Networking fc ies:
Architecture, Test Beds and Innovative Alliances
Contract no: 223994

M7.5
Deployment of the Integration Platform

SEVENTH FRAMEWORK PROGRAMME

Pedro Nunes Institute
António Cunha, Paulo Freitas, Luís Coelho, Lokesh Kumar
Postal address: Rua Pedro Nunes, 3030-199, Coimbra, Portugal
Tel: +351 239700934
Fax: +351 239700912
www.ipn.pt
cunha@ipn.pt, freitas@ipn.pt

ABSTRACT (Max 400 word)

Starting in May 2008, N4C is a 36 month research project funded by the Seventh Framework Programme (www.cordis.lu/fp7). This is a cooperation project between end-users, in Swedish Lapland and Kočevje region in Slovenian mountain, and technological partners (RTD performers and companies). Under the scope of the project we will design and experiment with an architecture infrastructure and applications in field trials and build two test beds.

This document, which is relative to milestone 7.5 described in the Description of Work of the project, is one of the outcomes of the integration and testing process that has been made under WP7 tasks. In the present document it will be discussed topics related with the deployment of the Integration Platform such as its features, advantages, dependencies that may block its operation, and how the user fits on the usage scenarios. Moreover, throughout this document will be addressed topics such as: the objective of this milestone, the importance for the N4C project, specification of the integration platform or future work.

Due date of deliverable: 31/03/2011 Actual submission date: 08/04/2011

Document history		
Status	**Date**	**Author**
Initial Draft based on DOW	08/04/2011	António Cunha, Paulo Freitas, Luis Coelho, Lokesh Kumar
First draft circulated to consortium		
Feedback		
Submission to EC		

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).	

CONTENT

INTRODUCTION

The objective of the *Networking for Communications Challenged Communities: Architecture, Test Beds and Innovative Alliances* (N4C) project is to provide network services to remote communities. This raises several problems because it is not economically viable to construct infrastructures in order to provide them with the usual network service. Nowadays the access to information is very important to the development of a community and improves significantly their quality of life. In order to overcome this problem N4C project uses Delay Tolerant Networks (DTN) connectivity to establish a virtual connection to these communities.

DTN is an approach to computer network architecture that seeks to address the technical issues in heterogeneous networks that may lack continuous network connectivity. A DTN Network provides a set of features that may overcome the known problems of the Legacy Internet as failing when there is not a continuous connection, solving this problem because it does not requires a continuous connection.

Each data package that travels within a DTN network is named as bundle. The difference between a bundle and a typical TCP or UDP packet is that the bundles, when there is not any connection available, wait within a DTN node for another DTN node to be available in order to be forwarded until it reaches its destination. In a situation where there is not a connection the heterogeneous networks, like the *World Wide Web*, usually fail to deliver the information.

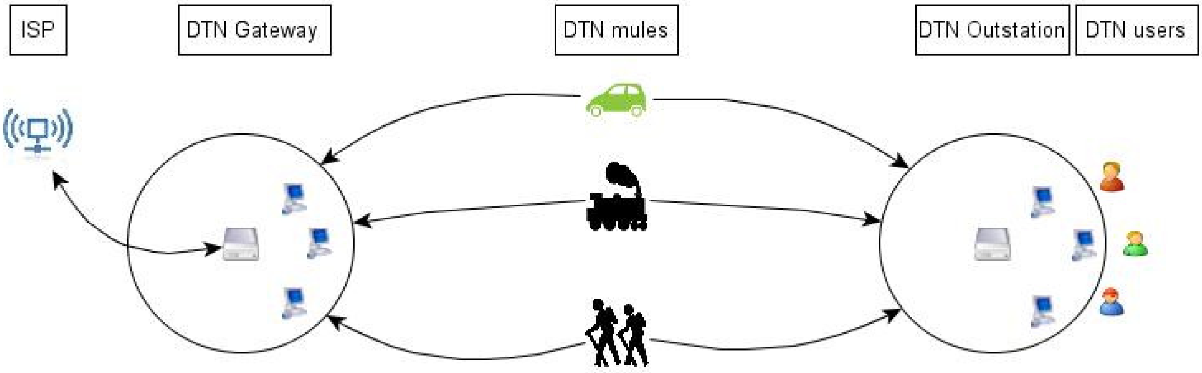


Figure 1 - N4C scheme

In Figure 1 can be seen the N4C approach to solve the problem of the network services in remote communities. The DTN Gateway is the network node that has a connection to a Internet Service Provider (ISP) and does the bridge between the DTN network and the legacy internet. The virtual connection between the DTN Gateway and the DTN Outstation, that will be on the remote community, is done by DTN mules that travel between the networks. Anything that can carry a running DTN and travels between the DTN networks, like hikers, trains or helicopters, can be considered as a

DTN mule. The DTN Outstation is the access point to the remote community and all the bundles that the DTN users send or receive go through this node. When the DTN user sends a bundle it will wait in the DTN Outstation till a DTN mule to be in range.

There are several applications that were done in the N4C project in order to provide network services to the remote communities. These applications include for example HTMLrequester, Pymail, Meteorological service, Not So Instant Messenger and Hicker's PDA.

The HTMLrequester application allows the DTN user to request web pages and put them available in cache in the DTN Outstation; Pymail application gives to the DTN users the ability to send and receive e-mails; and the Meteorological service provides to the DTN users meteorological data from meteorological stations, one in Veliki Lonik and the other in Nožice. The Not So Instant Messenger it is, like the name suggests, a messenger service that runs upon the DTN layer that allows users to exchange messages, even considering the delay that may exists during the message deliver. These applications will be explored further in this document.

In order to make all these applications work together, an integration platform was developed to provide to the end users a single interface that aggregates all of these applications, briefly explained before. In Figure 2 - Initial screen of the Integration Platform

can be seen the initial screen of the Integration Platform where the services provided by the N4C project are.

Beside the DTN user, it is still needed to set up the DTN Outstation, one or more DTN mules and the DTN gateway in order to have the full network chain completed. This document is intended to be a guideline for deploying all the N4C applications and the setting up of the entire N4C network.

This guide will provide a step-by-step installation to explain the installation procedure and to make it more simple and intuitive. All the examples and configurations provided are merely examples and can be changed by the users. After this introduction, follows the prerequisites of these applications followed by a software overview and detailed feature description. The guidelines for deployment are next followed by examples of configuration and testing. To finalize this document some final conclusions, recommendations and references are given to the user.

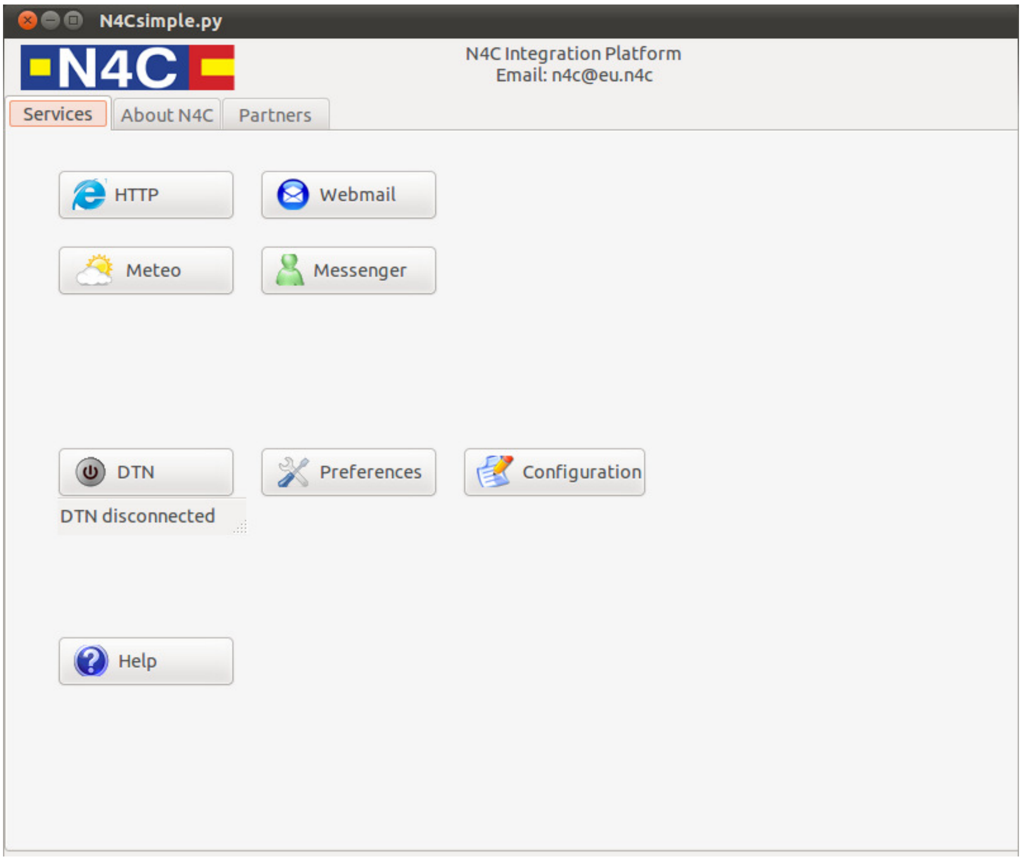


Figure 2 - Initial screen of the Integration Platform

PREREQUISITES

ASSUMPTIONS

It is assumed that some of the stakeholders that is installing this application, has minimal experience and knowledge in Linux OS in spite of this process being designed to be the most simple as possible. Is advised that the basic packages of the Linux OS installation are updated. This installation provides a local repository of the needed packages but, in some cases, it may not be sufficient.

RISKS

There are not any documented risks for the users' machine. If an unsuccessful installation (with errors) it breaks the installation process and returns error information to the user. After these errors are solved the installation process can restart without any problems. Usually the distributions of Linux OS are very secure because they do not let the user write in places that they should not write, so this decreases the risks for the users' machine.

CONFLICTS

There is a documented conflict with AVAHI. AVAHI is a free zeroconf implementation, including a system for multicast DNS/DNS-SD service discovery. AVAHI allows programs to publish and discover services and hosts running on a local network with no specific configuration. For example, a user can plug their computer into a network and AVAHI automatically finds printers to print to, files to look at and people to talk to, as well as advertising the network services running on the machine. The problem in using AVAHI is that it interferes with socket communications. All the N4C applications rely on socket communication so is unwanted that this process suffers interferences from other processes.

There is another problem that is related with the Apache version. It is desired that Apache has the 2.2 version because a latest version can generate problems on the DTN Gateway and DTN Outstation when using HTMLrequester.

HARDWARE AND SOFTWARE REQUISITES

All the components of the N4C scheme (DTN Gateway, DTN mule, DTN Outstation and DTN user) are recommended to have Intel's x86 architecture (32 or 64 bits) because they were developed with this architecture. If they have another architecture some applications could be unable to run. The DTN mule is different because it only has to carry a running DTN, so the architecture needed in this case could be different. This is approached on the DTN mule requisites chapter.

The applications of N4C do not have very high requisites. Follows all the requisites for the entire network scheme.

DTN Gateway requisites

The installation on the DTN Gateway requisites can be seen in Table 1.

	Minimum	Recommended
Processor	700 MHz	1 GHz
RAM memory	256 MB	1 GB
Hard disk	3 GB	8 GB

Table 1 - DTN Gateway requisites

The DTN Gateway must have a network connection, a graphics card able to do at least 1024x768 resolution and wireless capability so that the DTN mules do not need to have a physical connection to the network.

In terms of software there are no requirements because N4C provides a ISO file with a Ubuntu 10.10 installation and all the applications already installed. This is done because this should be a dedicated machine and the need for human intervention should be has minimal has possible. All the applications needed should be running at all time waiting for DTN mules to pass by.

DTN Mule requisites

The DTN mule is a special case because it only needs to carry a running DTN. Because the DTN mule was idealized to be mobile, in order to travel between the remote areas and the DTN Gateway, it has low requisites, so that simple machines like mobile phones have capability to be a DTN mule. The DTN mule can also be implemented on means of transportation that regularly travel between these sites like a train or a bus that can carry a small PC like for example Asus ePC that will serve as a DTN mule. So, the DTN mule requisites can be seen in Table 2.

	Minimum	Recommended
Processor	150 MHz	400 MHz
RAM memory	128 MB	256 MB
Hard disk	1 GB	2 GB

Table 2 - DTN Mule requisites

The DTN mule must have wireless capability in order to exchange bundles with other DTNs. These bundles that are exchanged between the DTNs have a very long expiration date, one year usually, and they are stored even when the DTN or the device that is carrying it are off line.

The DTN software has two versions, DTN1 and DTN2. DTN1 was tested on x86 and StrongARM architectures but this version is not the one that N4C uses. Until now DTN2 was only tested on x86 architecture so this is the safer choice but others architectures could work since DTN1 worked in other architectures and the DTN software is C based so almost any architecture should be able to run it. There is also a new DTN being developed that is based on Java so it will run in any architecture but at this time this is not available. In terms of software requirements DTN2 was tested on Linux and Mac OS without any problems so any software that is based on these should work perfectly.

DTN Outstation requisites

The DTN Outstation will serve as a local repository to all the applications of N4C. Since the local community does not have access to the internet all the things needed in order to have a successful installation should be available on the DTN Outstation. So, in order to set up a DTN Outstation the requisites are in Table 3.

	Minimum	Recommended
Processor	700 MHz	1 GHz
RAM memory	256 MB	1 GB
Hard disk	10 GB	20 GB

Table 3 - DTN Outstation requisites

The difference between the DTN Gateway is on the hard disk space because it has to hold all the repository of the applications. Beyond the local repository the DTN Outstation also has to store all the cached HTML pages, with the maximum size of 100 MB each because of the limitation of the bundle size that the DTN can carry. The DTN Outstation serves and replies to the requests of a entire community so the space of the Hard disk is vital to extend the period of intervention. The DTN Outstation must have wireless capability in order to be easily accessible to all the DTN users and the DTN mules.

Like in the DTN Gateway, in terms of software there are no requirements because N4C also provides an ISO file with an Ubuntu 10.10 installation and all the applications already installed. Also like DTN Gateway the DTN Outstation is a dedicated machine that requires minimum human intervention in spite of having the limitation of the Hard disk space. On the DTN Outstation is critical that all the applications run at all time because it has to store and satisfy all the requests from the users and wait for a DTN mule to be available in order to forward the stored bundles and to receive responses to the pending requests.

