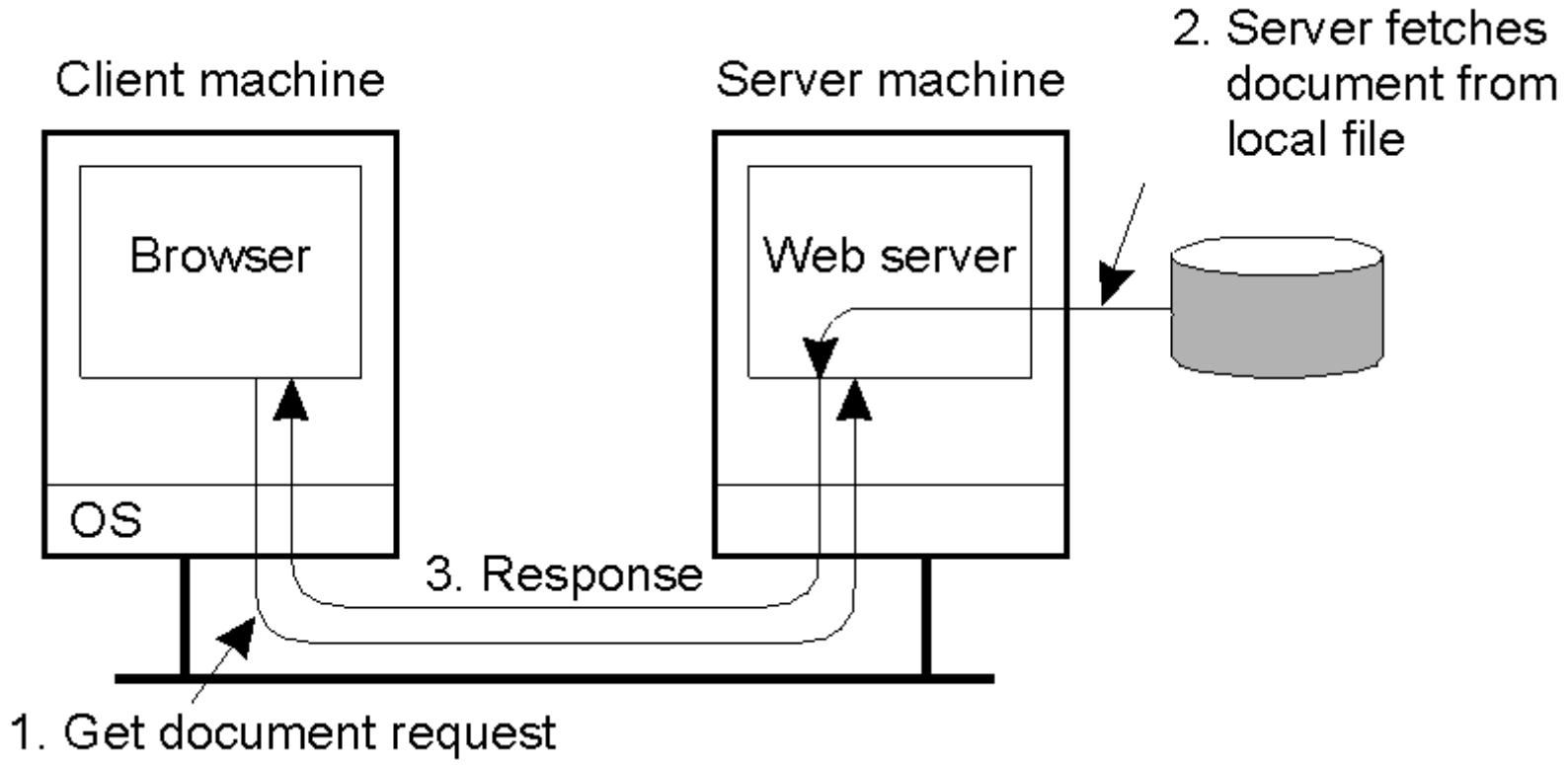
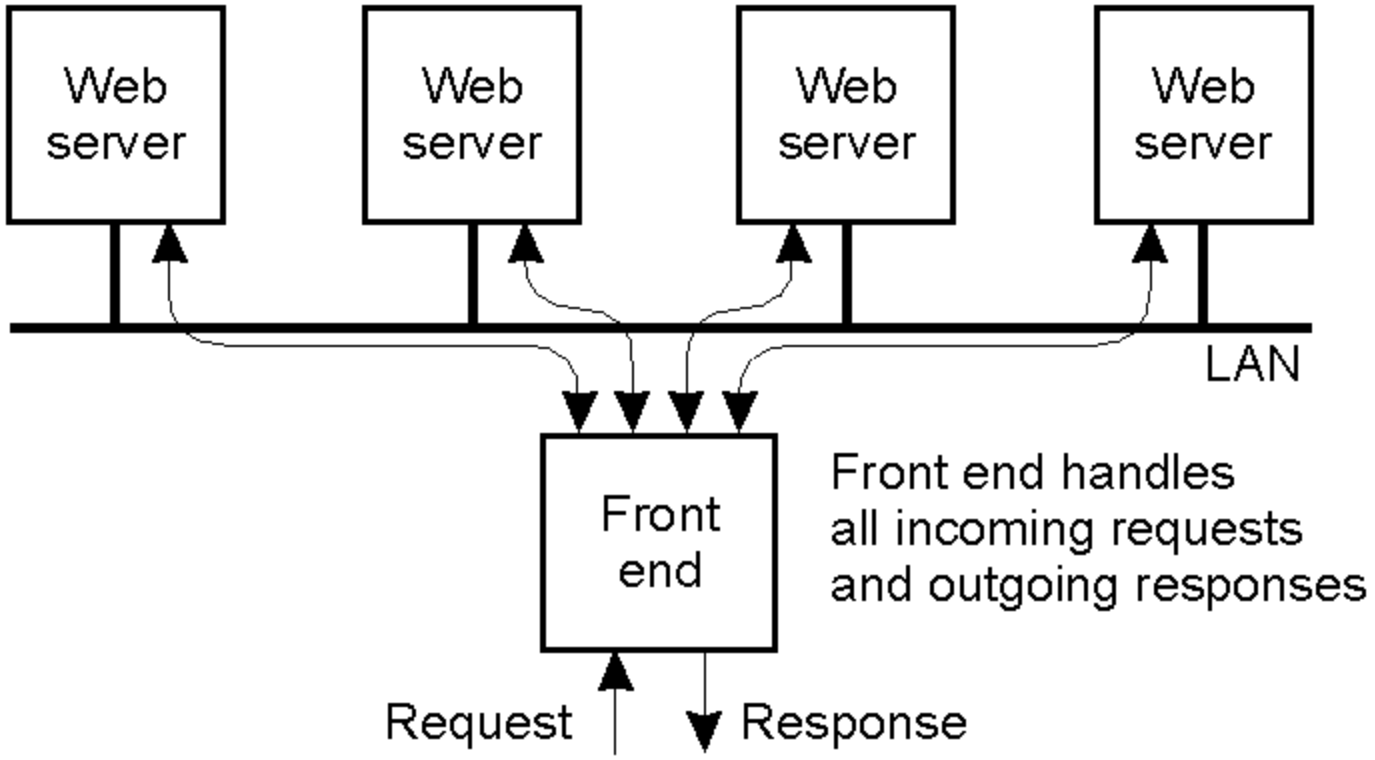
The World Wide Web



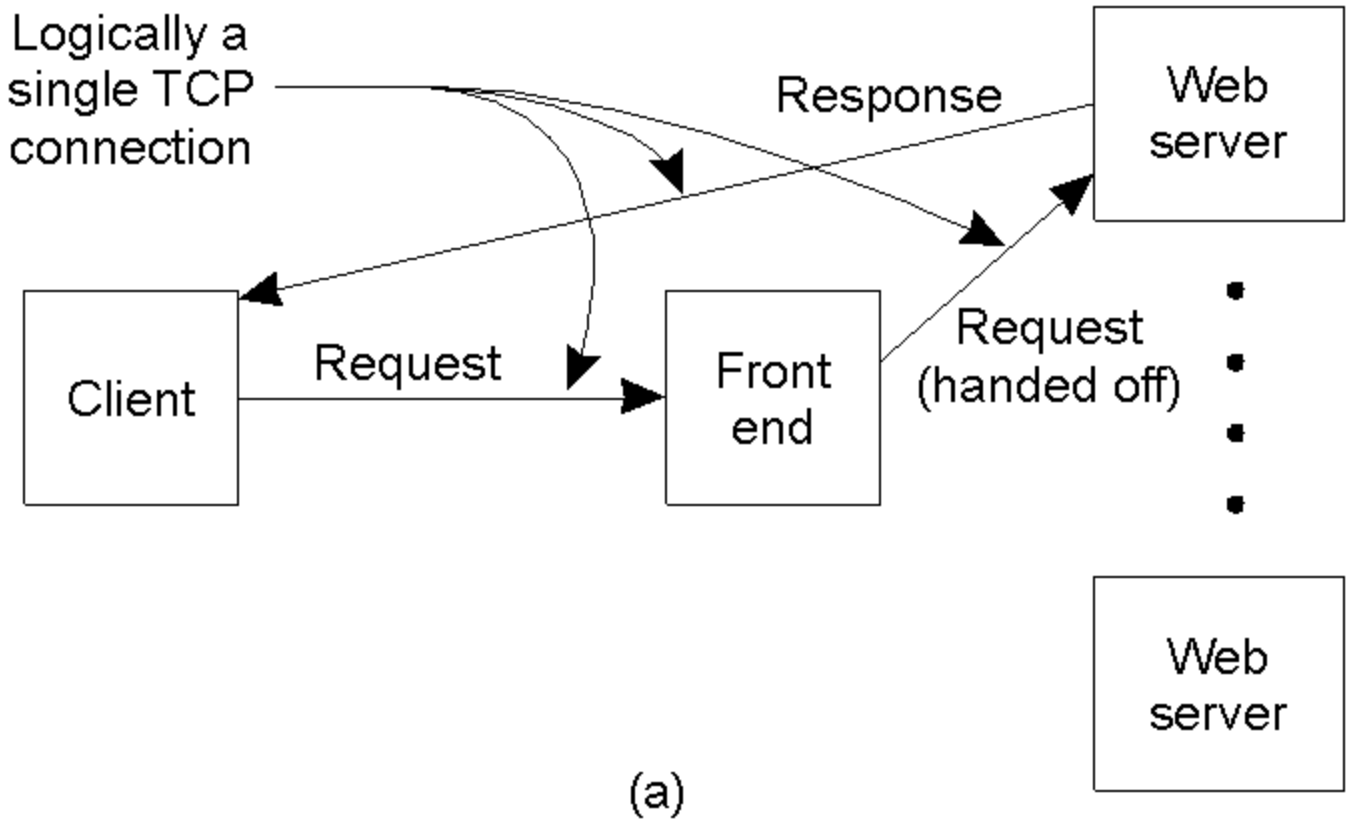
Overall organization of the Web.

