N4C

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

Draft for D7.2.2 N4C Laboratory Testing on Integrated Subsystems

SEVENTH FRAMEWORK
PROGRAMME

Pedro Nunes Institute
António Cunha, Francisco Barbosa
Postal address: Rua Pedro Nunes, 3030-199, Coimbra, Portugal
Tel: +351 239700934
Fax: +351 239700912
www.ipn.pt
cunha@ipn.pt

ABSTRACT (Max 400 word)

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 northern Sweden 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 explains how the testing and integration processes are planned, how is it going to be executed and what are the results of such processes. As it is explained later on this document, testing and integration of different faces from the same coin, and they cannot be taken apart, which results in a great interconnection between these.

Due date of deliverable: 31/12/2009 Actual submission date: 04/12/2009

Document history		
Status	**Date**	**Author**
Initial Draft based on DOW	04/12/2009	António Cunha, Francisco Barbosa
First draft circulated to consortium		
Feedback		
Submission to EC		

Dissemination level	
	Level
PU = Public	X
PP = Restricted to other programme participants (including the Commission Services).	
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).	

CONTENT

1. INTRODUCTION 6
1.1 - OVERVIEW 6
1.2 - ABBREVIATIONS 6
1.3 - REFERENCES 6
2. STRATEGY FOR THE INTEGRATION PROCESS 7
2.1 - INTEGRATION STRATEGY 7
2.2 - TESTING STRATEGY 7
2.3 – GLOBAL STRATEGY 8
3. PROCESS DEFINITION 9
3.1 - TYPES OF NODES 9
3.1.1 - PURE LEGACY NODE 9
3.1.2 - DTN-ONLY NODE 9
3.1.3 - CHAMELEON NODE 9
3.1.4 - GATEWAY NODE 9
3.2 - REQUIRED HARDWARE 10
3.2.1 - DTN ROUTER (A DTN-ONLY NODE) 10
3.2.2 - STANDARD PC (A GATEWAY OR PURE LEGACY NODE) 10
3.2.3 - NETBOOK (CHAMELEON NODE) 11
3.2.4 - PDA (CHAMELEON NODE) 11
3.2.5 - OTHER DATA MULES (CHAMELEON NODES) 11
3.3 – MODULES 11
3.3.1 - DTN2 11
3.3.2 - PROPHET / NSIM 11
3.3.3 - WEB CACHING 12
3.3.4 - DTN MAIL 12
4. TEST SCENARIO DEFINITION 12
4.1 – NETWORK'S NODES HARDWARE 13
5. NODES' ASSEMBLY 14
5.1 – GATEWAY *ASSEMBLY* 14
5.2 – DTN MULE 14
5.3 – DTN-ONLY NODE 15
5.4 – DTN-ROUTER ASSEMBLY 16
6. PERFORMED TESTS 17
WEB CACHING 17
MAIL SERVICE 18
6.1 - WEB CACHING 18
6.1.1 - ACCESS TO A CACHED WEB SITE 18
6.1.2 - ACCESS TO A NON-CACHED WEB SITE 18
6.1.3 - ACCESS TO PERIODICALLY-DELIVERED CONTENTS 18
6.1.4 - PERFORM A SEARCH 18
6.2 - MAIL SERVICE 18
6.2.1 - SEND AN EMAIL FROM THE DTN TO THE LI 18
6.2.2 - SEND AN EMAIL FROM THE DTN TO THE DTN 18
6.2.3 - RECEIVE AN EMAIL FROM THE LI, IN THE DTN 18
6.2.4 - SEND AND RECEIVE EMAILS WITH ATTACHMENTS 18
7. RESULTS OF TESTS 18

FIGURES INDEX

Figure 1 - Algorithm for integration and tests 8
Figure 2 - Test scenario no. 1 12
Figure 3 – Test scenario no. 2 13

EXECUTIVE SUMMARY

N4C – Networking for Communication Challenged Communities – is a project funded under European Seventh Framework Programme, initiated in 2008, that aims to use existing and novel technologies in vast remote areas with sparse population where traditional communications means are not viable.

