ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
UNIVERSITY OF CRETE

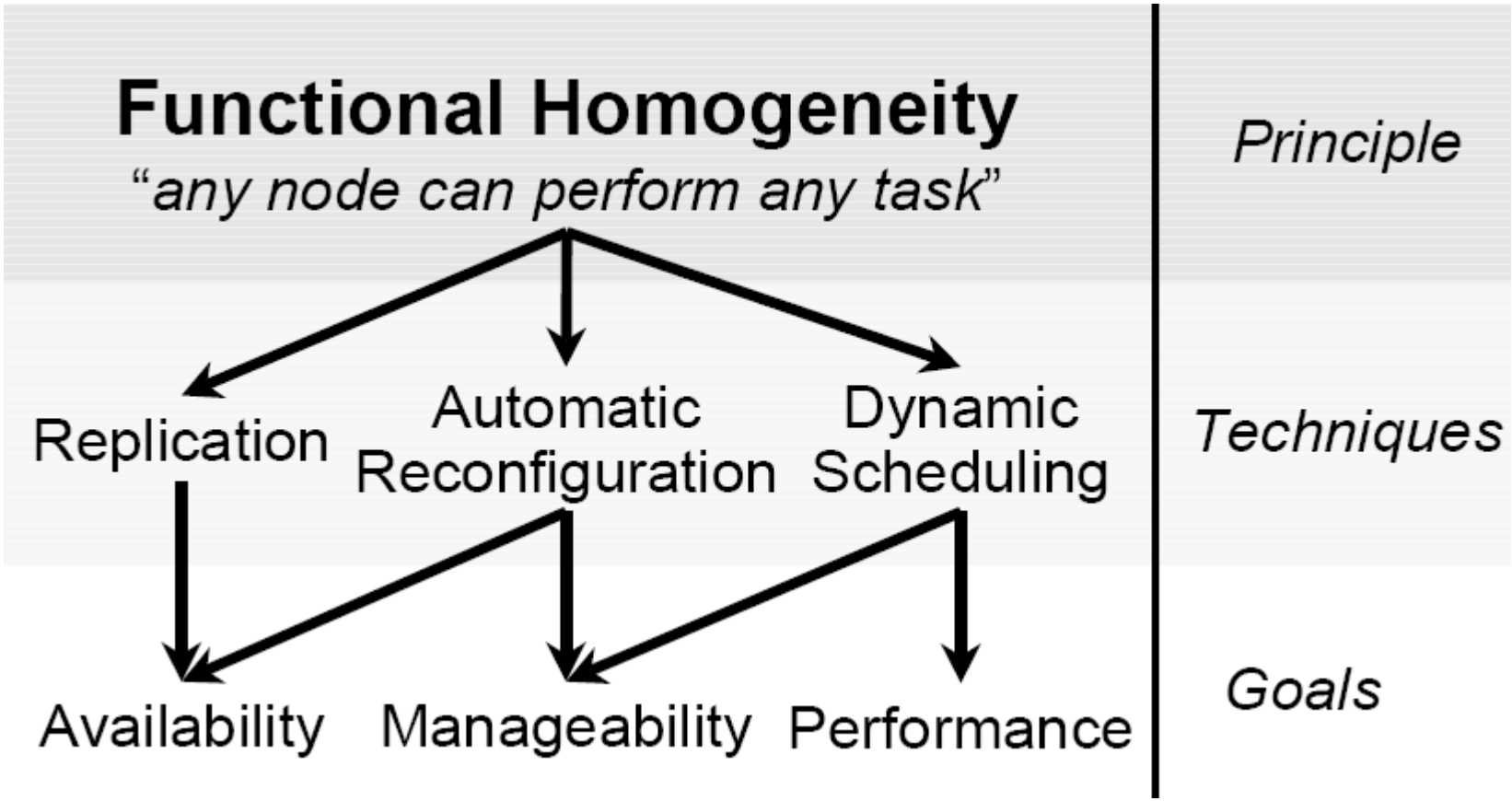
HY590.45
Modern Topics in Scalable Storage Systems

