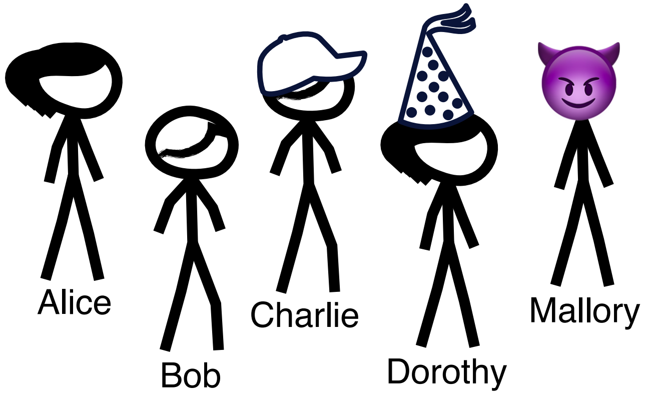
FORMALIZING DELAYED ADAPTIVE CORRUPTIONS AND THE SECURITY OF FLOODING NETWORKS

Christian Matt, *Concordium*
Søren Eller Thomsen, *Aarhus University*
Jesper Buus Nielsen, *Aarhus University*

NAKAMOTO-STYLE BLOCKCHAINS



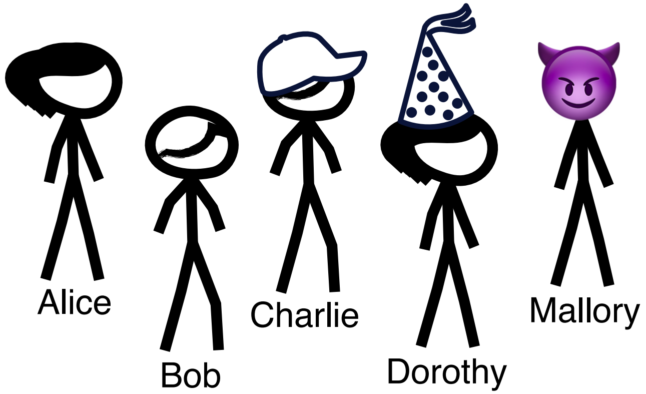
NAKAMOTO-STYLE BLOCKCHAINS

- Parties build a total order using a lottery.

NAKAMOTO-STYLE BLOCKCHAINS

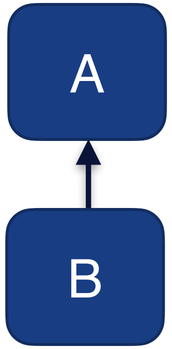
- Parties build a total order using a lottery.

A



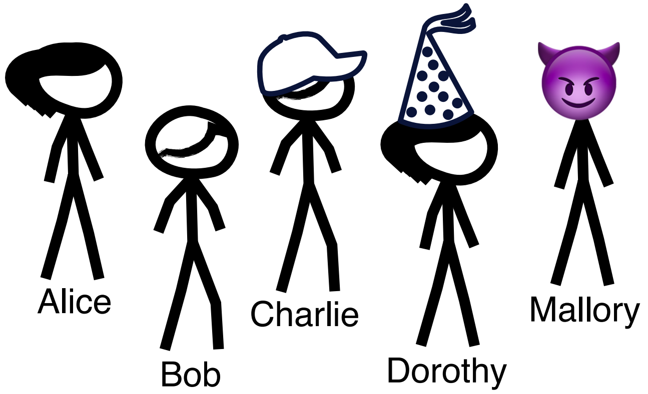
NAKAMOTO-STYLE BLOCKCHAINS

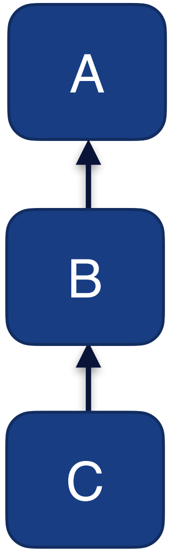
- Parties build a total order using a lottery.



NAKAMOTO-STYLE BLOCKCHAINS

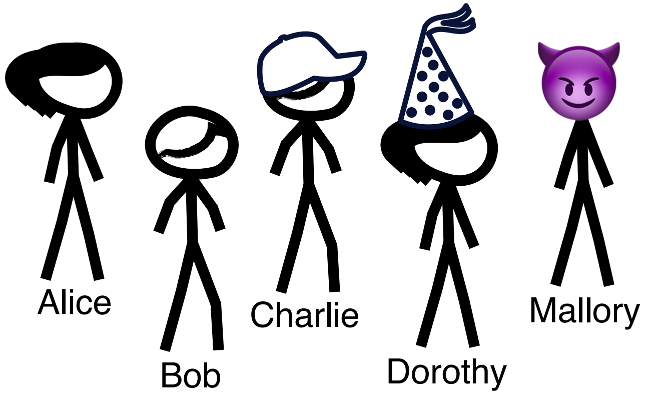
- Parties build a total order using a lottery.

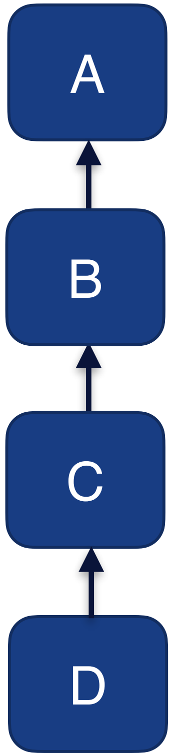




NAKAMOTO-STYLE BLOCKCHAINS

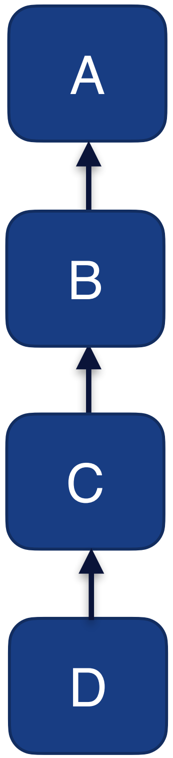
- Parties build a total order using a lottery.





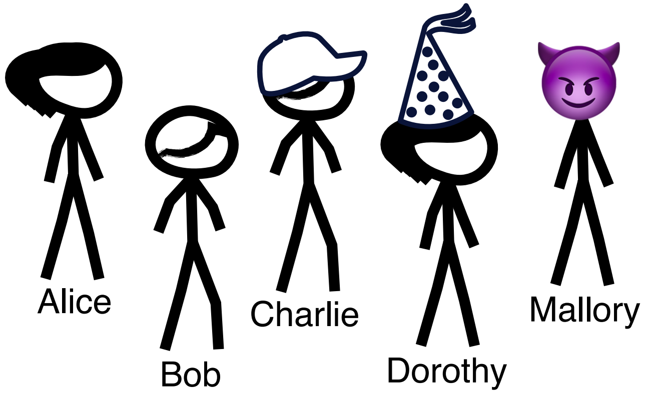
NAKAMOTO-STYLE BLOCKCHAINS

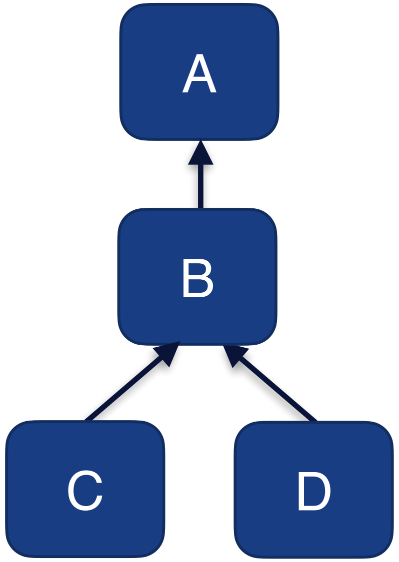
- Parties build a total order using a lottery.
- If a party wins the lottery without knowing about a previous extension of the order, a fork is created.



NAKAMOTO-STYLE BLOCKCHAINS

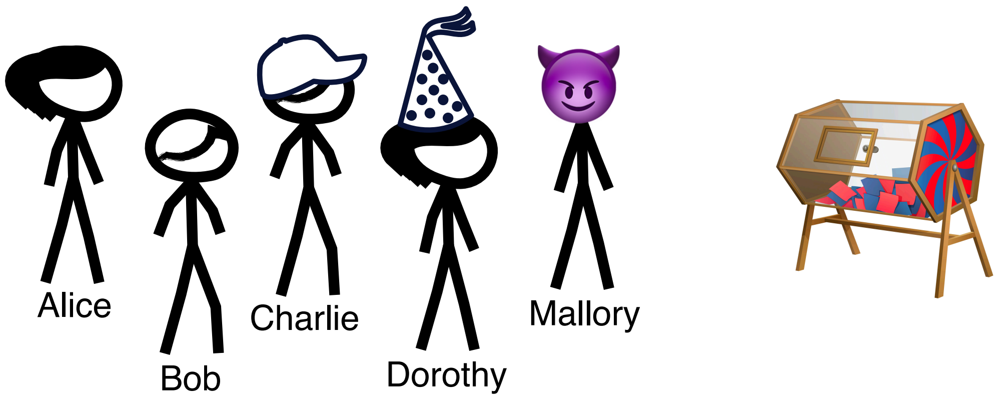
- Parties build a total order using a lottery.
- If a party wins the lottery without knowing about a previous extension of the order, a fork is created.

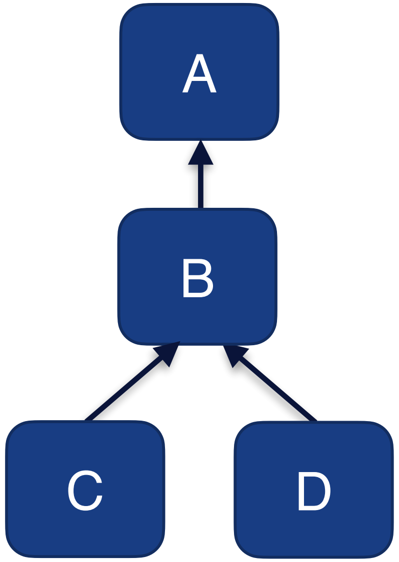




NAKAMOTO-STYLE BLOCKCHAINS

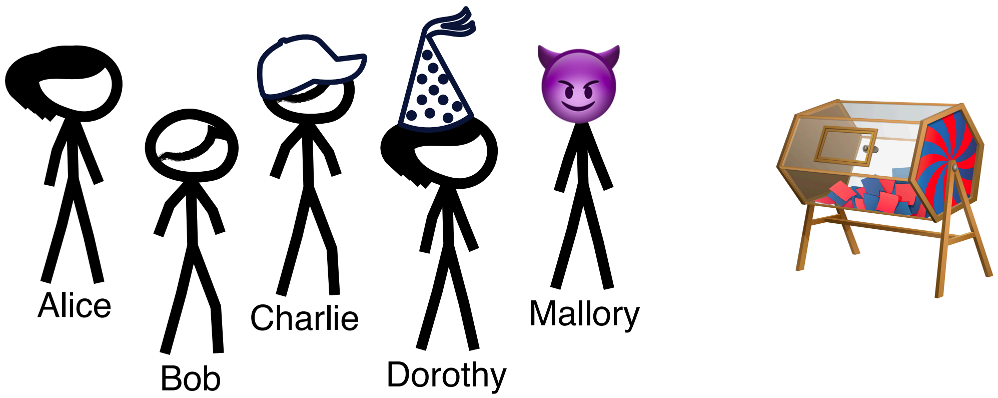
- Parties build a total order using a lottery.
- If a party wins the lottery without knowing about a previous extension of the order, a fork is created.
- Adversaries can also create forks.

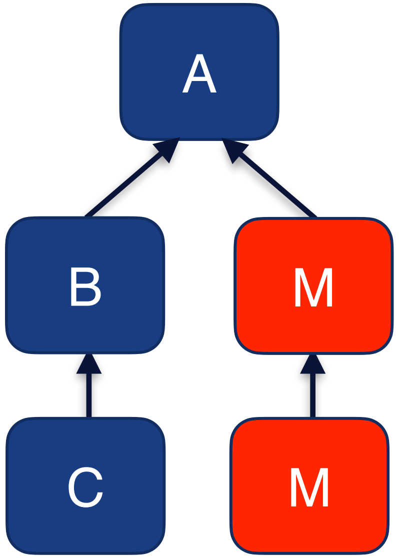




NAKAMOTO-STYLE BLOCKCHAINS

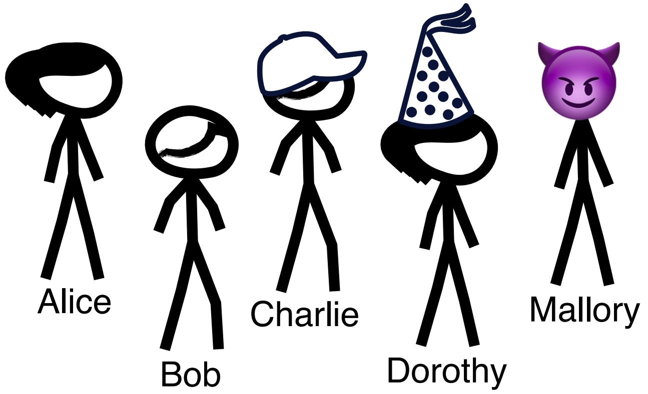
- Parties build a total order using a lottery.
- If a party wins the lottery without knowing about a previous extension of the order, a fork is created.
- Adversaries can also create forks.

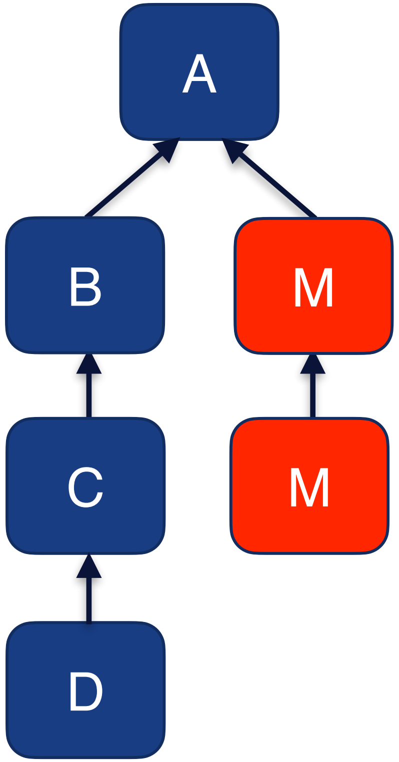




NAKAMOTO-STYLE BLOCKCHAINS

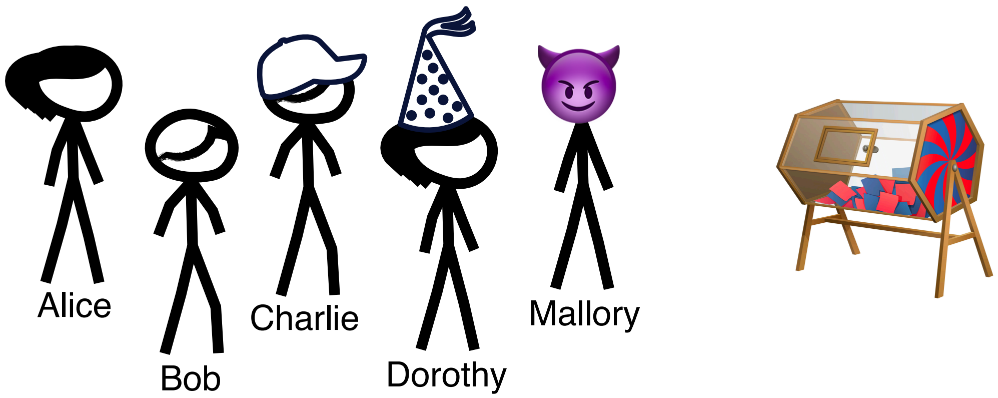
- Parties build a total order using a lottery.
- If a party wins the lottery without knowing about a previous extension of the order, a fork is created.
- Adversaries can also create forks.