DTN User requisites

The DTN user machine should have a Linux based Operating System (OS) because all the N4C applications are based on this OS. So the requisites for the DTN user will be the ones for a small Linux installation like in Table 4.

	Minimum	Recommended
Processor	400 MHz	1 GHz
RAM memory	128 MB	1 GB
Hard disk	1 GB	8 GB

Table 4 - DTN User requisites

The minimum requisites present in this table are for a very small and basic Linux installation so is better to have at least the recommended requisites. The DTN user should also have wireless capability in order to link itself to the DTN user. Beyond the Linux installation the DTN user should have a email client software like Evolution (recommended), Thunderbird or some other software similar to these.

Other requisites

In order to make the information transport process as easier as possible is advised to set up wireless local networks on the area of the DTN Outstation and the DTN Gateway. This means that should be a router creating a wireless network in these areas, and, in the case of the DTN Gateway, with access to the internet.

DETAILED FEATURE DESCRIPTION

With this remote communities can have access to information and communications means with a very low investment since the infrastructure needed are very few and at very low cost, or even none, to the DTN user. The DTN technology is the base of all the applications of N4C and because DTN is a relatively new technology it has much to be improved (for instance, the maximum size of the bundles it is a very big limitation), so the applications provided are confined and somehow limited by the development of the DTN technology. Next we will present all the software, with all applications, features and examples of use, that were developed in N4C.

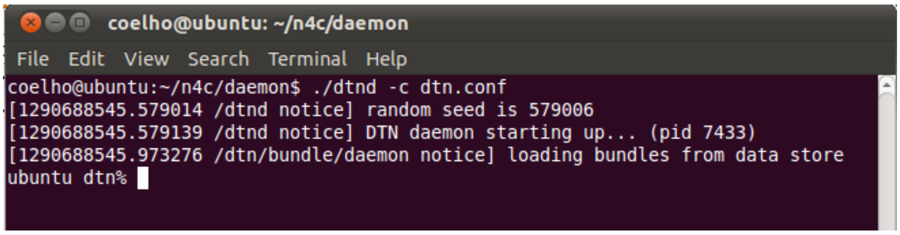
DTN

Nowadays the ways that the machines communicate rely on the fact that the latency of the communication channel, usually the internet, is almost none. This is a given fact for most of the time and for most of the applications, but sometimes, this low latency communication channel fails and some important packets are lost. In some applications is most imperative that all the packets arrive to their destination no matter how late.

Delay Tolerant Networking is a new form of networking that can handle delays and disruptions in traffic. The "normal" internet handles delays and disruptions in a different way of the DTN because when there is a delay on the delivery of a package usually the protocols that the internet uses just repeat the sending of this package and wait for it to arrive. The DTN works in a different way.

The original ideas for DTN where developed by among others Dr. Vinton G. Cerf, known as the father of the Internet. The Internet requires end-to-end connectivity, which limits its use for instance in space communications. Thus, Dr. Cerf developed with a group including NASA staff, the basis of the DTN technology. This means that for instance where masts are not possible to put close enough for a continuous network, DTN can be a bridge that saves traffic that is interrupted and that by using DTN computer-to-computer networks can be set up even where the population is small. Has the N4C project is intended for these small communities DTN is a perfect technology to carry information between DTN Gateway and DTN Outstation.

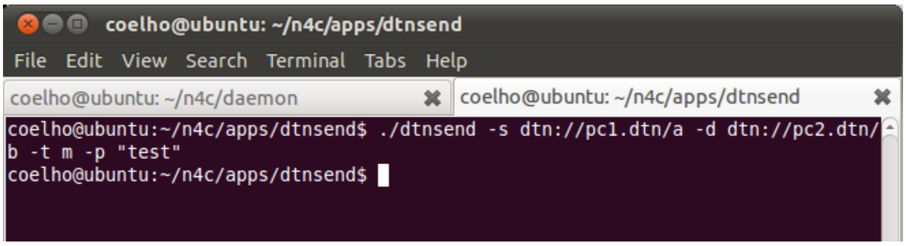
As already been said the DTN technology is the base for all the modules of the N4C project but it has also some applications within it. In Figure 3 we can see a DTN running in the command line above which all the applications of N4C run.


coelho@ubuntu: ~/n4c/daemon
File Edit View Search Terminal Help

```
coelho@ubuntu:~/n4c/daemon$ ./dtnd -c dtn.conf
[1290688545.579014 /dtnd notice] random seed is 579006
[1290688545.579139 /dtnd notice] DTN daemon starting up... (pid 7433)
[1290688545.973276 /dtn/bundle/daemon notice] loading bundles from data store
ubuntu dtn%
```

Figure 3 - Running DTN daemon

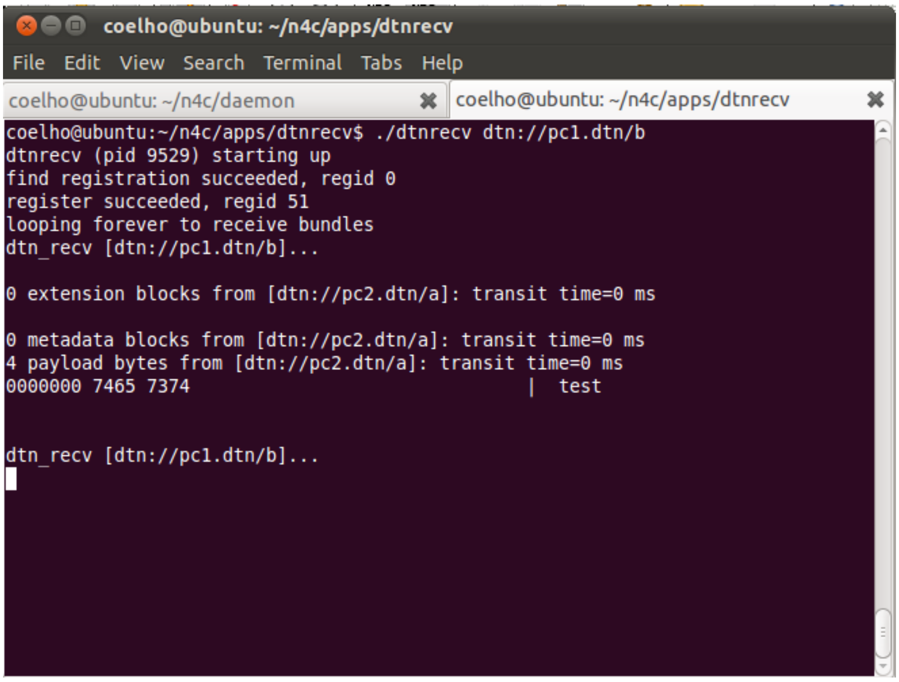
DTN applications are very useful in machine communications because they provide for instance a way to copy entire folders from another machine, send files or messages between these machines or simply pinging other machines. These look like very simple applications but they have the very big advantage of being able to do these things without a direct connection between them which is very important and useful in some situations.


coelho@ubuntu: ~/n4c/apps/dtnsend
File Edit View Search Terminal Tabs Help
coelho@ubuntu: ~/n4c/daemon
coelho@ubuntu: ~/n4c/apps/dtnsend

```
coelho@ubuntu:~/n4c/apps/dtnsend$ ./dtnsend -s dtn://pc1.dtn/a -d dtn://pc2.dtn/
b -t m -p "test"
coelho@ubuntu:~/n4c/apps/dtnsend$
```

Figure 4 - Send message through the DTN

In Figure 4 there is an example of how to send a simple message through the DTN on the command line. This message will wait inside the DTN waiting for a connection to the destination to be available. The receiving part can be seen on Figure 5. In order for the N4C project to run smoothly all the elements of the network have to have a running DTN because the DTN will be like a "transport layer" of the N4C network. This is a simple example of the utility of the DTN beside the use that it has inside the N4C project. On the provided DTN installation all these commands will be bash commands and the DTN could be running on background upon the initialization of the OS (dtnd process), but we will get to these details later in the document.


coelho@ubuntu: ~/n4c/apps/dtnrecv
File Edit View Search Terminal Tabs Help
coelho@ubuntu: ~/n4c/daemon
coelho@ubuntu: ~/n4c/apps/dtnrecv

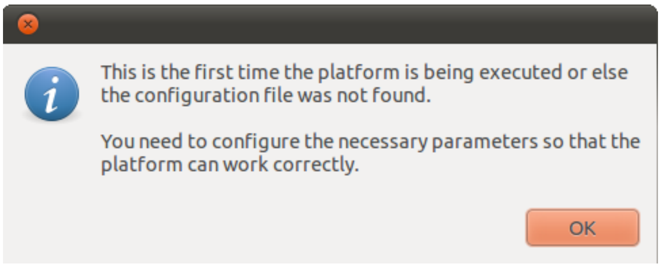
```
coelho@ubuntu:~/n4c/apps/dtnrecv$ ./dtnrecv dtn://pc1.dtn/b
dtnrecv (pid 9529) starting up
find registration succeeded, regid 0
register succeeded, regid 51
looping forever to receive bundles
dtn_recv [dtn://pc1.dtn/b]...

0 extension blocks from [dtn://pc2.dtn/a]: transit time=0 ms

0 metadata blocks from [dtn://pc2.dtn/a]: transit time=0 ms
4 payload bytes from [dtn://pc2.dtn/a]: transit time=0 ms
0000000 7465 7374                                 |  test


dtn_recv [dtn://pc1.dtn/b]...
```

Figure 5 - Receive message through the DTN

N4C INTEGRATION PLATFORM

The objective of the N4C Integration platform is to provide a easy and simple way for the DTN user to use all the modules. Other objective of the Integration Platform is to diminish the necessity of knowledge of the DTN user, in order for the service that N4C provides being accessible to almost all persons from the remote communities.

The first time that the platform is initialized the DTN user will receive a warning message that warns the DTN user that the platform is being initialized for the first time or for the absence of a configuration file. This warning message is displayed in Figure 6.

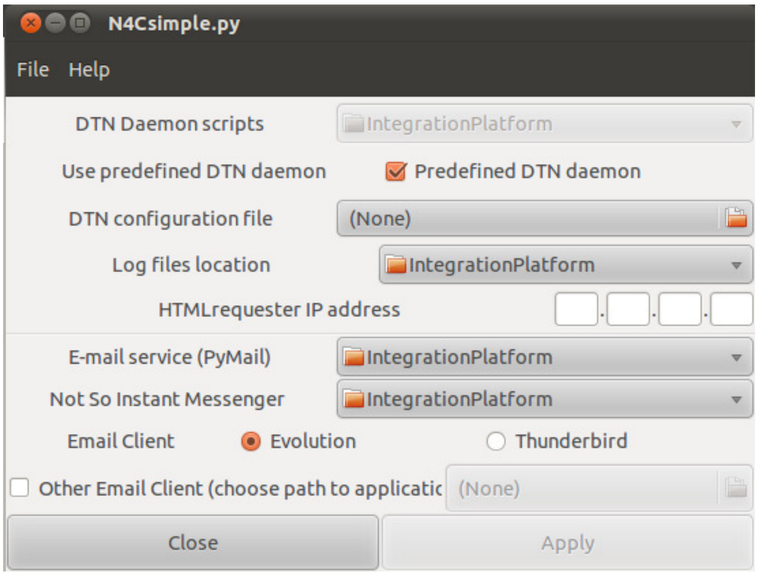

This is the first time the platform is being executed or else the configuration file was not found.

You need to configure the necessary parameters so that the platform can work correctly.

Figure 6 - Initial warning message

When the platform is initialized it will look for a configuration file and if this configuration file is not found it will display this message. If this warning message appears after the first time that the platform is initiated it means that there was some kind of error with the saved configuration file (it could have been deleted or corrupted).

When the DTN user presses the OK button the preferences window will be displayed in order for the user to make all the necessary configurations. This initial preferences window can be seen in Figure 7.



DTN Daemon scripts: IntegrationPlatform

Use predefined DTN daemon: ☑ Predefined DTN daemon

DTN configuration file: (None)

Log files location: IntegrationPlatform

HTMLrequester IP address: . . .

E-mail service (PyMail): IntegrationPlatform

Not So Instant Messenger: IntegrationPlatform

Email Client: ● Evolution ○ Thunderbird

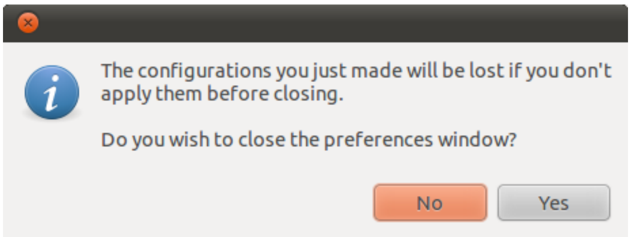
□ Other Email Client (choose path to applicatic (None)

Figure 7 - Initial preferences window

This preferences window has some fields to the DTN user to configure. If the user selects the predefined DTN daemon he will be using the DTN installation provided by N4C, but if he wants to use another installed DTN he cans by deselecting this option and selecting the folder where is located the DTN Daemon scripts.

DTN requires a configuration file so the next configuration needed is the location of this DTN configuration file. The DTN daemon will run on background when the DTN is initiated but to do this the DTN needs a log file where all the logs will be kept so the user must provide a folder to store this file. The next information needed is the HTMLrequester IP address. This IP address is the address of the DTN outstation because the DTN Outstation provides cached internet service. Next are the information about the E-mail service and the Not So Instant Messenger. The information required for the e-mail service is the location of the Pymail scripts and the e-mail client that the DTN user wants to use. The Not So Instant Messenger information needed is the location of Prophet executable file. The Prophet software has a tab with the Not So Instant Messenger allowing the DTN user to send messages to other DTN users.

If the Close button is pressed without having clicked on Apply the user can quit the preferences configuration without saving them, but a warning will be displayed in case of a involuntary click on it. This warning message can be seen in Figure 8. If No button is clicked it will return to the preferences window and if the Yes button is clicked the preferences window will definitely closed and the configurations lost.



