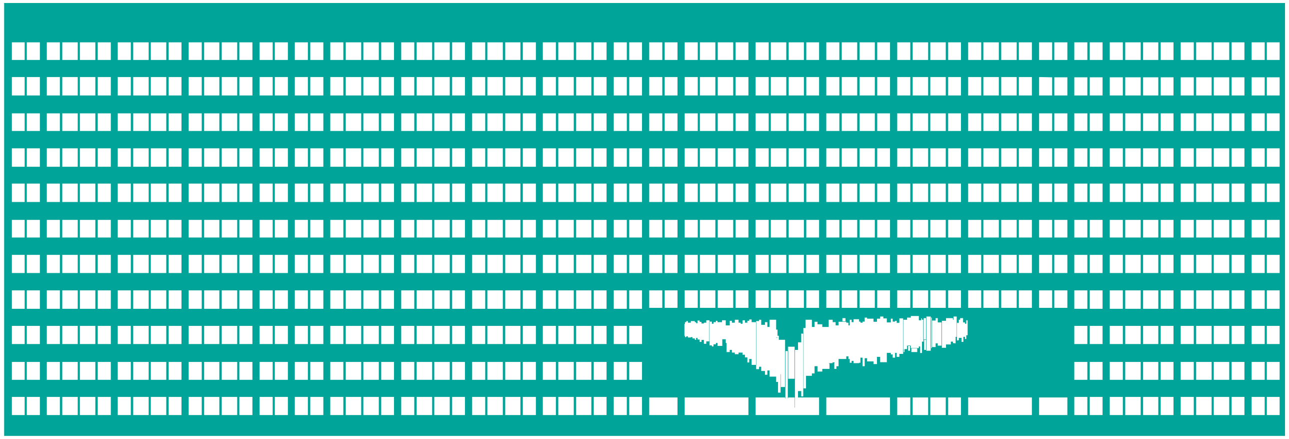
VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

Adapting applications for mobile use



MS (Mobile Computing)
Lecture 2

Mobility consequences

- The location of mobile elements and their access point to network changes during their movement
- The configuration of a system with mobile elements is not static:
  - the topology is not fixed unlike the original models
  - dynamic changes in the center of activity, locality and system load
    - usually necessary to locate objects through search
- Heterogeneous environment
- Variable offer of specific services, e.g. printing, local information

Location & connectivity

- Location
  - The search cost to locate mobile elements needs to be included in communication costs
  - Efficient data structures, algorithms, and query execution plans for representing, managing, and querying (fast changing) location of mobile elements

- Connectivity is highly variant as is the number of devices in a network cell at a time
  - bandwidth and reliability (obstacles, interference, …)
  - broadcast channel: base station → all mobile clients
  - charges (?): connection time, messages, kB (or mixed)

Mobile unit dilemma

- Mobile units are unreliable and more likely to suffer hard failures (theft, loss or accidental damage), with lower resources then static nodes → treat the mobile units as dumb terminals
  - run just a user-interface
  - offload all functionality from the mobile unit
- Slow and unreliable networks
  - put additional functionality into mobile unit
  - lessen its dependency on remote servers

Adaptability

- Varying resources and their quality:
  - totally disconnected → full connectivity
  - docked units may have access to additional resources
- Changes in:
  - location of mobile elements
  - network configuration
  - center of computational activity

Adaptivity: Need to adapt to the constantly changing environmental conditions

Implementing adaptivity

- Variable assignment of duties/tasks between the mobile and static elements, e.g.:
  - disconnected mobile host works autonomously
  - during strong connectivity, fixed network provides services and spares local resources

- Quality of data available at the mobile host may vary, *Fidelity* – degree to which a copy of data presented for use matches the reference copy, has many dimensions:
  - consistency (universal dimension)
  - frame rate, quality of frames, resolution, … (video)
  - sampling rate, lossless/lossy compression + degree, …

Mobility & adaptivity support placement

Strategies are developed between following two extremes:

A) Adaptivity is performed transparently by underlying system and applications are not aware of mobility

- ✓ no changes are necessary in existing applications
- ✗ no universal way to serve diverse applications → may be inadequate or even lower the performance
- ✗ may no be achievable, e.g. during long disconnections

B) Adaptation by individual applications, no OS support

- ✗ applications must be re-written → can be complicated
- ✗ no way to enforce limits on usage of resources
- ✗ incompatible resource requirements of different apps

Adaptation taxonomy

- System-transparent, application-transparent
  - conventional, “unaware” client/server model
- System-aware, application-transparent
  - client/proxy/server model
  - disconnected operation model
- System-transparent, application-aware
  - dynamic client/server model
  - the mobile agent (object) model
- System-aware, application-aware

