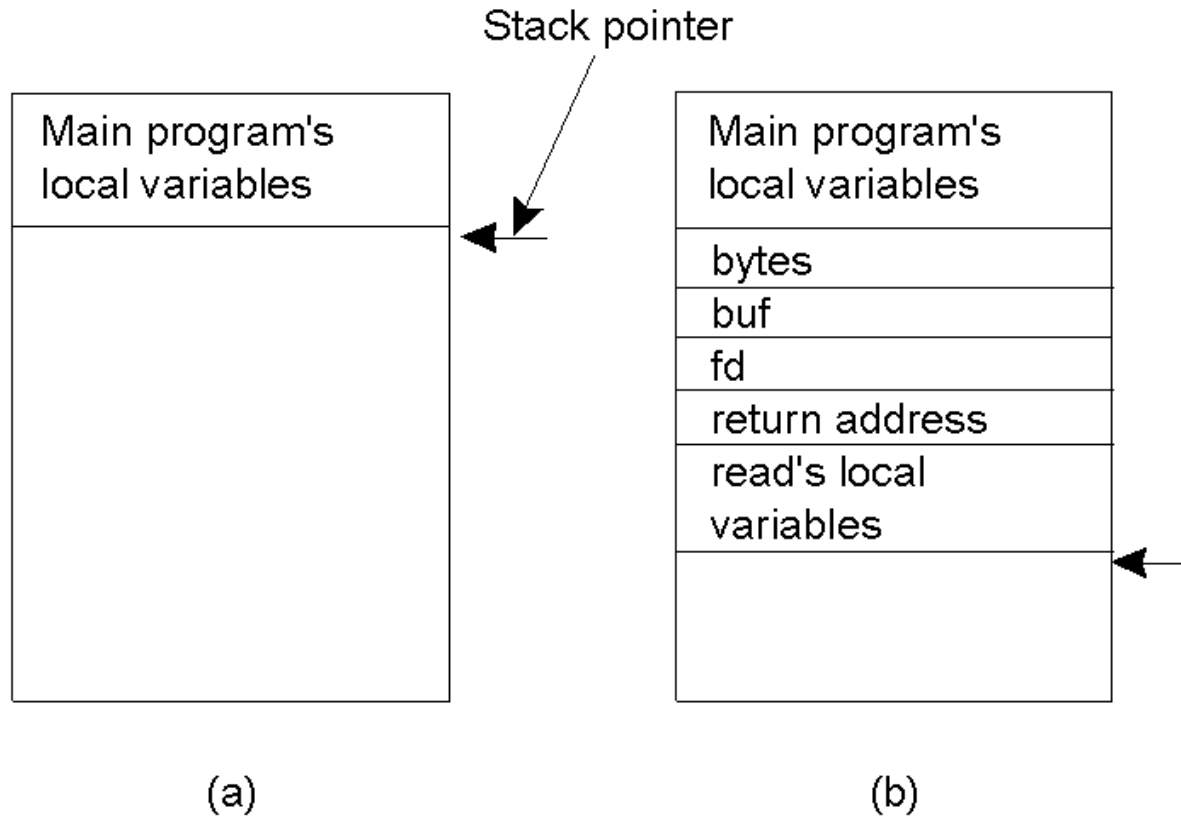


Communication

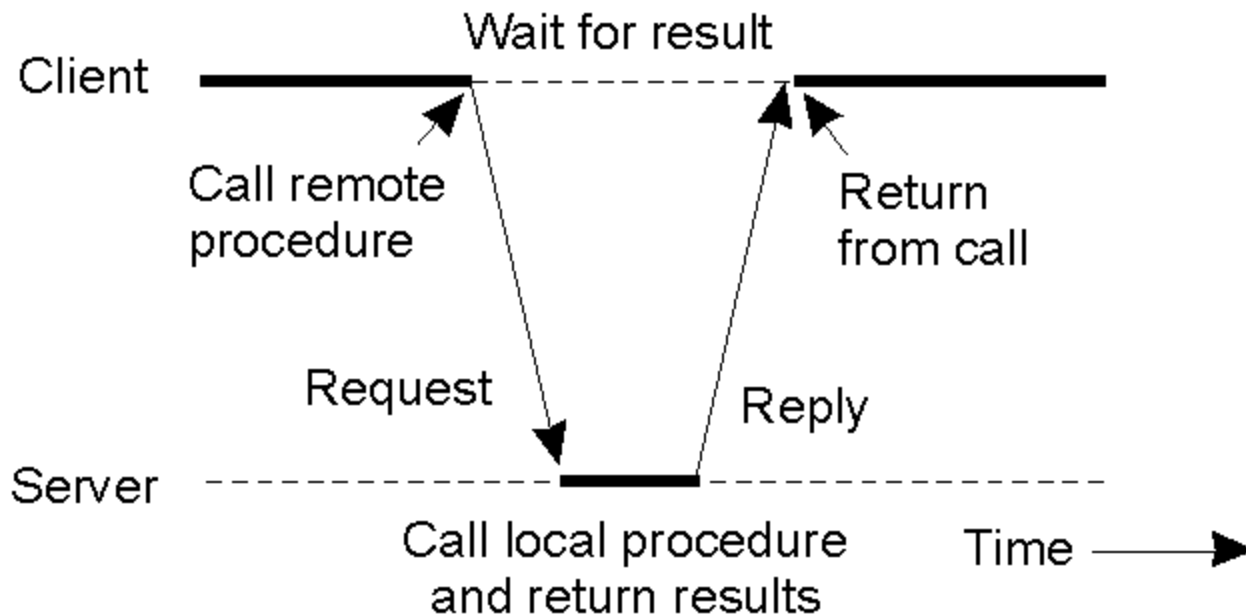
Chapter 2

Conventional Procedure Call



- a) Parameter passing in a local procedure call: the stack before the call to read
- b) The stack while the called procedure is active

Client and Server Stubs

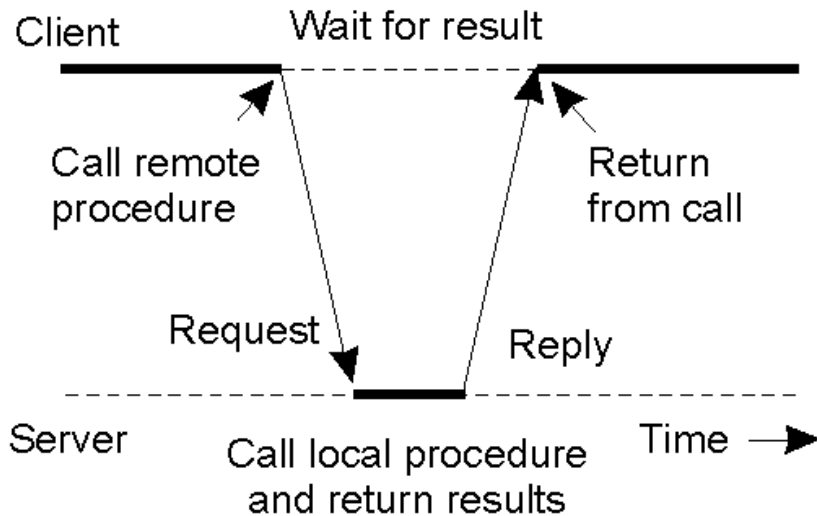


Principle of RPC between a client and server program.

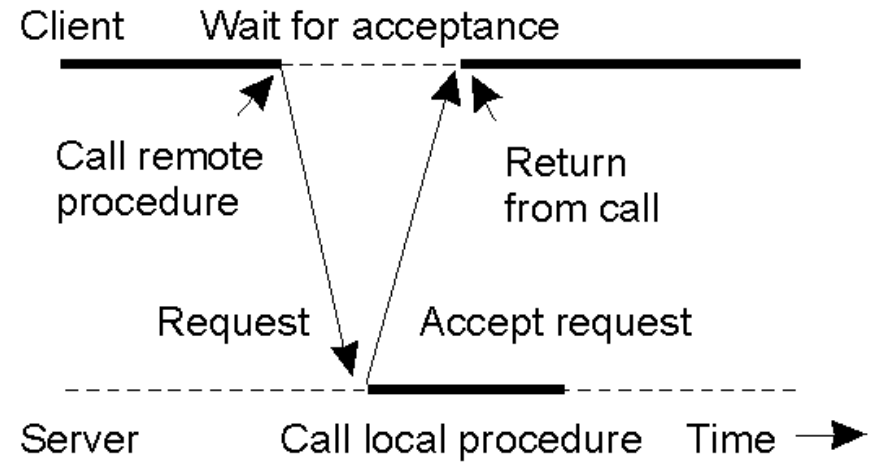
Steps of a Remote Procedure Call

1. Client procedure calls client stub in normal way
2. Client stub builds message, calls local OS
3. Client's OS sends message to remote OS
4. Remote OS gives message to server stub
5. Server stub unpacks parameters, calls server
6. Server does work, returns result to the stub
7. Server stub packs it in message, calls local OS
8. Server's OS sends message to client's OS
9. Client's OS gives message to client stub
10. Stub unpacks result, returns to client

