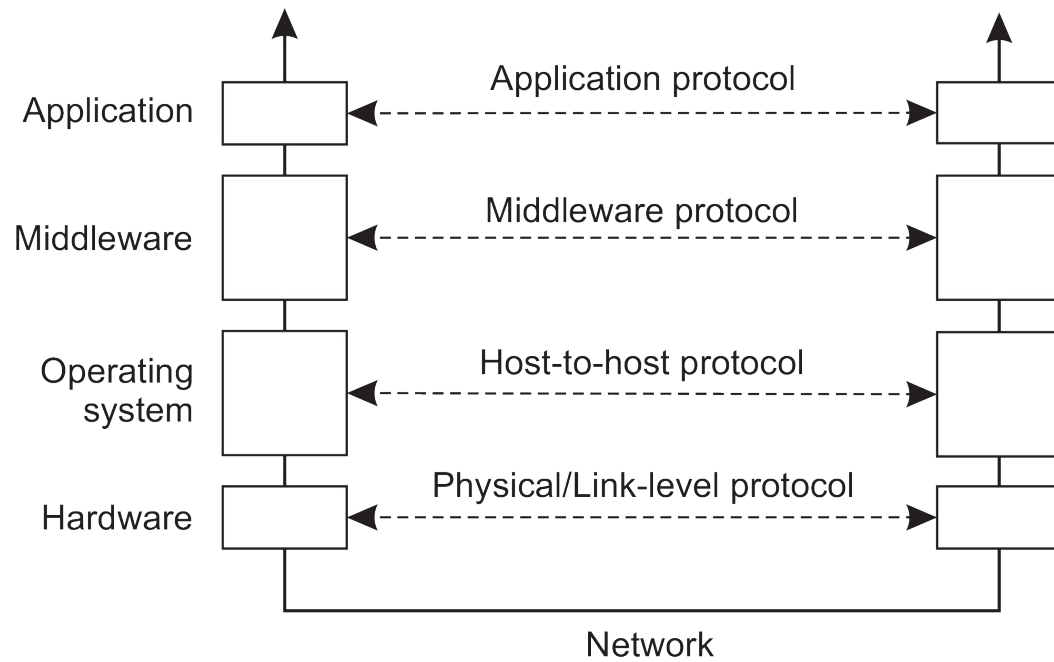


COMMUNICATION

Dr. Padraig Corcoran

- Interprocess communication is at the heart of distributed computing.
- In distributed systems communication is achieved using messages (shared memory not an option).
- Low-level communication facilities of computer networks do not offer **distribution transparency**.
- Therefore higher level middleware solutions are typically used.



An adapted reference model for networked communication.

Connection-oriented and connectionless communication

Connection-oriented communication:

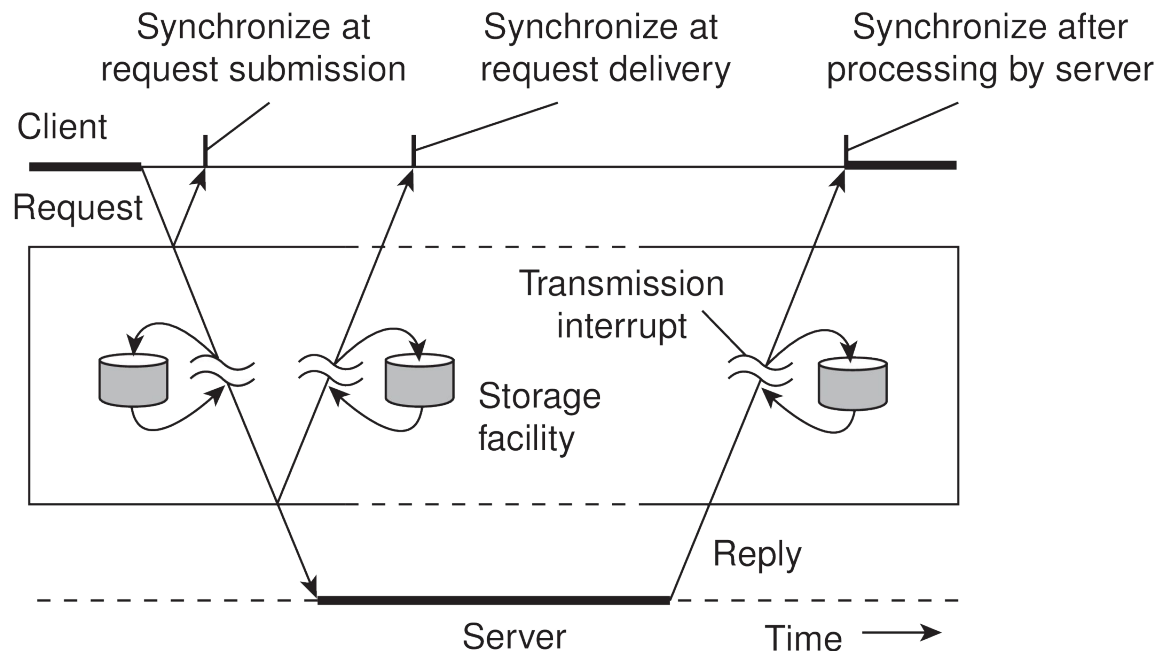
- Before exchanging data the sender and receiver first explicitly establish a connection.
- When they are done, they release (terminate) the connection.
- The telephone is a connection-oriented communication service.

