Middleware for Distributed Applications in Decentralized Wireless Networks

Wolfgang Golubski

Zwickau University of Applied Sciences
FB Physikalische Technik / Informatik
Dr.-Friedrichs-Ring 2a, 08056 Zwickau, Germany

golubski@fh-zwickau.de
Mobile Computing

- Devices equipped with wireless network interfaces
  - handhelds, personal digital assistants, mobile phones, laptops
- Main features
  - limited hardware resources
  - wireless communication
  - self-organization
  - spontaneous collaboration
Application fields

- **Tourist information**
  - Tourists in a city can collect a lot of interesting data about the city like sights or catering. When a tourist comes new to the city, he can use the experiences of other tourists by exchanging respectively downloading their information (i.e. experiences)

- **Conference situation**
  - participants would like to meet persons with similar interests

- **Entertainment**
  - Network gaming is a promising increasing business industry
The Goal

- Communication infrastructure
  - To find the other hosts
  - To find services in the network
  - To manage the connections
  - To manage the services

- As a case study: a game service
  - German card-game called Schwimmen (swimming)
The Middleware MaJo

- J2ME (Java 2 Micro Edition)
- iPAQ H3630 and H3850 handhelds equipped with WLAN
- Java virtual machine: J9 from IBM
- Java RockyRoad API (JRRRA)

- Support the application development
The Middleware MaJo

- Application Layer
- Host and Service Layer (MaJo)
- Network Layer (JRRA)

- must locally organize the complete network infrastructure
- must have self-organized features
Host Discovery

- Problem:
  - no administrative components
  - no information about participants in a central place
- Thus, the information about the complete network topology must be locally available for each participant
Host Discovery

- Initialization: find out which mobile devices are active in the environment
- The new mobile host (MH) sends a broadcast packet and informs the other hosts about its own network address
- The other hosts return their own addresses to the sender (MH)
- If all active hosts have received and answered the broadcast packet, the network topology is updated
Host Management

- New mobile host (MH) can leave
- Terminated connection
  - notify the other hosts
- Unexpected disruption
  - leaving must be noticed by hosts
  - sending a multicast packet (alive packet) in periodical intervals
Host Management

- implemented by UDP
- but UDP is very fragile
- if no alive packet of a MH was received in a specified time => MH has left the network or a UDP error has appeared
- the probability that the MH has definitively left the network increases the longer no alive packet arrives
- hence: alive packets are sent periodically
- after a predefined delay the network topology is up-to-date
Service Management

- permanently current view of all available services
- actual service list is stored locally
- a new user informs the other users about his services
- services of the other users must be collected
- service discovery is similar to the host discovery model
- each user can start, finish and delete a service
- the alive packets are extended by service information
The Swimming Service
The Swimming Service
Conclusion

- The development of the middleware was time-consuming
- The development time of the application was moderate
- The network performance up to 7 MHs are acceptable
- Future work: intelligent profile model based on user preferences