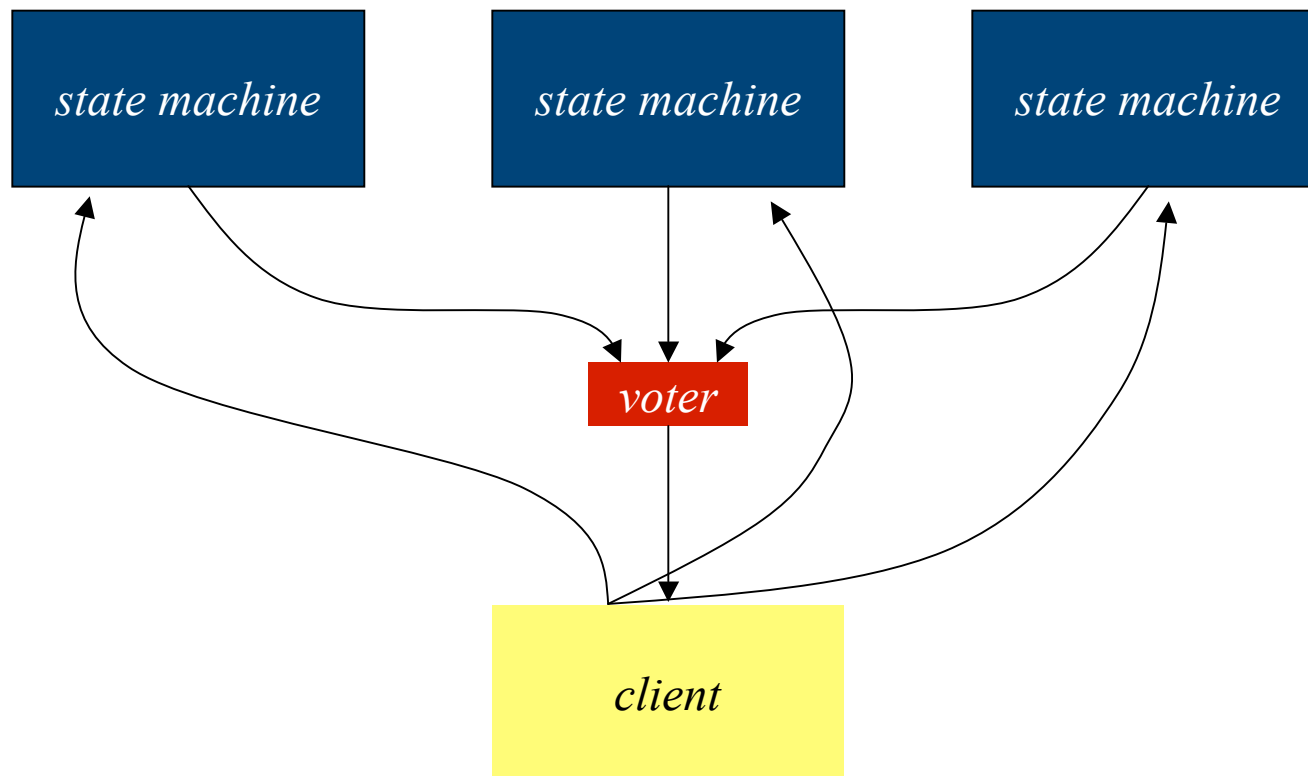


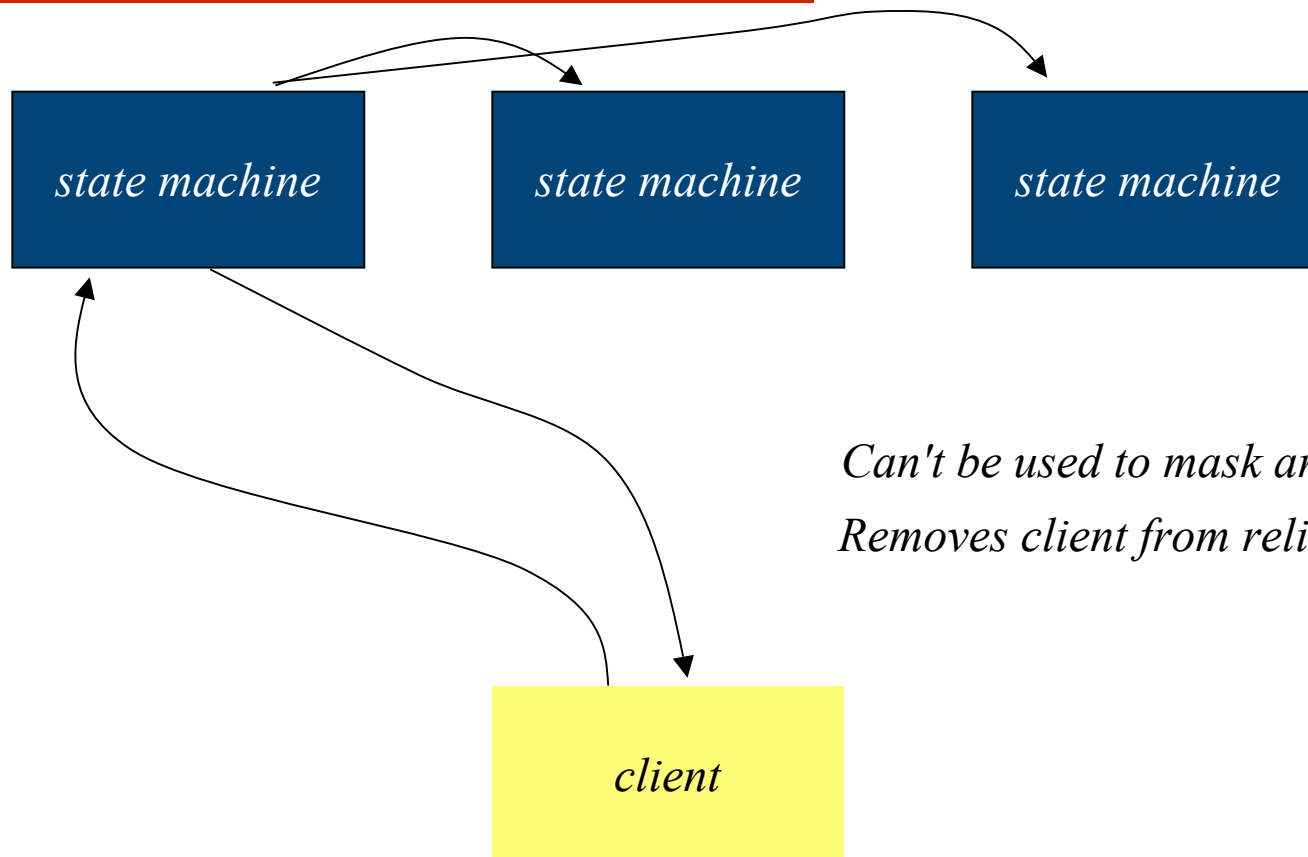
# Lowering the cost of state machines

---



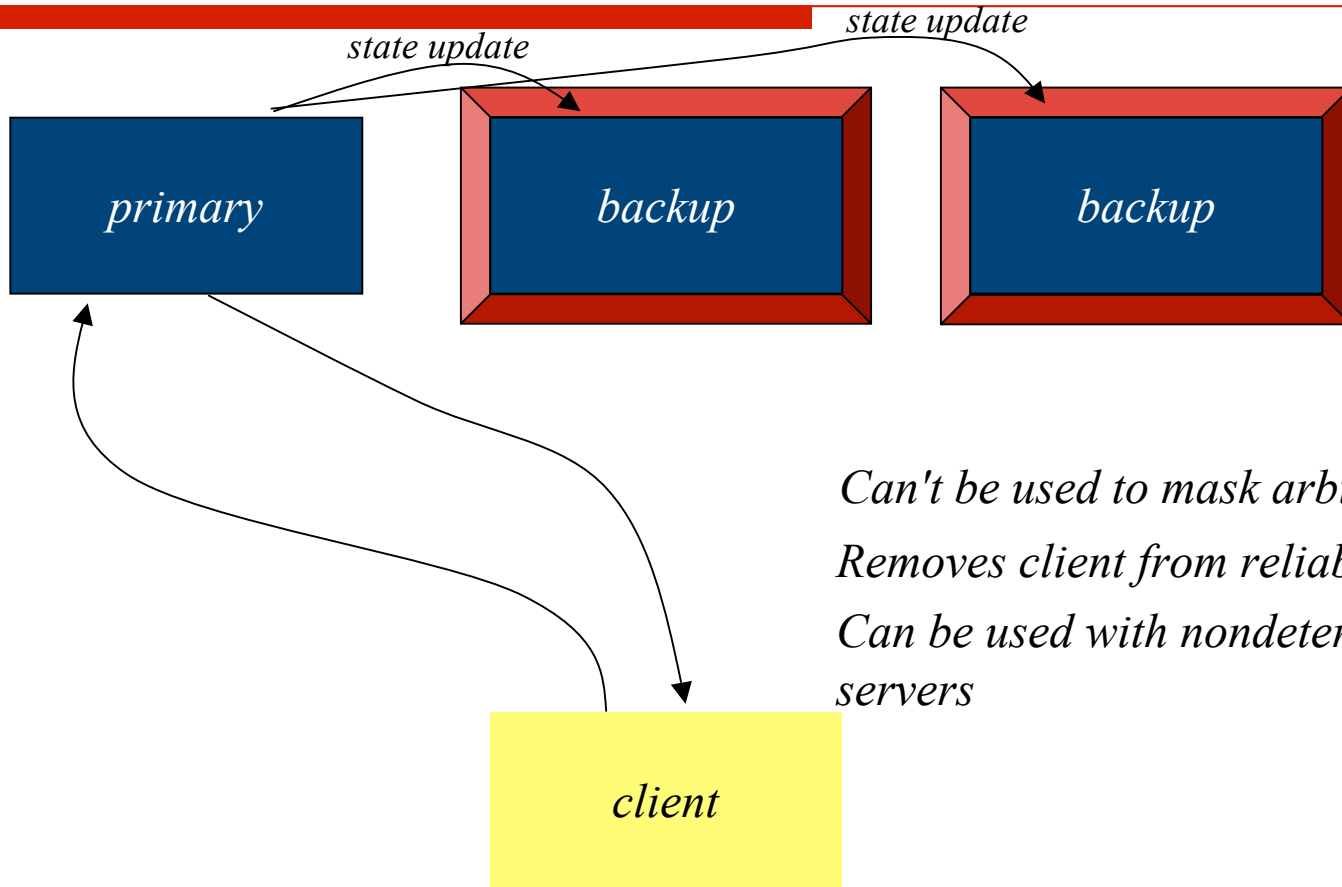
# Lowering the cost of state machines

---



*Can't be used to mask arbitrary failures  
Removes client from reliable broadcast*

# Lowering the cost of state machines



*Can't be used to mask arbitrary failures*  
*Removes client from reliable broadcast*  