Kostas Magoutis
magoutis@csd.uoc.gr
http://www.csd.uoc.gr/~hy590-45

Scalability requirements

- Manageability
  - Ability to self-configure, self-heal
  - System manager just adds or replaces machines
- Availability
  - Despite component failures, deliver good service to all users
  - Some users may be prevented from accessing some mail
- Performance
  - Single-node performance as good as single-node systems
  - Aggregate performance scale linearly with number of nodes

Key principle and techniques behind Porcupine



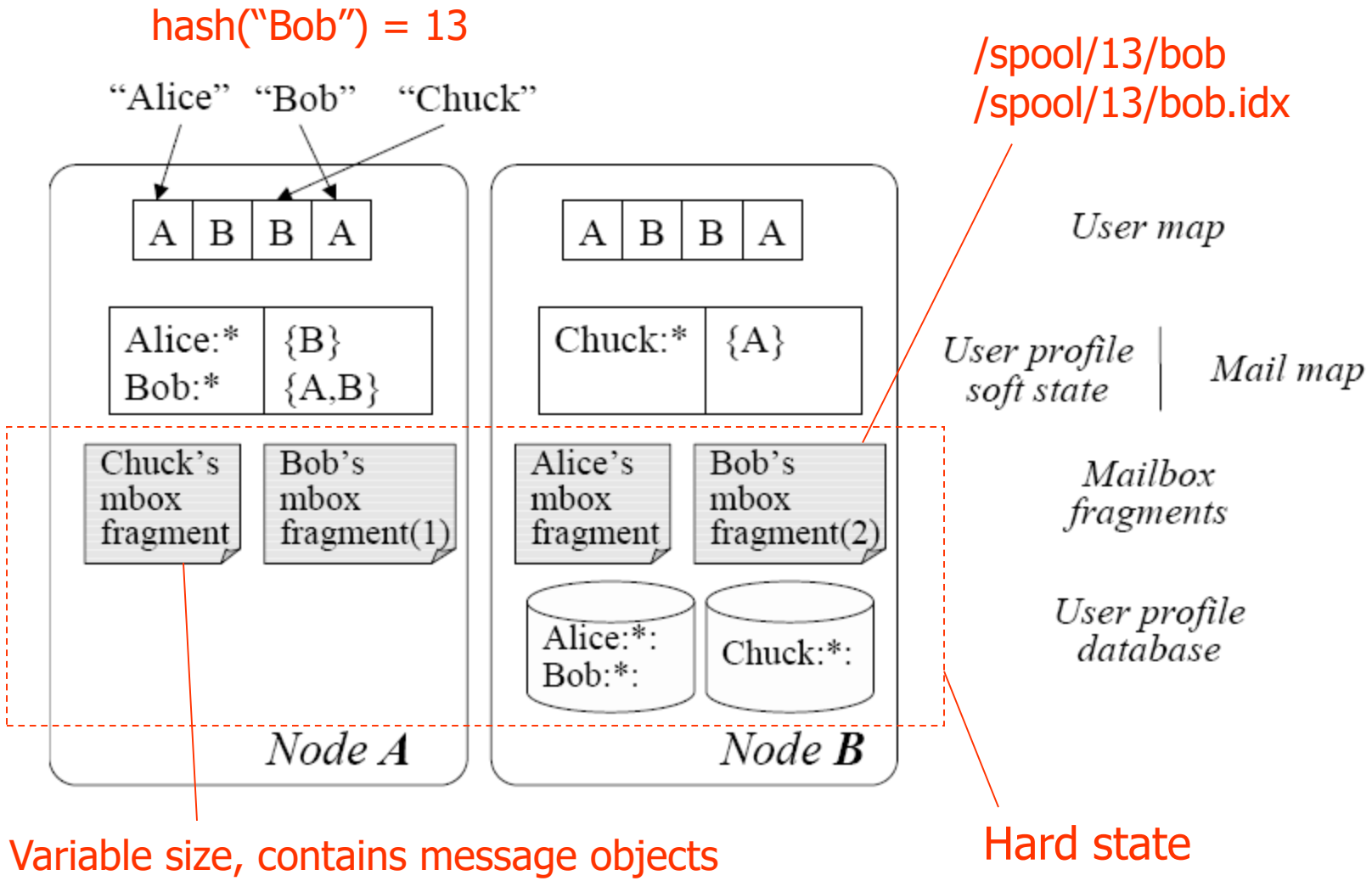
Why E-Mail

- Need systems that can scale to billions messages/day
- Frequent writes present a challenge
- Weak consistency

