see that many of them come with an open platform for software development, e.g. Android.

- Present Off-the-shelf technology may operate in field in a satisfactory way. However, more robust equipment with an improved and simplified user interface is desirable.

4.3.3 Evaluation of the software modules

Referring to the spiral development model, all the tested software modules are specified and implemented as prototype version 2, in the second iteration of the model.

Auto Discovery: Test of Auto Discovery using ad hoc Wi-Fi communication in the field. The module worked according to specifications. It uses IP v.6 and multicasting, which is not available in DTN at the present. A 'hybrid' solution running IP v.6 for Auto Discovery and IP v.4 for DTN is probably the best way forward.

Send Message with own location: Test of multicasting a message with one's own location to other hiker's PDAs in the CCR area. The module worked according to specifications. The problem of flooding of messages (repeated transfers) is handled by using a 'synchronisation service' in the middleware layer.

Maps for own location: Test of sharing map database (typically of the area around one's own location). The module worked according to specifications.

POI: Test of sharing of Point Of Interest (POI). This tests the use case of extraction of GPS co-ordinates with interest points derived from other hikers or web cache/internet. For instance, the location of nearest medical services (medical, physical or heart starter) can be one type of POI. The module worked according to specifications.

Geoblog – Photo blog: Test of geoblog and photo blog (sharing of textual information, annotated with photos, GPS co-ordinates, timestamp, etc) when there is a connection opportunity. In the winter test the activation of the synchronisation was done automatically without user intervention. In the summer test, the users activated the geoblog manually. The module worked according to specifications.

RSS feed in DTN: (Optional) Test of software for RSS feed in DTN environment in the field. The test assumed that the RSS DTN functionality was operational. At the time of the test, the RSS DTN functionality was not available, hence the test was not performed.

NSIM, DFTP: Test NSIM and DFTP with a public Wi-Fi Internet access point. In particular, we tested transfer of Geoblog/Photoblog messages to Legacy Internet by using DFTP. The test did not succeed when we tried to connect via a public Internet access point at a petrol station. The reason was not any code failure in our module, but lack of Linux support at the access point (only supported Internet Explorer from Microsoft). We repeated the test in the Norut office using the Telenor mobile Internet connection (EDGE). The module then worked according to specifications.

4.4 FUTURE DEVELOPMENT

N4C is an ongoing project with three development iterations, with two of them now completed. The second field tests have been completed during summer 2009. According to the spiral model we have completed iterations 1 and 2 where prototype 1 and prototype 2 have been developed and tested. At the beginning of 2010 we are entering the phase where we make an operational prototype based on the original plan and the lessons learned from development & testing so far. The operational prototype is planned to be ready for testing in the summer test 2010.

The main activity that remains is to integrate the different functions/modules into a unified and consistent application, which has an intuitive user interface (GUI) and is adequately robust to be operated in a field environment.

During the first two iterations automatic and manual synchronization have been implemented, none of these approaches are suitable to all the applications. Therefore a new improved version is being designed to support hybrid synchronization that can be customized to the specific needs of the different applications. Additionally, the next version will include the integration of new DTN technologies (e.g. DTN2 and/or PRoPHET).

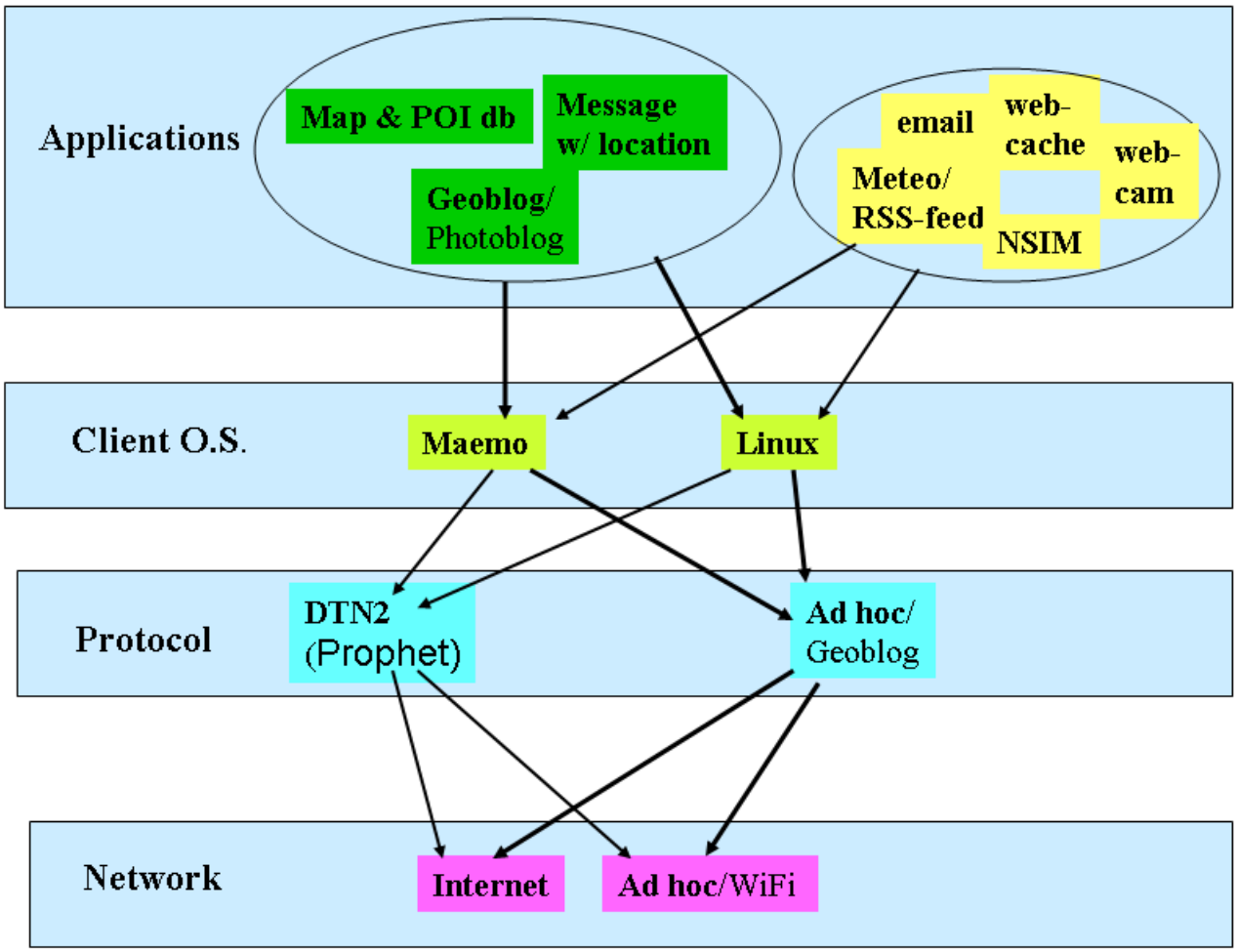

FIGURE 11: N4C APPLICATIONS SUMMER TEST 2010

5. WEB CACHING AND WEB SERVICES

5.1 WEB CACHING

Web caching for DTN is a service that provides users in remote areas access to web based information that normally would not be possible. The N4C aim of web caching is:

- Providing users of DTN at least limited access to information stored on the web, and making this as transparent as possible to the user.
- Building a basic service that can be used in by anyone interested to implement it to their DTN.
- It should be made up of free open source applications such as web proxies, servers that can be modified and controlled by using script languages.
- Expandability is important for the service to be useful in the long run. That could mean implementing new functionality and scenarios, handle certain types of web sites and supporting a growing number of users.

To have a future user/developer community that could have the opportunity to improve the existing code as much as possible for the web caching and web services.

5.1.1 From SNC to N4C

Already 2006 in the SNC project there was a functional web caching service. It was based on free applications and scripts, on both server and receiver. It was a one-way web forwarding that made it possible to view web pages "offline" on a computer in the mountains. Today the same technical application and scripts are used, modified to work with N4C hardware and with extended tests up in the northern region of Sweden in the Sapmi region, and possibly limited tests in Slovenia in the Kocevje region.

5.1.2 Components of web caching

N4C uses free and available open source applications and scripts on all hardware that we use for web caching.

Wwwoffle (server) – an application installed on a server connected to internet that N4C developers has control over. It fetches data from web sites and store it a special location on the N4C server.

Wwwoffle also acts as a web proxy on a client computer existing in the DTN realm, serving the cached web pages to the user.

Web caching script (server) - Packs and send the web data from the server to the DTN node at the border, where "data mules" then can pick up the data and later leave it at the destination.

Web unpack script (client) – Unpacks web data to a location on the receiver computer.

5.1.3 Expandability

With the DTN progress and user demand, web caching software used in N4C project allows for increasing functionality. As web caching is merely applications using the DTN as a transport, it will be possible to develop and expand for new cached web functionality for users as the network mature and grows. Web services development might a problem in cases where the applications connection has so much delay, that the time delay gets too long and data is no longer relevant. Developers for DTN web caching could make "plug-ins" that tells the application software when time-delay is an issue, so the application can behave differently depending on location. This could enhance the transparency for the users going to/from communication challenged regions.

5.1.4 Evaluation of web caching service

Through software developing and modification, using free software such as Wwwoffle and bash scripts, work in WP3 has resulted resulting in a working web cache service for within N4C. The web cache theoretical part has also been developed resulting in possible web caching scenarios that has been recognized and explained in Deliverable 3.1 – Functional specification [N4C-D3.1]. These scenarios will be used as reference when improving the web cache service, expanding it with new functionality. User end part of web caching has also progressed with the Web GUI, because it provides users a way to find, use and evaluate the application during field tests.

In latest test, summer 2009, web caching was tried out with users in Staloluokta, Ritsem and in Saltoluokta. Two Asus EEE laptops acted like client-computers and were available to normal users (hikers, residents and workers) for a limited time.

5.2 PODCASTING

Podcasting is a "proof of concept", but useful web service made by N4C LTU. It is providing users of the service with an ad hoc "radio" experience. The idea is to broaden the possibilities with DTN and get closer to the living labs experience by providing people with useful services:

- Enabling radio shows/podcasts being sent using DTN to remote areas.
- Expanding the "web caching" scenario from SNC to "web caching and services" in N4C – making not only web pages available, but also podcasts.

Using the techniques and free software similar to the web caching application.

5.2.1 Components

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

Gpodder – a free software for linux systems that downloads podcasts (audio or video files) whenever new ones appear on the web. This application exists on a stable normal internet-connection.

DTN podcast script – an application made by LTU for N4C, which controls the gPodder application and forwards new podcasts to the DTN. This application is installed to the same server as Gpodder

On the client side, in a CCR, the podcasts is shown to the users by the web GUI. Playback is then possible by using a free software MP3/audio decoder – usually included in operating systems.

