

Mobile Computing

Exercise 3

Mobile Computing 2005

Übersicht

- Sehr kurz: Review Übung 2
- Multihop – warum?
- Dynamic Source Routing
- Multihop – wie?

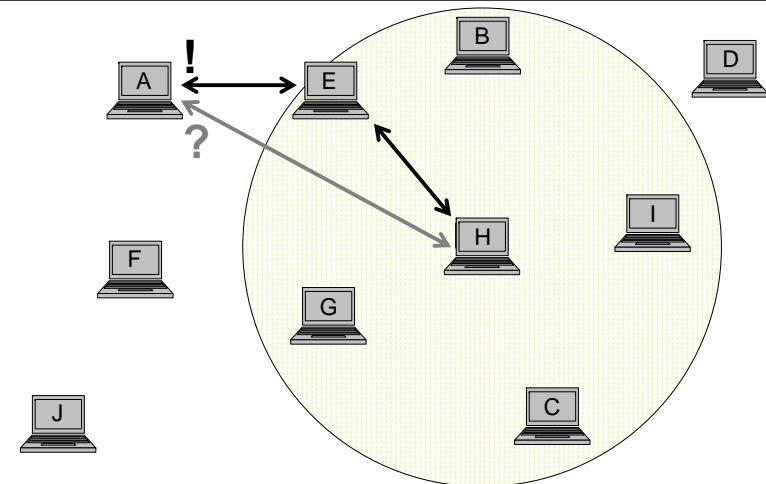
Mobile Computing 2005

Übung 2 - Review

- Sehr wenige Fragen 😊
- Musterlösung bald auf der Webseite

Mobile Computing 2005

Warum ein Multihop Layer?



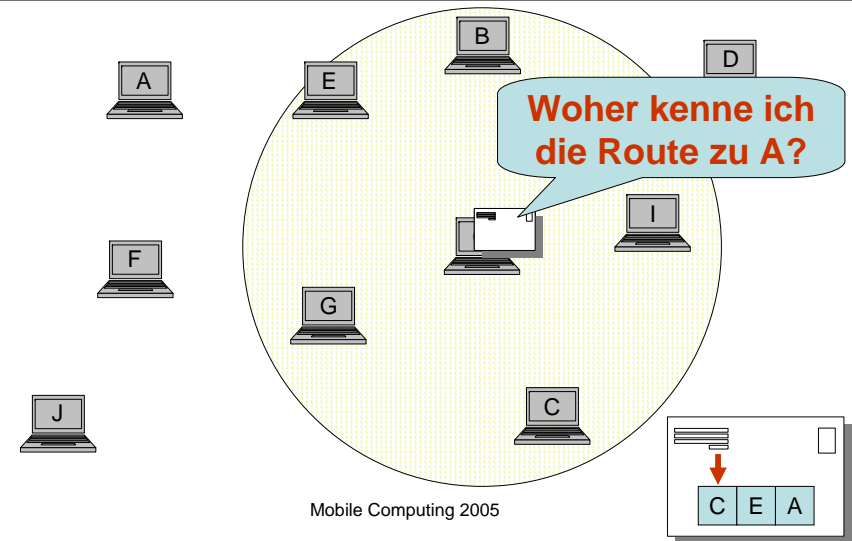
Mobile Computing 2005

Dynamic Source Routing (DSR)

- Route zum Ziel steht im Paket
- Zwischenknoten leiten das Paket weiter
- → „forwarding phase“

Mobile Computing 2005

Forwarding Phase



Mobile Computing 2005

Route Discovery - Request

- 2 Phasen: Flood / Reply
- Flooding
 - Sender schickt **Route Request**
 - Kontrolliertes Flooding → TTL
 - Knoten:
 - TTL vermindern
 - eigene ID anhängen
 - Request weiterleiten

Mobile Computing 2005

Route Discovery - Reply

- Ziel empfängt Route Request
- Antwort entlang der selben Route zurück zum Sender
- Tadaa! Sender kennt (schnellste) Route zum Ziel!

Mobile Computing 2005

Fazit

- „Demand driven“
 - Datenverkehr nur falls nötig
 - Null Verkehr bei Funkstille
- Völlig unabhängig von der Topologie
- Geeignet für Mobile Netze

- Aufwändig! → 2 x 2 Wochen Zeit

Mobile Computing 2005

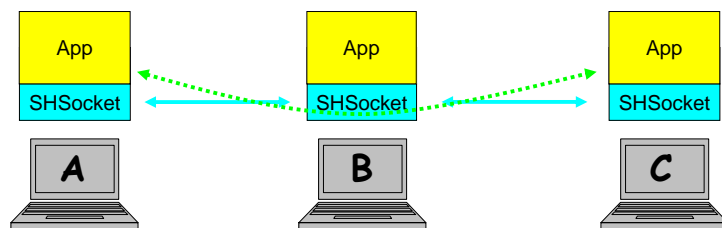
1. Teil: Route Request

- Einfach Flooding implementieren?
Ja, aber...

- ...nicht nur:
 - Dispatcher
 - TTL
 - Route / Route Index

Mobile Computing 2005

Multihop – Neue Schicht



Mobile Computing 2005

Multihop Layer

- Kann:
 - Route suchen
 - Routen speichern
 - MHPacket senden (wie bisher)
 - MHPacket empfangen und behandeln
 - neu **asynchron**

Mobile Computing 2005

Dispatcher (1)

- MHSocket wartet auf SHMessage Objekte aus der unteren Schicht
→ separater Thread

```
public class Noname extends Thread {  
    public void run() {  
        // work: read packet & call dispatcher  
    }  
}
```

```
Thread t = new Noname();  
t.start();
```

Mobile Computing 2005

Dispatcher (2)

- Verschiedene Paket-Typen müssen verschieden behandelt werden
→ je ein Handler pro Typ
- Handler registrieren sich beim Dispatcher
`dispatcher.addHandler(
 new RouteRequestHandler(foo, bar));`
- Dispatcher verteilt Pakete an die Handler

Mobile Computing 2005

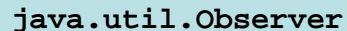
Dispatcher (3)

- Handler entscheiden, ob Paket interessant

- Nützliche Java Klassen:

– Dispatcher 

– Handler



- Observable:

– `addObserver(Observer obs);`
– `notifyObservers(Object param);`

Mobile Computing 2005

Mögliche Features

- Inkrementelles Flooding
- Ping/Pong Meldungen

Mobile Computing 2005

Fragen über Fragen?

- www.google.ch
- javadoc
- Mail oder Besuch im ETZ G63 oder G64.1