Bytewalla IV
Implementation of Delay Tolerant Networks on the Android platform

Progress Report

Author:
- Michel Hognerud

Examiner:
- Peter Sjödin

Supervisors:
- Bjorn Pehrson
- Hervé Ntareme
- Danilo Gligoroski

Coaches :
- Doria Avri
- Marco Zennaro

Revision History

Version	Date	Remarks
V1.0	2011-05-06	Document creation.
V1.1	2011-05-09	Thesis objectives introduction, corrections…

Table of Contents

Introduction

This document describes the thesis as of May 5, 2011. It describes the activities I performed recently and the future objectives.

Thesis Objective

In the thesis plan we have agreed for the objectives explained below. The first part “Bundle priority and queuing mechanism” was implemented earlier this semester, and the second part “Service Layer” is still under development.

Bundle priority and queuing mechanism

Due to storage limitation in mobiles, nodes may need to drop some bundles. Hence, each mode may have a queuing policy to determine which bundles to keep or to drop.

Some queuing policies have already been evaluated as part of the PRoPHET Internet-Draft:

- FIFO: Handle the queue in a First In First Out (FIFO) order.
- MOFO: Evict most forwarded first
- MOPR: Evict most favorably forwarded first
- Linear MOPR: Evict most favorably forwarded first
- SHLI: Evict shortest life time first
- LEPR: Evict least probable first

It is worth nothing that several queuing policies may be used together in an ordered set.
The queuing mechanism is defined along with PRoPHET in its Internet-Draft.

Two policies have been implemented. FIFO and MOFO, as they appeared to be the most efficient ones according to the evaluation from Anders Lindgren and Kaustubh S. Phans.

Service Layer

The objective here is to set up “service layers” which will be responsible for optimizing the DTN network.

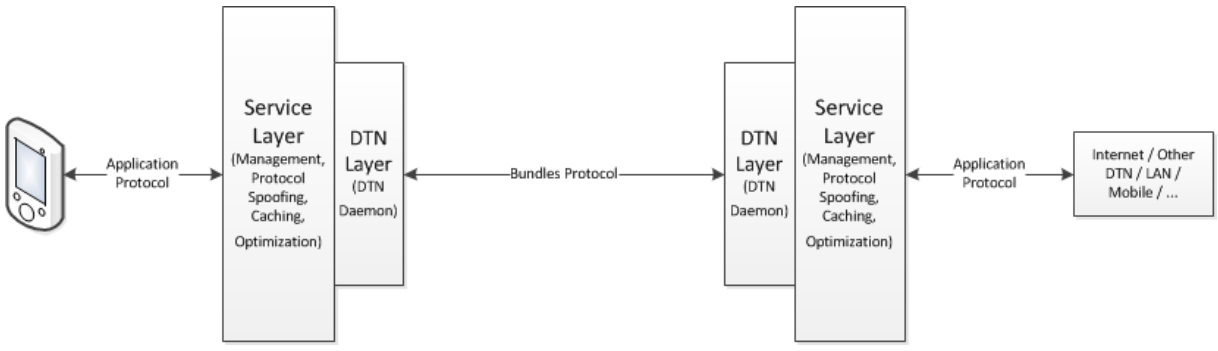

Figure 1

As the figure 2 shows, the streams will be going through proxies before and after going through the DTN. The service layers basically act as proxies between the Bundles protocol and common application protocols. This way, it will help implementing some optimization techniques such as:

- Protocol Spoofing
  In case of chatty protocols (e.g. FTP, IMAP or key-exchange mechanism [6]), it takes the client a tremendous amount of time to process all the consecutive requests, as the delay is already big for a single request in challenged networks. Hence, it would be very useful to modify the protocol such that several requests may be bundled into one request, semantically performing the same operation.
  This feature would also greatly improve applications interoperability. Existing applications designed for Internet would only have to communicate with the service layer, which will handle the transmission of the request to its final destination through DTN.
- Caching
  In order to avoid long requests over DTN and congestion, caching could be used to access the same data over and over.
- Compression
  it might be interesting to see if the bundles could be compressed to deal more efficiently with storage capacity limits.
- Management tools will help getting feedbacks from the DTN network (through the Bundle Status Reports mechanism) and managing the network by defining priorities and firewall rules.
  The Bytewalla 3 application already includes the Bundle Status Reports mechanism; however there is no way for the administrator to have access to these reports and take actions based on them.
  Also, administrators may want to deal with the priorities and add rules according to the situation and DTN environment

