Cost Model and Adaptive Scheme for Publish/Subscribe Systems on Mobile Grid Environments

APGAC 2005, Atlanta May 24th 2005

Sangyoon Oh, Sangmi Lee Pallickara,
Sunghoon Ko,
Jai-Hoon Kim, Geoffrey C. Fox
Community Grids Laboratory and Computer Science
Indiana University,
School of Information and Communication
Ajou University

Contents

- Introduction
- Cost model
- Cost analysis
- Adaptive scheme
- Experiments
- Conclusion

http://grids.ucs.indiana.edu/ptliupages/hhms/pub-sub.html

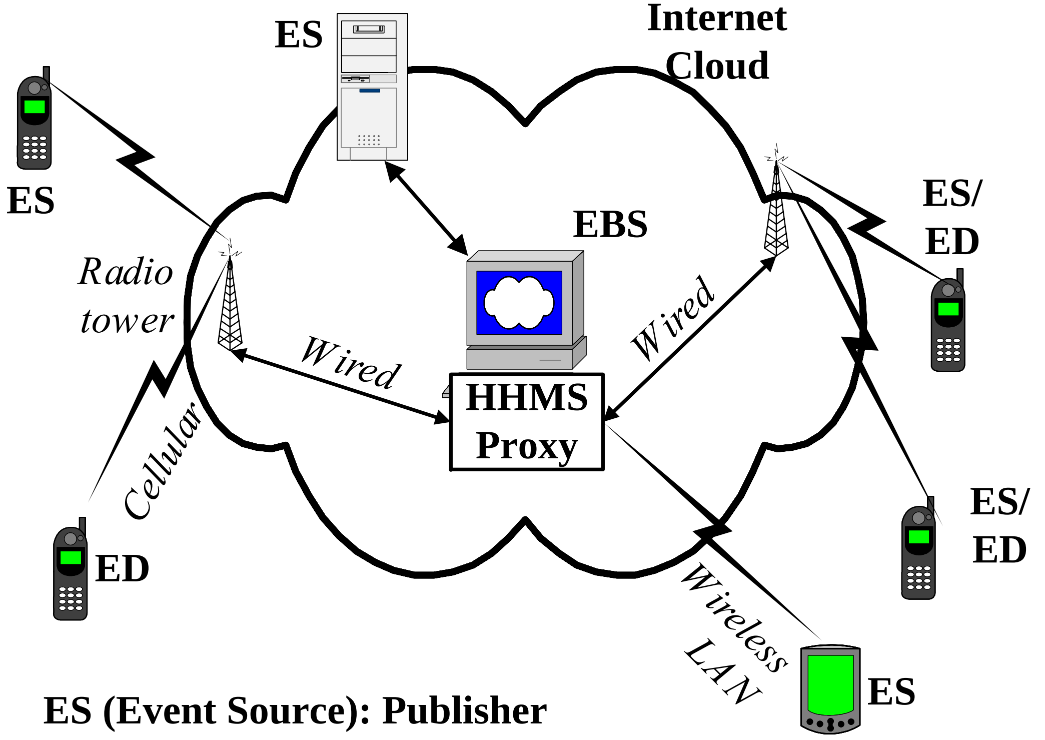
Introduction

Introduction

- Advantages of publish/subscribe systems in mobile computing
  - Intermittent and high latency wireless connections
  - Decoupling publisher and subscriber
    - Data dissemination services
    - Information sharing
    - Service discovery
  - Stock Ticker
- In this presentation:
  - Performance modeling
  - Cost analysis
  - Proposing an adaptive scheme

  → And its experiments

Pub/sub system configuration



ES (Event Source): Publisher
ED (Event Displayer): Subscriber
EBS (Event Brokering System): Server

Adaptive scheme

- Various types of mobile devices
  - Performance, Resources constrains, Application domain, Usage patterns
- Model selection

(publish/subscribe, request/reply)

1. *Static* model selection
2. *Hybrid* model selection

   Each device adopts appropriate model independently
3. *Dynamic* model selection

   Model can be changed during a service depending on change of status of system and network

