From Partial to Global Asynchronous Reliable Broadcast

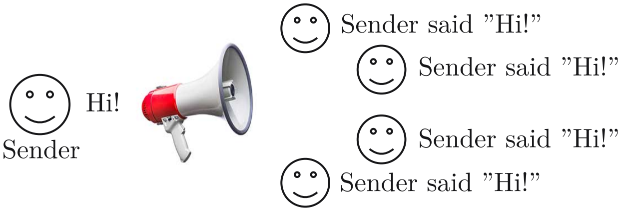
Diana Ghinea Martin Hirt Chen-Da Liu-Zhang

ETH Zurich

DISC 2020

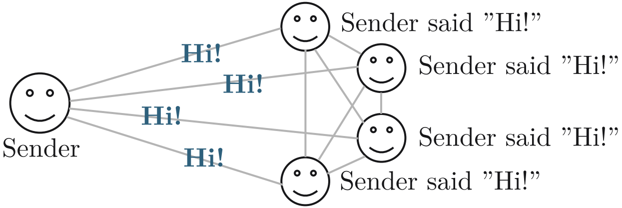
Broadcast

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



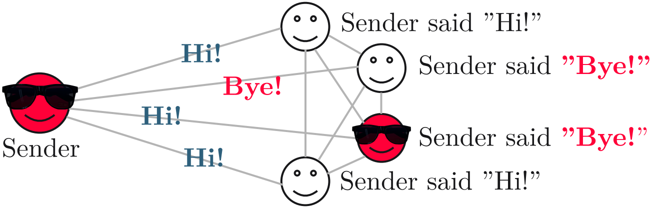
Broadcast

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



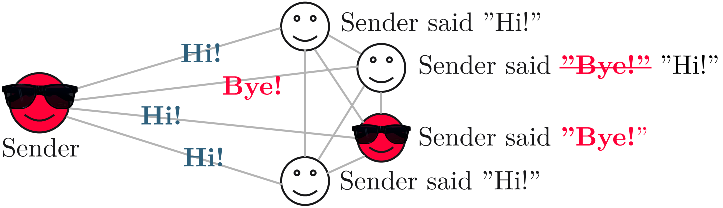
Broadcast

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



Broadcast

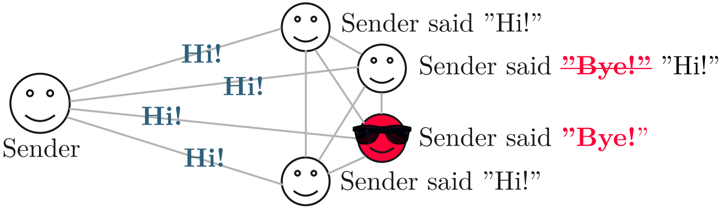
Broadcast allows a party to consistently distribute a message to n recipients.



(Consistency)

Broadcast

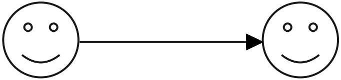
Broadcast allows a party to consistently distribute a message to n recipients.



(Validity)

Model

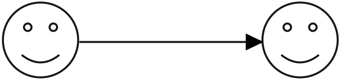
Synchronous channels



I will receive the
message in one hour.

Model

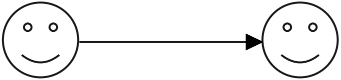
Asynchronous channels



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

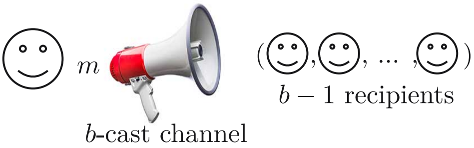
	Synchronous BC	Asynchronous RBC
Traditional model	$t < n/3$ [PSL80]	$t < n/3$ [BraTou85]
PKI		
b-cast		

Thresholds

	Synchronous BC		**Asynchronous RBC**
Traditional model	$t < n/3$ [PSL80]		$t < n/3$ [BraTou85]
PKI	$t < n$ [DolStr83]		$t < n/3$
b-cast			

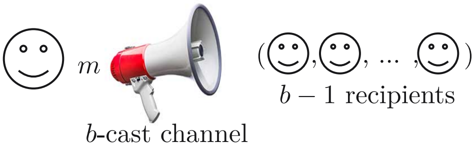
Thresholds

		Synchronous BC	Asynchronous RBC
Traditional model		$t < n/3$ [PSL80]	$t < n/3$ [BraTou85]
PKI		$t < n$ [DolStr83]	$t < n/3$
b-cast	$b = 3$	$t < n/2$ [FitMau00]	



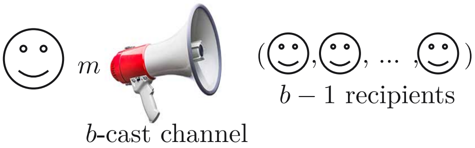
Thresholds

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



Thresholds

		Synchronous BC	Asynchronous RBC
Traditional model		$t < n/3$ [PSL80]	$t < n/3$ [BraTou85]
PKI		$t < n$ [DolStr83]	$t < n/3$
b-cast	$b = 3$	$t < n/2$ [FitMau00]	?
	$b < n$	$t < \frac{b-1}{b+1} n$ [CFFLMM05]	



Our Results

Feasibility

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 ( ) 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.