*Can be used with nondeterministic servers*

# Some metrics for primary-backup

---

- *Degree of replication*: the number of servers needed to implement a  $t$ -fault-tolerant service.
- *Blocking time*: the worst-case interval between a request and its response in any failure-free execution.
- *Failover time*: the worst-case interval during which requests can be lost because there is no primary.

# Specification

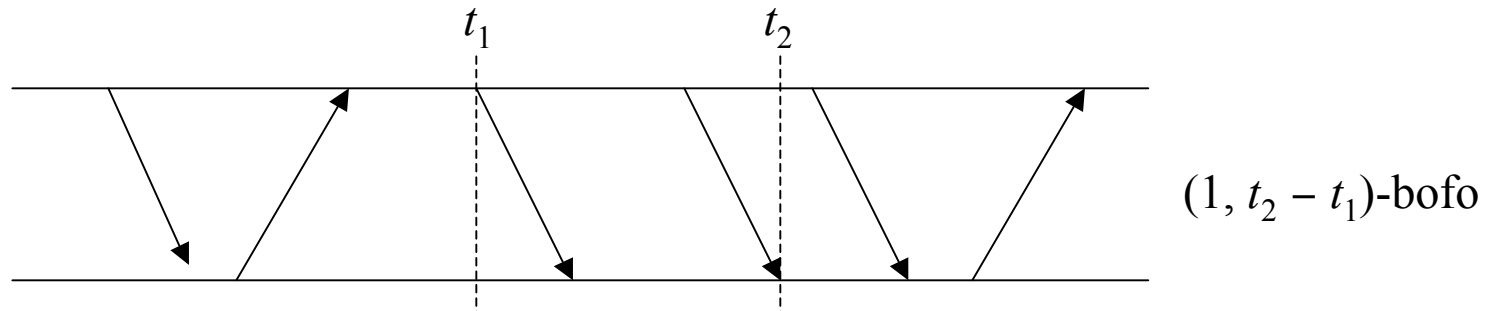
---

- PB1 There exists a local predicate  $Prmy_s$  on the state of each server. At any time, there is at most one server  $s$  whose state satisfies  $Prmy_s$ .
- PB2 Each client  $i$  maintains a server identity  $Dest_i$  such that to make a request, client  $i$  sends a message to  $Dest_i$ .
- PB3 If a client request arrives at a server  $s$  whose state does not satisfy  $Prmy_s$ , then the request is not enqueued (and therefore not processed).

