Consensus with Broadcast Busses


System Model (I)

*process fault*: crash, arbitrary
*channel fault*: can drop incoming message
*link fault*: can drop messages between channel and process

$n$ processes

$synchronized clocks, bounded message delivery, digital signatures$

$R$ channels

$R$ links
System Model (II)

\( \text{xcast}(v, i) \) sends \( v \) on link/channel \( i \)

- a null message \( \phi \) is generated by a timeout.
- all processes that receive a non-null message receive the same message.

\( \text{xcast}(v, S) \) for \( S \subseteq \{1, 2, \ldots, R\} \) is a parallel execution of
\( \text{xcast}(v, p_1) \parallel \text{xcast}(v, p_2) \parallel \ldots \parallel \text{xcast}(v, p_k) \)

where \( S = \{p_1, p_2, \ldots, p_k\} \). Parallel execution is done in no particular order and not done atomically.

\( \text{xcast}(v, *) \) is shorthand for \( \text{xcast}(v, \{1, 2, \ldots, R\}) \)
System Model (III)

A process is *properly connected* to a channel if its link to that channel is nonfaulty.

Let $P = \{p_1, p_2, \ldots p_n\}$ be the set of processes where $p_1$ is the transmitter.

$V$ is the set of possible message values (excluding $\phi$) and $v_0 \in V$ is the *default value*.
System Model (IV)

In a run, let
\[ \pi = \text{the number of processes that are faulty} \]
\[ 0 \leq \pi \leq n \]
\[ \psi = \text{the number of channels that are faulty} \]
\[ \lambda = \text{the number of links that are faulty} \]
\[ \psi \geq 0 \text{ and } \lambda \geq 0 \]
\[ R > \lambda + \psi, \text{ which we show next...} \]
Being Connected

If $R > \lambda + \psi$ then there will always be a direct path connecting any pair of processes.

Remove $\psi$ right nodes. Each node on the left has $R - \psi > \lambda$ edges incident.
Omission: 2 rounds

Round 1:
\[ p_1: \text{xcast}(v, \ast); \]
\[ \text{decide}(v); \]

Round 2:
\[ p_i, i \neq 1: \]
\[ \text{if} \ (|C^1_i| \neq 0) \ \text{then} \]
\[ \text{xcast}(m_i, \{1,2, \ldots R\} - C^1_i); \]
\[ \text{decide}(m_i); \]

After round 2:
\[ p_i, i \text{ not yet decided:} \]
\[ \text{if} \ (|C^2_i| \neq 0) \ \text{then} \ \text{decide}(m_i); \]
\[ \text{else} \ \text{decide}(v_0); \]

\[ C^r_i \text{ is set of channels over which } p_i \text{ receives non-null message } m_i \in V \text{ in round } r \]

Since there are only crash and omission failures, \( m_i \) is unique.
Omission: Example
Omission: Example
Omission: Example
Omission: Example
Omission: Proof (I)

If $\lambda + \pi \leq n$ then the protocol implements reliable broadcast.

Proof by case analysis:

1. Nonfaulty $p_1$.
   
   All nonfaulty $p$ receive $v$ in Round 1 because $p_1$ and $p$ are connected.
   
   All nonfaulty $p$ decide on $v$ in Round 2
Omission: Proof (II)

2. Faulty $p_1$ that broadcasts no message over nonfaulty channels to which it is properly connected.

No value from $p_1$ is received by anyone.

By end of Round 2, all nonfaulty $p$ decide on $v_0$. 
Omission: Proof (III)

3. Faulty $p_1$ and some nonfaulty $p_i$ receives initial broadcast.

Thus $m_i = v$.

Nonfaulty $p_j$ will either receive in Round 1,

if $p_j$ is well connected to channel over which $p_i$ received $v$,

or in Round 2

because $p_i$ and $p_j$ are connected.
Omission: Proof (IV)

4. Faulty $p_1$ and only faulty processes receive $v$ in Round 1

$p_1$ sent $v$ on at least one channel because some processes received $v$.

Hence all $n - \pi$ nonfaulty processes have faulty links to that channel, so $\lambda \geq n - \pi$.  