Asynchronous RPC (1)



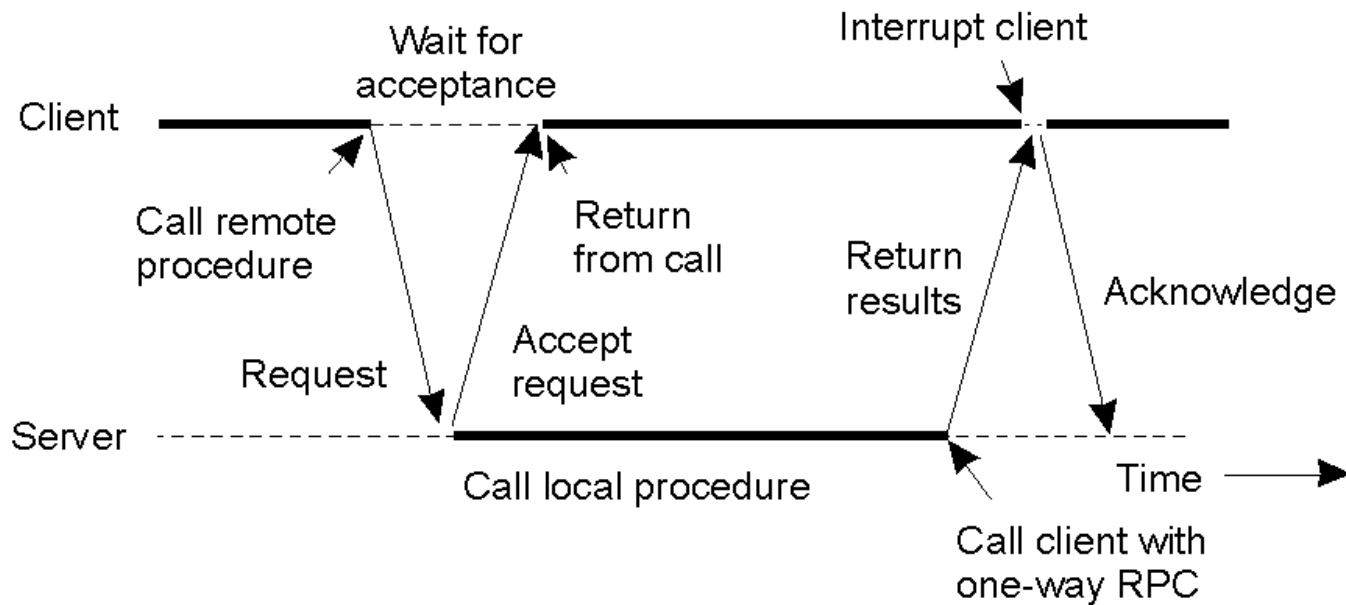
(a)



(b)