# Specification (continued)

---

$(k, \Delta)$ -bofo server: all server outages can be grouped into at most  $k$  intervals of time, each having length of no more than  $\Delta$ .



**PB4** There exists fixed values  $k$  and  $\Delta$  such that the service behaves like a single  $(k, \Delta)$ -bofo server.

# Specification (continued)

---

We assume that clients are unreliable enough that they cannot help in detecting failures and notifying other clients about failovers.



# Lower bounds on replication

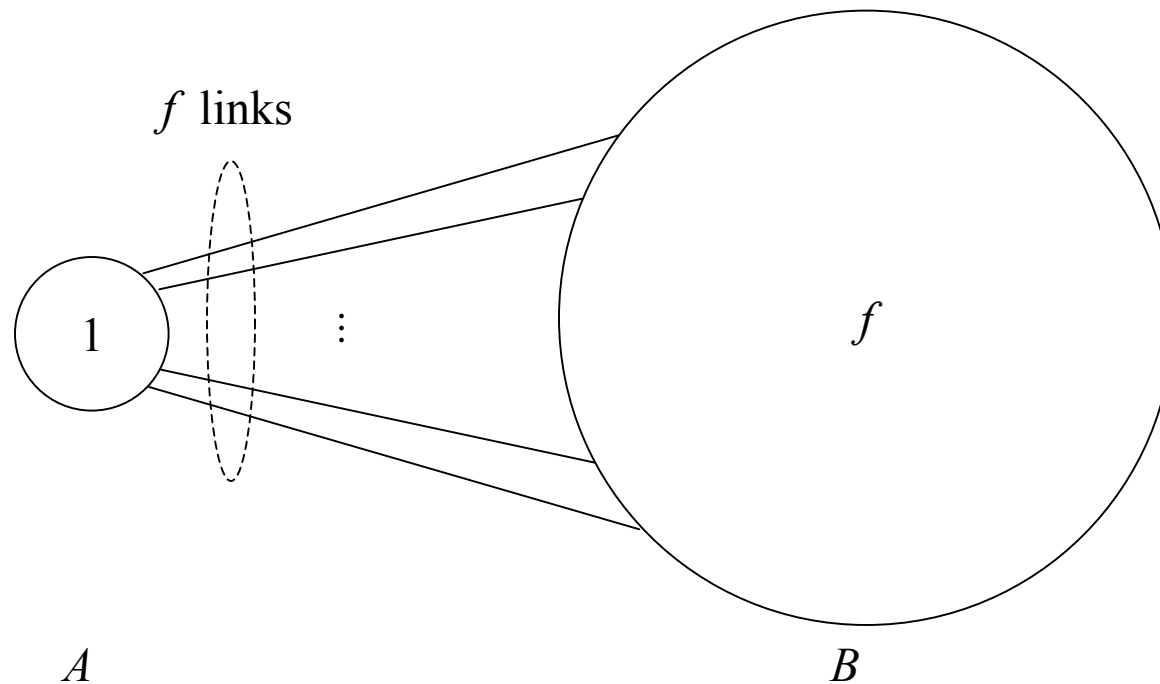
---

Both crash and send omission require only  $n > f$ .

