

Early-Stopping TRB Algorithm for General-Omission Failures

Sender s in round 1:

send m to all; deliver m ; halt

Every receiver $p \neq s$ in round i , $1 \leq i \leq t + 1$:

$quiet^0 := \Phi$

do

if delivered some message m (could be SF) in round $i - 1$ **then** send m to all ; halt
 else send ? to all

 [receive round i messages]

$quiet^i := quiet^{i-1} \cup \{q \mid p \text{ received no round } i \text{ message from } q \}$

if received some message $m \neq ?$ in round i **then** deliver m

else if $|quiet^i| < i$ **then** deliver SF

od

if did not deliver any message yet **then** deliver SF ; halt

Lemma (Validity): If the sender is correct and broadcasts m , then all correct processes deliver m .

Proof: If the sender is correct and broadcasts m , it sends m to all and delivers m in round 1. Thus, all other correct processes receive and deliver m by the end of round 1.

Lemma (Termination): All correct processes deliver some message by round $f + 1$, and halt by round $\min(f + 2, t + 1)$.

Proof: (Sketch) If $f = 0$, the sender is correct. In this case, all correct processes deliver the sender's message in round 1 and halt by round 2. Now assume that $1 \leq f \leq t$. There are two cases:

1. Some correct process p delivers a message by the end of round f . In this case, p sends that message to all by round $f + 1$. Thus, all correct processes decide by the end of round $f + 1$, and halt by the end of round $\min(f + 2, t + 1)$.
2. No correct process delivers a message by the end of round f . Thus, no correct process halts before the end of round $f + 1$. So, in all rounds i , $1 \leq i \leq f + 1$, all correct processes send a message to all. Consider any correct process p at the end of round $f + 1$. Since all correct processes send a message to all in rounds $1, 2, \dots, f + 1$, p 's set $quiet_p^{f+1}$ can only contain *faulty* processes. So, $|quiet_p^{f+1}| < f + 1$ at the end of round $f + 1$. By the algorithm, p delivers a message in round $f + 1$, and then halts by round $\min(f + 2, t + 1)$.

Lemma (Uniform Integrity): Every process delivers at most one message, and delivers $m \neq SF$ only if m was previously broadcast by the sender.

Proof: It is clear from the structure of the algorithm that no process delivers more than once. Since only benign failures may occur and the send/receive primitive satisfies Uniform Integrity, a simple induction shows that if a process delivers $m \neq SF$ then the sender sends m to some process in the first round.

Lemma: If some correct process delivers $m \neq SF$, then some correct process delivers m before round $t + 1$.

Proof: Suppose some process delivers $m \neq SF$ in round $t + 1$. This implies the existence of $t + 1$ distinct processes $\{p_1, \dots, p_{t+1}\}$, such that for $1 \leq i \leq t + 1$, p_i sends m in round i . One of them, say p_k , must be correct. If $k = 1$, then p_k delivered m in round 1. If $2 \leq k \leq t + 1$, then p_k delivered m in round $k - 1$. In both cases, p_k delivered m before round $t + 1$.

Lemma: If some correct process delivers $m \neq SF$, then no correct process ever delivers SF .

Proof: Let p be the first correct process to deliver $m \neq SF$. By the previous lemma, p delivers m in some round i , $i < t + 1$. This implies the existence of i distinct processes $\{p_1, \dots, p_i\}$, such that for $1 \leq k \leq i$, p_k sends m in round k .

Claim: No process (whether correct or faulty) delivers SF before or in round i .

Proof of Claim: By contradiction. Suppose the first process to deliver SF , say process q , does so by round $j \leq i$. In rounds $1 \leq k \leq j$, q did not receive m from p_k . Thus, at the end of round j , $quiet_q^j$ includes processes $\{p_1, \dots, p_j\}$, and so $|quiet_q^j| \geq j$. Since q is the *first* process to deliver SF , it did so because $|quiet_q^j| < j$ — a contradiction.

Since p delivers m in round $i < t + 1$, it sends m to all in round $i + 1$. Furthermore, from the above claim, no process sends SF in round $i + 1$. Thus, in round $i + 1$, any correct process that has not yet delivered a message receives and delivers m . Together with the above claim, this implies that no correct process ever delivers SF .

Lemma (Agreement): If some correct process delivers a message m (possibly SF), then every correct process eventually delivers m .

Proof: By Termination and Uniform Integrity, all correct processes deliver exactly one message. By Uniform Integrity, this message is either SF or the sender's message $m \neq SF$. Agreement now follows by the previous lemma.