A Service Adaptation Middleware for Delay Tolerant Network based on HTTP Simple Queue Service

Hao Zhuang [1,2], Herve Ntareme[2],
Zhonghong Ou[1], Bjorn Pehrson[2]
1.Aalto University, Finland
2.Royal Institute of Technology, KTH, Sweden

Aalto University

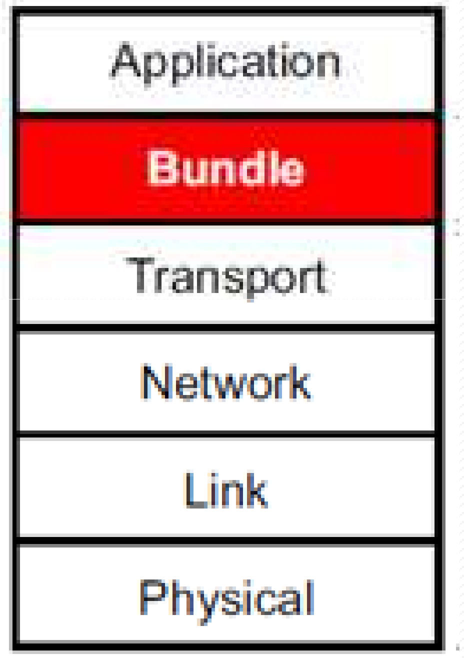
KTH VETENSKAP OCH KONST

Motivation

- Communication-challenged area
  - Environment
  - Health
- Services in need
  - Environment monitor
  - Secure drug distribution
- Delay tolerant network
  - Establish network quickly

Delay Tolerant Network

- Store-forward
  - Bundle Layer
  - RFC 5050
- DTN Prototypes
  - DTN2 : Linux C/C++
  - Bytewalla : Android



Application

Bundle

Transport

Network

Link

Physical

Problems

- Applications based on different development platform
  - Java, Python, J2ME and C/C++
- Applications deployed in different OS
  - Bifrost, Voyage, Ubuntu, Android
- Different ways to communicate with DTN service daemon
  - Embeded Linux command in Java or C
  - Shellscript
  - Python

Message-oriented Middleware

Background

Hardware

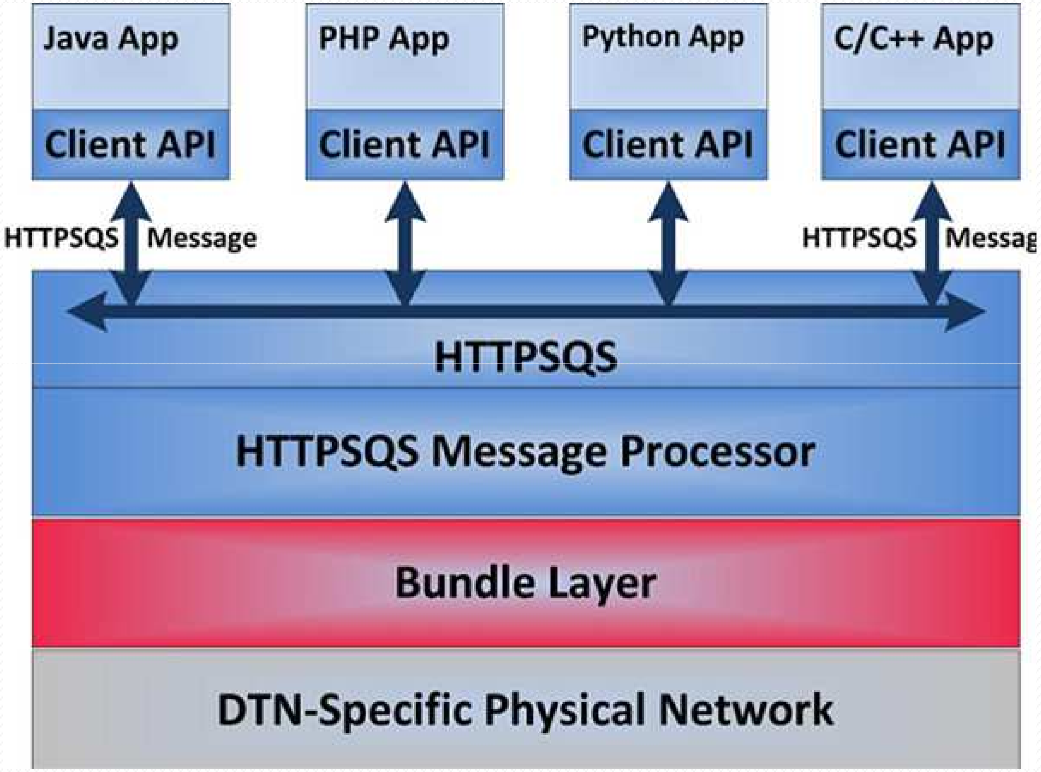
- ALIX board computer
  - Voyage Linux as DTN gateway
  - Bifrost Linux as DTN router
- Sun SPOTs
  - collects light level, temperature and battery
  - Small-footprint JVM
    - host multiple applications
    - no underlying operating system