# Lower bounds on replication: Crash+Link

---

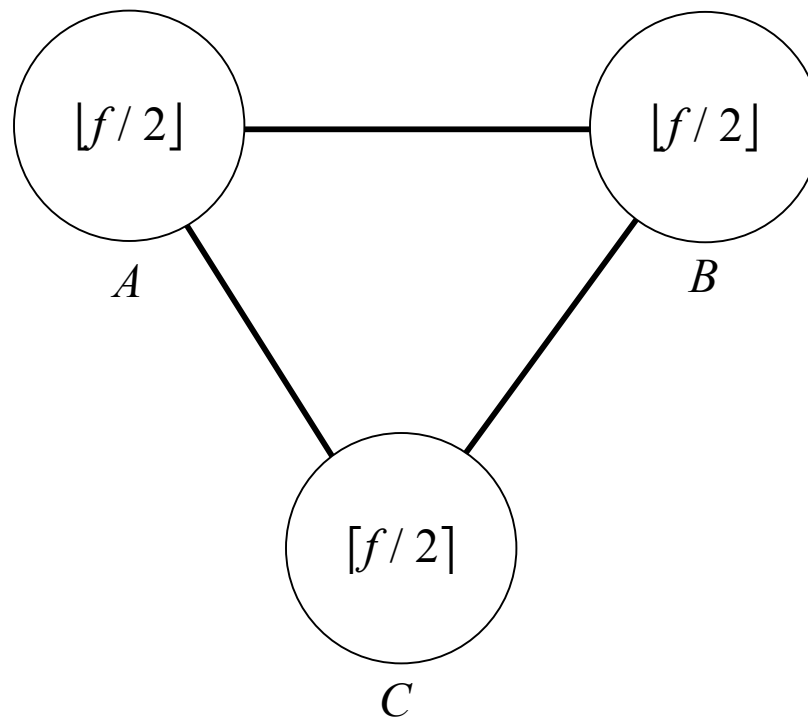
$$n > f + 1$$



# Lower bounds on replication: Receive Omission

---

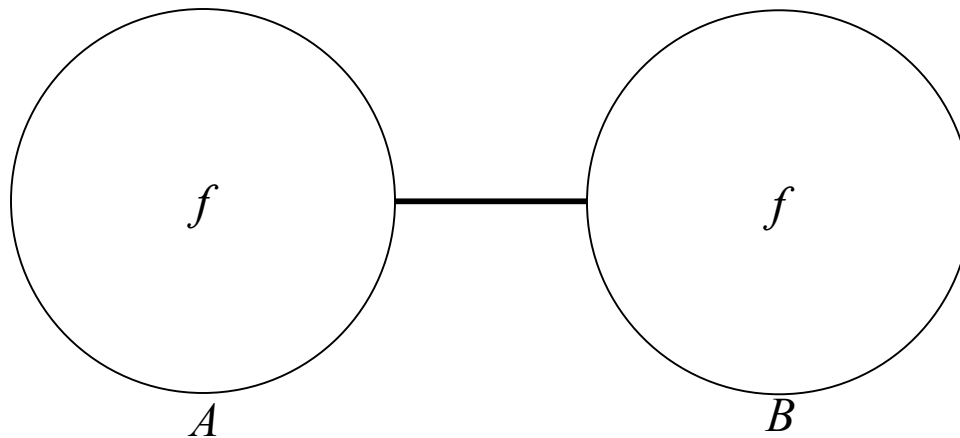
$$n > \lfloor 3f/2 \rfloor$$



# Lower bounds on replication: General Omission

---

$$n > 2f$$



# Lower bounds on blocking time

---

crash	0
crash + link	0
receive omission	$\delta$ when $f = 1$ and $n = 2$ $2\delta$ when $f > 1$ and $n \leq 2f$ 0 when $n > 2f$
send omission	$\delta$ when $f = 1$ $2\delta$ when $f > 1$
general omission	$\delta$ when $f = 1$ $2\delta$ when $f > 1$

# Lower bounds on failover time

---

crash	$f \delta$
crash + link	$2f \delta$
receive omission	$2f \delta$
send omission	$2f \delta$
general omission	$2f \delta$

Requires PB5: A correct server that is the primary remains so until there is a failure of some server or link.