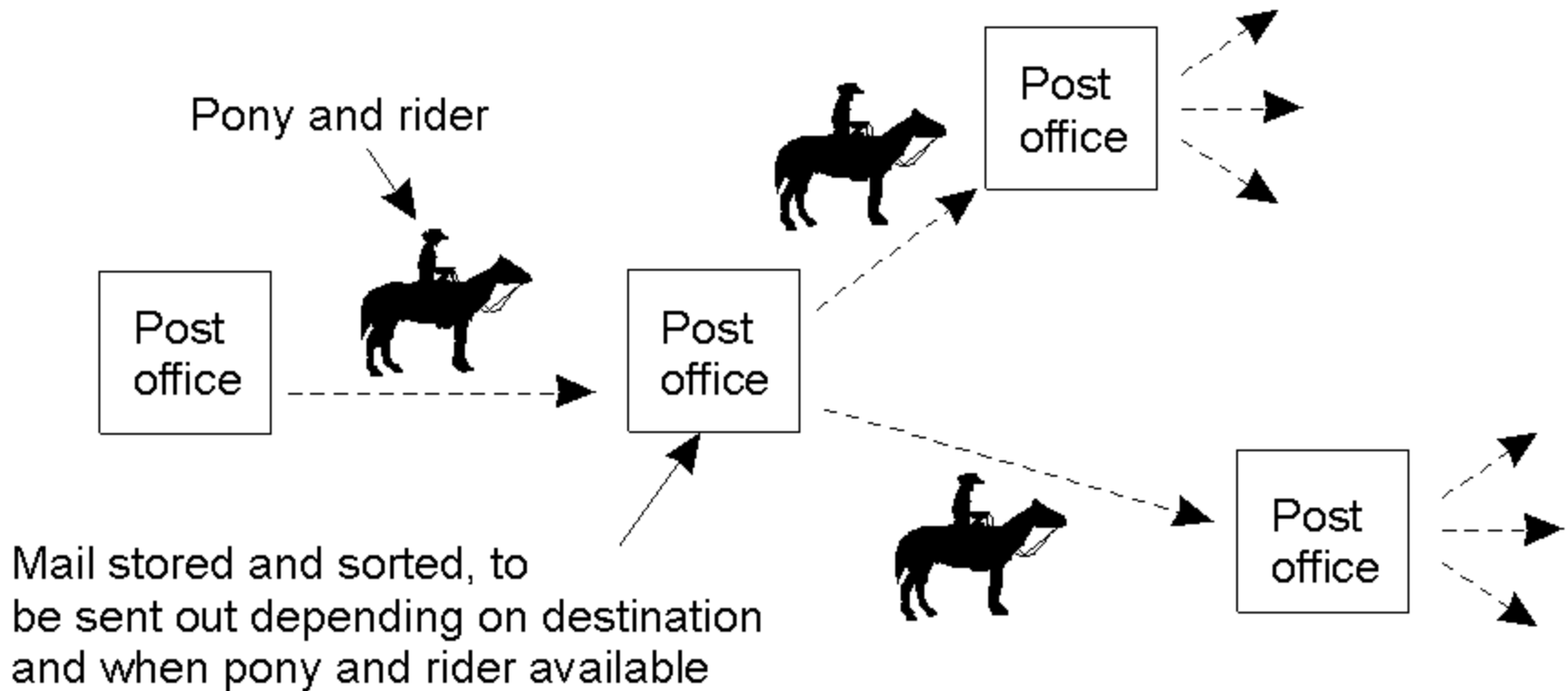
- a) The interconnection between client and server in a traditional RPC
- b) The interaction using asynchronous RPC

Asynchronous RPC (2)