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.

Model $\mathcal{N}_3$

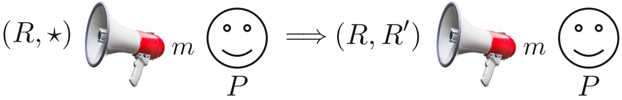
- 3-cast channels among any 3 parties.
- P **mega-sends** m:
  P sends m to every pair of recipients via 3-cast.
- R **mega-receives** m from P:
  R received m from P through all the available 3-cast channels.

$(R, \star)$  m 

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.



Protocol in $\mathcal{N}_3$

Code for Sender S

1. On input m:
   mega-send $(\texttt{MSG}, \mathbf{m})$

S

$(\texttt{MSG}, m)$

$(\star, \star)$

Protocol in $\mathcal{N}_3$

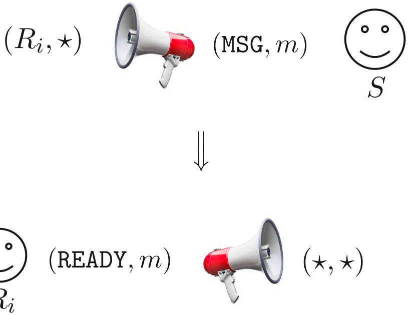
Code for Sender S

1. On input m:
   mega-send $(\texttt{MSG}, \mathbf{m})$

Code for Recipient R_i

1. When *mega-receiving* $(\texttt{MSG}, m)$ from S:

   mega-send $(\texttt{READY}, \mathbf{m})$



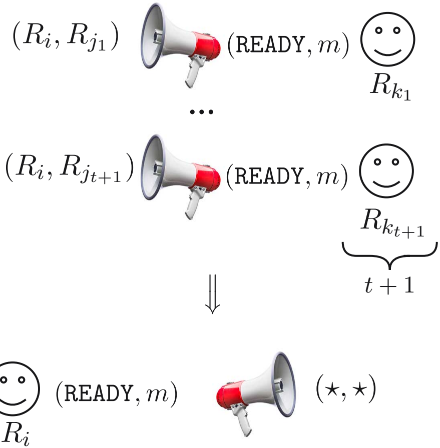
Protocol in $\mathcal{N}_3$

Code for Sender S

1. On input m:
   mega-send $(\texttt{MSG}, \mathbf{m})$

Code for Recipient R_i

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



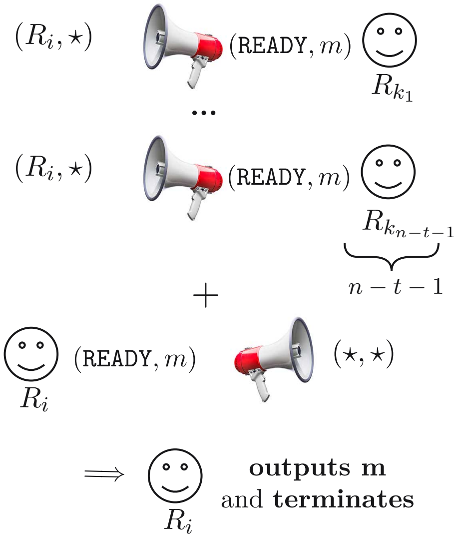
Protocol in $\mathcal{N}_3$

Code for Sender S

1. On input m:
   mega-send $(\texttt{MSG}, \mathbf{m})$

Code for Recipient R_i

1. When *mega-receiving* $(\texttt{MSG}, m)$ from S or when *receiving* $(\texttt{READY}, m)$ from $t+1$ recipients:
   mega-send $(\texttt{READY}, \mathbf{m})$
2. When *mega-receiving* $(\texttt{READY}, m)$ from $n-t-1$ recipients and $(\texttt{READY}, m)$ was *mega-sent*:
   output m and **terminate**