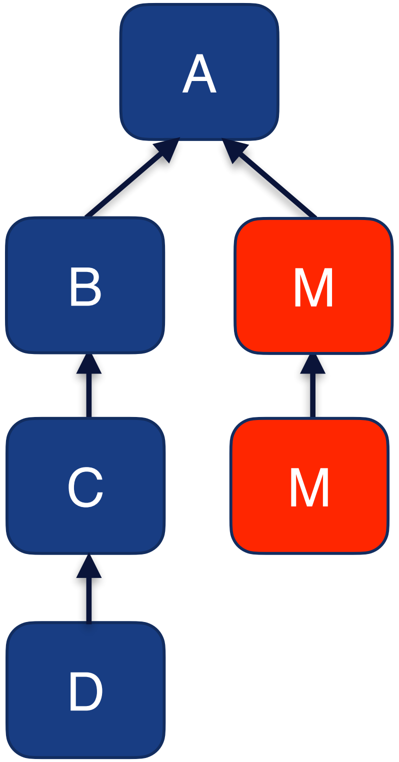




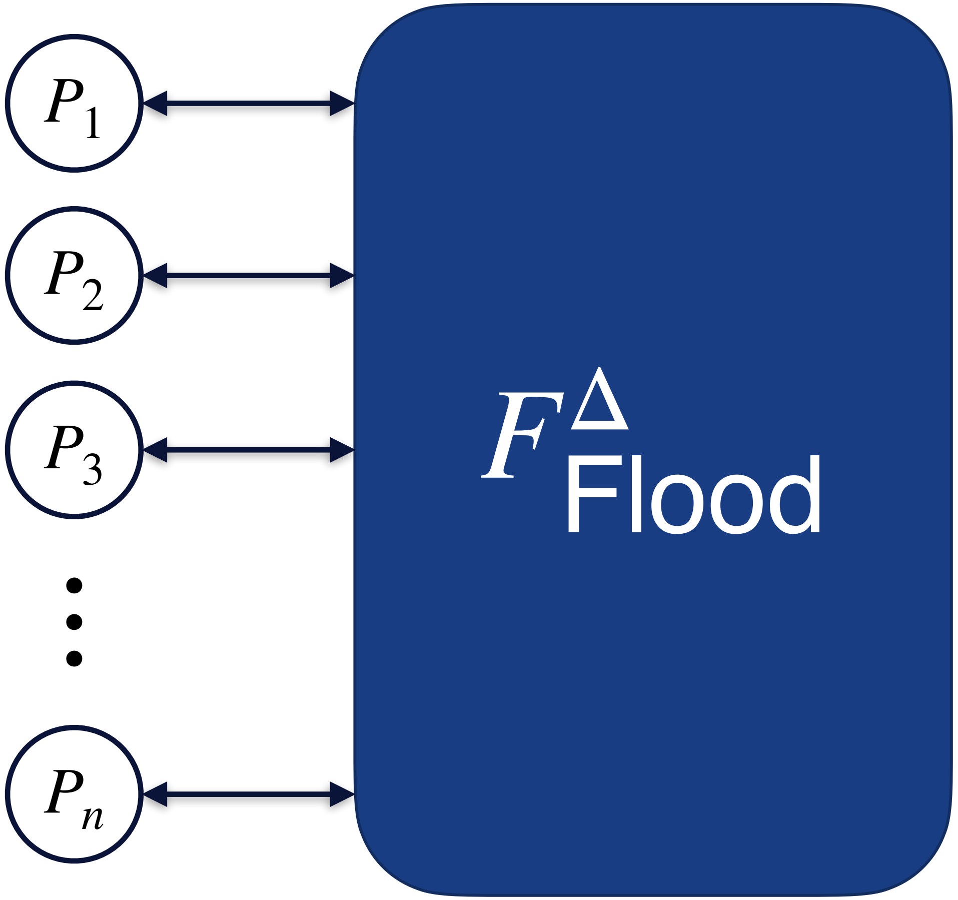
NAKAMOTO-STYLE BLOCKCHAINS

- Parties build a total order using a lottery.
- If a party wins the lottery without knowing about a previous extension of the order, a fork is created.
- Adversaries can also create forks.
- **Isolated honest blocks must outgrow adversarial blocks.**

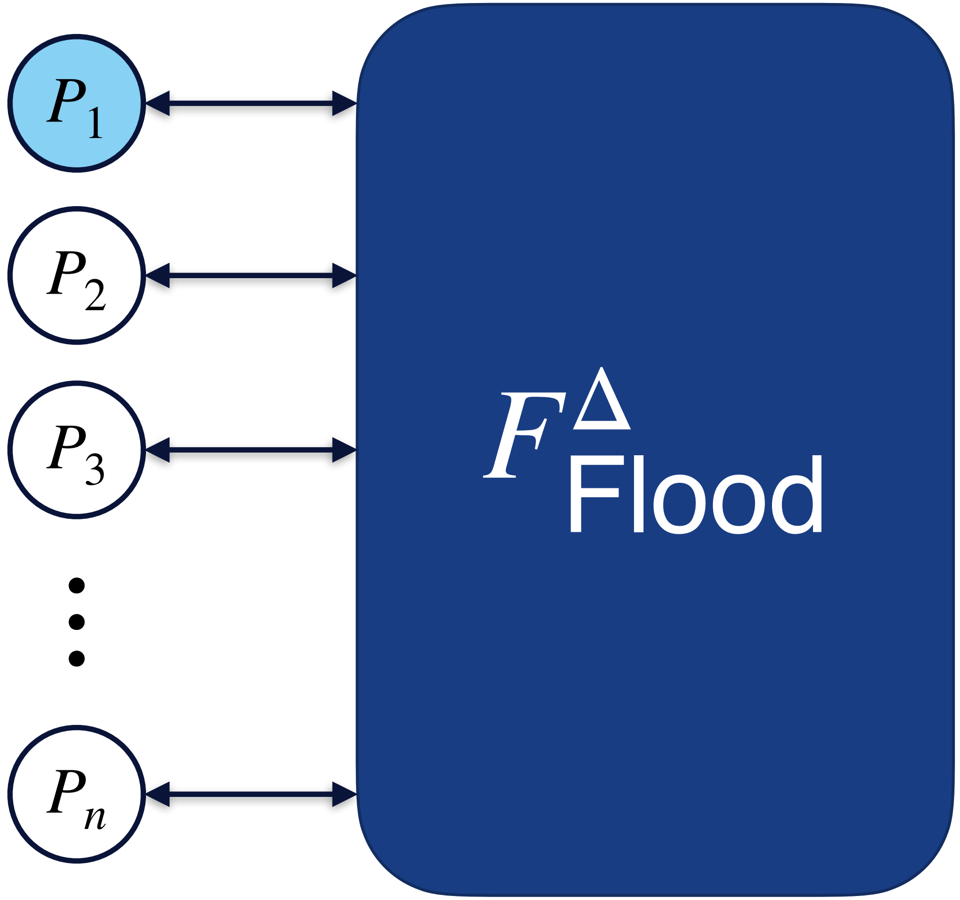




FLOODING FOR NSBS

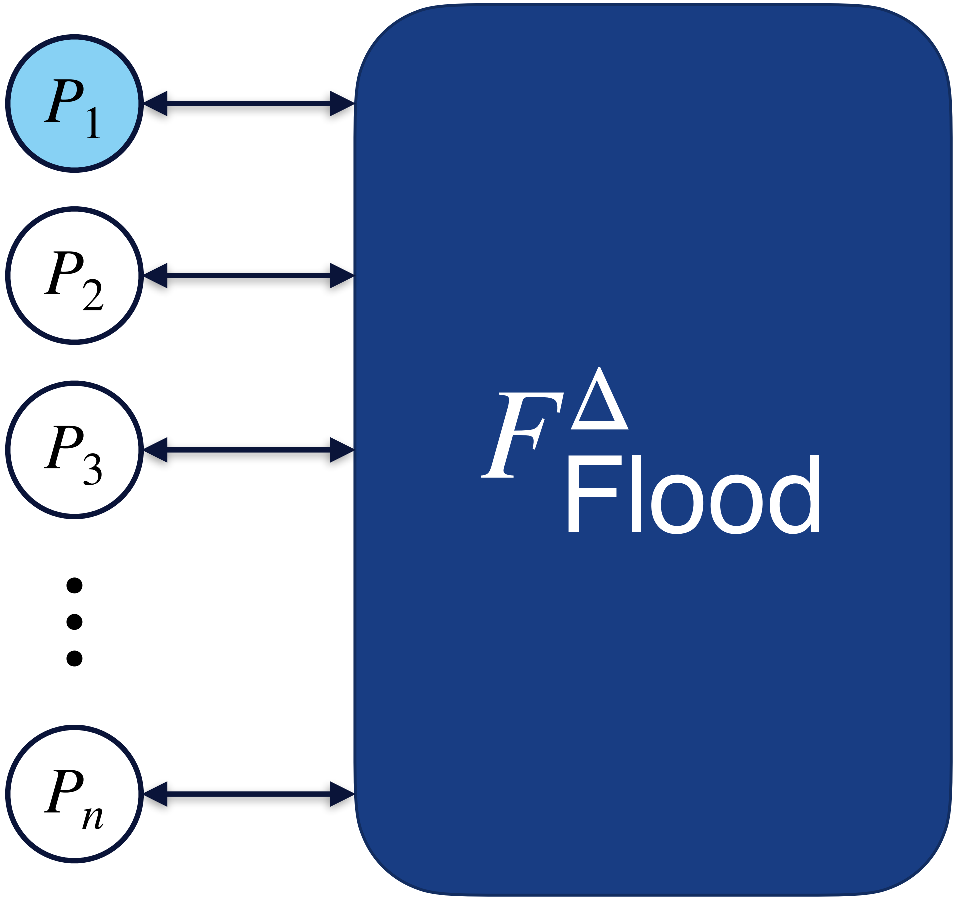


FLOODING FOR NSBS



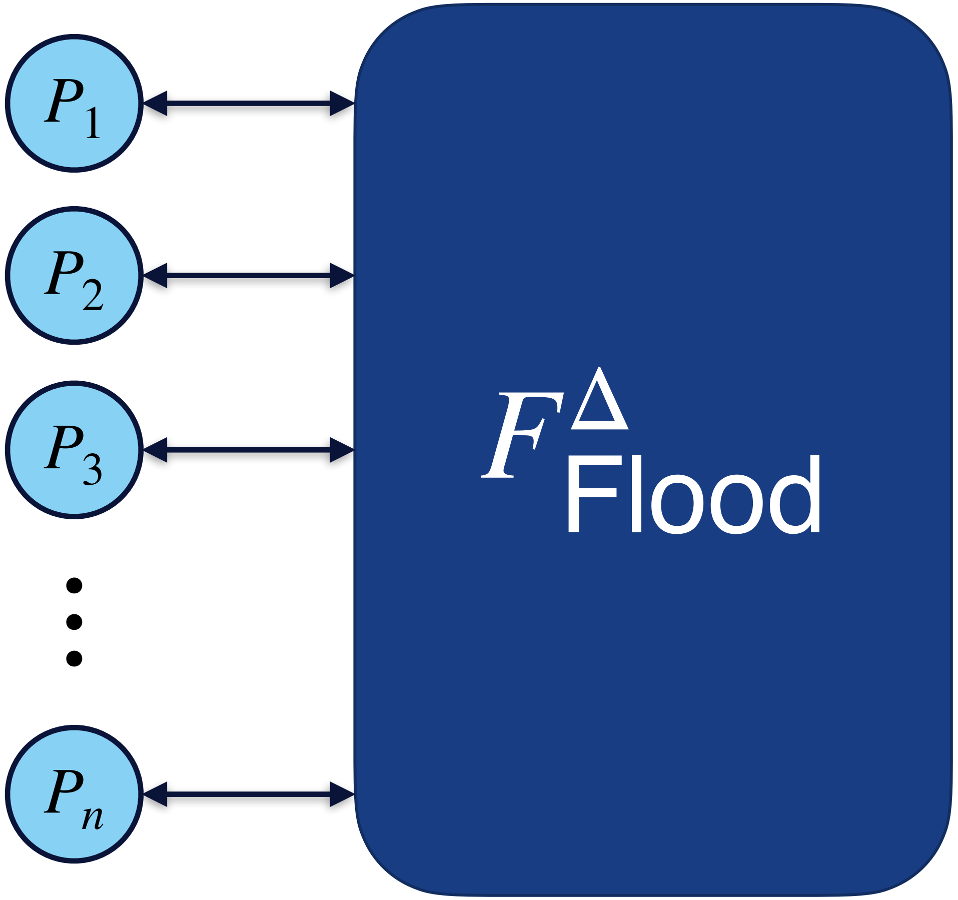
FLOODING FOR NSBS

- Any message input at time t must be delivered before time $t + \Delta$.



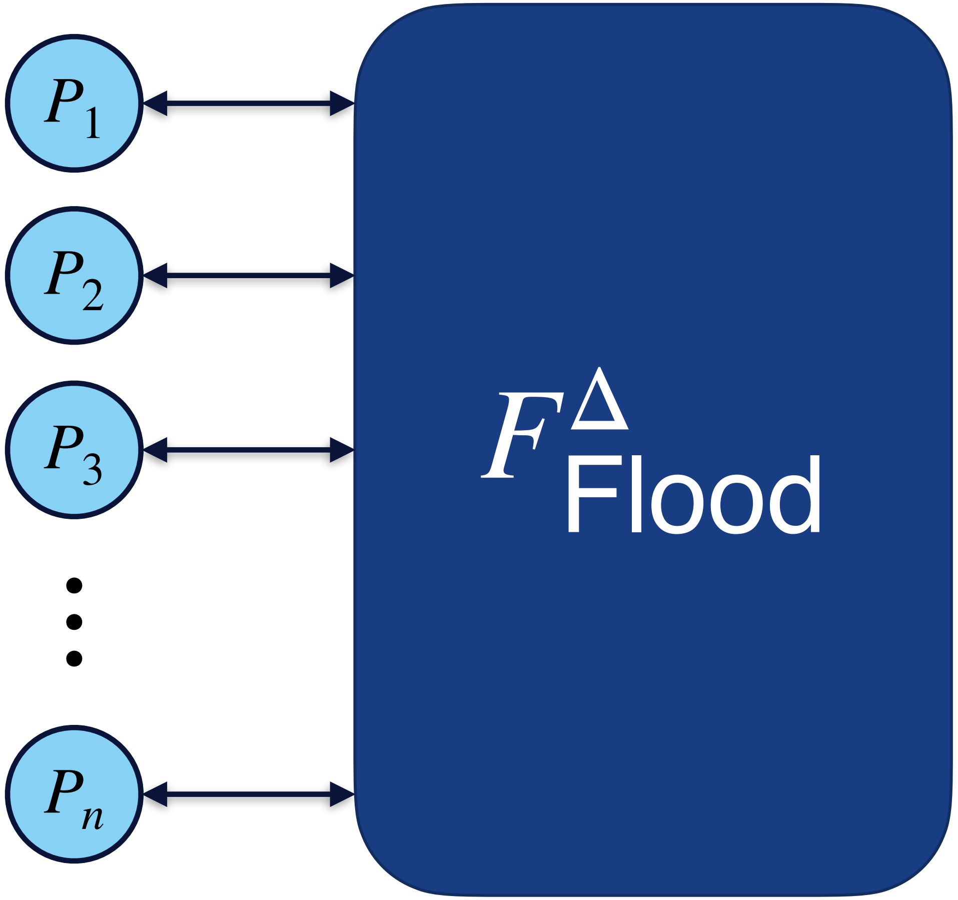
FLOODING FOR NSBS

- Any message input at time t must be delivered before time $t + \Delta$.

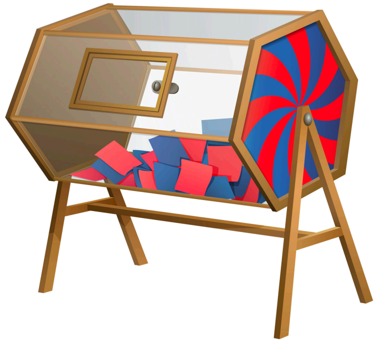


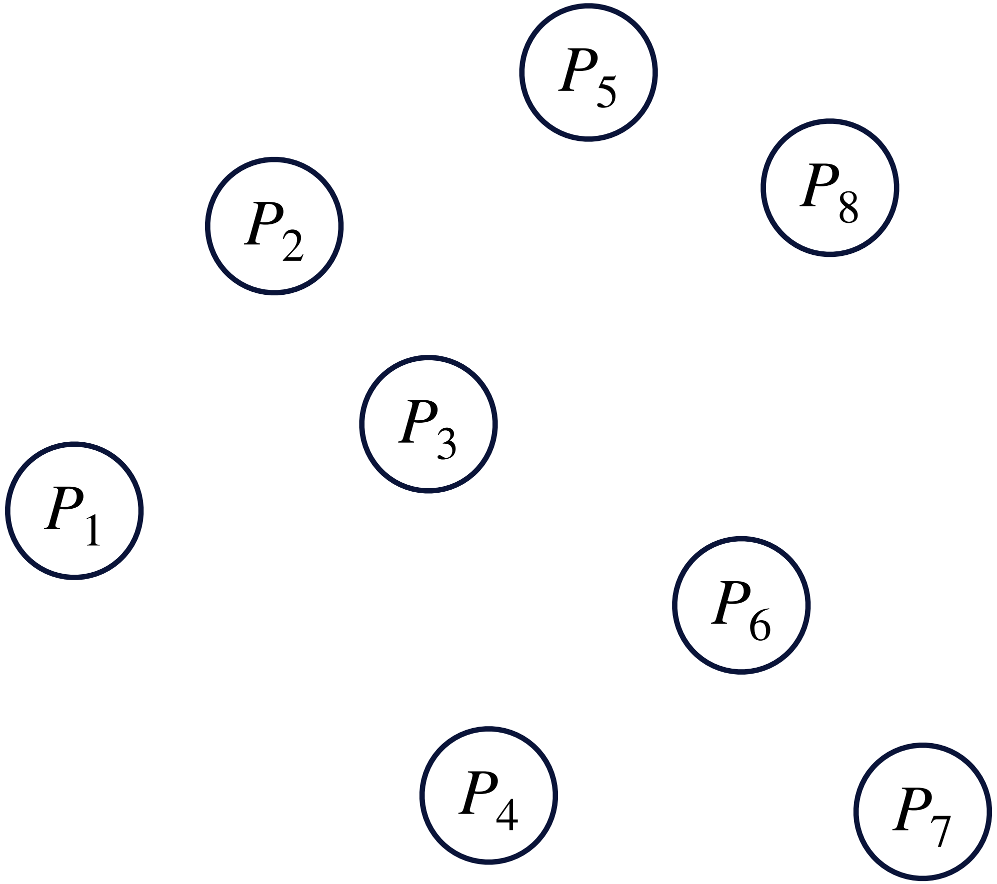
FLOODING FOR NSBS

- Any message input at time t must be delivered before time $t + \Delta$.
- Assumed to prove NSBs secure [GKL15,GKL17,PSs17,DGKR18].