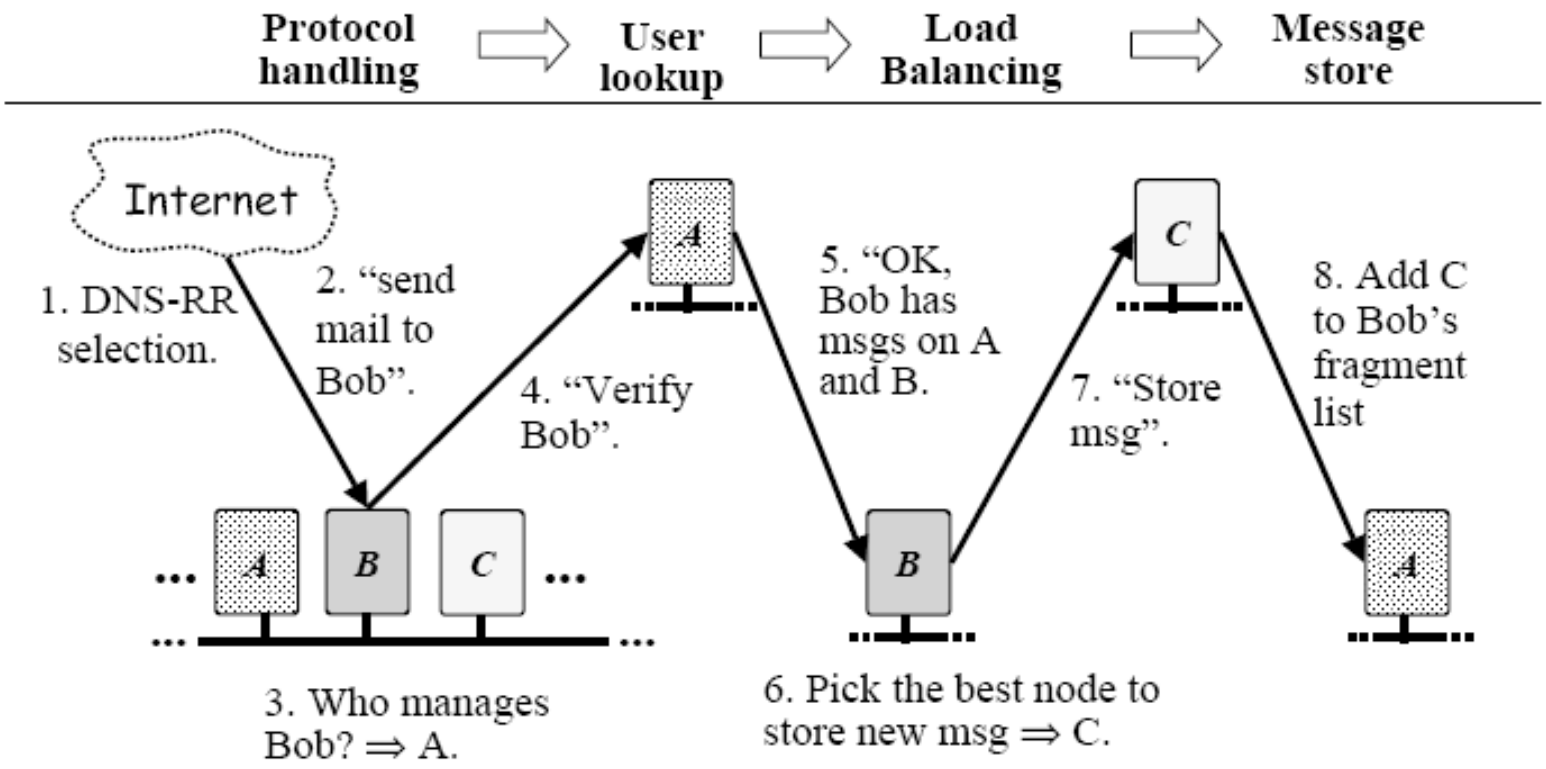
Semantics of Internet E-Mail

- May arrive out of order
- May arrive more than once
- May reappear after being deleted
- SMTP, POP, IMAP