Background
Software

- DTN Service Daemon
  - DTN2 : Linux C/C++, ALIX Board
  - Bytewalla : Android , HTC Desire Android Phone
- HTTPSQS
  - Lightweight MOM to provide HTTP Simple Queue Service
  - Only 800 lines source codes and easy to second development

Architecture

- HTTPSQS
  - Simple Queue Service
- Client API
  - Java,PHP,Python,C/C++
  - DNT2 API
- HTTPSQS Message Processor
  - Serialize and deserialize
  - Request and reply



Message Design

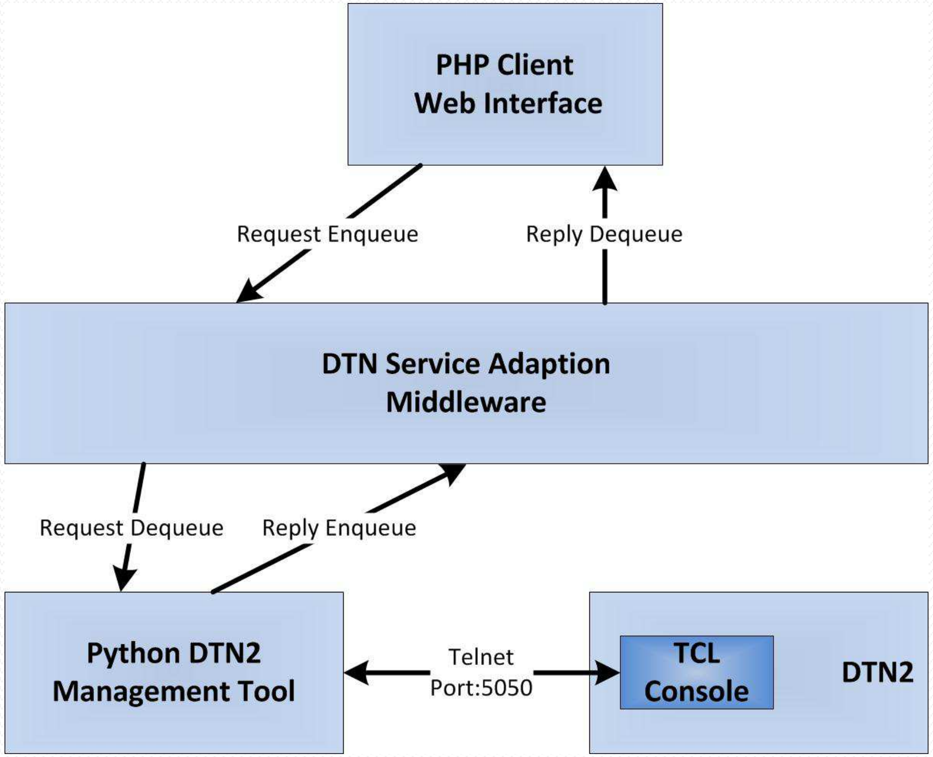
- Text Message
    - command [option]， …
        - Bundle stat/list/
        - Route dump/add/del
    - Simple and but poor extensibility
- XML Message
    - QoS control: expire/priority/correlation
    - Better extensibility but consumes more computing and power

HTTPSQS Message Processor

- Message serialize and deserialize
  - external store and transfer → serialize
  - External to local representation → deserialize
- Request-reply processor
  - Request, Reply, Retry and Dead queues
  - Message dispatcher process