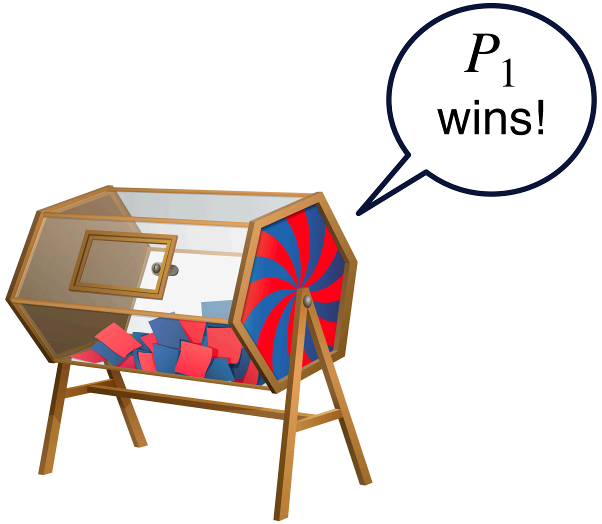


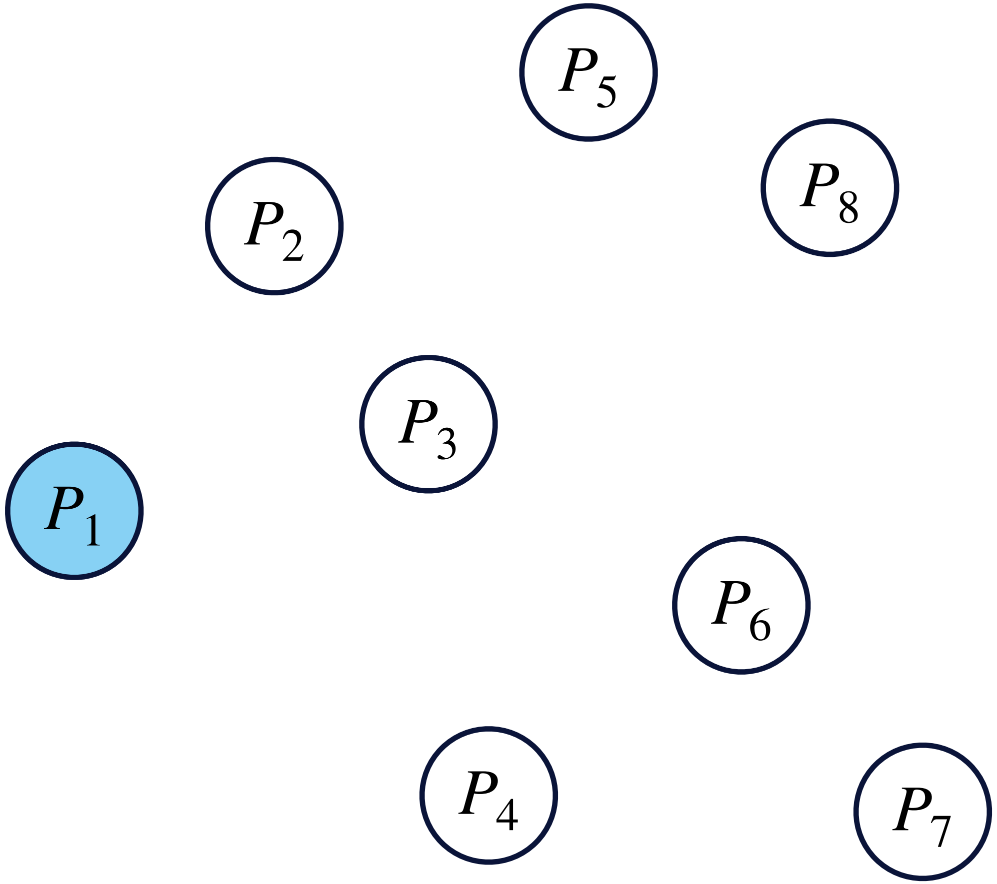
FLOODING FOR NSBS IN PRACTICE



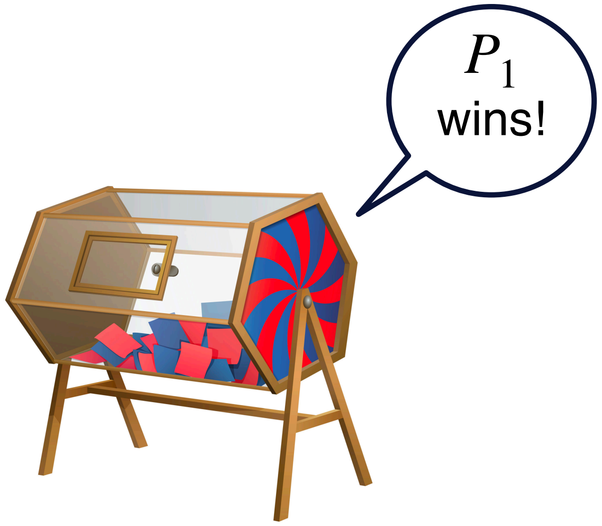


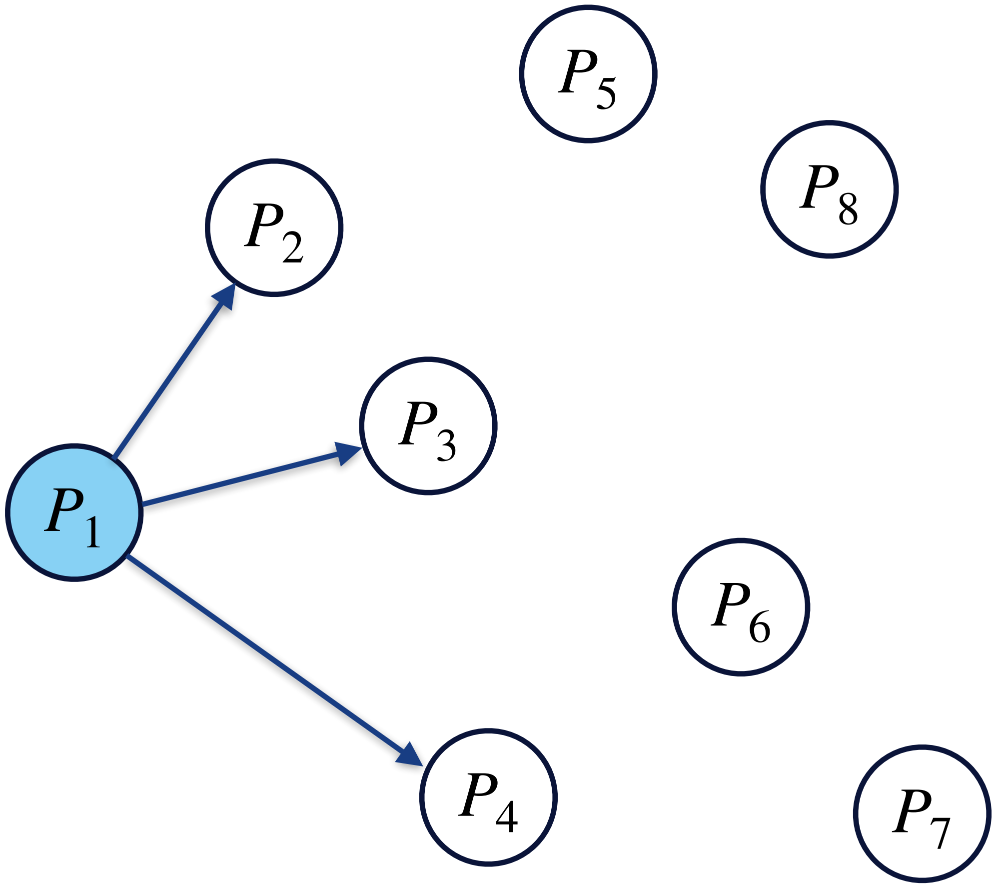
FLOODING FOR NSBS IN PRACTICE



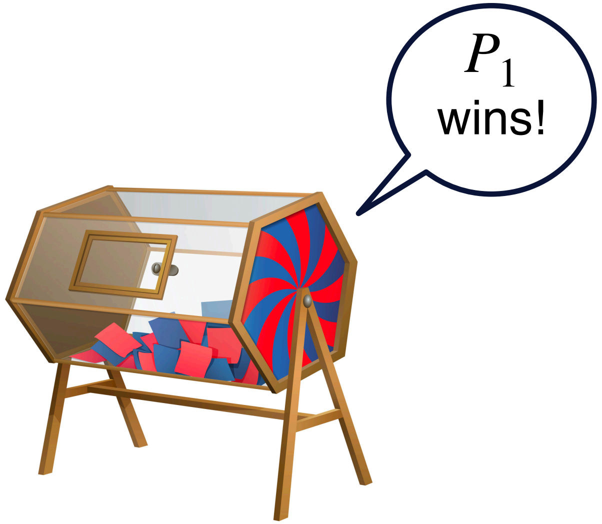


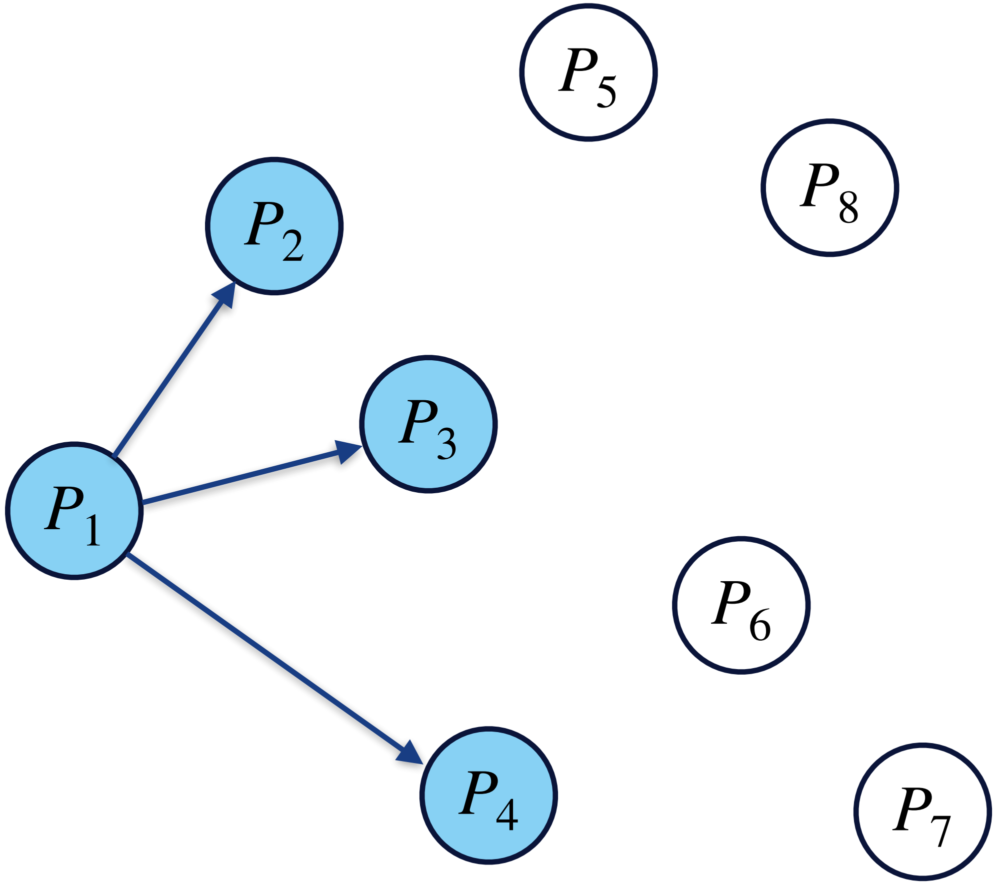
FLOODING FOR NSBS IN PRACTICE



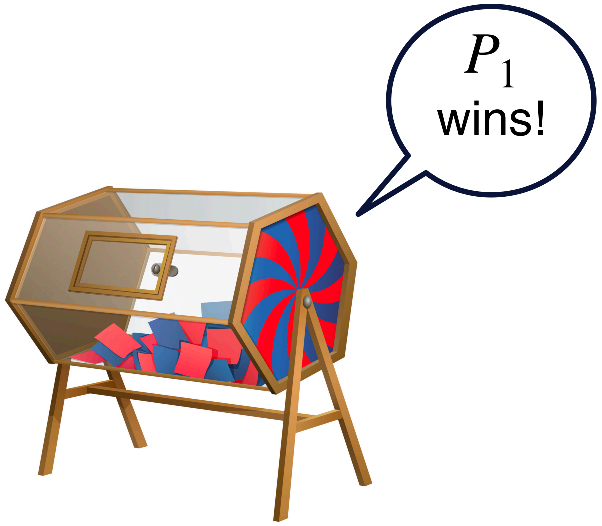


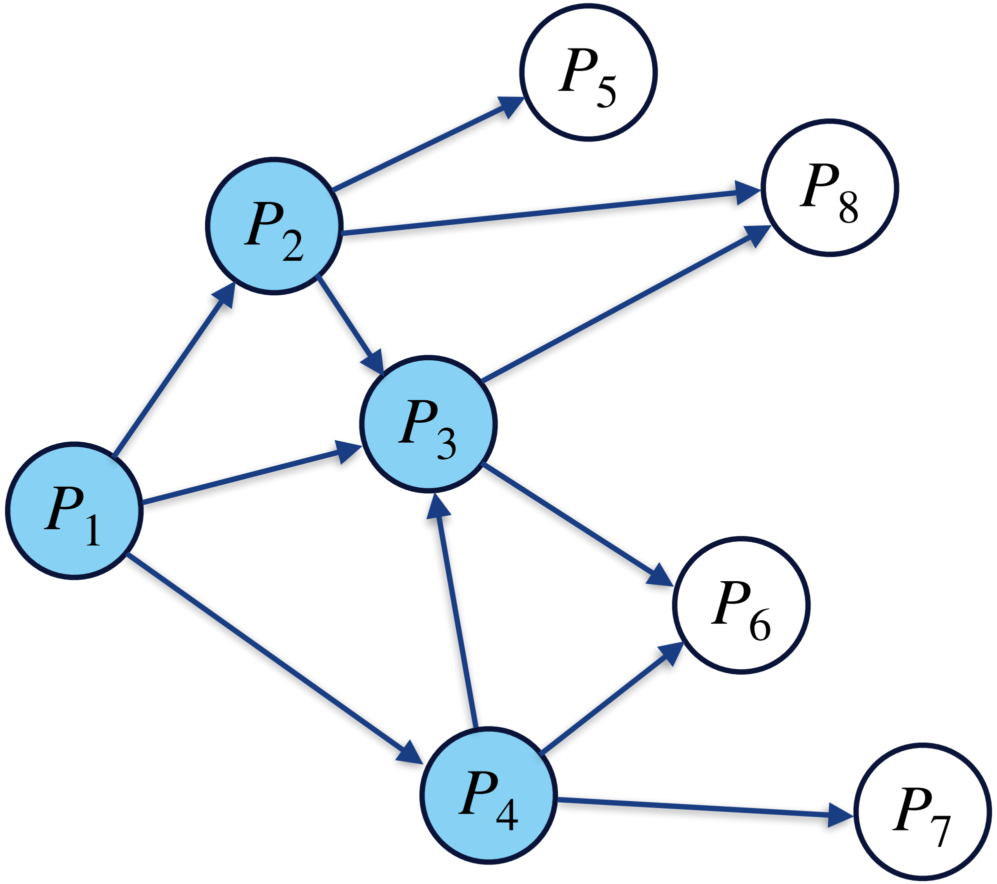
FLOODING FOR NSBS IN PRACTICE



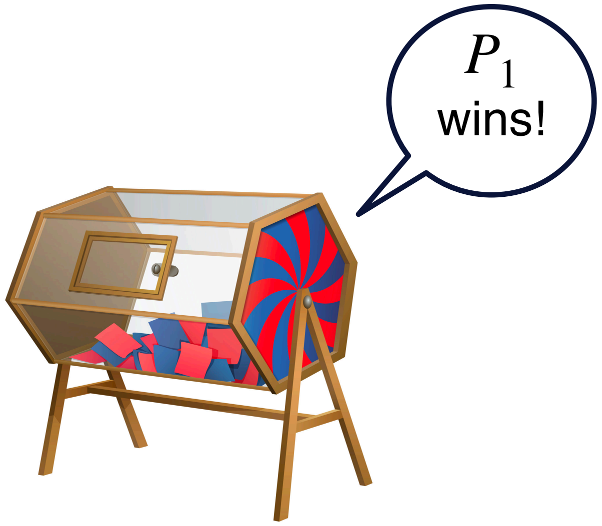


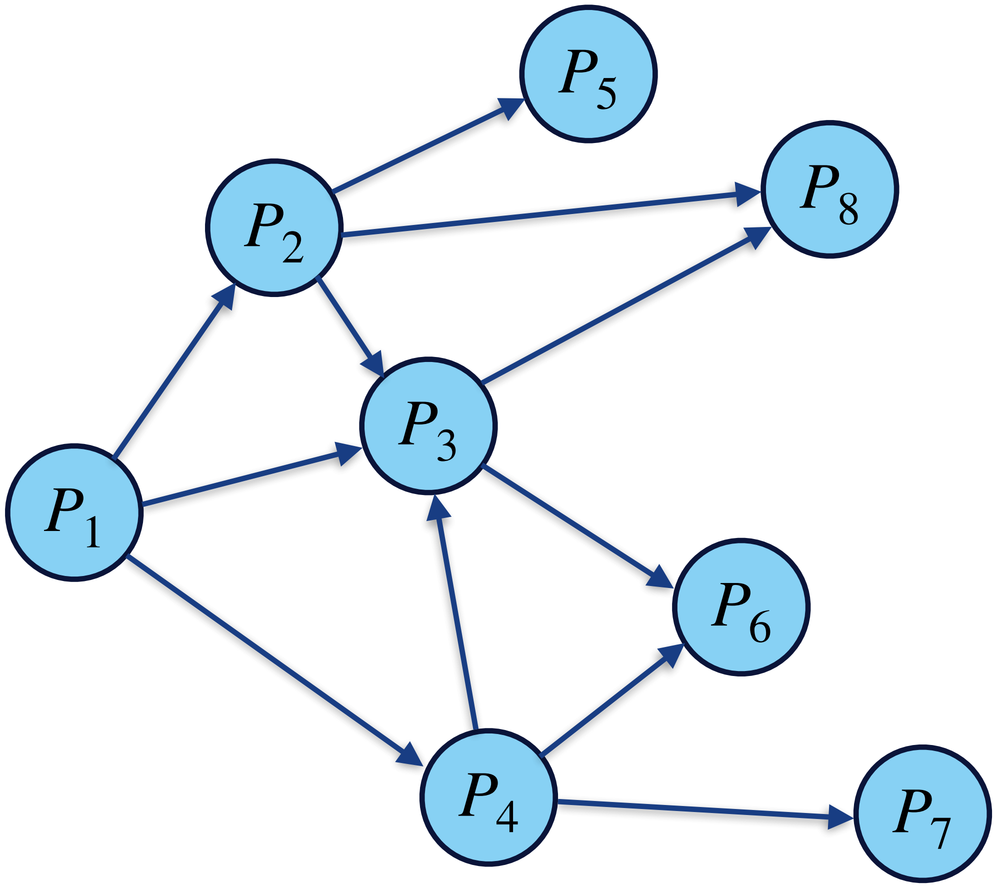
FLOODING FOR NSBS IN PRACTICE