Application-aware support for mobility

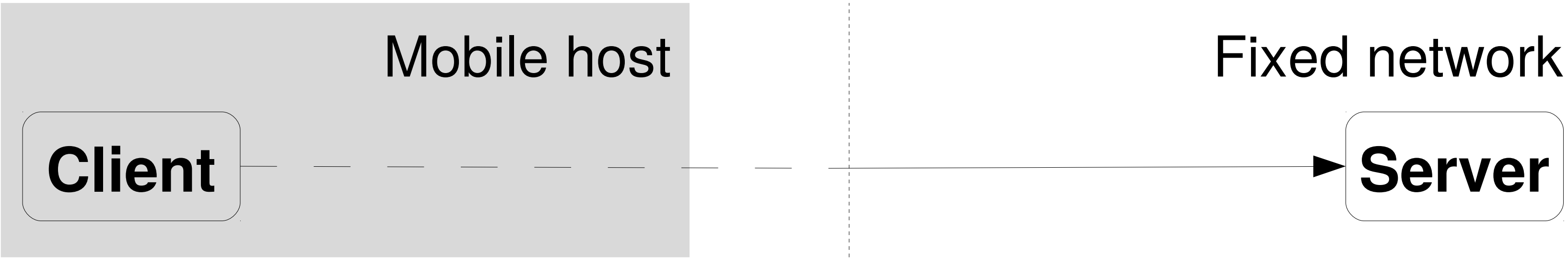
- Monitoring quality and level of resources, notifying applications about significant environmental changes: location, resources (bandwidth, memory, …)

- *Agile* applications, receiving events asynchronously and reacting to them accordingly. Detection & delivery of events may be separate (kernel × user level)

- *Central point* for resources management and authorization of application-initiated requests

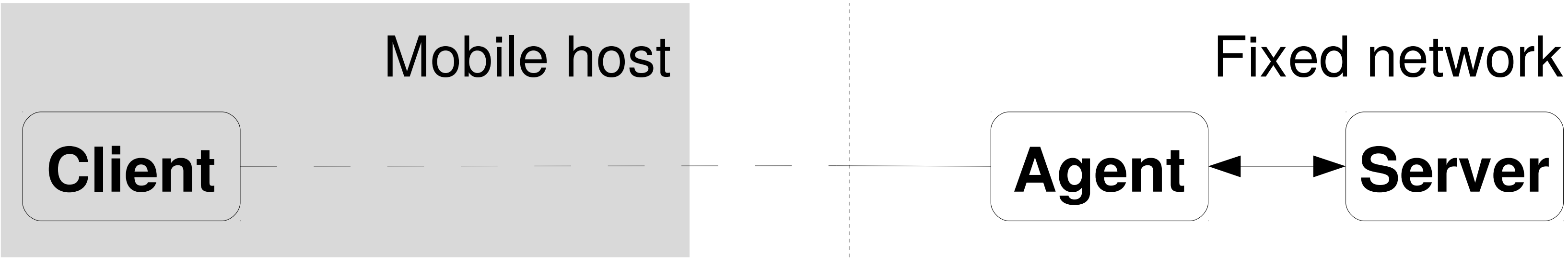
  Example: Application reserves a window of tolerance *for a particular resource. Application is notified if the resource leaves these boundaries.*

Client – Server



- Mobile host (*client*) requests services from *server* in FN
- If there are several interconnected servers, client can connect to the closest one – *service handoff*.
  - A) Mapping of clients to servers is transparent to applications, solved by server cache coherence control (cache invalidation, update propagation)
  - B) Selecting suitable server is application-based
- Suitable for: reliable & fast communication, stable connection, reliable static clients with rich resources

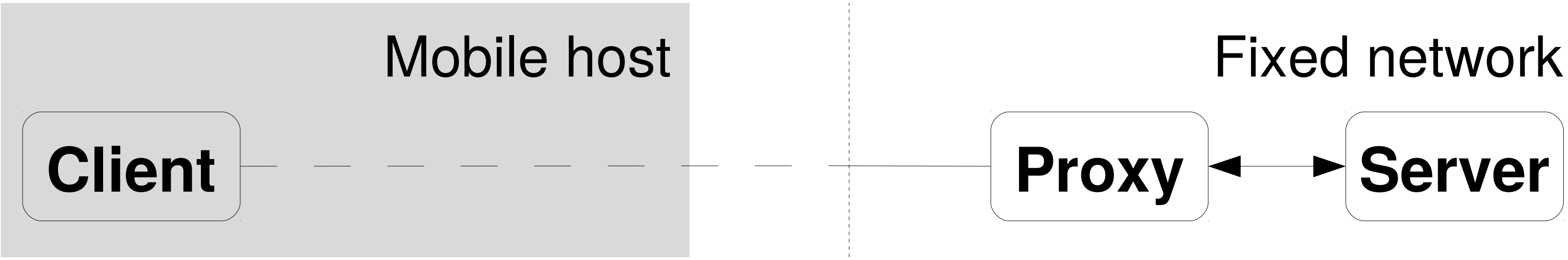
Client – Agent – Server



- Messaging & queuing infrastructure for communication between mobile client and agent

- Different forms & roles of agents: proxy × service-specific
  - Service-specific – mobile-enabled access to specific services or applications (web browsing, DB access,…). Client requests and server replies for given service go through the agent. Number of agents = number of services it needs to access.

