




Mobile Computing Model

 Files & media	04-MobileComputingModels.pdf
<input checked="" type="checkbox"/> Mid-Review	<input type="checkbox"/>
 Name	Lecture 4
<input checked="" type="checkbox"/> Review	<input type="checkbox"/>
 Status	Not started

Lecture 4 - Mobile Computing Models

Overview

- design appropriate model & archi to organize app components & specify relation

Mobile Computing Model

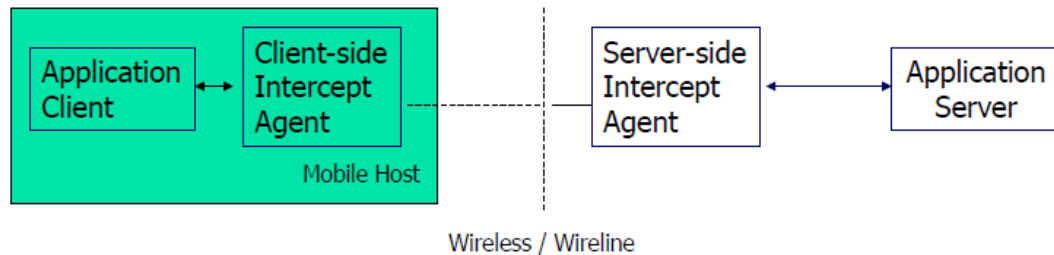
- client / server Model
 - ▼ client/server model
 - distributed system
 - client send request to server; server rich resource, respond request, result back to client
 - conn. b/t client & server; 2 client cannot comm.
 - ▼ mobile client / server model
 - mobile client request service from server located at fixed network
 - problems
 - no continuous network connectivity and powerful client ability
 - resource limitation on mobile client force client activity to move to server side(thin client)
 - client need to handle disconnection & low comm. bandwidth of network & provide persistent service on client side(smart client)

- mobility of client causes additional problem for server - when & where to transit reply back to client
- solution
 - extend c/s model to multiple tier / further optimization(data compressing/filtering/sync)
 - 3-tier approach
 - logic component(proxy) in middle tier
 - proxy can perform more processing operations &/ mask out mobile computing limitation

▼ Client / Agent / Server Model(CAS)

- agent proxy(deputy of client on fixed network)
 - comm. b/t client and server pass through agent
 - agent continuously maintain client's presence on fixed network
 - standard C/S interaction occurs b/t agent and server
 - diff. protocol can be used for interaction b/t mobile client-agent & agent-server
- Agent function
 - exchange msg & queue msg for comm. b/t mobile client-server
 - offload processing operation from client
 - handle disconnection b/t client-server
 - optimize transmission over wireless link
- ad:
 - client func. shift to agent - more appropriate for thin client app
 - complex client request managed by agent, final result to client
 - server shift some activity to agent(compression)
 - agent cache some result to improve performance
- dis: client side app need change to comm. to agent; no support disconnection operation for client

▼ Client / Intercept / Server Model(CIS)



- pair of agents - 1 stay on client side & other on fixed network
- agents transparent to both client & server
- 2 agents cooperate to facilitate effective data optimization & protocol translation
- client intercept agent function
 - pre-fetching & compression operation; cache data for satisfying clients' request during disconnection
- server intercept agent function
 - msging & queuing; offload processing client operation; handle disconnection b/t client & server; optimize transmission over wireless link
 - ad: more appropriate for clients with enough comp. power & storage; no change client&server -side app codes; diff protocol can be execute b/t client - client agent; server agent - server; optimize transm. masks disconnection
 - dis: client need more resource to run intercept agent; need develop intercept agent at both side; system overhead will increase more
- Peer-to-Peer Model
 - generalized C/S model
 - no center server; all nodes have dual role(resource-rich nodes needed); nodes can comm. directly; node op. completely decentralized & async.; server can move & suffer from disconnection
 - mobile P2P app
 - data sharing; instant msg; collaboration; entertainment
 - ad: good comp. model for ad hoc network; good archi for cooperative app; more flexible(server side can also move); good system scalability

- dis: both side need resource-rich device; protocol design complicated
- Mobile Agent Model
 - process on move
 - dispatched from 1 host into network; execute at host for a while, halts execution, dispatches itself to another host, resume execution there
 - Feature
 - mobility: mobile agent actively travel across network, carry partial patial / intermediate results calculated at the previous host; Not just alternative to msg passing; can perform additional tasks along its way of traveling
 - autonomy: Decides by itself what to do and how, when and where it ought to move
 - asynchrony: Mobile agents and users can execute at the same time without blocking each other
 - Application
 - long connection time / use bad connction/high mobility
 - can disconnect & do else while agent performing task
 - Advantage:
 - reduce remote comm; perform task locally(save bandwidth)
 - overcome msg passing latency
 - adaptive, intelligent
 - move around for best server
 - Problem
 - need more powerful mobile device
 - fault tolerance for autonomous operations of mobile agents
 - security issue

Mobile application architecture

▼ Thin Client

- internet app archi

- business logic & enterprise data on server
- micro browser on client(limited UI & capability)
- diff internet: how info transmitted to end user
- Components
 - Micro-browser Client: use URL address to contact specified wireless web server & parse response containing markup lang. & display result to user
 - Wireless Network: send request to & receive data from server
 - wireless Gateway: translate request & send it as HTTP to web server
 - convert diff. protocol; optimize comm. stream; push msg to wireless client; enhance security
- Server
 - web server: listen to HTTP request; corporate data source & format data appropriately for client; response back to the client
 - app server: provide app logics & core features
- Back-end System: provide enterprise data source; allow web & app server access enterprise data source using access mechanism
- Advantage: extend internet app to mobile env.; need minimal to 0 software deployment; keep fresh data; provide high data security
- Disadvantage: need persistent wireless connectivity; app performance is highly affected by network condition; availability; app testing difficult

▼ Smart Client

- allow user to access data even disconnected from network
- Components
 - Smart Client: run at client side
 - provide rich & customized UI to user
 - execute client-side app logic
 - persistent data storage locally to provide offline data access
 - comm to sync server via wireless conn
 - runs on mobile OS

- wireless network
- Application Server
 - interface - minimal data transferred, disconnected operation
 - execute server side application logic
 - integrate with back-end sys
 - perform data sync
 - b/t back-end sys & client side
 - ensure data consistency among copies
 - detect data conflicts
 - resolve data conflicts
 -
- back-end enterprise system
- Advantages: offline data access; performance; distributed computing; security
- Disadvantages: application deployment; development complexity; security risk