Bytewalla IV
Implementation of Delay Tolerant Networks on the Android platform

Thesis Plan v1.2

Author:
- Michel Hognerud

Examiner:
- Peter Sjödin

Supervisors:
- Hervé Ntareme
- Bjorn Pehrson
- Danilo Gligoroski

Coaches :
- Doria Avri
- Marco Zennaro

Revision History

Version	Date	Remarks
Draft	2011-02-21	Document creation.
V1.0	2011-03-16	Updates after verification report and discussion with supervisor
V1.1	2011-03-18	Complete objectives
V1.2	2011-03-24	Updated objectives, Issues section and Gantt Diagram

Table of Contents

Abbreviations used in this document

Abbreviation	Description
NordSecMob	Master's Programme in Security and Mobile Computing
KTH	Kungliga Tekniska Högskola
NTNU	Norwegian University of Science and Technology
DTN	Delay-Tolerant Network
PRoPHET	Probabilistic Routing Protocol using History of Encounters and Transitivity
SSA	Sentinel Surveillance Application

Introduction

Background

Nowadays and in many developed countries such as Sweden, Internet is omnipresent and plays a major role in the economy, people social life, research and other areas. More generally, it allows everyone to deliver and to get access to information.

However, there are still places or situations where access to Internet is extremely challenging. For instance, some rural parts of Africa have low levels of access to Internet [1]. This is mainly due to several obstacles including poor infrastructure and high costs of Internet services. Also there are situations where Internet suddenly becomes hardly available. For example, natural disaster governments firewall.

Hence, aiming to bring solutions for these situations, KTH started the development of Bytewalla. Bytewalla has already been through three iterations including two team projects and one Master's Thesis.

Bytewalla is based on Delay-Tolerant Networking (DTN). The Delay Tolerant Networking Research Group is concerned with "how to address the architectural and protocol design principles arising from the need to provide interoperable communications with and among extreme and performance-challenged environments where continuous end-to-end connectivity cannot be assumed"[2]. There are several implementations based on this research, but Bytewalla was the first to implement it a on an Android platform.

While Bytewalla 1 started the implementation on Android [3], and Bytewalla 2 focused on the security issues [4], Bytewalla 3 enhanced the implementation and developed an email application on top of the DTN protocol [5]. People can now send emails within bundles which are carried by mules from the village to the city as shown in figure 1.
In addition, another application named Sentinel Surveillance Application was built for the healthcare system. Doctors can manage and contact patients thanks to Bytewalla.
These two applications stand as proofs-of-concept.

Moreover, PRoPHET (Probabilistic Routing Protocol for Intermittently Connected Networks) was also implemented by Bytewalla 3. PRoPHET uses the history of encounters and transitivity to achieve the best case routing capabilities. Based on the history, it computes the probabilities of delivery for bundles whenever a node neighbor is discovered.

PRoPHET developed at Luleå University of Technology.