5.2.2 Expandability

The podcast service is one of many typical services that could be used within DTN. This particular service exist only with the "push" model, which means that predetermined podcast shows are being sent to the DTN x number of times per day. This type of service, also web caching, can be expanded to include even other scenarios like handling user requests. This could mean that a user existing on a DTN can connect to the web GUI with a web browser and tell the application what he/she wants to see or listen to.

5.2.3 Evaluation of podcasting web service

The point of the application has been to expand the web service concept, showing the possibilities with DTN applications and giving birth to further ideas. Podcasting was chosen because it is relatively easy to handle within DTN, it is free to use and can be used on most platforms running with free software only.

Podcasting was used in summer test 2009. Audio files from from "Samiska nyheter" (Sapmi news) and other "testing" podcasts were sent to the CCR and our Asus EEE machines so it could be tested by the users.

5.3 WEB GUI

When a DTN user wants to use the web services, the user need to be able to find it and use is properly, without having to a computer expert. It needs to provide high usability and bring satisfactory response to the user. They will be greeted by a "homepage" that will help the user as much as possible, connecting the user to the different web based services. Aim for the Web GUI is:

- Providing a "home" for DTN users using the web services like caching, podcast and more.
- Installable on a "village router" or an end client computer, depending on DTN setup.
- Collecting services or links to services in one place for easy access.
- Clear design make it is easy to use and expandable.

FIGURE 12: WEB GUI CAPTURE

5.3.1 Components

Web GUI is basically a web page build with XHTML and PHP scripts and can exist on any N4C DTN user computer, or be served from a central host. It may require user installation of a web server and PHP engine for using the service on a computer with DTN software. For connecting to the GUI one could be using own computer such as laptop or PDA and connect to a village wireless router. From there the Web GUI will be served and present itself with the user.

5.3.2 Expandability

When new services are developed, they can easily be added to work with the Web GUI. The platform independent XHTML and PHP language purpose in this application is to serve and show web services and caching. It can therefore be used for controlling what kind of functions the user can access, or hiding unnecessary complexity. For users with mobile devices, web GUI can be built with added support for smaller screens with smaller resolutions, following mobile web – best practices, and mobile web developing guidelines.

5.3.3 Evaluation of Web GUI

Web GUI exists only in its most basic form since summer 2009, providing the users in the last test access to services like web cache, podcasts, webcam images and temperatures. It is a natural step to implement some kind user interface, which will then grow and will be improved upon when more users start testing DTN. The plan is that this will happen working with the users, participants and trough field tests finding out what is the best way to provide and improve the web services. For example: If a user feels something is missing in the GUI or in his/her experience with it, it could mean that ideas for new services or functions come up. This could have a positive effect on any of the applications connected with the GUI.

6. METEOROLOGICAL DATA AND WILD-LIFE PICTURES CAPTURE: CHALLENGES IN THE LATEST ITERATION

6.1 CALIBRATION AND SYNCHRONIZATION OF CLOCKS

6.1.1 Introduction

Synchronization of the clocks is one of the most important issues in the environmental measurements. Each acquired measurement must have unique date/time stamp. This is especially important for long-term assessment of climate changes where the data evaluation is based on the data comparison on the time scale.

Some embedded and personal computers have integrated better clocks (quartz) that are precise up to several seconds per year while other can have relatively big errors up to 15 minutes per day. At boot-up procedure the clock of the operating system is initialized. If a hardware clock exists the operating system uses it to set its time and then forgets about this hardware clock. If there is no hardware clock available the operating system clock is set to some predefined time (i.e. 01.01.1970 00:00:00). After this initialization the precision of the clock is based just on the stability of the processor's frequency. Usually this processor's frequency is relatively very stable but it can be slightly different on the two equal computers from the same manufacturer. So to achieve best possible precision of the clock some fine tuning (calibration) of the clock must be made.

The problem of calibration and synchronization of computer's clocks is very well addressed and described in the "Network Time Synchronization Research Project" (http://www.eecis.udel.edu/~mills/ntp.html). The access to open source software is available at "Network Time Protocol project" (http://www.ntp.org) which is mostly targeted at Linux platforms. There exists also port of this software on the Windows platform (http://www.meinberg.de/english/sw/ntp.htm).

6.1.2 Procedure to synchronize and calibrate the clock

1. Step

Disable or kill all services that can change/modify system clock such as ntpclient (can be placed also in the folder /etc/hotplug.d). Disable also any scheduled processes that are defined to modify/change system clock (like scripts in crontab, terMit) that are invoking hwclock.

2. Step

Enable good internet connection with a relatively precise time server. It is even better if there is some local precise time server available. Set the correct time zone.

3. Step

Check your current system clock settings using adjtimex command. Read and save the settings tick and freq (on Cambria boards there are usually default values if tick=10000 and freq=0).

4. Step

Record the comparison of computers system clock with time server for 20 minutes in intervals of 1 minute. At the end of recording there should be 20 samples available in the text file (i.e. /MEIS/time_calibration_rough.txt) saved on the storage device. Command for recording is following:

ntpclient -c 20 -i 60 -h goodtime.ijs.si >/MEIS/time_calibration_rough.txt.

5. Step

Calculate following settings for parameters tick and freq by following rule described in adjtimex manual:

"If your system clock gained 8 seconds in 24 hours, you could set the

tick to 9999, and then it would lose 0.64 seconds a day (that is, 1

tick unit = 8.64 seconds per day). To correct the rest of the error,

you could set the frequency offset to (2^16)*0.64/.0864 = 485452."

6. Step

Readjust new settings for parameters tick and freq by following command:

adjtimex -t calculatedTick -f calculatedFreq

7. Step

Repeat 4. step for longer interval at least 24 hours long:

ntpclient -c 1440 -i 60 -h goodtime.ijs.si >/MEIS/time_calibration_fine.txt

8. Step

Calculate following settings for parameters tick and freq by following rule described in adjtimex manual

"If your system clock gained 8 seconds in 24 hours, you could set the

tick to 9999, and then it would lose 0.64 seconds a day (that is, 1

tick unit = 8.64 seconds per day). To correct the rest of the error,

you could set the frequency offset to (2^16)*0.64/.0864 = 485452."

9. Step

Add the adjtimex command in the boot-up script (i.e. /etc/init.d/ntpd). Be careful that some other script or service does not overwrite your new settings. The adjtimex command is available in the package ntpclient.

adjtimex -t calculatedTick -f calculatedFreq

10. Step

Enable time adjusting service (ntpd or ntpclient).

11. Step

Create a scheduled script that will be periodically executed every n-minutes to readjust the real/hardware clock from the system time.

hwclock -w

12. Step (optional for the most precise clock)

Install and configure the ntpd deamon/service to make fine adjustments of the system clock.

ntp.conf example:

this command enables access to our server from local machine

restrict 127.0.0.1

#

the server from which the clock is sync does not have permission to adjust settings

of our deamon and cannot make time queries

#

server goodtime.ijs.si iburst

restrict goodtime.ijs.si mask 255.255.255.255 nomodify notrap noquery

#

server atom.uhr.de iburst

restrict atom.uhr.de mask 255.255.255.255 nomodify notrap noquery

#

#

the second serves points to itself, so that the deamon can use it own clock for

syncronization until it meets again better server

server 127.127.1.0 iburst

fudge 127.127.1.0 stratum 10

#

#

```
# the file on the local system where the clock drift is saved
driftfile  /MEIS/ntp.drift
#
# the computer from N4C network can sync on the our time server
restrict 192.168.2.0 mask 255.255.255.0 nomodify notrap
```

IMPORTANT: if the ntpd is enabled, disable the ntpclient service to avoid strange behaviour of the operating system.

6.1.3 Conclusion

Described procedure of synchronization and calibration of the computer's clocks will be performed on all nodes for the following winter test 2010. During the permanently running tests in Slovenia it was determined that the clock drifts on all used nodes is too high for environmental measurements. So far the problem has been solved by using the synchronizator to hardware clocks that are available at nodes. The described method will improve the precision of the clocks because the operating system clock is more precise than hardware clock. The final results will be described in the following milestone reports and deliverables. If some new problems will occur they will be reported with possible solutions.

6.2 COMPACT FLASH CARDS

6.2.1 Introduction

In DTN networking storage of data is very important. Sent bundles from the source node must wait on intermediate nodes until they reach the destination node. On intermediate nodes are delete after the reception has been acknowledged or they TTL (time to live) expires. Providing certain bandwidth in DTN network requires also some minimum storage capacities of the nodes. In our test Compact Flash cards and USB pen drives had been selected as storage media because they consume relatively small energy, they are designed to work in extreme temperate conditions (from -20 deg. C to +70 deg. C) and are supported in most of embedded computers.

During the last summer test 2009 some problems occurred with used storage media. The data on two storage media had been lost: one CF on one of the Cambria nodes and one USB pen drive used in ASUS WL-500W wireless router. Recovery of the data has not been successful.

Another issue that also appeared was read and write data transfer rate to storage media. On Asus router only USB 1.1 transfer protocol has been supported so the transfer rate was relatively very low. The node was useful only for exchange of relatively small bundles (up to 300 KiB). It was also performing very badly when a large number of small bundles arrived. Similar problem with large number of small bundles occurred on Cambria nodes where the transfer could take also up to 2 hours.

6.2.2 Currently available solutions

During the recovery process of destroyed storage media it was determined that it would be more suitable to use ext3 file system instead of ext2. The newer ext3 file system has integrated journaling file system and it is more resistant to sudden power failures. Also the recovery of data from ext3 file system can be much more successful. The write and read transfer performances of ext3 file system are slower but not so significantly to have important effect on the system.

Improvement of transfer of large number of bundles during the exchange could be improved by upgrading to latest version of software were some changes has been made according to authors. The authors of the software were also advised to improve the exchange of large number of small bundles by optimising this exchange: several smaller bundles could be wrapped into on bigger bundle and sent in one step.

Transfer strongly depends also on the speed of storage media. On simple off-the-shelf routers (i.e. Asus WL-500W) the speed of data transfer to the USB pen drives could be improved by upgrading OpenWrt from currently used kernel 2.4 to latest kernel 2.6. This could be done when next stable release of OpenWrt will be available, which will also include support of Broadcom's chipset.

On the Cambria node the speed of data transfer to CompactFlash cards could be increased only by using additional SATA minPCI interfaces. Contact to the manufacturer Gateworks company confirmed the limitation of IXP expansion bus that allows speeds only up to 4MB/s.

The results of the extensive tests of different storage media will be reported in next winter test 2010 milestone report.

6.3 POWER MANAGEMENT ISSUES

6.3.1 Introduction

Low power embedded computers are currently consuming at least 5W without any external devices. This power consumption covers the processor, USB controller, Ethernet and IDE controller. The latest versions of Intels Atom processors have also managed to reach this consumption or even better. For the 5W of energy consumption we can get currently best computational performances and also video controller. Atom platform presents currently one of the best solutions especially when more embedded computers from different manufacturers of motherboards will be available on the market.