Connectionless communication:

- No setup in advance is needed.
- The sender just transmits the first message when it is ready.
- Dropping a letter in a mailbox is an example of a connectionless communication service.

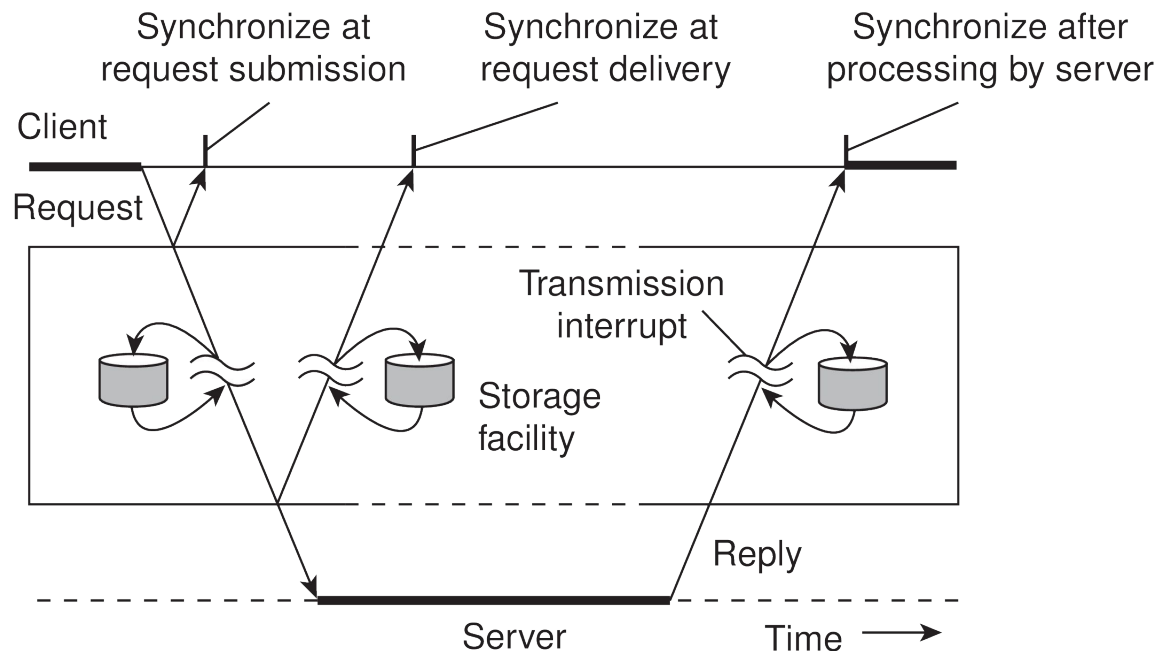
Persistent and transient communication

- **Persistent communication** - message submitted for transmission is stored by communication system as long as it takes to deliver it to the receiver.
- **Transient communication** - message stored by communication system only as long as the sender and receiver executing.