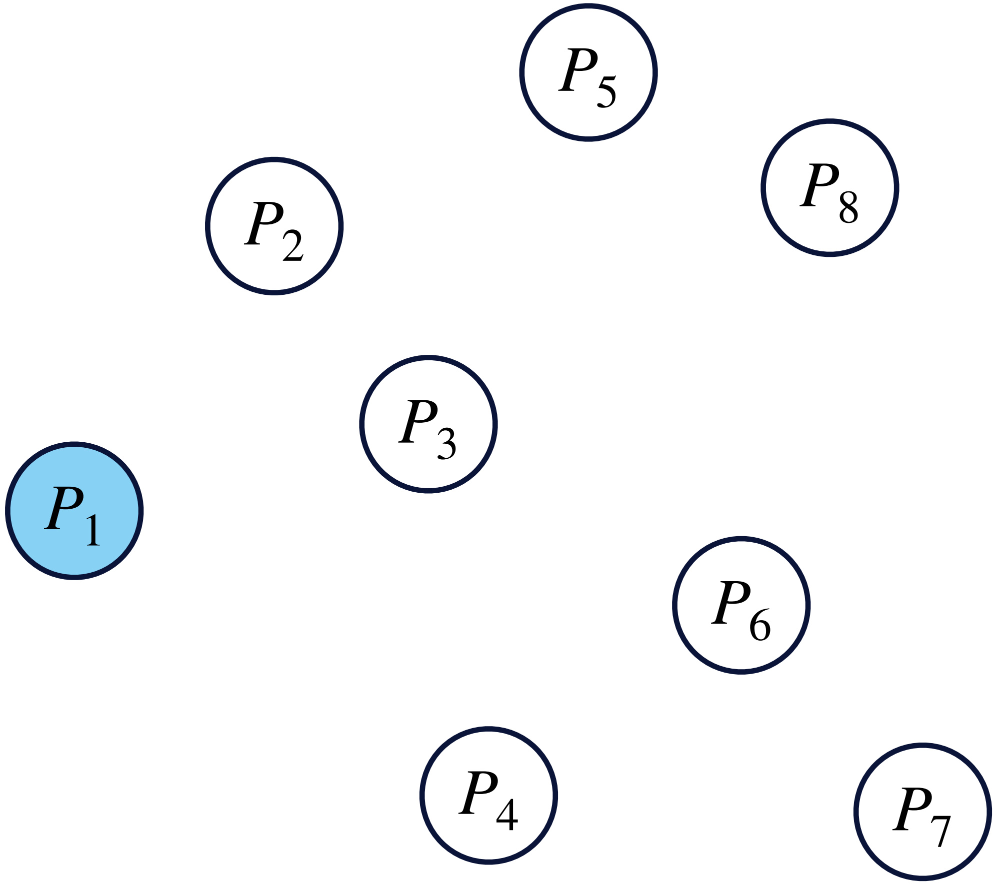


FLOODING FOR NSBS IN PRACTICE

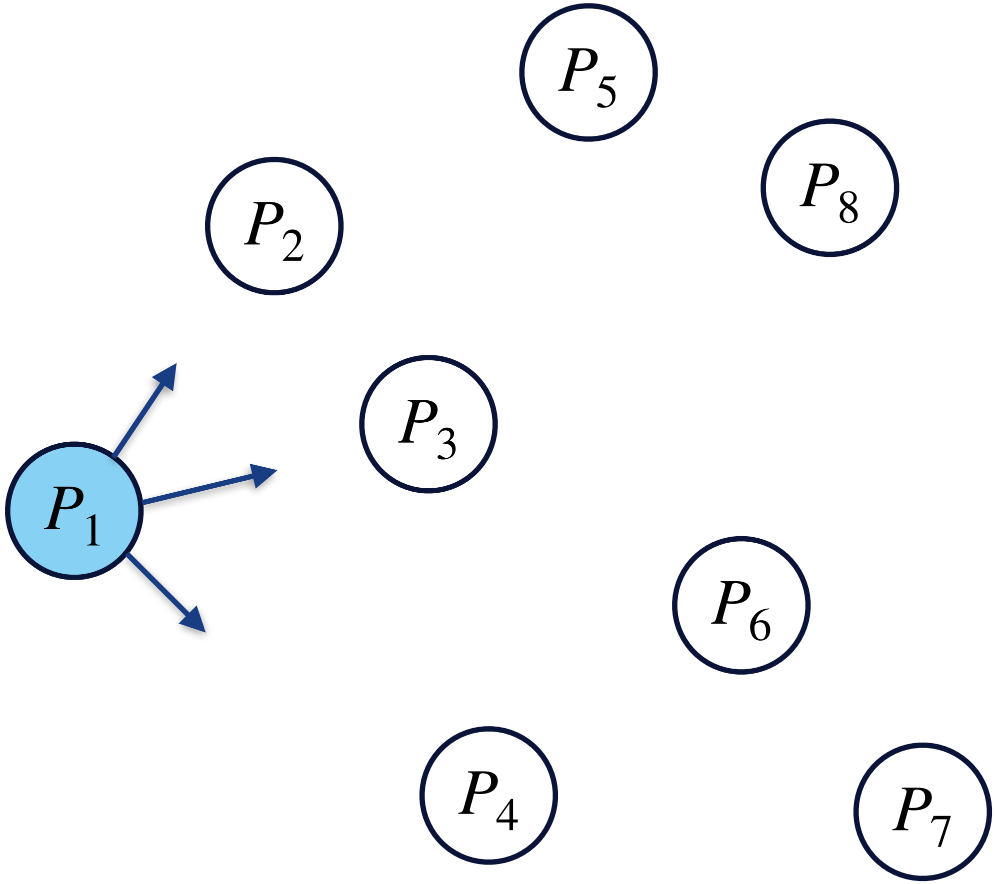




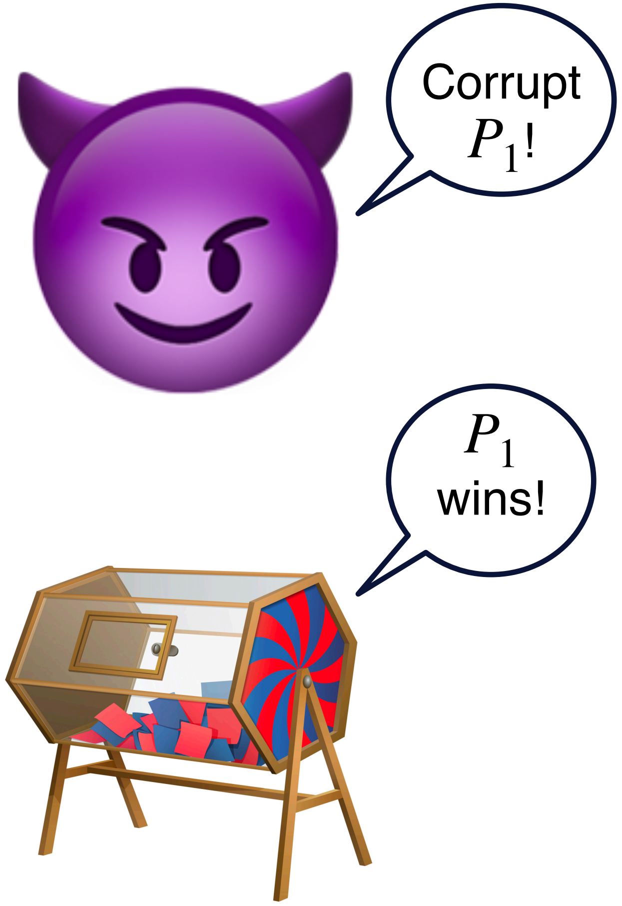
FLOODING WITH AN *ADAPTIVE* ADVERSARY AND *NON-ATOMIC MESSAGE SEND*

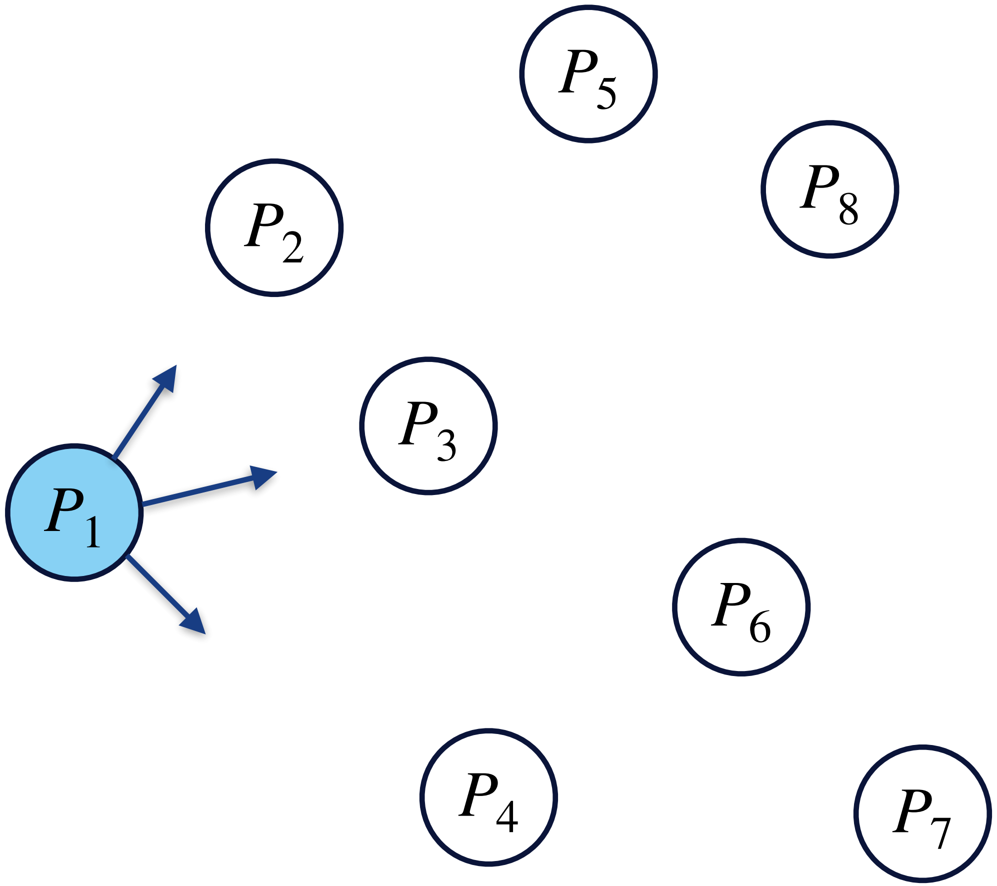


FLOODING WITH AN *ADAPTIVE* ADVERSARY AND *NON-ATOMIC MESSAGE SEND*

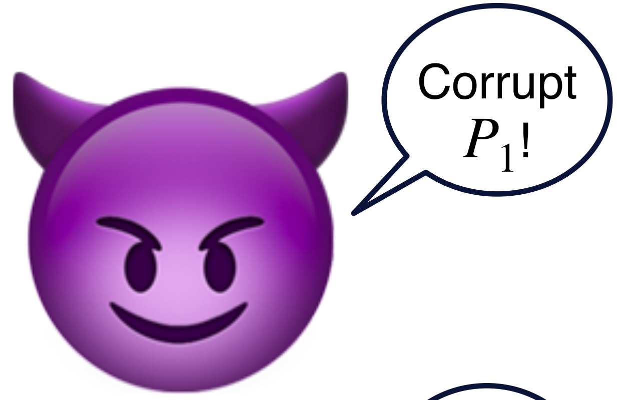


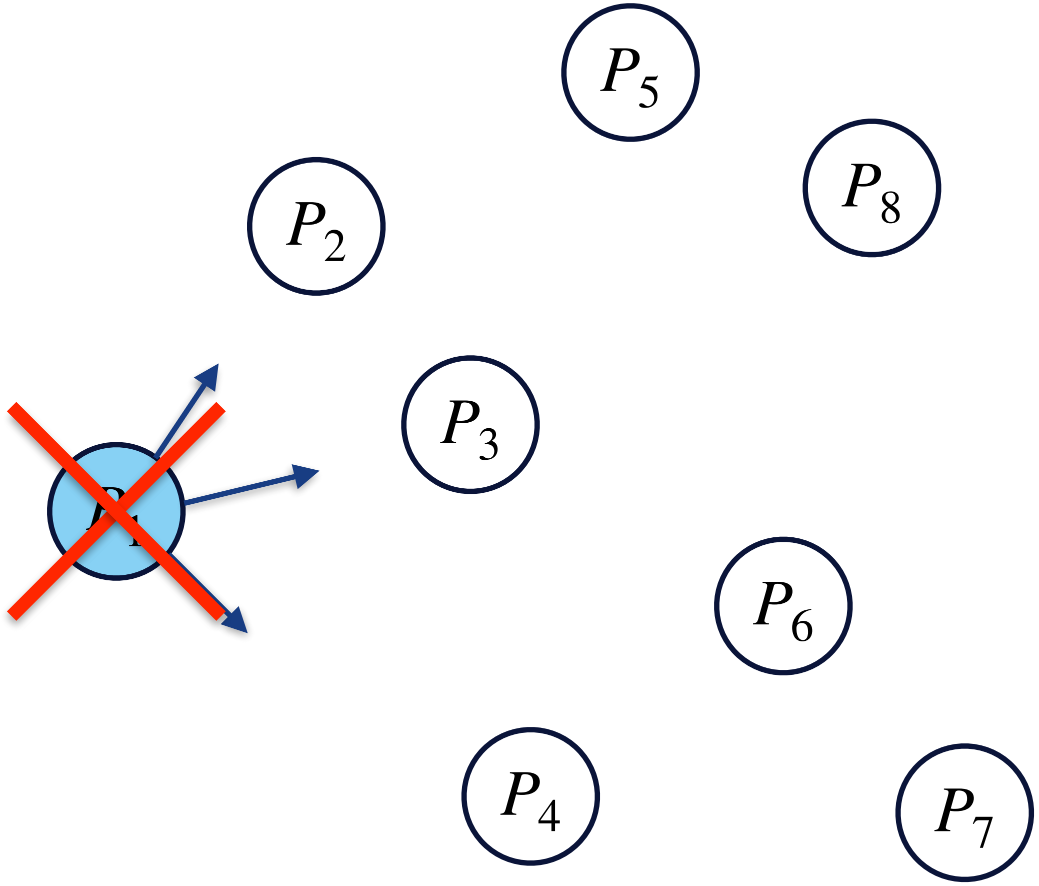
FLOODING WITH AN *ADAPTIVE* ADVERSARY AND *NON-ATOMIC MESSAGE SEND*



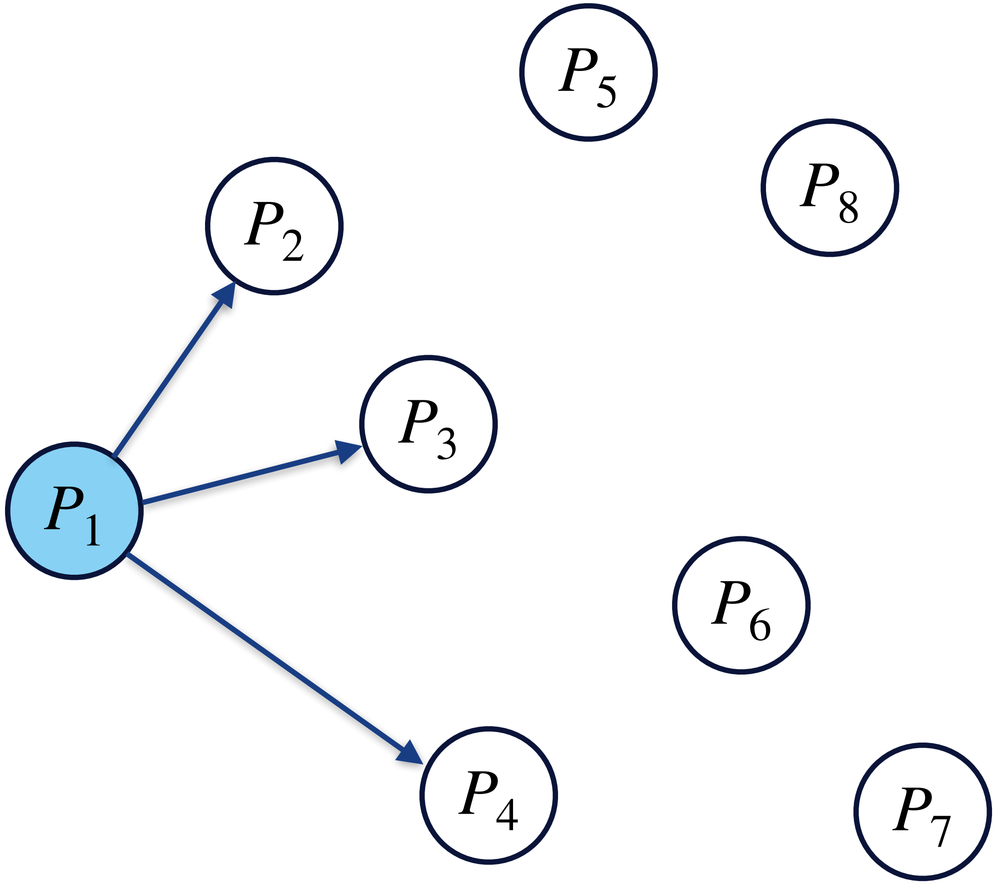