But for extreme conditions where special requirements are defined this level of power consumption (5W) is still too high. There is missing a platform with lower computational performances that would have at least 10 times lower power consumption. That kind of platform should also keep all the functionality except the video controller which is not important for internet routers and environmental equipment. For a decade lower power consumption is need in extreme environment where standard power lines are not available and only power from solar panels or wind generators is available for a relatively short period of time. This situation is not present only in the Lapland where in winter time there is almost no sun to derive power from, but also in Slovenia. Most of environmental stations in Slovenia are placed in locations that are following World Meteorological Organization (WMO) standards. These locations are usually away from urban environment at locations that represent the weather conditions for wider area. At the best possible locations the power lines are not available. Because in most parts of Slovenia (also in Kočevje region) is present

complex terrain (mountains, hills and deep valleys, basins) many stations are located also on the bottom of valleys and basins. During the winter time the thermal inversions can last for several weeks. These thermal inversions also block most of the sun light during these periods and valleys and basins are foggy for weeks. So the usage of solar panels in these periods can also cause power failures.

6.3.2 Currently available solutions

According to the available technology at the moment one of the possible solutions to save power would be a development of advanced power management system. Most of environmental parameters are not changing very quickly except of the wind. The solution would be the usage of simple data loggers that are based on really low power microcontrollers and are already available. These microcontrollers computational performances are relatively small but enough to log measured data. This logged data could be processed in certain periods (every few hours) by existing embedded computers that have much higher computational performances. After the data processing the embedded computer would go back to sleep until the next processing of data. To realize this idea a special power management system should be developed. Currently such system is not commercially available. Special power management system would be based on a really low power microcontroller that would turn on the embedded computer in certain periods. When the embedded computer would finish the data processing from data loggers it would return to sleep.

There is also missing a solution to save energy during the data transfer between DTN nodes. Usually DTN nodes (especially mobile nodes) are in waiting mode which means that they are waiting to meet other DTN nodes. To keep track of the presence of another node an embedded computer must be turned on and WiFi device must be listening and also emitting wireless data packets to inform about it's own presence. This procedure is very power consuming and there it has not been reported that this task could be made with less that 10W power consumption. Again possible solution would be a creation of special power management system that would keep track for another DTN with some less power consuming wireless technology that WiFi. When the presence of another node would be found the embedded computer would be turned on by this low power management system and automatically turned off when data exchange between nodes would finish. An investigation of such technology should be made in following part of project.

6.4 UNSTABLE AD-HOC CONFIGURATIONS

6.4.1 Introduction

Currently most spread technology for wireless data communications is WiFi. It's possible rival technology in the future could be the WiMax which enables much higher rates of data transfer because of better usage of available bandwidth and more synchronized data packets exchange algorithm. Within the WiFi technology used nowadays the most common network structure is so called AccessPoint. In AccessPoint structure there exists one central wireless node (Access Point router) that router the wireless data traffic between the clients that are currently connected and optionally to standard Ethernet network. The device drivers for wireless network cards are supporting this kind of network structure much more reliably than the Ad-Hoc network structure. In the Ad-Hoc wireless network structure all devices are acting as peers. Unfortunately this kind of network structure is very rare in common practice which is the main reason for many bugs in device drivers and poor support from the device drivers manufacturers. There are even some manufacturers that only partially support the AdHoc mode to cut down the development price of the wireless network card. The AdHoc mode is poorly supported just to the level that barely satisfies the WiFi standard. The usage of

AdHoc mode for the mesh networking will be limited for a while until some WiFi standards will be followed more strictly by the manufacturer.

6.4.2 Currently available solutions

During the N4C tests we keep track of available wireless network devices and their support for AdHoc mode in milestone reports. During the tests some problems also appeared with devices that proved to be performing well in the past. Some bugs had appeared in new versions of the operating system OpenWrt. This discourages us from using the AdHoc mode in the future. If the problems remain unsolved during the duration of N4C, the AdHoc mode will be replaced with AccessPoint (Infrastructure) mode while the AdHoc driver problems persist. Meanwhile we will keep track of the improvements of AdHoc drivers.

6.5 TESTING OF DTN2 REFERENCE IMPLEMENTATION AND PROPHET

6.5.1 Introduction

The DTN2 reference implementation is in the process of being ported to the OpenWrt platform. It is going to be used on the most of nodes for the next tests in Slovenia if the porting process proves successful and it will run in parallel with the existing Prophet implementation. The DTN2 has some restrictions and it was successfully tested only in static routing environment.

The Prophet implementation has been tested in previous tests in Slovenia and for static routing it was performing well. But when some more complex routing was added to network structure bundles began to accumulate on certain nodes.

6.5.2 Currently available solutions

Basic network structure in Slovenia is constructed in such manner that it is possible to use static routing. This basic structure will be used in next test for the DTN2 reference implementation tests. If dynamic routing will be successfully tested and proved by the authors then also dynamic routing will be tested in following tests.

In the last test the basic network structure was also extended with two additional nodes to test the dynamic routing of the Prophet implementation. The tests for Prophet implementation will be repeated on this complex structure in next tests in Slovenia until they will be successful.

7. PYMAIL: INTRODUCTION AND MOTIVATION

The Pymail system was initially developed to provide an interface between a Postfix mail server running in a DTN equipped Internet gateway and email clients running in small, portable nomadic clients connected through a DTN network. The system was designed to provide an email delivery system that would allow small mobile nodes to send and receive email anywhere in a DTN capable Communication Challenged region, and to use these nodes as data mules to carry DTN bundles. All the nodes are equipped with DTN bundle agent stacks, initially using the DTN2 reference implementation.

Apart from creating a nomadic email capability, the motivation for constructing this system was to provide a convenient means for testing the proposed developments of the DTN naming and addressing scheme in the DTN: uri scheme in conjunction with the PRoPHET dynamic DTN routing protocol. The intention is that nomadic devices would expose EIDs both for the basic name of the node and also the mail acceptance service for one or more users who have access to the node. The PRoPHET routing protocol would route bundles addressed to the relevant mail service accordingly.

The nomadic email system was intended to complement and work in parallel with the 'village' email system that was deployed by the N4C partner Trinity College Dublin (TCD) as a test application for their router implementation. In the village email system the DTN network is used purely as a transport mechanism to move email between Postfix mail servers. Clients access email using unmodified IP based clients from servers located in village enclaves after the mail is delivered to the server over DTN.

Returning to Pymail, the interface in the Internet gateway is mediated through a suite of Python programs that send and receive email messages from the mail server and client and convert them to and from email bundles that are handled by the bundle agent. Each email is encapsulated in a single bundle. For mail being sent to nomadic clients, a separate bundle is generated for each nomadic destination so that they can be routed independently. Email for other destinations that has to pass through a gateway may have multiple destinations specified in a single bundle. The email destination addresses are placed in a 'Delivered To:' extra header at the beginning of the email in its internal form. The Postfix email server is used in the Internet gateway to provide the email service.

The nomadic nodes are also equipped with a suite of similar Python programs that interface with the bundle agent and a local email client. Incoming mail is provided to the local email client through a minimalist POP3 server and outgoing mail is handled by a minimalist SMTP server. These programs are both implemented in Python. This allows the use of (almost) any mail client that may be used on the nomadic device. The initial implementation has been tried out on various machines, including Asus EEE PCs and Nokia N810 Internet tablets, using Wi-Fi communications for exchange of routing information and email bundles.

7.1 ARCHITECTURE

The Pymail system contains four independent components that run as processes within an OS environment. Two main combinations are provided for depending on the context in which the system runs. The system as originally envisaged is designed to transport email in DTN bundles (to RFC 5050 standard) between a gateway machine with access to the conventional SMTP-based RFC 2822 Internet email network and small) nomadic machines intended primarily for a single dedicated user that communicate primarily using DTN. The network architecture is illustrated in Figure 13.

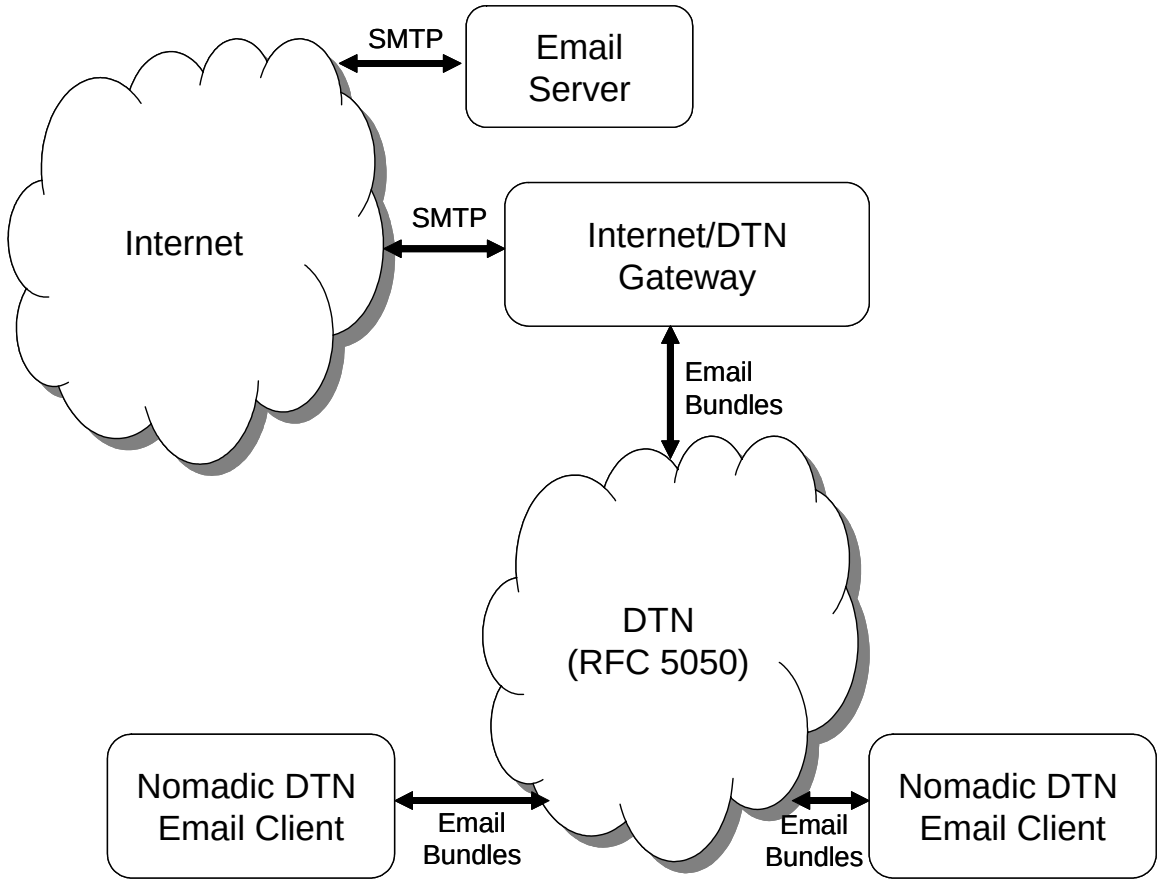

FIGURE 13: BASIC PYMAIL NETWORK ARCHITECTURE

In the basic case, there are two distinct configurations used on the Internet gateway and on the 'outstations'. The Internet gateway is intended to run the Postfix mail server which provides the routing for emails, a DTN2 bundle daemon to transmit and receive bundles, and some interface logic written in Python. The outstation is intended to run one of the various email clients that provide a user mail interface, a DTN2 bundle daemon to transmit and receive bundles, and some interface logic written in Python.