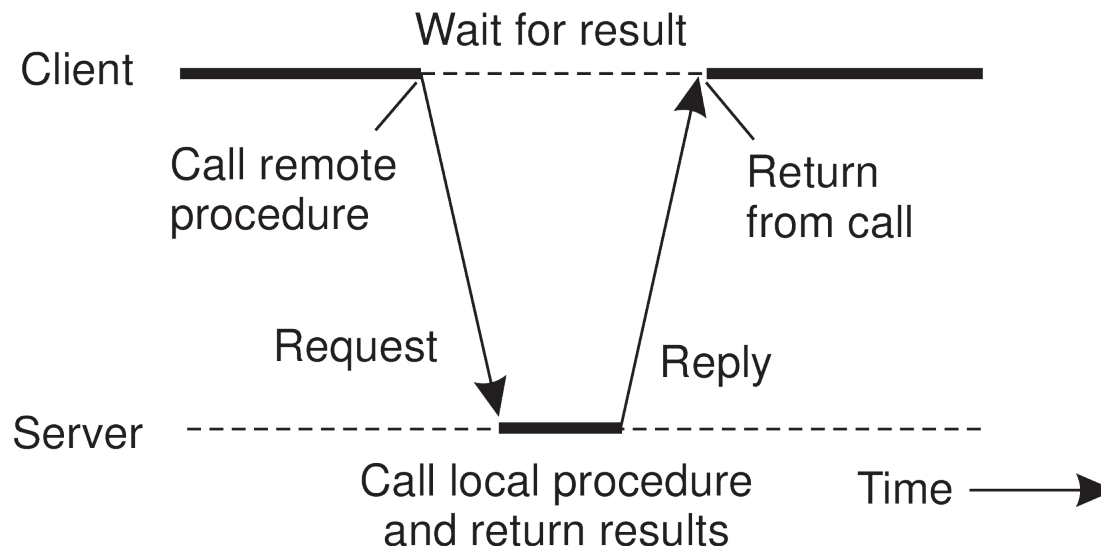
Asynchronous and synchronous communication

- **Asynchronous communication** - sender continues immediately after it has submitted its message for transmission.
- **Synchronous communication** - sender is blocked until its request is known to be accepted.



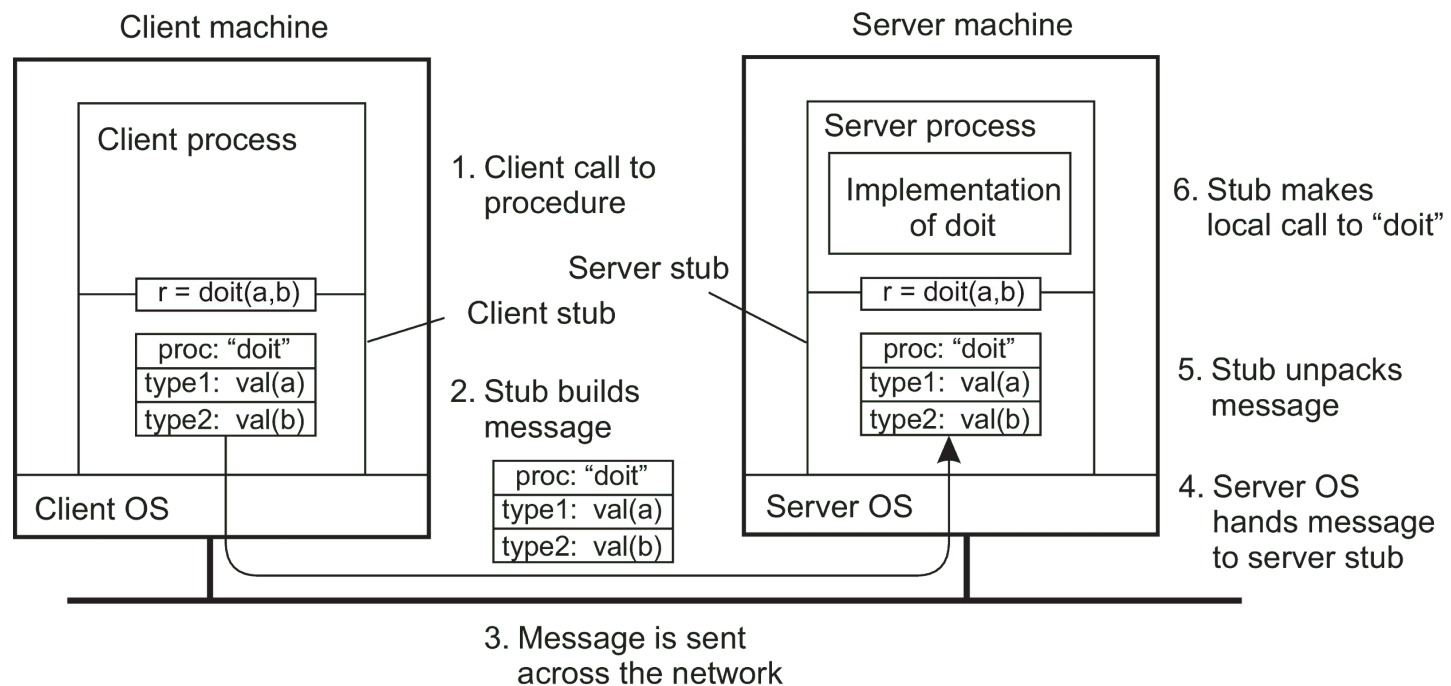
Remote Procedure Call (RPC)

