Distributed Computing and Systems
Chalmers university of technology

Group Communication

Phuong Hoai Ha & Yi Zhang

Introduction to Lab. assignments
March 24th, 2004

Today's schedule

- Introduction to group communication
- Desired group communication
- Multicast communication
- Group membership service

Coordination in distributed systems

- Coordination is needed by distributed systems but hard to achieve:
  - Events happen concurrently
  - Communication links are not reliable
  - Computers can crash
  - New nodes can join the systems
  - Asynchronous environments

⇒ respect an efficient way to coordinate a group of processes

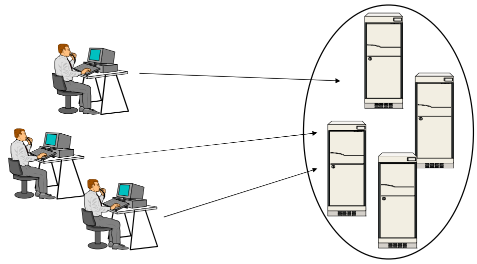
Group communication

- What is a group?
  - A number of processes which cooperate to provide a service.
  - An abstract identity to name a collection of processes.
- Group Communication
  - For coordination among processes of a group.

Who Needs Group Communication?

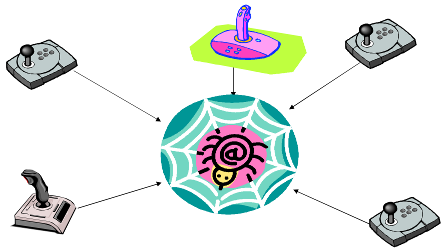
- Highly available servers (client-server)
- Database Replication
- Multimedia Conferencing
- Online Game
- Cluster management
- …

Distributed Web Server



- High availability

Online Game



- Fault-tolerance, Order

Different Comm. Methods

- Unicast
  - Point-to-Point Communication
  - Multiple copies are sent.
- Broadcast
  - One-to-All Communication
  - Abuse of Network Bandwidth
- **Multicast**
  - One-to-multiple Communication

Today’s schedule

- Introduction to group communication
- Desired group communication
- Multicast communication
- Group membership service

Group Comm. Properties

- Name Abstraction
- Efficiency ⇒ Multicast
- Delivery Guarantee
  - Ordering
  - Failure behavior
  - Reliability
  - …
- Dynamic Membership ⇒ Group membership service

Properties of Communication

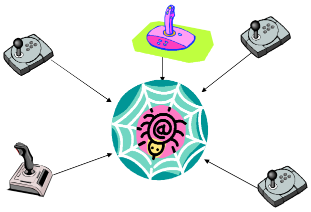
- Ordering
  - Total order, causal order
- Failure behavior
  - Failure atomicity
- Reliability
  - Validity, integrity, agreement

Properties of Group

- Name of group
- Addresses of group members
- Dynamic group membership
- Options:
  - Peer group or client-server group
  - Closed or Open Group

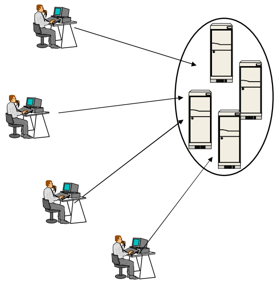
Peer Group

- All the members are equal.
- All the members send messages to the group.
- All the members receive all the messages.



Client-Server Group

- Replicated servers.
- Clients do not care which server answers.



Desired Group Communication

- Name Abstraction
- Efficiency ⇒ Multicast
- Delivery Guarantees ⇒ Reliability, Ordering
- Dynamic Membership ⇒ Group membership service

Today’s schedule

- Introduction to group communication
- Desired group communication
- Multicast communication
- Group membership service

Multicast communication

- Send message over a distribution tree.
- Use network hardware support for broadcast or multicast when it is available.
- Minimize the time and bandwidth utilization

Reliability

Correct processes: those that never fail.

