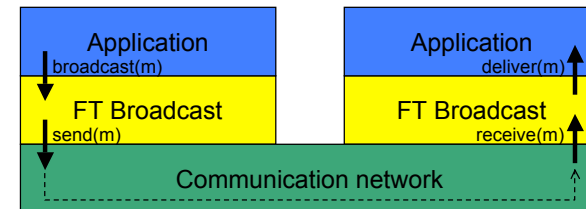


Fault-tolerant broadcasts

Part I: Definitions

Fault-tolerant broadcasts

- Specification
 - broadcast(m)
 - deliver(m)



© Fernando Pedone

Fault-tolerant broadcasts

- Reliable broadcast
 - Validity: if a correct process broadcasts m , then all correct processes eventually deliver m .
 - Agreement: if a correct process delivers m , then all correct processes eventually deliver m .
 - Integrity: for any message m , every correct process delivers m at most once, and only if m was previously broadcast by sender(m).

© Fernando Pedone

Fault-tolerant broadcasts

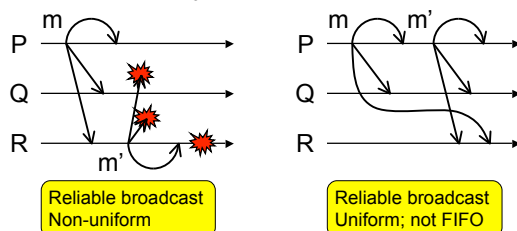
- Uniform reliable broadcast
 - Validity: if a correct process broadcasts m , then all correct processes eventually deliver m .
 - Uniform agreement: if a process delivers m , then all correct processes eventually deliver m .
 - Uniform integrity: for any message m , every process delivers m at most once, and only if m was previously broadcast by sender(m).

© Fernando Pedone

Fault-tolerant broadcasts

FIFO broadcast

- FIFO order: if a process broadcasts m before m' , then no correct process delivers m' unless it has previously delivered m .

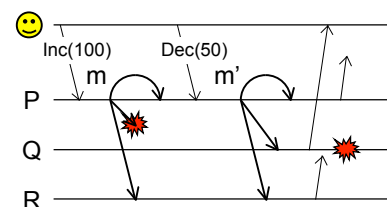


© Fernando Pedone

Fault-tolerant broadcasts

FIFO broadcast (cont'd)

- Ambiguous FIFO: if a process broadcasts m before m' , then no correct process delivers m' before m .



© Fernando Pedone

Fault-tolerant broadcasts

Causal broadcast

- Example

- In a network news application, if users distribute their articles with FIFO broadcast, the following could happen: User A broadcasts an article. User B at a different site delivers that article and broadcasts a response to A's article. User C receives B's response before receiving A's article.

© Fernando Pedone

Fault-tolerant broadcasts

Causal broadcast (the happens-before relation, Lamport '78)

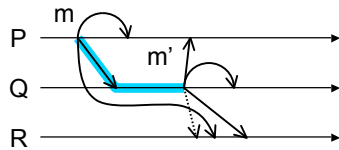
- An execution of a broadcast or deliver primitive by a process is called an **event**. We say that event e **causally precedes** f , $e \rightarrow f$, iff
 - (a) a process executes e and f in that order, or
 - (b) e is m 's broadcast and f is m 's delivery, or
 - (c) there is some h such that $e \rightarrow h$ and $h \rightarrow f$.

© Fernando Pedone

Fault-tolerant broadcasts

■ Causal broadcast

- Causal order: if the broadcast of a message m causally precedes the broadcast of a message m' , then no correct process delivers m' unless it has previously delivered m .



© Fernando Pedone

Fault-tolerant broadcasts

■ Uniform FIFO broadcast

- Uniform FIFO order: if a process broadcasts m before m' , then no process delivers m' unless it has previously delivered m .

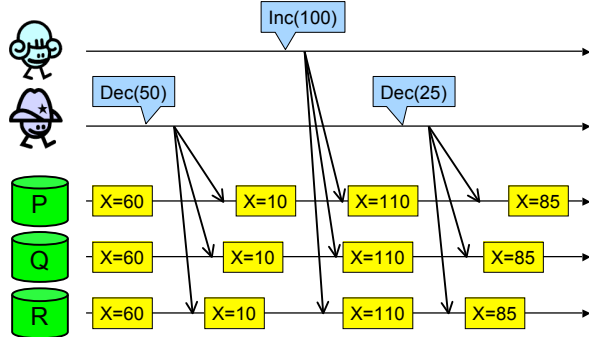
■ Uniform causal broadcast

- Uniform causal order: if the broadcast of a message m causally precedes the broadcast of a message m' , then no process delivers m' unless it has previously delivered m .