7.2 Experience and progress

The development of Pymail is still work-in-progress. The reason for creating this application suite is to provide a convenient environment in which service oriented DTN naming and dynamic routing can be tested, in addition to providing a useful application for use on portable machines in a communications challenged environment.

Work up to the end of N4C summer testing in 2009 has revealed a number of challenges which we will seek to address during the remainder of the N4C project.

This section gives an outline of the work that has been carried out up to the middle of August 2009.

The aim of this work was to provide a nomadic email system that could be trialled by the hikers taking part in the Extremecom 2009 event in the Swedish Arctic during early August.

7.2.1 Laboratory Work

During the period of May to July 2009 the Pymail application was designed and the coding of the Python interface programs was carried out by Folly Consulting Ltd.

The choice of Python for the interface programs was driven by the desire to have a system which could be simply portable to a wide variety of environments as well as a demonstration of the Python interfaces in DTN2 which have not seen much use in practical applications.

The application suite was then integrated with DTN2 and Postfix, initially using laboratory machines. An Intel Atom based desktop machine running Gentoo Linux provided the main development environment including a Postfix mail server. The outstation was initially implemented on an Asus EEE PC 1000 running Xandros Linux, before moving to Nokia N810.

To compile DTN2 for the N810 (ARM Atmel processor) the Scratchbox virtual development environment was used.

It was also necessary to install Python 2.5 on the N810s. This requires the downloading of a large suite of packages to be installed on the N810s. Given that eventually ten N810s were in use, a short cutting mechanism was used to bulk download the relevant Debian packages on to the N810, avoiding the need to download all the packages individually over the Internet.

It was intended to also load the Norut Hiker's PDA application onto the N810s but due to logistical problems with holidays and installation instructions, we were not able to install this application until very late in the day whilst out in the field.

In addition to testing the new Python programs, the laboratory testing also included work on the PRoPHET implementation in DTN2. It became clear that the PRoPHET implementation still contained a number of bugs, and a significant amount of work was required to identify some of these problems and reach a point where emails could be exchanged using the PRoPHET routing mechanism. The debugging work on this delayed some of the testing and streamlining of installation.

The working assumption during this period was that it would be possible for bundles to be routed from the Folly Consulting server in the UK to the field trial area using the PRoPHET routing protocol with the bundles delivered into the heart of the field trial area via the Fiskflyg helicopter relay from Ritsem to Staloluokta Tourist Camp. However it became apparent that the DTN2 software could only work with one routing protocol in a given bundle agent - it is not currently possible for a single bundle agent to handle different routing protocols on separate links. This proved to be a problem later.

To avoid the need for extensive configuration work on site, the N810s were each allocated a predefined email account and user name. Because of limitations in the EID naming scheme in DTN2 at present, this effectively determined the local EID of the N810 nodes. It also allowed a simple algorithmic mapping from email address to DTN EID.

7.2.2 DTN Research Group Disconnectathon

Just prior to the N4C summer trials the IRTF DTN Research Group held a meeting in Stockholm, Sweden in association with the IETF 75 meeting (July 26 -31). During the IETF meeting a number of groups working on DTN implementations gathered for the 'Disconnectathon'. The object of this

gathering was to carry out interoperability trials for the various implementations of the RFC 5050 Bundle Protocol and other DTN components.

During the Disconnectathon the N810 and EEE PC implementations were exercised and some additional configuration work was done in conjunction with the N4C partner Trinity College Dublin (TCD) to set up static DTN routes that would allow bundles to be routed from Folly Consulting's 'rosebud' server which was providing the email gateway to the Internet through TCD's server 'basil' to the machines which would be deployed during the field trials. As part of this exercise - and in conjunction with the Disconnectathon, TCD and Folly Consulting machines were interconnected in a nine hop 'route' demonstrating the interoperability of a number of different DTN implementations.

7.2.3 Pre-Deployment Work in Lulea and Ritsem

Subsequent to the Disconnectathon, TCD, Folly Consulting and LTU staff involved in the summer trials spent a week in LTU and later a couple of days at the tourist lodge in Ritsem prior to flying out to the field trial area at Staloluokta.

During this period further work was done on the installation of the Pymail program on the N810s and the Norut Hiker's PDA software was added to the N810s. The opportunity was taken to create menu options on the N810 to control the start-up and shutdown of the Pymail system.

A number of practical issues with both the TCD and Pymail systems became critical during this period. Both systems were initially relying on Wi-Fi ad hoc mode for certain communication links. A good deal of time was spent during this period attempting to improve the reliability of the ad hoc links. However it became clear that the ad hoc mode drivers on the various systems were individually unreliable and could not be relied upon to provide the network merge functionality needed in order to ensure that units physically coming together would be able to start a communication when they were in radio range. This resulted in a rapid reworking of the TCD village email system to use only Wi-Fi infrastructure mode. The N810s and Asus EEE PC proved to be somewhat better but it was still decided that for relaying to the Internet infrastructure mode would need to be used.

Furthermore, there were ongoing difficulties with the PRoPHET implementation in DTN2 especially in the N810s. In particular, there were difficulties with ensuring that bundles were correctly rescheduled for transmission after the bundle agent had been shutdown and restarted.

These limitations in the capabilities of DTN2 and the available hardware at Ritsem and in the helicopters meant that it proved impossible to run a PRoPHET based routing system from the Pymail Internet gateway directly to the nomadic email clients. All these difficulties resulted in the need to make some belated changes to the way in which nomadic email bundles would be delivered to the field trial area and then distributed further.

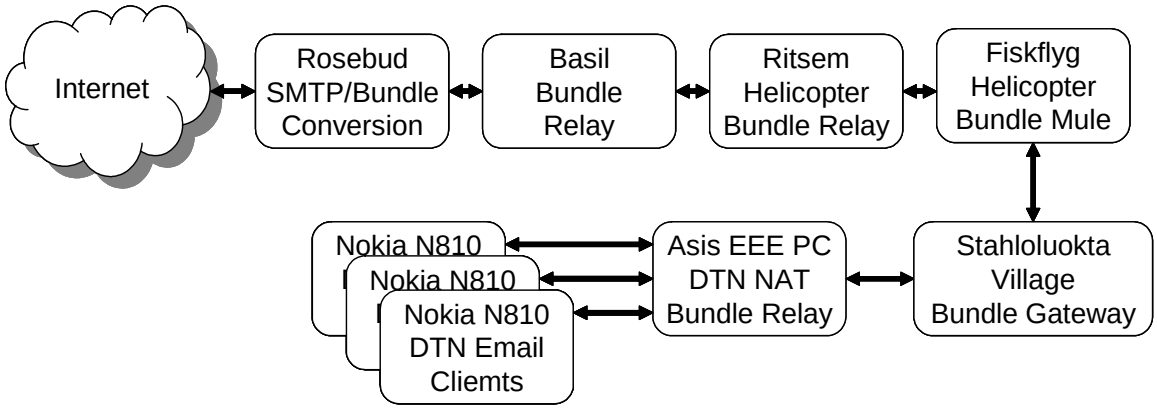

FIGURE 14: BUNDLE TRANSFER PATH DURING SUMMER TESTS

The scheme as finally deployed, shown in Figure 14, involved statically DTN routing bundles to/from the Internet gateway on Rosebud (in the UK) via Internet links via TCD's Basil gateway (in Dublin, Ireland) to the field gateway at the Fiskflyg offices at Ritsem heliport. Here the bundles were transferred over Wi-Fi to a 'data mule' machine (Asus EEE PC) that was carried in one of the helicopters on its regular (semi-scheduled) trips to Staloluokta. On arrival, the data mule connected over Wi-Fi to the solar powered 'village router/gateway' developed by TCD and bundles were transferred again. The nomadic email bundles were then picked up from the village router by another EEE PC which acted as the base station or relay for all nomadic email within the field test area. Nomadic email clients were able to exchange bundles both with other nomadic clients and with the base station.

These changes also necessitated some changes in the way in which bundles were addressed. The original plan was to address bundles directly to the nomadic email clients when they were generated at the Internet gateway. However the amount of configuration of static routes in the links from rosebud to Staloluokta with this scheme would have been prohibitive. Thus an address translator was implemented in the Staloluokta nomadic email base station relay. All bundles from the gateway were now addressed to the relay. They were locally delivered to a Python program that rewrote the destination DTN address based on the email delivery address and passed them to a second bundle agent that provided the delivery to nomadic email clients.

The intention was to provide a number of the hiking participants in Extremecom who passed through Ritsem on Saturday 8 August on their way to the start of the Extremecom walking phase with N810 units. This would allow them to experiment with during the hike. Although this was put into practice, the amount of time available with the hikers before they set off was inadequate to provide training on the new applications. The weather during the hiking was also very wet and problems with limited battery lifetime meant that no significant email traffic was generated during the hiking.

7.2.4 Work in the Field

After the Extremecom hikers set out form Ritsem, the Folly Consulting, TCD and LTU developers flew to Staloluokta to set up the TCD village router and await the arrival of the hikers. During the three days before the hikers arrived, the helicopter data mule service was exercised and some nomadic emails were sent from a local station in Staloluokta.

Ensuring that the various pieces of equipment had adequate power proved to be challenging even with reasonably good weather and a small generator.

Some further work on configuring the N810s was also carried out.

As mentioned, the hikers were unable to make significant use of the email application whilst hiking and it will be necessary to consider the best usage of the system for the future.

The whole party travelled back to Ritsem and then onto the conference venue at Saltoluokta where some additional emails were dispatched. The village router system was dismantled before leaving.

7.3 EVALUATION AND WAY FORWARDS

The development of Pymail and the trial deployment during the Summer 2009 field trials has provided a very useful view of the user and technical problems that stem from trying to provide seamless application integration across the boundaries of the existing Internet and DTN deployments in communications challenged environments.

In this section we summarize the successes and the issues that arose in the development and deployment of the Pymail system during summer 2009, and then propose a way forwards that will make it a more useful part of the test beds during the remainder of the N4C project.

7.3.1 Successes

7.3.1.1 Use of Python DTN2 API

The DTN2 Python API proved to be convenient and worked as advertised. The only difficulty encountered was during cross-compilation where some adaptation of the Python distribution makefile was necessary due to some macros being set inappropriately (this is not a problem with the DTN2 distribution).

