

# 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!

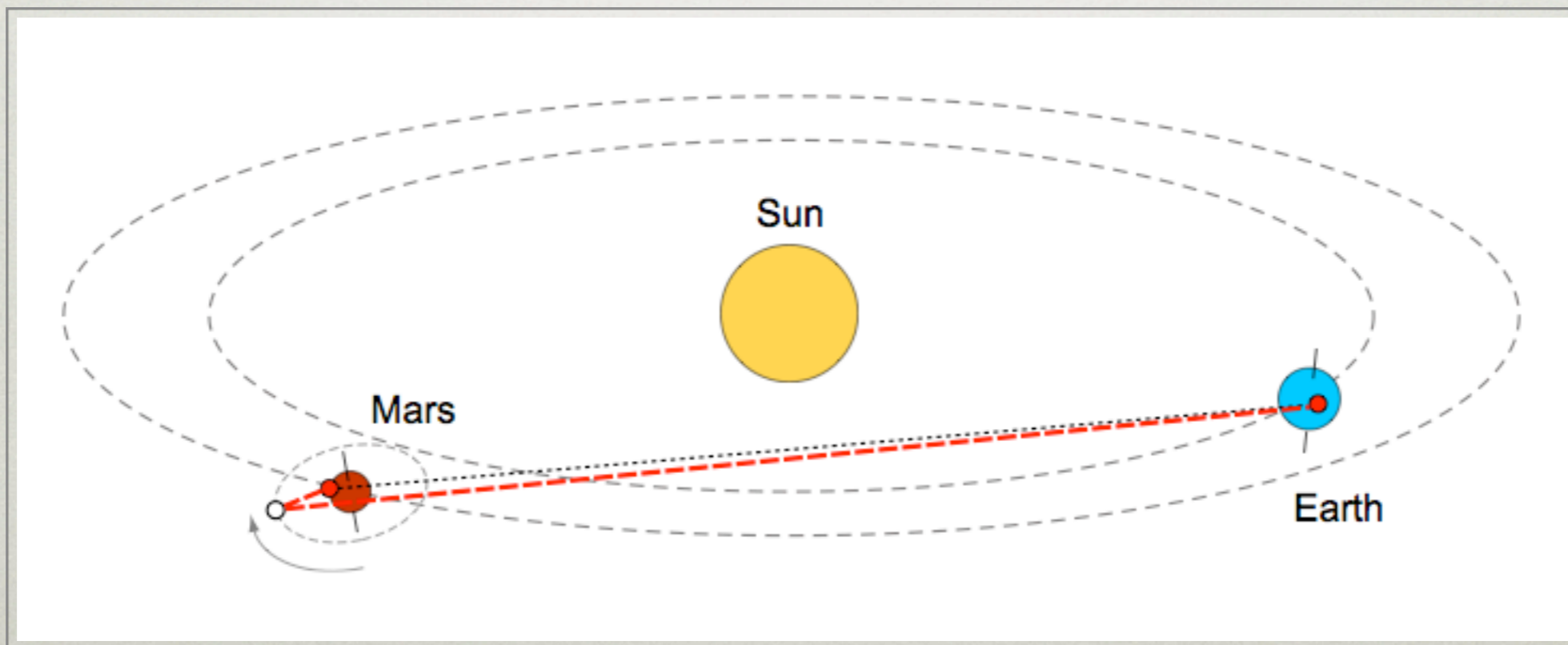


Figure: Forrest Warthman

# 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

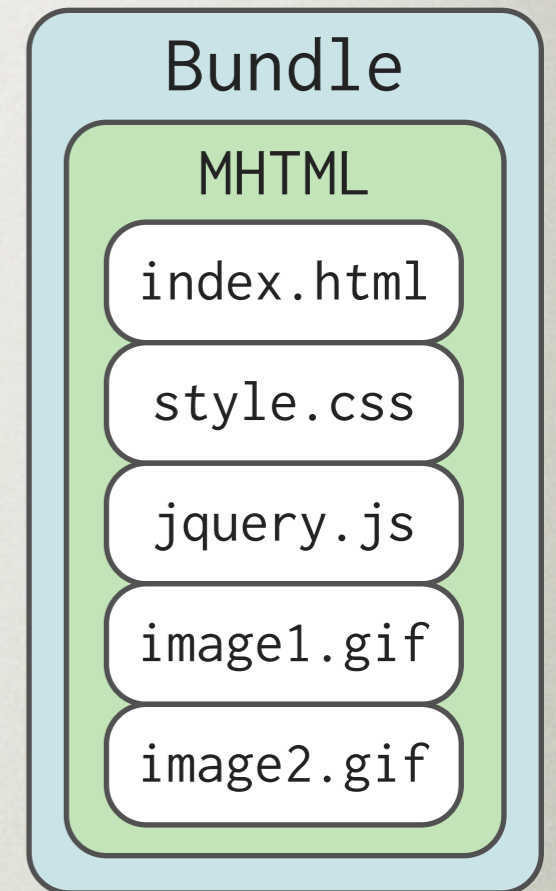
---

- 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)