- **Locally** call a procedure (aka a function) that is implemented on a **remote** machine.
- Parameters and results are transported but no message passing is visible to the programmer.
- Achieves **access** and **location** transparency.



The principle of RPC between a client and server program.

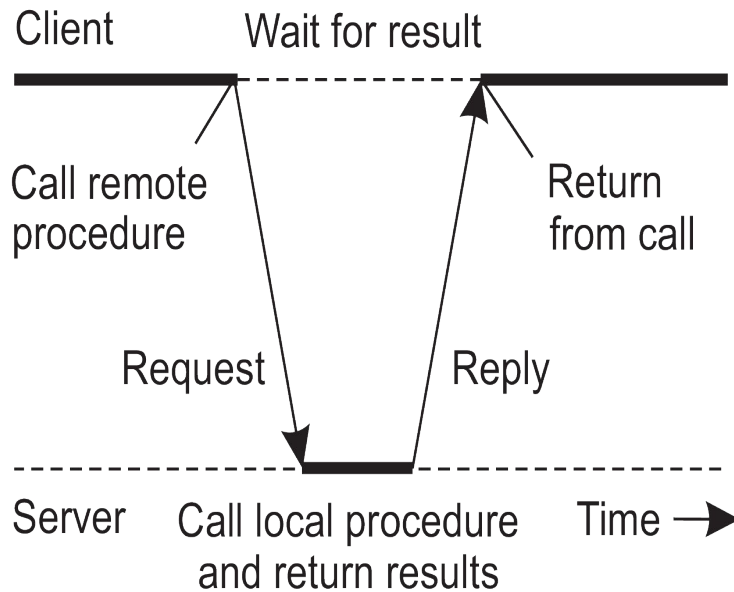
- Client calls directed to a **client stub** which provides the local procedure call interface to the remote function.
- A **server stub** transforms calls coming in over the network into local calls.



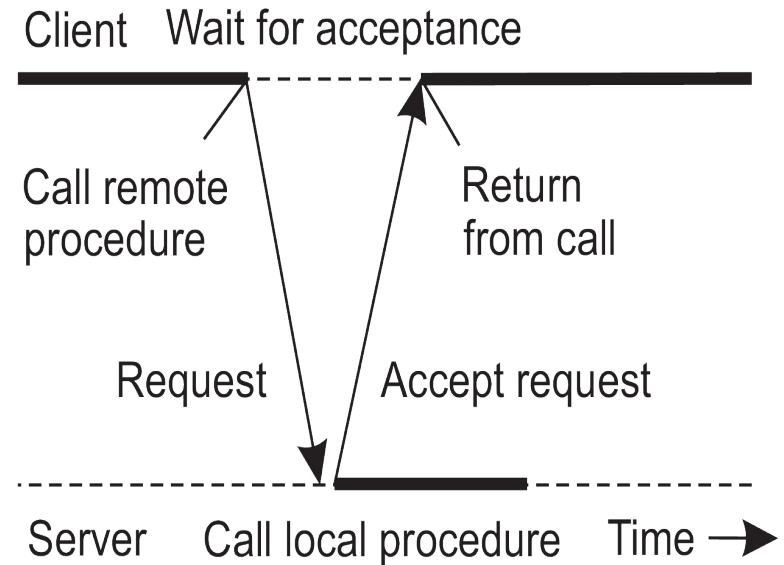
*The steps involved in calling a remote procedure `doit(a,b)`.
The return path for the result is not shown*

- RPC challenges include:
 - Handling different data representations (e.g. little endian vs. big endian).
 - Handling pointers or references which refer to local memory.