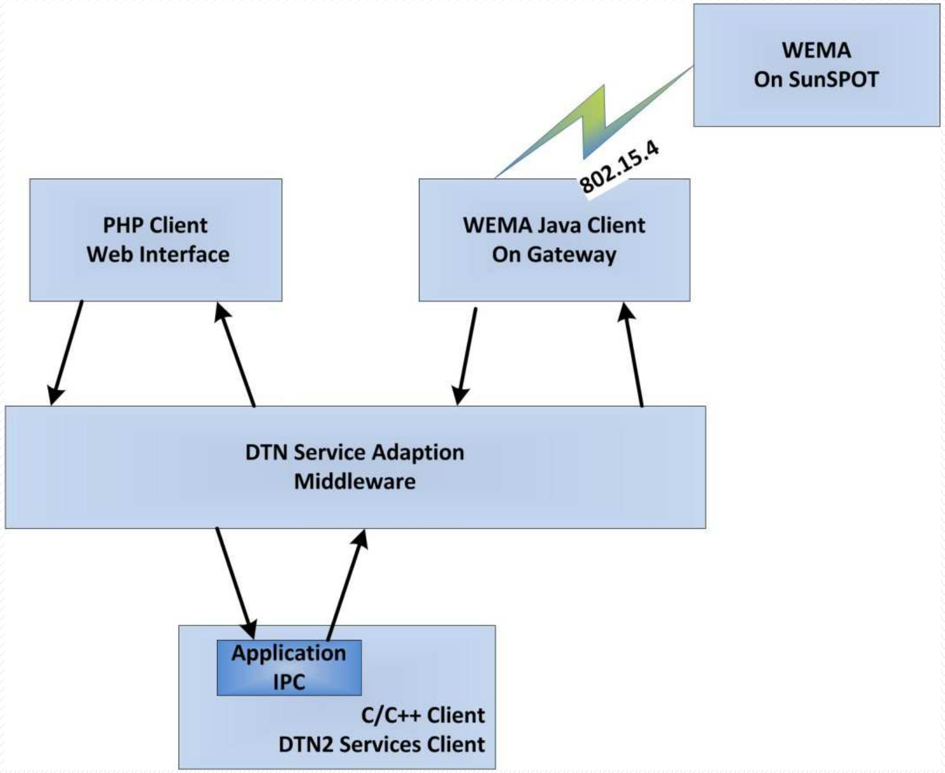
DTN2 Network Management Tool

- Provide a user-friendly DTN network management tool
  - Bundles stat/list
  - Route dump/add/del
  - Link add/del
- Socket communication between DNMT and DTN2

