Week 4 – Wireless Networking, measurement and Internet Connectivity

Su, Scott, Hui, et al. – Haggle: Seamless Networking for Mobile Applications

Adam Goldstein
abg@wpi.edu
CS 525w – Mobile Computing (02/15/2011)

Haggle: The idea

- **Separate application logic from transport bindings**
  - i.e. applications can focus on their task; someone else will make sure the data gets in an out successfully.
- **Authors provide a proof of concept for web browsing and e-mail.**

Haggle: The idea

- **Haggle uses late binding**
  - Don’t worry about network connectivity etc. until actually trying to transmit data
- **Applications can communicate**
  - Share data and metadata
- **Manage local and shared resources**
  - Options/preferences for all sources can be managed on any device

Author Examples

- **Send an e-mail to person next to you**
  - Ideally use Bluetooth or 802.11.
  - In practice, phone->e-mail server, recipient’s e-mail server->phone. Slow!
- **Reading news while on public trans**
  - Internet connection dies, goodbye news.
  - Ideally, we can borrow the same news stories from browsers all around us.

Our Examples

- **A few additional ideas:**
  - – Shared GPS information
  - – Haggle chains. Devices like repeaters.

- **What else could be made possible?**

- **Concerns?**

Haggle: How it works

- **Just-in-time binding**
  - – Provide alternate routes for data when usual channels are slow or unavailable
- **Persistent data/metadata**
  - – Stored as key/value pairs, united in direct relationships, ownership & dependency
- **Centralized resource management**
  - – Device preferences define behavior

Just-in-time binding

Connectivity Interfaces

- **Must support many networking technologies**
  - – Differ by range, latency, bandwidth, cost, availability, power, etc.
- **Connectivity - A schedulable resource**
  - – Even two of the same kind of connection are considered separate resources
- **Haggle currently focuses on 802.11**

Just-in-time binding

Protocols and Forwarding

- **Different protocols, different needs**
  - – HTTP -> web server & request objects
  - – P2P -> direct in and out from a peer
- **Once connection is made, forwarding**
  - – Haggle can maintain multiple connections and can forward to all
  - – Choices are made by running many algorithms in sync