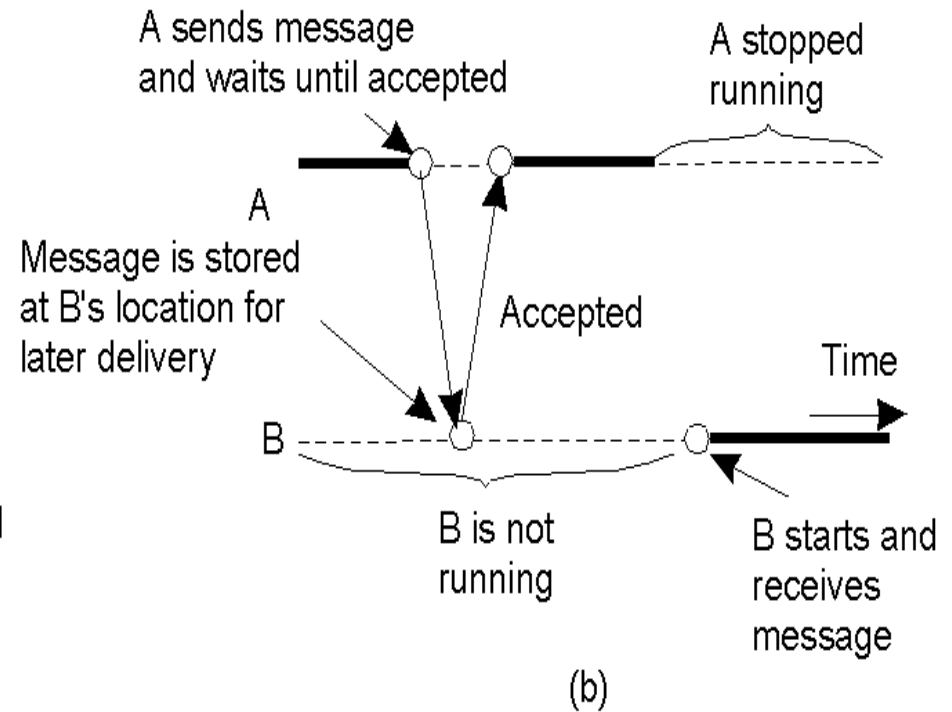
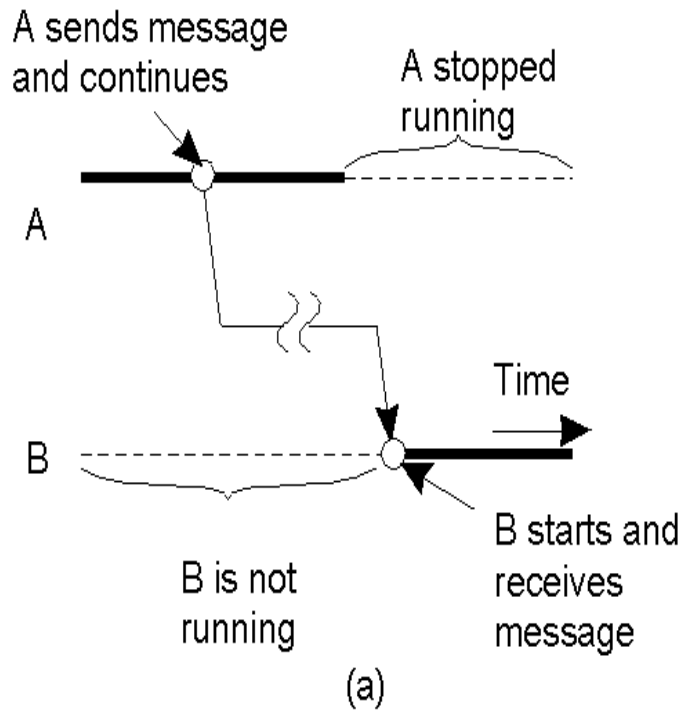
A client and server interacting through two asynchronous RPCs

Persistence and Synchronicity in Communication (2)



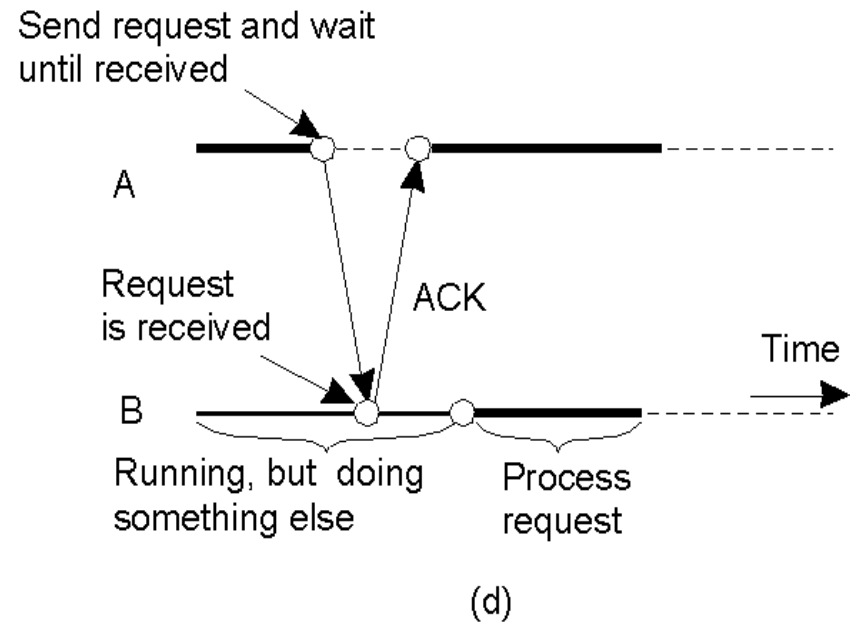
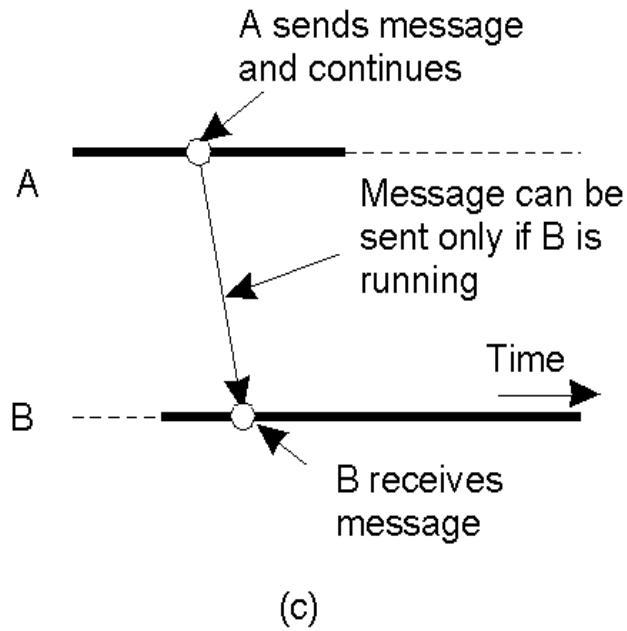
Persistent communication of letters back in the days of the Pony Express.