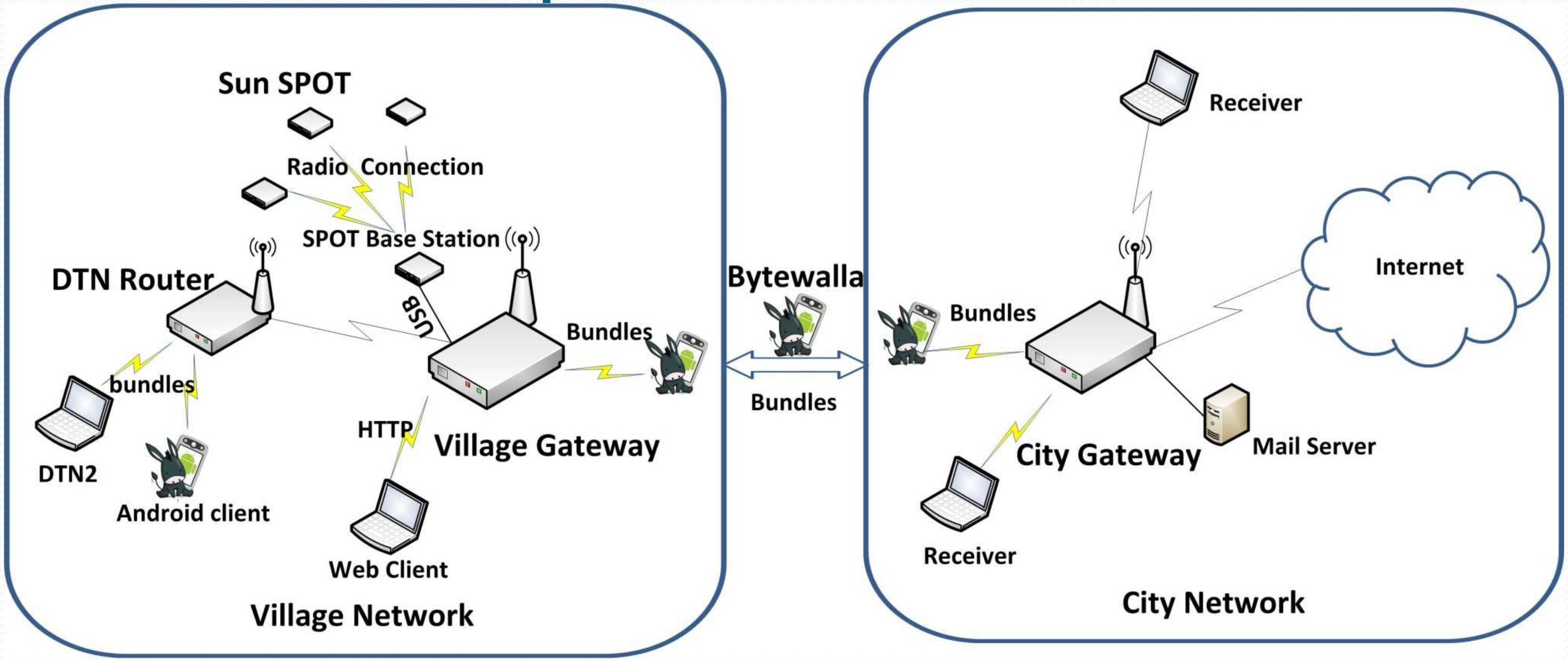


Environmental Monitoring Application

- PHP web client
  - User Interface
- Java client on SunSPOT
  - Collect data
  - Broadcast data
- Java client on DTN Gateway
  - Setup and teardown connection
  - Process data
- C/C++ DTN2 Services Client
  - Httpdtnsend/httpdtnrecv
  - Httpdtncp/httpdtncpd



Evaluation
Network Setup



- Cover all the DTN nodes : Gateway, Router and Host
- Bytewalla: DTN Mobile router based on Android
- 2 base station and 10 SunSPOTs

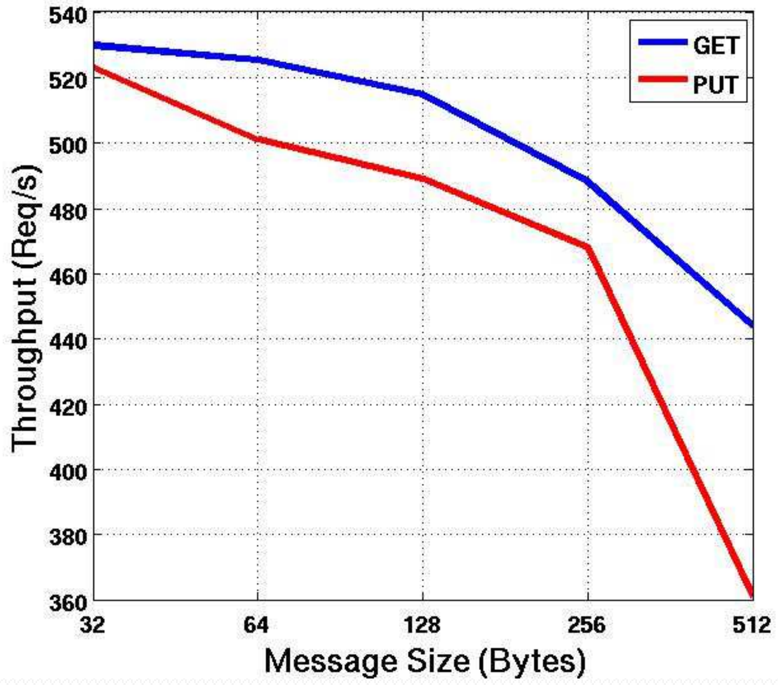
Evaluation
DTN2 Send/Recv Bundles

File Size(MB)	Time(Seconds)	Throughput(MB/s)	Power(Watt)
100	199	5.14	4.975
70	199	5.14	4.912
50	177	5.79	4.962
30	159	6.44	4.941
20	154	6.65	5.032
10	194	5.28	4.802
1	338	3.03	4.973

- Send 1 GB data in different file size
- Best file size 20MB-30MB

Evaluation
DSAM Put/Get Httpsqs Message

- Put/Get 10,000 HTTPSQS Message with various size
- Power remains constant at 4.6 watts
- When size is same, Get are large than PUT
- When msg size > 256 bytes, throughput decreases dramatically



Conclusion

- Provide a communication layer between DTN service daemons and different applications
- Develop two applications to validate the effectiveness of DTN Service Middleware
- Provide an environmental monitoring solution for developing regions

Thank you