In the heart of N4C is DTN, a network topology that allows computers in such network to communicate between them and with the traditional Internet users and machines independently of the delay that communications may have. Delays in DTNs are significantly bigger than in the Internet, which can vary from a few hours to several days, as opposed to few milliseconds of nowadays Internet. Such delays are, however, a small price to pay in such areas, if it can provide a reliable and best-effort service.

N4C comprehends a set of services and modules aimed at providing people not only a delayed Internet connection, but also a set of services that are thought to improve people's life and education, in an obvious way to promote eInclusion and eLearning. These services, however useful, can become even more powerful and useful if they can act and work together. With this purpose in mind, integration appears as a fundamental process, which target is to pick every module and integrate all them in a set of cohesive network nodes. Associated with this objective comes one more, which arises as a consequence of the integration process: the testing. So, when integrating modules, testing is a need and an advantage that WP7 can bring to the project, and both of these processes are being described in this document, as well as the results obtained from the plans here described.

1. INTRODUCTION

1.1 - OVERVIEW

This document was created under the scope of the N4C's work package 7, responsible for the system's integration and it is intended to be the second part of the document "D7.2.1 – Laboratory Testing on Integrated Subsystems". In that document, IPN described the several possible approaches to the integration that was intended to be made and choose the option that fitted best: the use of an integration platform, designated as D-Bus. However, as the project was developed, the consortium reflected about the implications that such integration method would have and the consortium as a whole decided that it was best to slightly change the integration method, under penalty of not achieving the desired goal in a reasonable time or with reasonable results.

So, in order to properly realize the integration, a new approach is being explored, consisting in taking all the modules that are being developed and test them, first alone and then in hardware running other modules. This process implies that not only the integration of the modules is made, but also an independent set of tests can be performed in an environment different from the one in which the module was developed.

Obeying to the refinement that the integration process suffered, this document will describe the actions already taken and the conditions on how integration will occur, as well as the initial results of laboratory tests.

1.2 - ABBREVIATIONS

CCR	Communications Challenged Region
DTN	Delay Tolerant Networking
LI	Legacy Internet

1.3 - REFERENCES

1. **Davies, Elwyn.** *D2.1 - N4C System Architecture v0.4.* Folly Consulting. 2009.
2. **Farrel, Stephen.** *D5.1 - N4C Node Design.* TCD / Intel. 2009.
3. **Cunha, António and Barbosa, Francisco.** *D7.2.1 - Laboratory Testing on Integrated Subsystems.* Instituto Pedro Nunes. 2009. Deliverable.
4. **Farrel, Stephen.** *D4.1 - N4C Generic DTN Documentation.* TCD / Intel. 2009.

2. STRATEGY FOR THE INTEGRATION PROCESS

The change that was decided to perform in the integration process leads to a modification in the strategy that was initially defined. As it has been explained in section 1.1, this change implies that, when performing the integration, it is necessary to perform tests at the same time, which, being under the WP7 context, needs to be outlined in the following two sections.

2.1 - INTEGRATION STRATEGY

Referring to section 6 from [3] – "Subsystem Test Planning" – and, more specifically, to the methodology that is going to be used, it is required to change the integration perspective from black-box to gray-box perspective, i.e., the integration teams will need to have some knowledge, albeit limited, of each module that is going to be integrated.

To accomplish this process, the partners working in the integration – IPN and ITTI – need to collect, from all developers, the sources of the software that is being developed, together with a set of information that is relevant to make it work and to understand a little bit of its internal structure and position in the "big picture".

The information collected from partners is going to be used in the process of getting things to work in hardware pieces. To do this, it is required to define first what kind of nodes are going to exist in laboratory tests, what is their role in the network, which hardware will they require and which software they need in order to run the modules properly, in an attempt to replicate, in laboratory, the nodes that are going to be deployed, but this time with some integration made.

2.2 - TESTING STRATEGY

After collecting the sources and information on each module, each one of them will be tested in a clean system, preferably with a fresh installation, so that its proper functioning can be assured. This step is where the gray-box perspective adopted by the integration teams come to play: integration teams will need to know a little about each module (dependencies, compilation methods, …) so installation can be possible. In this process, the contact with developers is fundamental, as they are the people who can help us the most.

Once the proper functioning of the module is guaranteed, the module passes to a "production platform", a network node already running with other modules. In this step, not only the behavior of the module in test is analyzed, but it is also analyzed other modules' behavior.