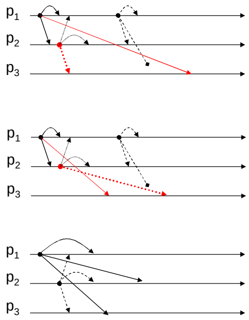
- Integrity
  A *correct* process *delivers* a message at most once.
- Validity
  A message from a *correct* process will be *delivered* by the process eventually.
- Agreement
  A message *delivered* by a *correct* process will be *delivered* by all other *correct* processes in the group.

⇒ Validity + Agreement = Liveness

Ordering

Assumptions: a process belongs to at most one group.

- FIFO
  - if $m_p \rightarrow m'_p$, all correct processes *that deliver* m'_p will deliver m_p before m'_p.
- Causal
  - if $m \rightarrow m'$, all correct processes *that deliver m'* will deliver *m* before *m'*.
- Total
  - if a correct process delivers *m* before *m'*, all other correct processes *that deliver m'* will deliver *m* before *m'*.



Examples

- Assumption:
  - Reliable one-to-one *send* operation (e.g. TCP)
- Basic multicast
  - Requirement:
    - All correct processes will eventually deliver the message from the *correct* multicaster.
  - Implementation:
    - ***B-multicast(g, m)****: ∀p ∈ g: send(p, m);*
    - *On receive(m) at p:* ***B-deliver(m)*** *at p.*
    - ⇒ Properties: integrity, validity.

Examples (cont.)

- Reliable multicast
  - Requirements: integrity, validity, *agreement*
  - Implementation:

```
• Received := {};
• R-multicast( g, m) at process p: B-multicast( g, m);
• On B-deliver( m) at process q
  if( m ∉ Received)
          Received := Received ∪ {m};
          if( q ≠ p) B-multicast( g, m);
          R-deliver( m);
  end if
```

⇒ Inefficient: each message is sent |g| times to each process

  - Encourage to implement in more efficient ways (e.g. IP-multicast, etc.)

Examples (cont.)

- FIFO-ordered multicast:
  - Assumption:
    - a process belongs to at most one group.
  - Implementation:

```
• Local variables at p: S_p = 1, R_p[ |g| ]={0};
• FO-multicast( g, m) at p:
          B-multicast( g, <m, S_p>);
          S_p++;
• On B-deliver( <m, S>) from q:
  if( S = R_p[q] + 1)
          FO-deliver( m);
          R_p[q] := S;
  else if( S > R_p[q] + 1)
          place <m, S> in the queue until S = R_p[q] + 1;
          FO-deliver( m);
          R_p[q] := S;
  end if
```

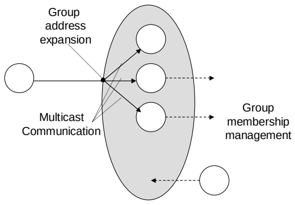
  - Encourage to implement causally ordered, totally ordered multicasts.

Today’s schedule

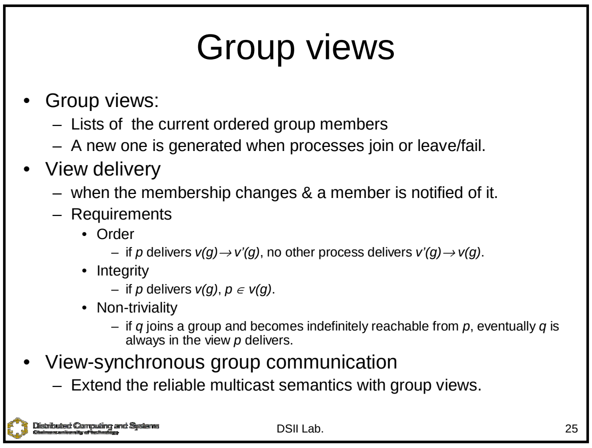
- Introduction to group communication
- Desired group communication
- Multicast communication
- Group membership service

Group membership service

- Four tasks:
  - Interface for group membership changes
  - Failure detector
  - Group address expansion
  - Membership change notification