Asynchronous RPC

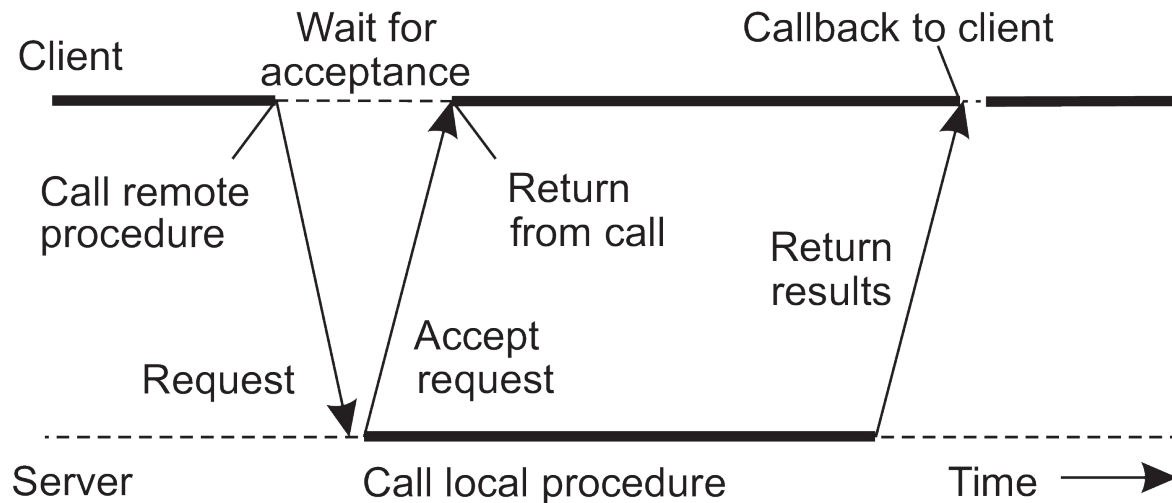


The interaction between client and server in a traditional RPC.



The interaction using asynchronous RPC.

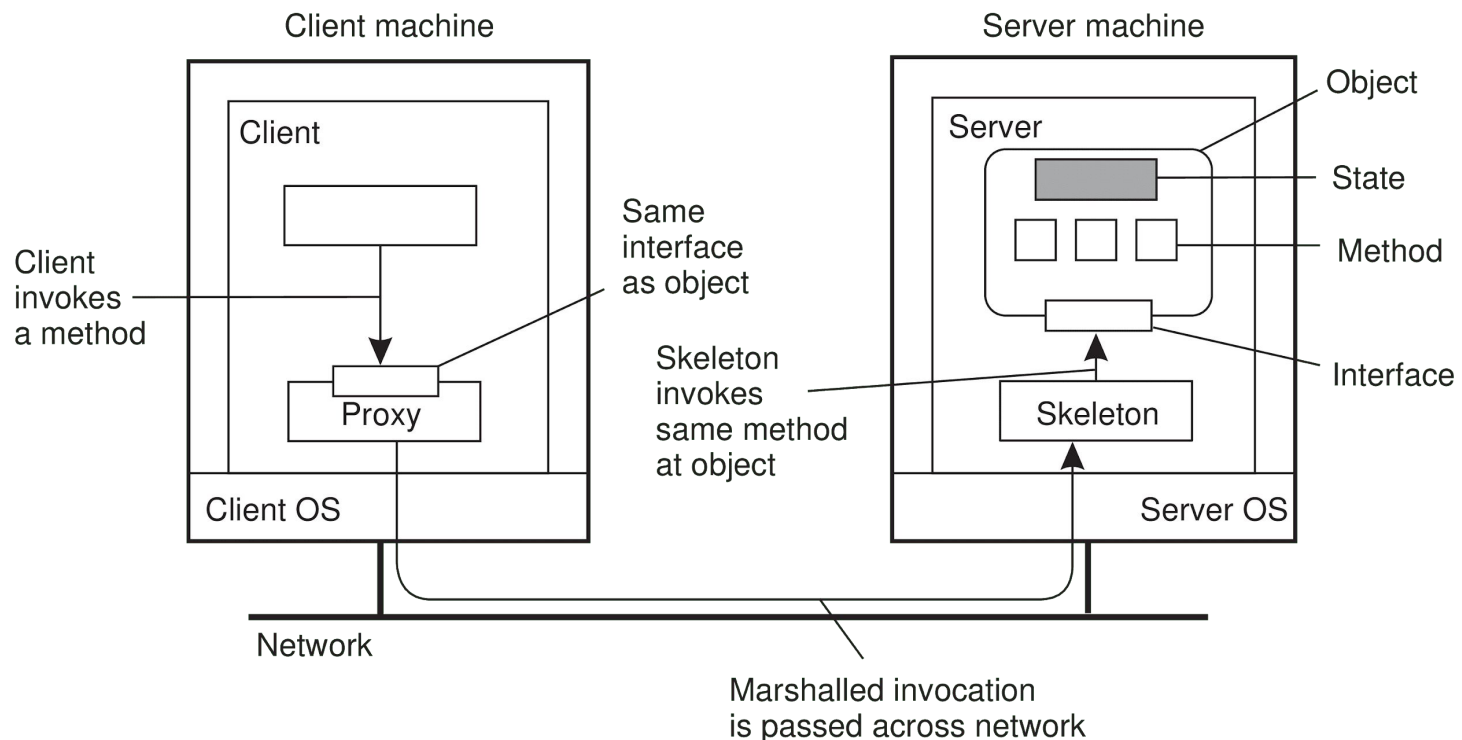
Asynchronous RPC combined with a callback