2.3 – GLOBAL STRATEGY

It is clear by now that strategies described in previous sections are highly entwined. To better comprehension, they are depicted in the figure below.

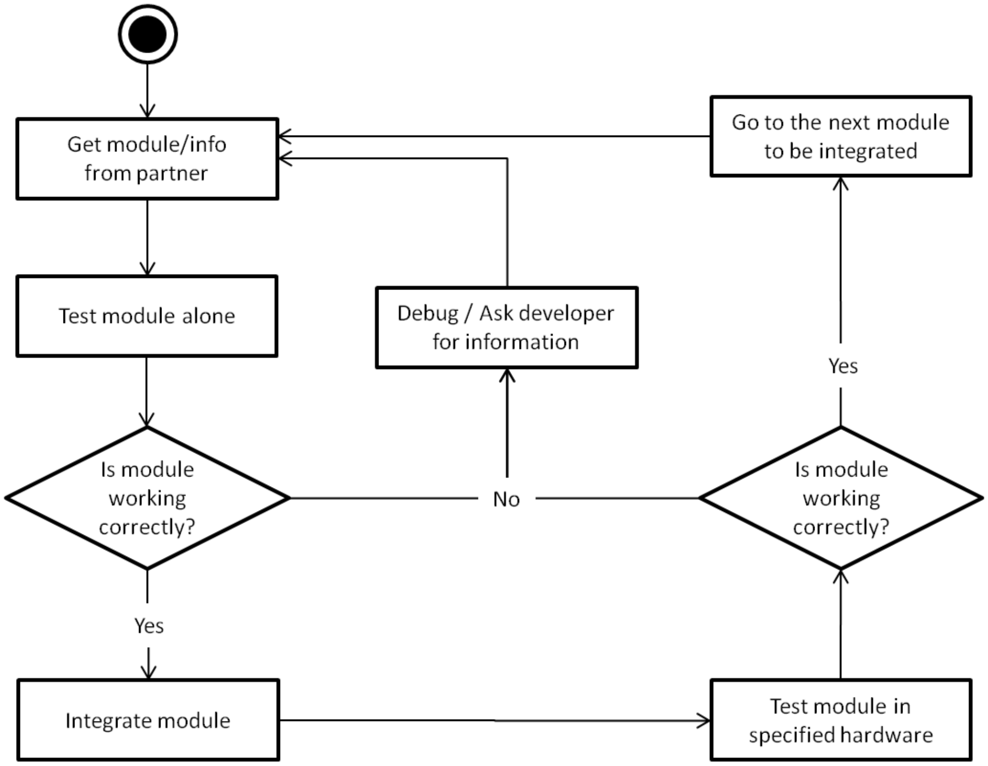
Get module/info from partner
Go to the next module to be integrated
Test module alone
Debug / Ask developer for information
Yes
Is module working correctly?
No
Is module working correctly?
Yes
Integrate module
Test module in specified hardware

Figure 1 - Algorithm for integration and tests

The entire process comprehends the following steps:

1. Definition of a lab test scenario;
2. Definition of which nodes will participate in the scenario defined in 1);
3. Definition of the modules that are going to be integrated;
4. Definition of use cases to test modules;
5. Perform tests on modules, alone and integrated.

3. PROCESS DEFINITION

3.1 - TYPES OF NODES

Integration teams have the interest to perform their tasks in scenarios that are as close as possible to the real-life use. To accomplish that, the network nodes used must be the same as the ones that are going to be used. According to [1] and [2], the nodes that are being used in N4C are divided by four main types.

3.1.1 - PURE LEGACY NODE

These types of nodes are standard nodes used within Legacy Internet (LI). Despite not being able of using DTN functionalities directly, they can operate in CCR enclaves without any modification in its operating system and/or protocols, although being restricted to existing applications that can operate unchanged in DTN regions, like the email application.

3.1.2 - DTN-ONLY NODE

Nodes expected to be in pure DTN regions, which do not need LI connectivity. These nodes use the DTN infrastructure and can be deployed as a router node, providing store-and-forward capabilities. These nodes can be applied in isolated sensor stations, or in places which travelers almost certainly will pass by. An example of a DTN-only node is a DTN router, a node that is designed to be used in summer villages as the core of a local network, and that may be powered through a battery or a solar panel.

