Enabling DTN-based Web Access

The Server Side

Lauri Peltola

Supervisor: Prof. Jörg Ott
Delay-tolerant Networking

Delay-Tolerant Networking (DTN) is an approach to computer network architecture that seeks to address the technical issues in mobile or extreme environments that lack continuous network connectivity.

– Wikipedia
Delay-tolerant Networking

• Designed to work in situations where traditional Internet protocols (TCP) will fail

• No end-to-end path, high delay, high packet loss

• Examples: interplanetary communication, mobile ad-hoc networks, sensor networks
Example: Interplanetary Communication

- Very high propagation delays: e.g., from Earth to Mars 4–20 minutes
- TCP will not work!
DTN Architecture

• Send data in variable-length packets, called “bundles”

• Store-and-forward operation, similar to email

• Opportunistically move the bundle closer to the destination, hop-by-hop

• Minimize round-trips: no connection setup, optional acknowledgments
Objectives of Thesis

1. Conceptualize DTN-based web access, i.e., design the mechanisms needed to run HTTP on top of the DTN architecture

2. Implement the concepts in a web server application
HTTP-over-DTN

- The problem with HTTP is that it is a conversational protocol – web resources are fetched one-by-one
- A web page comprising 10 resources requires 10 round-trips to the server – in a high-delay environment this is unacceptable
- Solution: resource bundling
Resource bundling

• Instead of retrieving resources one-by-one, aggregate them into a single bundle

• Ideally, this reduces the number of round-trips to just one

• Bundle the resources in some suitable format, e.g. MHTML (used in our implementation)

Bundle
MHTML

index.html
style.css
jquery.js
image1.gif
image2.gif
Resource bundling

Without resource bundling:
- **Client**: Request: `index.html`
  - **Response**: `index.html`
- **Client**: Request: `styles.css`
  - **Response**: `styles.css`
- **Client**: Request: `logo.png`
  - **Response**: `logo.png`

With resource bundling:
- **Client**: Request: `index.html`
  - **Response**: `index.html, styles.css, logo.png`
Resource bundling

- Server needs to identify which resources should be bundled
- Explicitly: create a dependency file that contains the structure of a website
- Implicitly: parse the source of an HTML document and find the dependencies
- Our implementation supports both
Implementing a DTN-enabled web server

- Goal: create a web server with native support for DTN that implements the designed resource bundling scheme
- Don’t start from scratch – build on an existing open source web server
- Implementation is built in the Ruby programming language
Implementing a DTN-enabled web server

- The implementation is based on an open source server called Mongrel
- 2500 lines of code – simple enough to understand (Apache 2.0: ~90000 LOC)
- Written mostly in Ruby – easy to extend
- Tested, stable, fast – can be used in production
Implementation: Features

• Resource bundling in MHTML format
• Dependency file and parser-based resource gathering
• Caching of outgoing response bundles
• TCP and DTN interfaces can be used simultaneously
Implementation: Measurements

Measurements were conducted to observe server performance in both low-delay and high-delay environments.

Page retrieval times of www.mozilla.com (seconds):

<table>
<thead>
<tr>
<th>Delay</th>
<th>DTN</th>
<th>TCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>50 ms</td>
<td>0.62</td>
<td>0.89</td>
</tr>
<tr>
<td>100 ms</td>
<td>0.66</td>
<td>1.45</td>
</tr>
<tr>
<td>500 ms</td>
<td>1.12</td>
<td>5.48</td>
</tr>
<tr>
<td>1000 ms</td>
<td>1.69</td>
<td>10.49</td>
</tr>
</tbody>
</table>
Future Work

• The implementation works with static files – how to deal with dynamic content? AJAX technology is especially problematic

• Security issues: e.g., DoS prevention
Thank you
Implementation: How it works

Socket processing

HTTP parser

HTTP handlers

TCP interface

TCP data

HTTP data

HTTP response

HTTP request
Implementation: How it works

Socket processing

HTTP parser

HTTP handlers

HTTP response

HTTP request

Resource fetching

DTN interface

TCP interface

Bundles

TCP data

HTTP data