Model Selection

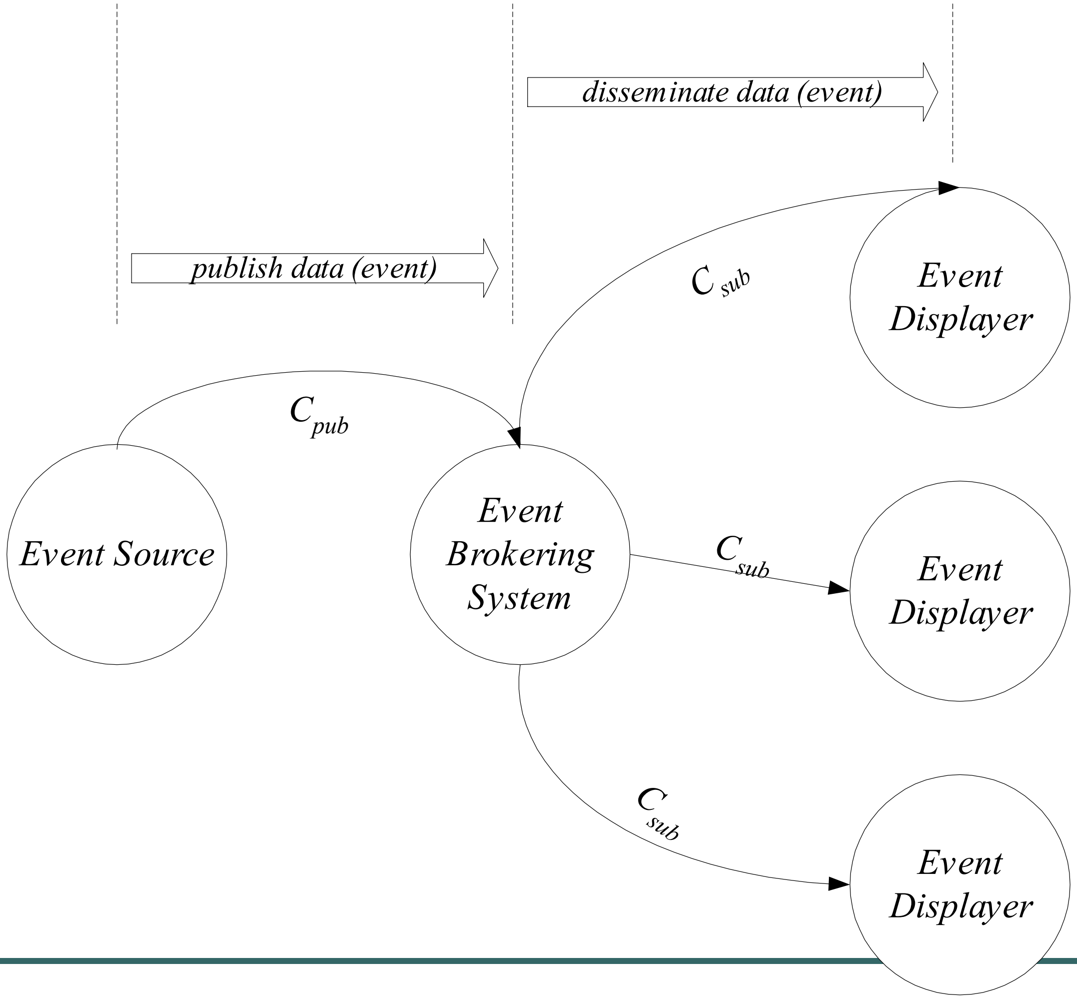
	models		*remarks*
	publish/ subscribe	*request/ reply*	
Number of node	*large*	*small*	*Pub/sub model has advantage when system is large and data transfer is shared among many clients*
Number of event (data update) *per client's access*	*small*	*large*	*Pub/sub model is appropriate when events or data update occurs infrequently.*
Access rate of client	*high*	*low*	*When clients seldom use published data, pub/sub model is not appropriate.*
Degree of common interest	*high*	*low*	*Pub/sub model is appropriate to disseminate data of common interest*
Cost of user's intervention	*high*	*low*	*Pub/sub model requires less user's intervention than request/reply model*
delay cost of event (data) transfer to user	*high*	*low*	*Events (data update) are immediately delivered to subscribers.*