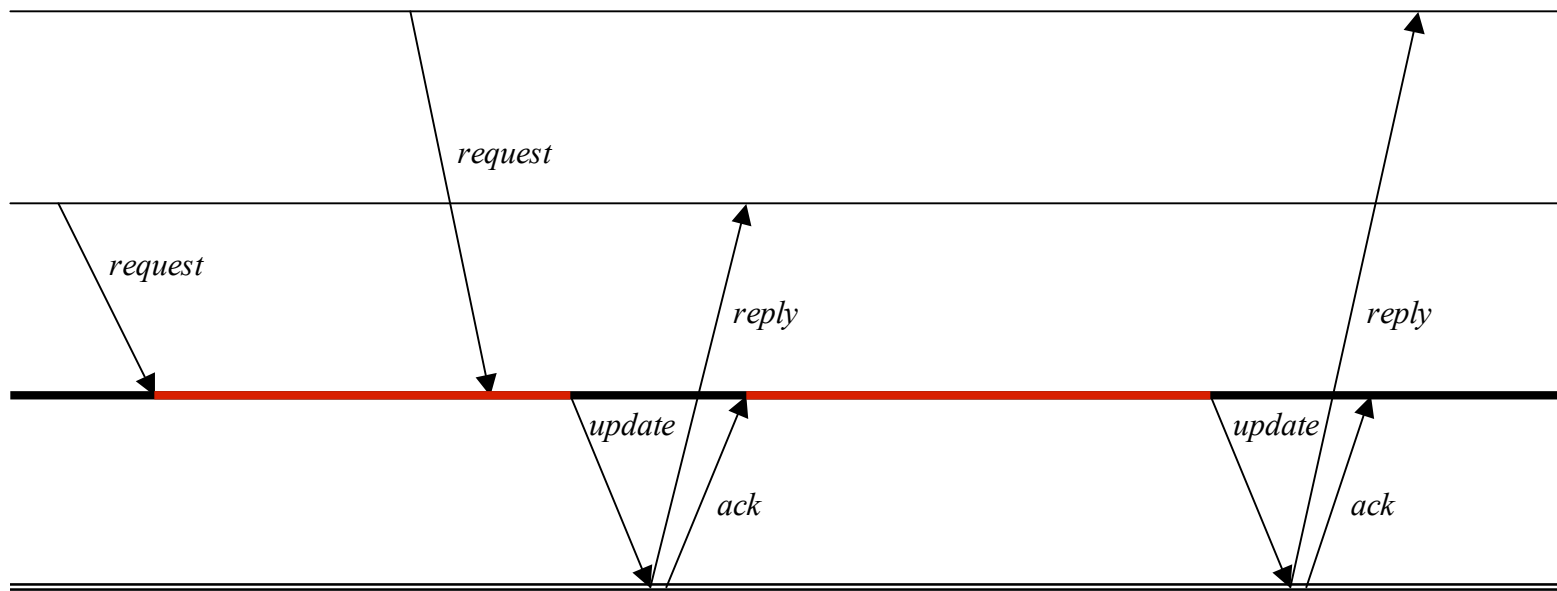
# Alsberg/Day protocol (1976): single crash

---

- If request arrives at primary then
  - Primary executes request
  - Primary sends state update message, and waits for ack.
  - Backup processes state update message, replies to client, and sends ack to primary.
- If request arrives at backup then
  - Backup forwards to primary
  - Primary executes request, sends reply to client, and sends state update message to backup.
  - Backup discards request when gets state update message.
- Missing acks and heartbeats used to detect failure.

# Alsberg/Day

---



*blocking time*  $\delta > 0$  (backup sends reply)

*failover time*  $\tau + 2\delta > \tau + \delta$   
(no synchronized clocks)