Server Clusters (1)



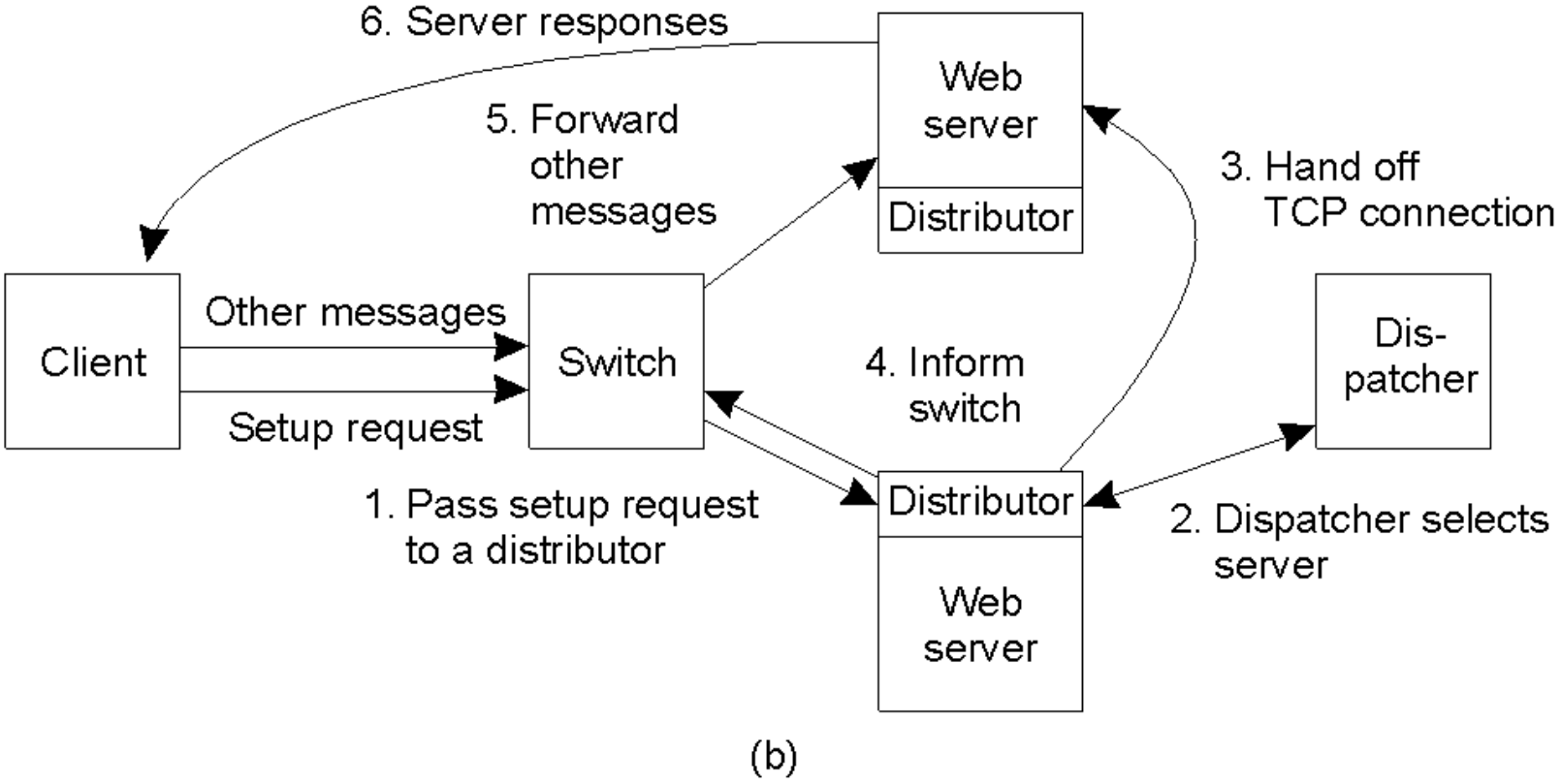
The principle of using a cluster of workstations to implement a Web service.

Server Clusters (2)



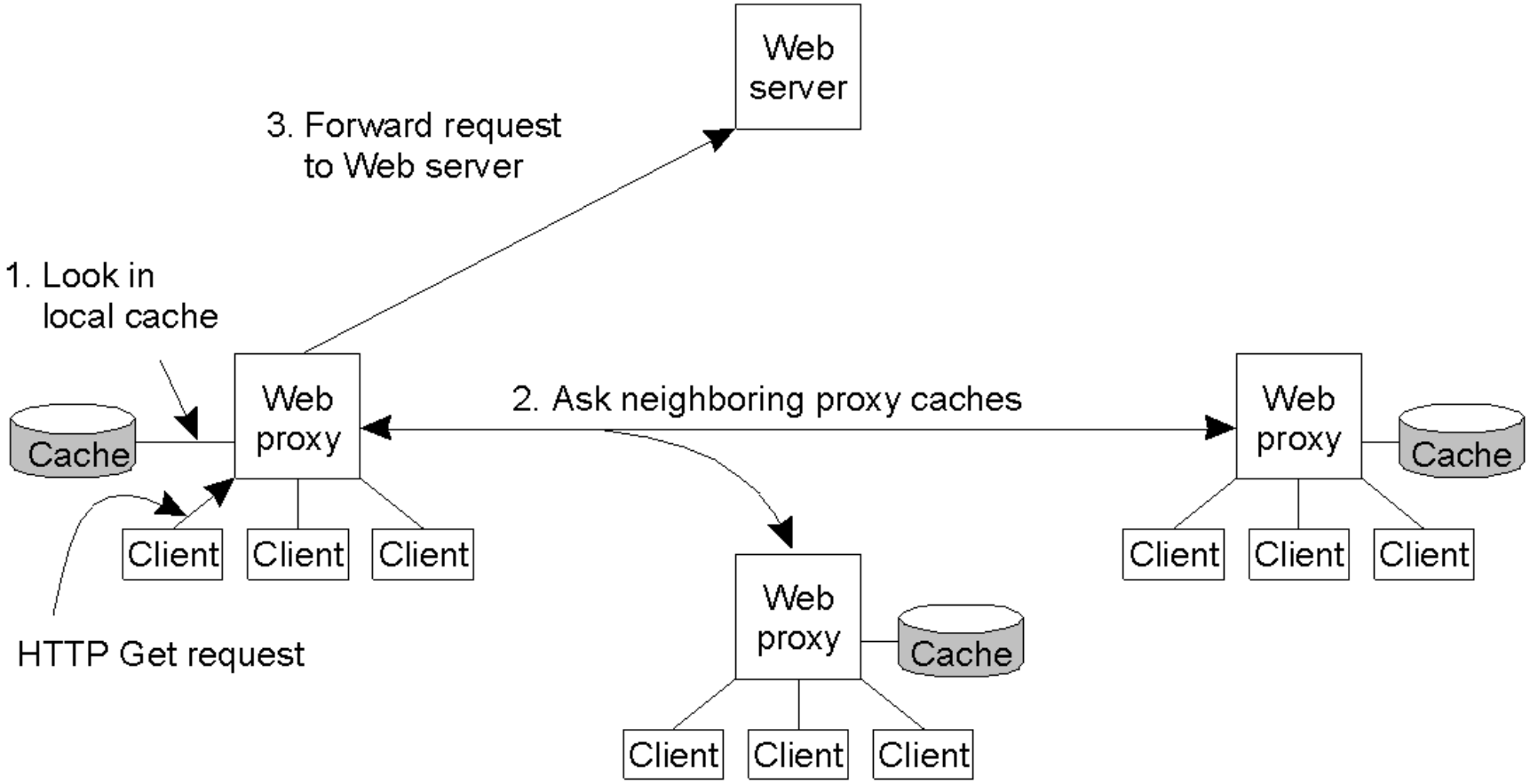
(a) The principle of TCP handoff.

Server Clusters (3)



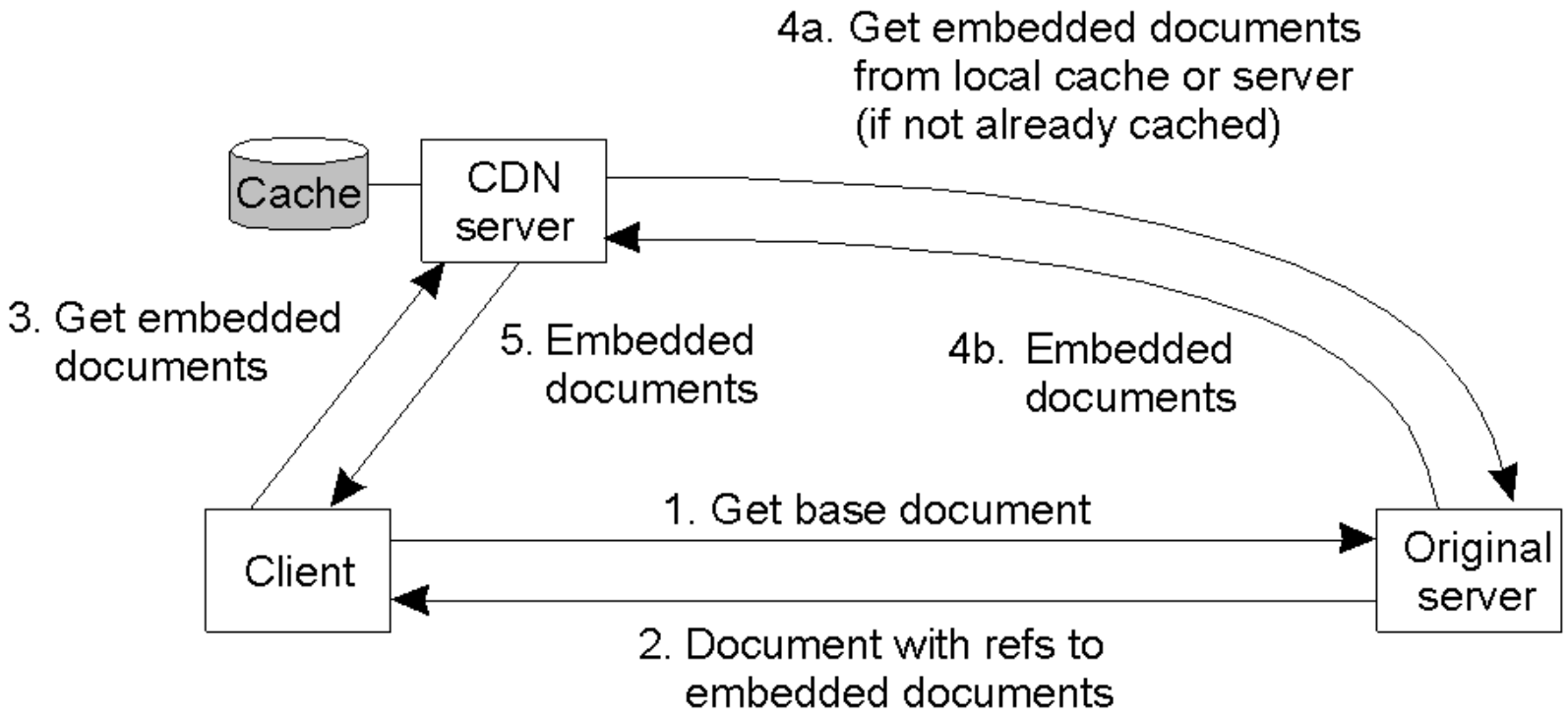
(b) A scalable content-aware cluster of Web servers.