Examples publish/subscribe system

- Broadcast notification services in many areas such as real-time sports news, stock market, etc. (publish/subscribe model)
- Many applications such as location based services are available using many types of devices and communication protocols (adaptive scheme: hybrid model selection).
- Users can alternatively choose on/off-line or power on/off to save communication cost or batter power, or during their movement (publish/subscribe model, adaptive scheme: dynamic model selection).
- Users can alternatively use wired or wireless connection (Ethernet or GPRS) during services (adaptive scheme: dynamic model selection)
- Programmer can choose model according to data access patterns and system parameters for designing application (adaptive scheme: static model selection)
- System manager can choose model according to service characteristics (adaptive scheme: static model selection)
- Users can choose model according to their preferences (adaptive scheme: hybrid model selection)
- System can automatically choose model for each user according to his/her reference or use pattern (adaptive scheme: hybrid model selection)

Cost Model

System model



System parameters

α (publish rate)
β (request rate or process
(reference access) rate)
$c_{ps}(\alpha)$ (publish/subscribe cost per event)
$c_{rr}(\beta)$ (cost per request and reply)
$c_{poll}(\alpha,T)$ (cost of periodic publish or polling)
$c_{d}(\alpha,T)$ (cost of delaying publish)
$s(n)$ (effect of sharing among n subscribers)
t_{ps} (time delay for publish/subscribe)
t_{rr} (time delay for request and reply)

Assumption and consideration

- Assumption:

  "***No communication link or node failure***"
- Consideration
  - Conceptual total cost per unit time
  - Cost for each access by client (or subscriber)
  - Time delay for access after subscriber's intension
  - Time delay between event occurrence and notification to subscriber (or recognition by client)
  - *Cost can be number of message, amount of message or time delay*

Cost Analysis

Cost of publish/subscribe model

- **Conceptual total cost per time unit**
  - *Cost of each publish/subscribe event:* $(c_{pub} + n\, c_{sub})$
  - *Sharing effect among n nodes:* $s(n)$
  - *Publish rate:* α
  - *Total cost per time unit* = α $(c_{pub} + n\, s(n) c_{sub})$

- **Performance metrics**

1. *Conceptual cost for each access:*
   - *aver. number of event before each access* $= \sum_{i=0}^{\infty} \frac{\beta}{\alpha+\beta}\left(\frac{\alpha}{\alpha+\beta}\right)^{i} = \frac{\alpha}{\beta}$
   - c_{pub} *is shared among n subscriber and* c_{sub} *is for each subscriber*
   - *aver. cost for each access* $= \alpha \;/ \beta\, (\, c_{pub} \;/ n + c_{sub})$
2. *Time delay between intention and access*

   ← *No time delay, since the event is already received*
3. *Time delay for occurrence and notification/recognition (or access)*

$$t_{ps} = t_{pub} + t_{sub}$$

Cost of request/reply model

- **Conceptual total cost per time unit**
  - *Cost for each request/reply is assumed as c_{rr}*
  - *Request rate: β , number of client: n*
  - *Total cost per time unit = β n c_{rr}.*

- **Performance metrics**
  1. *Cost for each access* = c_{rr}
  2. *Time delay between intention and access = t_{rr}* **(assumed)**
  3. *Time delay for occurrence and recognition = 1 / (2 β)*

     (depends on request rate)