A client and server interacting through asynchronous RPCs.

Remote Method Invocation (RMI)

- RMI is the object-oriented equivalent of (RPC).



Common organisation of a remote object with client-side proxy.

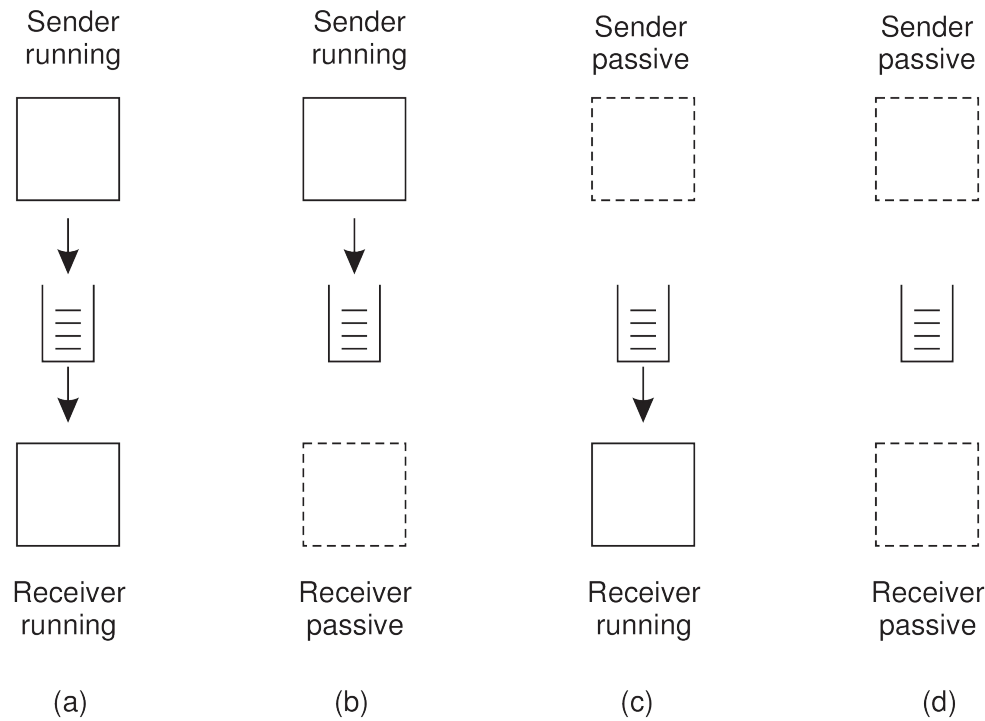
- The object resides on the server's machine and is known as a *remote object*.
- An object for which the instance of the data associated with it is distributed across machines is known as a *distributed object* (e.g. a distributed database).

Message Oriented Middleware (MOM)

- RPC and RMI require client and server to be running at time of communication (transient communication).
- RPC and RMI block the client until its request has been processed (synchronous communication).