Web (Proxy) Caching



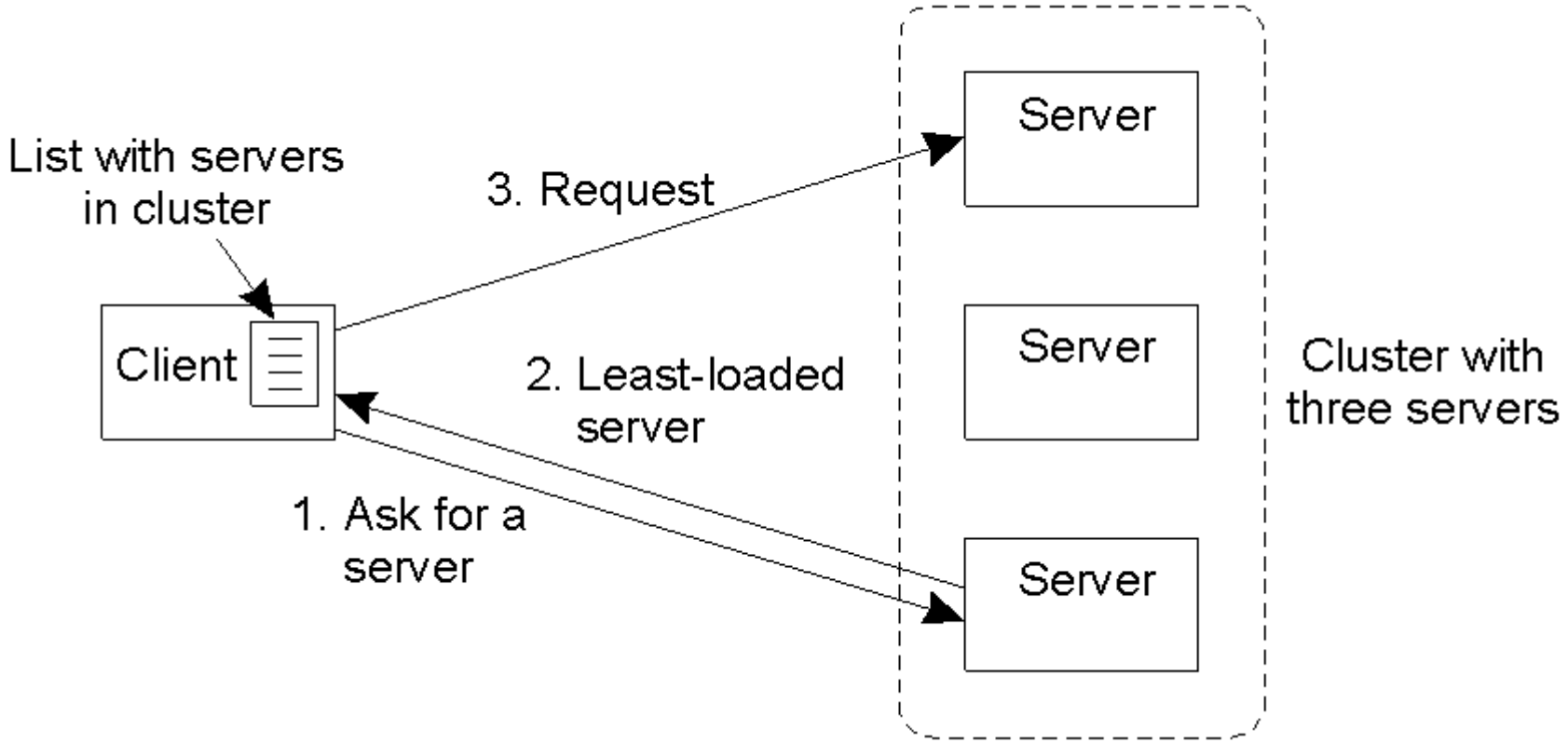
The principle of cooperative caching;
not only caching locally but also at neighborhood

Server Replication



In a CDN (content distribution network), parts of a document can come from different sources. Sources are chosen based on load, location, latency, cost, etc.

Server Replication (2)

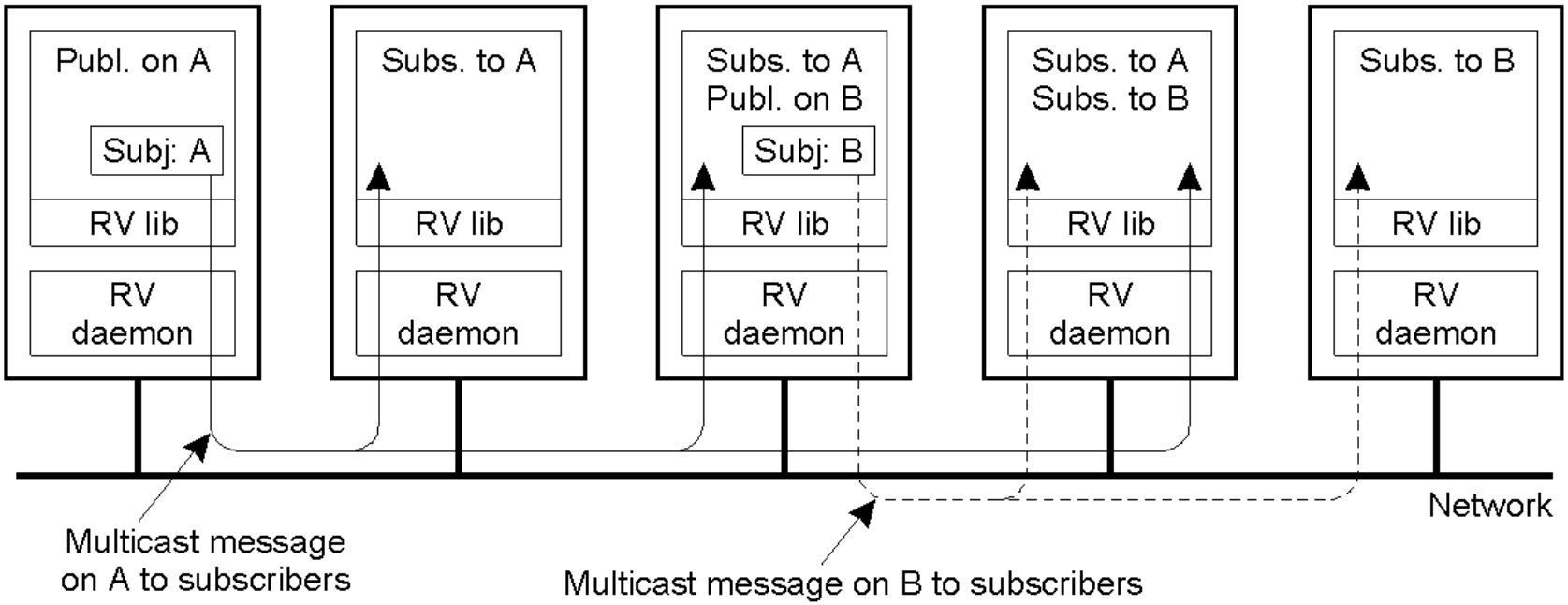


Request handling in a cluster of Domino servers.

Replication Schemes

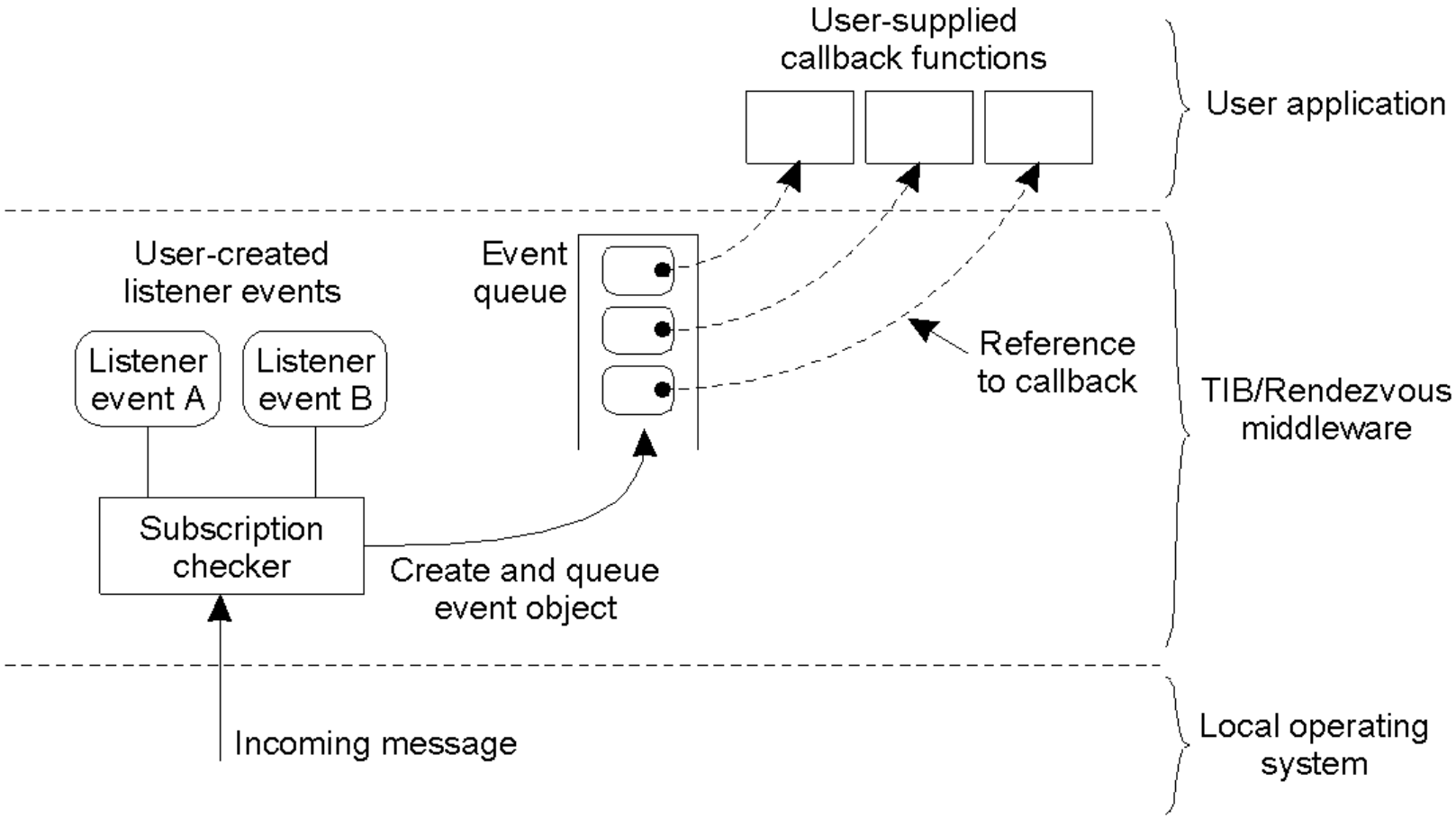
Scheme	Description
Pull-push	A replicator task pulls updates in from a target server, and pushes its own updates to that target as well
Pull-pull	A replicator task pulls in updates from a target server, and responds to update fetch requests from that target
Push-only	A replicator task only pushes its own updates to a target server, but does not pull in any updates from the target
Pull-only	A replicator only pulls in updates from a target server, but does not give any of its own updates to that target

