From Partial to Global Asynchronous Reliable Broadcast

Diana Ghinea    Martin Hirt    Chen-Da Liu-Zhang

ETH Zurich

DISC 2020
Broadcast

Broadcast allows a party to consistently distribute a message to $n$ recipients.

Sender said ”Hi!”
Sender said ”Hi!”
Sender said ”Hi!”
Sender said ”Hi!”
Broadcast

Broadcast allows a party to consistently distribute a message to $n$ recipients.
Broadcast

Broadcast allows a party to consistently distribute a message to $n$ recipients.
Broadcast

Broadcast allows a party to consistently distribute a message to $n$ recipients.

(Sender said “Hi!”)
(Sender said “Bye!” “Hi!”)
(Sender said “Bye!”)
(Sender said “Hi!”)

(Consistency)
Broadcast

Broadcast allows a party to consistently distribute a message to \( n \) recipients.

(Sender)

Sender said "Hi!"

(Sender)

Sender said "Bye!"  "Hi!"

(Sender)

Sender said "Bye!"

(Sender)

Sender said "Hi!"

(Validity)
Model

Synchronous channels

I will receive the message in one hour.
Model

Asynchronous channels

I will receive the message eventually.

If it was sent.
Model

Asynchronous channels

I will receive the message eventually.
If it was sent.

Adversary

- Controls the delay time of the messages.
- Corrupts up to $t$ parties: they send wrong messages or they do not send some of the messages.
Achieving Asynchronous Reliable Broadcast

To achieve asynchronous reliable broadcast, a protocol must satisfy the following properties:

Validity
Honest Sender with input \( m \)
\[ \implies \] Every honest recipient terminates and outputs \( m \).

Consistency
An honest recipient terminates with output \( m \)
\[ \implies \] Every honest recipient terminates with output \( m \).
## Thresholds

<table>
<thead>
<tr>
<th></th>
<th>Synchronous BC</th>
<th>Asynchronous RBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional model</td>
<td>$t &lt; n/3$ [PSL80]</td>
<td>$t &lt; n/3$ [BraTou85]</td>
</tr>
<tr>
<td>PKI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b-cast</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Thresholds

<table>
<thead>
<tr>
<th></th>
<th>Synchronous BC</th>
<th>Asynchronous RBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional model</td>
<td>$t &lt; n/3$ [PSL80]</td>
<td>$t &lt; n/3$ [BraTou85]</td>
</tr>
<tr>
<td>PKI</td>
<td>$t &lt; n$ [DolStr83]</td>
<td>$t &lt; n/3$</td>
</tr>
<tr>
<td>b-cast</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Thresholds

<table>
<thead>
<tr>
<th></th>
<th>Synchronous BC</th>
<th>Asynchronous RBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional model</td>
<td>$t &lt; n/3$ [PSL80]</td>
<td>$t &lt; n/3$ [BraTou85]</td>
</tr>
<tr>
<td>PKI</td>
<td>$t &lt; n$ [DolStr83]</td>
<td>$t &lt; n/3$</td>
</tr>
<tr>
<td>b-cast</td>
<td>$b = 3$</td>
<td>$t &lt; n/2$ [FitMau00]</td>
</tr>
<tr>
<td></td>
<td>$b - 1$ recipients</td>
<td></td>
</tr>
</tbody>
</table>

**b-cast channel**

$m$ (😊,😊, ... ,😊)

$b - 1$ recipients
## Thresholds

<table>
<thead>
<tr>
<th></th>
<th>Synchronous BC</th>
<th>Asynchronous RBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional model</td>
<td>$t &lt; n/3$ [PSL80]</td>
<td>$t &lt; n/3$ [BraTou85]</td>
</tr>
<tr>
<td>PKI</td>
<td>$t &lt; n$ [DolStr83]</td>
<td>$t &lt; n/3$</td>
</tr>
<tr>
<td>b-cast</td>
<td>$b = 3$</td>
<td>$t &lt; n/2$ [FitMau00]</td>
</tr>
<tr>
<td></td>
<td>$b &lt; n$</td>
<td>$t &lt; \frac{b-1}{b+1}n$ [CFFLMM05]</td>
</tr>
</tbody>
</table>