---

Winter Quarter 2009  
CSE 223A: Consensus with Broadcast 
Busses
Omission: Proof (V)

But by assumption $\lambda + \pi \leq n$ or $\lambda \leq n - \pi$ and so there can be no further link failures: only these $n - \pi$ links are faulty.

If any faulty process succeeds in sending $\nu$ on a channel that does not drop the message, then all nonfaulty $p$ will receive and decide on $\nu$ in Round 2.

Else they decide on $\nu_0$. 

\[\text{Diagram of process}$}
Omission: Summary

\[ 0 \leq \lambda + \psi < R \text{ and } 0 \leq \lambda + \pi \leq n \]

Choose \( R \) based on probability of link failures and channel failures. Then choose \( n \) based on probability of process crashes.

Some boundary cases:

- when \( \pi = 0 \) then \( \lambda < n \) and \( \lambda + \psi < R \)
- when \( \lambda = 0 \) then \( \psi < R \) and \( \pi \leq n \)
- when \( \psi = 0 \) then \( \lambda < R \) and \( \lambda + \pi \leq n \)
Arbitrary Failures: 2 rounds

Requires $n > t + \pi + 2\lambda$ where $t$ is the maximum possible number of process failures.
Still need $R > \lambda + \psi$.

Values sent are in $W = V \cup \{\phi, \bot\}$ where $\bot$ denotes bad value.

$S^k$ denotes all multisets of $k$ elements drawn from $S$.

For example, $\{0, 1\}^2 = \{\{0, 0\}, \{0, 1\}, \{1, 1\}\}$
Arbitrary Failures (II)

Every nonfaulty process broadcasts at most one message over a channel in any single round.

$B^r_{i}(j) \in W^R$ is the multiset of messages $p_i$ received from $p_j$ in round $r$, one message per channel.

If $p_i$ received more than one message per channel in Round $r$ from $p_j$, then $B^r_{i}(j) = \{\bot\}^R$.

$B^r_{i}(j)$ is consistent in $v$ if $B^r_{i}(j) \in \{\emptyset, v\}^R$ for some $v \in W$ and $v \neq \bot$. 
Arbitrary Failures (III)

Some useful functions:

\( \sigma: W^R \rightarrow V: v \) if parameter is consistent in \( v \); \( v_0 \) otherwise.

\( \gamma^z: W^n \rightarrow V: v \) if \( v \) is most frequent non-\( \phi \) value in parameter and there are at least \( z \) of them; \( v_0 \) otherwise.

... defined only if \( 2z > n \).
Arbitrary Failures: Protocol

Round 1:

\[ p_1 : \text{xcast}(v, \ast); \]

Round 2:

\[ p_i : \quad m_i = \sigma(B_1^1(1)); \]
\[ \text{xcast}(m_i, \ast); \]

After round 2:

\[ p_i : \quad M_i = \{ \forall j : 1 \leq j \leq n : \sigma(B_2^1(j)) \}; \]
\[ \text{decide on } \gamma^{t+1}(M_i) \]
Arbitrary Failures: Example

\[ t = \pi = 1, \lambda = 1, \psi = 2 \]
Arbitrary Failures: Example

\[ t = \pi = 1, \lambda = 1, \psi = 2 \]

Round 1
Arbitrary Failures: Example

\[ t = \pi = 1, \lambda = 1, \psi = 2 \]

Round 2
Arbitrary Failures: Example

\[ t = \pi = 1, \lambda = 1, \psi = 2 \]

Round 2
Arbitrary Failures: Example

\[ t = \pi = 1, \lambda = 1, \psi = 2 \]

After round 2
Arbitrary Failures: Proof (I)

Sending process \( p_j \):

- *is overtly malicious* when its multiset of message values as observed on channels is not consistent;

  - \( p_i \) is a *witness* if \( B^r_i(j) \) is not consistent in \( \nu \) for some \( \nu \).
Arbitrary Failures: Overtly Malicious
Arbitrary Failures: Proof (II)

Sending process $p_j$:

- *exhibits overt omission failure* if it is not overtly malicious and broadcasts null messages over at least one channel to which it is well-connected;
  - $p_i$ is a witness if $B^r_i(j) = \{\phi\}^R$.
  - $p_j$, crashing subsequently exhibits overt omission failure.
Arbitrary Failures: Exhibits Overt Omission Failure
Arbitrary Failures: Proof (III)

Sending process $p_j$:

- *is correct* if it does neither.
Arbitrary Failures: Proof (IV)

Lemma: Let $k$ be the number of witnesses to an overt failure of $p_1$ by the end of Round 1 (not counting the transmitter).
**Arbitrary Failures: Proof (V)**

**Overt omission:** $0 \leq k \leq \lambda$ or $k = n - 1$

Let $x$ be the number of nonfaulty channels to which $p_1$ is properly connected and to which $p_1$ sends non-null messages: $0 \leq x \leq R$.