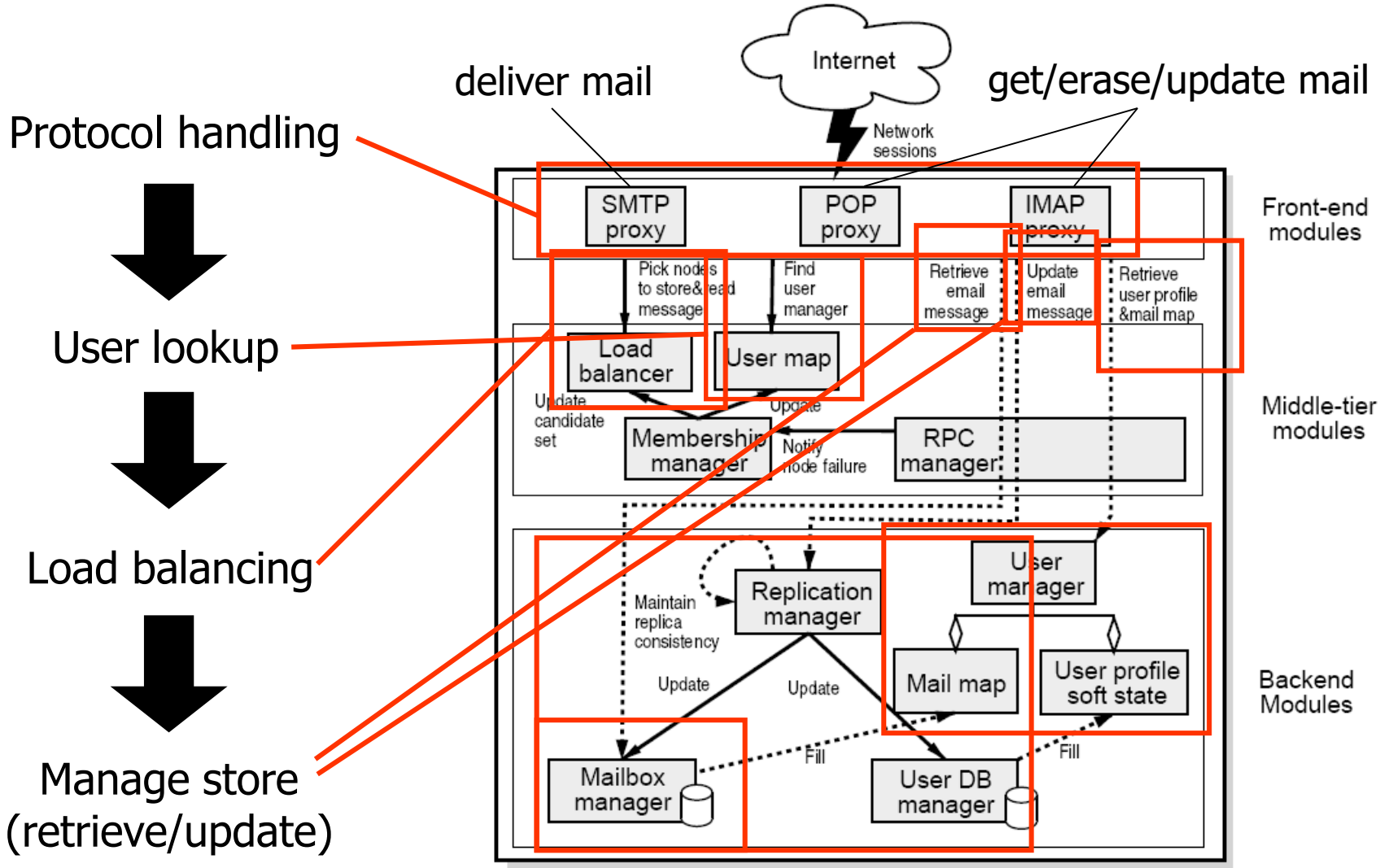
Mailbox management and storage



Mail delivery to "Bob"