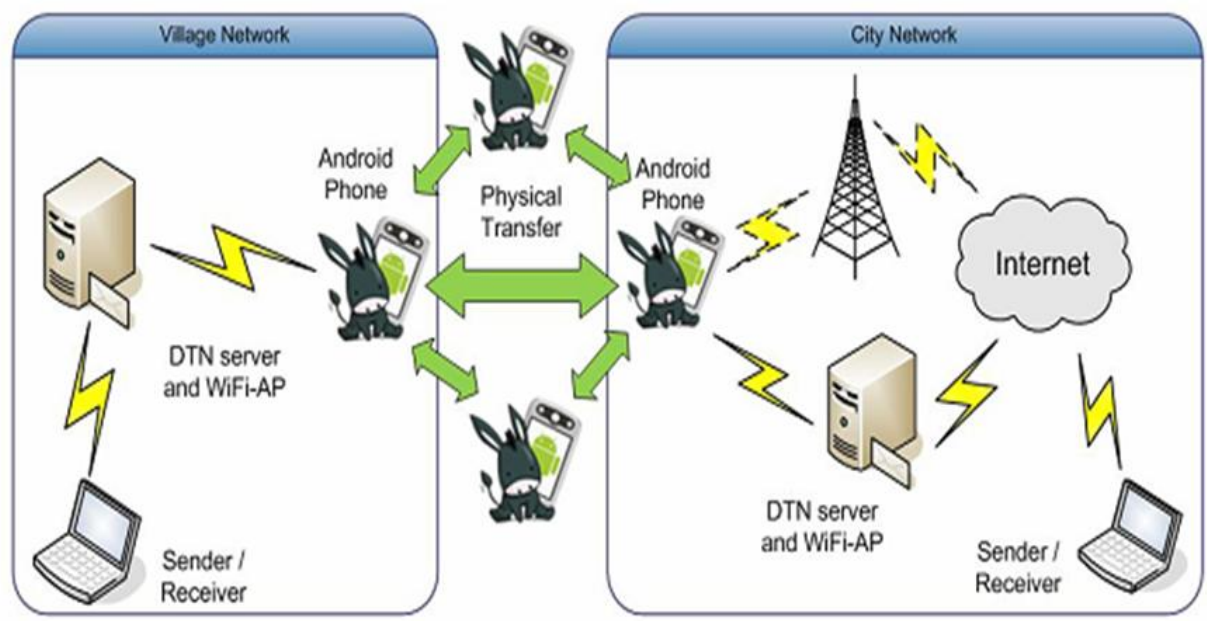


Figure 1

Motivation

The main functionalities have been implemented in Bytewalla 1 and 3 while Bytewalla has been focusing the security aspects.

However, there is room for improvement in optimizing DTN and application interoperability. Hence this thesis aims to make DTN networks more efficient and easier to manage and build or port applications on them.

We hope that it will be improve DTN's penetration into real-world cases as people would then get more control on DTN services.

Goals

Overarching goals

One of the overarching goals of this thesis is to optimize DTN networks regarding the delivery rate and delays. This will be achieved with the help of the existing specifications as well as with new functionalities.

Also, the output of this work should help developer porting and developing applications for DTN, as the challenged networks have very different requirements and capacities compared to standard networks such as Internet.

Approach

As this thesis aims to enhance Bytewalla 3, the first phase will consist of studying the current system along with DTN, bundle protocol and PRoPHET. During this phase, the system will be tested in regards of Bytewalla's 3 objectives. Issues will be reported. Also, new enhancements should be proposed out of the investigations.

Then, will come the implementation part. Issues reported before and new functionalities will be implemented during this period. This period will be followed by an analysis of the results of the thesis. This includes measuring the improvements on the DTN network, comparison with the previous Bytewalla system and comments in the results.

In order to keep supervisors aware of the progress, documents will all be available on the thesis website.

Measurable objectives

- Verification report of Bytewalla 3
    - Investigate Bytewalla 3 functionalities and issues.
    - Determine current issues, missing functionalities, and give suggestions.
- Literature Study (Related Work)
- Implementation according to the verification report and specifications, as described below.

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.

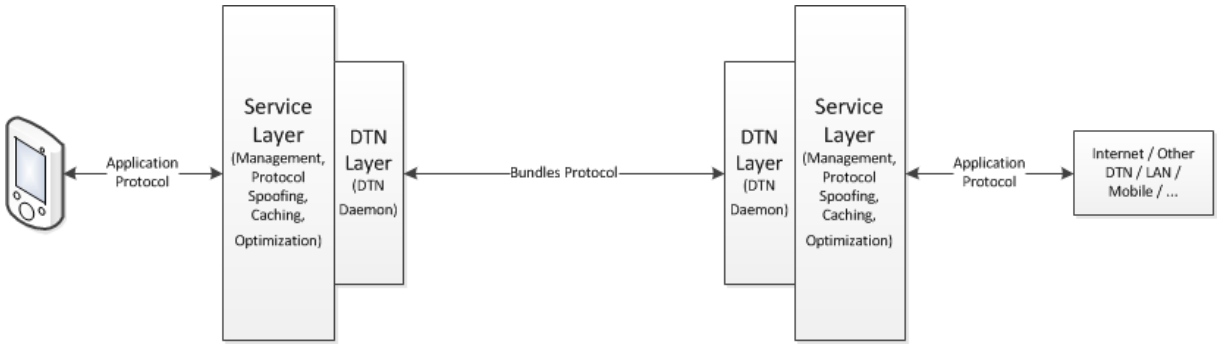


Figure 2

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

Deliverables

Generic Deliverables

- Master's Thesis Project Website
- Thesis Plan
- Thesis Draft
- Final Thesis Presentation
- Final Thesis Report

Master's Thesis Project Specific Deliverables

- Verification Report of Bytewalla 3
- Android Bytewalla Application Source code with queuing mechanism
- DTN Management Application (PHP, Python)
- Service Layer (Python)
    - Caching
    - Protocol Spoofing
    - Optional Additional Optimization Techniques such as compression

Resources

Space

One room containing the hardware is allocated for this thesis, in 8th floor, Forum.