$b$-cast channel: $b - 1$ recipients

$m$
## Thresholds

<table>
<thead>
<tr>
<th></th>
<th>Synchronous BC</th>
<th>Asynchronous RBC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Traditional model</strong></td>
<td>$t &lt; n/3$ [PSL80]</td>
<td>$t &lt; n/3$ [BraTou85]</td>
</tr>
<tr>
<td><strong>PKI</strong></td>
<td>$t &lt; n$ [DolStr83]</td>
<td>$t &lt; n/3$</td>
</tr>
<tr>
<td><strong>b-cast</strong></td>
<td>$b = 3$</td>
<td>$t &lt; n/2$ [FitMau00]</td>
</tr>
<tr>
<td></td>
<td>$b &lt; n$</td>
<td>$t &lt; \frac{b-1}{b+1} n$ [CFFLMM05]</td>
</tr>
</tbody>
</table>

$\begin{align*}
\text{b-cast channel} & \quad m \\
\text{b-recipients} & \quad (\circled{\text{m}}, \circled{\text{m}}, \ldots, \circled{\text{m}}) \\
\text{b - 1 recipients} & \quad \circled{\text{m}}
\end{align*}$
Our Results

Feasibility

An asynchronous reliable broadcast protocol for \( b = 3 \), secure against \( t < n/2 \) corruptions.

An asynchronous reliable broadcast protocol, secure against \( t < \frac{b-4}{b-2} n \) corruptions.

A nonstop reliable broadcast protocol, secure against \( t < \frac{b-4}{b+1} n \) corruptions.

Impossibility

In the asynchronous setting, there is no protocol achieving ( ) reliable broadcast secure against \( t \geq \frac{b-1}{b+1} n \) corruptions.
Our Results

Feasibility

- An asynchronous reliable broadcast protocol for $b = 3$, secure against $t < n/2$ corruptions.

Impossibility

In the asynchronous setting, there is no protocol achieving (nonstop) reliable broadcast secure against $t \geq \frac{b-1}{b+1} n$ corruptions.
Our Results

Feasibility

- An asynchronous reliable broadcast protocol for $b = 3$, secure against $t < n/2$ corruptions.
- An asynchronous reliable broadcast protocol, secure against $t < \frac{b-4}{b-2}n$ corruptions.

Impossibility

In the asynchronous setting, there is no protocol achieving ( ) reliable broadcast secure against $t \geq \frac{b-1}{b+1}n$ corruptions.
Our Results

Feasibility

- An asynchronous reliable broadcast protocol for $b = 3$, secure against $t < n/2$ corruptions.
- An asynchronous reliable broadcast protocol, secure against $t < \frac{b-4}{b-2}n$ corruptions.
- A *nonstop* reliable broadcast protocol, secure against $t < \frac{b-1}{b+1}n$ corruptions.

Impossibility

In the asynchronous setting, there is no protocol achieving (*nonstop*) reliable broadcast secure against $t \geq \frac{b-1}{b+1}n$ corruptions.
Model $\mathcal{N}_3$

- 3-cast channels among any 3 parties.

\[ m \rightarrow (\smiley, \smiley) \]

3-cast channel
Model $\mathcal{N}_3$

- 3-cast channels among any 3 parties.

- $P$ mega-sends $m$:
  
  $P$ sends $m$ to every pair of recipients via 3-cast.

\[ \begin{array}{ccc}
\text{P} & m & (\ast, \ast) \\
\end{array} \]
Model $\mathcal{N}_3$

- 3-cast channels among any 3 parties.

- $P$ mega-sends $m$:
  $P$ sends $m$ to every pair of recipients via 3-cast.

- $R$ mega-receives $m$ from $P$:
  $R$ received $m$ from $P$ through all the available 3-cast channels.

\[
(R, \star) \quad m \quad \odot \quad P
\]
Model $\mathcal{N}_3$

- 3-cast channels among any 3 parties.

