

# Bytewalla IV

## Implementation of Delay Tolerant Networks on the Android platform

### Verification Report v1.0

**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	2011-03-02	Document creation.



## Table of Contents

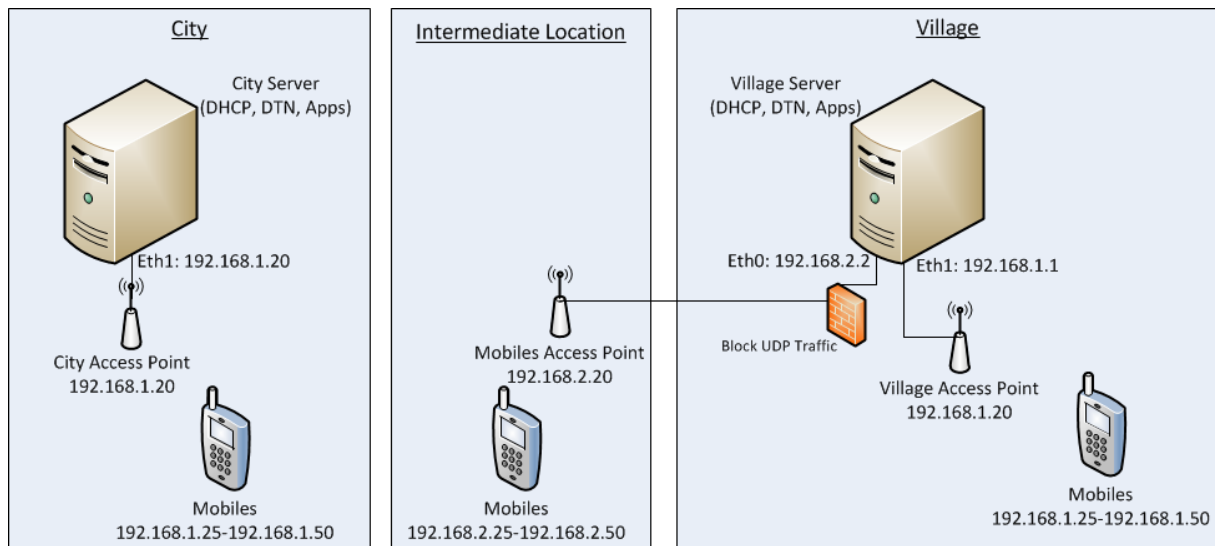
Revision History .....	2
Introduction.....	4
Network Architecture.....	4
Network Components .....	4
Installation/Configuration .....	5
Functionalities Testing.....	6
Hardware .....	6
Bytewalla Verification .....	6
PRoPHET .....	6
Neighbor Discovery .....	7
SSA.....	7
NMT .....	8
Issues Summary.....	9
Hardware .....	9
Bytewalla 3 .....	9
Future work from Bytewalla 3.....	10
DTN Implementation.....	10
Security .....	10
Network.....	10
PRoPHET .....	10
SSA.....	10
Suggestions.....	11
Bundle Status Reports .....	11
Bundle priority and queuing mechanism .....	11
Application Protocol over DTN.....	11
References.....	12



## Introduction

This document aims to verify the existing Bytewalla system, detect issues and bring recommendations on how to solve them and future work.

## Network Architecture



## Network Components

- The servers are running on Ubuntu with DHCP, DTN Daemon and applications such as SSA.
- Access Points: Ubiquiti Bullet
- Mobiles running with Android and DTN Daemon
  - HTC Wildfire 2.1
  - HTC Tattoo 1.6



## Installation/Configuration

The network and DTN applications were set up following the Bytewalla 3 installation guide [[http://www.tslab.ssvl.kth.se/csd/projects/1031352/sites/default/files/Quick%20 Easy%20 Installati on %20Guide.pdf](http://www.tslab.ssvl.kth.se/csd/projects/1031352/sites/default/files/Quick%20Easy%20Installation%20Guide.pdf)], and Bytewalla 1 Installation guide [<http://www.tslab.ssvl.kth.se/csd/projects/092106/sites/default/files/DTN2.6 Installation Guidelines.pdf>].

SSA was set up following the SSA installation guide [[http://www.online.kth.se/csd/projects/1031352/sites/default/files/confmanual\\_0.pdf](http://www.online.kth.se/csd/projects/1031352/sites/default/files/confmanual_0.pdf)].