To sidestep the problems caused by the effectively half-duplex nature of the DTN2 API, separate connections to the DTN bundle agent daemon were used for sending and receiving bundles. This allows the receiving connection to be permanently ready for reception and minimises the processor cycles used by the receiver by having the receiver sitting in an IO wait state in a 'select' system call. The send and receive aspects are handled by separate threads with their own connections and consequently there is no need to break out of the receive 'select' call in order to send a bundle - the functions are essentially independent.

7.3.1.2 Implementation of DTN2 on Nokia N810 and Asus EEE PC 1000

The intention with both of these devices was to implement Pymail on the default operating system. For the N810 this is Maemo and for the Linux based EEE PC this is Xandros. (Some of the other systems being developed in N4C have moved to Ubuntu on the EEE PC because of problems with Xandros.)

This was reasonably straightforward. Compiling the DTN2 code did not require any special modifications to the code.

Due to the very limited memory on the N810, it was necessary to use 'stripped' executables - by default, DTN2 is linked with debugging symbol tables which makes the programs very large. Removing the symbol tables using the 'strip' command reduces the size of the DTN2 bundle agent daemon (*dtnd*) by about 90% (from approximately 25MB to around 2.5MB).

For the N810 it was also necessary to install

- the Berkeley Database system as the N810 does not support a *dbm* style database by default,
- the Python 2.5 system to allow the use of the Python scripts, and
- the TCL interpreter to support the DTN2 control connections.

Both the bundle security extensions and the XML-based extension system were omitted for the N810 build as they were not necessary for this deployment. The intention was to use the built-in memory card in the N810 to provide storage for bundles and emails but this was not completed in time for the trials, so all files were stored on the internal memory of the device. This was not a problem for these trials as data volumes were very low, but for the future this would be essential. The speed of writing is a potential problem for this memory. One solution is to place the files in main memory initially and copy the files in the background as necessary - further work will be done on this problem in due course. Scripts to start and stop Pymail were connected to the N810 menu system (as has been done with the Hiker's PDA system).

No particular issues were encountered on the EEE PC - the standard local development environment provided all the necessary tools.

Since the trials the rationale for maintaining Xandros as the EEE PC operating system has been negated by a change of policy at Asus. Asus are no longer supporting a Linux distribution as a pre-installed option, and the Xandros support has been steadily withdrawn. This and other matters are discussed in Section 7.3.3.7.

7.3.1.3 Delivery of Emails Directly to and from a Nomadic User

A small number of emails were successfully delivered in both directions from the lab and field locations.

7.3.2 Issues

7.3.2.1 User Acceptance and Usability

The limited amount of testing that we were able to carry during Extremecom 2009 revealed that it is unlikely that recreational hikers will voluntarily activate a nomadic device during normal hiking. In particular requiring some action from a person carrying a nomadic device to exchange data with another passing hiker is not likely to meet with user acceptance. The weather during a good part of the hike was inclement (heavy rain) and hikers would be unwilling to open their packs or even necessarily pause when passing others in such circumstances.

Hence any exchange of data needs to be triggered without user intervention. This has implications for the design of the system and interacts with the battery lifetime and Wi-Fi modes as described in the next two issues when in use by Hikers and other casual users.

Such casual users need to be incentivized to make active use of the device while actually hiking - it is necessary to actively try out information services on a wider scale to see if these can give sufficient incentive. On the other hand email usage at tourist villages would certainly seem to be popular.

The system was not trialled by employees for work related purpose this year. It seems that a work related system would have a greater chance of acceptance.

7.3.2.2 Ad-Hoc Wi-Fi Mode

The intention was to use the Wi-Fi Ad-hoc mode to connect nomadic devices to each other and the Internet relays and gateways. Although the Nokia devices and the EEE PC using Xandros performed reasonably well with ad-hoc mode, the field trials encountered a large number of issues with ad-hoc mode. In particular the connections to data mules had to be reconfigured to use infrastructure mode in most cases because of difficulties with getting arbitrary pairs of machines of different types and with different operating systems to connect reliably.

The problem here appears to be that ad-hoc mode is not very well tested by manufacturers and software developers, so that the standard behaviour expected does not always occur. In many cases the network merge function that is supposed to occur when nodes come into range of an existing communication cluster appears not to work correctly, leading to a partitioned network. Overall the difficulties with ad-hoc mode are a major stumbling block for nomadic operation.

7.3.2.3 Battery Lifetime and Application Management

The battery lifetimes of many of the devices that are potentially useful as nomadic clients are limited. Typically only a few hours of active use is possible. The problem is exacerbated by using ad-hoc Wi-Fi mode which requires periodic active beacon transmission which is a major power drain both from the point of view of radio transmission and the need to wake up the device fully.

Power drain could be lessened by requiring the user to start the application that uses ad-hoc Wi-Fi but as discussed in Section 7.3.2.1, casual users would not like this sort of scheme.

Very careful attention to application design is needed to ensure that applications do not keep the node processor active unnecessarily, preventing the device from entering as deep a sleep mode as possible.

7.3.2.4 N810 GPS System

The Nokia N810 contains a built-in GPS system which can provide location information to aid in logging of events. Unfortunately it is very slow and power hungry. However, since nomadic email clients are mostly expected to be carried by people moving around outside buildings and vehicles, GPS location is a viable means of locating clients.

7.3.2.5 DTN2 PRoPHET Implementation

The PRoPHET implementation in DTN2 has a number of issues, both of architectural design and buggy implementation. A number of these bugs were identified and fixed during lab testing but it has become clear that there is a more generic issue with DTN2 which does not appear to maintain sufficient state about bundles that have or have not been transmitted when the bundle agent is shutdown and restarted. Both PRoPHET routing and static routing seem to have encountered problems with bundles either being transmitted multiple times or not transmitted at all because of this problem. This will require considerable work to ensure that shutdown/restart works as expected.

The PRoPHET implementation also does not match either the PRoPHET specification or the other main implementation of PRoPHET (resulting from the SNC project) as regards how routing metadata is exchanged between meeting nodes. The DTN2 implementation transfers this information in bundles whereas the specification uses an out-of-band link. This leads to issues when nodes are connected for a significant length of time and should re-exchange metainfomation.

Difficulties with the PRoPHET scheme meant that most of the field deployment had to fall back to static routing which limited the possibilities of multi-hop forwarding. In practice other issues meant that this did not affect the email delivery significantly, but it obviated the idea of using the nomadic clients as data mules.

7.3.2.6 DTN2 Addressing

Currently, DTN2 can only route bundles to a node based on a single EID base component (similar to a host name) - the bundles can be demultiplexed based on additional components once they arrive at the node. This is inconvenient for 'service based' delivery as is needed for email addresses. It would be highly desirable to be able to route to an EID that represents an email address that does not depend on the node EID of the (nomadic) node where the user email client is currently located. Essentially a node should be able to advertise multiple EIDs, some of which refer to services as discussed in recent IRTF drafts relating to the dtn: URI scheme.

7.3.2.7 Heterogeneous Routing in DTN2

A DTN2 bundle agent cannot use more than one routing mechanism in a single instance. Thus a single bundle agent cannot provide a means of interlinking areas of a DTN network using static routing and PRoPHET routing. This meant that the eventual field deployment of Pymail had to use a DTN NAT scheme in which bundles were passed between two bundle agent instances in a relay node, one providing static routing through the 'village router' back to the Internet gateway and the other providing PRoPHET routing to the nomadic clients.

Some thought need to be given to how heterogeneous routing can be added to DTN2.

7.3.3 Way Forwards

Further development of the Pymail application will require some further development and

7.3.3.1 User Model

As noted in Section 7.3.2.1, casual users were not inclined to use the email system while hiking, partly because of the effort needed to work with the prototype system in difficult conditions and partially because there were not clear benefits to using the system with the facilities available.

We need to work with the Norut Hiker's PDA application and information providers to see if incoming emails (e.g., regarding weather and local facilities) could be a useful 'killer application' that would move casual users to make use of the nomadic clients.

We also need to put the tool in the hands of local professionals for employment related purposes.

7.3.3.2 Wi-Fi and Battery Lifetime

Apart from ensuring that the Pymail application is as friendly as possible as regards allowing the nomadic clients to enter sleep mode for as much time as possible, the existing hardware does not appear to offer many options for improvement. Increasing time between ad-hoc mode beacons might reduce power consumption at the risk of missing communication opportunities.

One way forward may be to implement the system on cellular telephones that are already optimized for long battery lifetimes. However the use of ad-hoc mode Wi-Fi for communications may still reduce the effective battery lifetime even on cellular phones. One advantage appears to be that Nokia cellular phones (at least) have good ad-hoc mode implementations.

For the future it is possible that either Intel's 'My Wi-Fi' proposal or the upcoming 'Wi-Fi Direct' proposal in the 802.11 standards committee may provide a lower power consumption solution, but details are sparse and it is unlikely that commercial hardware will be available during the remaining part of the N4C project.

Finding a solution to the general problem of low power mobile neighbour encounter discovery in a wireless environment is a key research issue that would greatly enhance the utility of nomadic clients.

7.3.3.3 DTN Improvements - Addressing

Proposals exist to enhance the dtn: URI scheme[2]. Experimentation with these proposals will require significant changes to the DTN2 code. It is anticipated that this work will be carried out, but it is not absolutely essential to experimentation with Pymail.

7.3.3.4 DTN Improvements - PRoPHET Implementation

In order to make further progress with dynamic routing an improved implementation of PRoPHET is required. The DTN2 implementation requires major rework to make it fit for purpose and to implement the standard specification that is now approaching publication as an experimental standard RFC.

Folly Consulting is investigating ways to create a bundle agent that will provide the required functionality and have a smaller 'footprint' than the DTN2 implementation. See Section 7.3.3.7 for further details.

7.3.3.5 DTN Improvements - DTN NAT Workaround

Modifying DTN2 to allow it to provide different routing protocols on specified links requires a great deal of effort and may require considerable restructuring of the bundle agent. It is unlikely that this work can be carried out during the N4C project. A workaround that provides the required functionality was implemented during the 2009 testing period. Separate bundle agents are deployed on a relay node with one agent providing (say) static routing for some links and another agent providing (say) PRoPHET dynamic routing for other agents. In order to make this work, an addressing scheme similar to the Network Address Translation (NAT) scheme familiar in IP networks has to be adopted. An interface program ('Application Layer Gateway' in NAT parlance) routes bundles between the two bundle agents on the relay node, and modifies the EID destination addresses to the appropriate value for the target network based on information from the email addresses in the bundle payload.

This scheme also allows different bundle agent implementations to be used for the various routing protocols which may be important if the DTN2 PRoPHET implementation is not upgraded during the N4C project (see Sections 7.3.3.4 and 7.3.3.7.)