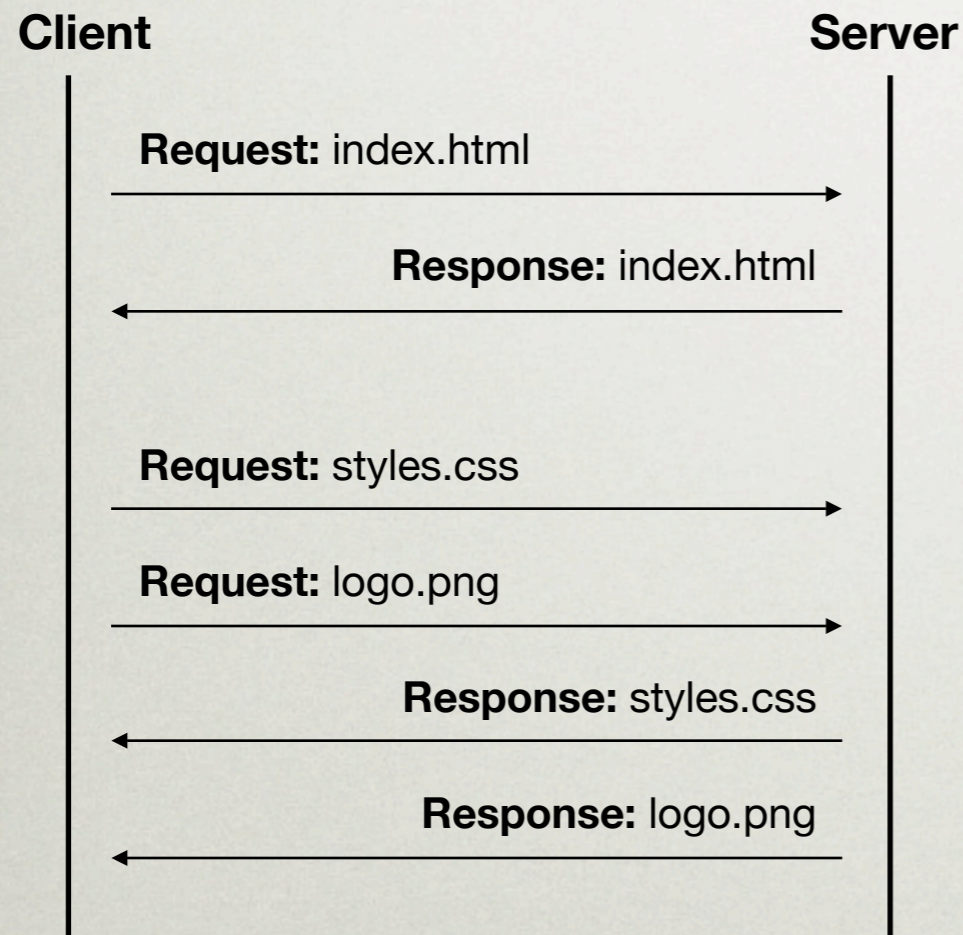




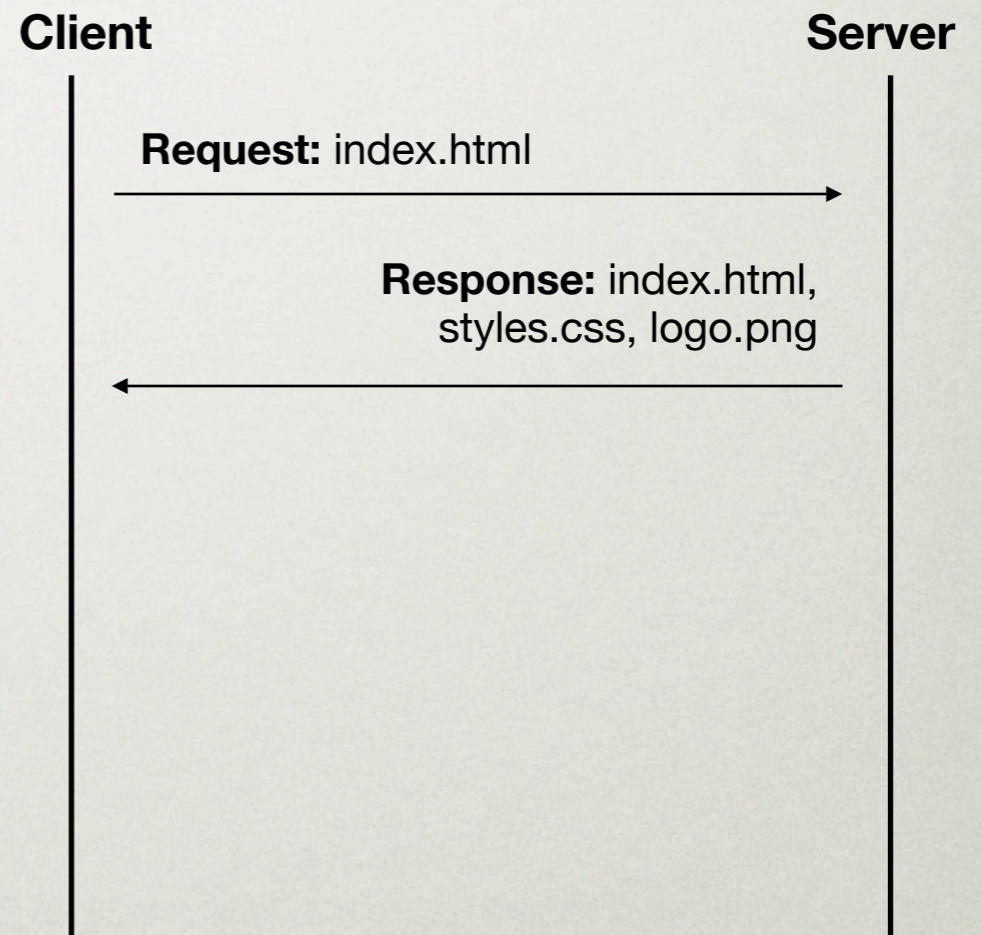
# Resource bundling

---

Without resource bundling



With resource bundling



# 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](http://www.mozilla.com) (seconds)

<b>Delay</b>	<b>DTN</b>	<b>TCP</b>
<b>50 ms</b>	<b>0.62</b>	<b>0.89</b>