3.1.3 - CHAMELEON NODE

These nodes are able to operate in LI, DTN or even in CCR enclaves and its behavior varies according to the place where the node is operating, but in a way that is transparent to users. According to its location, these types of nodes will use Internet protocols to communicate with other LI or CCR nodes whenever they are in LI or CCR enclaves, and will use DTN mechanisms when operating in CCR regions. An example of a chameleon node is the mules that are going to be used to carry information, which can be a PDA or a netbook.

3.1.4 - GATEWAY NODE

These gateway nodes include nodes operating in the boundaries between LI and CCR regions where DTN protocols are used and nodes in the boundaries between CCR enclaves and the rest of CCR regions. Gateways will require one or more logical pair of interfaces: one for LI protocols and the other for DTN protocols. Together with the pair of interfaces, it is also needed a mechanism to mediate communications between the different zones and different interfaces. The gateways nodes are standard computers, which do not have to face connection or power supply problems or limitations.

3.2 - REQUIRED HARDWARE

3.2.1 - DTN ROUTER (A DTN-ONLY NODE)

In [2], it is proposed a design to a DTN router composed of a single board computer (SBC), network, power supply and enclosure modules:

Part	Chosen hardware
SBC	Eurotech Proteus, Mikrotik RB411, or Kronton nanoETXexpress-SP
Network	Engenius EMP-8603 wireless card
Power Supply	Sunshine Solar FastFIX 20w12v (Solar Panel) Camden 5085329 7AH 12V gel
Enclosure	Eurobox IP66 cabinet

This node may run Ubuntu Linux 8.04 with the kernel 2.6.24-lpia[1] (for the PROTEUS board) or OpenWRT Kamikaze (for instance, with MikroTik board), and its main role it to act as hotspot to village users. In what concerns to applications, these nodes will primarily provide e-mail (act as MTA, with Postfix/Exim and Dovecot) and web (Apache and Squid) services. The DTN router will also provide DNS through Bind9. Configurations for a DTN router running Ubuntu 8.04 are available at http://basil.dsg.cs.tcd.ie/code/DTN-gateway-doc/file/566861cbee4d/build/dtnrouter%20build%20version%201.01. An interesting approach to another DTN node could also be to use the Asus WL-500G router with OpenWRT firmware.

3.2.2 - STANDARD PC (A GATEWAY OR PURE LEGACY NODE)

Standard PCs can serve two purposes in the network: act as a gateway or as a pure legacy node.

In order to configure a computer as a gateway, a document is available with some configuration at http://basil.dsg.cs.tcd.ie/code/DTN-gateway-doc/file/566861cbee4d/build/dtngateway%20build%20version%201.00.

[1] Low Power on Intel Architecture

3.2.3 - NETBOOK (CHAMELEON NODE)

Netbooks can act as data mules or as DTN nodes. The way these nodes can be configured is described in [4] and in the wiki, at the 2009 summer tests section. The equipment being used is Asus eeePC901, which configurations are available at http://basil.dsg.cs.tcd.ie/code/DTN-gateway-doc/file/566861cbee4d/build/dtnmule%20build%20version%201.00.

3.2.4 - PDA (CHAMELEON NODE)

The equipment that will act as a chameleon node and is being carried by hikers is a Nokia tablet, the N810, although N900 is being analyzed as an alternative. Nokia tablets will have combined behavior, since it will operate as a DTN node and as a mule. Its operating system is Maemo, a Linux distribution, and will have installed the "Hiker's Applications" and NORUT AutoDiscovery module, for ad-hoc communication between hikers. Further information on how to setup a tablet with all the applications is required so the testing team can properly setup these devices.

3.2.5 - OTHER DATA MULES (CHAMELEON NODES)

Much similar to the devices being used in helicopters during the tests, a SBC-like device can be used as a data mule. For this purpose, a configuration like the one presented in section 3.2.1 or even the Asus WL-500 option can be used.

3.3 – MODULES

3.3.1 - DTN2

This module is being developed by TCD, LTU and Folly, aimed to be run in Linux platforms and it is going to be part of all the nodes that operate in any part of the DTN infrastructure: DTN-only, chameleon and gateway nodes.