Client – Proxy – Server



- Special case of Client – Agent – Server
- The agent is a complete surrogate (or proxy) of the mobile host in the fixed network.
- Any communication to and from the mobile host goes through the proxy. Application/user dependent settings.
- The agent can act as a proxy of multiple mobile hosts.
- One of service models in Mobile Ad-hoc networks.
- Allows thin client design for resource-poor mobile hosts

Agent functions

- Messaging and queuing
- Optimizations for weak connectivity
  - data manipulation before their transmission
    - changing transmission order so that the most important information is transfered first,
    - data specific lossy compression
  - tailored to the client × batching multiple replies
- More active role - notify the client on predefined events
- Offload functionality from client, start/stop specific functions at the mobile node, execute client-specific services.

Position of the agent
(depends on its role)

- On the fringe of the fixed network (at the base station)
  - personalized information available for the client
  - information about the wireless link is easier to obtain
  - special link-level protocol between the mobile host and the agent is available
    - ✗ agent may need to move along with its mobile host
    - ✗ current base station may not be trustworthy

- In the fixed network (usually service-specific agents) there are two possibilities:
  - closer to the majority of their clients × closer to the server

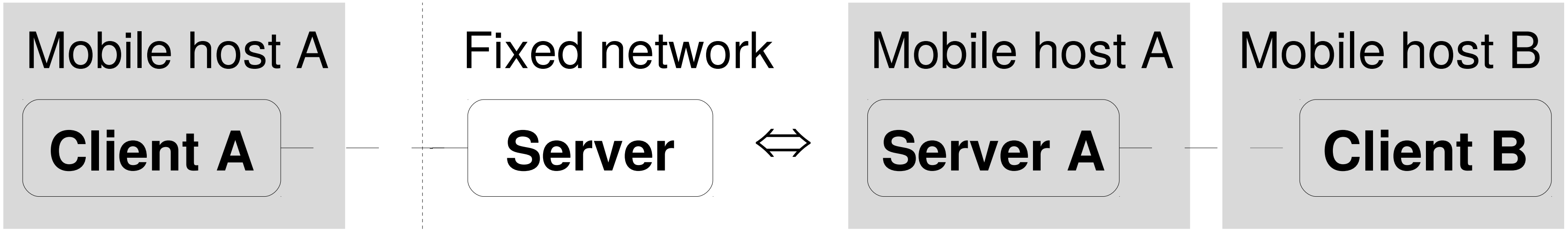
Client – Agent – Server closing remarks (1)

- Moving the agent in fixed network: prediction of the future location of a mobile user by mobile motion prediction algorithm, placing a new agent
- Strong points
  - reduces the impact of limited bandwidth and low reliability of the wireless link by maintaining the client's presence on the fixed network via an agent
  - splits interaction between mobile clients and fixed servers into two parts: client ↔ agent and agent ↔ server. Different protocols and independent execution for each part of the interaction are possible

Client – Agent – Server closing remarks (2)

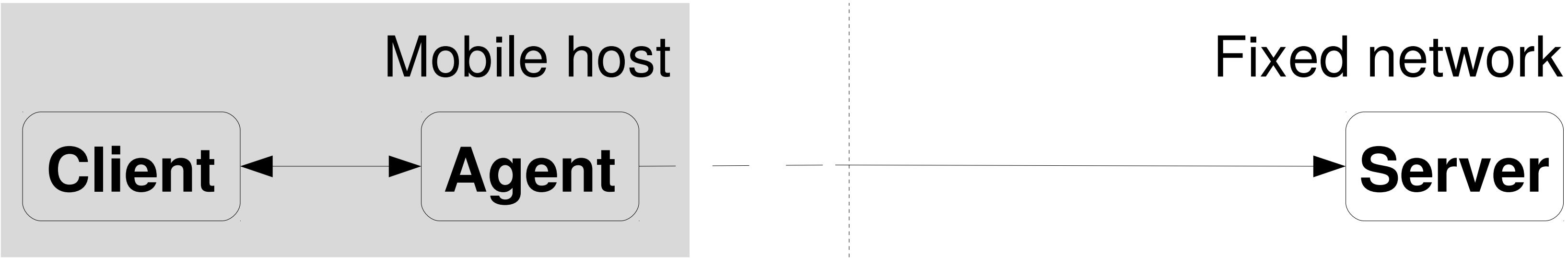
- Drawbacks
  - Fails to keep running the computation at a mobile client during disconnection periods.
  - Changes in the client code for the development of the client – agent interaction are necessary
  - The agent directly controls/optimizes data transmission from the fixed network to the mobile client over the wireless link and not in the opposite direction.