FLOODING WITH AN *ADAPTIVE* ADVERSARY AND *NON-ATOMIC MESSAGE SEND*

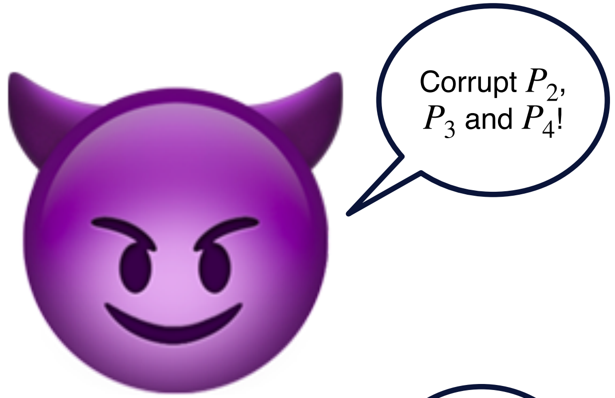


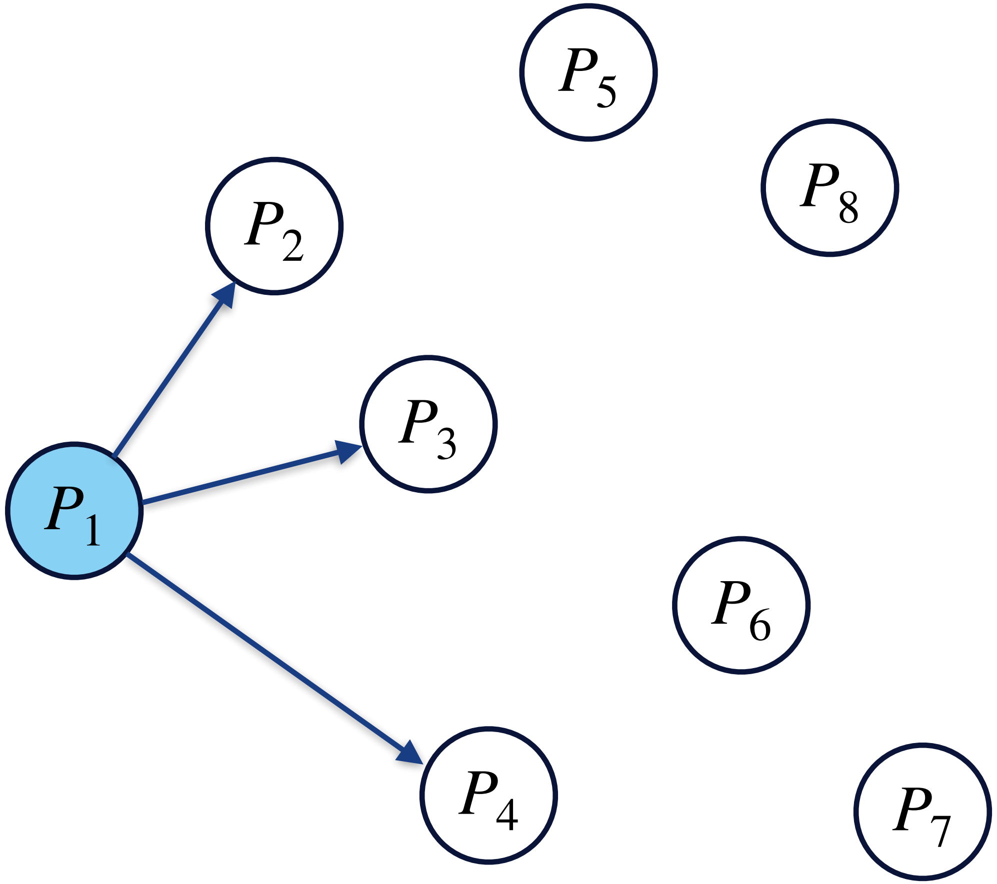


FLOODING WITH AN *ADAPTIVE* ADVERSARY AND *ATOMIC MESSAGE SEND*

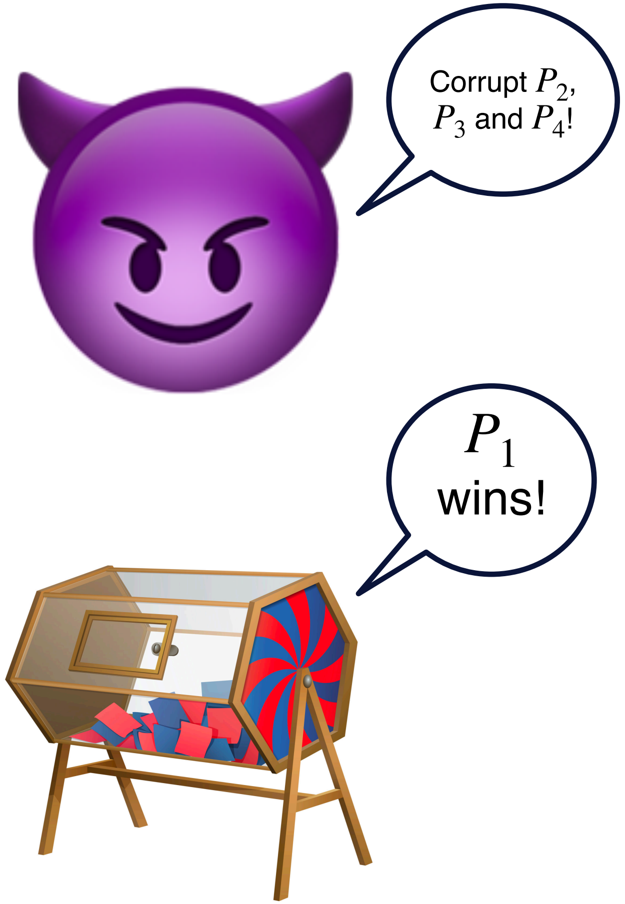


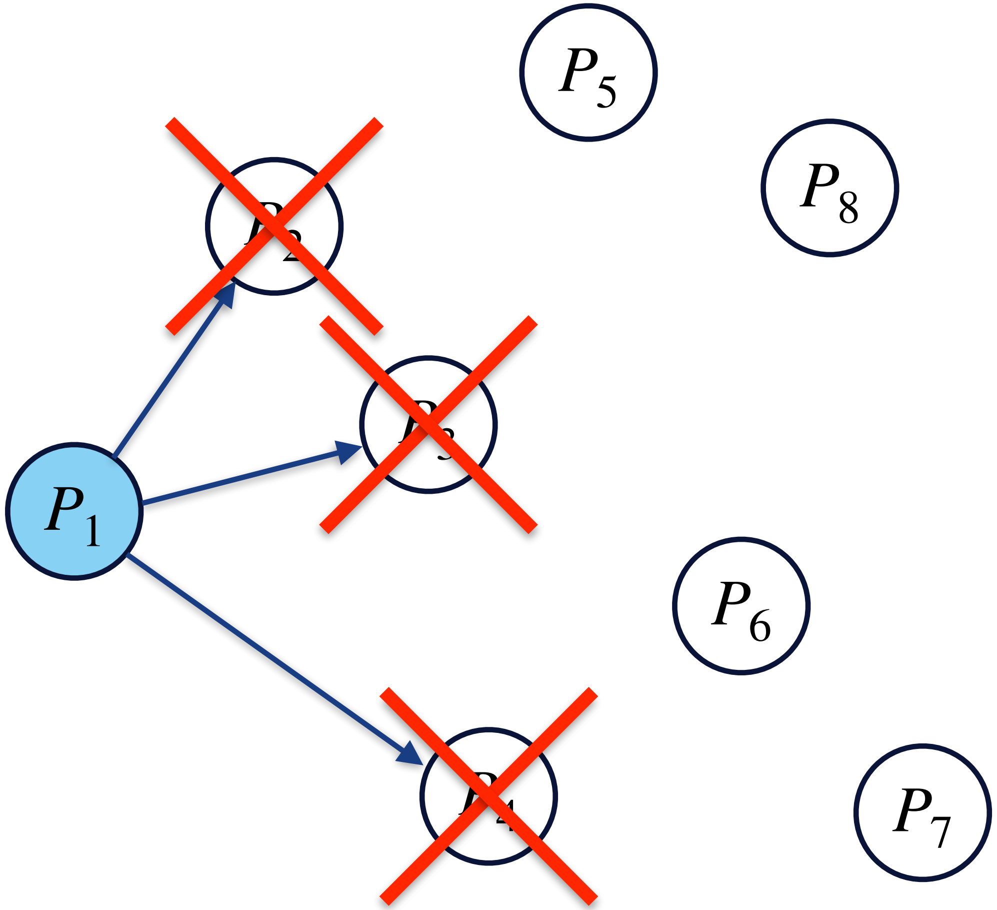
FLOODING WITH AN *ADAPTIVE* ADVERSARY AND *ATOMIC MESSAGE SEND*



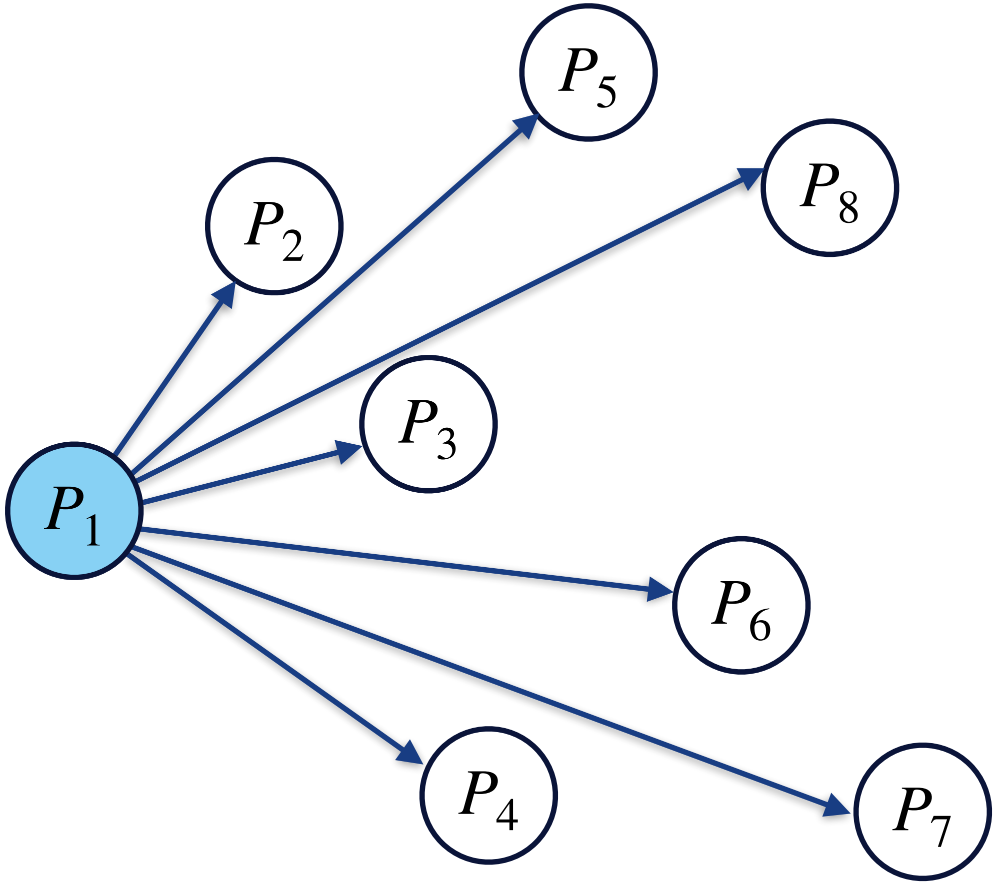


FLOODING WITH AN *ADAPTIVE* ADVERSARY AND *ATOMIC MESSAGE SEND*

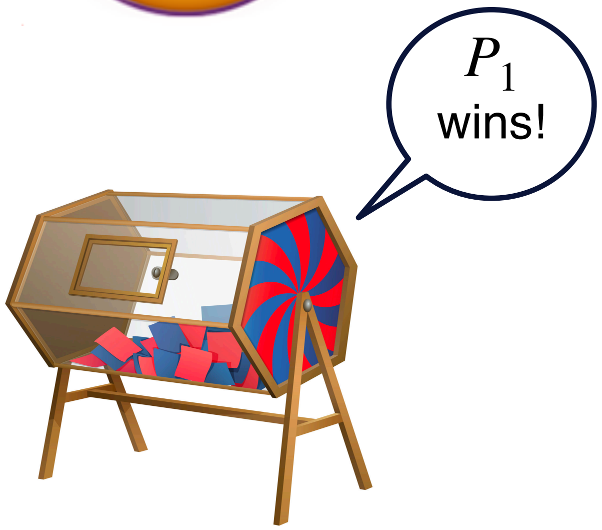


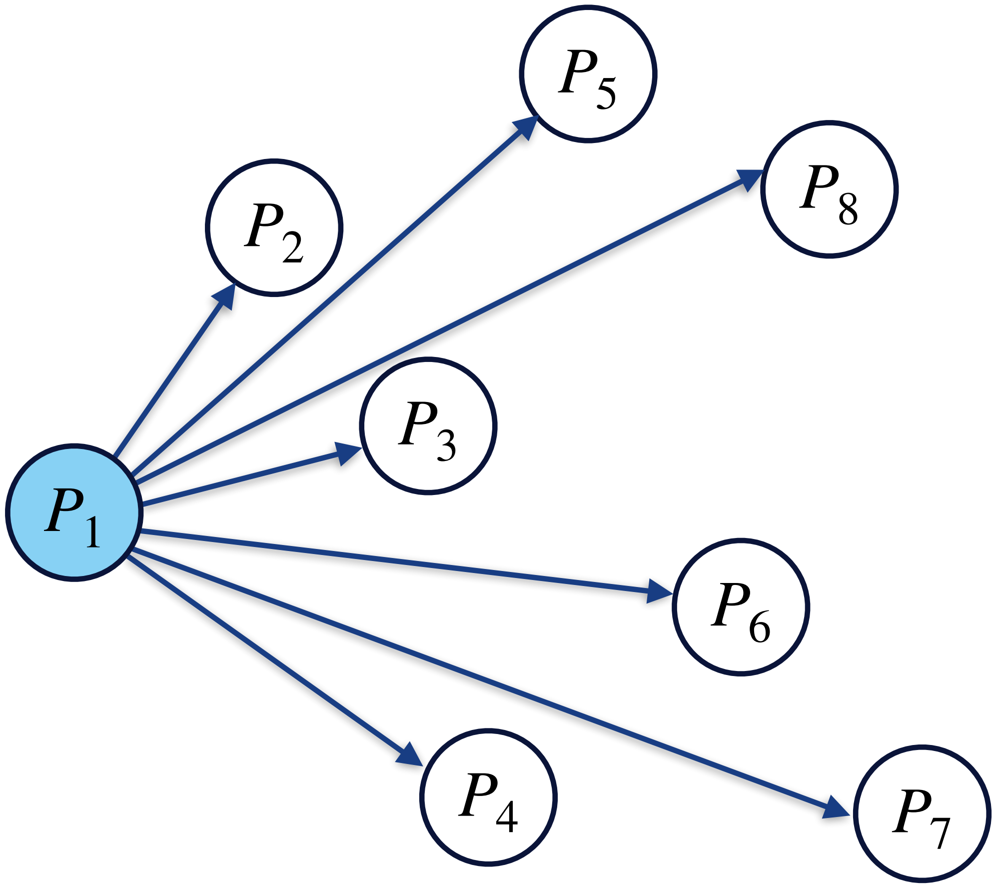


FLOODING WITH AN *ADAPTIVE* ADVERSARY AND *ATOMIC MESSAGE SEND*