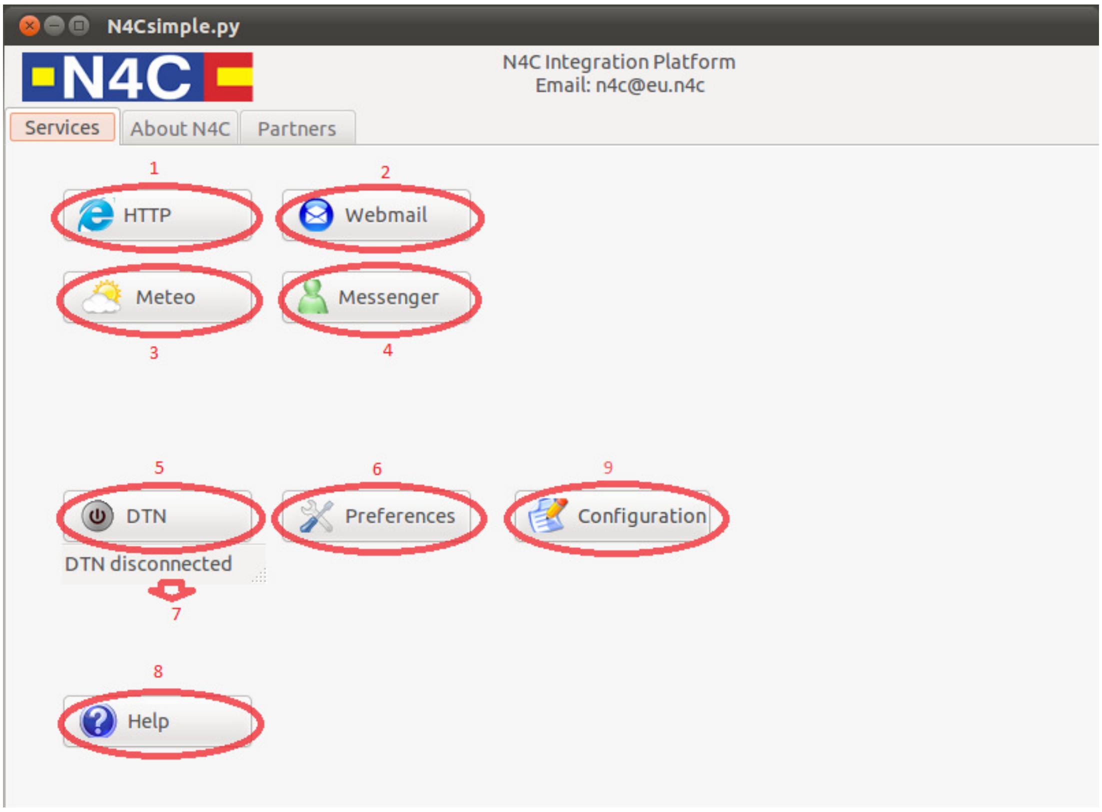
The configurations you just made will be lost if you don't apply them before closing.

Do you wish to close the preferences window?

Figure 8 - Quit preferences warning

The Apply button is to save all the configurations in a configuration file that will be retrieved every time that the platform is initialized. After this button is clicked the user can press the Close button without receiving any warnings and without losing the information. There are two drop down menus at the top of the window, File and Help. The File menu has options that allow the user to quit or save the configurations and the Help menu opens a web page with help content.

After all the configurations are done the N4C Integration Platform opens and the DTN user can use the provided services. In order to explain the N4C Integration Platform we will use the support of Figure 9.



N4Csimple.py

N4C Integration Platform
Email: n4c@eu.n4c

DTN disconnected

Figure 9 - Highlighted N4C Integration Platform

The button marked with 1 is the HTMLrequester button. When this button is pressed it opens a new tab where the HTMLrequester interface will appear.

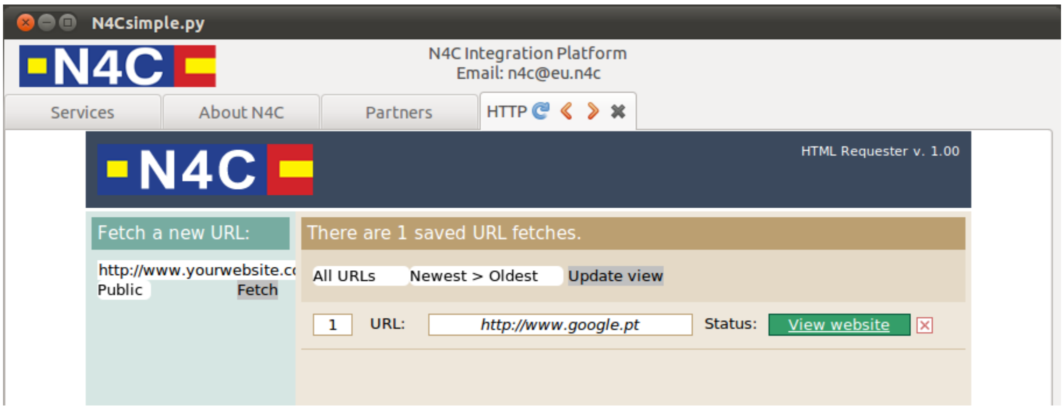


Figure 10 - HTMLrequester tab

As can be seen in Figure 10 the tab from the HTMLrequester is different from the other tabs. This tab will work as a browser allowing the DTN user to navigate the requested pages from here without the need of a web browser being able to refresh the page or pass to the next or previous page. The HTMLrequester web page is located at the DTN Outstation, being provided by an Apache server, and if the DTN Outstation is not available or out of reach it will displayed a blank page. More of HTMLrequester will follow.

In order to use the webmail service in the button 2 the DTN and Pymail scripts must be running. If this button is pressed without this being running it will not open the client email and will appear a warning that can be seen in Figure 11.

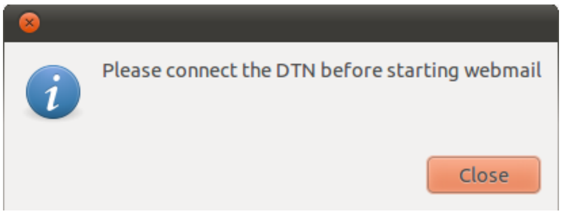
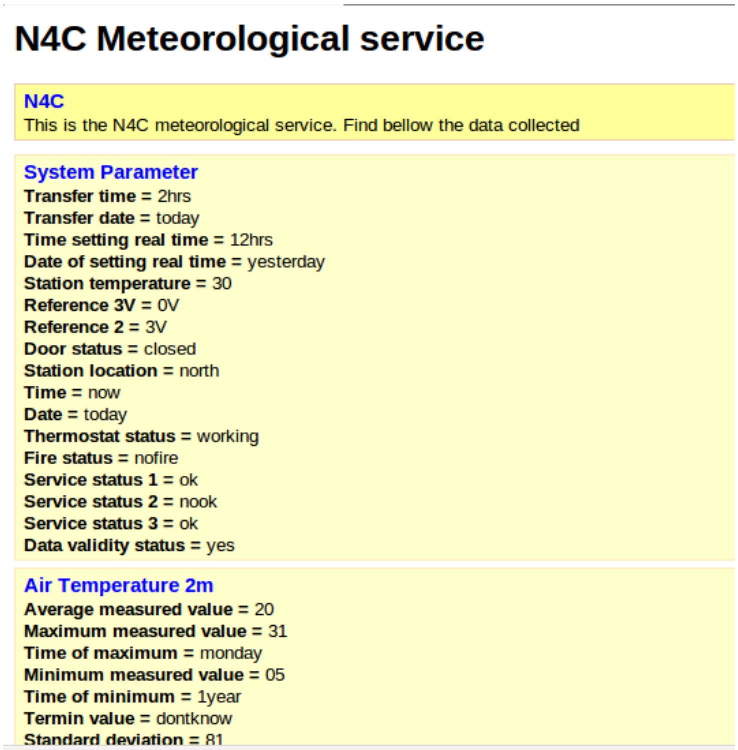

Please connect the DTN before starting webmail

Figure 11 - Webmail warning

If all the required processes are running this button will open the selected client email service. More information about webmail service will follow.

Button 3 is the Meteorological Service. As HTMLrequester this opens a tab with a web page that is available on the DTN Outstation. This web page will contain updated meteorological information collected in two meteorological stations like in Figure 12.



N4C Meteorological service

N4C
This is the N4C meteorological service. Find bellow the data collected

System Parameter
Transfer time = 2hrs
Transfer date = today
Time setting real time = 12hrs
Date of setting real time = yesterday
Station temperature = 30
Reference 3V = 0V
Reference 2 = 3V
Door status = closed
Station location = north
Time = now
Date = today
Thermostat status = working
Fire status = nofire
Service status 1 = ok
Service status 2 = nook
Service status 3 = ok
Data validity status = yes

Air Temperature 2m
Average measured value = 20
Maximum measured value = 31
Time of maximum = monday
Minimum measured value = 05
Time of minimum = 1year
Termin value = dontknow
Standard deviation = 81

Figure 12 - Meteorological tab

Button 4 is the Not So Instant Messenger service. As said before this service is embedded in the Prophet application and the Not So Instant Messenger tab can be seen in Figure 13.

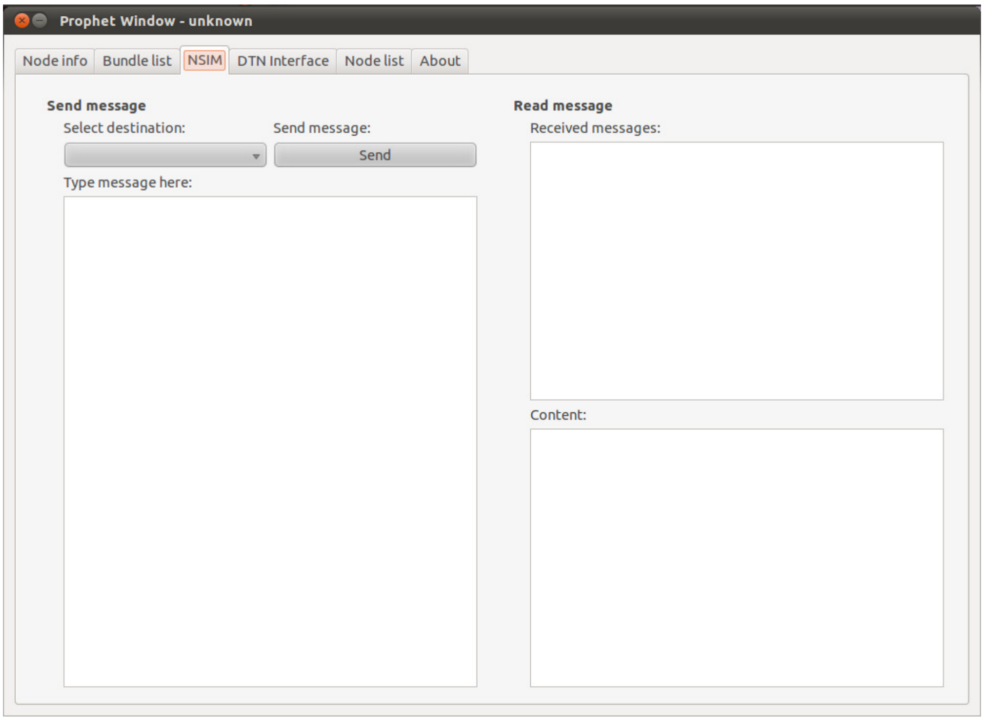
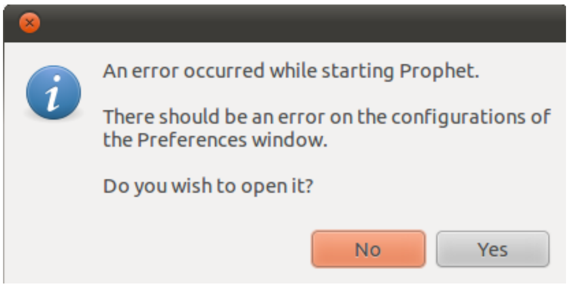


Figure 13 - Not So Instant Messenger

If there is a problem with the initialization of Prophet the DTN user will get a warning message like in Figure 14. This error usually indicates a problem with the configurations on the preferences window so this warning window provides a shortcut to the preferences window.



An error occurred while starting Prophet.

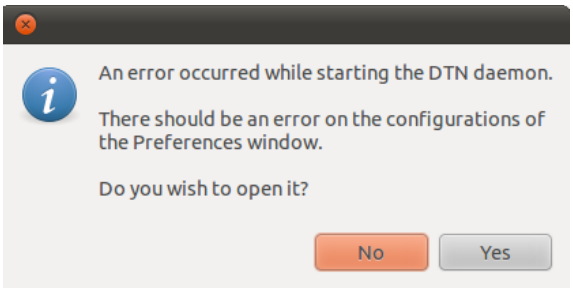
There should be an error on the configurations of the Preferences window.

Do you wish to open it?

Figure 14 - Prophet warning message

This application allows the DTN user to send messages to other DTN users. As others messages within the DTN it will wait for the other client to be available to deliver the message.

The next part of the N4C Integration Platform is about establishing a DTN connection and the configurations needed in order to do it. The button marked by 5 is the On/Off DTN button that initializes the local DTN and Pymail scripts necessary to use the email service. When this button is pressed, if the operation is successful the field marked by 7 will change it status to DTN connected, but if the operation is unsuccessful the DTN user will receive a warning message that can be seen in Figure 15.



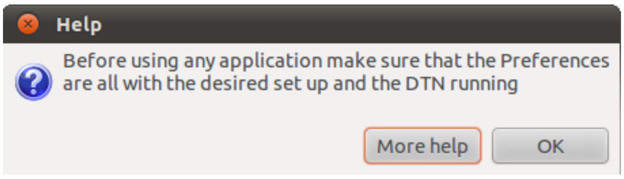
An error occurred while starting the DTN daemon.

There should be an error on the configurations of the Preferences window.

Do you wish to open it?

Figure 15 - DTN warning

If the DTN is not connected usually it indicates that is a problem with the preferences configurations so this warning asks the DTN user if he wants to open the preferences window. This warning window is similar to the one with Prophet and if the DTN user answers Yes the preferences window will open as well if the user presses the button with number 6. The preferences window is the same as in Figure 7. The next button present in this tab is the Help button with the number 8. This button pops up a window with some generic information that can be seen in Figure 16.

