IBR-DTN
A lightweight, modular and highly portable Bundle Protocol implementation

Sebastian Schildt, Johannes Morgenroth, Bastian Pöttner, Lars Wolf
WASA-NGI 2011
Outline

Introduction

Architecture and Usage

Evaluation

Conclusion
Delay- and Disruption tolerant networks are networks which do not provide continuous end-to-end connectivity
  - Store-and-Forward principles are applied under these conditions
Bundle Protocol (RFC 5050 and friends) is a protocol specification for realising DTN networks
  - It is implemented - among others - by DTN2, which acts as a reference implementation for Bundle Protocol
DTN Applications

DTN@IBR

- Equipping trams with DTN nodes
- Cooperation with BBR Verkehrstechnik
- Update passenger information displays

Other applications
Interplanetary Communication, Distributed Sensing, VANET’s and providing connectivity to sparsely populated rural areas
Lightweight and Portable

Targeted platforms

- Embedded Linux platforms (OpenWRT as main target)
  - (We are POSIX compliant, so porting to others OS’s should be feasible)
- Scalable to “Big Iron”, no compromises compared to DTN2 running on similar hardware

Not Targeted

- No sensor nodes with 8 bit µCs
  - (Although we are doing some work for the convergence of BP based DTNs with classical sensor node platforms)
## Supported Platforms

<table>
<thead>
<tr>
<th>CPU Architectures</th>
<th>Software Stack</th>
<th>Hardware Platforms</th>
</tr>
</thead>
<tbody>
<tr>
<td>x86</td>
<td>OpenWrt</td>
<td>Ubiquiti RouterStation Pro</td>
</tr>
<tr>
<td>x86-64</td>
<td></td>
<td>Memsic (CrossBow) iMote2</td>
</tr>
<tr>
<td>MIPS</td>
<td>uClibc</td>
<td>QNAP TS219-P NAS</td>
</tr>
<tr>
<td>ARM</td>
<td>Linux 2.6, Linux 2.4</td>
<td>Buffalo TeraStation Pro NAS</td>
</tr>
</tbody>
</table>
Modular architecture

Written in C++
Interacting with the DTN Daemon

C++ API

- Link IBR-DTN C++ library to your program
- Communicates with the DTN daemon using a TCP or Unix socket
- Full flexibility, but can be complex

Commandline Tools

- Diagnostics: dtnping, dtntracepath
- Simple data transfer: dtnsend, dtnrecv
- DTN enabled scripting: dtntrigger ("Poor man’s DTN API")
Storage Backends

Memory

- Non persistent, RAM based

File

- Persistent, file based
- Bundles survive scheduled daemon restarts as well as power failures

SQLite

- Stores more meta information for bundles
- Useful for more complex routing modules
Routing modules

Static

- Routes and available neighbours are configured statically

Neighbor

- Route packets to nodes seen by a Discovery Agent
- When using IPND or DTN2 announcements, nodes within the same subnet are reachable

Epidemic

- Epidemic routing (a flooding scheme) implementation
- Uses Bloom filters for summary vectors and is extended with purge vectors
API Send Performance

Send 100 bundles using dtnsend and store in respective storage

- ~ 265 MBit peak throughput
API Receive Performance

Receive 100 bundles using dtnrecv from respective storage

- ~ 250 MBit peak throughput
Transfer Test

- 10 runs of 1000 bundles for each size
- \( \sim 310 \text{ MBit peak throughput} \)
Conclusions

- IBR-DTN provides a full featured Bundle Protocol stack, which is interoperable with DTN2
- Specifically targeted for uClibc based embedded Linux platforms but scalable to non-embedded environments
- Comparable or better raw performance compared to DTN2
- Modularized, lightweight C++ codebase
- Available from our website to be used in your projects now!
Thank you! Questions?

Sebastian Schildt
schildt@ibr.cs.tu-bs.de

http://www.ibr.cs.tu-bs.de/projects/ibr-dtn/

New version 0.6 released yesterday evening!
API receive (retrieve)
Throughput

Bundle Throughput [MBit/s] vs Bundle Payload Size [bytes] for DTN2, IBR-DTN, and ION (T).

DTN2, IBR-DTN, and ION (T) are shown with different markers and line styles in the graph.

Sebastian Schildt | IBR-DTN | 18
Chunksize

Throughput [MBit/s] vs. Bundle Payload Size [bytes]

- 1024 Bytes
- 4096 Bytes (Default)
- 10240 Bytes
- 102400 Bytes
- 1048576 Bytes
- 10485760 Bytes

4096 Bytes, NODELAY
Caching / Sync

Throughput [MBit/s] vs Bundle Payload Size [bytes]

Default
Caches cleared
Caches cleared, Sync