- $P$ mega-sends $m$:
  $P$ sends $m$ to every pair of recipients via 3-cast.

- $R$ mega-receives $m$ from $P$:
  $R$ received $m$ from $P$ through all the available 3-cast channels.

- $R$ mega-receives $m$ from $P$ $\implies$ $R'$ receives $m$ from $P$. 

\[
\begin{array}{c}
(R, \star) \quad m \quad P \quad \implies (R, R') \quad m \quad P
\end{array}
\]
Protocol in $\mathcal{N}_3$

Code for Sender $S$

1. On input $m$:
   \[ \text{mega-send (MSG, } m) \]
Protocol in $N_3$

Code for Sender $S$

1. On input $m$:
   mega-send ($MSG, m$)

Code for Recipient $R_i$

1. When mega-receiving ($MSG, m$) from $S$:
   mega-send ($READY, m$)
Protocol in $\mathcal{N}_3$

Code for Sender $S$
1. On input $m$:
   \text{mega-send} (\text{MSG}, m)$

Code for Recipient $R_i$
1. When \text{mega-receiving} (\text{MSG}, m) from $S$ or when \text{receiving} 
   (\text{READY}, m) from $t + 1$ recipients:
   \text{mega-send} (\text{READY}, m)
Protocol in $\mathcal{N}_3$

**Code for Sender $S$**

1. On input $m$:
   
   \[ \text{mega-send} \ (\text{MSG}, m) \]

**Code for Recipient $R_i$**

1. When \( \text{mega-receiving} \ (\text{MSG}, m) \) from $S$ or when \( \text{receiving} \ (\text{READY}, m) \) from $t + 1$ recipients:
   
   \[ \text{mega-send} \ (\text{READY}, m) \]

2. When \( \text{mega-receiving} \ (\text{READY}, m) \) from $n - t - 1$ recipients and \( (\text{READY}, m) \) was \text{mega-sent}:

   \[ \text{output } m \text{ and terminate} \]

\[ \Rightarrow \quad \exists R_{k_1} \text{ outputs } m \text{ and terminates} \]
Validity: \( t < n - t \)

**Code for Sender** \( S \)

1. On input \( m \):
   
   \[
   \text{mega-send (MSG, m)}
   \]

**Code for Recipient** \( R_i \)

1. When \( \text{mega-receiving (MSG, m)} \) from \( S \) or when \( \text{receiving (READY, m)} \) from \( t + 1 \) recipients:
   
   \[
   \text{mega-send (READY, m)}
   \]

2. When \( \text{mega-receiving (READY, m)} \) from \( n - t - 1 \) recipients and \( \text{(READY, m)} \) was mega-sent:
   
   \[
   \text{output m and terminate}
   \]

Honest Sender’s input: \( m \)

**Fact:**

Honest \( R \) cannot mega-send \( \text{(READY, m')} \)  

\[ \Rightarrow R \text{ mega-sends (READY, m)} \]

\[ \Rightarrow R \text{ outputs m} \]
Consistency: \( t < n - t \)

**Code for Sender \( S \)**

1. On input \( m \):
   
   \[ \text{mega-send} \ (\text{MSG}, m) \]

**Code for Recipient \( R_i \)**

1. When \textit{mega-receiving} \((\text{MSG}, m)\) from \(S\) or when \textit{receiving} \((\text{READY}, m)\) from \(t + 1\) recipients:
   
   \[ \text{ mega-send } (\text{READY}, m) \]

2. When \textit{mega-receiving} \((\text{READY}, m)\) from \(n - t - 1\) recipients and \((\text{READY}, m)\) was \textit{ mega-sent}: 
   
   \[ \text{output } m \text{ and terminate} \]

**Fact \#1:**

An honest \( R \) \textit{ mega-sends } \((\text{READY}, m)\) 

\( \implies \) No honest \( R' \) \textit{ mega-sends } \((\text{READY}, m')\) 

\( \implies \) No honest \( R' \) \textit{ outputs } \( m' \)

**Fact \#2:**

An honest \( R \) \textit{ outputs } \( m \) 

\( \implies \) Any honest \( R' \) \textit{ mega-sends } \((\text{READY}, m)\) 

\( \implies \) Any honest \( R' \) \textit{ outputs } \( m \)
Model $\mathcal{N}_b$

- **Model $\mathcal{N}_b$ ($b > 3$):** $b$-cast channels among every group of $b$ parties.
Model $\mathcal{N}_b$

- Model $\mathcal{N}_b \ (b > 3)$: $b$-cast channels among every group of $b$ parties.