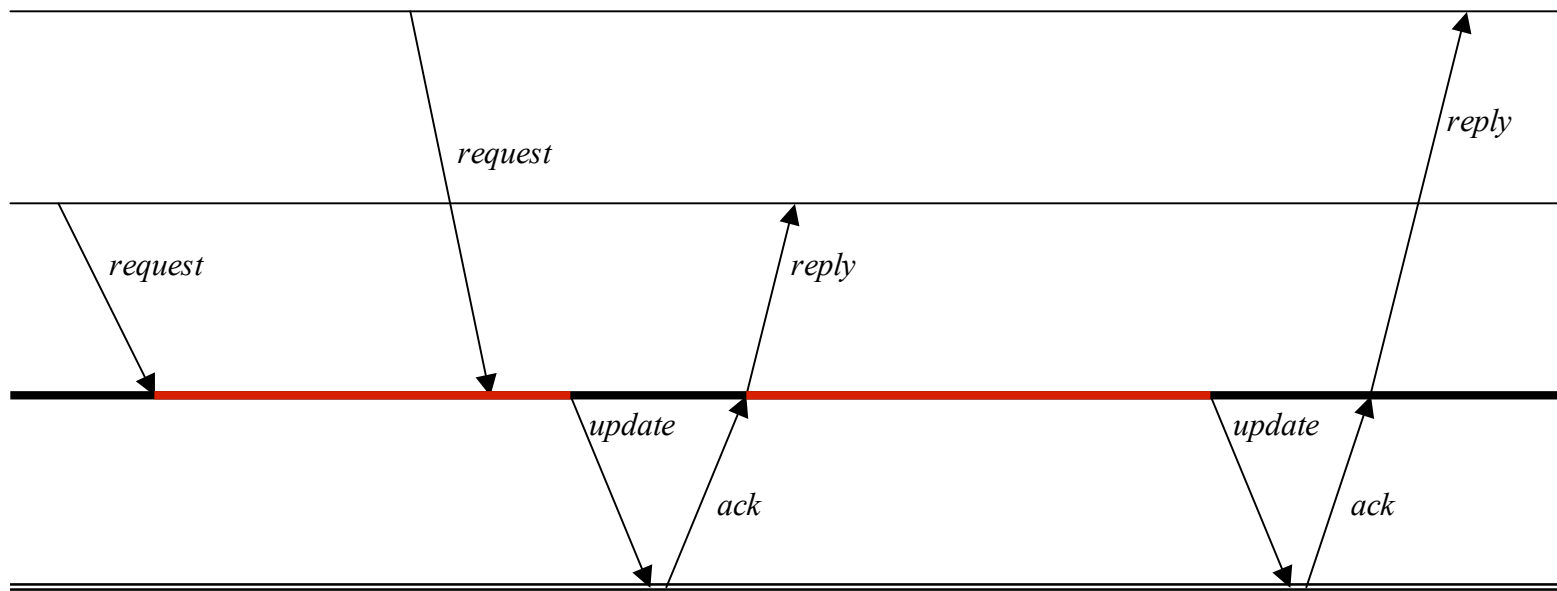
# Tandem (1981): single crash+link

---

- Based on *process pairs*, which replicate a process on two processors that are connected by two links (broadcast busses).
- Primary receives request:
  - Primary executes request, sends state update message on one link, to backup, and waits for ack.
  - If no ack by a deadline, sends state update message on other link.
- Heartbeats used to detect crash.

# Tandem

---



*blocking time*  $2\delta > 0$

*failover time*  $\tau + 2\delta > \tau + \delta$   
(no synchronized clocks)

# HA-NFS (1991): single crash+link

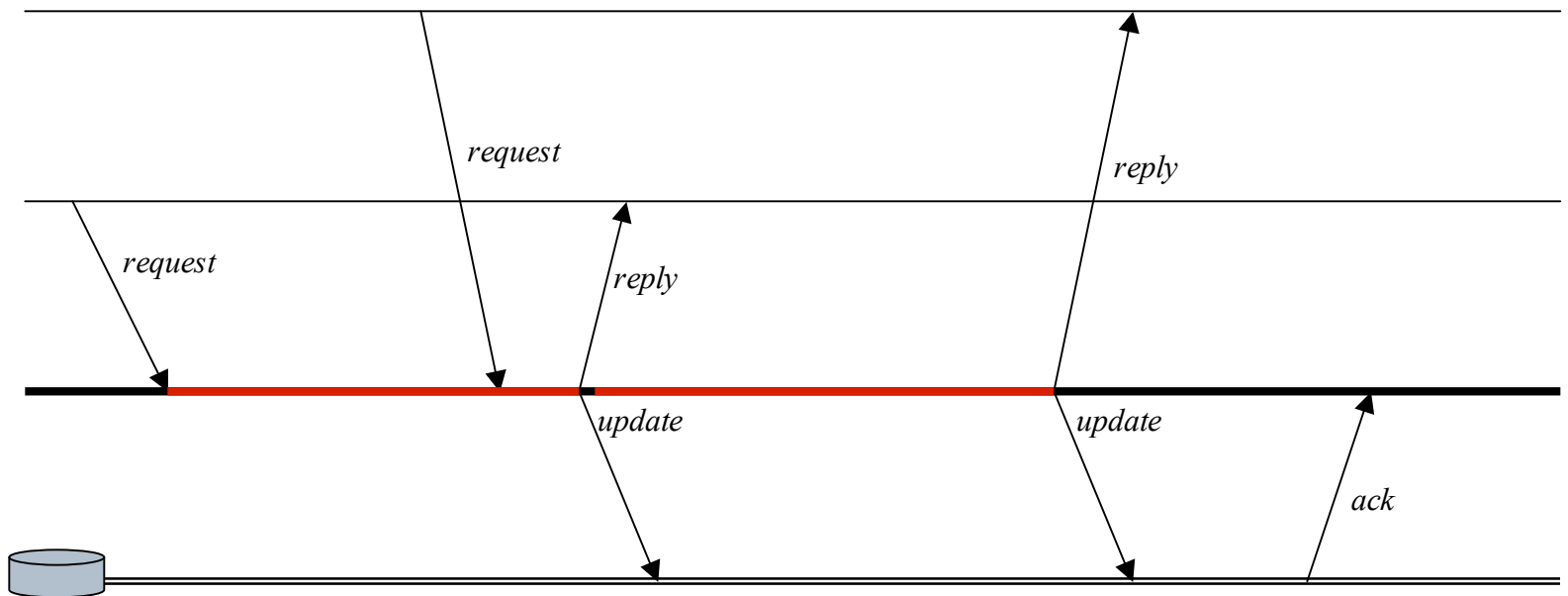
---

Primary and backup connected by a link and a dual-ported disk.