- **Message Oriented Middleware** offers intermediate storage for messages, without requiring sender or receiver to be active during transmission.
- Applications communicate by inserting messages in specific queues.
- Sender is given only the guarantees that its message will eventually be inserted in the recipient's queue.
- No guarantees are given about when, or even if the message will actually be read.
- Achieves **persistent** and **asynchronous** communication.

- Four combinations with respect to the execution mode of the sender and receiver.



Four combinations with respect to the execution mode of the sender and receiver.

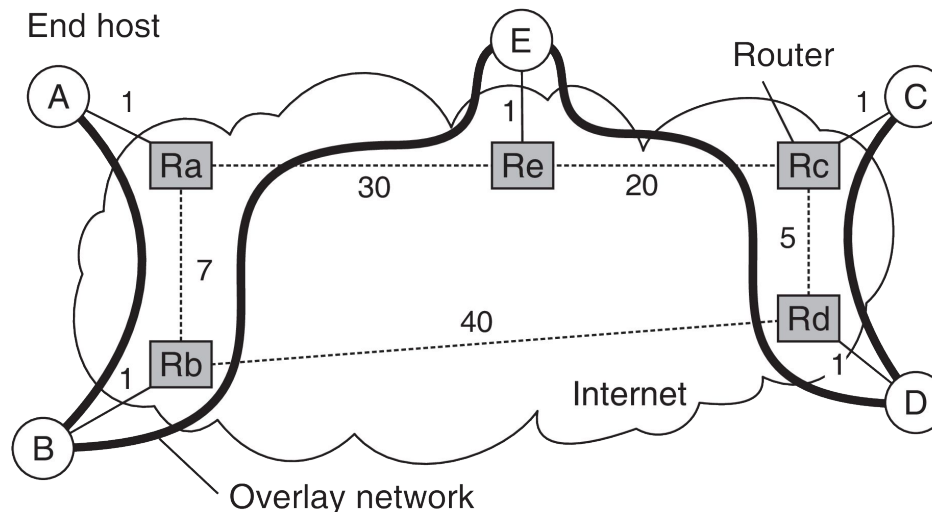
- *Advanced Message Queuing Protocol (AMQP)* is a protocol for MOM.

Multicast Communication

- Multicast communication concerns sending data to multiple receivers.
- Many specialized algorithms to perform this task including **application-level tree-based multicasting**.

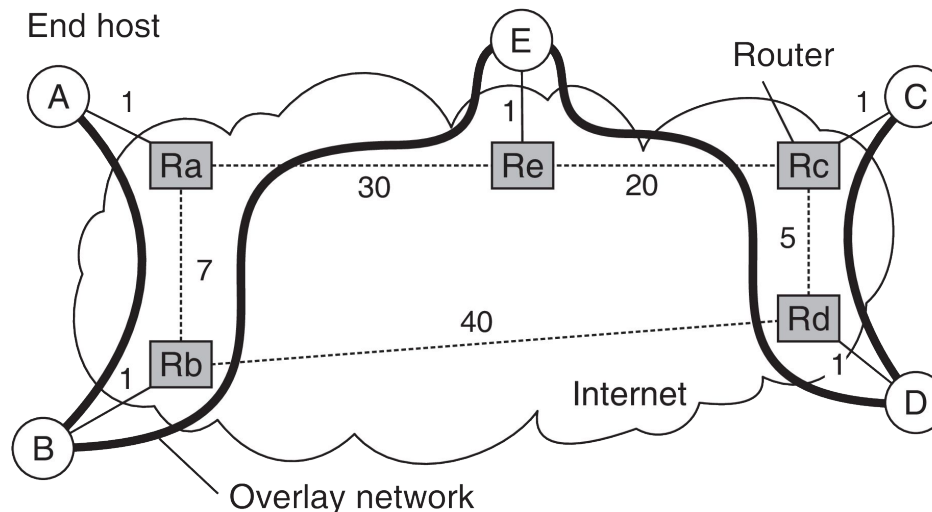
Application-level tree-based multicasting

- Nodes organize as overlay network with tree topology; unique (overlay) path between every pair of nodes.
- Each node forwards a message m to each of its neighbors, except to the one from which it received.



The relation between links in an overlay and actual network-level routes.

- Building an efficient tree is challenging.
- A message multicast by A will traverse the links (B, Rb), (Ra, Rb), (E, Re), (Rc, Rd), and (D, Rd) twice.



The relation between links in an overlay and actual network-level routes.