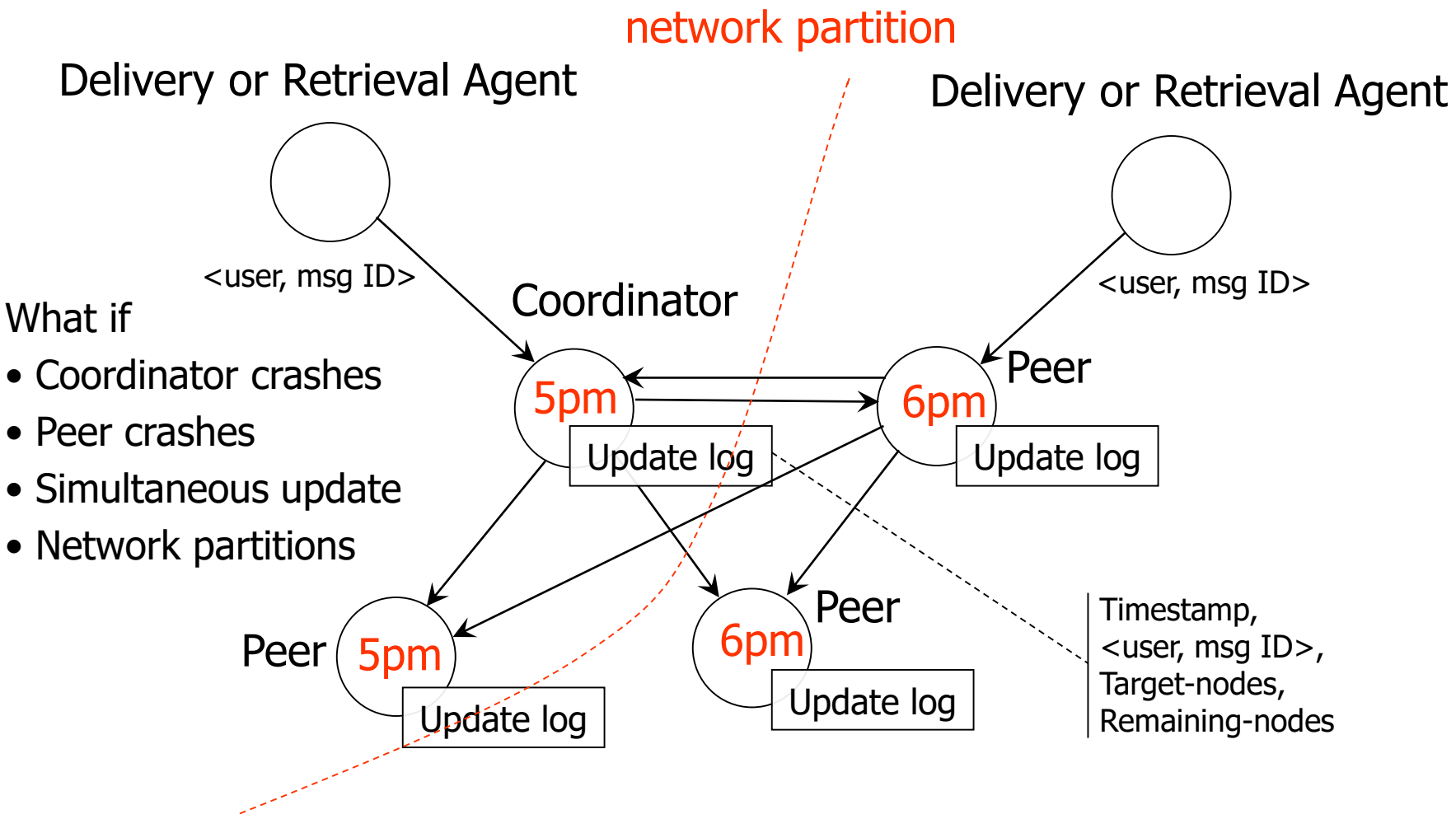
System architecture



Replication properties

- Update anywhere / Retrieve from anywhere
- Eventual consistency (weaker than “single-copy”)
- Total update
- Lock free
- Ordering by loosely synchronized clocks