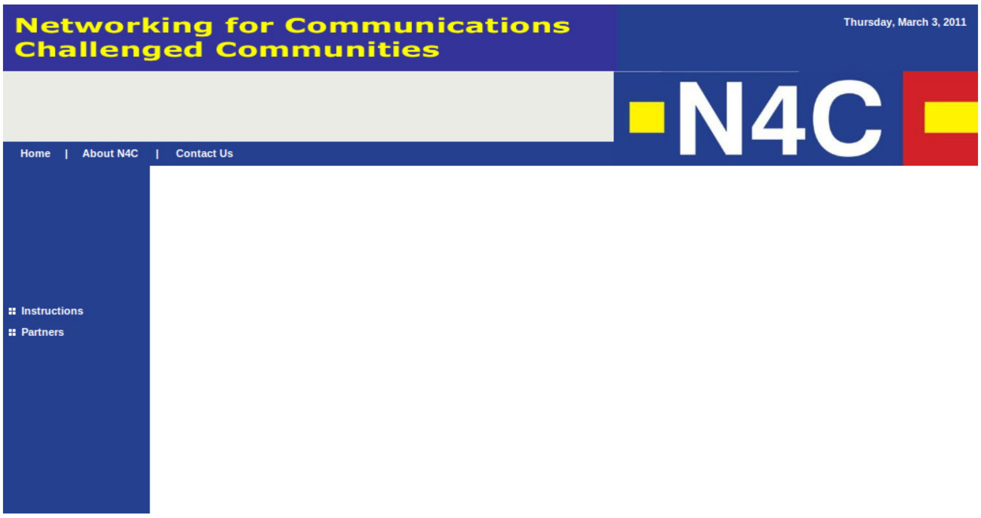


Help

Before using any application make sure that the Preferences are all with the desired set up and the DTN running

Figure 16 - Help window

This window gives a very brief overview from what the user should do in order to have access to the N4C services, but if he needs more help the More Help button opens a tab with a more complete help text. This opened page is part of a local website set up on the DTN Outstation that was designed to provide all the information the DTN User needs in order to use the N4C services. This page layout will be as in Figure 17.

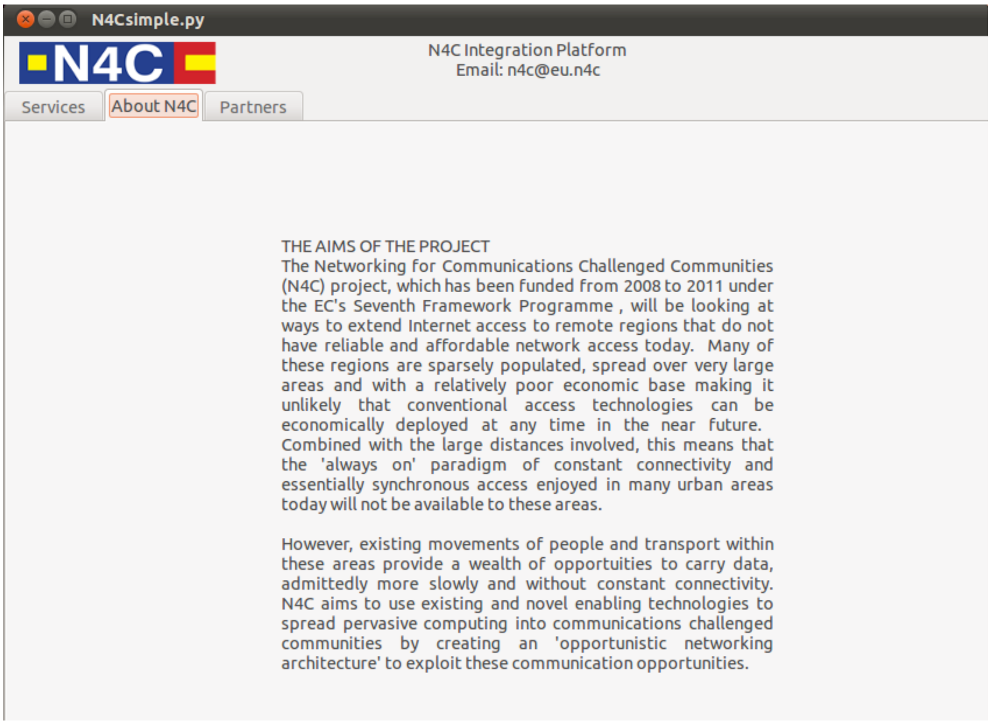


Networking for Communications Challenged Communities
Thursday, March 3, 2011
N4C
Home | About N4C | Contact Us
Instructions
Partners

Figure 17 - Local help website's format

This local website is still under development but was designed to give the local users all the information needed and access to all the downloads needed in order to set up and install all the components needed.

There are two more tabs on the N4C Integration Platform, the About N4C tab and the Partners tab. The about N4C tab is to provide the DTN users some information about the N4C project. This tab can be accessed on the main screen and it looks like Figure 18. The partners tab was designed to give the DTN users information about all the entities involved on the N4C project. This tab as one button for each partner like in Figure 19.



THE AIMS OF THE PROJECT
The Networking for Communications Challenged Communities (N4C) project, which has been funded from 2008 to 2011 under the EC's Seventh Framework Programme , will be looking at ways to extend Internet access to remote regions that do not have reliable and affordable network access today. Many of these regions are sparsely populated, spread over very large areas and with a relatively poor economic base making it unlikely that conventional access technologies can be economically deployed at any time in the near future. Combined with the large distances involved, this means that the 'always on' paradigm of constant connectivity and essentially synchronous access enjoyed in many urban areas today will not be available to these areas.

However, existing movements of people and transport within these areas provide a wealth of opportunities to carry data, admittedly more slowly and without constant connectivity. N4C aims to use existing and novel enabling technologies to spread pervasive computing into communications challenged communities by creating an 'opportunistic networking architecture' to exploit these communication opportunities.

Figure 18 - About tab

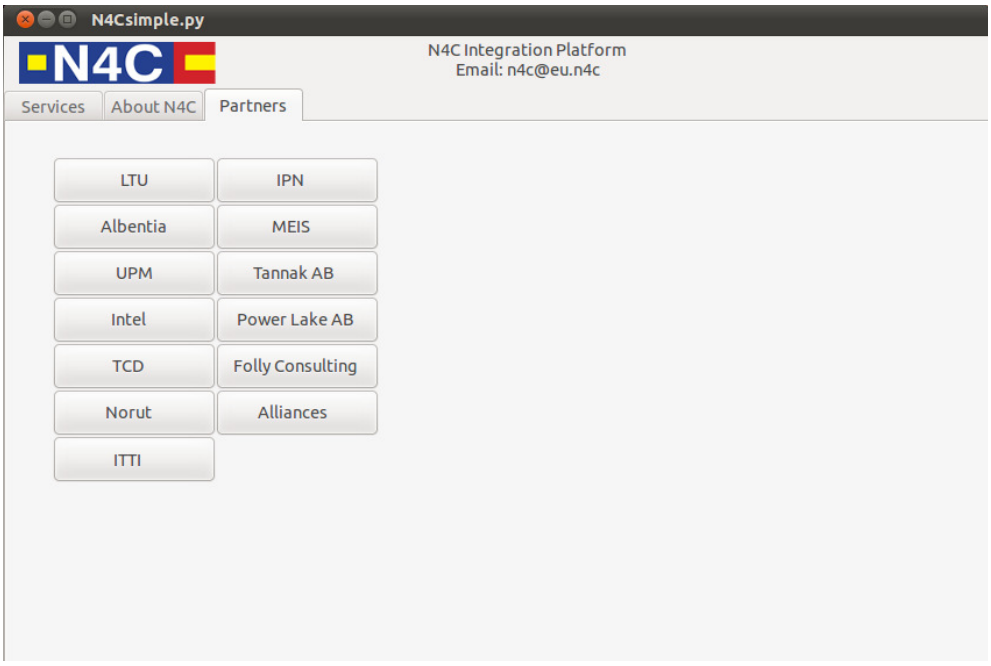
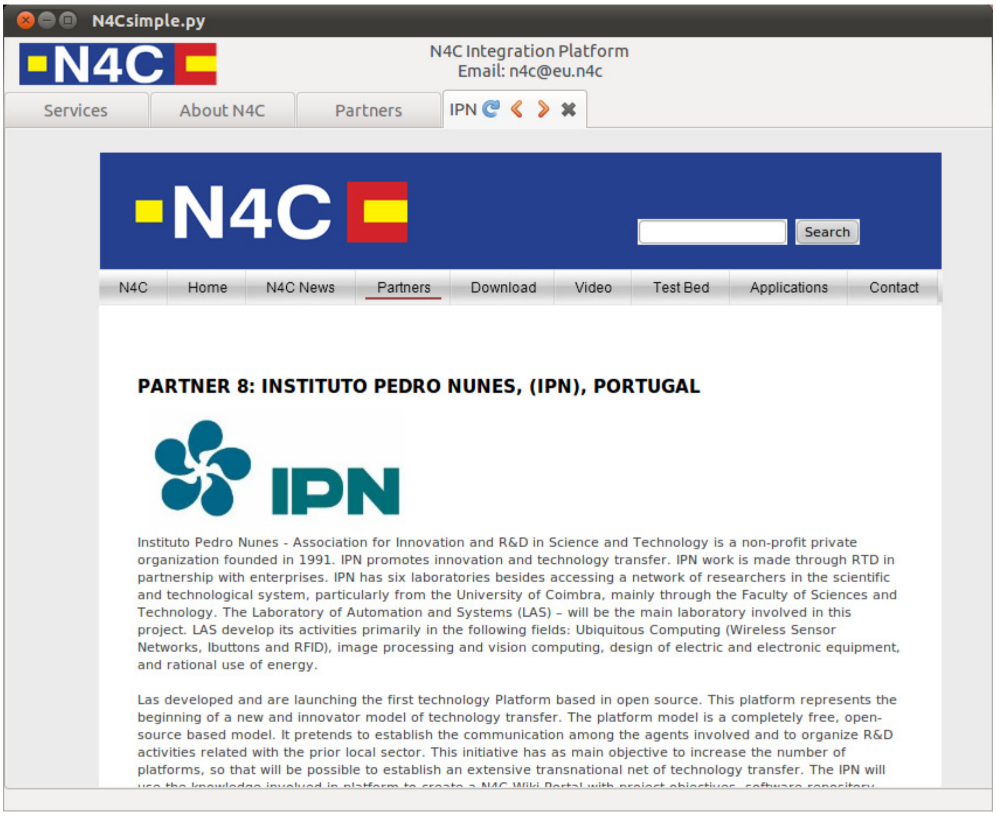


Figure 19 - Partner's tab

The N4C web page was downloaded and is available on the DTN Outstation in order to give the DTN user more complete information about the N4C project. This buttons will open the respective page of each partner like Figure 20.



PARTNER 8: INSTITUTO PEDRO NUNES, (IPN), PORTUGAL

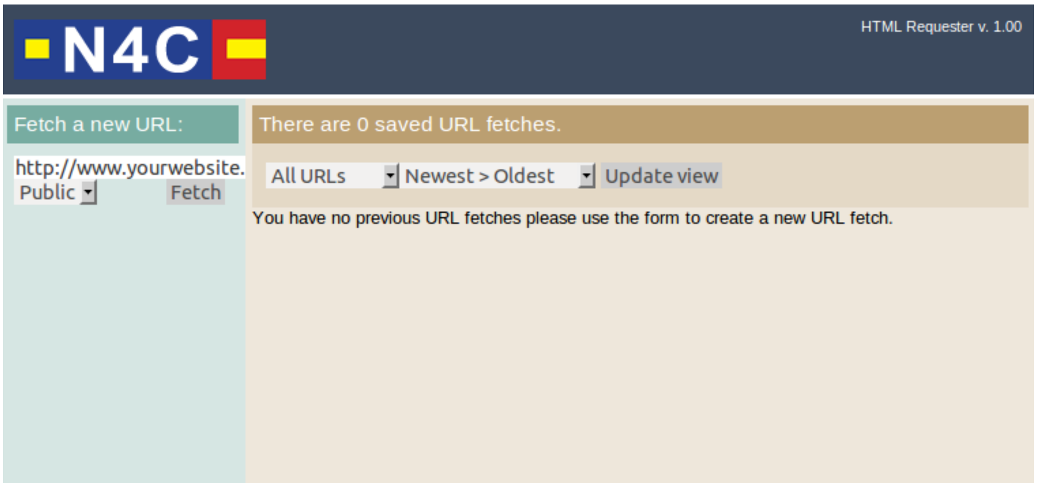
Instituto Pedro Nunes - Association for Innovation and R&D in Science and Technology is a non-profit private organization founded in 1991. IPN promotes innovation and technology transfer. IPN work is made through RTD in partnership with enterprises. IPN has six laboratories besides accessing a network of researchers in the scientific and technological system, particularly from the University of Coimbra, mainly through the Faculty of Sciences and Technology. The Laboratory of Automation and Systems (LAS) – will be the main laboratory involved in this project. LAS develop its activities primarily in the following fields: Ubiquitous Computing (Wireless Sensor Networks, Ibuttons and RFID), image processing and vision computing, design of electric and electronic equipment, and rational use of energy.

Las developed and are launching the first technology Platform based in open source. This platform represents the beginning of a new and innovator model of technology transfer. The platform model is a completely free, open-source based model. It pretends to establish the communication among the agents involved and to organize R&D activities related with the prior local sector. This initiative has as main objective to increase the number of platforms, so that will be possible to establish an extensive transnational net of technology transfer. The IPN will use the knowledge involved in platform to create a N4C Wiki Portal with project objectives, software repository

Figure 20 - Partner's example

HTMLREQUESTER

HTMLrequester is the module that allows the DTN user to have ability to request HTML pages and navigate them offline. In Figure 21 we can see the initial screen of HTMLrequester.