Validity: $t < n - t$

Code for Sender S

1. On input m:
   mega-send $(\texttt{MSG}, \mathbf{m})$

Code for Recipient R_i

1. When *mega-receiving* $(\texttt{MSG}, m)$ from S or when *receiving* $(\texttt{READY}, m)$ from $t + 1$ recipients:
   mega-send $(\texttt{READY}, \mathbf{m})$
2. When *mega-receiving* $(\texttt{READY}, m)$ from $n - t - 1$ recipients and $(\texttt{READY}, m)$ was *mega-sent*:
   output m and **terminate**

Honest Sender's input: m

Fact:
Honest R cannot mega-send $(\texttt{READY}, m')$

$\implies$ R mega-sends $(\texttt{READY}, m)$

$\implies$ R **outputs** m

Consistency: $t < n - t$

Code for Sender S

1. On input m:
   mega-send $(\texttt{MSG}, \mathbf{m})$

Code for Recipient R_i

1. When *mega-receiving* $(\texttt{MSG}, m)$ from S or when *receiving* $(\texttt{READY}, m)$ from $t + 1$ recipients:
   mega-send $(\texttt{READY}, \mathbf{m})$
2. When *mega-receiving* $(\texttt{READY}, m)$ from $n - t - 1$ recipients and $(\texttt{READY}, m)$ was *mega-sent*:
   output m and **terminate**

Fact #1:
An honest R mega-sends $(\texttt{READY}, m)$
$\Longrightarrow$ No honest R' mega-sends $(\texttt{READY}, m')$

$\Longrightarrow$ **No honest R' outputs m'**

Fact #2:
An honest R outputs m
$\Longrightarrow$ Any honest R' mega-sends $(\texttt{READY}, m)$

$\Longrightarrow$ **Any honest R' outputs m**

Model $\mathcal{N}_b$

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

Model $\mathcal{N}_b$

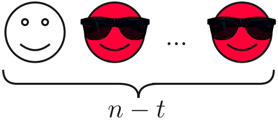
- **Model $\mathcal{N}_\mathbf{b}$** ($b > 3$): b-cast channels among every group of b parties.

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

Model $\mathcal{N}_b$

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

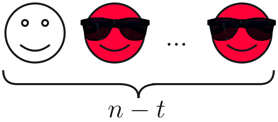
- Goal when $b = 3$: $t < n - t$
- Goal when $b > 3$: $t \geq n - t$



Model $\mathcal{N}_b$

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

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



$\Longrightarrow$ **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.
  $$(R_1, \star, \star, \star, ..., \star, \star)$$

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

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

...

- R_{b-1} $(\mathbf{b-1})$**-receives** m:
  R_{b-1} receives m from S through **all** the b-cast channels **shared with** $\mathbf{b-2}$ **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 **k-receives** m: R_k receives m from S through all the available b-cast channels shared with $k-1$ other recipients $R_1, R_2, \ldots, R_{k-1}$.

$$(R_1, R_2, ..., R_{k-1}, R_k, \star, \star, ..., \star)$$

$\implies$ Any recipient R $(k+1)$-receives m.

$$(R_1, R_2, ..., R_{k-1}, R_k, R, \star, ..., \star)$$

Messages Received from S

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

$$(R_1, R_2, ..., R_{k-1}, R_k, \star, \star, ..., \star)$$

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

$$(R_1, R_2, ..., R_{k-1}, \cancel{R_k}\ \star, \star, \star, ..., \star)$$

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.

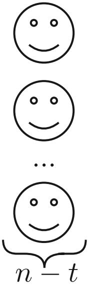
Level 1

  Level 1

Level 1

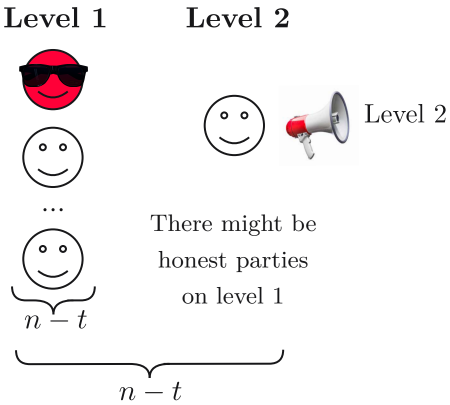
The recipients on level 1 output m if there are $n - t$ recipients that sent notifications for level 1.

Level 1



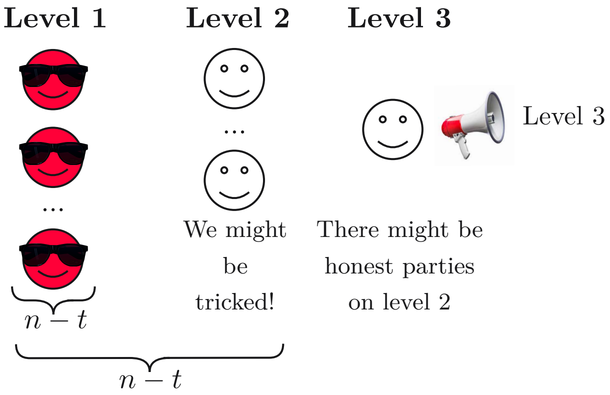
Levels 1 and 2

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.