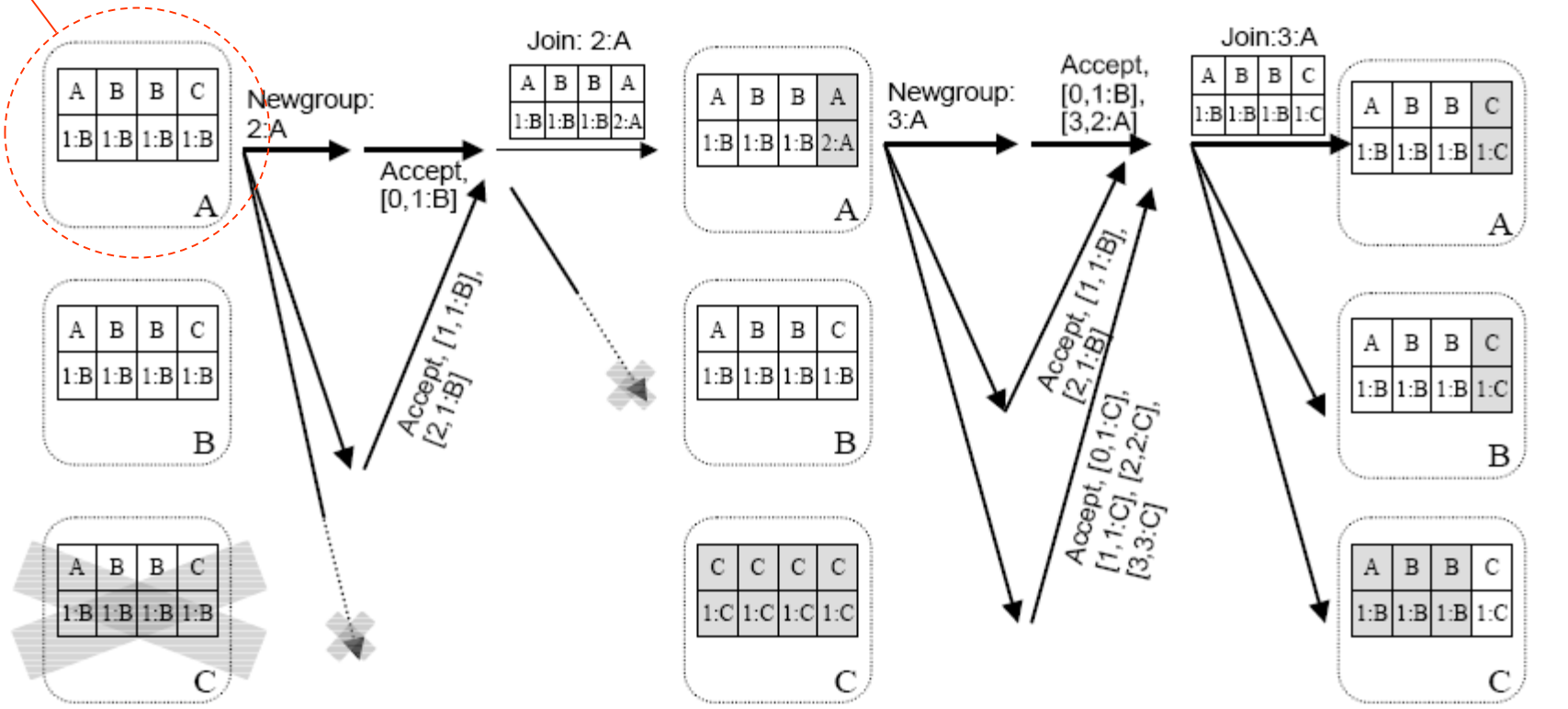
Update protocol



What if

- Coordinator crashes
- Peer crashes
- Simultaneous update
- Network partitions

Membership



C crashes. *A* detects the crash and starts TRM.

A and *B* reply with the epoch IDs of the buckets they manage.

A broadcasts the new membership and the user map, but *B* misses the packet.

C recovers. *A* detects the recovery and starts TRM.

A, *B*, and *C* reply with the epoch IDs of the buckets they manage.

A broadcasts the new membership and the user map.

Nodes finally agree on the membership and the user map.

User profile, mail-map reconstruction

- Identify buckets with new manager assignments
  - Compare Epoch IDs
- Send relevant soft state to new managers
  - Update their mail map with mailbox fragments
  - Update their user profile soft state with user profiles
- Replica with highest IP does the transfer
- Hard state bucketed into directories for quick search

Mail-map consistency

- Mail-map pointers lead to valid fragments
- Valid fragments reachable from mail-map pointers
- Example: {Bob, *} : {{A, B}, {A, B, C}, {A, C}}

Mail-map consistency

(1) A node fails just after a message is stored in a new mailbox fragment on its disk, but before the corresponding mail map is updated. This case causes no problem because this copy of the message becomes non-retrievable after the node failure. The replication service (Section 4) ensures that another copy of the message is still available.