MNT was set up following the SSA installation guide [<http://www.tslab.ssvl.kth.se/csd/projects/1031352/sites/default/files/Configuration-Manual-of-Network%20Management%20Tool2.pdf>].



## Functionalities Testing

### Hardware

TEST MATRIX			
Number	Test Description	Result	Comments
1	HTC Wildfire 2.1	No	The phones are unable to receive UDP broadcast. <a href="http://code.google.com/p/android/issues/detail?id=8407">http://code.google.com/p/android/issues/detail?id=8407</a>
2	HTC 1.6	Yes/No	One of the two phones is freezing when sending or receiving a bundle. The other one is working properly.
3	Access Points	Yes	Configured and operational
4	City and Village Servers	Yes	Configured and operational

### Bytewalla Verification

TEST MATRIX			
Number	Test Description	Result	Comments
1	Basic Network setup (DHCP, Firewall, ..)	Yes	
2	Wireless router configuration	Yes	
3	City server software installation	Yes	
4	Village server Software Installation	Yes	
5	Bytewalla application installation in the android Phone	Yes	
6	Phone connection with Wi-Fi in City network	Yes	
7	Phone connection with Wi-Fi in Village server	Yes	
8	Message transferred from tattoo to bytewalla village Server	Yes	
9	The message received by bytewalla village server	Yes	
10	The message transferred from bytewalla village server to mobile	Yes	
11	The message received by mobile	Yes	
12	Message transferred from mobile to bytewalla City server	Yes	
13	The message received by bytewalla City server	Yes	

### PRoPHET

Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) enhances the probability, reliability and speed of a message delivery. It computes probabilistic routing decisions based upon the history of encounters between nodes.

PRoPHET was implemented in Bytewalla 3.

TEST MATRIX			
Number	Test Description	Result	Comments
1	Android Phone get connection with server	Yes	
2	Hello SYNC	Yes	
3	Hello ACK	Yes	



4	RIB Dictionary exchange	Yes	
5	RIB Exchange	Yes	
6	Bundle OFFER (Optional)	Yes	
7	Bundle Response (Optional)	Yes	
8	Bundle transfer	Yes	

## Neighbor Discovery

Neighbor Discovery allows nodes to discover each other whenever they are using the same access points. Hence, it is not anymore necessary to use static IPs and routes.

PRoPHET was implemented in Bytewalla 3.

TEST MATRIX			
Number	Test Description	Result	Comments
1	Dynamic ND configuration on android phone	Yes	
2	Dynamic ND configuration on Village server	Yes	
3	Dynamic ND configuration on City server	Yes	
4	Beacons are broadcasted by android phone	Yes	
5	Android Phone can receive beacons from neighbors	Yes	Except HTC Wildfire, see above
6	Android phone interpret the beacons	Yes	
7	Android phones exchange bundle	Yes	
8	Android Phone receive bundle from City Server	Yes	
9	Android Phone receive bundle from Village Server	Yes	
10	Movement of bundle from Village Server to City Server	Yes	

## SSA

The Sentinel Surveillance Application (SSA) is an application running over DTN, developed by Bytewalla 3. It provides access to medical information for the health care system and distributes the information to the village and the city.

TEST MATRIX			
Number	Test Description	Result	Comments
1	Application Installation	Yes	
2	Web Interface	Yes	
3	All back ground services running	Yes	
4	Input from web interface	Yes	
5	File transfer to server	Yes	
6	Bundle creation with time stamp	Yes	
7	Bundle transfer to android phone	Yes	
8	Bundle received at server	Yes	
9	Bundle transfer to client	Yes	

## Comments



- Some changes to the code such as files locations parameters were to be done before the application would start working properly.

## NMT

The Network Management Tool (NMT) is an application aimed to manage the DTN network.

For instances, you can look at the statistics (number of bundles sent/received) and modify the node's configuration.

TEST MATRIX			
Number	Test Description	Result	Comments
1	NMT Installation files	Yes	
2	Web Interface working	Yes	
3	All back ground services running	Yes	
4	Input from web interface	Yes	
5	Changes in dtn.conf file in city server	Yes	
6	Changes in dtn.conf file in village server	Yes	
7	Bundle statics exactness	Yes	

## Comments

- Files locations are not relative (/var/www/).
- The new configuration file still has to be replaced manually.