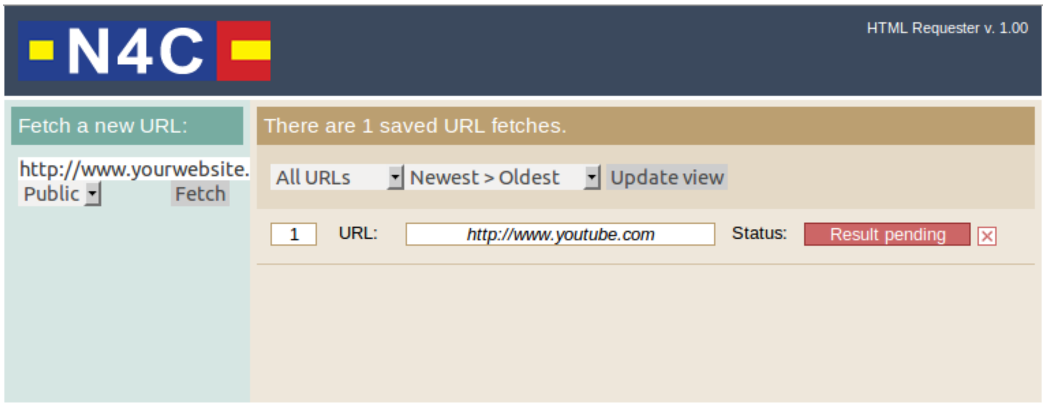
There are 0 saved URL fetches.

You have no previous URL fetches please use the form to create a new URL fetch.

Figure 21 - HTMLrequester initial screenshot

The HTMLrequester module is required to be installed on the DTN Gateway and on the DTN Outstation. Has the DTN Gateway does the interaction between the N4C network and the legacy internet it holds the HTMLrequester part that fetches and encapsulate the requested HTML pages. There is a process that is always running on the DTN Gateway (dtnN4Cmiddleware) that detects when the DTN receives a request for a HTML page and uses wget to download the requested page.

Wget will download three levels of the HTML page, this is, it follows, and downloads, all the links three times, and does all the settings needed in order for the user to navigate these three levels in off line mode. After the downloading is completed it encapsulates it into a tar.gz file and sends it back to the users DTN Outstation. Besides this it makes a copy of the downloaded content in case of an error occurring on the encapsulating or sending process.



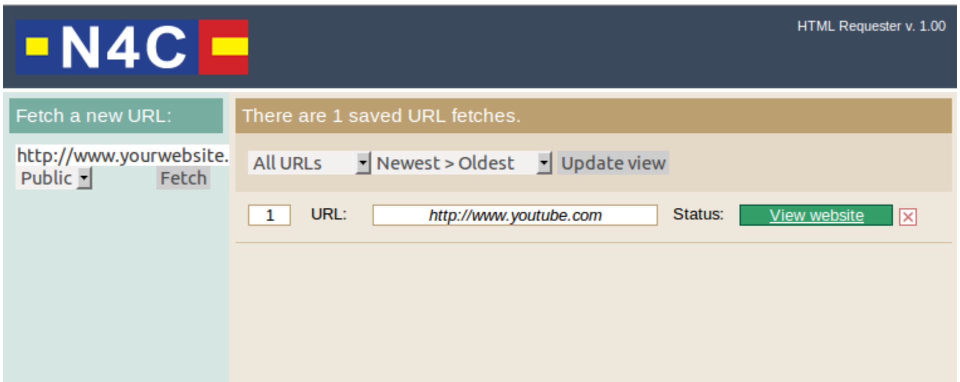
http://www.yourwebsite.

There are 1 saved URL fetches.

1 URL: http://www.youtube.com Status: Result pending

Figure 22 - Request waiting to be satisfied

In Figure 22 can be seen a request for www.youtube.com that has not yet been satisfied. In order to do this request the DTN user must type the address that it wants to see on the left side box and click on the Fetch button. This will result in a window similar to the one in Figure 22 that will wait for the response in order to update the status of the request. After the request is satisfied and the response bundle is received by the DTN Outstation the interface will change the status of the request and present it like in Figure 23. The View website link will be turned active and when the user clicks on it the requested HTML page will be presented.



http://www.yourwebsite.

There are 1 saved URL fetches.

1 URL: http://www.youtube.com Status: View website

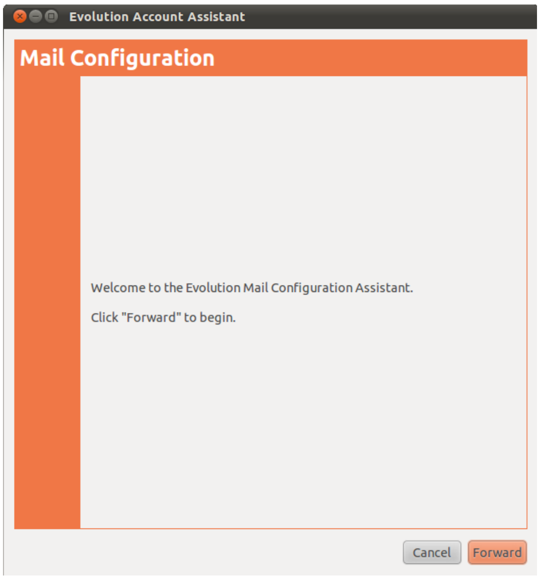
Figure 23 - Satisfied request

This request will be available to all DTN users if the request is made as public but if the request is made as private it will available just for DTN user that made it. This feature, controlled by the drop menu beneath the new URL box, is very useful in a remote community because if the link that a DTN user wants to see was already asked by another DTN user he will have a faster access to the information wanted without having to make a request. This will also be useful for the DTN itself because it will reduce the number of requests made and therefore the number of bundles that will be in transit inside the N4C network.

PYMAIL

Pymail is the module that gives the DTN users ability to have email service and its the reason for the DTN users to have a DTN installation. This module works on the DTN Gateway and on the DTN user. On the DTN user side its necessary to have an email client, we recommend Evolution, and running DTN and Pymail scripts. The Pymail scripts do the link between the email client and the DTN and when an email is sent these scripts make a bundle out of it having has destination a DTN Gateway. To get to the DTN Gateway this bundle will have to pass through the DTN Outstation and one or more DTN mules. When the bundle that contains the email arrives to the DTN Gateway it will wait within the DTN until the Pymail scripts for the DTN Gateway are running. These scripts on the DTN Gateway do the inverse of the ones on the DTN user because they identify when DTN receives a bundle that contains a email and forwards it to Postfix. Postfix will do the bridge between the local machine and the legacy internet.

The webmail button from the N4C Integration Platform opens the selected email client, and a client configuration will be asked. In the case of Evolution the DTN client will get a screen like in Figure 24 when this is initialized for the first time.

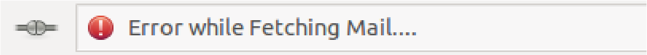


Welcome to the Evolution Mail Configuration Assistant.

Click "Forward" to begin.

Figure 24 - Evolution initialization

The DTN user has to configure the email client with some specific configurations but we will get to this latter on the document. If the configurations are not correct when the DTN tries to connect the email client he will receive an error message like the one in Figure 25.



Error while Fetching Mail....

Figure 25 - Evolution connection error

This example is from Evolution and it will appear on the left bottom corner of the screen. If this error appears it can mean one of two things: or the Pymail scripts are not running or the client configuration was not done correctly.

METEOROLOGICAL SERVICE

The Meteorological Service was designed to provide meteorological data from different meteorological stations, one in Veliki Locnik and the other in Nozice.

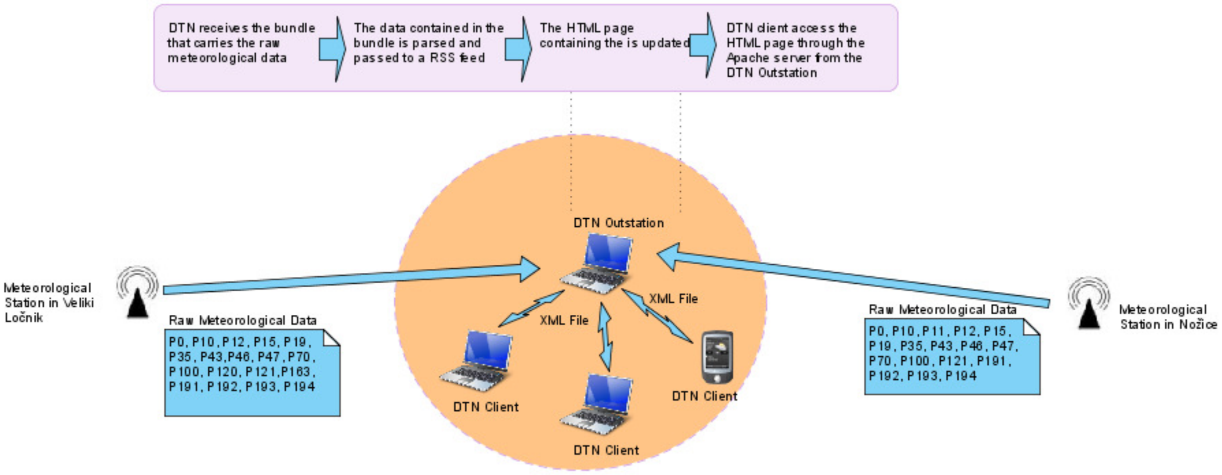
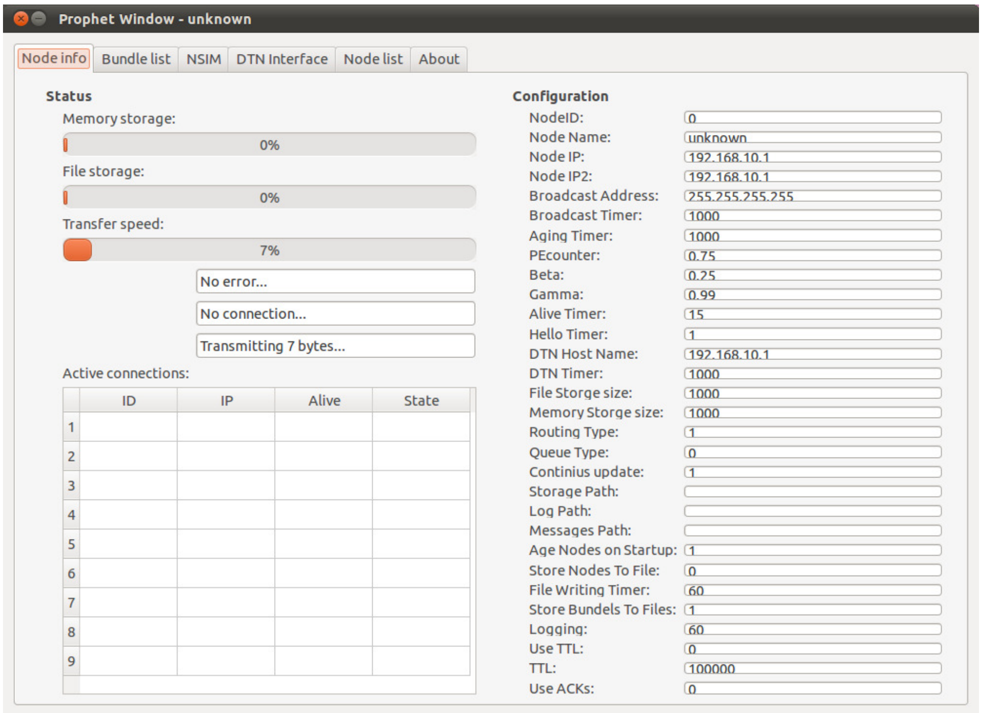

DTN receives the bundle that carries the raw meteorological data
The data contained in the bundle is parsed and passed to a RSS feed
The HTML page containing the is updated
DTN client access the HTML page through the Apache server from the DTN Outstation
DTN Outstation
Meteorological Station in Veliki Ločnik
Raw Meteorological Data
P0, P10, P12, P15, P19, P35, P43, P46, P47, P70, P100, P120, P121, P163, P191, P192, P193, P194
XML File
XML File
DTN Client
DTN Client
DTN Client
Raw Meteorological Data
P0, P10, P11, P12, P15, P19, P35, P43, P46, P47, P70, P100, P121, P191, P192, P193, P194
Meteorological Station in Nožice

Figure 26 - Meteorological service scheme

As can be seen in Figure 26 the bundles sent from the meteorological stations will send raw meteorological data that will be received and treated by the DTN Outstation. The meteorological information will be parsed and updated in a XML file that will be accessed by the web page present on the Apache server of the DTN Outstation that all the DTN Users can access. A example of this web page is present in Figure 12.

NOT SO INSTANT MESSENGER

Not So Instant Messenger works within Prophet. Prophet monitors all the activity inside the DTN and is able to solve the routing problem of the bundles. Prophet requires a proper configuration in order for it to work alongside the DTN and to be able to deliver the messages to their destinations. So, when the DTN user clicks on the Messenger button the Prophet will be initialized as in Figure 27.



Status

Memory storage: 0%

File storage: 0%

Transfer speed: 7%

No error...

No connection...

Transmitting 7 bytes...

Active connections:

	ID	IP	Alive	State
1				
2				
3				
4				
5				
6				
7				
8				
9				

Configuration

NodeID:	0
Node Name:	unknown
Node IP:	192.168.10.1
Node IP2:	192.168.10.1
Broadcast Address:	255.255.255.255
Broadcast Timer:	1000
Aging Timer:	1000
PEcounter:	0.75
Beta:	0.25
Gamma:	0.99
Alive Timer:	15
Hello Timer:	1
DTN Host Name:	192.168.10.1
DTN Timer:	1000
File Storge size:	1000
Memory Storge size:	1000
Routing Type:	1
Queue Type:	0
Continius update:	1
Storage Path:	
Log Path:	
Messages Path:	
Age Nodes on Startup:	1
Store Nodes To File:	0
File Writing Timer:	60
Store Bundels To Files:	1
Logging:	60
Use TTL:	0
TTL:	100000
Use ACKs:	0

Figure 27 - Prophet initial screen

It can be seen that Prophet has several tabs. This first one as information of the actual configurations on the right side and a monitor system of all the nodes active on the DTN network as well as the DTN running on the DTN user. The next tab is the Bundle List tab. This lists all the bundles on the DTN of the user that are going out or coming in. This provides several data about all the bundles like ID, size, source, destination or the options that the bundle has.

The Not So Instant Messenger tab already seen in Figure 13 is a simple messenger service where the DTN user can chose the destination of the messages, write and send them. It can also receive messages from other DTN users and see them.

In 0 was seen a DTN working in a console but Prophet has its own DTN interface, present in the next tab, in order for the DTN user do not need to work on the console. The Node list tab contains a list of all known nodes. This is useful for the routing protocol because it can establish the route dynamically. At this time the routing protocol is at a testing phase so more details will be latter released.