Publish/subscribe system



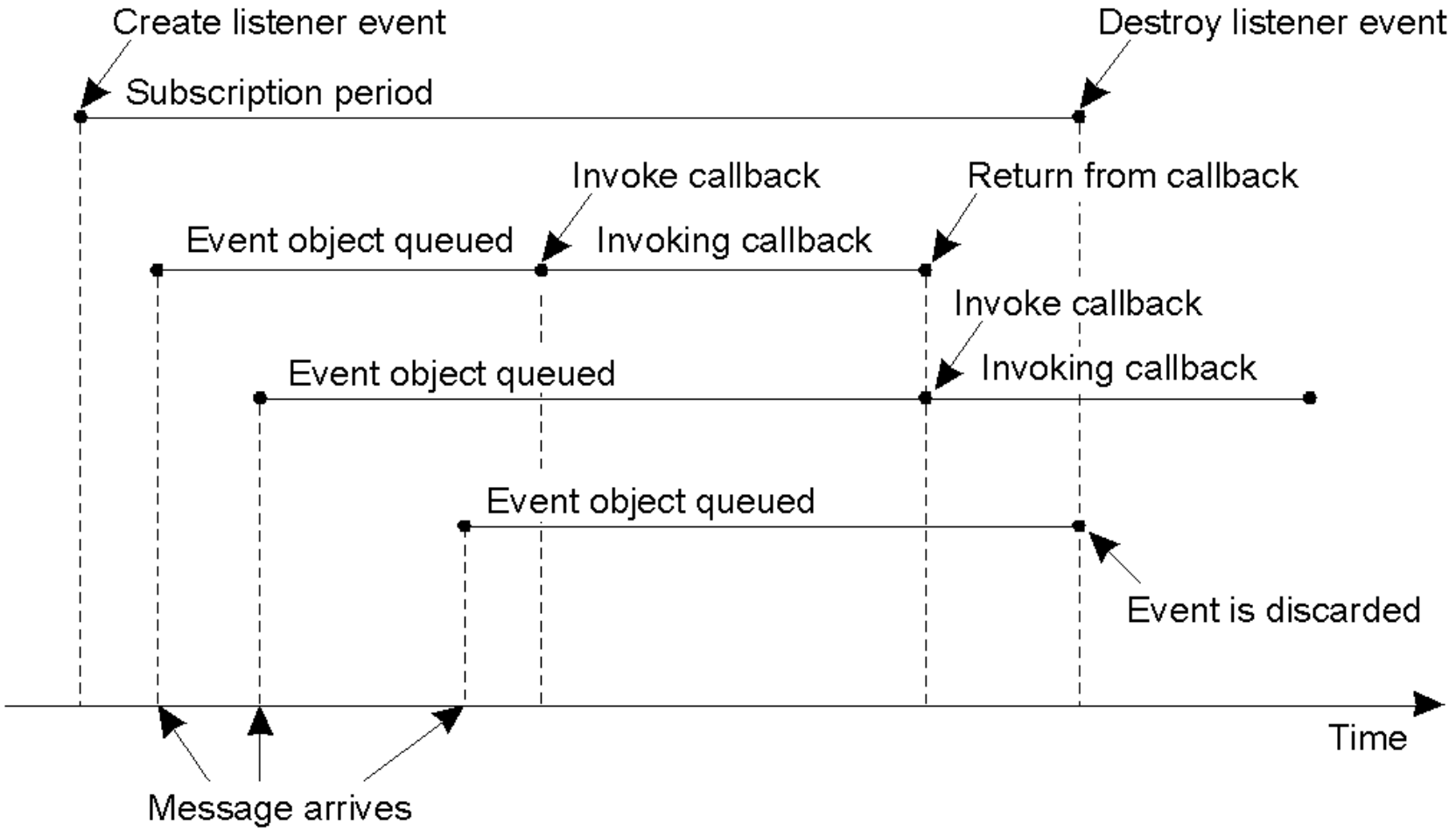
RV = Rendezvous, or consistentcy/synchronization primitives

Events (1)



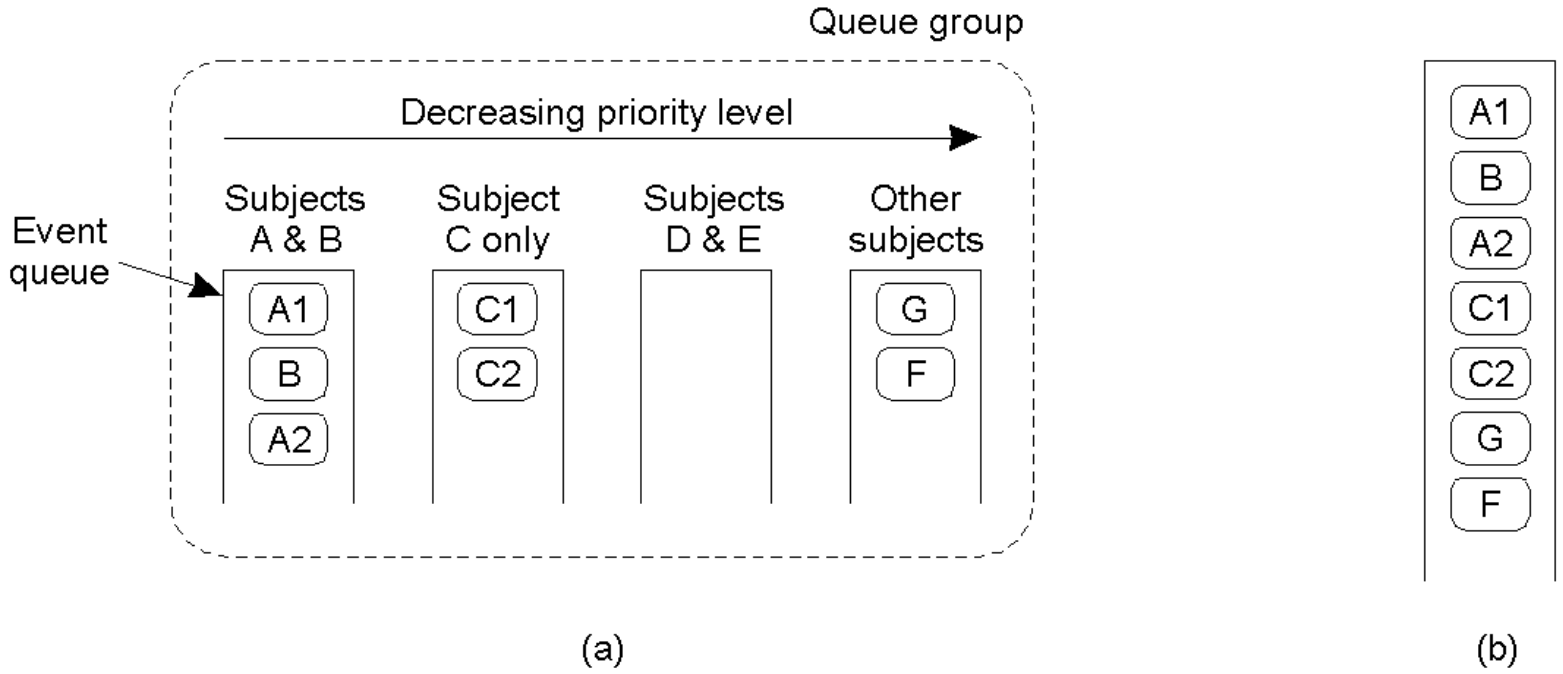
Processing listener events for subscriptions in TIB/Rendezvous.

Events (2)



Processing incoming messages in TIB/Rendezvous.

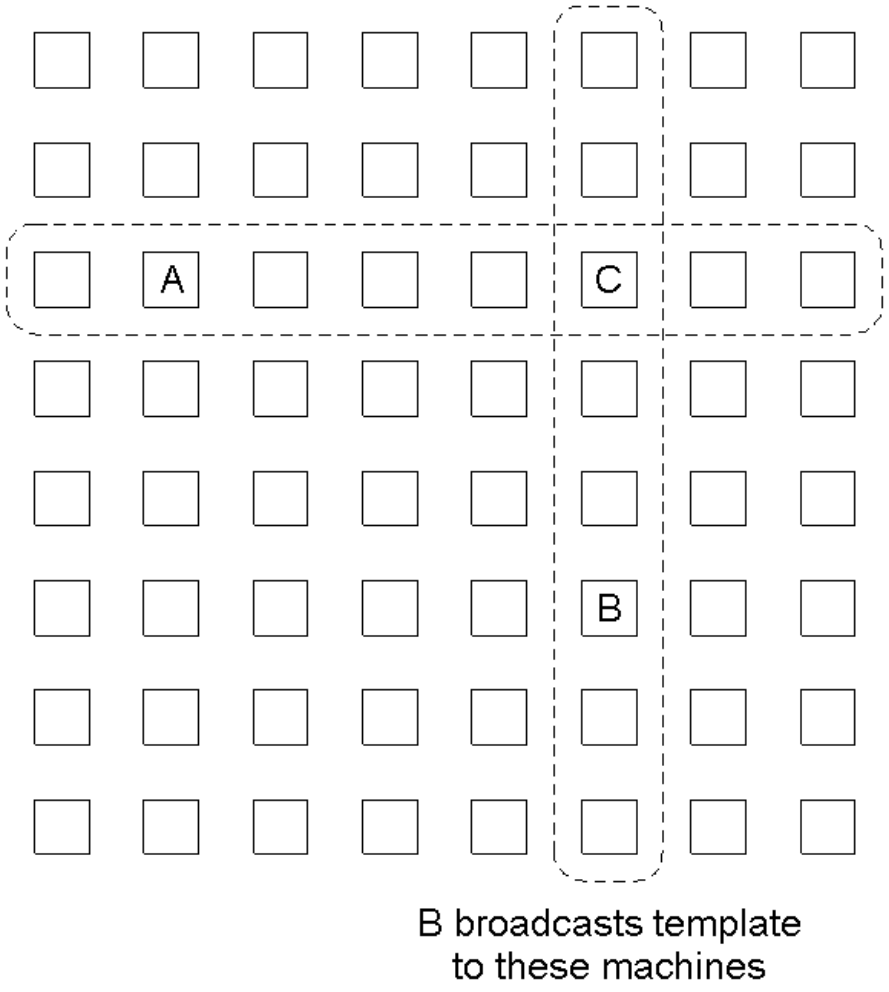
Processes



a) Priority scheduling of events through a queue group.

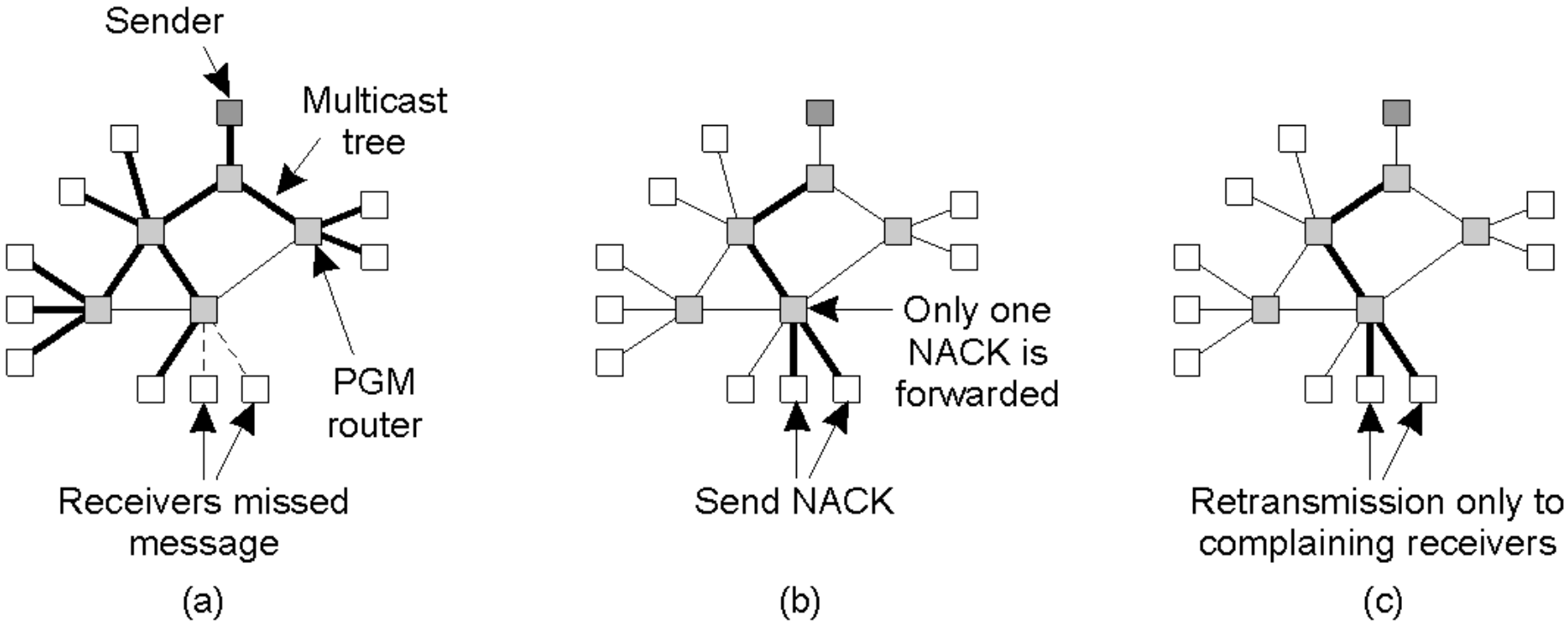
b) A semantically equivalent queue for the queue group with the specific event objects from (a).