Dynamic Client – Server



- Servers (or thin versions of servers) dynamically relocate between mobile and fixed hosts.
- Proxies can be created and relocated dynamically.
- Dynamic availability/performance tuning
- Distributed architecture – replicating information among mobile hosts
- Error tolerant system, communication can be optimized

Local Agent



- A client-side agent to
  - reduce the amount of transmitted data, offer off-line data cache (obtained by hoarding – background prefetching) and reintegrate the data with real server after reconnection
  - emulate necessary server functions in disconnected state
  - improve availability and sustain the computation at the mobile host uninterrupted in special conditions
- From the point of view of the client: local server proxy residing on the local mobile host

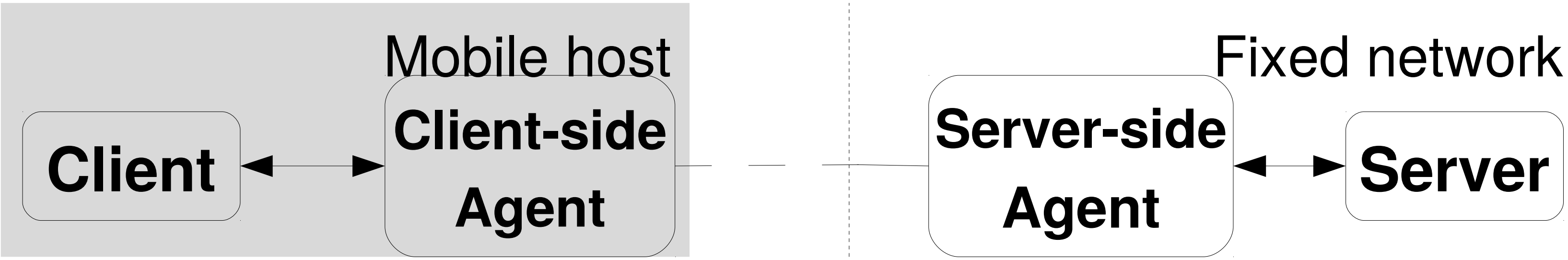
Disconnected operation

A) Provide full client and a thin version of the server on the mobile platform.

- Required data replicated into the mobile platform.
- On (re)connection, updated copies are synchronized with the server.
- Conflict resolution strategies are necessary

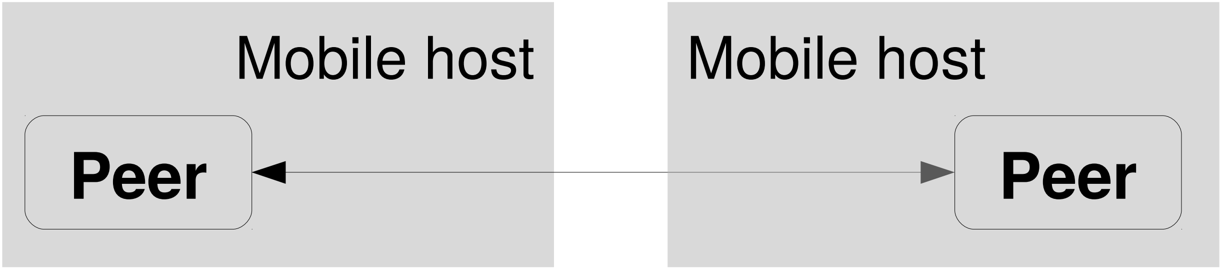
B) Provide a full client and a mobile agent that intercepts requests to the unreachable server, emulates the server buffering requests, and forwards them on (re)connection

Pair of agents between C&S



- Variety of optimizations available
- It is possible to relocate computation between the agents
- Clear separation of responsibilities exists
- Communication protocol can implement a highly effective data reduction and protocol optimization
- Drawbacks: relatively heavy-weight clients, changes necessary both on the server and the client side.
- A different pair of agents per application type is enough

Peer-to-peer model



- Mobile hosts act both as servers and clients and are equal partners in distributed computation.
- Tasks or work loads are partitioned between peers, which are equally privileged and potent participants in the application.
- Heavy-weight mobile hosts

Push/Pull data propagation

- *Pull data*: clients request (validate) data by sending uplink messages to the server.

- *Push data*: servers push data (and validation reports) usually through a broadcast channel, to a community of clients.