GUIDELINES FOR DEPLOYMENT

This section will provide a full step-by-step detailed installation process of all entities needed in order to successfully deploy a N4C network that, as has been explained, requires four entities: DTN Gateway, DTN Mule, DTN Outstation and DTN user. These installations will be provided in the form of installation packages or in the form of ISO files with a full installation. These installation methods will be described next.

INSTALLATION OF PACKAGES

One of the options of installation process is the installation by packages. There are two main distributions of Linux OS, Debian that is based on DEB packages and the RedHat that is based on RPM packages. The Debian based installation will be the only form explored in this document but the procedure for RedHat based OS should be very similar. All screenshots provided here are using Ubuntu 10.10 that is our recommended OS, but the procedure is applied to other OS also.

The selected package to install should be the suited to the users OS because if it is not the installation procedure will not work. The user should be careful in order to do not install different packages on the same machine because it can generate some problems and cause the N4C network to fail. Primarily there are four major packages mentioned below.

1. **nccc-gateway – Networking for Communications Challenged Communities gateway**
2. **nccc-mule – Networking for Communications Challenged Communities mule**
3. **nccc-outstation – Networking for Communications Challenged Communities outstation.**
4. **Nccc-node – Networking for Communication Challenged Communities node.**

DTN Gateway

The installation package for the DTN Gateway is **nccc-gateway**. This package will install all the dependencies needed to a successful installation of a DTN Gateway and all the applications that it needs as DTN, Pymail and HTMLrequester. As been said Pymail uses Postfix to handle the connection between the legacy internet and the DTN so Postfix is also installed upon installation of nccc-gateway. Also like Pymail, HTMLrequester depends on other applications that have to be installed. When nccc-gateway is installed it will also install mysql-server that will allow HTMLrequester to keep track of the requests.

Beyond doing the installation procedure nccc-gateway will set up a local repository of all the packages needed for the installation of nccc-gateway, as well as other packages for other entities of the N4C network.

DTN Mule

The DTN mule package is the simpler one because it only needs to install the DTN. This package is called **nccc-mule**. This package will be present on the repository of any DTN Gateway or DTN Outstation and upon the installation it will install DTN and all the applications related to it.

DTN Outstation

The installation package for the DTN Gateway is **nccc-outstation**. The installation procedure on the DTN Outstation is similar to the one on DTN Gateway but the applications installed are different because on the DTN Outstation will be only installed DTN and HTMLrequester.

DTN installation is similar to all other packages but the HTMLrequester installation is different because on the DTN Outstation HTMLrequester is designed to make possible for the user to request the HTML pages and then making it available for the user to navigate them. It also needs some extra applications like mysql-server that will be installed upon installation of the original package. Similar to DTN Gateway, DTN Outstation will set up a local repository for DTN Mules and DTN Users. It is assumed that the DTN Users do not have access to legacy internet so the local repository must contain every package needed in order for the installation process to be successful and the usage without problems.

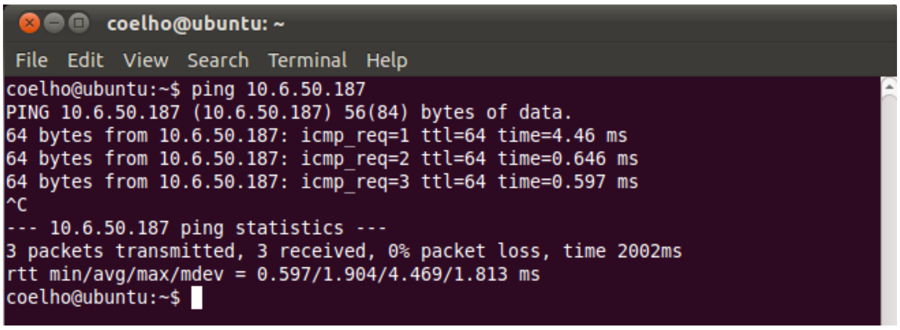
DTN User

Similar to the other already discussed packages DTN User package is **nccc-node**. This package will install DTN, Pymail, Prophet and N4C Integration Platform. Once more DTN installation is similar to all other entities but the Pymail installation is different because in this case it does not do the bridge between legacy internet and DTN but it links the chosen email client to the DTN.

Prophet application will work as a routing protocol and will allow the DTN User to send messages to other DTN Users inside the DTN network. N4C Integration Platform, already described in 3.2, is installed in DTN user allowing the user a simple way to manage all the services provided by the N4C network. Because the DTN User does not have connection to legacy internet the installation will be done through the DTN Outstations repository.

INSTALLATION PREPARATION

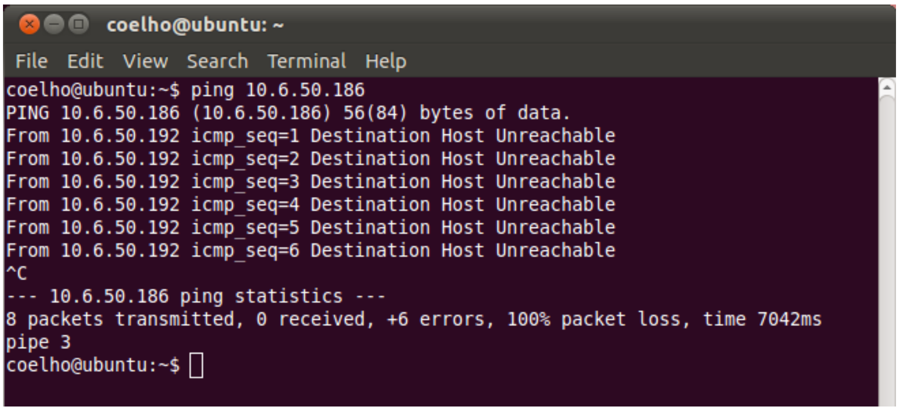
The preparation for the installation process is very simple because the user just should make sure that he is within reach of a local repository that could be in a DTN Gateway or in a DTN Outstation. To do this the user can open a console and use the ping command that will send a ping signal through the network and if the IP address is reachable it will receive a signal back. An example of this command output is in Figure 28.



```
coelho@ubuntu:~$ ping 10.6.50.187
PING 10.6.50.187 (10.6.50.187) 56(84) bytes of data.
64 bytes from 10.6.50.187: icmp_req=1 ttl=64 time=4.46 ms
64 bytes from 10.6.50.187: icmp_req=2 ttl=64 time=0.646 ms
64 bytes from 10.6.50.187: icmp_req=3 ttl=64 time=0.597 ms
^C
--- 10.6.50.187 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.597/1.904/4.469/1.813 ms
coelho@ubuntu:~$
```

Figure 28 - Successful ping output

On the example of Figure 28 the command ping **10.6.50.187** was used, where 10.6.50.187 is our repository IP, and three packets were sent and three received has can be seen. This means that the local repository is well within the reach. If the local repository is unreachable the output of the command will be like in Figure 29.



```
coelho@ubuntu:~$ ping 10.6.50.186
PING 10.6.50.186 (10.6.50.186) 56(84) bytes of data.
From 10.6.50.192 icmp_seq=1 Destination Host Unreachable
From 10.6.50.192 icmp_seq=2 Destination Host Unreachable
From 10.6.50.192 icmp_seq=3 Destination Host Unreachable
From 10.6.50.192 icmp_seq=4 Destination Host Unreachable
From 10.6.50.192 icmp_seq=5 Destination Host Unreachable
From 10.6.50.192 icmp_seq=6 Destination Host Unreachable
^C
--- 10.6.50.186 ping statistics ---
8 packets transmitted, 0 received, +6 errors, 100% packet loss, time 7042ms
pipe 3
coelho@ubuntu:~$
```

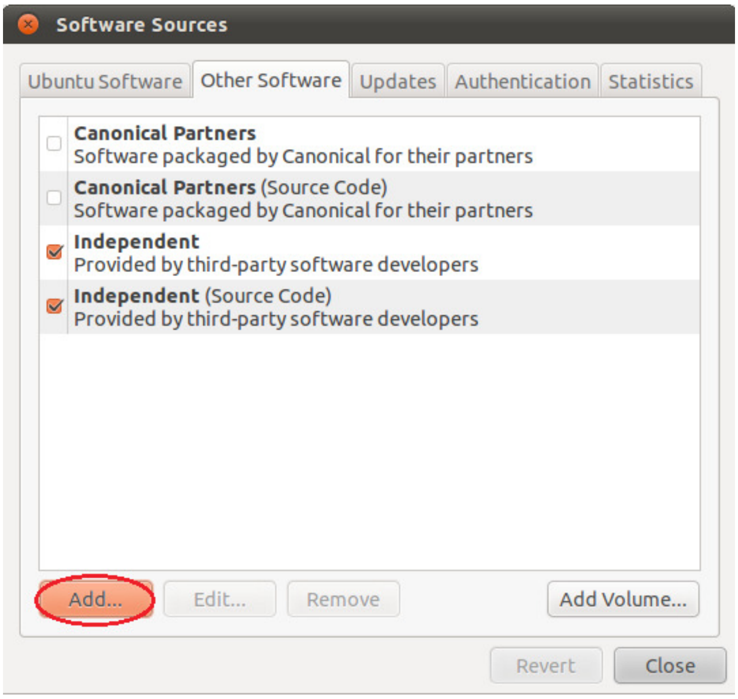
Figure 29 - Unsuccessful ping output

If the ping is unsuccessful like in Figure 29, where there was 100% of packet loss, it means that either the IP of the repositories machine is wrong or is out of reach. If this happens the installation will not be possible.

GUI INSTALLATION

The easiest way to install packages in Linux based systems is to use Package Manager. In the case of Ubuntu 10.10, the OS recommended, the Synaptic Package Manager is located on the System drop down menu at the top left corner of the screen inside the Administration sub menu.

The local repository is not known for the OS so it has to be manually added in order to be able to download the necessary packages. This is most crucial to DTN Users because they do not have access to legacy internet and therefore to the basic repositories. To do the manually addition of the local repository after starting the Synaptic Package Manager and inserting the password there will appear the initial screen that has on the top left corner the **Settings** menu. Inside this menu will be the **Repositories** option. This will open a menu like the one in Figure 30.



□ **Canonical Partners**
Software packaged by Canonical for their partners

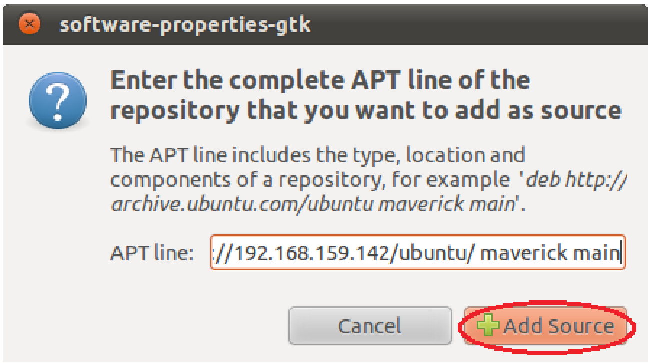
□ **Canonical Partners** (Source Code)
Software packaged by Canonical for their partners

☑ **Independent**
Provided by third-party software developers

☑ **Independent** (Source Code)
Provided by third-party software developers

Figure 30 - Repositories menu

In order to add the local repository click on the highlighted button of the Figure 30. This will open an extra window were the address of the local repository can be added. This window is present in Figure 31.



Enter the complete APT line of the repository that you want to add as source

The APT line includes the type, location and components of a repository, for example '*deb http://archive.ubuntu.com/ubuntu maverick main*'.

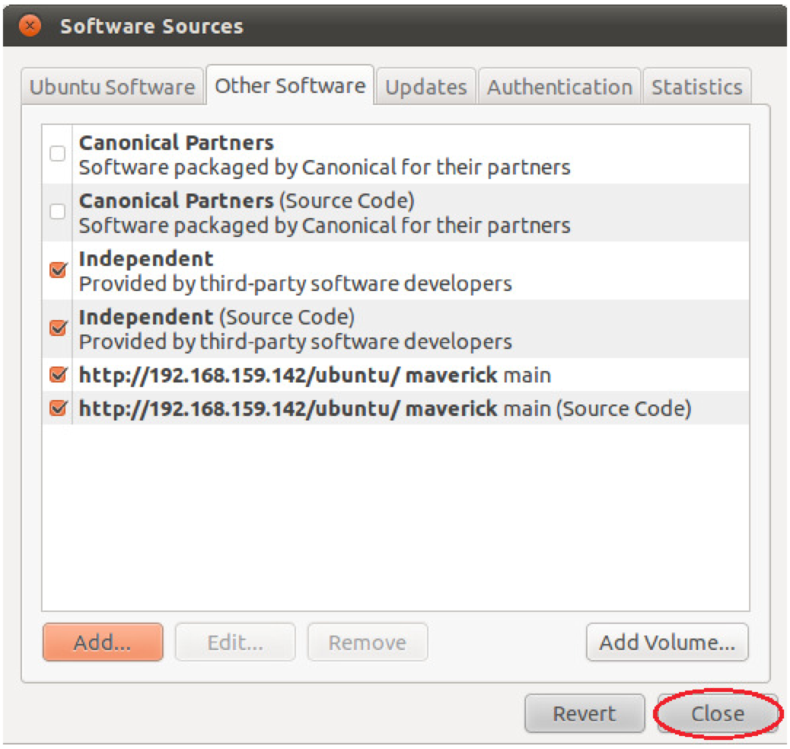
APT line: ://192.168.159.142/ubuntu/ maverick main

Figure 31 - Window to add repositories

There is an example from the APT line that should be inputted to add the local repositories. In this case the line should be as follows:

```
deb http://<IP address>/ubuntu/ maverick main
```

The IP address is the IP from the local repository, the one that was pinged in the installation preparation. After adding the APT line click on the highlighted button to add the source has a repository and the result should be like in Figure 32 where the new repositories can be seen.



□ **Canonical Partners**
Software packaged by Canonical for their partners

□ **Canonical Partners** (Source Code)
Software packaged by Canonical for their partners

☑ **Independent**
Provided by third-party software developers

☑ **Independent** (Source Code)
Provided by third-party software developers

☑ **http://192.168.159.142/ubuntu/ maverick** main

☑ **http://192.168.159.142/ubuntu/ maverick** main (Source Code)

Figure 32 - Updated list of repositories

After the highlighted Close button is pressed the initial screen will appear and it is necessary to click the Reload button like in **Erro! A origem da referência não foi encontrada.**.

