Switching On.

How Processes Initialize for Consistent Broadcast

Josef Widder
widder@auto.tuwien.ac.at

October 9, 2002

Overview

- The Partially Synchronous System Model
- Desired Service: Round Synchronization
- Consistent Broadcasting
- Problems of Initialization
- Modified Consistent Broadcasting
- A Solution for Initialization

System Model

A system of N distributed processes with a maximum of f faulty ones.

Transmission Delay. δ is the end-to-end computational + transmission delay of a message sent between two correct processes.

Bounds on the Transmission Delay.
$0 < \tau^- \leq \delta \leq \tau^+ < \infty$. The bounds τ^- and τ^+ are not known in advance.

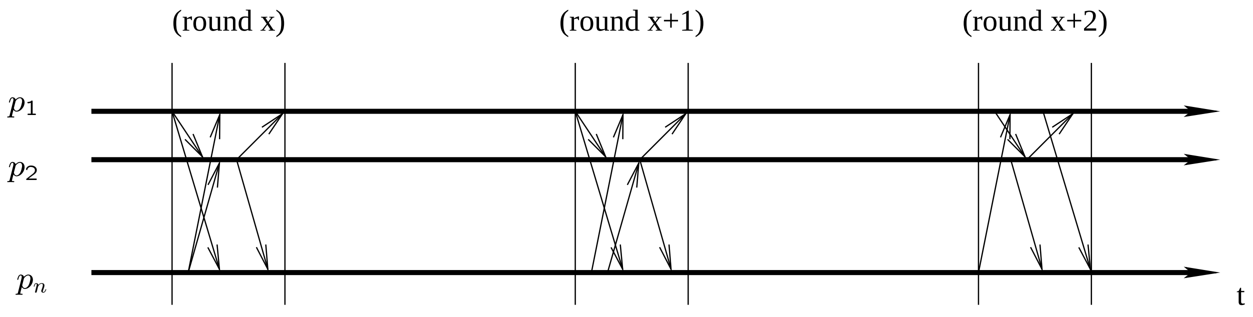
Transmission Uncertainty. $\epsilon = \tau^+ - \tau^-$.

Transmission Delay Ratio. $\mathcal{P} = \frac{\tau^+}{\tau^-}$.

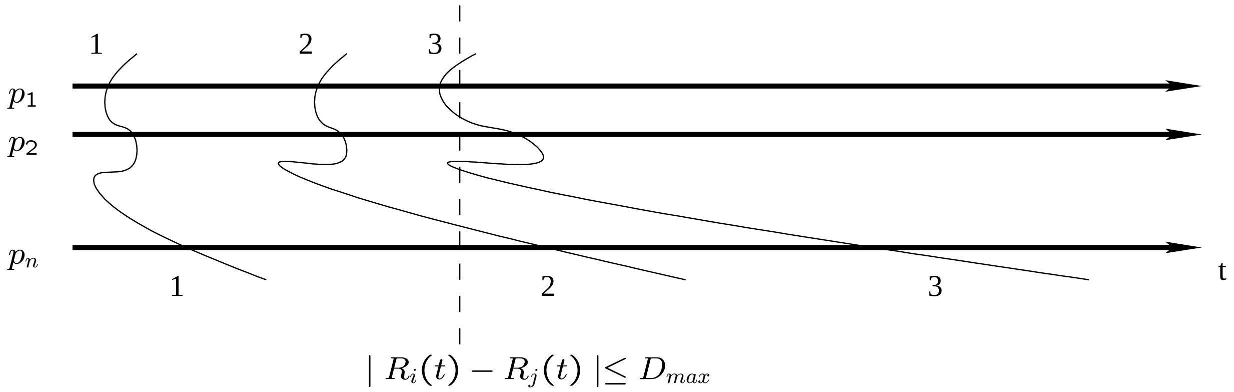
Processes boot one by one. No assumption on the maximum difference of booting times (**timer free initialization**).

Messages that reach a process before it is booted are lost.

Consistent Broadcast for Round Synchronization



Consecutive CB



Properties of round synchronization using CB

- P1. *Correctness.* If at least $f+1$ correct processes broadcast *(round k)* messages by time t, then every correct process accepts the message by time $t+\tau_{rt}$, where $\tau_{rt} = 2\tau^{+}$.

- P2. *Unforgeability.* If no correct process broadcasts a *(round k)* message by time t, then no correct process accepts *round k* by $t+2\tau^{-}$ or earlier.

- P3. *Relay.* If a correct process accepts the message *(round k)* at time t, then every correct process does so by time $t+\tau_{\Delta}$, where $\tau_{\Delta} = \epsilon + \tau^{+}$.

