

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	2011-04-04	Document creation.



Table of Contents

Revision History	2
Introduction.....	4
Thesis Objective.....	4
Bundle priority and queuing mechanism	4
Service Layer.....	4
Management	5
Tasks Description.....	5
Next Steps.....	6



Introduction

This document describes the thesis as of April 4, 2011. It describes the activities I performed recently and the future objectives.

Thesis Objective

As described in the thesis plan:

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.

The priority here is to compare the policies and determine the most suitable ones according to situations, and to design it for Bytewalla application. At least one of the above queuing policy should be implemented.

Service Layer

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

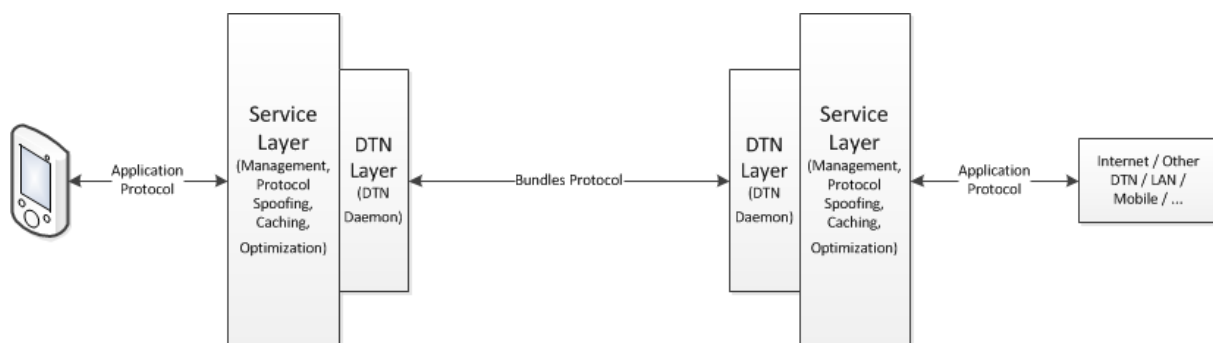


Figure 1

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.

The service layer will be tested with at least one kind of service which is already implemented and running over Internet.

Management

Management tools would 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

During the first weeks I have been studying the documents from the previous projects Bytewalla 1, 2 and 3. All documents are available on the three projects websites:

- Bytewalla 3: <http://www.tslab.ssvl.kth.se/csd/projects/1031352/>
- Bytewalla 2: <http://www.tslab.ssvl.kth.se/csd/projects/1011248/>
- Bytewalla 1 :<http://www.tslab.ssvl.kth.se/csd/projects/092106/>

I also studied the DTN and PROPHET specifications as given in the RFC 5050, 4838 and the PROPHET Internet draft.

Following this period, and as soon as all hardware and file sources were available, around the 2nd and the 3rd week, I started setting up the network architecture as described in the verification report. This was done with some difficulties as the source code files were at first unavailable and the Android phones used for testing the Bytewalla application had issues with UDP broadcasting.

Finally, I could set up the network and start testing the previous Bytewalla projects. All the testing is described in the verification report. This step was to check that the previous Bytewalla projects were



meeting their objectives and functioning as they are supposed to.

Hence, I tested the DTN implementation, the PROPHET implementation, as well as the Sentinel Surveillance Application and the management application.

This was basically the work done for the first month.

Then, I could finally meet with my PhD supervisor and discuss more precisely about the objectives which had to be redefined. Indeed, the feedback mechanism which was supposed to be implemented as part of the thesis, is already part of the Bytewalla application. Also we discussed about my ideas concerning a service layer which could help optimizing and managing the DTN network.

The updated thesis plan has been submitted and as for now, I am still waiting for feedbacks from everyone.

Meanwhile I have started working on the queuing mechanism following the PROPHET specification. I have been trying to design a generic implementation so that the queuing policy may be easily replaced by another.

Two queuing policies are already implemented, FIFO and MOFO (Evict most forwarded first), but still being tested. FIFO is the easiest queuing policy to implement, as we just have to drop the oldest bundle (i.e. the bundle with the lowest id). Concerning MOFO, we need to update the number of transmission for the bundle every time there is transmission event.

Next Steps

Concerning the queuing mechanism, the next step is to verify the implementation and give some comparison between the different queuing policies. The paper "Evaluation of Queueing Policies and Forwarding Strategies for Routing in Intermittently Connected Networks" from Lindgren, A. and K. Phanse gives some recommendations.

Depending on the needs, it may be useful to implement another queuing policy which would be more efficient.

Also, as two other students will start working on Bytewalla, we will have to coordinate and agree on how we will plan things together.