3.3.2 - PROPHET / NSIM

PRoPHET and NSIM are two modules grouped as one. In one side, we have the PRoPHET routing protocol, used for intermittently connected networks. In the other side, we have NSIM, which acts as a service that uses the protocol, and that is like a proof-of-concept of the protocol's functionalities.

3.3.3 - WEB CACHING

Web caching is the module that is responsible for taking DTN users' requests for a particular web page and retrieve the content contained in that same page, or group of pages. This modules has to, obligatory, be present at the gateway (in the LI) and, in the case of scenario 2 (Figure 3) in the DTN-router, that may act as a proxy-like system, or in the DTN nodes, in situations similar to the one in scenario 1 (Figure 2).

3.3.4 - DTN MAIL

Much similar to the traditional email service, DTN mail provides mailing functionality to users whether they are in a CCR or in the LI. Beyond the mail, this module allows users to send attachments, just like regular email services.

4. TEST SCENARIO DEFINITION

The scenarios in which we want to test the integration are represented in the following figures, in which there is a computer operating in a CCR, downloading from and uploading data through a data mule, which is periodically travelling between two points. In the other end we have a gateway, bridging the CCR enclave and the Legacy Internet, making the translation between bundles and packets. This is the scenario depicted in figure 1, which is slightly modified in scenario number 2, in which the DTN-only node communicates with the mule with a DTN-router in the middle.