If $x = 0$ then all of the remaining $n - 1$ processes witness the failure.

If $x > 0$ then each process that witnesses the failure has $x$ faulty links.

Given $k$ witnesses, there were $kx \leq \lambda$ link failures.

$kx$ may be less than $\lambda$ because there may be subsequent link failures.

So, $k$ is bounded from above by $[\lambda/x]$, and thus is in the range from 0 to $\lambda$. 
Arbitrary Failures: Proof (VI)

**Overtly malicious failure:** \( n - \lambda - 1 \leq k \leq n - 1 \)

\( p_1 \) failed by successfully broadcasting two different non-null messages over two channels.

\( p_j \) will witness unless it is not well-connected to at least one of these channels. At most \( \lambda \) processes can be not well-connected in this manner.

Thus, between \( n - \lambda - 1 \) and \( n - 1 \) processes will witness \( p_1 \)’s overtly malicious failure.
Arbitrary Failures: Proof (VII)

Proof of correctness when $n > t + \pi + 2\lambda$ by case analysis.

1. Nonfaulty $p_1$.

   By end of Round 1 and because all processes are connected, all $B^1_i(1)$ are consistent in $\nu$.

   Thus all $m_i = \nu$ and are sent in Round 2 by nonfaulty processes $p_i$.

   After Round 2, at all nonfaulty processes $p_i$, at least $n - \pi$ occurrences of $\nu$ are in in $M_i$ along with at most $\pi$ occurrences of some other value or $\bot$.

   Since $n > t + \pi + 2\lambda$ and $t \geq \pi$, $n > 2\pi$ and so $n - \pi > \pi$. Thus, the $\nu$ values dominate. And, $n - \pi > t + 2\lambda \geq t$. So $n - \pi \geq t + 1$.

   Thus $\gamma^{t+1}(M_i)$ computes $\nu$. 
Arbitrary Failures: Proof (VIII)

2. Overt omission failure by $p_1$.
   There will be $k \in \{0, 1, \ldots, \lambda, n - 1\}$ witnesses.
   If $k = n - 1$ then at least $n - \pi$ broadcast $\phi$ in Round 2.
   As in Case 1, the nonfaulty processes decide on $v_0$.
   If $k \leq \lambda$, then at least $n - \pi - k - 1$ nonfaulty processes do not
   witness the failure. These, along with $p_1$, set $m_i = \nu$ and send
   $\nu$-consistent messages in Round 2.
   All nonfaulty processes will have at least $n - \pi - k$ $\nu$-values in $M_i$.
   $n > t + \pi + 2\lambda$ or $n - \pi - k > t + (2\lambda - k) \geq t$
   or $n - \pi - k > t$
   Thus $\gamma^{t+1}(M_i)$ computes $\nu$. 

Arbitrary Failures: Proof (IX)

3. Overtly malicious $p_1$.
   
   Number of witnesses $k \geq n - \lambda - 1$.
   
   At least $k - (\pi - 1)$ processes send $v_0$-values in Round 2.
   
   All nonfaulty processes $p_i$ have at least $k - \pi + 1$ $v_0$-values in $M_i$.
   
   $k - \pi + 1 \geq n - \lambda - \pi$
   
   but $n > t + \pi + 2\lambda$ or $n - \lambda - \pi \geq t + \lambda + 1 \geq t + 1$
   
   So $k - \pi + 1 \geq t + 1$
   
   Thus $\gamma^{t+1}(M_i)$ computes $v_0$. 
Arbitrary Failures: Summary

\[ 0 \leq \lambda + \psi < R \text{ and } 0 \leq t + \pi + 2\lambda < n \]

More simply, \[ n \geq 2(t + \lambda) + 1 \]

Compare with \[ n \geq t + \lambda \] for send omission failure model.