CB Primitive
by Srinkath & Toueg 1987

if received *(init, round k)* from $f+1$ p's
  $\rightarrow$ send *(echo, round k)* to all;
fi

if received *(echo, round k)* from $f+1$ p's
  $\rightarrow$ send *(echo, round k)* to all;
fi

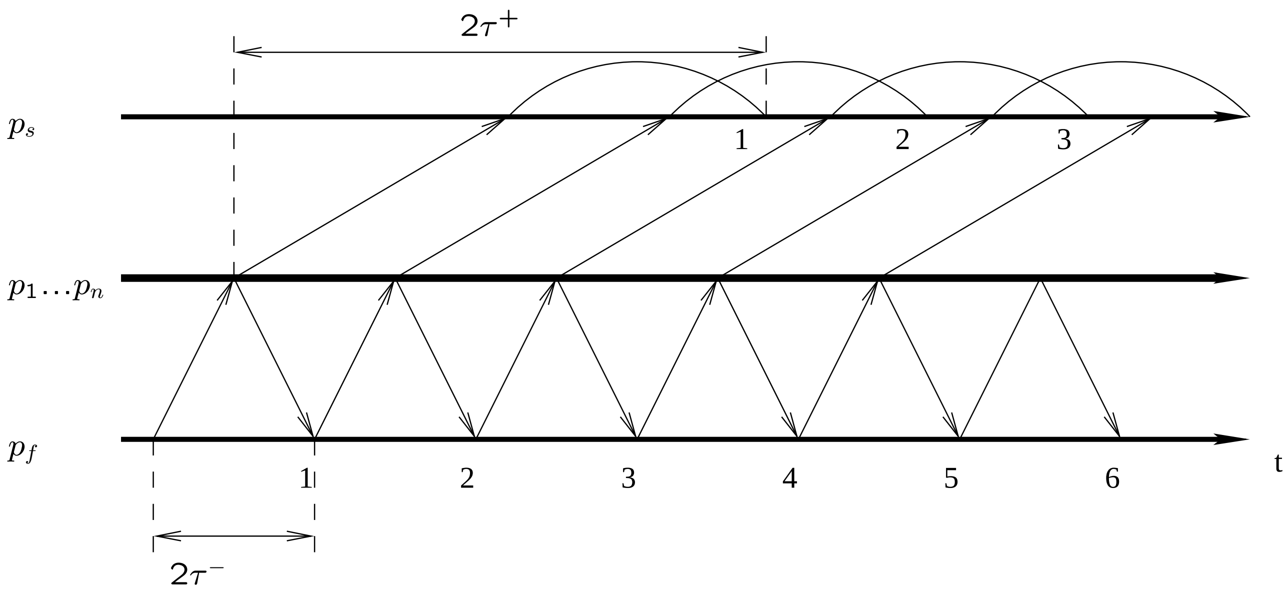
if received *(echo, round k)* from $2f+1$ p's
  $\rightarrow$ unblock *accept (round k)*
    send *(init, round k+1)* to all;
fi

Maximum Deviation of CB

$R_i(t)$ is the round number of p_i at real time t.

Theorem. For the consistent broadcast primitive there exists a constant D_{max} such that $\mid R_i(t) - R_j(t) \mid \leq D_{max}$ for all processes p_i and p_j at every instant t.

$D_{max} = \lfloor \mathcal{P} + \frac{1}{2} \rfloor$.



The algorithm does not need to know $\mathcal{P}$!

Initialization

- $2f + 1$ correct processes needed to ensure P1 - P3
  - wait until received messages from $3f+1$ processes to start *round 0*
  - $N \geq 4f + 1$ processes needed
- $4f$ is too much !

Solution:

- Find a primitive that does the job with less processes
- Wait only for $2f + 1$ processes
- Integrate the late starters

Problems

- Primitive needs $2f + 1$ correct processes to ensure properties P1 - P3 $\Rightarrow$ weaken properties

- round switching is only possible with the help of faulty processes.

- new system behaviour makes integration difficult.

Modified CB Primitive

VAR k : integer;

if received *(init, round k)* from $f + 1$ p's
  $\rightarrow$ send *(echo, round k)* to all;
fi

if received *(echo, round k)* from $f + 1$ p's
  $\rightarrow$ send *(echo, round k)* to all;
fi

if received *(echo, round k)* from $2f + 1$ p's
  $\rightarrow$ unblock *accept (round k)*;
    $k := k + 1$;
    send *(init, round k)* to all;
fi

if received *(echo, round* l*)* from $f + 1$ p's
  where $l > k$
  $\rightarrow$ unblock *accept (round* $l - 1$*)*;
    $k := l$; /* **jump to new round** */
    send *(echo, round k)* to all;
fi

Properties of MCB

even with $N_{act} \leq 3f$:

- P1W. *Weak Correctness.* If at least $f+1$ active correct processes broadcast *(round k)* messages by time t, then every active correct process accepts *round* **k-1** by time $t + \tau^{+}$.