© Fernando Pedone

Fault-tolerant broadcasts

■ Atomic broadcast

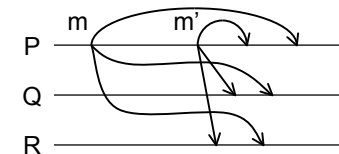


© Fernando Pedone

Fault-tolerant broadcasts

■ Atomic broadcast

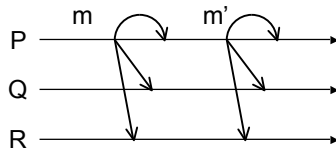
- Total order: if correct processes p and q both deliver messages m and m' , then p delivers m before m' if and only if q delivers m before m' .



© Fernando Pedone

Fault-tolerant broadcasts

- FIFO atomic broadcast
 - FIFO + total order: ...



© Fernando Pedone

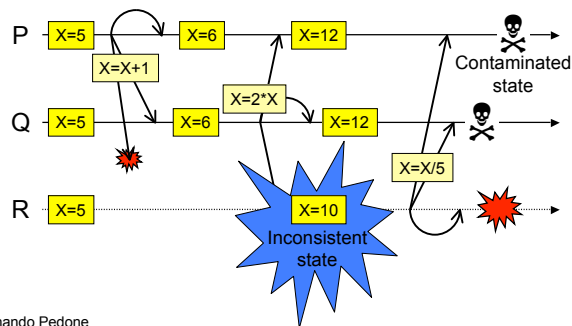
Fault-tolerant broadcasts

- Causal atomic broadcast
 - Causal + total order: ...
 - FIFO abcast doesn't ensure causal order
 - Reconsider the network news example, implemented with FIFO atomic broadcast. Faulty user A broadcasts an article; faulty user B is the only one to deliver it, broadcasts a response, and fails before delivering its own message. Correct users C delivers the response although it never delivers the original message.

© Fernando Pedone

Fault-tolerant broadcasts

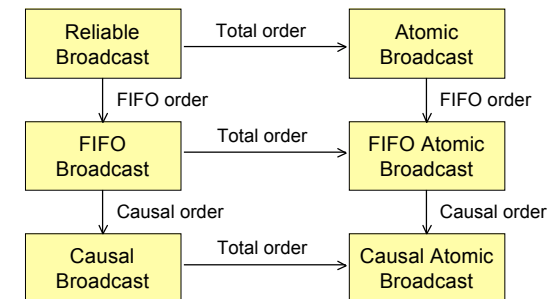
- Inconsistency and contamination



© Fernando Pedone

Fault-tolerant broadcasts

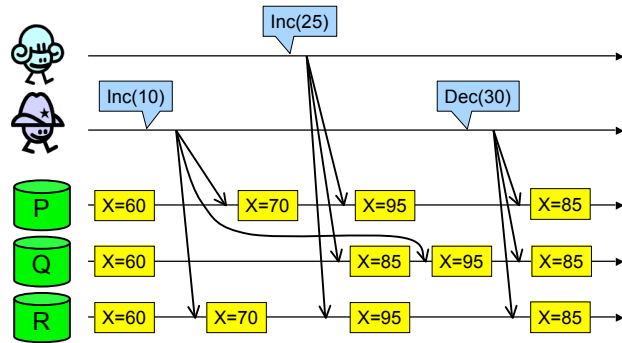
- Relationship among broadcast primitives



© Fernando Pedone

Fault-tolerant broadcasts

Generic broadcast



© Fernando Pedone

Fault-tolerant broadcasts

Generic broadcast

- Conflict relation ~
 - Messages m and m' may conflict (i.e., $m \sim m'$)
- Order: if correct processes p and q both deliver messages m and m' , and $m \sim m'$, then p delivers m before m' if and only if q delivers m before m' .

© Fernando Pedone

Fault-tolerant broadcasts

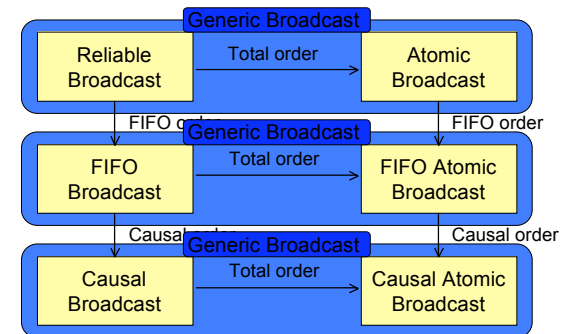
Generic broadcast

- Cheaper and faster delivery
- Generalizes order guarantees
- Reliable broadcast
 - Empty conflict relation
- Atomic broadcast
 - Cartesian product: $M \times M$

© Fernando Pedone

Fault-tolerant broadcasts

Relationship among broadcast primitives



© Fernando Pedone