Processes (2)



- Partial broadcasting of tuples and template tuples
- Achieved through multicasting or unicasting

Reliable Multicasting



The principle of PGM (Pragmatic General Multicast)

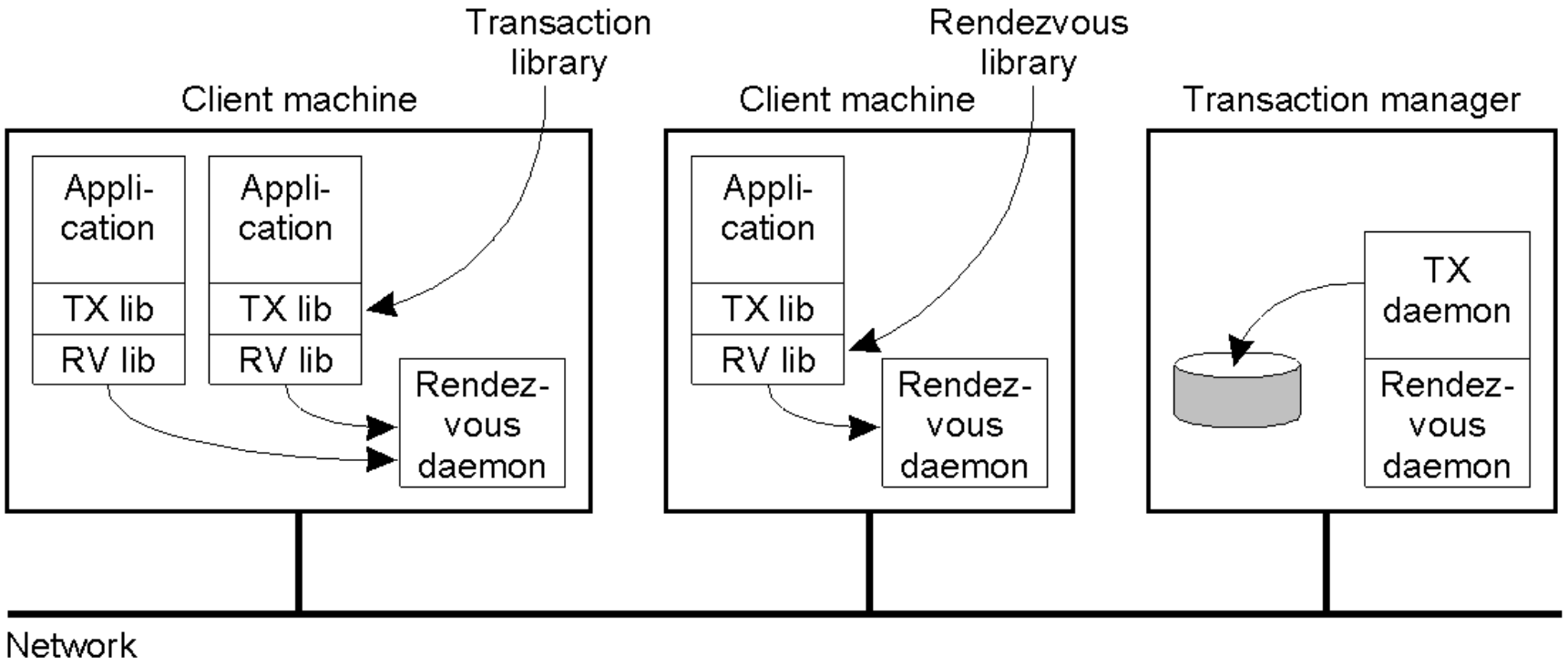
a) A message is sent along a multicast tree (n nodes, n-1 links: efficient)

b) A router will pass only a single NACK for each message

c) A message is retransmitted only to receivers that have asked for it.

Synchronization (1)

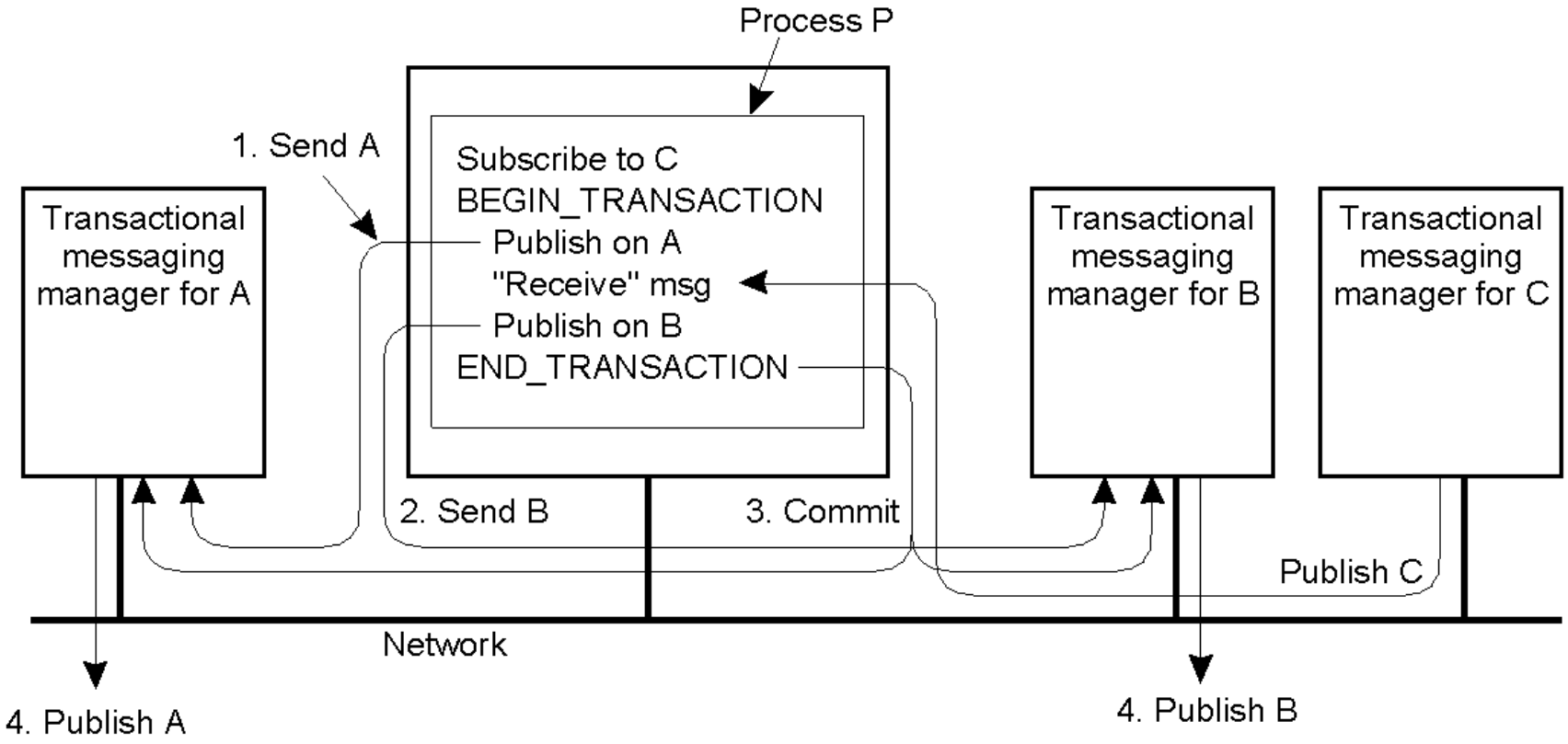
interesting info: not in exam



The organization of transactional messaging as a separate layer in TIB/Rendezvous.

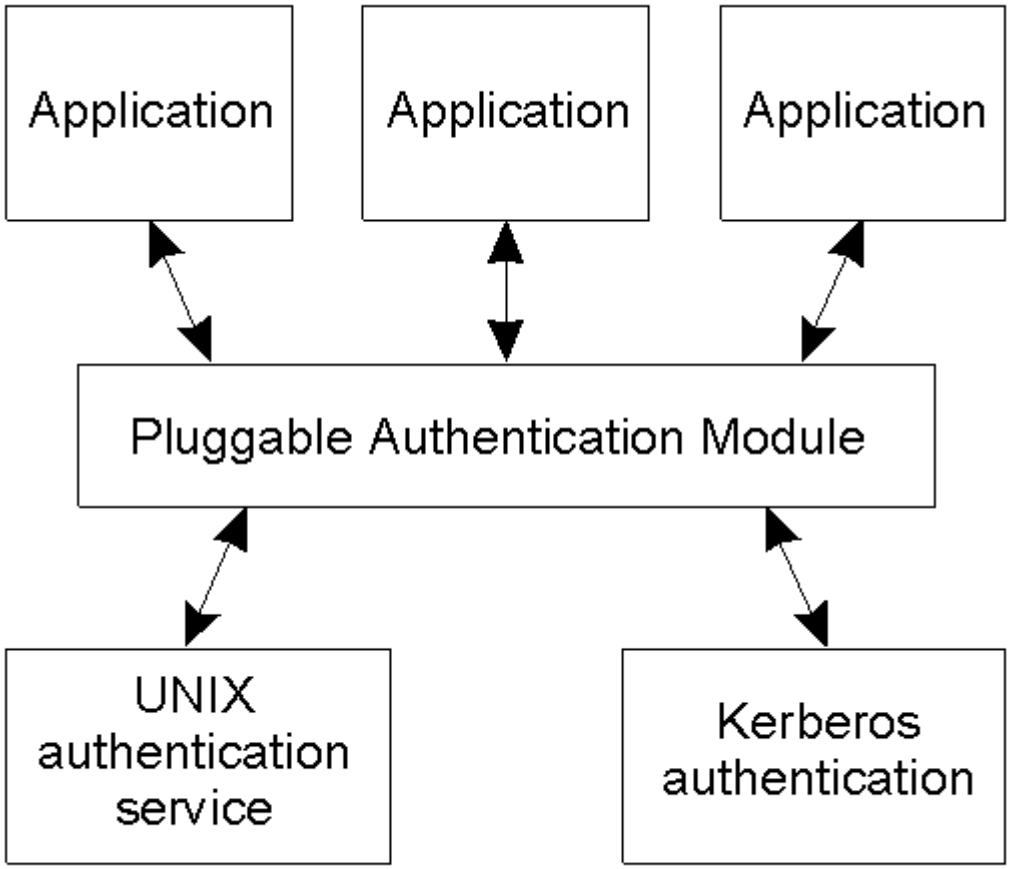
Synchronization (2)