Periodic (polling) model

- **Appropriate for applications where delayed message is acceptable**
- **Conceptual total cost per time unit**
  - *Cost for each polling* $= c_{poll}(\alpha, T) + c_{delay}(\alpha, T)$
  - *Total cost per time unit* $= (c_{poll}(\alpha, T) + c_{delay}(\alpha, T)) / T$

    where $c_{rr} < c_{poll}(\alpha, T) < \alpha T c_{rr}$

- **Performance metrics**

1. *Cost for each access* $= (c_{pub}(\alpha, T) + n\ c_{sub}(\alpha, T)) + c_{delay}(\alpha, T)) / T$

   where $c_{pub} < c_{pub}(\alpha, T) < \alpha T c_{pub}$ *and* $c_{sub} < c_{sub}(\alpha, T) < \alpha T c_{sub}$
2. *Aver. time delay between intention and access* $= T/2$
3. *Aver. time delay for occurrence and recognition* $= T/2$

Summary of Cost Analysis

Model	*Publish/Subscribe*	*Request/Reply*	*Polling*
conceptual total cost per time unit	$\alpha\ (c_{pub} + n\ s(n) c_{sub})$	$\beta\ n\ c_{rr}.$	$(c_{poll}(\alpha,T) + c_{delay}(\alpha,T)) / T$
cost for each access	$\alpha\ / \beta\ (\ c_{pub}\ / n + c_{sub})$	c_{rr}	$c_{poll}(\alpha,T) + c_{delay}(\alpha,T)$
time delay between intention and access	0	t_{rr}	$T/2$
time delay between event occurrence and notification/recognition (or access)	$t_{ps} = t_{pub} + t_{sub}$ $(t_{ps} = t_{pub} + t_{sub} + 1 / \beta)$	$1 / 2\ \beta$	$T/2$

Adaptive scheme

Adaptive scheme I

- Choosing appropriate model among publish/subscribe and request/reply models
  - Hybrid model: each client can select its own model independently
  - Dynamic model: change model during its service
- Considering cost for each client's access a cost metric

  Aver. Number of event occur

  - Aver. number of event per client's access

$$= \alpha \ / \beta \ (c_{pub} \ / n + c_{sub}) \ and \ c_{rr}$$

  - Aver. number of event and number of subscriber are obtained experimentally

Number of subscriber

Adaptive scheme II

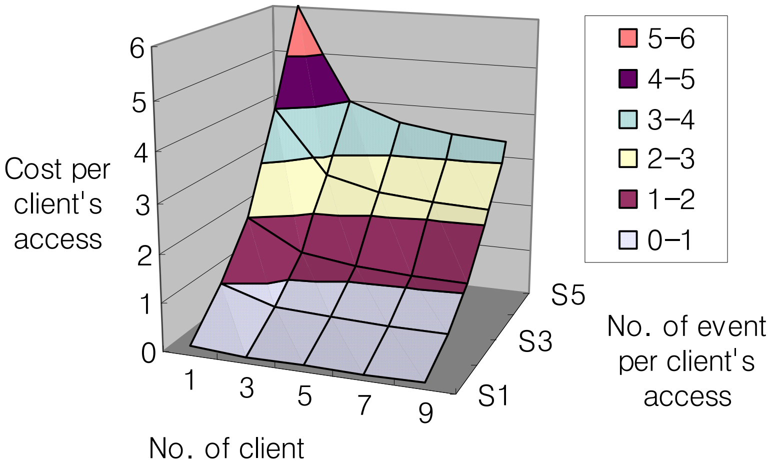
- Number of event per client's access
  - Request/reply: increment of counter for each client's access
  - Publish/subscribe: using event ID and client ID
- Steps
  1. During the period of time, average number of event occurred per client's access is measured for each client.
  2. If $\alpha / \beta (c_{pub} / n + c_{sub}) > c_{rr}$, choose request/reply model for the next period.
  3. else, choose publish/subscribe model.
  4. Repeat step1 and step3

Illustration of adaptive scheme

- Assumption
  - $c_{rr} = 2$
  - $c_{pub} = 1 \text{ and } c_{sub} = 1$
- Conclusion
  - number of client is large
  - number of event per client access is small