## Issues Summary

### Hardware

- HTC Wildfire phones: They don't receive UDP broadcast packets. Hence, it is impossible to have neighbor discovery between two of these phones.  
They are still able to communicate with the Village and City servers, because servers receive phones beacons, however there can't be any exchange between mobiles as they can not "see" each others.
- HTC Tattoo phones:  
one of the phone is unable to receive/send bundles. The other one is working fine.

However, the Tattoo phone is able to discover the Wildfire phone, as the Wildfire can still send UDP broadcast, and Tattoo receive it. So this way we can have bundle transfer between phones.

### Bytewalla 3

- The work performed in Bytewalla 2 –on security implementation- was not included in Bytewalla 3.
- A minor change to the code is necessary to handle the configuration correctly.
- Few coding issues (e.g. Directory locations).



## Future work from Bytewalla 3

Here are some potential areas of future work as suggested by Bytewalla 3.

### DTN Implementation

- Recent versions of Android
- Bluetooth neighbor discovery
- An acknowledgment system: A system in which, for each bundle delivered to a destination, the destination sends back an acknowledgement bundle to the source.
  - Extension:  
Mule Pay Back System  
Introduce a mule pay back system by keeping a record of the number of bundles transferred by each mule (Android Phone) and then rewarding the mule based upon this record.

### Security

The NMT should be able to provide an option to encrypt the data communicated to the mules based upon the work done in Bytewalla 2.

### Network

We suggest that powerful server machine should be used to make service availability 24/7. More than one server machine can be installed in the city so that data can transfer as soon as possible. Instead of android phones a specifically designed device, containing Android, can be used for this purpose that can be fixed on a bus and any traveling resource to transfer bundle from village to city and vice versa. We also suggest that AP can be attached with any traveling service that would help in exchanging bundle in the middle of city and village network.

### PRoPHET

- Priority bundles: Bundles can be given priority depending upon their importance. Queuing mechanisms.
- PRoPHET on Linux DTN: It would be efficient to run PRoPHET on the Linux DTN nodes when the PRoPHET compatible to the PRoPHET running on the android DTN node is implemented

### SSA

- Update function: It is sometimes required to update the records; in such case implementing update record function would be a valuable addition.
- Replace function: It can be used to replace some fields of the records.
- Search by disease: For now application searches only using Case\_ID, First Name and Last name; however implementation of search by disease is also a good feature to add.



## Suggestions

### Bundle Status Reports

Bundle status reports provide feedback and information on error conditions related to the bundle layer (ref. rfc4838). The specification for the Bundle status reports is defined in the rfc4838.

There are several kinds of reports: Bundle Reception, Custody Acceptance, Bundle Forwarded, Bundle Deletion, Bundle Delivery and Acknowledged by application.

This mechanism could also be used as part of the Mule Pay Back system.

### Bundle priority and queuing mechanism

Due to a limited storage in mobiles, there should be a priority/queuing mechanism in order to keep the most important bundles in regards of their priority and drop the low priority bundles when necessary.

The RFC4838 is giving some indications on the queuing mechanism.

### Application Protocol over DTN

The development of applications using DTN should be made easier by setting up a set of guidelines and a framework providing basic functionalities for using DTN.

As for now, each application is developed in “its own way” and differs from other implementations.



## References

[http://www.tslab.ssvl.kth.se/csd/projects/1031352/sites/default/files/Bytewalla3\\_Network\\_architecture\\_and\\_PROPHET\\_v1.0.pdf](http://www.tslab.ssvl.kth.se/csd/projects/1031352/sites/default/files/Bytewalla3_Network_architecture_and_PROPHET_v1.0.pdf)

<http://www.tslab.ssvl.kth.se/csd/projects/1031352/sites/default/files/Verification%20and%20Testing%20PROPHET.pdf>

[http://www.tslab.ssvl.kth.se/csd/projects/1031352/sites/default/files/Bytewalla\\_Verification.pdf](http://www.tslab.ssvl.kth.se/csd/projects/1031352/sites/default/files/Bytewalla_Verification.pdf)

[http://www.tslab.ssvl.kth.se/csd/projects/1031352/sites/default/files/Configuration%20and%20Testing%20Dynamic%20neighbor%20discovery%20\(2\).pdf](http://www.tslab.ssvl.kth.se/csd/projects/1031352/sites/default/files/Configuration%20and%20Testing%20Dynamic%20neighbor%20discovery%20(2).pdf)

<http://www.tslab.ssvl.kth.se/csd/projects/1031352/sites/default/files/FDDSSAArchitecture1.pdf>