7.3.3.6 Duplicate Suppression and Retransmission

Since a DTN network is not able to guarantee that an email will be delivered (e.g., if a node does not come in range of a relay before the bundle expires), it may be necessary to retransmit a email if no delivery notification is received before the bundle is likely to have expired. It is also possible that an email will be received multiple times because of DTN routing injecting multiple copies into the

[2] There are currently two DTN Research Group drafts addressing this issue: http://tools.ietf.org/internet-drafts/draft-irtf-dtnrg-dtn-uri-scheme-00.txt [4] and http://tools.ietf.org/internet-drafts/draft-davies-dtnrg-uri-find-01.txt [5].

network. It was intended that the Pymail interface programs support retransmission and duplicate suppression. This will be implemented for the next session of testing.

7.3.3.7 Hardware and Operating System Developments

The hardware and available operating systems on equipment suitable for use in nomadic email clients is evolving rapidly at the moment. The Nokia N810 used in the 2009 trials is now obsolete and has been replaced by the N900 which incorporates cellular telephone technology in addition to the Wi-Fi technology in the N810 but retains the Maemo Linux operating system for the non-telephony applications. Asus EEE PCs no longer provide the Xandros operating system as standard; they are now sold with Microsoft Windows as default.

In order to cope with these developments it seems essential to have an implementation of DTN that will support the RFC 5050 bundle protocol and a small set of routing protocols (at least epidemic, static and PRoPHET routing) on an implementation platform that is deployable on both Linux and Windows platforms, and is compatible with DTN2 at the bundle protocol layer. Folly Consulting is investigating how this can be achieved by improving the SNC PRoPHET code using Qt as the implementation platform.

7.3.3.8 Testing in a Less Power Challenged Area

Testing the Pymail application in the Sámi areas proved a salutary experience. The basic environmental challenges of power availability and weather tend to overwhelm the protocol and implementation issues. It seems important to do considerable additional live testing in an area where power is less of an issue to ensure that the application is stable and dependable before retrying in an environment with the full set of challenges. This would give the best chance to determine the usability of the application for the future.

7.4 SUMMARY FOR NOMADIC EMAIL APPLICATION

Development of the Pymail Nomadic Email system has demonstrated that the nomadic email system has potential, but considerable work is needed on the DTN infrastructure to make it a realistic, deployable system. The work during the Summer 2009 testing campaign has highlighted a number of areas where work is needed and a plan for development and further testing, especially in less power challenged areas, is being studied.

8. FUTURE WORK FOR WP3

Other tasks that need to be done in WP3, and integration with other WPs:

- WP2: Make DTN2 infrastructure a realistic, deployable system [N4C-D2.3].
- WP3: New relevant hardware for Hiker's PDA: Nokia N900 and several Android phones.
- WP4: Heterogeneous routing and DTN2: Hiker's app assumes dynamic routing, and makes Hiker's PDA a data mule. Static routing makes it difficult to make use of the nomadic clients. Hence, a hybrid infrastructure, that uses static topology to cover longer distances, villages and main routes of traffic, while dynamic topology can be used within a single or a group of villages, is needed.
- WP5: Integration with meteo station, village router and data mule.
- WP6: Wi-Fi is used by Hiker's PDA and for short distance communication between village router and data mule. Need to look into integration to WiMax [N4C-D6.3].
- WP7: The method from WP3 test plans has been adopted by the whole project for future tests and integration. Integration and laboratory testing of subsystem [N4C-D7.2.2].
- WP8: Test beds: Participation in winter/summer tests and integration of the test beds [N4C-M8.3] [N4C-D8.3].

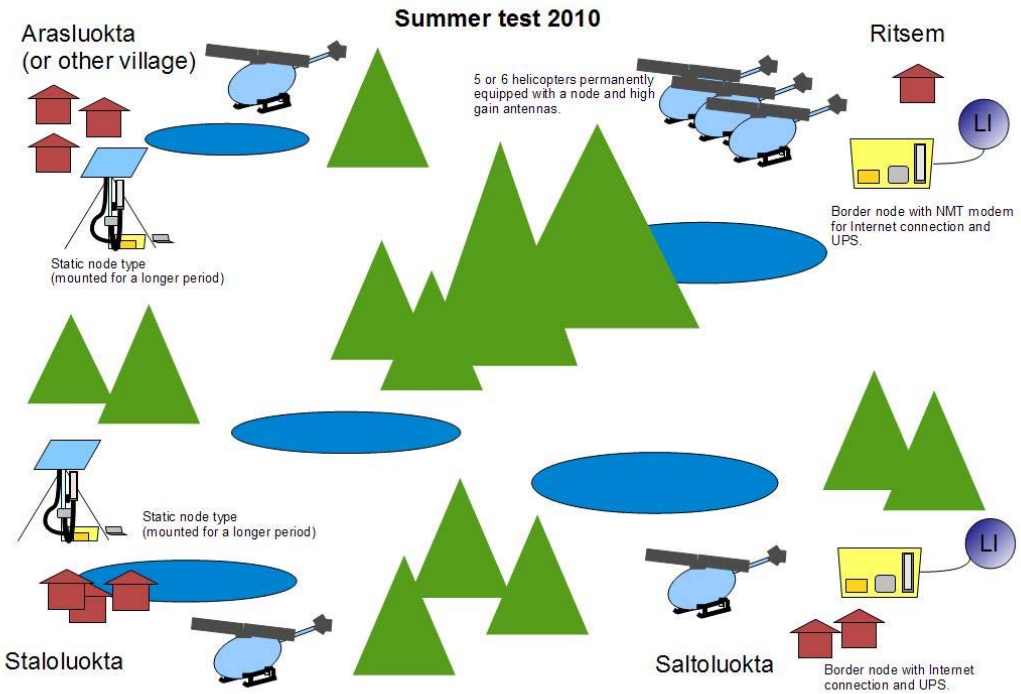

FIGURE 15: N4C SKETCH OF SUMMER TEST 2010

9. DISSEMINATION ACTIVITIES

9.1 CONTACTS WITH TELEMEDICINE

The partner Norut has engaged in dissemination of the N4C project by (informally) presenting the project at several occasions to the academic community in Tromsø. Communications challenged regions are a familiar concept to many research activities in the Arctic regions. The opportunities that open up with DTN, on the other hand, are less well known, and we observe a growing interest for more information in this area. There is one institution that is particularly interesting for future development: The Norwegian Centre for Integrated Care and Telemedicine (designated as WHO Collaborating Centre for Telemedicine).

So far the dissemination activity has resulted in three follow-up meetings with Telemedicine, who will look at the possibilities for DTN-based services in remote areas in Norway and Northwest Russia.

9.2 BOOK CHAPTER FOR TEXTBOOK IN PERVASIVE COMPUTING

By invitation from professor Jose Luis Sevillano Ramos at the University of Sevilla, Norut submitted a contribution to a textbook "Standards and implementation of pervasive computing applications" in August 2009. The reason for the invitation is that Norut and the University of Sevilla are partners in several telemedicine projects, and through this contact the N4C project has been presented to them. Our contribution, authored by Karl Johan Grøttum, was a Case Study with the title "Pervasive Computing in Extreme Areas: The Hiker's Personal Digital Assistant" [N4C-PCA].

9.3 SUBMISSION OF ABSTRACT AND POSTER PRESENTATION TO FIA

Norut has, together with LTU, answered to the invitation from FIA to submit an extended abstract for the FIA Book 2010. The abstract [N4C-FIRE] was accepted, and we proceeded to make a poster presentation [N4C-FIA] for the Future Internet Assembly meeting in Stockholm 23rd-24th November 2009.

The poster was presented during the conference, and was voted to be accepted as a candidate for a full paper submission before the FIA Valencia event in April 2010.

9.4 SUBMISSION OF FP7 PROPOSAL BASED ON RESULTS FROM N4C

In September 2009 TCD and Norut were invited to join a consortium headed by University of Surrey, UK to prepare a proposal under the call FP7-ICT-2009-5.

We regard such an invitation as a fruitful spin-off of the dissemination of N4C and as an acknowledgement of the preliminary results created in N4C. It also underscores the great advantage of building networks between research institutions in the Framework Program.

The proposal has the acronym SYMBIOSIS (Research of Systems for Scalable Intelligent Delay Tolerant Architectures for Wireless Sensor/Satellite Networks). The SYMBIOSIS abstract (below) clearly demonstrates the influence from N4C:
At its core, SYMBIOSIS brings together a strong industrial and academic partnership, with support from user organisations, to further develop the Delay/Disruption Tolerant Networking (DTN) architecture and associated implementations to be ready for commercial-grade deployment of real Wireless Sensor Networks (WSNs) in real-world environments, thus paving the way to a future

Internet of Things (IoT). The research challenges that need to be considered relate mainly to scaling and the need to explore new techniques: additional integration and security mechanisms are required for the large-scale systems that will justify industrial investment in this technology. We will develop and validate technology and services for two environmental monitoring applications: one to monitor oil spillages on the sea, (in Norway) and another to monitor water pollution of Napo River (in Peru). In both cases, SYMBIOSIS will deploy a DTN overlay infrastructure over WSNs to ensure reliable data collection, aggregation and transmission to remote control centres for appropriate action. Endusers will be involved and play a central role in the design and evaluation of these SYMBIOSIS trials. The challenging environments considered in SYMBIOSIS demand the use of a range of communication link technologies, including WiFi for local communications and satellite for coverage of remote locations and seas. Unmanned Aerial Vehicles (UAVs) will also be used both as sensor platforms and as "data mules", to transfer data from WSNs to intermediate terrestrial DTN gateways (or, if possible, also directly to control centres via satellite links). SYMBIOSIS will develop working DTN-based applications for both environmental monitoring scenarios and, if successful, will have convincingly shown that DTN is ready for commercial use. In addition, SYMBIOSIS will along the way seek to achieve a number of technology "firsts", including the real-world use of UAVs as DTN "data mules", new sensor data aggregation, lightweight network control and security mechanisms for such challenging environments, and the use of DTN protocols to integrate a commercial satellite offering into the Internet of Things (IoT).

The proposal was submitted to the EC 27/10/2009. The result from evaluation of the proposal is expected in the next couple of months.

10. REFERENCES

[1] Delay-Tolerant Networking Research Group, http://www.dtnrg.org/wiki, accessed 22nd August 2009

[2] Farrell, S. and Cahill, V. (2006) *Delay- and Disruption-Tolerant Networking,* Artech House.

[3] Beck, Richard et al (2007) GPSDTN: *Predictive Velocity-Enabled Delay-Tolerant Networks for Arctic Research and Sustainability.* The First International Workshop on Tracking Computing Technologies, IARIA-IEEE Track 2007, July 1-6, 2007 - Silicon Valley, USA

[4] Fall, K., Burleigh, S., Doria, A., and Ott, J. (2009) *The DTN URI Scheme*, http://tools.ietf.org/internet-drafts/draft-irtf-dtnrg-dtn-uri-scheme-00.txt (work in progress), March 2009.

[5] Davies, E. and Doria, A. (2009) *Adding the "find" Operation to the dtn: URI Scheme*, http://tools.ietf.org/internet-drafts/draft-davies-dtnrg-uri-find-01.txt (work in progress), October 2009.