- Primary executes requests and updates disk and heartbeats to the backup.
- When backup stops receiving heartbeats, it attempts to communicate with primary via disk. If not possible, then it takes over as primary.

# HA-NFS

---



*blocking time 0*

*failover time  $\tau + 2\delta > \tau + \delta$   
(no synchronized clocks)*

# Nonblocking primary-backup: receive omission (experimental protocol)

---

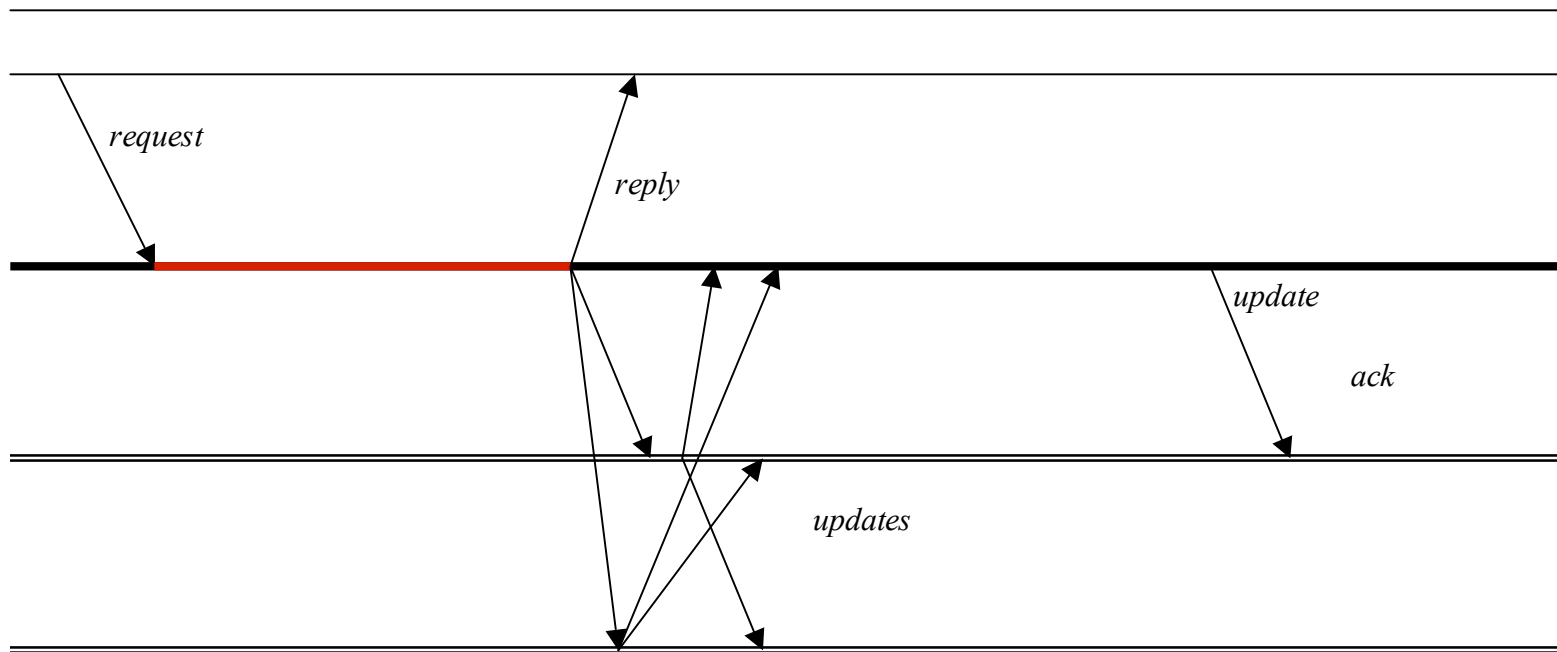
$2f + 1$  servers.

- ❑ Primary executes request, sends state updates to backups, and immediately replies to client.
- ❑ Backup applies state updates.
- ❑ Failures detected using a scheme whereby a faulty process will detect its own receive omission failure.

... when implemented, failure detection created so much contention that it limited response time.

# Non-blocking RO

---



*blocking time*  $\delta > 0$  (backup sends reply)

*failover time*  $\tau + 2\delta > \tau + \delta$   
(no synchronized clocks)