  → Use publish/subscribe model,

  if cost per access <= 2

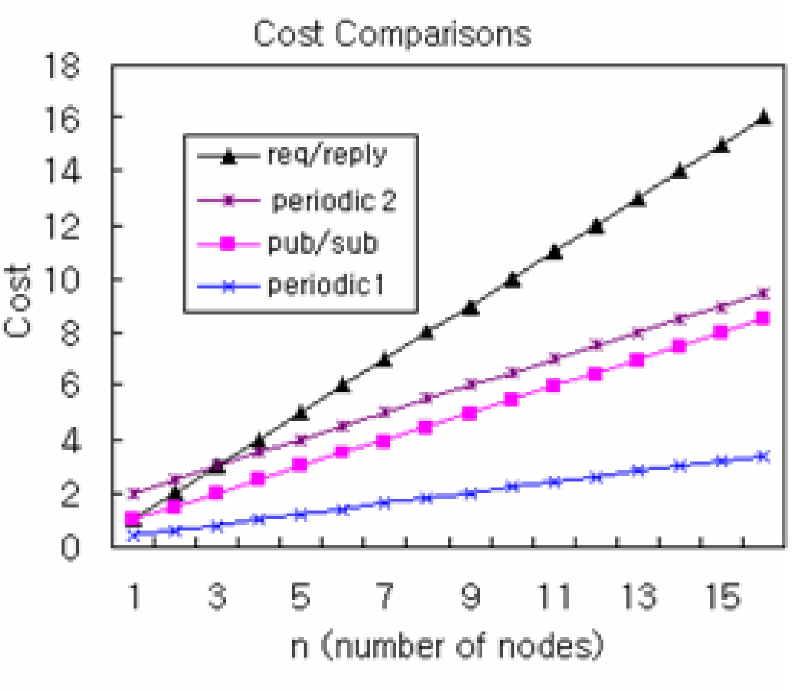


Cost per client’s access of publish/subscribe model

Performance Comparisons

Parametric Analysis of Performance comparison

Param.	Values
$a,$	0.5
β	0.5
c_{ps}	2
c_{pub}	1
c_{sub}	1
c_{rr}	2
$c_{poll}(a, T)$	1 or aT
$c_{delay}(a, T)$	$0, T,$ or aT
$s(n)$	1
t_{ps}	1
t_{proc}	1 or 5
t_{rr}	1
$t_{poll}(a, T)$	$1, T,$ or aT



Communication cost per transaction by varying number of clients

$c_{pub}(\alpha,T)= c_{pub},\ c_{sub}(\alpha,T)= c_{sub},$ *and* $c_{delay}(\alpha,T)= 0$ ← *For periodic 1*

$c_{pub(\alpha,T)}= \alpha Tc_{pub},\ c_{sub}(\alpha,T)= \alpha Tc_{sub},\ c_{delay}(\alpha,T)= 2\alpha Tc_{delay}$ ← *For periodic2*

Experimental Setup

- Using NaradaBrokering as message brokering system -- MOM (Message Oriented Middleware) **for publish/subscribe**
- Using HHMS (Handheld Message Service) as primary application level transport protocol **for publish/subscribe** between mobile device and conventional wired environment
- Conventional RPC code using J2SE and J2ME MIDP 2.0 **for request/reply**
- Benchmarking Applications

  Measures Round Trip Time (RTT) → RTT/2 for pub/sub and RTT for request/reply
  - Echo clients for cost (time) per message by varying size of message
  - ACK clients for message publishing cost for various number of clients
- Experimental Specifications
  - Treo600:

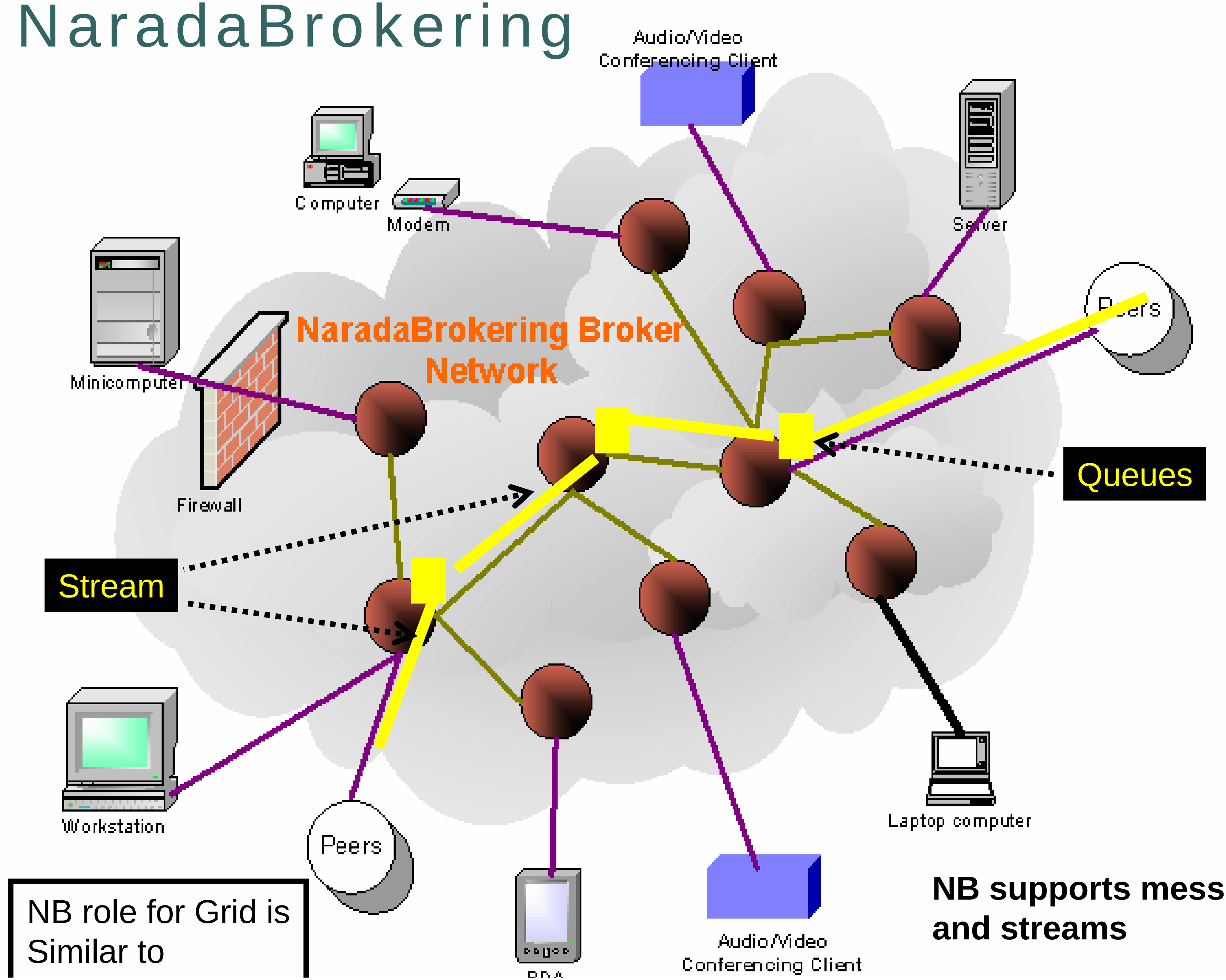
    PalmOS 5.2 144MHz ARM, 32MB, Sprint PCS Service (<14.4kbps)
  - HHMS Gateway and NaradaBrokering:

    Linux 7.3, Pentium III 1GHz, 512MB
  - Timer: Linux native timer by JNI

NaradaBrokering

- Developed by Community Grids Laboratory of Indiana University
- Message Oriented Middleware (MOM)
  - Multiple protocol transport support: In publish-subscribe Paradigm with different Protocols on each link
  - Subscription Formats
  - Reliable delivery
  - Ordered delivery
  - Recovery and Replay
  - Security
  - Message Payload options
  - Messaging Related Compliance
  - Grid Feature Support
  - Web Services supported

