CSC 2221: Introduction to Distributed Computing Class Notes
University of Toronto

Specification of Terminating Reliable Broadcast (TRB)

A distinguished process s ("the sender") intends to broadcast some message $m \in \mathcal{M}$. Processes know that s intends to broadcast a message, and they can deliver a message in $\mathcal{M} \cup \{\text{SF}\}$ (intuitively, "SF" means "Sender is Faulty"). TRB satisfies:

- *Validity:* If the sender s is correct and broadcasts a message m, then it eventually delivers m.
- *Agreement:* If a correct process delivers a message m, then all correct processes eventually deliver m.
- *(Uniform) Integrity:* Every process delivers at most one message, and it delivers a message $m \neq SF$ only if m was previously broadcast by the sender s.
- *Termination:* Every correct process eventually delivers a message.

Specification of Consensus

Every process is supposed to *propose* some value (from a universe $\mathcal{U}$ of possible values), and each process must *decide* on a value such that the following properties hold:

- *Agreement:* No two correct processes decide differently.
- *Validity:* If a correct process decides v, then v was proposed by some process.
- *Termination:* Every correct process decides exactly one value.

1. Using TRB to Solve Consensus:

Process i ($1 \leq i \leq n$):

Initially:

$D[1 \ldots n] := \langle \perp, \perp, \ldots, \perp \rangle$

To **propose** v_i:

broadcast v_i [using copy i of TRB]

To **decide**:

while ($\exists k : D[k] = \perp$) ("array is not full")

if **deliver** some v_j [from copy j of TRB]

then $D[j] := v_j$

decide first non-SF value in array D.

---

2. Using Consensus to Solve TRB:

The following reduction assumes:

- synchronous rounds;
- crash and send-omission failures only;
- no link failures.

Sender s in round 1:

send m to all; **deliver** m ; halt

Every receiver $p \neq s$ in rounds > 1:

Run a t-tolerant Consensus algorithm by **proposing**
the message that p received in round 1
(or SF if p did not receive any message)

if p **decides** v, then it **delivers** v.