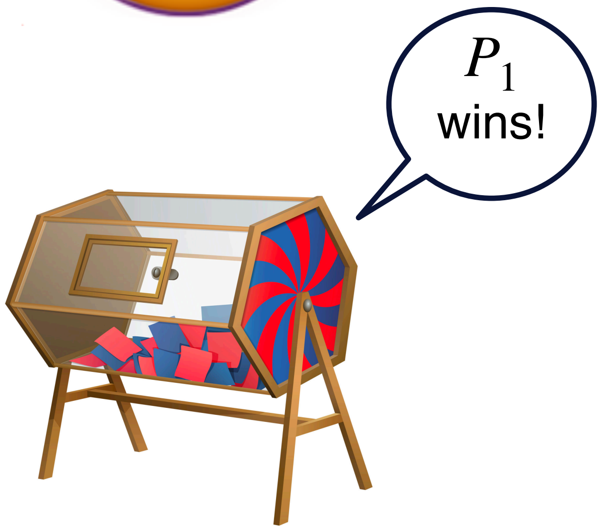


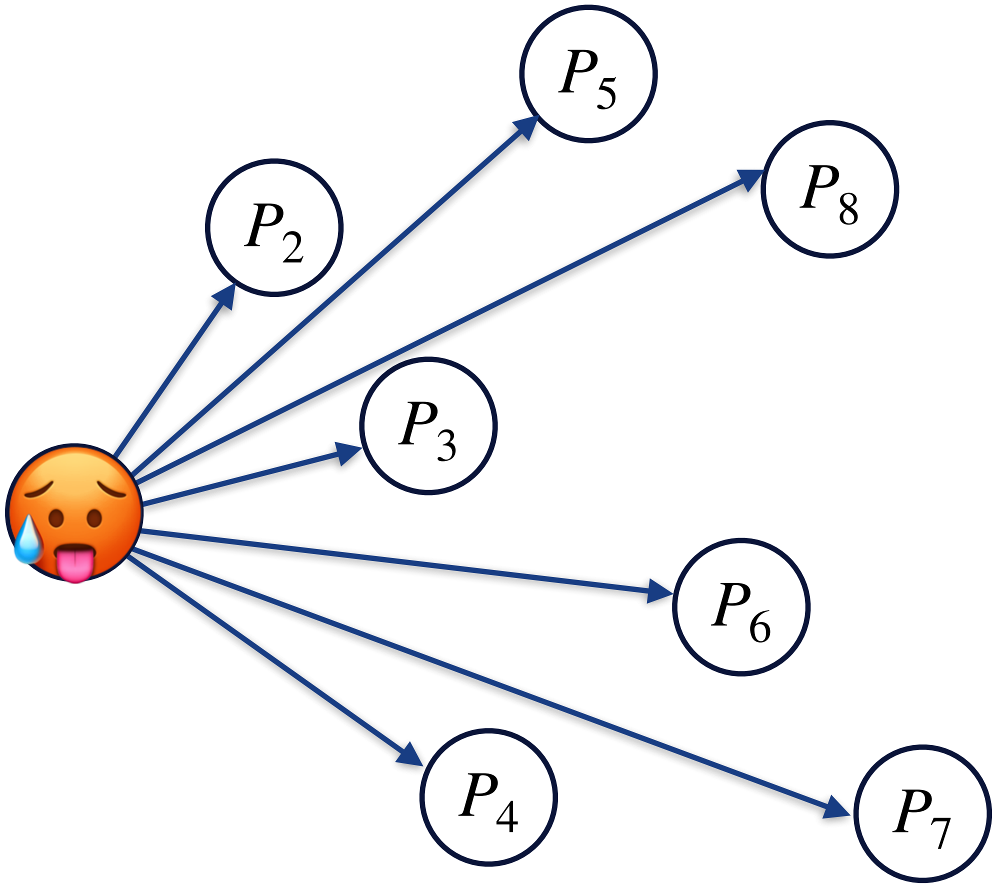
FLOODING WITH AN *ADAPTIVE* ADVERSARY AND *ATOMIC MESSAGE SEND*





FLOODING WITH AN *ADAPTIVE* ADVERSARY AND *ATOMIC MESSAGE SEND*

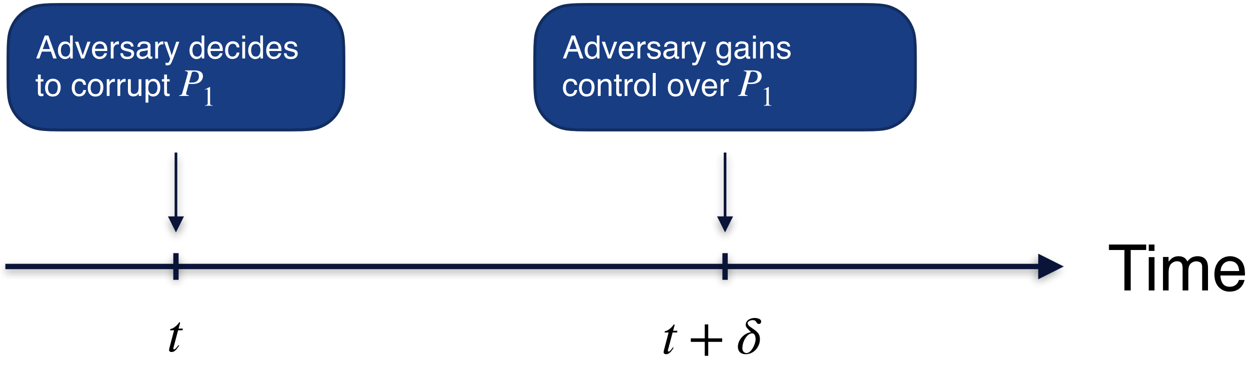




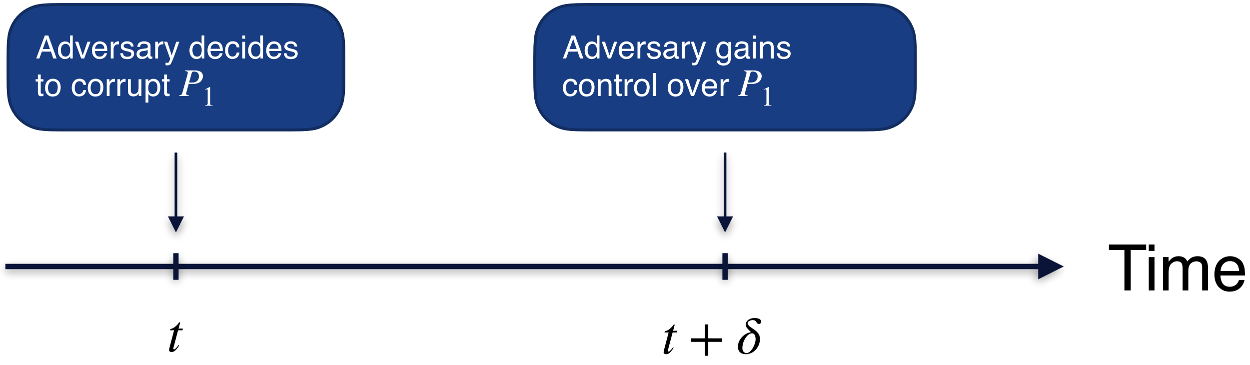
δ-DELAYED ADVERSARIES

δ-DELAYED ADVERSARIES

δ-DELAYED ADVERSARIES

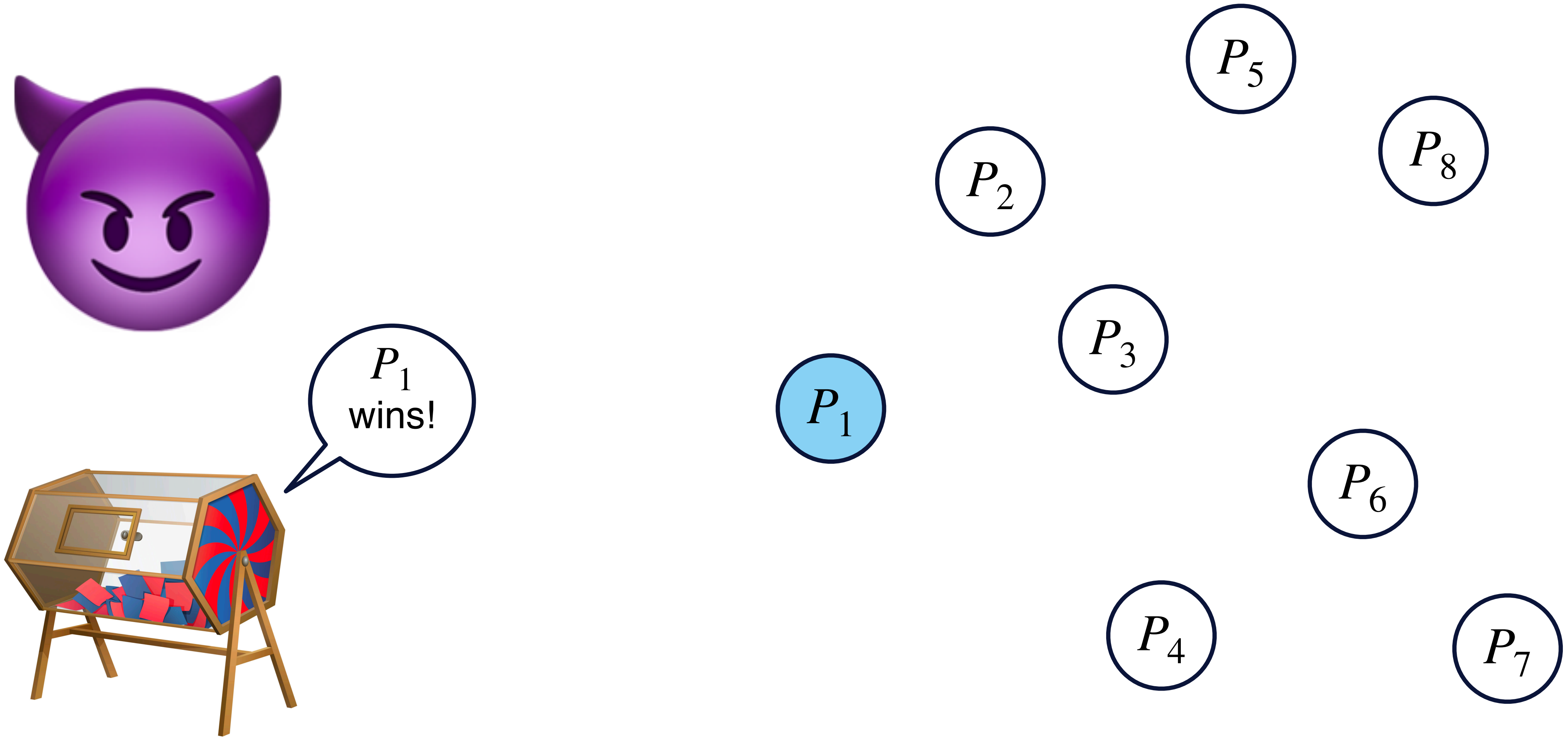


δ-DELAYED ADVERSARIES

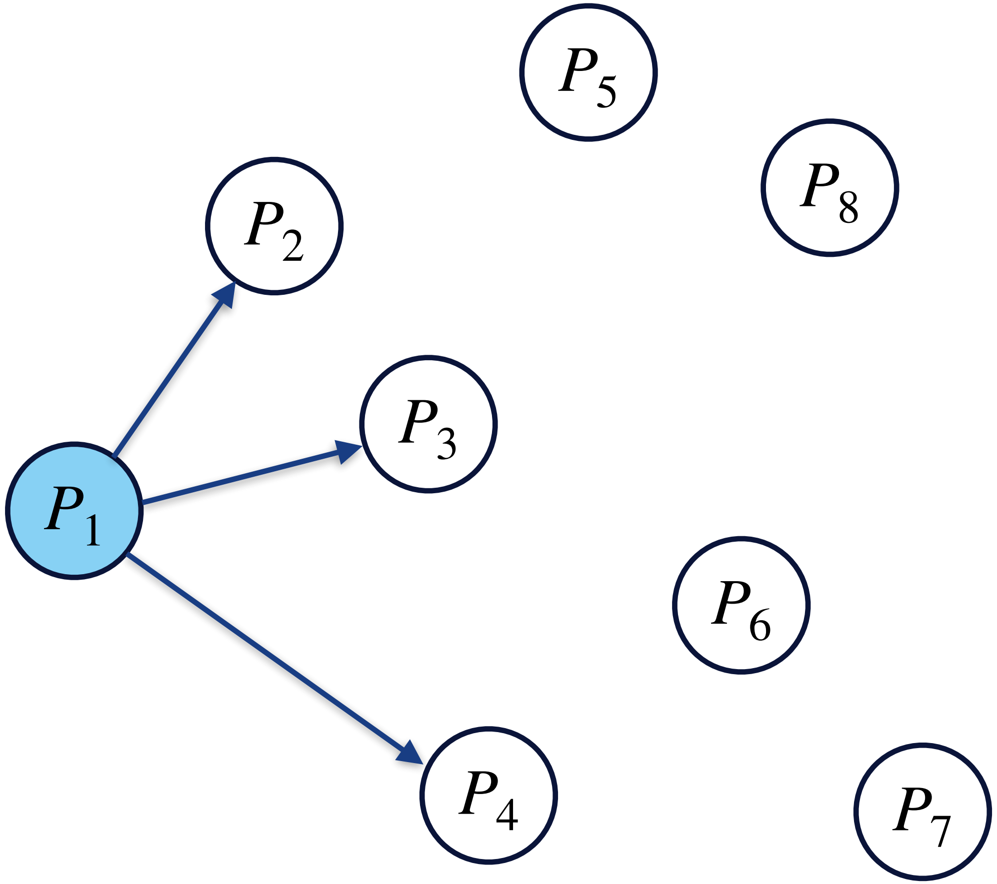


- Informally introduced by [PS17] for long-lived committees.