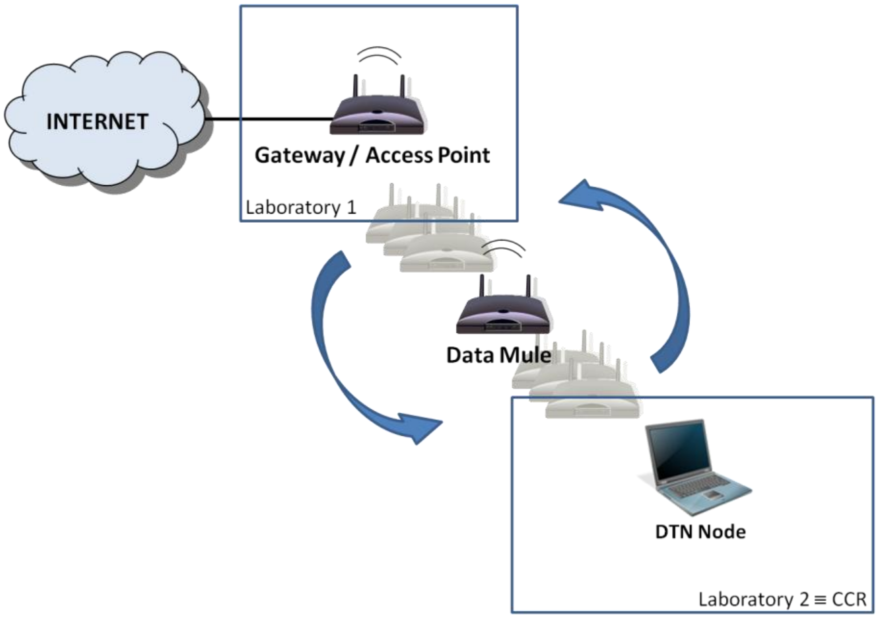

Figure 2 - Test scenario no. 1

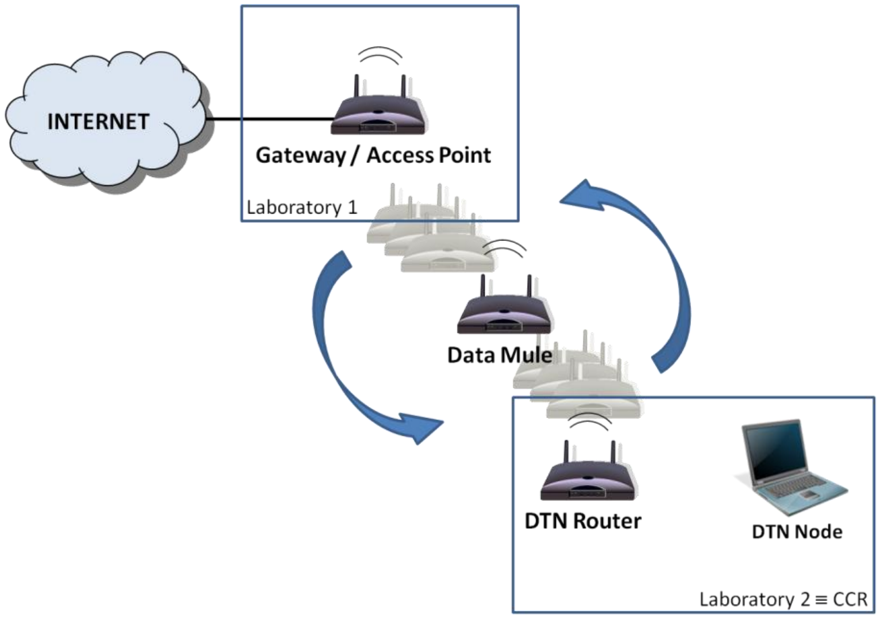

Figure 3 – Test scenario no. 2

4.1 – NETWORK'S NODES HARDWARE

- Gateway/Access Point

- Data Mule

- DTN Router

- DTN Node

5. NODES' ASSEMBLY

5.1 – GATEWAY *ASSEMBLY*

Kernel	2.6.24-16-lpia
Modules	DTN2, HTMLrequester, LTPlib, oasys, PRoPHET
Responsible Partner(s)	LTU / TCD / Folly
Source Location	http://basil.dsg.cs.tcd.ie/code/DTN2/ http://basil.dsg.cs.tcd.ie/code/HTMLrequester/ http://basil.dsg.cs.tcd.ie/code/LTPlib/ http://basil.dsg.cs.tcd.ie/code/oasys/ http://basil.dsg.cs.tcd.ie/code/PRoPHET/
Hardware Requirements	Two network interfaces: one (LAN) for LI, other (WLAN) for the DTN side.

Software Requirements	• **DTN2**
	`tcl8.5-dev, tclreadline, tclx8.4-dev, tcllib, libxerces28-dev, libdb4.6-dev`
	• **PRoPHET**
	`qt4- dev-tools`
	• **LTP-T**
	`cgic205, libssl-dev, openssl`
	• **Development**
	`g++-4.2, libstdc++6-4.2-dbg, gcc`
	• **Benchmarking, ssh, firewall, system info**
	`info, lmbench, mercurial, lshw, ssh, ntp, iptables, postfix, dovecot-common, squid, dhcp3-server, libapache2-mod-php5, php5, php5-dev, mysql-server, mysql-client, libapr1-dev, apache2-threaded-dev, libmysqlclient15-dev, libapache2-mod-auth-mysql, automake1.9`

Platform in which the module is going to be tested	

5.2 – DTN MULE

Kernel	2.6.24-16-lpia

Modules	DTN2, HTMLrequester, LTPlib, oasys, PRoPHET
Responsible Partner(s)	LTU / TCD / Folly
Source Location	http://basil.dsg.cs.tcd.ie/code/DTN2/ http://basil.dsg.cs.tcd.ie/code/HTMLrequester/ http://basil.dsg.cs.tcd.ie/code/LTPlib/ http://basil.dsg.cs.tcd.ie/code/oasys/ http://basil.dsg.cs.tcd.ie/code/PRoPHET/
Hardware Requirements	Wi-Fi card

Software Requirements	• **DTN2**
	`tcl8.5-dev, tclreadline, tclx8.4-dev, tcllib, libxerces28-dev, libdb4.6-dev`
	• **PRoPHET**
	`qt4- dev-tools`
	• **LTP-T**
	`cgic205, libssl-dev, openssl`
	• **Development**
	`g++, libstdc++6-4.2-dbg, gcc`
	• **Benchmarking, ssh, firewall, system info**
	`info, lmbench, mercurial, lshw, ssh, ntp, iptables, postfix, dovecot-common, squid, apache2, dhcp3-server, libapache2-mod-php5, php5, php5-dev, automake1.9`