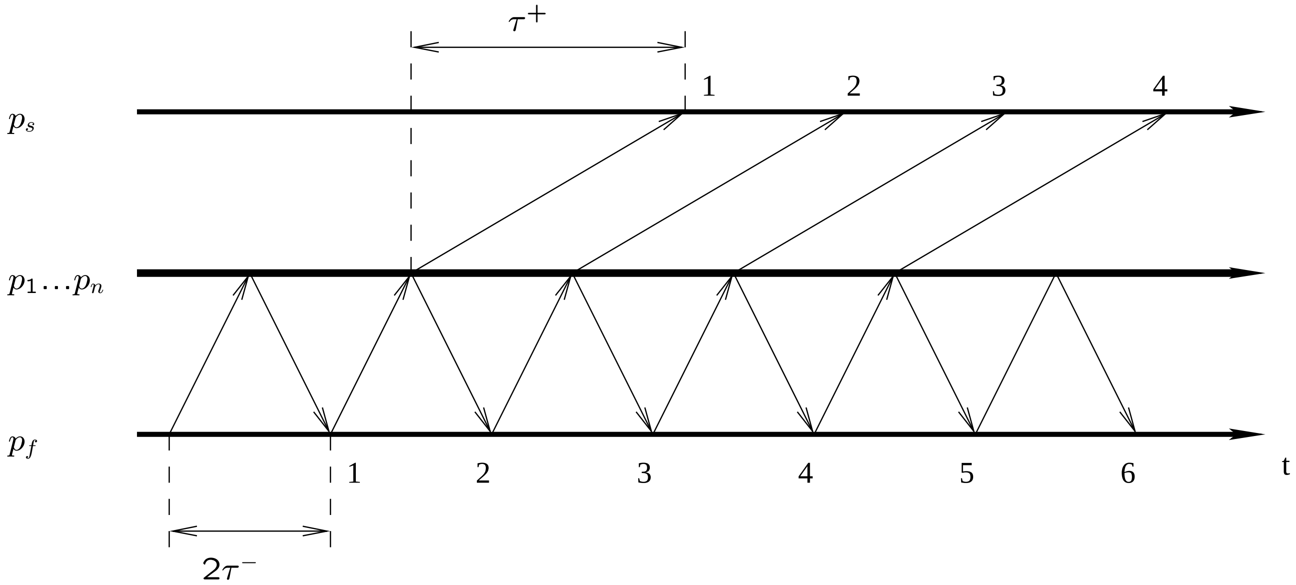
- P2. *Unforgeability.* If no correct process broadcasts a *(round k)* message by time t, then no correct process accepts *round k* by $t + 2\tau^{-}$ or earlier.

- P3W. *Weak Relay.* If a correct process accepts the message *(round k)* at time t, then every active correct process accepts *round* **k-1** by time $t + \epsilon$.

Maximum Deviation using the MCB Primitive

Theorem. For the modified primitive there exists a constant D_{max} such that $| R_i(t) - R_j(t) | \leq D_{max}$ for all processes p_i and p_j at every time t.

$D_{max} = \lfloor \frac{1}{2}\mathcal{P} + \frac{3}{2} \rfloor$.



System Behaviour during Initialization

- $N_{act} \geq 2f + 1$
- No Progress Guarantee
- Known solutions for integration need progress $\Rightarrow$ active integration
- initializing process asks the others about their local round numbers
- When am I synchronized ?

Initialization

- Active and Passive Primitive are needed to ensure D_{max}

- Passive Primitive

  - is started when the process starts

  - learns about the system state

  - does not accept rounds

  - switches to active primitive when process can be sure that its local round number is good enough

- Active Primitive

  - is started when process is synchronized

  - accepts rounds

Results

- Bounded time it takes to get system running after enough correct processes are up. $\Delta_{init} \leq 6\tau^+$.

- Bounded D_{max}^{system} during the whole system life-cycle including booting phase. $D_{max}^{system} = \lceil \frac{3}{2}\mathcal{P} + 4 \rceil$

  - if assumption on τ^+ fails, assumption on $\mathcal{P}$ could still hold

  - if assumption on $\mathcal{P}$ temporarily fails, algorithm keeps working properly with larger D_{max} and returns to calculated D_{max}^{system} if $\mathcal{P}$ holds again

What is it good for ?

- weak timeout mechanism

- can be used in a muteness detector

Future Work

- transformation into hybrid failure model

- analysis in the crash recovery model

- initialization of other services