- Goal when $b = 3$: $t < n - t$
- Goal when $b > 3$: $t \geq n - t$
Model $\mathcal{N}_b$

- **Model $\mathcal{N}_b$ ($b > 3$):** $b$-cast channels among every group of $b$ parties.

- Goal when $b = 3$: $t < n - t$
- Goal when $b > 3$: $t \geq n - t$
Model $\mathcal{N}_b$

- **Model $\mathcal{N}_b$ ($b > 3$):** $b$-cast channels among every group of $b$ parties.

- Goal when $b = 3$: $t < n - t$

- Goal when $b > 3$: $t \geq n - t$

$\Rightarrow$ Levels of confidence
Messages Received from $S$

Initially, $S$ forwards his input $m$ to every group of $b - 1$ recipients.
Messages Received from $S$

Initially, $S$ forwards his input $m$ to every group of $b - 1$ recipients.

- $R_1$ **1-receives** $m$:
  
  $R_1$ receives $m$ from $S$ through all the available $b$-cast channels.
  
  $(R_1, *, *, *, ..., *, *)$
Messages Received from $S$

Initially, $S$ forwards his input $m$ to every group of $b - 1$ recipients.

- $R_1$ **1-receives** $m$:
  - $R_1$ receives $m$ from $S$ through all the available $b$-cast channels.
    
    $(R_1, *, *, *, ..., *, *)$

- $R_2$ **2-receives** $m$:
  - $R_2$ receives $m$ from $S$ through all the $b$-cast channels **shared with one other recipient** $R_1$.
    
    $(R_1, R_2, *, *, ..., *, *)$
Messages Received from \( S \)

Initially, \( S \) forwards his input \( m \) to every group of \( b - 1 \) recipients.

- **\( R_1 \) 1-receives \( m \):**
  \( R_1 \) receives \( m \) from \( S \) through **all** the available \( b \)-cast channels.
  \((R_1, *, *, *, ..., *, *)\)

- **\( R_2 \) 2-receives \( m \):**
  \( R_2 \) receives \( m \) from \( S \) through **all** the \( b \)-cast channels **shared with one other recipient** \( R_1 \).
  \((R_1, R_2, *, *, ..., *, *)\)

  ...

- **\( R_{b-1} \) (\( b - 1 \))-receives \( m \):**
  \( R_{b-1} \) receives \( m \) from \( S \) through **all** the \( b \)-cast channels **shared with \( b - 2 \) other recipients** \( R_1, \ldots, R_{b-2} \).
  \((R_1, R_2, R_3, R_4, ..., R_{b-2}, R_{b-1})\)
Messages Received from $S$

$R_k$ \textbf{k-receives} $m$: $R_k$ receives $m$ from $S$ through all the available broadcast channels shared with $k - 1$ other recipients $R_1, R_2, \ldots, R_{k-1}$.

$$(R_1, R_2, \ldots, R_{k-1}, R_k, *, *, \ldots, *)$$

\implies \text{Any recipient $R$ ($k + 1$)-receives $m$.}$$$(R_1, R_2, \ldots, R_{k-1}, R_k, R, *, \ldots, *)$$
Messages Received from $S$

$R_k$ \textbf{k-receives} $m$: $R_k$ receives $m$ from $S$ through all the available $b$-cast channels shared with $k - 1$ other recipients $R_1, R_2, \ldots, R_{k-1}$.

$$(R_1, R_2, \ldots, R_{k-1}, R_k, \ast, \ast, \ldots, \ast)$$

$\implies$ It is possible that $R \in \{R_1, \ldots, R_{k-1}\}$ $(k - 1)$-receives $m$.

$$(R_1, R_2, \ldots, R_{k-1}, \ast, \ast, \ast, \ldots, \ast)$$
Levels of Confidence

For a message $m$, we build the following levels:

- **Level 1**: recipients that 1-receive $m$ and *believe* that $S$ is honest.
Levels of Confidence

For a message $m$, we build the following levels:

- **Level 1**: recipients that 1-receive $m$ and *believe* that $S$ is honest.