Equipment

Android Phones

To transmit bundles to their destination, two phones are needed. However, to work with PROPHET and the priority mechanisms, three phones are recommended.

As we want to customize the application to a recent version of the Android system, the phones should be on Android 2.1 or more recent.

Two HTC Wildfire 2.1 are now available as well as one HTC Tattoo 1.6.

Servers

Two servers are required. One for the village network and the other one for the city network.

The two servers which were used by Bytewalla 3 will be used as well as part on this thesis.

Here is their configuration as given by Bytewalla 3 [7]:

Specification	**Requirements**
CPU	2.26-GH Core Duo p8400
Hard Disk	300 GB or more
Network Interface	Wi-Fi compatible with IEEE802.11 b/g

Wi-FI Access Points

Three access points are required. One for the village network, another one for the city network, and the third one to setup the intermediate network for Android phones.

Three Ubiquiti Bullets are now providing the three access points.

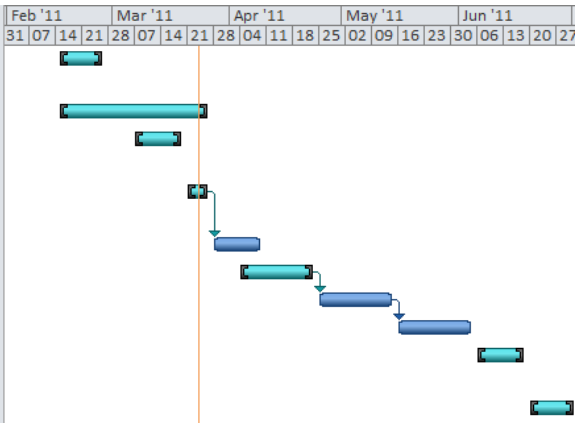
Time Plan

Activity Breakdown

Weeks	Activity Breakdown	Deliverable
1-2 (Feb 14th-25th)	• Review of previous Bytewalla projects. • Literature Study: DTN, Bundle Protocol, PRoPHET etc. • Thesis topic definition.	• Thesis plan • Thesis Website
3-6 (Feb 28th-March 11th)	• Testing and verifying Bytewalla system. • Continue literature study.	• Bytewalla verification report • New objectives / Updated thesis plan
7-15 (March 14th-May 27th)	• Implementation of queuing mechanism • Implementation of the service layer (protocol spoofing, caching) • Implementation of the management tool	• Bytewalla application with queuing mechanism • Service layer • Management tool
16-18 (May 30th-June 17th)	• Implementation analysis and Conclusion	Final thesis draft
19-20 (June 20th-30th)	• Thesis final version	• Printed version of approved final thesis • Thesis presentation

Gantt Diagram

	Task Name	Duration	Start	Finish
1	Review of previous Bytewalla Projects	9 days	Tue 15/02/11	Fri 25/02/11
2	Literature Study	29 days	Tue 15/02/11	Sat 26/03/11
3	Testing and verifying Bytewalla	10 days	Mon 07/03/11	Fri 18/03/11
4	Discussing objectives and updating thesis plan	5 days	Mon 21/03/11	Sat 26/03/11
5	Thesis Draft	10 days	Mon 28/03/11	Fri 08/04/11
6	Queuing Mechanism	15 days	Mon 04/04/11	Fri 22/04/11
7	Service Layer	15 days	Mon 25/04/11	Fri 13/05/11
8	Management	15 days	Mon 16/05/11	Fri 03/06/11
9	Implementation analysis and conclusion	10 days	Mon 06/06/1	Sat 18/06/11
10	Thesis final version	9,5 days	Mon 20/06/11	Fri 01/07/11



BYTEWALLA

Issues

Objectives

One of the main objectives –the Bundle Status Reports- had to be cancelled.

It wasn't really stated whether this had been implemented as part of Bytewalla, but after deep testing of the Bytewalla application, I noticed it was included in the Bytewalla application and working exactly as defined in the Bundles Protocol specifications [8]. However, it will be used anyway as part of the management system.

System Setup

Hardware

The HTC Wildfires 2.1 phones have a manufacturer issue which makes them unable to received UDP broadcast. It wasn't easy to figure out where the problem was coming from but it is confirmed by other people having the same issue: http://code.google.com/p/android/issues/detail?id=8407.

Installation

Some parts of the installation guides from Bytewalla 1 and 3 are sometimes unclear or incorrect. Also, guides were edited by two different groups, depending on the component.

Source Code

The Bytewalla application source code wasn't available until the middle of the second week.

Supervisors availability

It was hardly possible to agree on objectives before mid-March because of their occupations. But from this time, it is all fine.