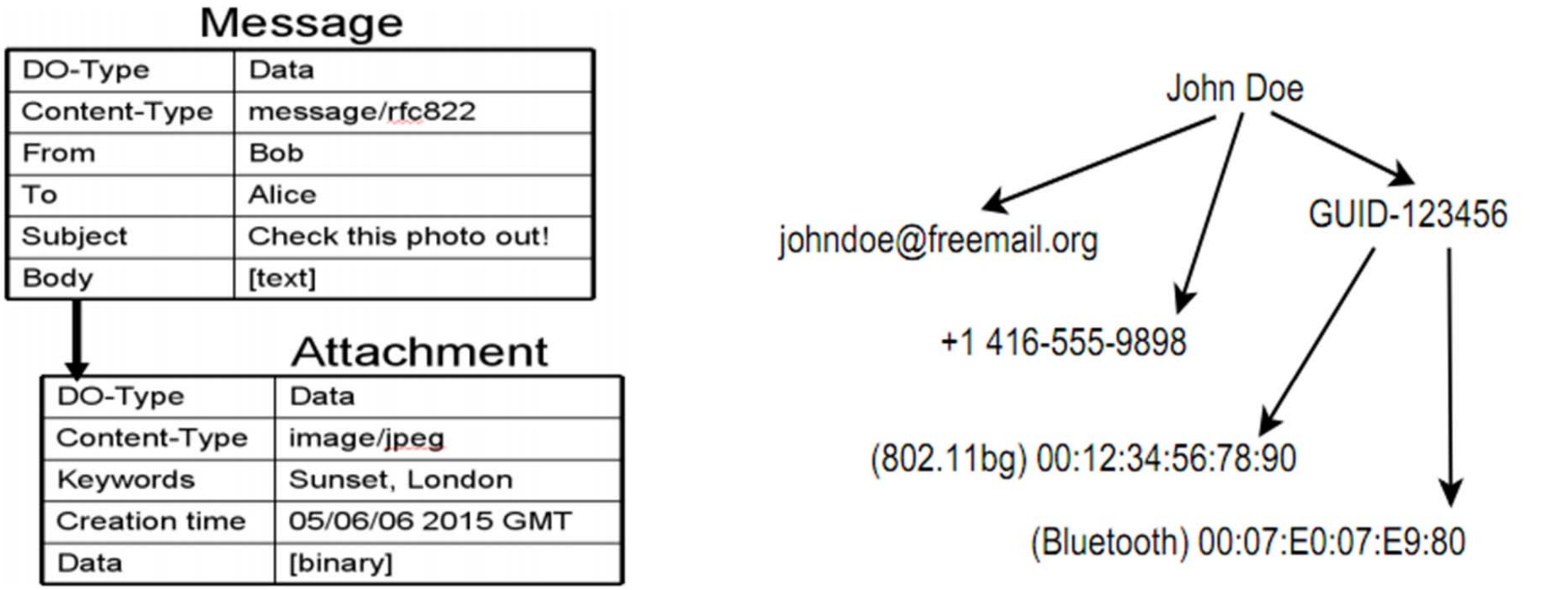
Just-in-time binding

Forwarding algorithms

- Epidemic – Spread to all like a virus
- MANET – [Minimum Exposed Path to the Attack] in Mobile Adhoc Network. Can be based on:
  - Geography
  - Distance-vectors
  - Mobility-based
  - Store and forward

Just-in-time binding

Naming for Forwarding Algorithms



Message

DO-Type	Data
Content-Type	message/rfc822
From	Bob
To	Alice
Subject	Check this photo out!
Body	[text]

Attachment

DO-Type	Data
Content-Type	image/jpeg
Keywords	Sunset, London
Creation time	05/06/06 2015 GMT
Data	[binary]

(a) Message and Attachment

(b) Name Graph

Fig. 2: Example Data and Name Object Graphs

Data Management

Data Objects

- Haggle data is structured & searchable
  - Information is findable and searchable for Haggle and its client applications
  - Think: Google Desktop
- **Data objects are type/value pairs**
  - Usually strings, binary also works
  - Metadata is usable, encouraged, but not mandatory

Data Management

Relationships

- Data is connected
- Can represent prerequisites
  - Photo album links to its pictures
  - E-mail links to its attachments
  - Webpage links to
- **Can represent ownership**
  - Browser owns cached items
  - Mail client owns stored e-mail

Scheduling and Managing

Data Objects

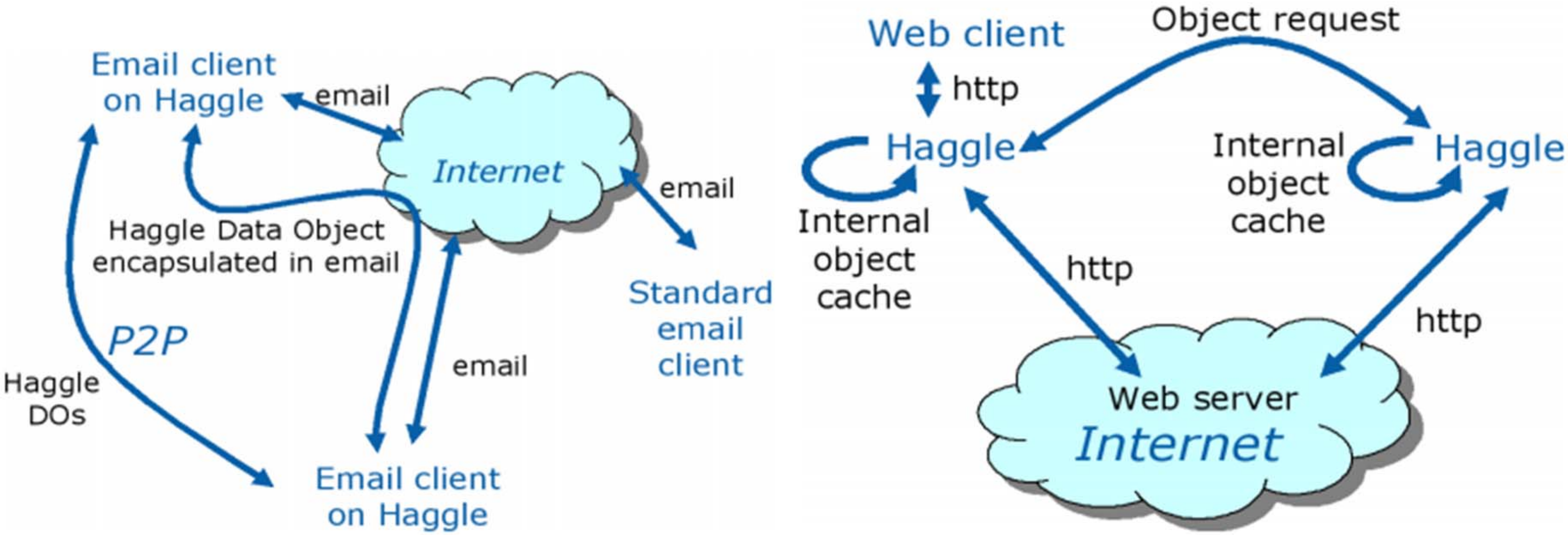
- Resource manager schedules tasks
  - Operations are asynchronous or immediate
- Priority can vary over time as interfaces because more and less costly
- Tasks can ask for extensions
- The shared data management is utilized with just-in-time binding to make these scheduling decisions.