interesting info: not in exam



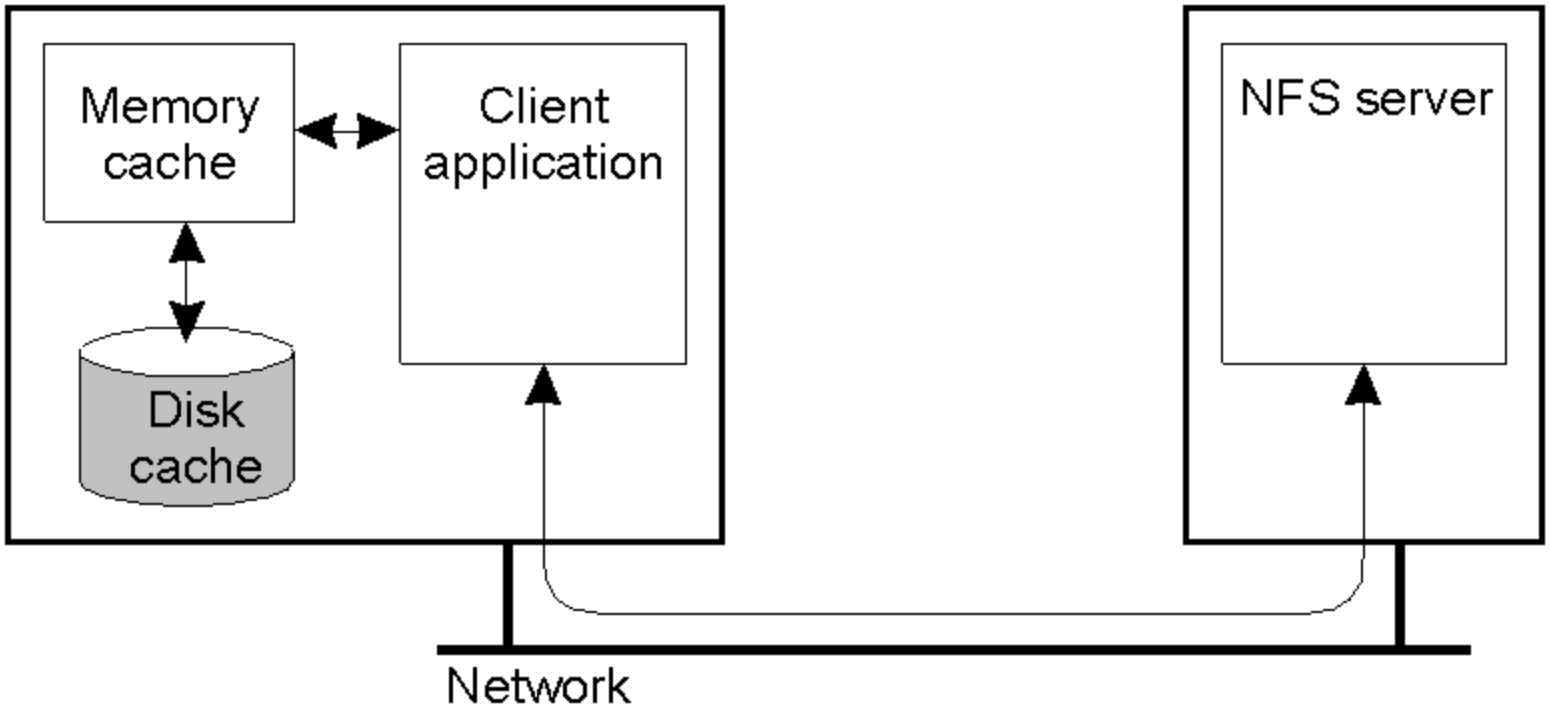
The organization of a transaction in TIB/Rendezvous.

Caching and Replication



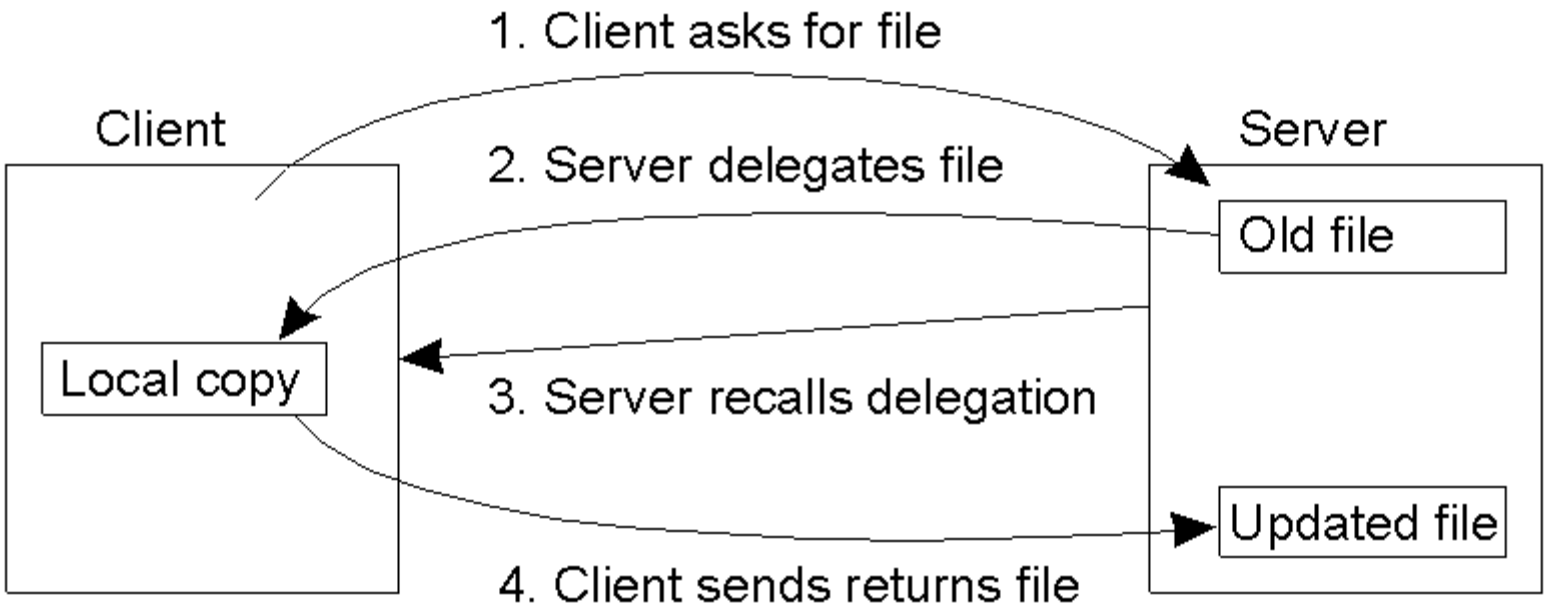
The position of PAM with respect to security services.

Client Caching (1)



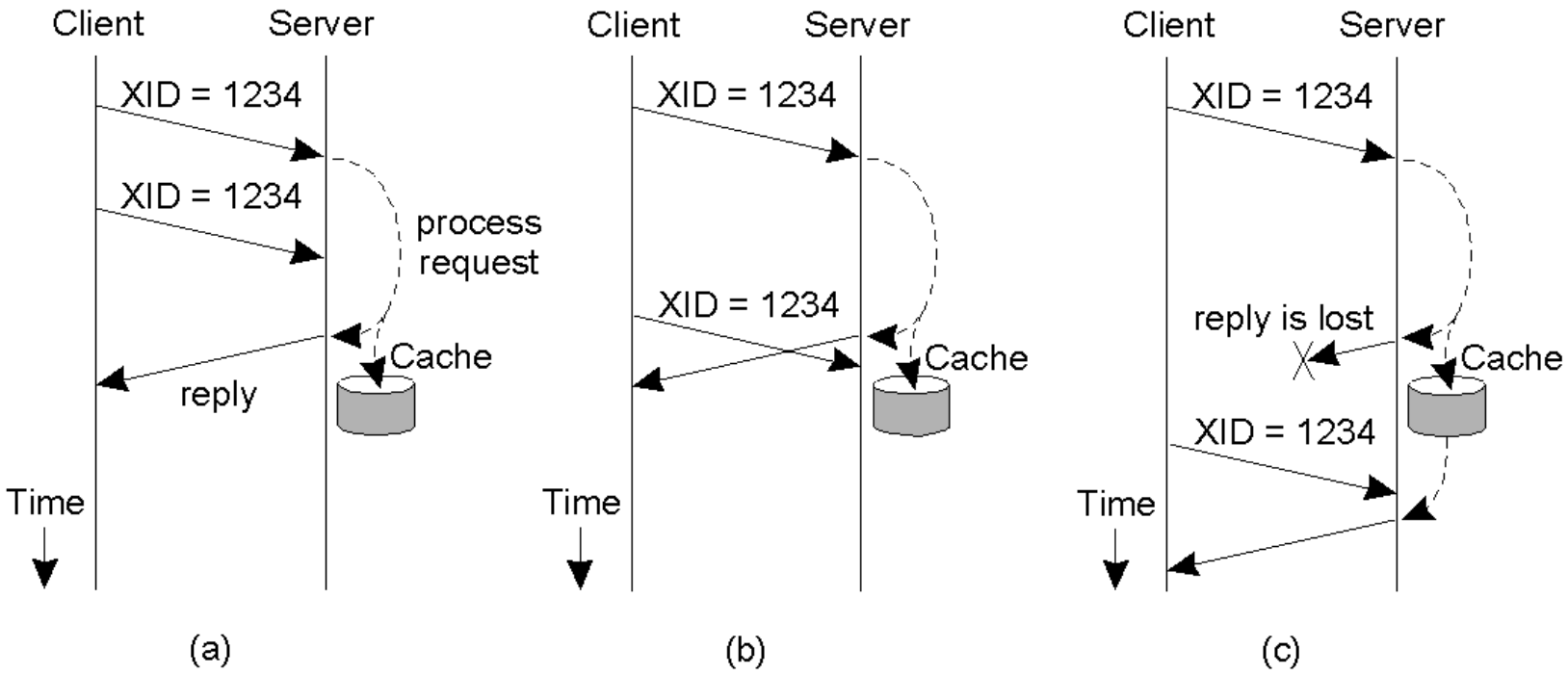
Client-side caching in NFS.

Client Caching (2)



Using the NFS version 4 callback mechanism to recall file delegation.

Remote Procedure Call (PRC) Failures



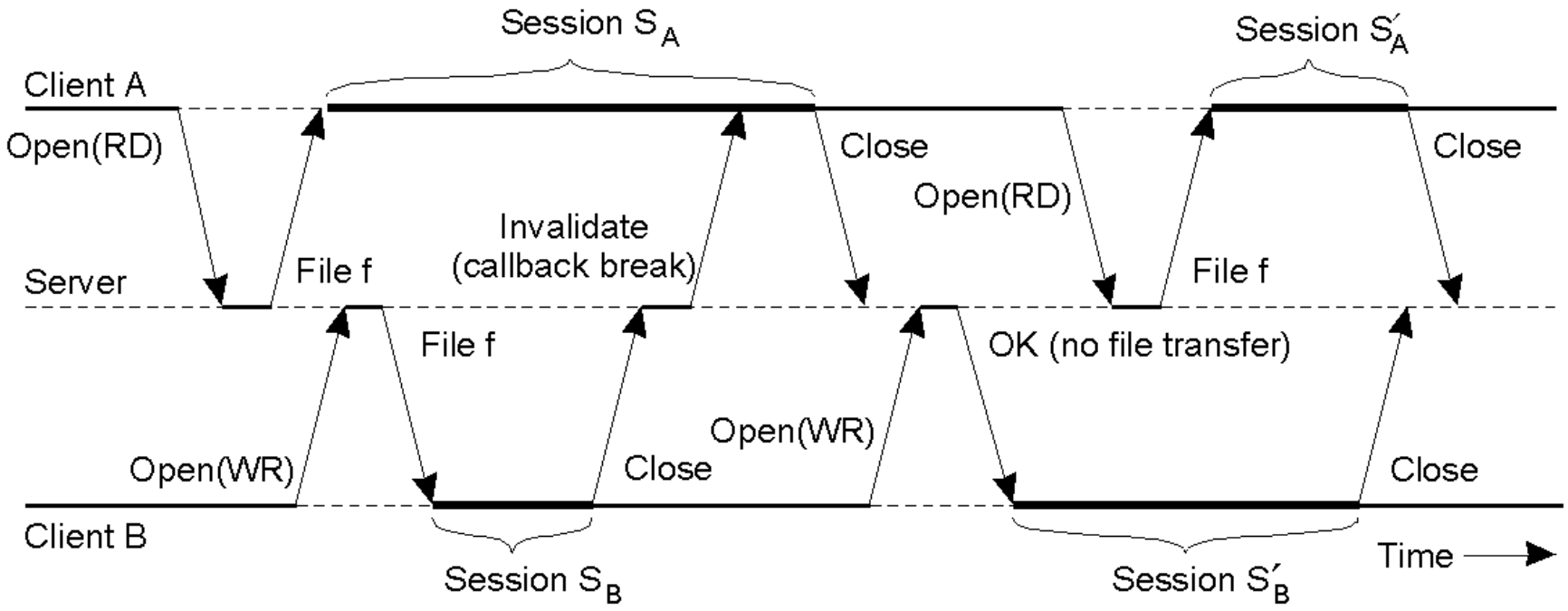
Three situations for handling retransmissions.