[N4C-D2.3] Functional Specification for DTN Infrastructure Software comprising RFC 5050 bundle Agent and Associated Components (Version 0.2)

[N4C-D3.1] D3.1 Functional Specification

[N4C-D6.3] D6.3 N4C Mesh Networking for static long distance wireless sensor networks

[N4C-D7.2.2] D7.2.2 N4C Laboratory Testing on Integrated Subsystems

[N4C-M8.2] M8.2 Winter Tests Report 2009

[N4C-M8.3] M8.3 Summer 2 interconnected tests report

[N4C-D8.3] D8.3 Test bed creation and methodology

[N4C-HPU] WP3 Scenario for Hiker's PDA, Use cases

[N4C-STP] N4C Plan for summer test 2009 - Hikers PDA

[N4C-VST] N4C Verbal log Norut Summer Test 2009

[N4C-WTP] N4C Plan for winter test 2009 - Hikers PDA

[N4C-WTS] N4C Post Winter Test Summary

[N4C-FIRE] Maria Udén, Karl Johan Grøttum and Sigurd Sjursen. (2009) *Realizing Delay Tolerant Networking as Access Enabler: Services Arizing in New Realms and the Driving Applications.* Extended abstract for FIA Valencia book 2010.

[N4C-FIA] Maria Udén, Samo Grasic, Karl Johan Grøttum and Sigurd Sjursen. (2009) *Realizing Delay Tolerant Networking as Access Enabler: Services Arizing in New Realms and the*

Driving Applications. Poster for FIA Stockholm 2009.

[N4C-PCA] Daniel Cascado and Jose Luis Sevillano. Luis Fernandez-Luque, Karl Johan Grøttum and Lars Kristian Vognild. Tatjana M. Burkow. *Standards and implementation of pervasive computing applications.*

[N4C-RAR] Risk Analysis Mid Project Report

[Karlsson et al., 2008] Karlsson, S., Osvalder, A.-L., Rose, L., Eklund, J., & Odenrick, P. (2008). Utvecklingsprocesser. In Bohgard, M. (ed.), *Arbete och teknik på människans villkor*. (pp. 569-610). Stockholm: Prevent.

[Pahl and Beitz 1996] Pahl, G., & Beitz, W. (1996). *Engineering Design – A Systematic Approach* (2nd ed.). London: Springer.

[Ulrich and Eppinger 2008] Ulrich, K., & Eppinger, S. (2008). *Product Design and Development* (4th ed.). New York, NY: McGraw-Hill.

A APPENDIX A

A.1 USING THE KINDLE E-BOOK READER IN A DTN CONTEXT

While at first the Kindle, and other such devices seemed perfect for remote applications, the evaluation showed that at this time, it would not be a reasonable use of N4C project resources to engage in development of this as an N4C application. The primary reasons for this conclusion are:

- Most of the books available were only in English, not Swedish, Sámi or Slovenian
- While recently the device has been made available to European markets, it is still expensive for the user community presented by the N4C project.
- Although delivery of reading material is becoming possible in regions of Europe's 3G network, the delivery would still require development of a server/client pair to reach remote regions. The project resources are not available for more then one such client/server pair in the project at the moment and the previous two reasons indicate that that this application would not have sufficient utility to the current participants.
- A possible implementation using web caching and email does not offer enough application experience or reader advantage to be worth engaging in at this point in time.

It is, however, recommended that consideration be given to developing such an application in the future when there may be more resources available and more literature in the required languages.

A.1.1 DESCRIPTION OF KINDLE APPLICATION

The Kindle is a standalone appliance created primarily as a book reader. The Kindle is one of two new appliances that use liquid ink instead of an active light source, the other being a Sony product.

A.1.2 REASON FOR THE KINDLE APPLICATION

In terms of remote application, an application that allows for the remote acquisition of books to a remote locations seemed at first glance as something that would useful to those who were not able to obtain new reading material easily.

Additionally the use of liquid ink in advice that only needed to use power when a new page was being printed or when books were being downloaded from a 3g network, meant that the device had very low power consumption. Because of the use of liquid ink, the devise uses very little power and when not in communications mode can be used for a week or more on one charge.

A.1.3 USE OF THE KINDLE APPLICATION

There are several basic use models for the Kindles.

1. Using a credit card number left on file with Amazon, order books using what they call their Whispernet network to order books on the Kindle and have them delivered to the Kindle. Whispernet refers to an arrangement they have with Sprint for use of their telephone network. This was originally restricted to the US, but has now been extended to Europe at a premium cost.

2. Using the Amazon website from a computer to order material and have it delivered directly to the reader's Kindle. Until recently direct delivery only worked in the US. It now also works in Europe at a premium price.

3. Using the Amazon website to order material and have it delivered via the Internet to the users computer to then be transferred to the Kindle via USB. Until recently this was the mode that had to be used when travelling outside of the US. For cost reasons, it is still the primary method.

4. Using ones own documents in PDF or HTML form, a reader can submit these for conversion to Amazon's proprietary (.AZW) format for use on the Kindle. These can either be downloaded automatically to the Kindle for a fee (10 cents in the US, more in Europe) or via the Internet to the computer for free. Amazon has set up an interesting method for submitting documents to conversion. To have them automatically converted and downloaded via Whispernet, one send the to username@kindle.com and to download them via the Internet one sends them to username@free.kindle.com.

5. There are many book length works available on the Internet for free. These are works whose copywrite has expired. Two of the main sources are the Gutenberg Project (http://www.gutenberg.org/wiki/Main_Page) where books can be download in several form including plain text and Mobipocket[3] (http://www.mobipocket.com/) that produces many books that can be downloaded for free in their semi-propietary .MOBI format. Books and other files in either pain text or mobipocket format can be downloaded to the Kindle directly using the USB cable as the Kindle accepts these two formats in addition to its own proprietary format. It should be noted that many of the features of Amazon books including functional indices only work with books in the .AZW format. It should also be noted that most of the works available are in English.

It is clear that there are many uses for the Kindle, from new books to classics, to documents of one's own, and that there are many ways in which documents can be downloaded and read, without being limited to purchase from Amazon. The only thing that was really limited to purchase from Amazon are books still under copywrite. It should also be noted that except for these new works, there were other ways to read any of the other materials, though they would consume more energy then a liquid ink device.

A.1.3.1 USE OF EXTENDED AND NON LATIN CHARACTER SETS

As part of the investigation, the Kindle as checked for its ability to render non Latin characters. While the expectation was that the rendering would work, it needed checking. The test was done by submitting a PDF containing various characters to see how they would be rendered. In the case of these tests, the rendering was not done properly in the case of Hebrew or Korean script, and there were a few errors in extended Latin scripts, though Swedish was conveyed properly. An investigation was done with Mobipocket books in Danish (they were the most available outside of English) and it appeared to be rendered properly. I was not able to find any materials in Slovenian to test. Before committing to a project involving the Kindle it would be important to do further testing into available materials in non English character sets. In this investigation it was determined that at least one of the languages used by the readers who would be resident in a Living Laboratory would work.

[3] It should be noted that Mobipocket in an Amazon company.

A.1.4 IMPLEMENTATION SCENARIO FOR KINDLE N4C APPLICATION

In order for the Kindle to be used as an application in the N4C project, the following scenario could be followed:

- Remote reader would need use the remote caching capabilities to order a book from the Amazon catalogue. While it might be possible to have the caching ability added to the Kindle software capabilities, there is no interest at Amazon in this change at this point in time.[4]
- This request would need to be mediated by a client process in the remote network.
- In making the request, the client process would also need to notify the server to monitor for the availability of the requested document for download.
- Once the server downloaded the document it would need to transfer it, probably as a Mime object using email, to the requestor who could then download it to the Kindle.
- Remote users would also be able to use the username@free, kindle.com interface for insertion of user generated text.
- Remote readers would also be able to download book from Gutenberg Project on Mobipocket using web caching techniques.
- All transfers to the Kindle would need to use the USB interface.

There are other variants, but most of the practical alternatives involve use of web caching and email response. And all involve transferring using the USB connection method.

A.1.5 RECOMMENDATION

As appealing a devise as the Kindle it, its use in the N4C project does not appear to add much to experimental application space given that it just uses other capabilities being build into the DTN systems such as email and web caching. also given the limitation of the implementation and the limitation of available work in languages other then English, it also does not seem as if it would offer the readers enough of an advantage at this point.

The recommendation is that this application be shelved at this point and perhaps reviewed again at a later date. If at some point Amazon could be convinced of the value of opening up its system to DTN capabilities, the situation would be radically different and merit a close re-review.

[4] Several communications were sent to Amazon regarding the implementation a DTN capability. These went unanswered. In a personal conversation with a manager in Kindle development, I was informed that the development goals involved spreading as much as possible to the existing 3G network. As there would be a way to work around this limitation by using the online capability of a DTN enable Web access of a regular computer this lack of interest was not seen as a showstopper, though it was a limitation.

B APPENDIX B

Author: Martin Bonnevier, master student at Luleå University of Technology.

B.1 PRODUCT DESIGN FOR N4C

B.1.1 PROBLEM

This part of the N4C project has been carried out as a Master's thesis project in the Ergonomic Design and Production Engineering MSc programme at the division of Industrial Design at Luleå University of Technology (LTU). The project aimed to investigate and prepare a proposal on how the product, which makes the DTN technology useable for the intended users in practice, should be designed. The intended users were, in this case delimited to the Sámi people in Swedish Laponia.

B.1.2 METHODOLOGY & IMPLEMENTATION

This chapter concern the methods used during the project. Furthermore, it describes how the methods have been used in the different stages of the project. Overall, a method for systematic product development has been used, which involves following a certain iterative methodology to achieve as good result as possible.

A cyclical work procedure result in a good understanding of the overall picture in a short period of time, and the chances of getting a good result increases thanks to the iterative method [Karlsson et al., 2008]. The steps included in the process are repeated 2-3 times. During these iterations, the development work is gradually moved forward. This is illustrated in Figure 16 below.