Persistence and Synchronicity in Communication (3)



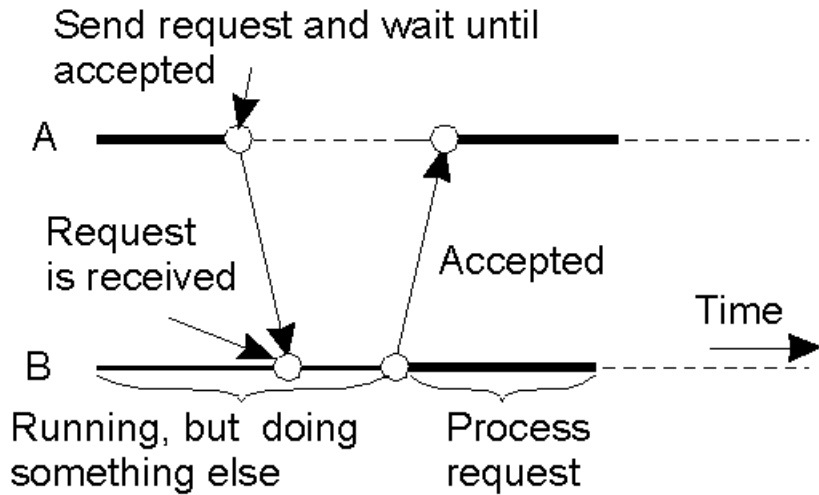
- a) Persistent asynchronous communication
- b) Persistent synchronous communication

Persistence and Synchronicity in Communication (4)

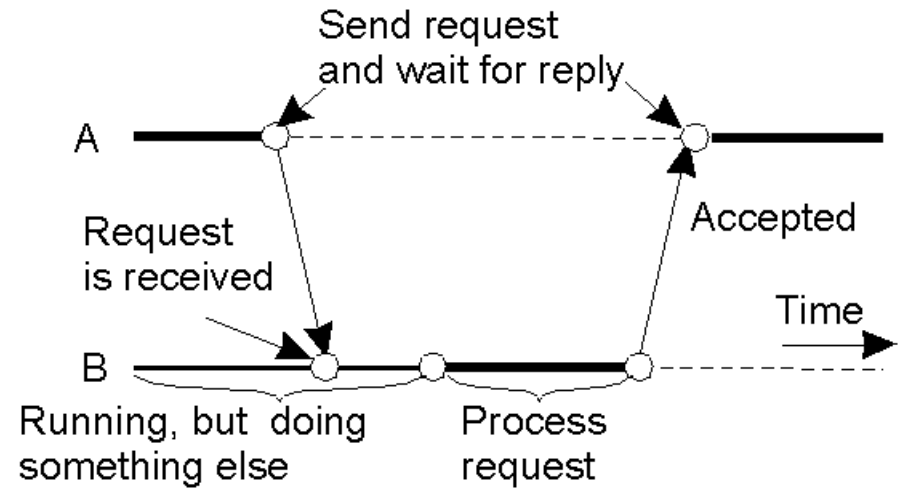


- c) Transient asynchronous communication
- d) Receipt-based transient synchronous communication

Persistence and Synchronicity in Communication (5)



(e)



(f)

- e) Delivery-based transient synchronous communication at message delivery
- f) Response-based transient synchronous communication