<b>100 ms</b>	<b>0.66</b>	<b>1.45</b>
<b>500 ms</b>	<b>1.12</b>	<b>5.48</b>
<b>1000 ms</b>	<b>1.69</b>	<b>10.49</b>

# 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

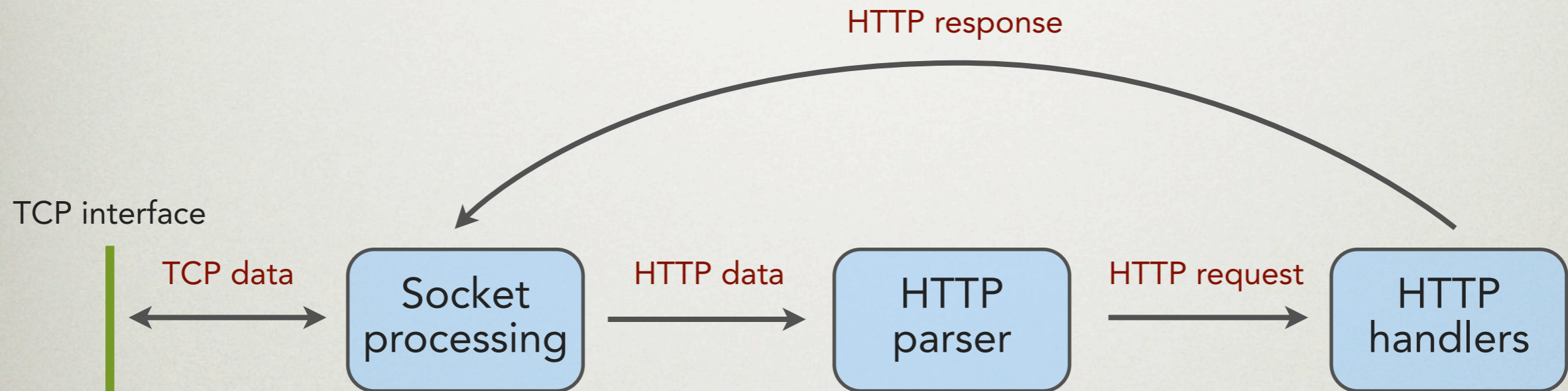
Thank you

---



# Implementation: How it works

---



# Implementation: How it works