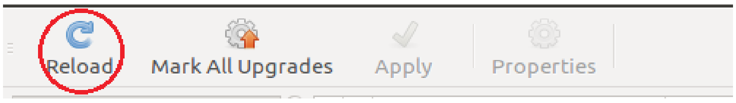
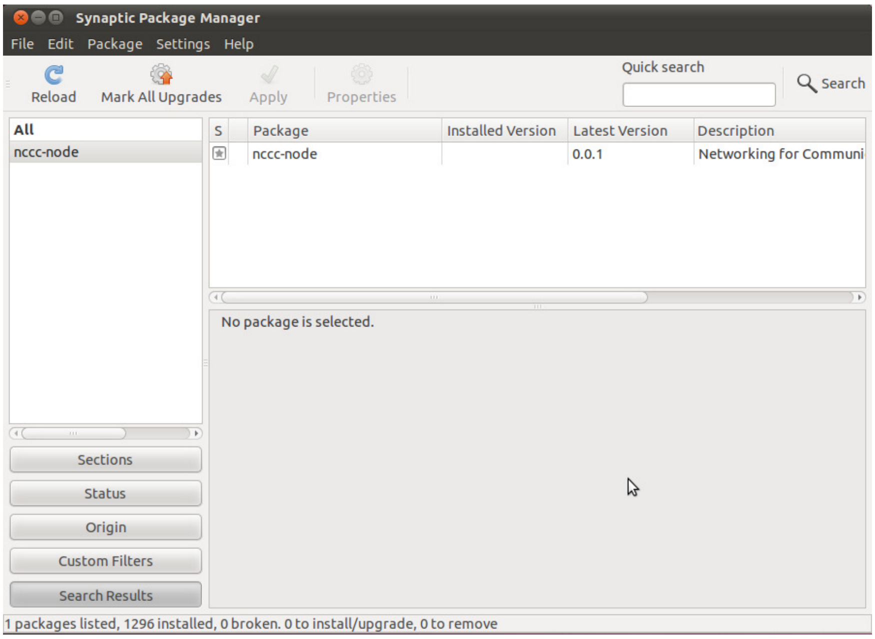


Figure 33 - Reload button

This Reload button will end the adding of the repositories. If this button is not pressed before continuing the new repositories will not be accessible and the required packages will not be available.

At this time the repository should be accessible and installation process should be ready to begin and a search should be done for the desired package. In **Erro! A origem da referência não foi encontrada.** is shown the result of a search result for the nccc-node package. If the result present in **Erro! A origem da referência não foi encontrada.** is empty it indicates that there is a problem with the repository that could have been badly added or out of reach. If this happens make sure that all the procedures until this point are correct.



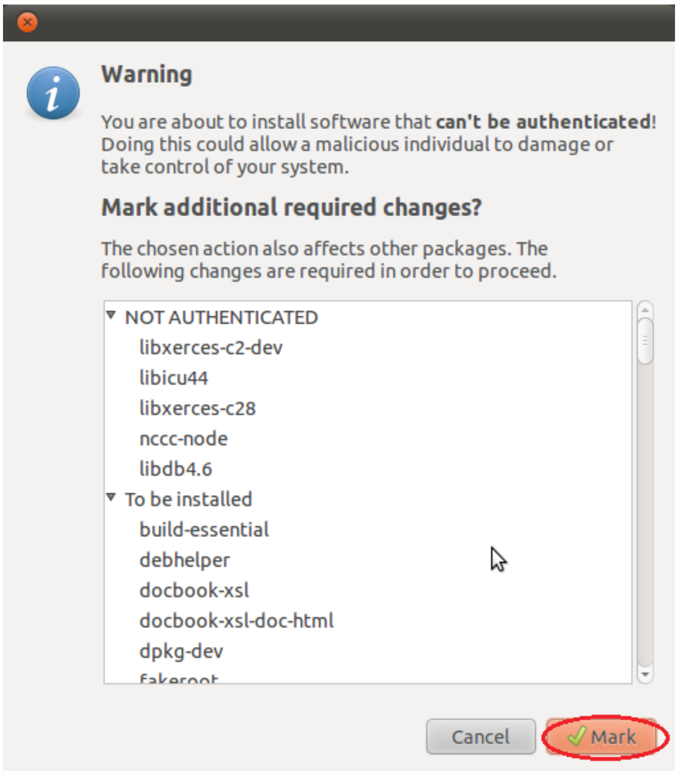
S	Package	Installed Version	Latest Version	Description
	nccc-node		0.0.1	Networking for Communi

No package is selected.

1 packages listed, 1296 installed, 0 broken. 0 to install/upgrade, 0 to remove

Figure 34 - nccc-node search result

The next step is to check the check-box of the selected package and marking it for installation and after this is done a pop up window will appear with all the dependencies that this package will install. The example of this pop window for the nccc-node package can be seen in Figure 35.


Warning

You are about to install software that **can't be authenticated!** Doing this could allow a malicious individual to damage or take control of your system.

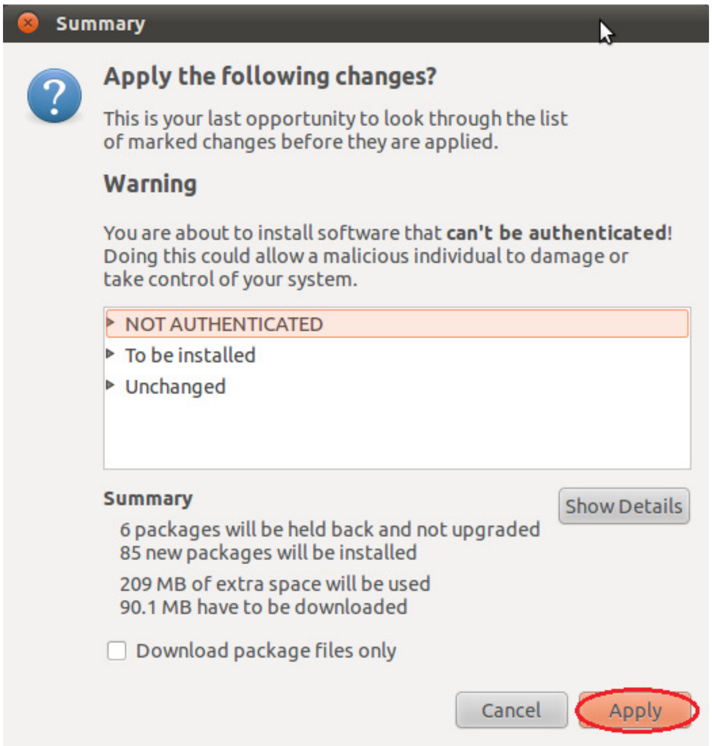
Mark additional required changes?

The chosen action also affects other packages. The following changes are required in order to proceed.

- NOT AUTHENTICATED
  - libxerces-c2-dev
  - libicu44
  - libxerces-c28
  - nccc-node
  - libdb4.6
- To be installed
  - build-essential
  - debhelper
  - docbook-xsl
  - docbook-xsl-doc-html
  - dpkg-dev
  - fakeroot

Figure 35 - Dependencies warning

Figure 35 presents the warning message that the Synaptic Package Manager returns when trying to install a new package. This is nothing but a safety feature in order for the user to be completely sure about what his about to install. The Mark button should be pressed in order for the window to disappear and the package to be marked for installation, followed by pressing the Apply button to make all the necessary downloads and to finish the installation process. When this button is pressed another warning will appear like in **Erro! A origem da referência não foi encontrada.**. Once more this is just a safety feature so just press the apply button and the installation process will begin.



Summary

Apply the following changes?

This is your last opportunity to look through the list of marked changes before they are applied.

Warning

You are about to install software that **can't be authenticated**! Doing this could allow a malicious individual to damage or take control of your system.

- NOT AUTHENTICATED
- To be installed
- Unchanged

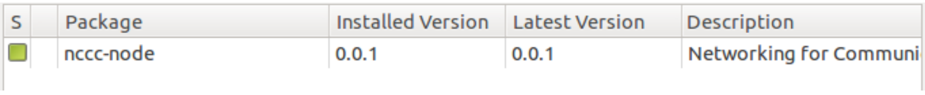
Summary

6 packages will be held back and not upgraded
85 new packages will be installed
209 MB of extra space will be used
90.1 MB have to be downloaded

□ Download package files only

Figure 36 - Installation warning

After the process of installation is done the state of the package should be like in Figure 37 but if the installation is not successful an error will appear.



S	Package	Installed Version	Latest Version	Description
🟩	nccc-node	0.0.1	0.0.1	Networking for Communi

Figure 37 - Package successful instalation

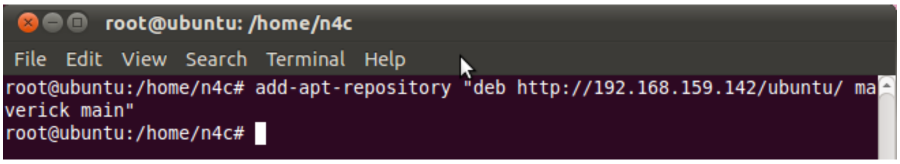
CONSOLE INSTALLATION

The console installation is intended for more experienced users but all steps are very simple and easy to follow. The example here presented is from the installation of nccc-mule but the procedure is similar to all the packages of the N4C network and the preparation for installation is the same as in 0 because it always needs to be in range of a local repository.

Because the steps follow the same order has with the GUI installation, the first step is to add the local repository to the repositories list with the command

```
sudo add-apt-repository “deb http://<IP address>/ubuntu/ maverick main”
```

where <IP address> is the IP address from the local repository. This command output is displayed in Figure 38.

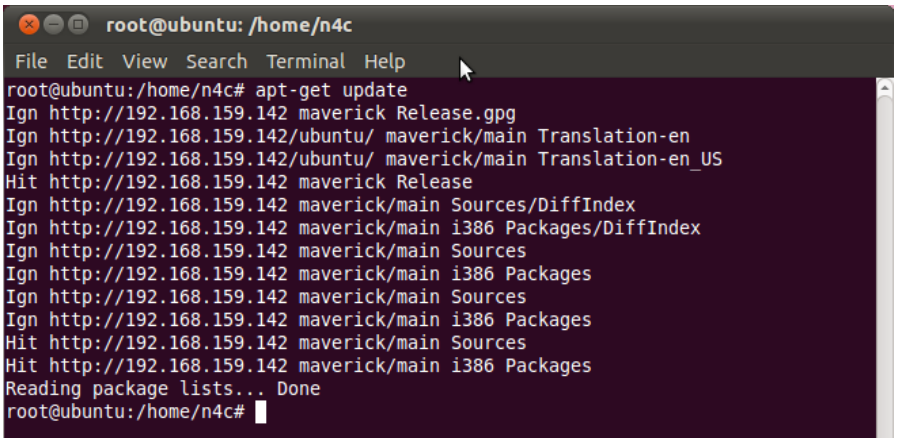


```
root@ubuntu:/home/n4c# add-apt-repository "deb http://192.168.159.142/ubuntu/ ma
verick main"
root@ubuntu:/home/n4c# 
```

Figure 38 - Adding repositorie by console

Next step is to update the repository with the content of the local repository. To do this execute the following command like in Figure 39.

```
sudo apt-get update
```



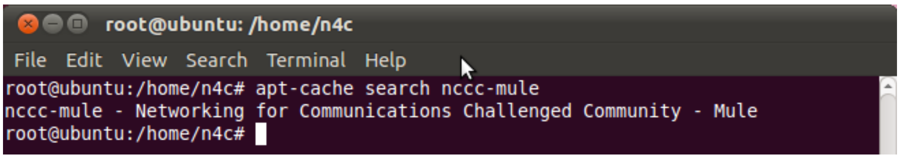
```
root@ubuntu:/home/n4c# apt-get update
Ign http://192.168.159.142 maverick Release.gpg
Ign http://192.168.159.142/ubuntu/ maverick/main Translation-en
Ign http://192.168.159.142/ubuntu/ maverick/main Translation-en_US
Hit http://192.168.159.142 maverick Release
Ign http://192.168.159.142 maverick/main Sources/DiffIndex
Ign http://192.168.159.142 maverick/main i386 Packages/DiffIndex
Ign http://192.168.159.142 maverick/main Sources
Ign http://192.168.159.142 maverick/main i386 Packages
Ign http://192.168.159.142 maverick/main Sources
Ign http://192.168.159.142 maverick/main i386 Packages
Hit http://192.168.159.142 maverick/main Sources
Hit http://192.168.159.142 maverick/main i386 Packages
Reading package lists... Done
root@ubuntu:/home/n4c# 
```

Figure 39 - Repositories update

The contents available to download should be displayed as result of the command. After this the desired package should be checked. To do this run the command

```
sudo apt-cache search <package>
```

where <package> is the desired package. The result of this command with nccc-mule can be seen in Figure 40.