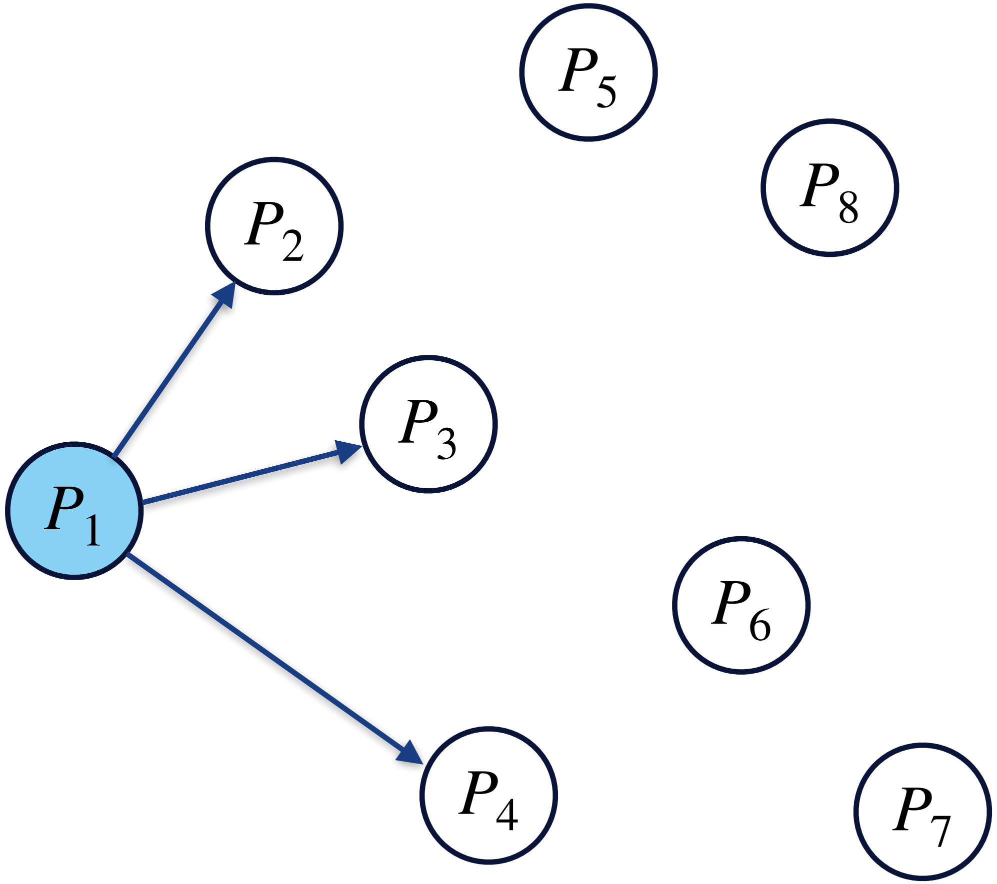
FLOODING WITH A DELAYED ADVERSARY



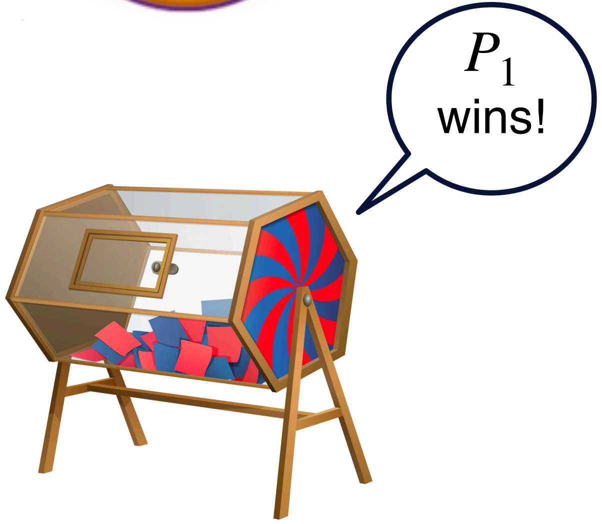
FLOODING WITH A DELAYED ADVERSARY

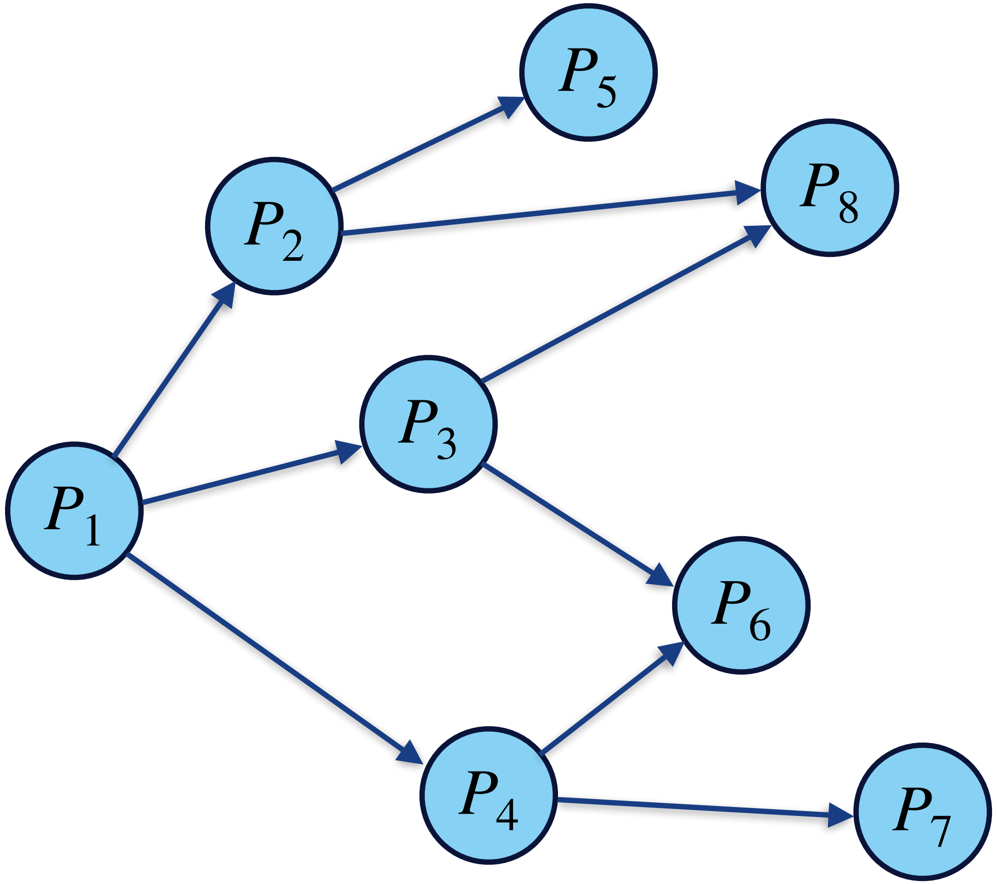


FLOODING WITH A DELAYED ADVERSARY

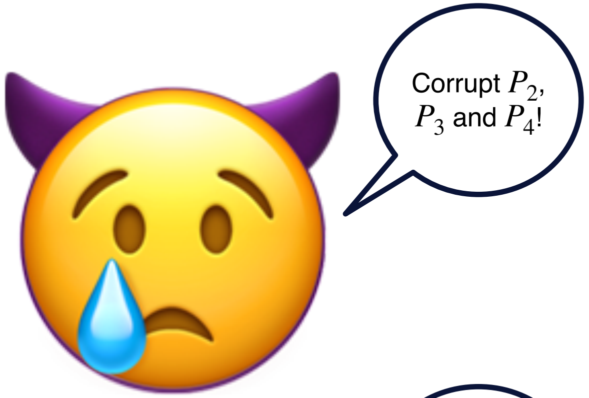


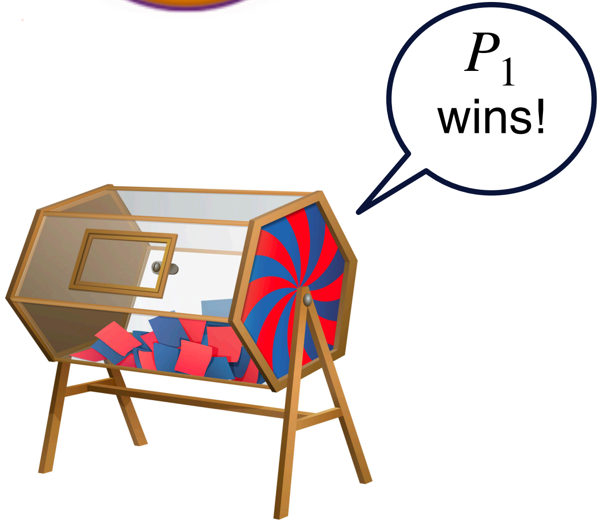
FLOODING WITH A DELAYED ADVERSARY

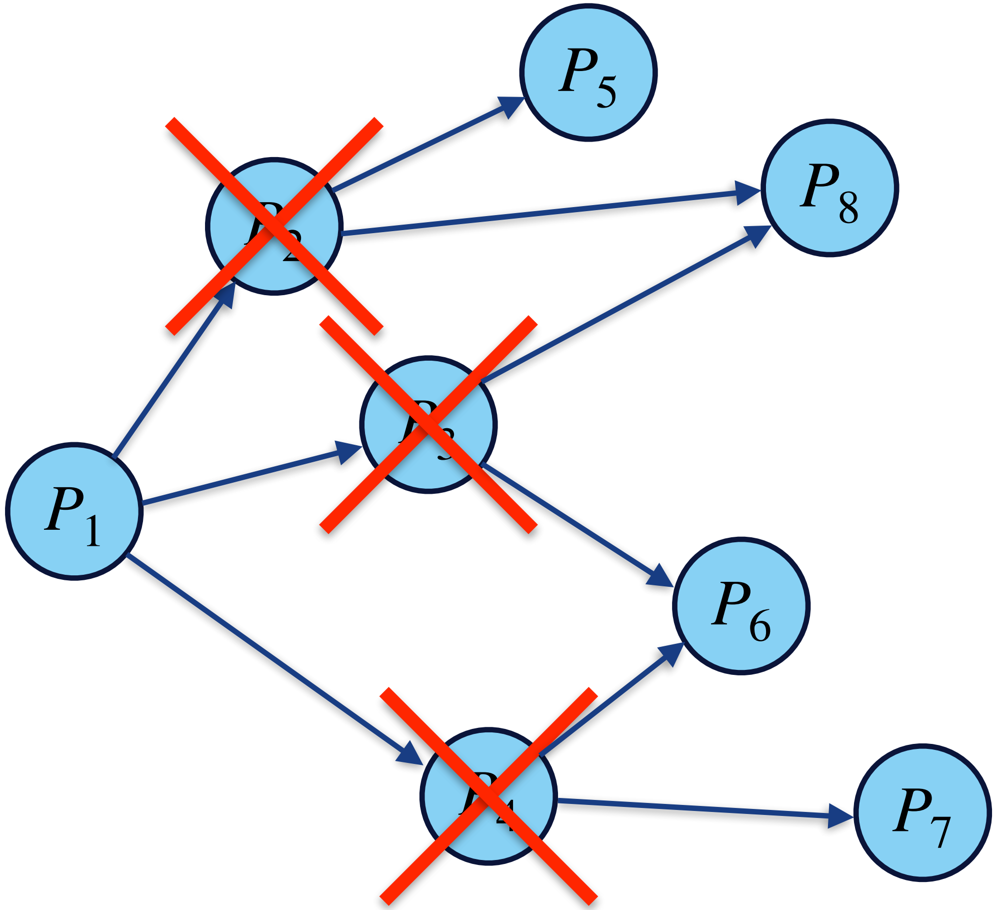




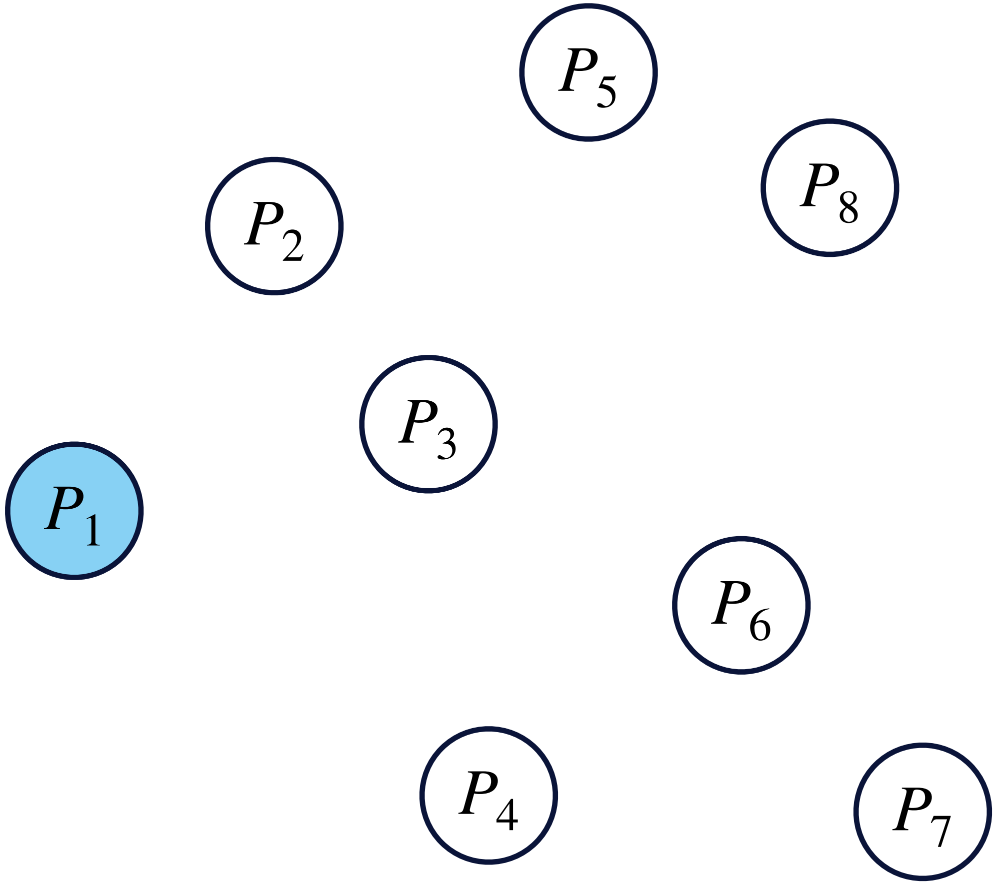
FLOODING WITH A DELAYED ADVERSARY



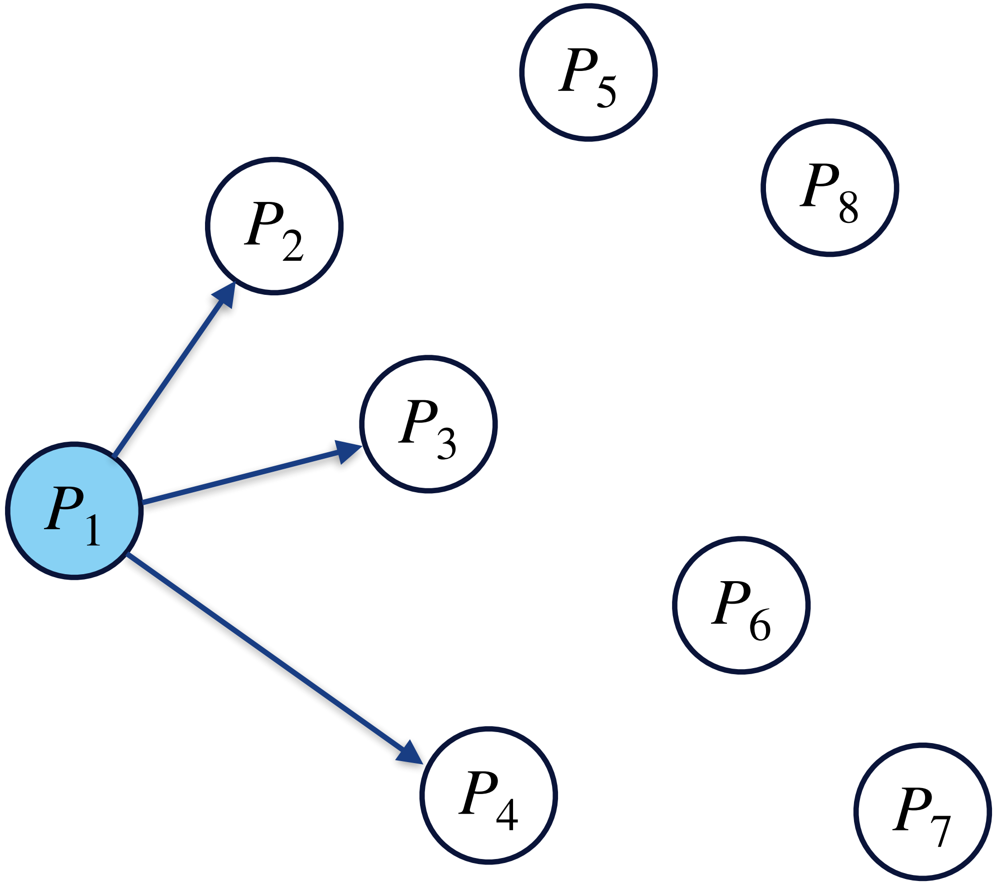




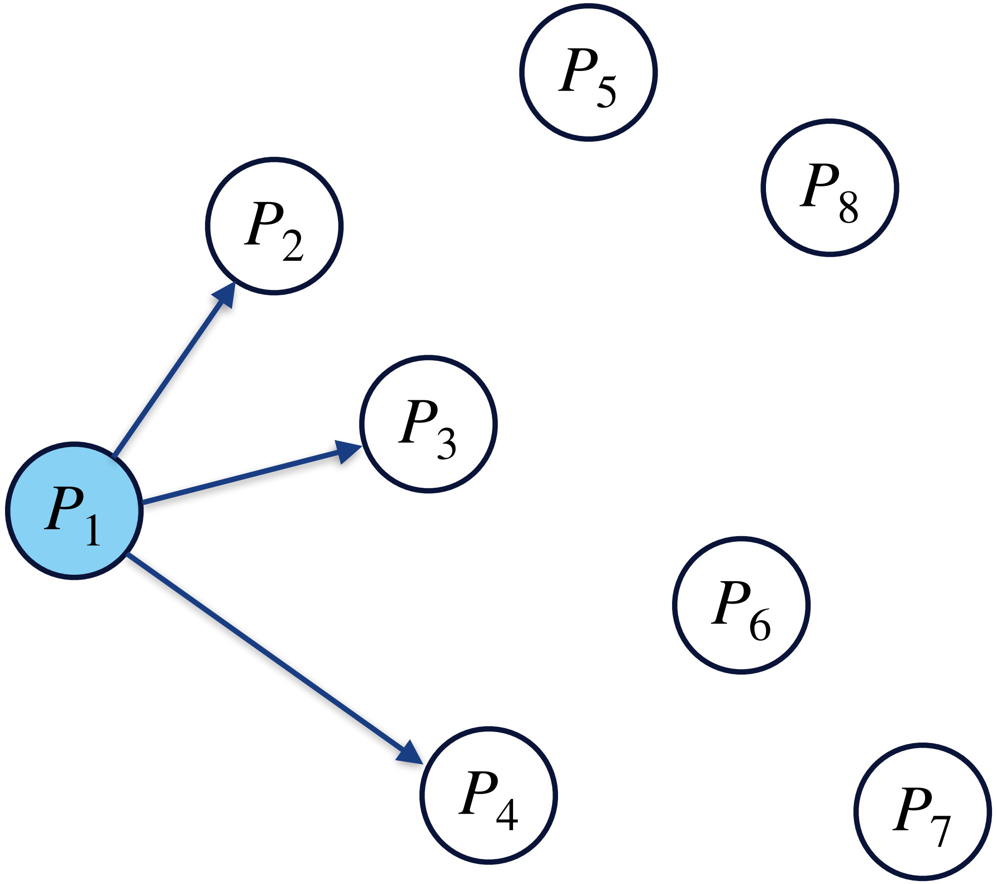
FLOODING WITH A DELAYED ADVERSARY



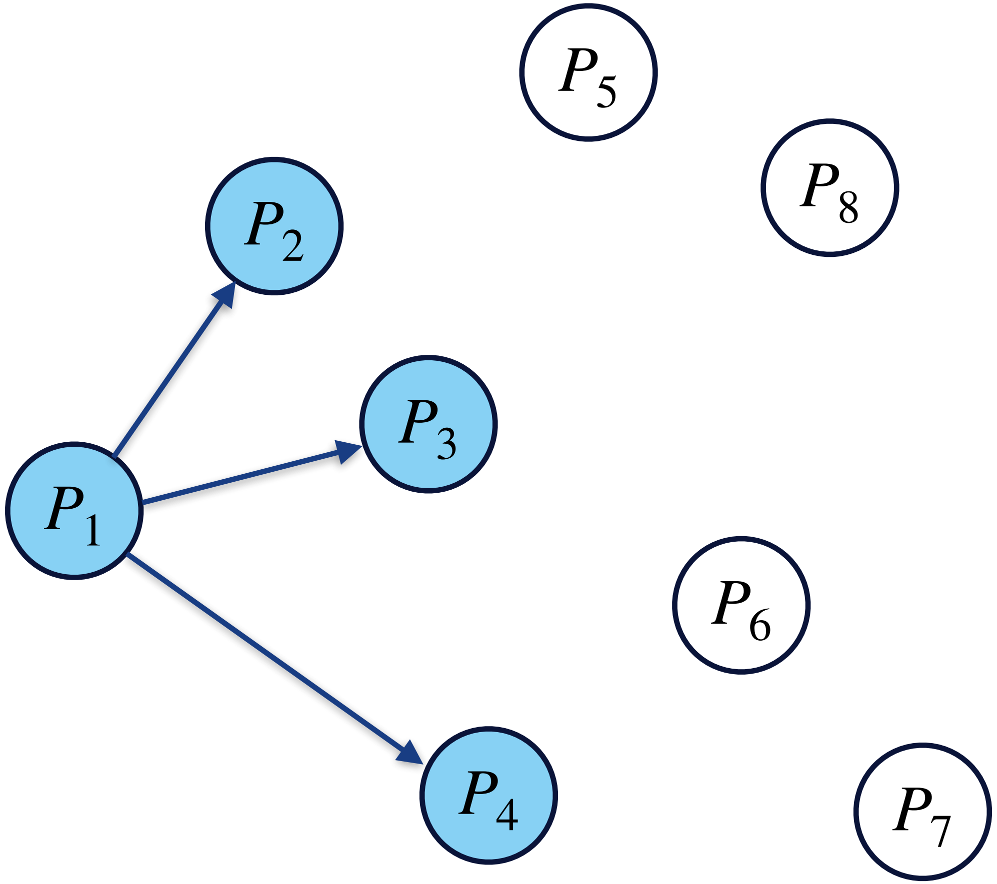
FLOODING WITH A DELAYED ADVERSARY



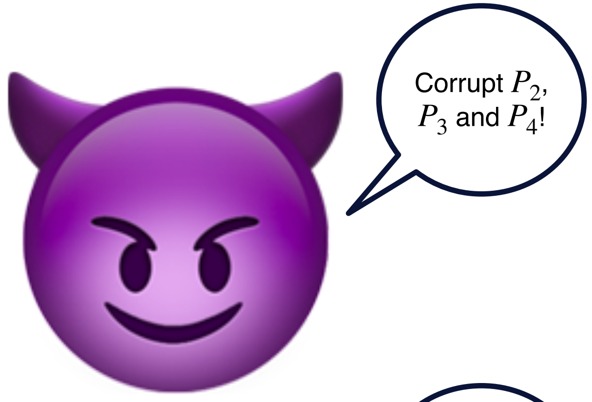
FLOODING WITH A DELAYED ADVERSARY

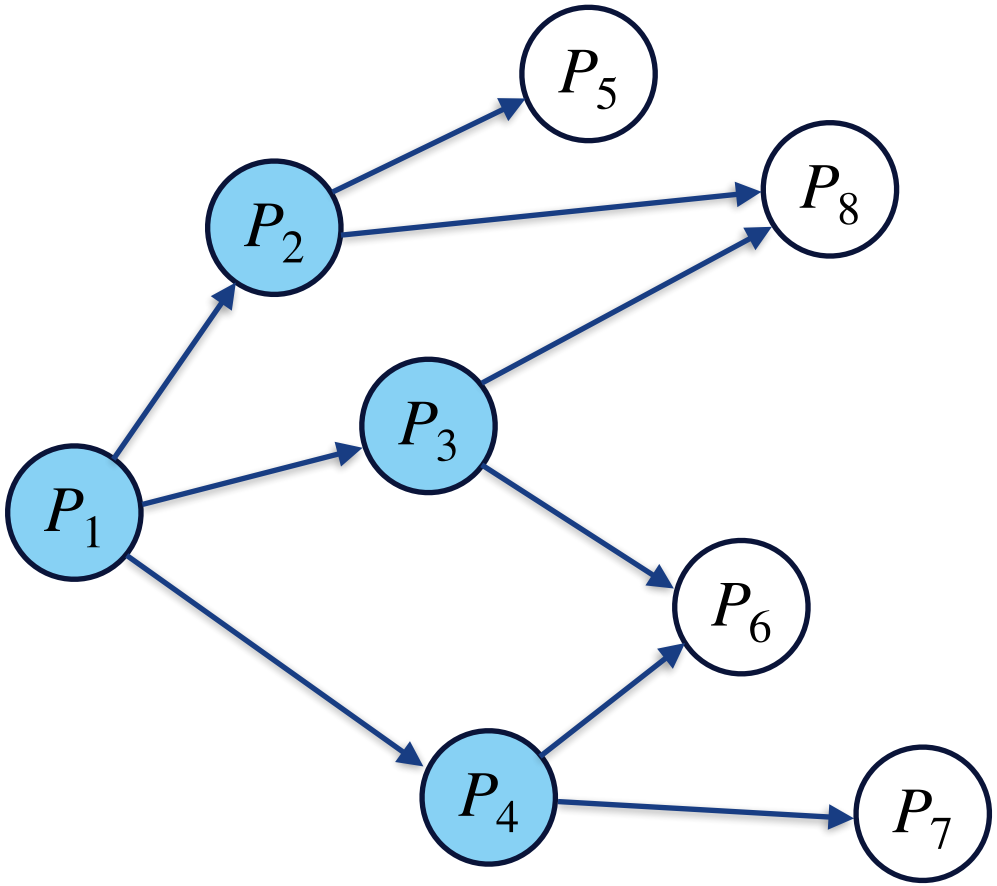


FLOODING WITH A DELAYED ADVERSARY

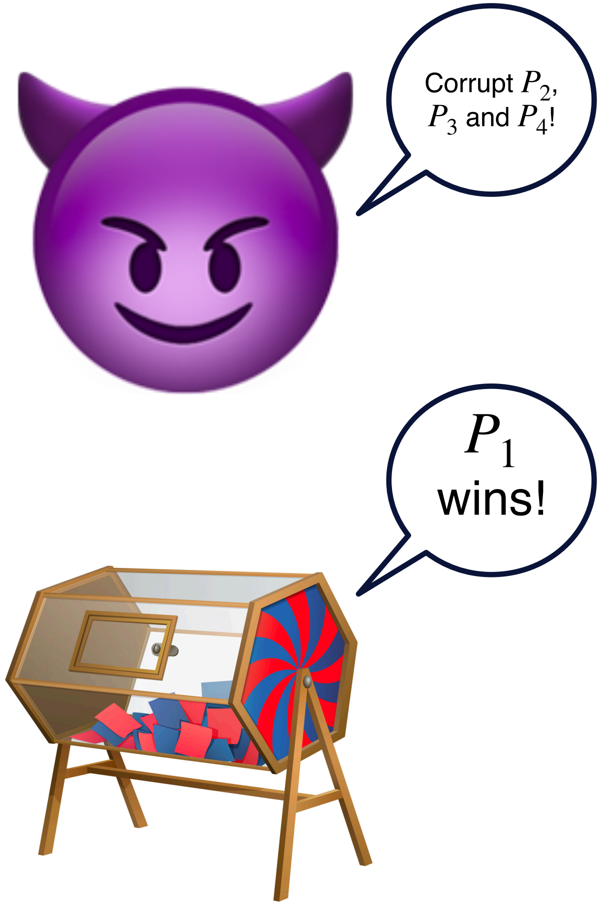


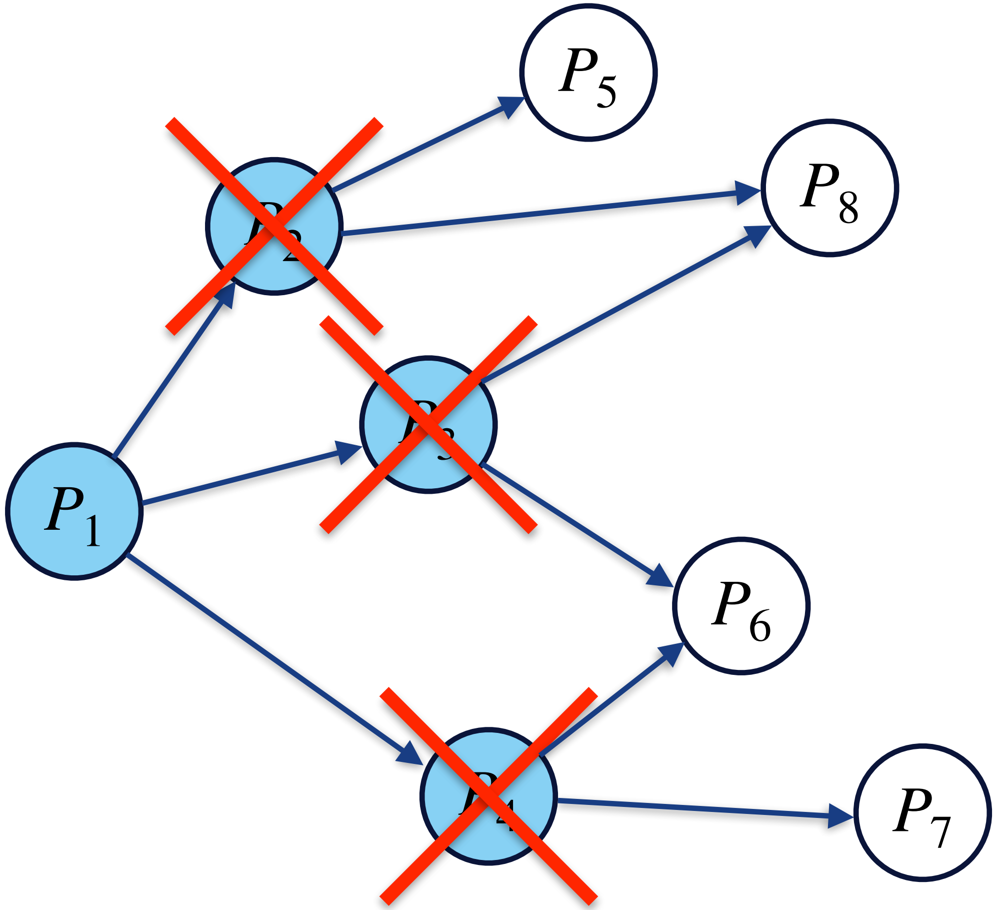
FLOODING WITH A DELAYED ADVERSARY



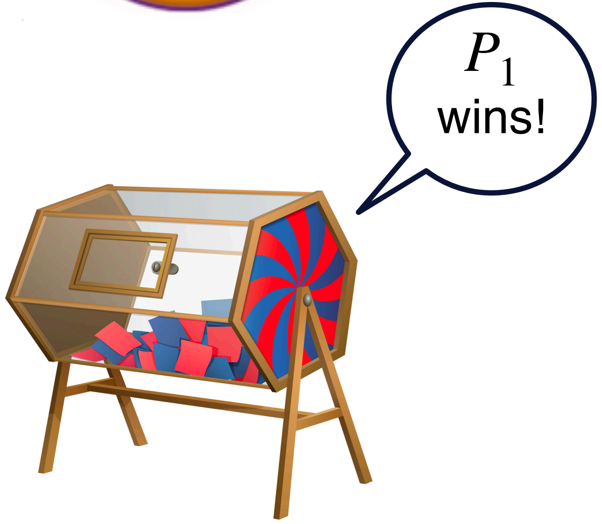


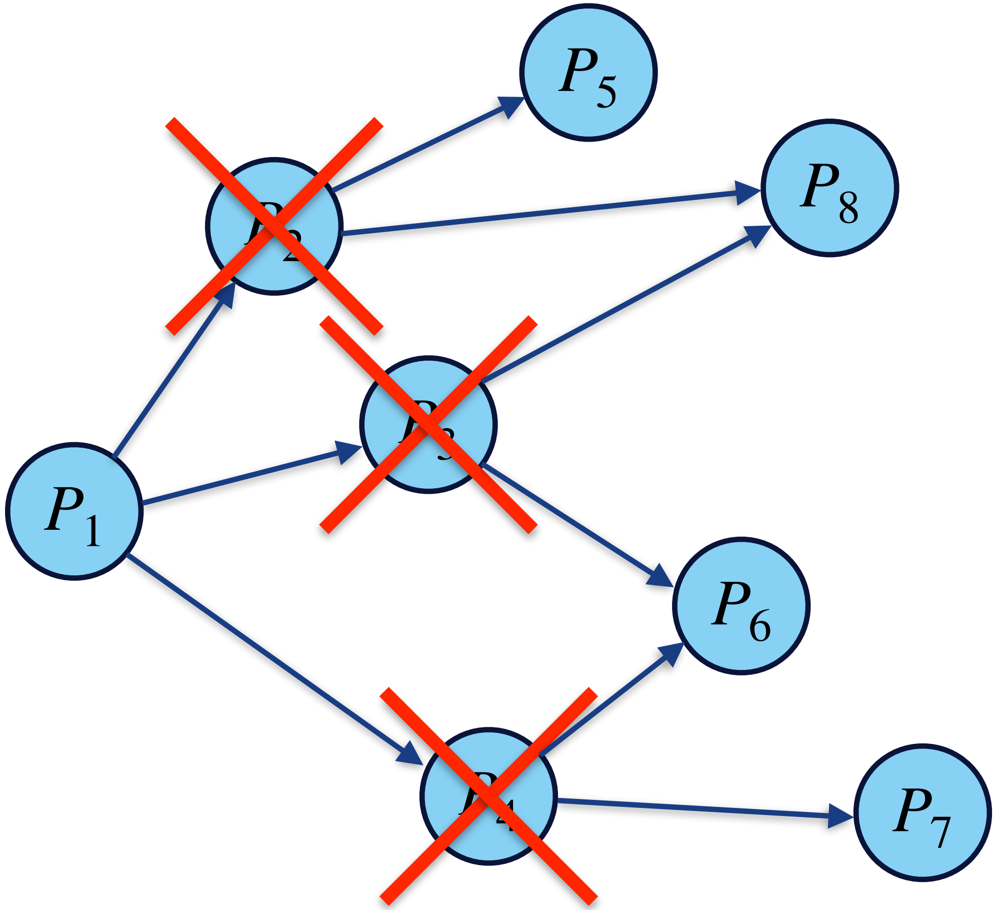
FLOODING WITH A DELAYED ADVERSARY





FLOODING WITH A DELAYED ADVERSARY





OUR WORK

Formalizing Delayed Adaptive Corruptions and the Security of Flooding Networks

Christian Matt[1], Jesper Buus Nielsen*[2], and Søren Eller Thomsen[2]

[1]Concordium, Zurich, Switzerland
cm@concordium.com
[2]Concordium Blockchain Research Center, Aarhus University, Denmark
{jbn, sethomsen}@cs.au.dk

June 27, 2022

Abstract

Many decentralized systems rely on flooding protocols for message dissemination. In such a protocol, the sender of a message sends it to a randomly selected set of peers. These peers again send the message to their randomly selected peers, until every network participant has received the message. This type of protocols clearly fail in face of an adaptive adversary who can simply corrupt all peers of the sender and thereby prevent the message from being delivered. Nevertheless, flooding protocols are commonly used within protocols that aim to be cryptographically secure, most notably in blockchain protocols. While it is possible to revert to static corruptions, this gives unsatisfactory security guarantees, especially in the setting of a blockchain that is supposed to run for an extended period of time.

To be able to provide meaningful security guarantees in such settings, we give precise semantics to what we call δ-*delayed adversaries* in the Universal Composability (UC) framework. Such adversaries can adaptively corrupt parties, but there is a delay of time δ from when an adversary decides to corrupt a party until they succeed in overtaking control of the party. Within this model, we formally prove the intuitive result that flooding protocols are secure against δ-delayed adversaries when δ is at least the time it takes to send a message from one peer to another plus the time it takes the recipient to resend the message. To this end, we show how to reduce the adaptive setting with a δ-delayed adversary to a static experiment with an Erdős–Rényi graph. Using the established theory of Erdős–Rényi graphs, we provide upper bounds on the propagation time of the flooding functionality for different neighborhood sizes of the gossip network. More concretely, we show the following for security parameter κ, point-to-point channels with delay at most Δ, and n parties in total, with a sufficiently delayed adversary that can corrupt any constant fraction of the parties: If all parties send to $\Omega(\kappa)$ parties on average, then we can realize a flooding functionality with maximal delay $\mathcal{O}(\Delta \cdot \log(n))$; and if all parties send to $\Omega(\sqrt{\kappa n})$ parties on average, we can realize a flooding functionality with maximal delay $\mathcal{O}(\Delta)$.

