Consistency Challenges
of Service Discovery in
Mobile Ad Hoc Networks

Christian Frank
ETH Zürich*, Switzerland

Holger Karl
Universität Paderborn*, Germany

*presented work created while both authors at
Technische Universität Berlin
Scenario

- Users form ad-hoc network
- Usage scenarios:
  - Find nearest store
  - Hail taxi cab
  - Connect to access point
- Stores, cabs, etc. publish services on the network
- Clients perform “Service Discovery“
Requirements

- Service-Discovery
- Efficient at times of few requests
  - Only act on-demand (reactive)
  - No requests → no network traffic
- Efficient when requests frequent
  - Prepare information in advance (proactive)
- Completely independent of infrastructure
- Post-discovery routing
Design Goals

- **Re-use routing functionality:**
  - On-demand (reactive) routing and service discovery similar
    - Request
    - Reply
    - Caching
    - Expanding ring search

- **Extendable architecture**
  - Exchangeable routing component
Related Work

- **Managed Networks**
  - Service Location Protocol
  - UPnP
  - Jini Lookup Service
  - Intentional Naming Service (INS)

- **Ad Hoc Networks**
  - Multicast trees (e.g., L. Cheng, CSCW’02)
  - Integration with routing (Koodli, Perkins, *Internet Draft ’02*)
Outline

- Introduction
- Design / Implementation
- Model
- Results
- Conclusion
Approach [kp02]

- Based on reactive ad-hoc network routing
- Attach service discovery headers to routing messages

**Service Request (SREQ)**

- SD Header
- RREQ
  - service id
  - destination address

**Service Reply (SREP)**

- SD Header
- RREP
  - service id
  - lifetime
  - destination address
Approach [kp02]

- Each node maintains service bindings:

<table>
<thead>
<tr>
<th>service-id</th>
<th>lifetime</th>
<th>provider</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>...</td>
<td>c</td>
</tr>
<tr>
<td>y</td>
<td>...</td>
<td>e</td>
</tr>
<tr>
<td>y</td>
<td>...</td>
<td>f</td>
</tr>
</tbody>
</table>

- SREQ flood
  - Destination in RREQ message is initially empty
- If node has binding for sought service
  - insert *provider* as *destination* and forward
- Nodes storing a route to destination will reply SREP
- Client receives SREP
  - Client knows provider and route to provider
Architecture

Application (Client, Store, AccessPoint, etc.)

Service Discovery Component

Routing Component

Link Layer

servRequest, servAvailable, servUnavailable

servReply

sendData

recvData

RREQ, RREP, SREQ, SREP

RERR, DATA, other
Proactive Elements?

- Be more efficient for frequent requests
- Periodic Announcements?
  - Caching enough with frequent requests
- Negative Announcements?
  - Good when providers become actively unavailable
Provider Unavailable

- Benefit grows with #requests
Announcement Message

Service Announcement (SANM)

- SD. Header
  - service id
  - lifetime
  - ...
  - destination address
  - originator address=BCAST

- RREP

  positive/negative flag

- Very similar to SREP
- Provider initiates SANM flood:
  - Service disc. header adds / removes service binding
  - RREP part creates route
- Nodes only forward if own decision is affected
Outline

- Introduction
- Design / Implementation
- Model
- Results
- Conclusion
Model: Applications

- **Clients**
  - Inter-request time ~ exponential
  - Mobile
- **Stores, institutions**
  - Constant service availability
- **Cabs, bike couriers**
  - Very volatile service availability
- **Internet access points**
Access Point Providers

- Clients are engaged in sessions
- Provider becomes unavailable if max. capacity is reached

\[
\text{session num} = \text{max}
\]

Avail.  \[\rightarrow\]  Unavail.

\[
\text{session num} < \text{threshold}
\]

- Becomes available if capacity free (threshold)
- Session length $\sim$ gamma($\alpha,1$)
- Traffic during session
Model: Mobility

- Variant of Random Waypoint Model
- Participants travel to random destination
  - Speed $\sim$ normal($\mu$, $\sigma$)
  - Different $\mu$, $\sigma$ for provider types
- Minimum speed
Model: Network

- Physical Layer
  - Simple path loss → radius

- Link Layer
  - Broadcast and unicast message delivery
  - Latency based on message length
  - MAC modelled by random delay
Outline

- Introduction
- Design / Implementation
- Model
- Results
- Conclusion
Simulation: Access Point Providers

- Study protocol behaviour with AP
- 800m x 800m area (see paper for parameters)
- AODV as routing component
- Prominent metrics:
  - Messages per request
  - Ratio of incorrect replies
- Runs:
  - Increase cache lifetime (under high load: 50 clients and 5 AP)
  - Increase number of requests (with a given lifetime)
Method

- Implementation in C++
  - Discrete Event Simulator (Omnet++)
  - TKN Mobility Framework
- Transient removal with Akaroa
- Parallel runs until reaching:
  - confidence level of 95%
  - 5% precision (or 0.05)
Varying Caching Lifetime

Messages per Request (Sent)

Incorrect Replies

caching
negative ann.
Varying Network Load

Messages per Request (Sent)

Incorrect Replies

- no caching
- caching (20s)
- negative anm.

Clients (equal to AP * 10)
Results

- Reactive without caching:
  - Most consistent
  - Least performant

- Reactive with caching:
  - Least consistent
  - Most performant

- Negative Announcements
  - Good balance
  - Better than customized lifetime
Conclusion

- Implemented and evaluated Internet Draft [kp02]
- Modular architecture for integration of routing and discovery
- Negative announcements perform well:
  - Make no difference for STORE providers
  - Provide consistency for CAB providers
  - Balance between performance and consistency for ACCESS POINT providers
- Explicitly removing cached entries is well-invested effort
Consistency Challenges of Service Discovery in Mobile Ad Hoc Networks

Christian Frank
ETH Zürich, Switzerland

Holger Karl
Universität Paderborn, Germany

Thank you!
Questions?

chfrank@inf.ethz.ch