a) The request is still in progress

b) The reply has just been returned

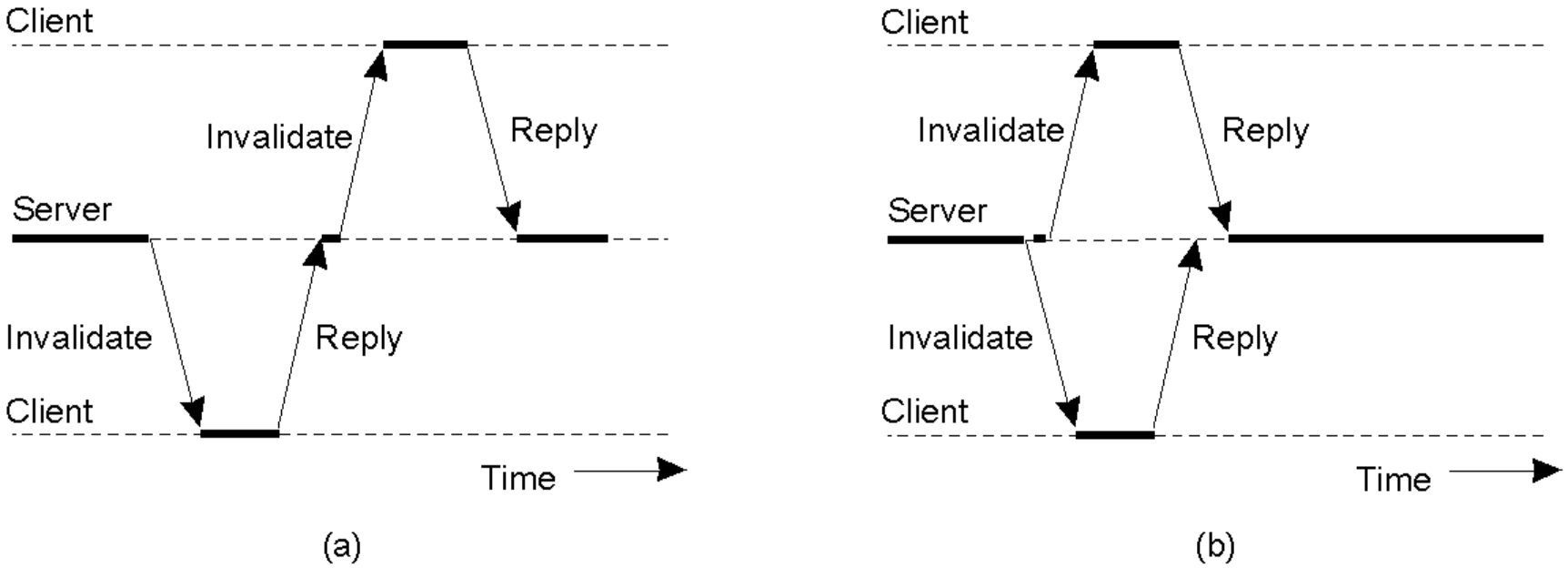
c) The reply was done earlier, but was lost.

Client Caching



The use of local copies when opening a session in Coda.

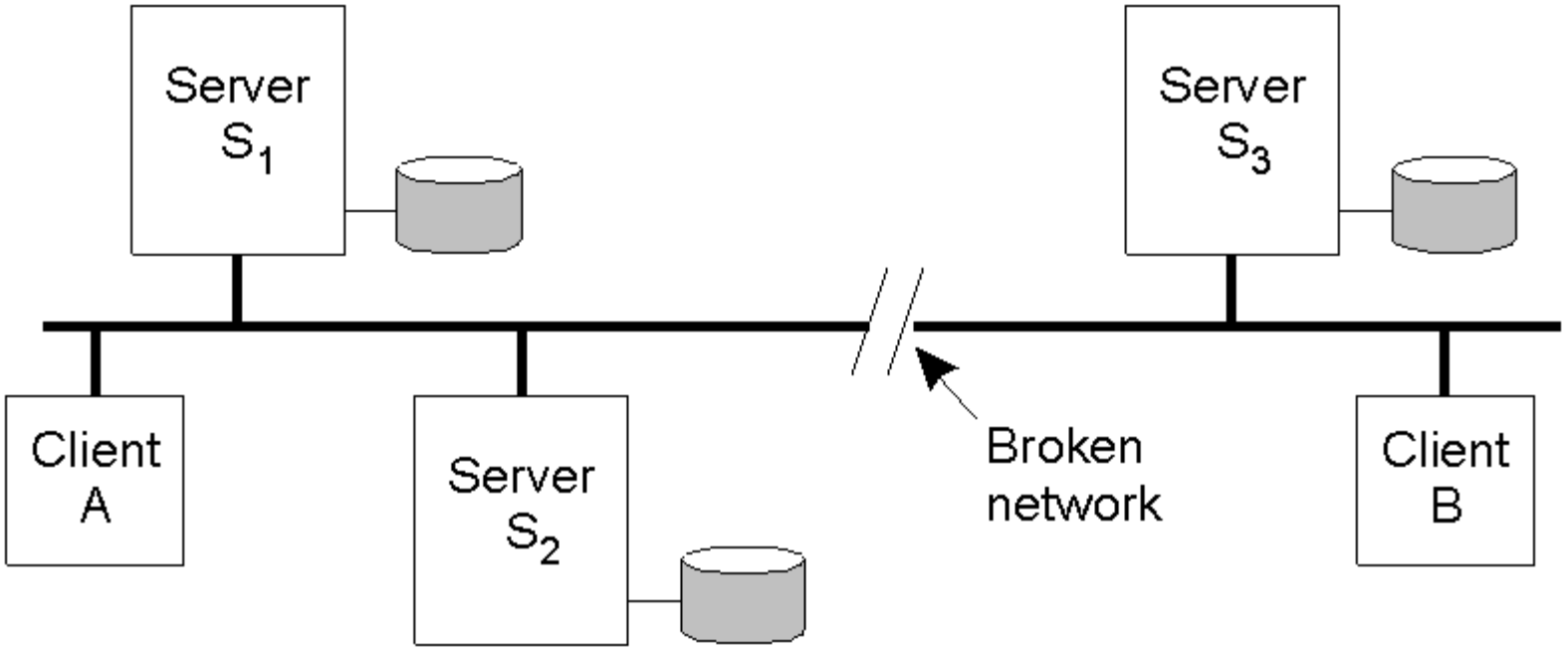
Cache invalidation



a) Sending an invalidation message one at a time.

b) Sending invalidation messages in parallel.

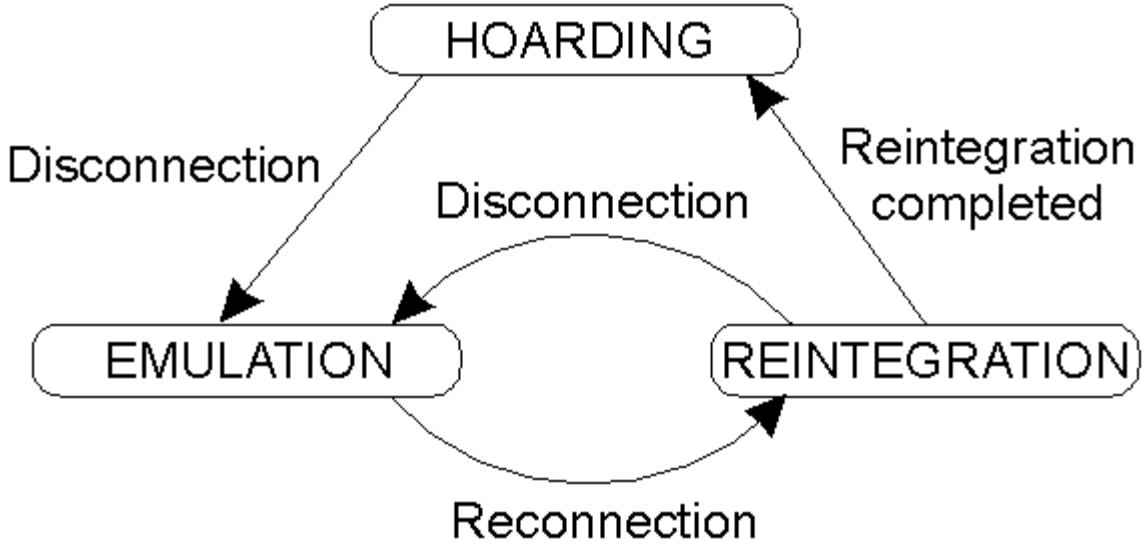
Server Replication



Two clients with different copies of the same replicated file.

Network partitions may cause problems and solutions

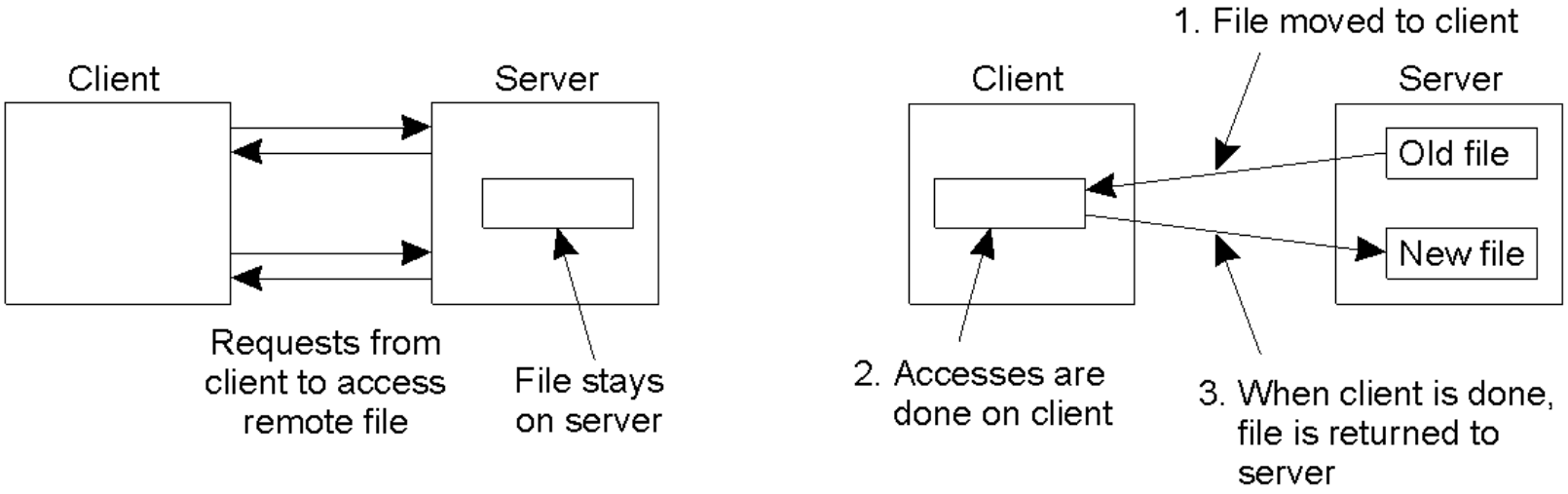
Disconnected Operation (network partitions on purpose)