(2) A node fails just after the last message in a mailbox fragment on its disk is deleted, but before the corresponding mail map is updated. Each node periodically scans the mail maps it manages and removes all "dangling" links to nodes not in the membership. The links will be restored when the failed nodes rejoin the cluster.

(3) A node stores a message in a new mailbox fragment on its disk, but the corresponding user manager node fails before the mail map is updated. The message will be discovered by the disk scan algorithm that runs after membership reconfiguration and will be added to the mail map on a new user manager node.

(4) A node deletes the last message in a mailbox fragment on its disk, but the corresponding user manager node fails before the mail map is updated. The same argument as

Dynamic load balancing

- Decentralized
- Fine grained
- Support heterogeneous clusters
- Automatic; minimize manual tuning
- Maximize throughput
- Resolve tension between load and affinity

Load balancing information

- Load characterization
  - Whether or not disk is full
  - Number of pending RPC that are disk-bound
- Load dissemination
  - Piggybacked on RPCs
  - Exchanged on virtual ring

Affinity-based scheduling

- Prefer adding email to nodes that already have a fragment, unless too busy
  - Decreases number of inter-node RPCs
  - Increases sequentiality of disk accesses
  - Reduces mail-map memory requirements
- Define “spread”
  - Soft limit, can be exceeded

Scalability