NaradaBrokering

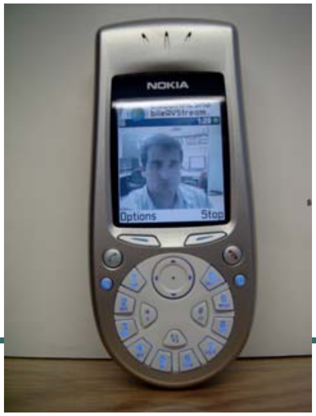

Audio/Video Conferencing Client
Computer
Modem
Server
Minicomputer
Firewall
NaradaBrokering Broker Network
Peers
Queues
Stream
Workstation
Peers
PDA
Audio/Video Conferencing Client
Laptop computer

NB role for Grid is Similar to

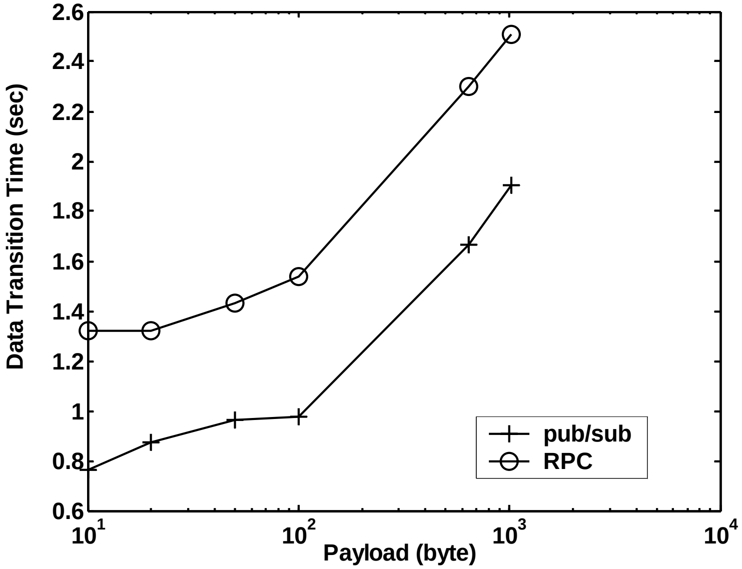
NB supports messages and streams

Handheld Messaging Service

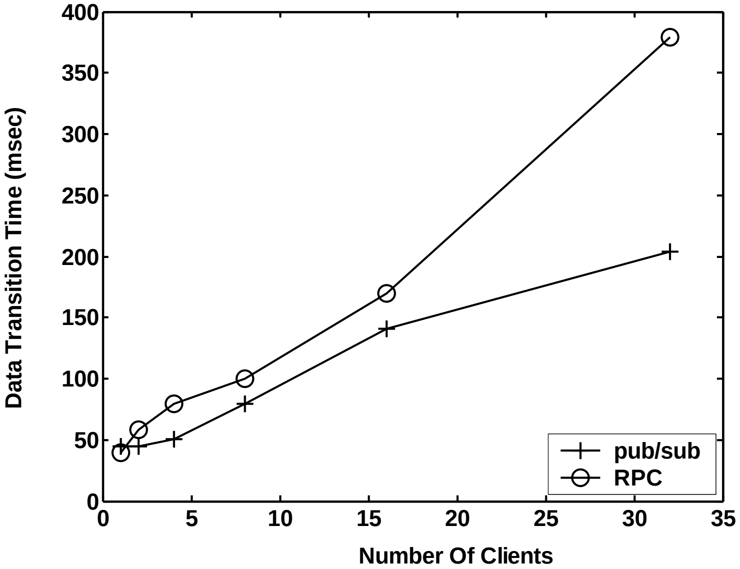
- Light-weight publish/subscribe message service framework for mobile devices
- Optimized application level transport protocol using byte message format
- Provide core-subset of JMS API



Experimental Results



Delay time by Payload *(Treo 600 Smart phone Over Sprint PCS cellular connection)*



Delay Time by Number of Clients *(J2ME Wireless Toolkit Emulator over 802.11b WLAN)*

Conclusion

Conclusion

- We presents cost model and cost analysis for publish/subscribe system, request/reply, and polling model
- Our proposed adaptive scheme improves the communication performance based on the dynamic parameters for individual mobile clients
- Experiments show a matching result with the theoretical analysis.