Tasks Description

New thesis students on DTN

Two new students are now working with DTN.
During their first month here, they have been studying about DTN and Bytewalla I have been helping them with setting up their system and understanding Bytewalla/DTN.
Also, as they were writing their thesis plan, we discussed and shared the work to do. I will keep working on the objectives as described at first in the thesis plan, and they will be working on parallel objectives.

Bundle priority and queuing mechanism

About the PRoPHET priority mechanism implementation. It has been tested manually, but a test in bigger scale would be more accurate. It is not something easy to test the implement with PRoPHET and limited resources (two servers and two phones).
According to PRoPHET, a phone needs to meet the recipient node before meeting the sender node, to have it in its history when they will be exchanging RIB information and dictionary, following the PRoPHET protocol.
Also, I noticed that the Bytewalla application sometimes crashes unexpectedly. I was not able to reproduce the error and I couldn't allocate much time on investigating it.

Service Layer

Following the priority mechanism implementation, I started to work on the proxy. The proxy, standing "in front" of the DTN daemon, is responsible for capturing the bundles and treating them before they are delivered.
Now, the proxy is able to capture and transfer UDP and TCP traffic. UDP discovery beacons are modified before they reach the "proxified" DTN node. This way, all TCP traffic goes through the proxy.
It's completely transparent for the DTN daemon. Only a few lines have to be changed in the configuration file. There is a small issue with the discovery mechanism, wish I hope can be solved soon.

With the TCP traffic, the proxy captures all the bundles and treats them according to some parameters such as its source, its destination, its type. All "regular" bundles are stored in a MySQL table and their control flags are updated so that they require status reports upon reception.
When the proxy receives the status reports it updates the table so that the bundle is marked as delivered. This way, we can keep track on bundles delivery and with an administration interface, we could do some management.

Besides this, I have developed a tool which will make developing application for DTN, and porting existing application for DTN, easier. On the dtn client, we have a proxy server, which is responsible for converting the application protocol to a bundle. The bundle is then transmitted to its final destination, which is responsible for converting the bundle back to the application protocol.

Based on this work, I have developed a small extension which is capable of capturing the emails sent from a client, and send them as bundles to a server having internet connectivity (here in my experience: dtn://city.bytewalla.com). The city DTN then handles this bundle a send the mail with the SMTP protocol. Hence, it is now possible to send a mail with no internet connection, and the set up is much easier than the one from Bytewalla 1. Moreover, the generic design makes it easy to port any kind of application protocol.

However, to achieve it, I had to add an extension block to the bundles. It seems that the Bytewalla application on Android doesn't support it and crashes. Instead I'm now working only with dtn://village.bytewalla.com and dtn://city.bytewalla.com.

Challenges

- Unplanned time allocated for helping two new students
- Not much help received. Discovering things and developing by myself.
- Inconvenient hardware. Two machines and one screen/keyboard/mouse set. Need to unplug/plug consistently from one on the other. Also, the hardware is quite old and not smooth/slow.
- The Android application doesn't support Bundle Extension block.
- It is not yet possible to send discovery beacons from a proxified DTN node.
  Sending discovery beacons would make other nodes connect directly to the node without going through the proxy. Hence the discovery beacons should be sent directly from the proxy instead of the DTN.

Next Steps

- Solve the discovery issue
- Reliability: since the bundles are stored and confirmations are received with status reports, it is possible to implement a mechanism wish would retransmit a bundle if no confirmation is received.
- Management:
    - Based on the kind of service, we could give an option for managing priorities. E.g. Healthcare service is more important than casual emails.
    - Give an administration interface to bundles management.
- Testing: how long it takes to discovery another node/transfer a bundle, memory footprint, bandwidth...
- Start to write the thesis text.