Contact Information

Royal Institute of Technology (KTH)

Examiner

- Peter Sjödin (psj@kth.se)

Supervisors

- Björn Pehrson (bpehrson@kth.se)
- Hervé Ntareme (ntareme@kth.se)

Coaches

- Doria Avri (avri@acm.org)
- Marco Zennaro (mzennaro@ictp.it)

Norwegian University of Science and Technology (NTNU)

Supervisor

- Danilo Gligoroski (danilo.gligoroski@item.ntnu.no)

References

[1] BBC, “Africa waiting for net revolution”, http://news.bbc.co.uk/2/hi/technology/7063682.stm, Last visited – March 21st, 2011

[2] DTNRG, http://www.dtnrg.org/wiki, Last visited – March 21st, 2011

[3] “Bytewalla 1: Final Report”, http://www.tslab.ssvl.kth.se/csd/projects/092106/sites/default/files/Bytewalla_Final_Report_v1.0.pdf, Last visited – March 21st, 2011

[4] Sebastian Domancich, “Bytewalla 2: Security in Delay Tolerant Networks for the Android Platform”, http://www.tslab.ssvl.kth.se/csd/projects/1011248/sites/default/files/Domancich%20-%20Thesis.%20Security%20in%20DTN%20for%20Android.pdf, Last visited – March 21st, 2011

[5] “Bytewalla 3: Final Report”, http://www.online.kth.se/csd/projects/1031352/sites/default/files/Final%20reportv1.2_0.pdf, Last visited – March 21st, 2011

[6] Jorg Ott, “Application Protocol Design Considerations for a Mobile Internet”, http://portal.acm.org/citation.cfm?id=1186720, Last visited – March 21st, 2011

[7] “Bytewalla 3: Project Plan”, http://www.online.kth.se/csd/projects/1031352/sites/default/files/Project_Planv2.0.pdf, Last visited – March 21st, 2011

[8] Scott & Burleigh , “Bundle Protocol Specification”, http://www.faqs.org/rfc/rfc5050.txt, Last visited – March 21st, 2011

Resume

MICHEL HOGNERUD

ADDRESS
Hanstavägen 883a
164 53 Kista, Sweden
(+46) 0736372946

CONTACT
www.hognerud.net
michel@hognerud.net

OBJECTIVE

Any position as software engineer. Preferably related to web, mobile or network.

EDUCATION / INTERNATIONAL EXPERIENCE

Master's Programme in Security and Mobile Computing (NordSecMob)	Royal Institute of Technology Stockholm, Sweden	August 2010-June 2011
	Norwegian University of Science and Technology Trondheim, Norway	August 2009-August 2010
Exchange Student	University of Iowa Iowa City, USA	August 2008-May 2009
Bachelor in Conputer Science	University de Franche-Comté Besançon, France	August 2006-June 2008

PROFESSIONAL EXPERIENCE

Master's Thesis — Royal Institute of Technology, Stockholm, Sweden — February 2011-June 2011

Implementation of Delay Tolerant Networks on the Android platform.

Technical Manager — Royal Institute of Technology, Stockholm, Sweden — August 2010-January 2011

Developing an "instant-talk" (VoIP) application on the Android platform, with a team of 8 people. Project includes: defining specifications, product development, communication around the project and validation from the project owner. More details at http://nebula.hognerud.net
From February 2011: Continuation of the project with subset of original team for a Swedish start-up.

Freelance — Independant, Telecommuting — July 2007-January 2008, And March 2010-...

Subcontracting for Web-Agencies. Developing a new project for small businesses (to be released).

Research Assistant — University of Iowa, Iowa City, USA — January 2009-May 2009

Contributed to a project whose goal was to automate the detection of malicious script using malicious scripts features. My objective was to automate samples recolting, filtering, and analysis with Weka (Weka is a collection of machine learning algorithms for data mining tasks).

SKILLS SUMMARY

Coding: Java, Python, C(++)
Web Techs: PHP, (X)HTML, XML, CSS, Web Services, Web Semantics
Database: SQL, MySQL
Engineering/Project: Scrum, GIT, SVN, SDL, UML, ASN-1, ProcessAlgebra, Design Patterns
Network: DHCP, MPLS, TCP/IP, DNS, RIP, OSPF, BGP, NAT, SNMP
Security: Web App. Security, Software Security, Fuzzing, Viruses/Rootkits, Cryptography
System: Windows, Android, Linux
Language: English (Fluent), French (Native), Norwegian/Swedish (Beginner)