The state-transition diagram of a disconnected client

Hoarding = gathering data; emulation = doing operations

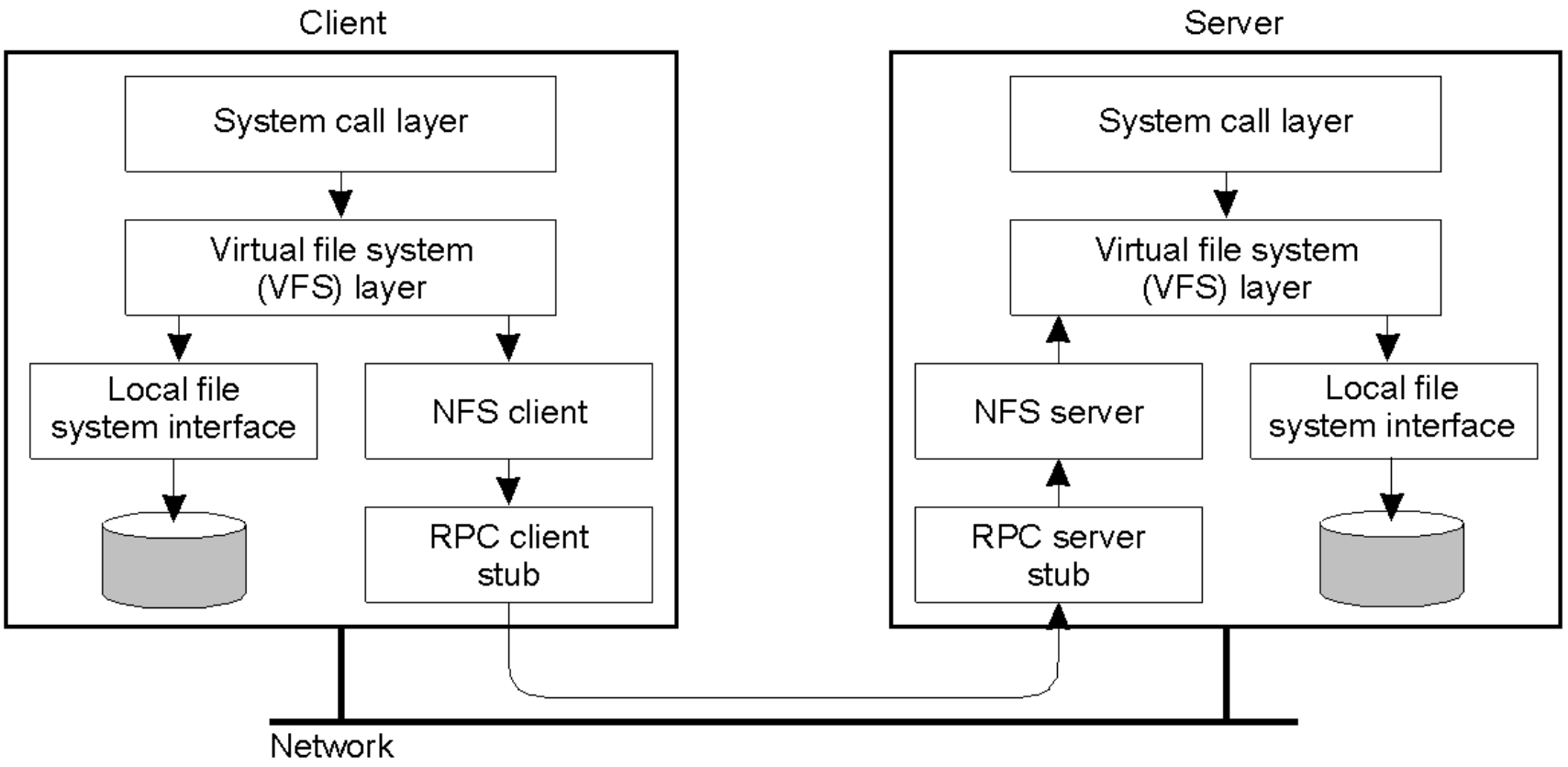
Case study: NFS (network file system) (1)



a) The remote access model.

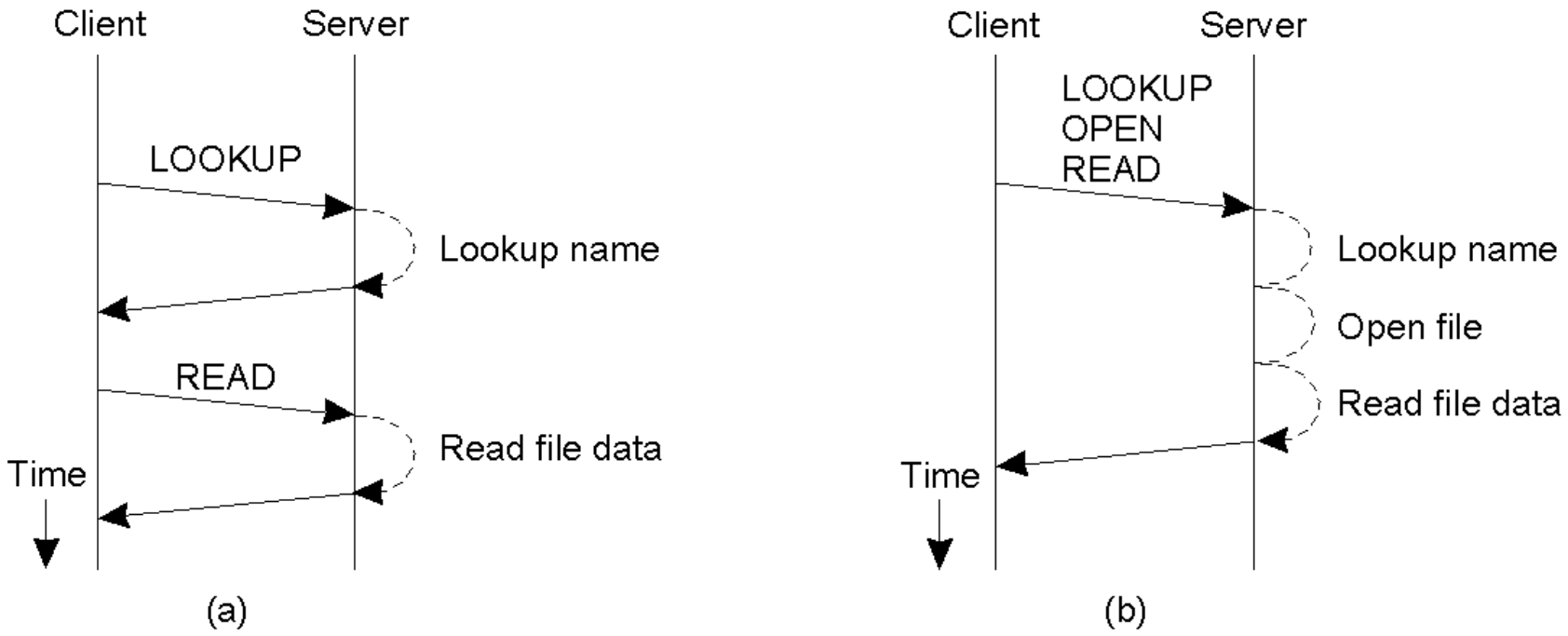
b) The upload/download model

NFS Architecture (2)



The basic NFS architecture for UNIX systems.

Communication



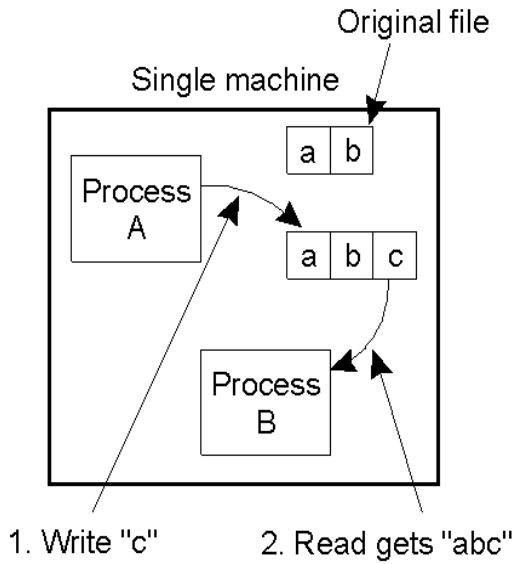
a) Reading data from a file in NFS version 3.

b) Reading data using a compound procedure in version 4.

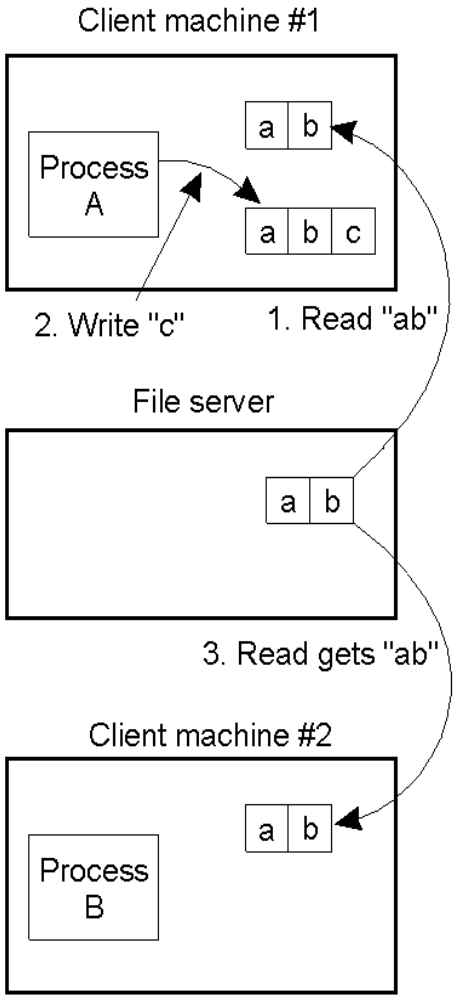
Semantics of File Sharing (1)

a) On a single processor, when a *read* follows a *write*, the value returned by the *read* is the value just written.

b) In a distributed system with caching, obsolete values may be returned.



(a)



(b)

Semantics of File Sharing (2)

Method	Comment
UNIX semantics	Every operation on a file is instantly visible to all processes
Session semantics	No changes are visible to other processes until the file is closed
Immutable files	No updates are possible; simplifies sharing and replication
Transaction	All changes occur atomically

Four ways of dealing with the shared files in a distributed system.

File Locking in NFS (1)

interesting info : not in exam

Operation	Description
Lock	Creates a lock for a range of bytes
Lockt	Test whether a conflicting lock has been granted
Locku	Remove a lock from a range of bytes
Renew	Renew the lease on a specified lock

NFS version 4 operations related to file locking.

File Locking in NFS (2)

interesting info: not in exam

Current file denial state

Request access	NONE	READ	WRITE	BOTH
READ	Succeed	Fail	Succeed	Succeed
WRITE	Succeed	Succeed	Fail	Succeed
BOTH	Succeed	Succeed	Succeed	Fail

(a)

Requested file denial state

Current access state	NONE	READ	WRITE	BOTH
READ	Succeed	Fail	Succeed	Succeed
WRITE	Succeed	Succeed	Fail	Succeed
BOTH	Succeed	Succeed	Succeed	Fail

(b)

The result of an *open* operation with share reservations in NFS.

a) When the client requests shared access given the current denial state.

b) When the client requests a denial state given the current file access state.