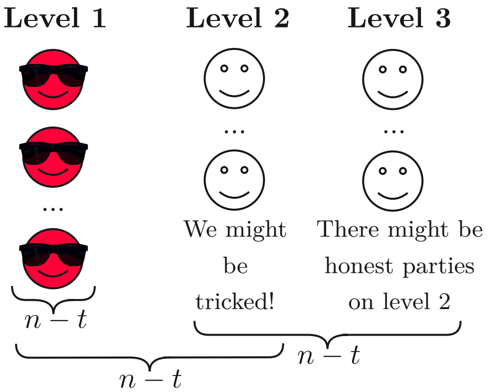
Levels 2 and 3

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.

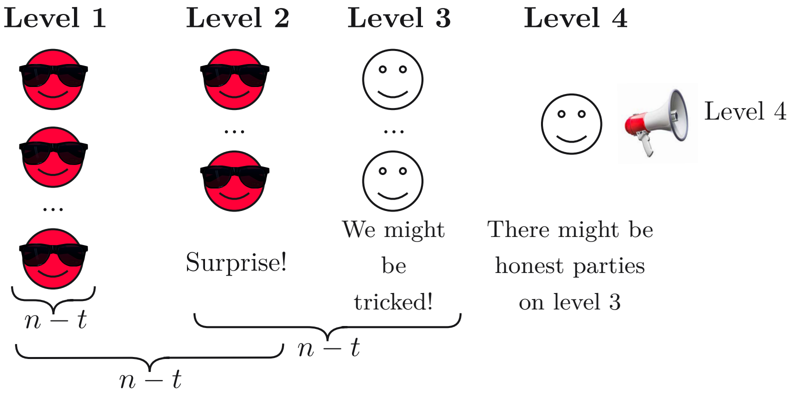


Levels 2 and 3

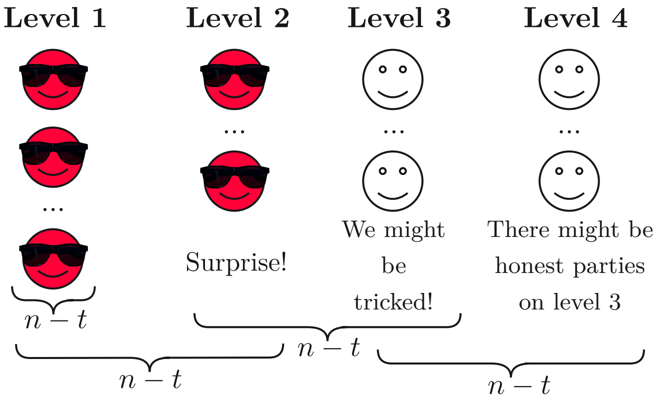
When there are $n - t$ recipients that sent notifications for levels 2 and 3, the recipients on level 3 output m.



Levels 3 and 4

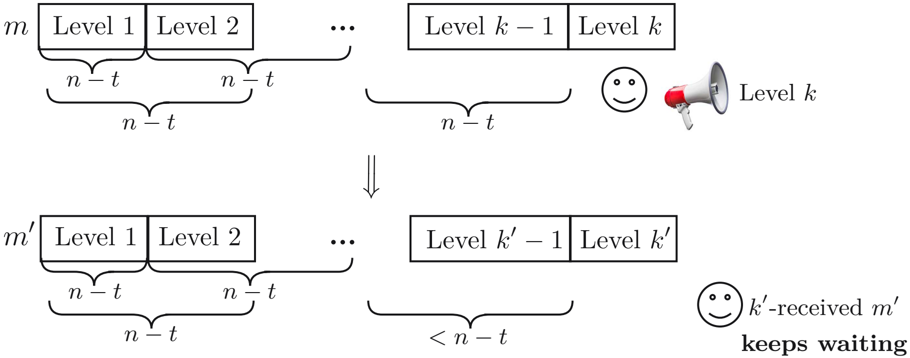


Levels 3 and 4



Different Outputs?

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



Summary

Can we achieve asynchronous reliable broadcast secure against more than $t < n/3$ corruptions by assuming b-cast channels? **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 **optimal** reliable broadcast protocol for $b = 3$.
- An **almost optimal** reliable broadcast protocol.
- An **optimal** *nonstop* reliable broadcast protocol.