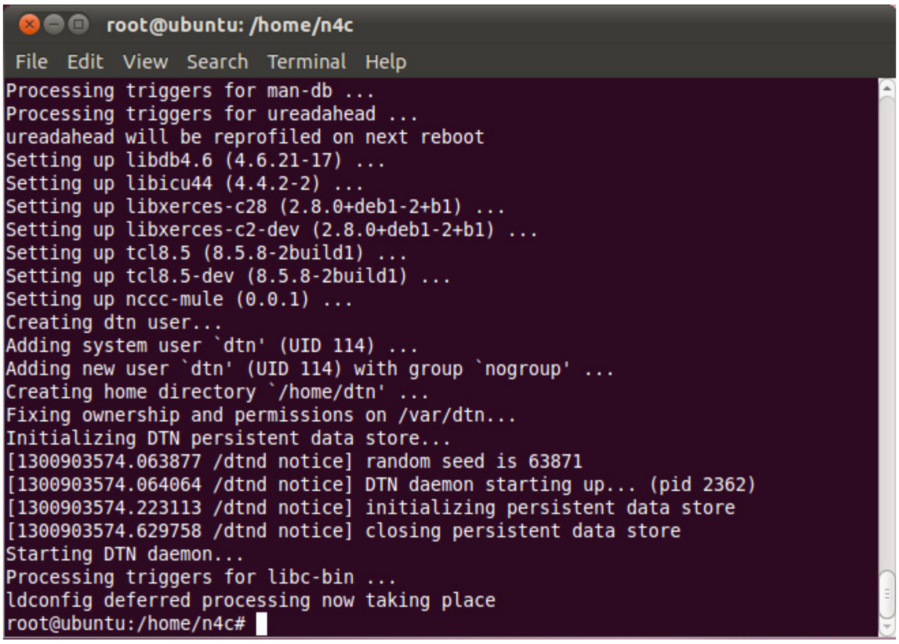
```
root@ubuntu:/home/n4c# apt-cache search nccc-mule
nccc-mule - Networking for Communications Challenged Community - Mule
root@ubuntu:/home/n4c# 
```

Figure 40 - nccc-mule search

The final step is to install the desired package by doing

```
sudo apt-get install <package>
```

where <package> is desired package, an answer yes to all the questions asked in the middle of the installation. The final screen should be like in Figure 41.



```
Processing triggers for man-db ...
Processing triggers for ureadahead ...
ureadahead will be reprofiled on next reboot
Setting up libdb4.6 (4.6.21-17) ...
Setting up libicu44 (4.4.2-2) ...
Setting up libxerces-c28 (2.8.0+deb1-2+b1) ...
Setting up libxerces-c2-dev (2.8.0+deb1-2+b1) ...
Setting up tcl8.5 (8.5.8-2build1) ...
Setting up tcl8.5-dev (8.5.8-2build1) ...
Setting up nccc-mule (0.0.1) ...
Creating dtn user...
Adding system user `dtn' (UID 114) ...
Adding new user `dtn' (UID 114) with group `nogroup' ...
Creating home directory `/home/dtn' ...
Fixing ownership and permissions on /var/dtn...
Initializing DTN persistent data store...
[1300903574.063877 /dtnd notice] random seed is 63871
[1300903574.064064 /dtnd notice] DTN daemon starting up... (pid 2362)
[1300903574.223113 /dtnd notice] initializing persistent data store
[1300903574.629758 /dtnd notice] closing persistent data store
Starting DTN daemon...
Processing triggers for libc-bin ...
ldconfig deferred processing now taking place
root@ubuntu:/home/n4c#
```

Figure 41 - Installation result

ISO INSTALLATION

The ISO installation process is available for DTN Gateway and for DTN Outstation. This is the simplest way to install these entities but it installs a completely new OS, Ubuntu 10.10, with all the components needed already installed. To do this just download the desired ISO file from one already set repository and burn it into a DVD or pen drive and then just install it as a new OS.

EXAMPLES OF COMNFIGURATION AND TESTS

After the installation is ready the objective is to provide a configuration guide and some test procedures in order to finish the installation proceedings. The configurations here given will be at the DTN User side, using the N4C Integration Platform.

CONFIGURATIONS

After the installation is done there will be a shortcut on the Applications menu, inside the internet section to open the N4C Integration Platform. As has already been said in 0, when the N4C

Integration Platform is executed for the first time it opens the Preferences window, present in Figure 7, for the DTN User to do the configurations.

Default DTN configuration file is **dtn.conf** and is present on the **/etc/** folder. This should be the initial option of the DTN User for a configuration file that can be later edited. The location of the Log files is not vital for this process but there is a folder special to save these files. This folder is by default **/var/log/** and the log file of the DTN will be created, if there is not one already created, and all the reports of the DTN will be inside this file. This is a very useful to solve DTN problems so it is advised for it to be kept in a correct and safe location.

HTMLrequester IP is the IP address of the DTN Outstation that serves the DTN User. This IP address will give ability for the N4C Integration Platform to communicate with the DTN Outstation and most importantly retrieve HTMLrequester and Meteorological Service pages. The Pymail folder asked next is the location of the Pymail scripts that will allow the N4C Integration Platform to control the execution of Pymail. The default location of this folder is **/usr/share/pymail/outstation/dtn/**.

As has been said the Not So Instant Messenger is part of Prophet module as can be seen in Figure 13 so the required location is the Prophet executable file folder. The default location of this file is **/usr/share/IntegrationPlatform/Prophet/x86/x86/**. The last parameter needed is the email client that the DTN User wants to use. It is recommended to the DTN User to use Evolution because it is usually installed by default on the Linux distributions and it is simple to use.

After these configurations are done the N4C Integration Platform should be ready to run. To stress out that all these locations are default with the nccc-node package.

EMAIL CLIENT CONFIGURATION

The Email client account needs some specific configuration in order to be able to work with Pymail.

When Evolution is started for the first time it helps you creating and configuring a new e-mail client. The most important configurations are the following.

Identity:

```
Name: <client>
Email Address: <client>@<system mail name>
```

With <client> being the client name and <system mail name> is the system mail name that is set in the DTN Gateways Postfix.

Receiving Email:

```
Server Type: Pop
Server: 127.0.0.1:2110
Username: dtnin (make sure you use this username or it wont work!)
Secutiry: No Encryption
```

Sending Email:

```
Server Type: SMTP
Server: 127.0.0.1:2125
Server Requires Authentication: No (uncheck box)
Use secure connection: No encryption
```

EDITING CONFIGURATION FILES

There are three files on the DTN User that may need some configuration and editing. In order for the DTN user to be able to do this editing in a simple way a text editor was integrated into the platform. This is available in the Configuration button highlighted in Figure 42.

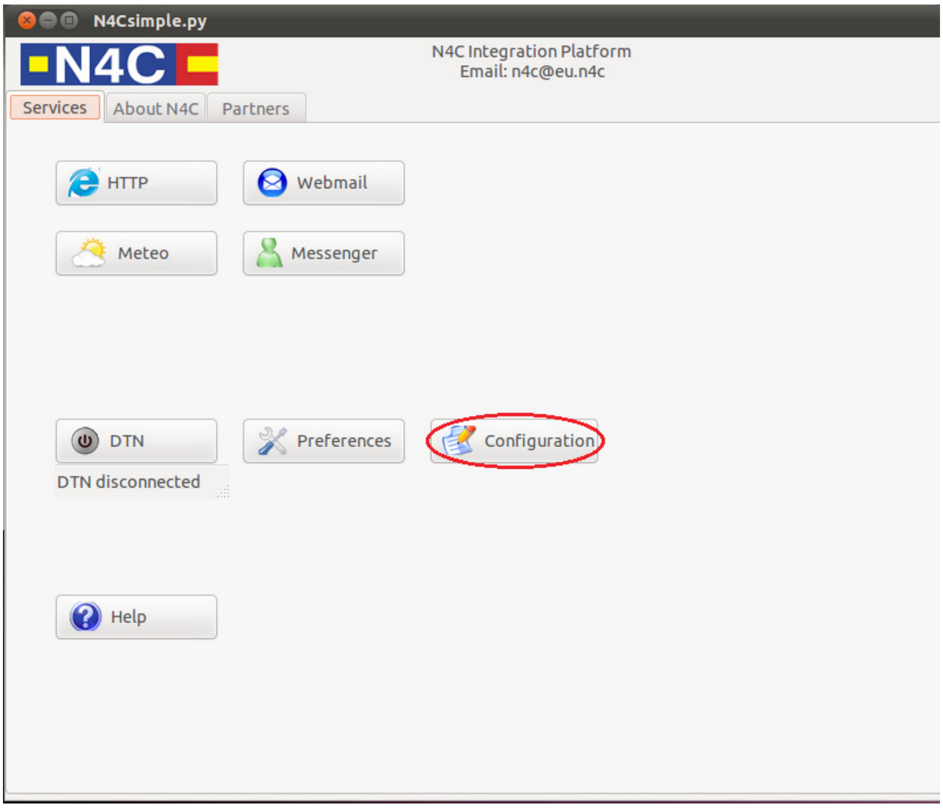


Figure 42 - Integration Platform configuration button

This Configuration button will open a new menu window with a selection of the file that the DTN user wants to edit. This new window is present in Figure 43.

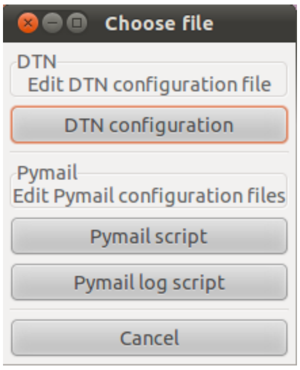
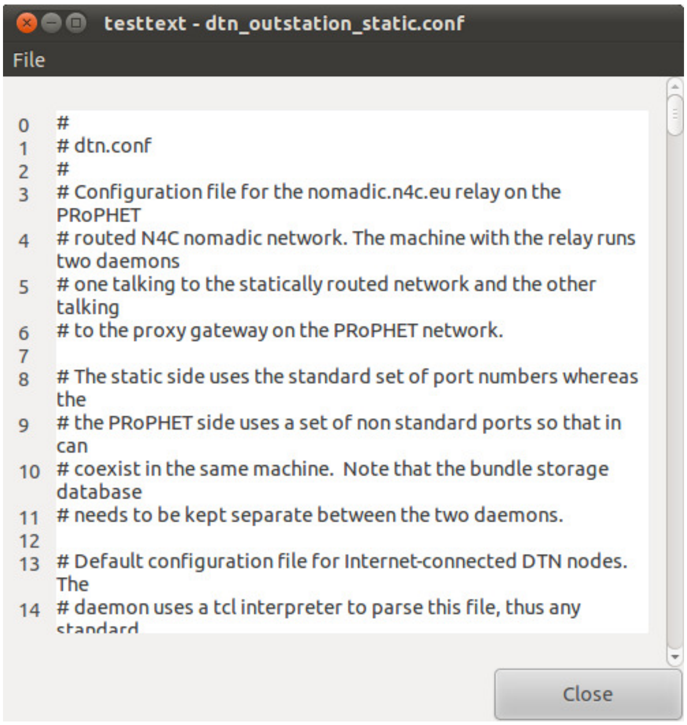


Figure 43 - Choose file window

In this window DTN User can choose between editing the DTN configuration file, Pymail script or Pymail log script that are the three files that may need to be edited in DTN User. After one of this files has been selected the text editor will open with the selected file ready to be edited has can be seen in Figure 44.



```
#
# dtn.conf
#
# Configuration file for the nomadic.n4c.eu relay on the PRoPHET
# routed N4C nomadic network. The machine with the relay runs two daemons
# one talking to the statically routed network and the other talking
# to the proxy gateway on the PRoPHET network.

# The static side uses the standard set of port numbers whereas the
# the PRoPHET side uses a set of non standard ports so that in can
# coexist in the same machine.  Note that the bundle storage database
# needs to be kept separate between the two daemons.

# Default configuration file for Internet-connected DTN nodes. The
# daemon uses a tcl interpreter to parse this file, thus any standard
```

Figure 44 - dtn.conf editing

There is some editing that can be done in these files. In the DTN configurations file the main things to edit are the routing options. The **route local_eid** command is the one that sets the identification of the DTN User inside of the DTN network. Another of the configuration needed should be the **route add** command that gives a static routing to all the bundles inside the DTN. The route add command has the following format

```
route add <destination local_eid>/* <next hop local_eid>
```

being <destination local_eid> the local_eid of the destination of the bundle and the <next hop local_eid> is the local_eid from the machine that the bundle has to go through in order to get to the destination.

In the Pymail script the **mail_domain** parameter can be edited because it should be the system mail name that is set in the DTN Gateways Postfix. Also the main function parameters can be edited on the bottom of the file with

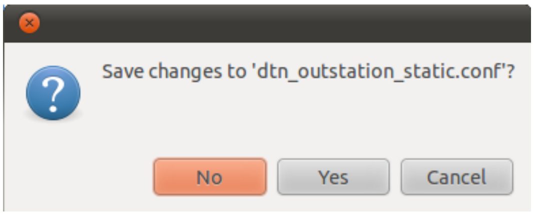
```
    main("/usr/share/pymail/maildir", "/usr/share/pymail/log",
"/usr/share/pymail/dtn/python/dtn_pymail_log.conf")
```

This line is most crucial for Pymail.

Inside the Pymail log script file the args arguments can be edited with

```
args=("/usr/share/pymail/log/dtn_pymail.log", "midnight", 1, 10)

    and

args=("/usr/share/pymail/log/dtn_postfix_pipe.log", "midnight", 1, 10)
```

If the Close button is clicked before the changes being saved the DTN User will get a warning message like in Figure 45.



Save changes to 'dtn_outstation_static.conf'?

Figure 45 - Warning message to save
When this message is displayed the DTN User has the option to save the changes made to the configuration file by pressing the Yes button, do not save the changes by selecting the No button or to go back to the edition by pressing the Cancel button. If the configurations in the preferences window, Figure 7, the configurations are not correct, when DTN User tries to edit a configuration file a warning message will be displayed like the one in Figure 46 - Error opening configurations files

TESTING

Throughout this document has already been discuss and prese.

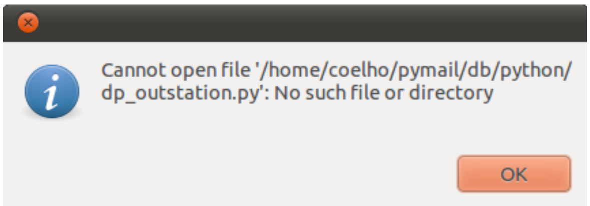

Cannot open file '/home/coelho/pymail/db/python/dp_outstation.py': No such file or directory

Figure 46 - Error opening configurations files

TESTING

Throughout this document has already been discuss and presented some error messages that give indication that something is wrong with configuration or usage, but there are two simple tests that can be easily made in order to know if all the installation and configurations are correct. The first test is to try to connect and disconnect the DTN with the button number five in Figure 9. It has already been discussed the DTN error present in Figure 15 but the status of the DTN can change to connect with the Pymail configuration wrong.

To test if the Pymail configuration is wrong after the DTN status change to DTN connected, press the DTN button again and if the Pymail configuration is wrong the N4C Integration Platform will return a fatal error and it will immediately shut down. Another simple test is to open the partners tab, Figure 19, and trying to open one of the partners page. If a blank page is displayed it means that the HTMLrequester IP from the Preferences window is wrong or the DTN Outstation is not available.

CONCLUSIONS

At this point already has been achieved a high level of interoperability, getting all the modules working properly when they are gathered and deployed. There are some modules that require further development and some refining because these modules are at their early stages but they already achieved a very good level that allows this software to be deployed at real sites.

REFERENCES