Haggle: Existing Applications

E-mail

- Consists of two elements:
  - SMTP/POP proxy for e-mail clients
  - SMTP/POP protocols for e-mail servers
- Haggle acts as an intelligent mailbox
  - If connected to the internet, send away
  - If not, client sends through proxy, Haggle uses available network interface to find easiest path out the door.

Haggle: Existing Applications



(a) Email Application

(b) Web Application

Fig. 3: Haggle Email and Web Applications

Haggle: Existing Applications

- Other features that would be nice?
- Data or relationships we can store for browsers or e-mail clients?
- More existing applications that would be improved by Haggle?

Haggle: Experiments

- Deployed using Java J2ME CDC
  - Useable on laptops and mobile platforms
- Experiments were conducted with two Windows XP machines.

Haggle: Experiments

E-mail

- Used Gmail
  - – Has a 10 MB cap for outgoing messages
- Sent messages from one laptop to other
  - – 0 bytes < sent messages < 10MB
  - – Faster performance with Haggle when allowed to ad hoc transmit messages

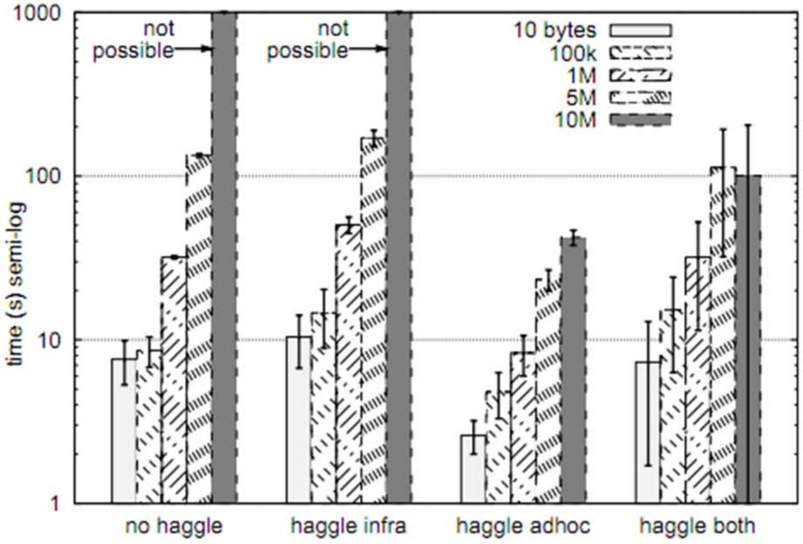
Haggle: Experiments

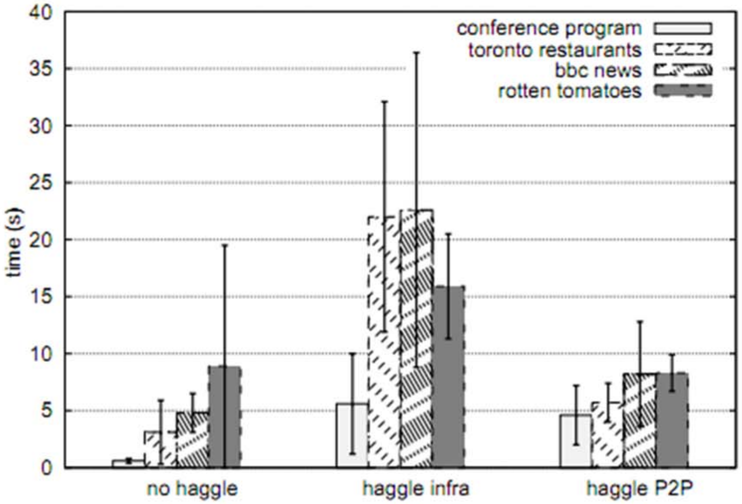
Browser

- Used Firefox with FasterFox plug-in
- Measured with four different web sites
  - Different characteristics, like text heavy, image heavy, update heavy, etc.
  - Tried 7 times, cleared cache before each
- Did not give better performance
  - Possibly due to parsing overhead, HTML parsing time, inefficiencies in the data manager

Haggle: Experiments

Results





Haggle: Experiments

Thoughts

- What do you think about their experiments? And the results?
- What other testing would be useful?

Haggle: Discussion

Authors' future work

- Future ideas:
  - – Resource-friendly media sharing
    - Sync with home, share with friends
  - – Predictive/preemptive browser fetching
- Preferences, preferences, preferences!

Haggle: Discussion

What we think

- Is Haggle a good idea? Are there additional good uses for it?
- How successful were the authors?
- How feasible is “good enough” security?
- Ideas for apps in a related field?