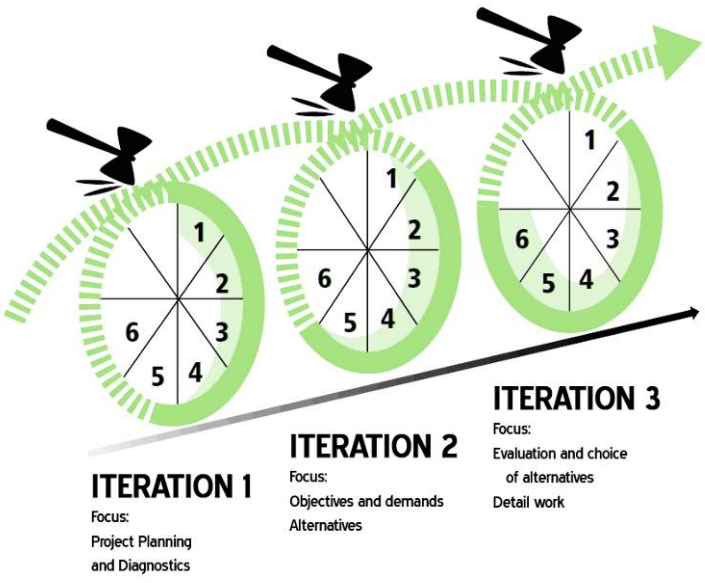

FIGURE 16: THE PROJECT SPIRAL SHOWS HOW THE PROJECT FOCUS IS GRADUALLY MOVED TOWARDS CONCRETE WORK WITH CONCEPTS AND PROPOSALS. (ADAPTED FROM KARLSSON ET AL., 2008)

The product development process has been described in many books and irrespective of how many the different steps are and what they're called; the overall process is in practice the same. Examples on how this process might look are described in e.g., [Karlsson et al. 2008], [Pahl and Beitz 1996] and [Ulrich and Eppinger 2008]. Even if the level of detail varies from the different descriptions of the product development process, the classification into the four steps Definition of demands/function/task, Conceptual design, Detail design and Final design are common for most opinions [Karlsson et al., 2008].

B.1.3 NEEDFINDING

Even though needs don't generate specific product or service solutions, the science of Needfinding can be a dynamic platform for design (Patnaik & Becker, 1999). In this section terms related to this process will be defined and the steps that help identify development opportunities will be outlined. Critical need-finding principles include letting consumers guide the flow of research, collecting data in a variety of different forms, and integrating research and design in a series of iterative stages as a way to fine-tune results (Patnaik & Becker).

The Needfinding process is based on the understanding of five statements. These statements are listed below. (Patnaik & Becker)

- Needs lasts longer than any specific solution.
- Needs are opportunities waiting to be exploited, not guesses at the future.
- Needs provide a roadmap for development.
- Needs spur action.
- Needs are obvious after the fact, not before.

The Needfinding process has been used iteratively in the first half of the project, with focus on the Sámi future DTN users in Jokkmokk municipality in northern Sweden. The process of identifying the users' needs has been the Needfinding process described by Patnaik and Becker, refined with methods for interpreting and organizing needs by [Ulrich and Eppinger 2008].

B.1.3.1 CENTRAL PRINCIPLES OF NEEDFINDING

Uncovering needs reliably requires an organized research effort, even if previously undiscovered needs occasionally do appear to designers by happenstance. The following points characterize the philosophy behind Needfinding. (Patnaik & Becker, 1999)

- Look for needs, not solutions.
- Make research and design seamless.
- Go to the customer's environment.

- Look beyond the immediately solvable problem.
- Let the customer set the agenda.
- Collect eclectic forms of data.
- Make findings tangible and prescriptive.
- Iterate to refine the findings.

B.1.3.2 THE NEEDFINDING PROCESS

The principles described in the section above manifest themselves as an iterative four-stage process for studying people. They determine the approach used by researchers at every step of that process. The goals of each stage are described in general terms below. (Patnaik & Becker, 1999)

- *Frame & Prepare.* Determine the research goals, the customer group being researched, and the specific sites to visit. Preparation before going to the customer's environment helps researchers to know what questions to ask and what information to look for.
- *Watch & Record.* People are often so accustomed to certain problems in their lives that they become oblivious to them. When asked about the situations in which these latent problems occur, they frequently fail to recognize that the problems exist at all. Directly observe people's behaviour in their own environments to gain a clearer understanding of their situations.
- *Ask & Record.* Observation alone can't tell researchers everything they want to know. Observation may offer occasional indirect indications, but generally doesn't give clear access to people's reasoning and emotions. To better understand these motivating factors, interview people after the observed activities have been completed to understand the context in which those activities just occurred. Answers to questions and further discussions can give researchers insight into why a person acted in a certain way and what he or she felt during the observed situation. This is crucial information for determining people's needs.
- *Interpret & Reframe.* Once data is collected, the final stage of the Needfinding process is to interpret the findings and revise the research questions. Information collected in the customer's environment helps refine one's understanding and prepares the team for another iteration of research. Product development can then continue in parallel to the ongoing Needfinding activity. Because Needfinding is about studying people, as well as developing products, always frame interpretations in terms of what problems need to be solved to improve the customer's situation. The interpretation process is discussed in section 2.3.4 below.

In this project, five one on one (or in a couple of cases one on two) interviews has been made, and were complemented with one focus group. The focus group was carried out in the common office of the Jokkmokk Sámi villages in Jokkmokk. The session was recorded with both video camera and MP3-player. The interpretation of the needs was made according to the suggested guidelines in Ulrich Eppinger (2008).

A sense of the relative importance of the various needs is essential to making trade-offs correctly. For this activity, an internet based survey was made and sent out to many members of the Sámi

community in northern Sweden. The results from the survey were the main source for establishing the relative importance of the customer needs.

B.1.4 CONCEPTS GENERATION

The concept generation process begins with a set of customer needs and target specifications and results in a set of product concepts from which the team will make a final selection. Good concepts generation leaves the team with confidence that the whole spectrum of alternatives has been explored, and reduces the risk of stumbling upon a superior concept later in the development process or that a competitor will introduce a better product. [Ulrich and Eppinger, 2008]

Ulrich and Eppinger suggest a five-step method for developing product concepts. The method breaks complex problems into simpler sub problems, for which solution concepts will be identified. The space of solution concepts is then systematically explored, and integrated into a total solution. The five steps are shown in Figure 17 below.

FIGURE 17: THE FIVE-STEP CONCEPTS GENERATION METHOD.
(ADAPTED FROM ULRICH AND EPPINGER, 2008)

B.1.5 CONCEPT SELECTION

The concept selection was made systematically according to the methodology suggested by Pahl and Beitz (1996). In brief, this method compares concepts by giving points for how well each concept meets a specific and weighted property. The properties are weighted relative to each other, with points 0-2 where 0 denotes that the weighted property is less important than the one it is compared to, and 2 denotes that it is more important than the one it is compared to.

B.1.6 RESULTS

The Needfinding process resulted in a hierarchical list of customer needs with relative importance ratings. This list is shown in table 1 below.

Table 1. Hierarchical list of primary and secondary customer needs for the Communicator (abbreviated into CCR). Importance ratings for the secondary needs are indicated by the number of bullets (•), with ••• denoting critically important needs. Latent needs are denoted by a dagger (†).

The Communicator is optimal for outdoor use during all four seasons.

- ••† The CCR is waterproof.
- •• The CCR functions normally in all temperatures of use.
- •• The CCR power life is good in all temperatures of use.
- ••• The CCR operates normally in all weather.
- •† The CCR can be used in direct sunlight.
- ••• The CCR can be used daily in cold weather.

The Communicator offers reliable and simple communication.

- ••† The CCR can replace other communication products.
- ••• The CCR can be used in isolated areas.
- •• The CCR offers fast communication.
- •• The CCR can be used despite topographic difficulties.
- ••† The CCR is independent of network signal strength.
- ••† The CCR makes it possible for remote discussions.
- •• The CCR has long range.
- •• The CCR offers contact with the surrounding world.
- •• The CCR can send messages.

The Communicator is user friendly.

- ••• The CCR is intuitive and easy to use for everyone.
- ••† The CCR interface keeps the product clean.
- ••• The CCR interface can be used with gloves on.
- ••† The CCR offers peace and quiet.
- ••† The CCR can be used on different types of vehicles.
- ••• The CCR is useful in daily work.
- •• The CCR is useful in the spare time.
- •• The CCR has a big display.
- •• The CCR display offers good visibility and detail level.
- ••• The CCR is wireless.

The Communicator is easy to bring along.

- •• The CCR is easy to keep safe.
- •• The CCR is light-weight.
- •• The physical design of the CCR is clean.
- •• The CCR has no loose parts.

The Communicator is environment friendly.

- •• The CCR consumes a minimum of energy.
- ••• The CCR has long power life.
- ••

•† **The Communicator is compatible with other physical products.**
••† The CCR has wireless connections for display and interface.
•† The CCR can be connected to a computer for more functionality.
••† The CCR can be connected to an external screen.
The CCR can be used on different types of vehicles.

•• **The Communicator is affordable.**
Using the CCR continuously is cheap.

• **The Communicator has practical functions apart from those for communication.**
•† The CCR makes it possible to read newspapers electronically.
•• The CCR makes it possible to pay bills from remote places.
••† The CCR offers quick and easy access to information.
••† The CCR offers weather information.
••† The CCR offers visual tracking of its movement.
• The CCR can estimate reindeer movement.
•• The CCR reminds the user of activated functions.
The CCR can be integrated with products from a third party.

•• The CCR can be turned off.
The CCR can be powered by different sources.

•• **The Communicator offers security.**
••• The CCR can be located by another one.
••• The CCR can send emergency messages independent of network availability.
••• The CCR offers fast connection to emergency service centres.
The CCR offers a secure connection for emergencies.

•• **The Communicator makes it possible to communicate.**
•• The CCR makes it possible to make calls.
•• The CCR can browse the internet.
The CCR can send and receive e-mails.

••• **The Communicator lasts a long time.**
••• The CCR operates normally after shock and impact.
••• The CCR keeps in tough conditions.
••† The CCR is shockproof.
The CCR can be used and viewed during strong vibrations.

B.1.7 CONCEPTS

The two final concepts are called Plausus (latin for applause, for getting most points in the concept weighting process) and Levis (latin for light, for being a very small and light concept). They are shown in simple sketches in figure 3 and 4 below. Both concepts are under further development, and are presented in the defence of the Thesis project on December 15th at LTU.

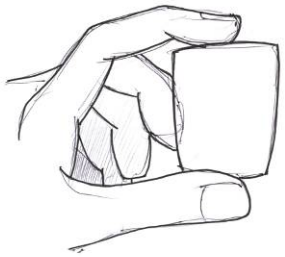

FIGURE 18: A SIMPLE SKETCH OF THE LEVIS CONCEPT

The Levis concept is a small module that can be connected to a computer or a newer mobile phone by USB or WLAN. The Levis contain all software and hardware required to send and receive data via DTN. It also contains a small satellite transmitter that can send a signal with exact position to an emergency service centre in case of emergency. To send or read incoming data the module must be connected to a computer or mobile phone. The Levis harvests kinetic energy from the user's movement.

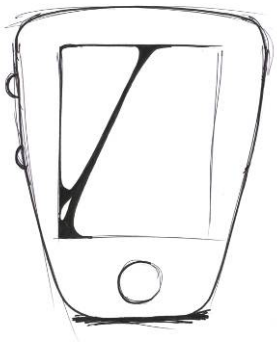

FIGURE 19: AN EARLY AND SIMPLE SKETCH OF THE PLAUSUS CONCEPT

The Plausus is a PDA concept adapted for harsh conditions. It has very few buttons to make the product easy to use with gloves on, for example. It can be connected to a computer for some more functionality. Just as the Levis, it contains a small satellite transmitter, and harvests kinetic energy as a supplement for the rechargeable batteries.