- **Level 2**: recipients that 2-receive $m$ and *believe* that someone on level 1 is honest and terminated with output $m$. 

...
Levels of Confidence

For a message $m$, we build the following levels:

- **Level 1**: recipients that 1-receive $m$ and believe that $S$ is honest.

- **Level 2**: recipients that 2-receive $m$ and believe that someone on level 1 is honest and terminated with output $m$.

  ...

- **Level $k$**: recipients that $k$-receive $m$ and believe that someone on level $k - 1$ is honest and terminated with output $m$. 
Levels of Confidence

For a message $m$, we build the following levels:

- **Level 1**: recipients that 1-receive $m$ and believe that $S$ is honest.

- **Level 2**: recipients that 2-receive $m$ and believe that someone on level 1 is honest and terminated with output $m$.

  ...

- **Level k**: recipients that $k$-receive $m$ and believe that someone on level $k - 1$ is honest and terminated with output $m$.

  ...

- **Level b**: recipients that do not receive $m$, but believe that someone on level $b - 1$ is honest and terminated with output $m$. 
Level 1

When a recipient 1-receives $m$, it places itself on level 1 and sends notifications to the other recipients.
Level 1

The recipients on level 1 output $m$ if there are $n - t$ recipients that sent notifications for level 1.
Levels 1 and 2

If a recipient 2-receives $m$ and receives notifications for level 1 from $n - t$ recipients, it sends notifications for level 2 and outputs $m$.

Level 1

Level 2

There might be honest parties on level 1
Levels 2 and 3

If a recipient 3-receives $m$ and receives $n - t$ notifications for level 1 and at least one for 2, it places itself on level 3 and sends notifications.

Level 1

Level 2

Level 3

We might be tricked!

There might be honest parties on level 2

Level 3
Levels 2 and 3

When there are \( n - t \) recipients that sent notifications for levels 2 and 3, the recipients on level 3 output \( m \).

- **Level 1**: \( n - t \) recipients
- **Level 2**: We might be tricked!
- **Level 3**: There might be honest parties on level 2
Levels 3 and 4

Level 1

Level 2

Level 3

We might be tricked!

Level 4

There might be honest parties on level 3

Surprise!
Levels 3 and 4

Level 1

Level 2

Level 3

Level 4

Surprise!

We might be tricked!

There might be honest parties on level 3

\[ n - t \]

\[ n - t \]

\[ n - t \]
Different Outputs?

$t$ must be small enough such that the honest recipients cannot place themselves on levels for different messages.

\[ m \quad \text{Level 1} \quad \text{Level 2} \quad \ldots \quad \text{Level } k - 1 \quad \text{Level } k \]

\[ m' \quad \text{Level 1} \quad \text{Level 2} \quad \ldots \quad \text{Level } k' - 1 \quad \text{Level } k' \]

\[ k'-\text{received } m' \quad \text{keeps waiting} \]
Summary

Can we achieve asynchronous reliable broadcast secure against more than \( t < \frac{n}{3} \) corruptions by assuming \( b \)-cast channels? \textbf{Yes!}

What is the trade-off between the strength of the communication network and the corruptive power of the adversary?

- There is no protocol achieving (nonstop) reliable broadcast secure against \( t \geq \frac{b-1}{b+1}n \) corruptions in the asynchronous setting.
- An \textbf{optimal} reliable broadcast protocol for \( b = 3 \).
- An \textbf{almost optimal} reliable broadcast protocol.
- An \textbf{optimal nonstop} reliable broadcast protocol.