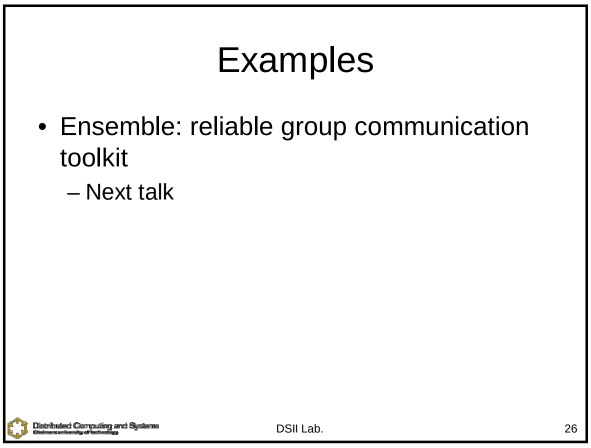


- Group partition:
  - Primary-partition
  - Partitionable



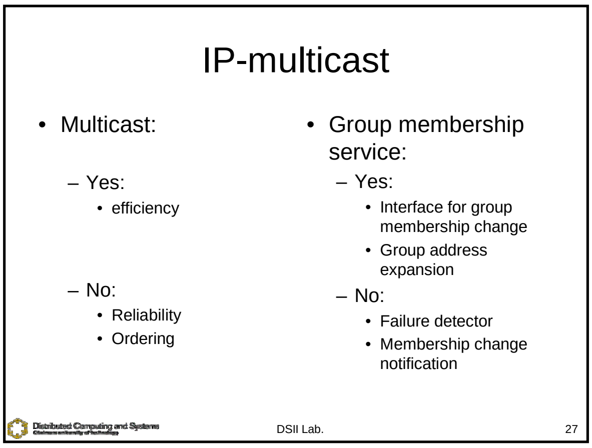
Group views

- Group views:
  - Lists of the current ordered group members
  - A new one is generated when processes join or leave/fail.
- View delivery
  - when the membership changes & a member is notified of it.
  - Requirements
    - Order
      - if *p* delivers $v(g) \rightarrow v'(g)$, no other process delivers $v'(g) \rightarrow v(g)$.
    - Integrity
      - if *p* delivers $v(g)$, $p \in v(g)$.
    - Non-triviality
      - if *q* joins a group and becomes indefinitely reachable from *p*, eventually *q* is always in the view *p* delivers.
- View-synchronous group communication
  - Extend the reliable multicast semantics with group views.



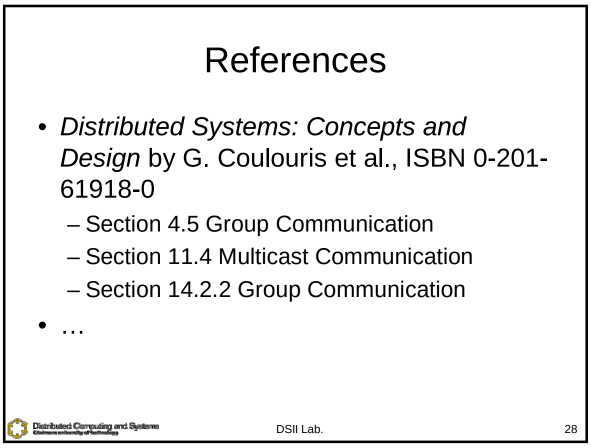
Examples

- Ensemble: reliable group communication toolkit
  - Next talk



IP-multicast

- Multicast:
  - Yes:
    - efficiency
  - No:
    - Reliability
    - Ordering
- Group membership service:
  - Yes:
    - Interface for group membership change
    - Group address expansion
  - No:
    - Failure detector
    - Membership change notification



References

- *Distributed Systems: Concepts and Design* by G. Coulouris et al., ISBN 0-201-61918-0
  - Section 4.5 Group Communication
  - Section 11.4 Multicast Communication
  - Section 14.2.2 Group Communication
- …