Platform in which the module is going to be tested	

5.3 – DTN-ONLY NODE

Kernel	2.6.24-16-lpia
Modules	DTN2, HTMLrequester, LTPlib, oasys, PRoPHET, DTNmailex
Responsible Partner(s)	LTU / TCD / Folly
Source Location	http://basil.dsg.cs.tcd.ie/code/DTN2/ http://basil.dsg.cs.tcd.ie/code/DTNmailex/ http://basil.dsg.cs.tcd.ie/code/HTMLrequester/ http://basil.dsg.cs.tcd.ie/code/LTPlib/ http://basil.dsg.cs.tcd.ie/code/oasys/

	http://basil.dsg.cs.tcd.ie/code/PRoPHET/
Hardware Requirements	

Software Requirements	• **DTN2**
	`tcl8.5-dev, tclreadline, tclx8.4-dev, tcllib, libxerces28-dev, libdb4.6-dev`
	• **PRoPHET**
	`qt4- dev-tools`
	• **LTP-T**
	`cgic205, libssl-dev, openssl`
	• **Development**
	`g++-4.2, libstdc++6-4.2-dev, gcc-4.2-doc, lib32stdc++6, libgcc1-dbg, libstdc++6-4.2-dbg`
	• **Development (64-bit architecture)**
	`g++-4.2-multilib, g++-multilib, gcc-4.2-multilib, lib32gcc1, lib32gomp1, libc6-dev-i386, libc6-i386`
	• **Benchmarking, ssh, firewall, system info**
	`info, lmbench, mercurial, lshw, ssh, ntp, iptables, postfix, dovecot-common, squid, apache2, dhcp3-server, libapache2-mod-php5, php5, php5-dev, automake1`

Platform in which the module is going to be tested	

5.4 – DTN-ROUTER ASSEMBLY

Kernel	2.6.24-16-lpia
Modules	DTN2, HTMLrequester, LTPlib, oasys, PRoPHET
Responsible Partner(s)	LTU / TCD / Folly
Source Location	http://basil.dsg.cs.tcd.ie/code/DTN2/ http://basil.dsg.cs.tcd.ie/code/HTMLrequester/ http://basil.dsg.cs.tcd.ie/code/LTPlib/ http://basil.dsg.cs.tcd.ie/code/oasys/ http://basil.dsg.cs.tcd.ie/code/PRoPHET/

Hardware Requirements	
Software Requirements	• **DTN2** tcl8.5-dev, tclreadline, tclx8.4-dev, tcllib, libxerces28-dev, libdb4.6-dev • **PRoPHET** qt4- dev-tools • **LTP-T** cgic205, libssl-dev, openssl • **Development** g++-4.2, libstdc++6-4.2-dev, gcc-4.2-doc, lib32stdc++6, libgcc1-dbg, libstdc++6-4.2-dbg • **Development (64-bit architecture)** g++-4.2-multilib, g++-multilib, gcc-4.2-multilib, lib32gcc1, lib32gomp1, libc6-dev-i386, libc6-i386 • **Benchmarking, ssh, firewall, system info** info, lmbench, mercurial, lshw, ssh, ntp, iptables, postfix, dovecot-common, squid, apache2, dhcp3-server, libapache2-mod-php5, php5, php5-dev, automake1
Platform in which the module is going to be tested	

6. PERFORMED TESTS

To infer about the modules, a set of use cases has been developed to test the basic integration that has been performed in the scenario described. All the following use cases, grouped by the service (the following services are based in the functionalities provided by the modules described in sections 3.3.3 and 3.3.4) that is being tested, will have its results commented in section 7 of this document.

WEB CACHING

- Access to a cached web site;

- Access to a non-cached web site;
- Access to periodically-delivered contents;
- Perform a search.

MAIL SERVICE

- Send an email from the DTN to the LI;
- Send an email from the DTN to the DTN;
- Receive an email from the LI, in the DTN;
- Include attachments.

6.1 - WEB CACHING

6.1.1 - ACCESS TO A CACHED WEB SITE

6.1.2 - ACCESS TO A NON-CACHED WEB SITE

6.1.3 - ACCESS TO PERIODICALLY-DELIVERED CONTENTS

6.1.4 - PERFORM A SEARCH

6.2 - MAIL SERVICE

6.2.1- SEND AN EMAIL FROM THE DTN TO THE LI

6.2.2 - SEND AN EMAIL FROM THE DTN TO THE DTN

6.2.3 - RECEIVE AN EMAIL FROM THE LI, IN THE DTN

6.2.4 - SEND AND RECEIVE EMAILS WITH ATTACHMENTS

7. RESULTS OF TESTS