*Partially funded by The Concordium Foundation; The Danish Independent Research Council under Grant-ID DFF-8021-00366B (BETHE); The Carlsberg Foundation under the Semper Ardens Research Project CF18-112 (BCM).

OUR WORK

1. Semantics for δ-delayed adversaries within UC.

Formalizing Delayed Adaptive Corruptions and the Security of Flooding Networks

Christian Matt[1], Jesper Buus Nielsen*[2], and Søren Eller Thomsen[2]

[1]Concordium, Zurich, Switzerland
cm@concordium.com
[2]Concordium Blockchain Research Center, Aarhus University, Denmark
{jbn, sethomsen}@cs.au.dk

June 27, 2022

Abstract

Many decentralized systems rely on flooding protocols for message dissemination. In such a protocol, the sender of a message sends it to a randomly selected set of peers. These peers again send the message to their randomly selected peers, until every network participant has received the message. This type of protocols clearly fail in face of an adaptive adversary who can simply corrupt all peers of the sender and thereby prevent the message from being delivered. Nevertheless, flooding protocols are commonly used within protocols that aim to be cryptographically secure, most notably in blockchain protocols. While it is possible to revert to static corruptions, this gives unsatisfactory security guarantees, especially in the setting of a blockchain that is supposed to run for an extended period of time.

To be able to provide meaningful security guarantees in such settings, we give precise semantics to what we call δ-*delayed adversaries* in the Universal Composability (UC) framework. Such adversaries can adaptively corrupt parties, but there is a delay of time δ from when an adversary decides to corrupt a party until they succeed in overtaking control of the party. Within this model, we formally prove the intuitive result that flooding protocols are secure against δ-delayed adversaries when δ is at least the time it takes to send a message from one peer to another plus the time it takes the recipient to resend the message. To this end, we show how to reduce the adaptive setting with a δ-delayed adversary to a static experiment with an Erdős–Rényi graph. Using the established theory of Erdős–Rényi graphs, we provide upper bounds on the propagation time of the flooding functionality for different neighborhood sizes of the gossip network. More concretely, we show the following for security parameter κ, point-to-point channels with delay at most Δ, and n parties in total, with a sufficiently delayed adversary that can corrupt any constant fraction of the parties: If all parties send to $\Omega(\kappa)$ parties on average, then we can realize a flooding functionality with maximal delay $\mathcal{O}(\Delta \cdot \log(n))$; and if all parties send to $\Omega(\sqrt{\kappa n})$ parties on average, we can realize a flooding functionality with maximal delay $\mathcal{O}(\Delta)$.

*Partially funded by The Concordium Foundation; The Danish Independent Research Council under Grant-ID DFF-8021-00366B (BETHE); The Carlsberg Foundation under the Semper Ardens Research Project CF18-112 (BCM).

OUR WORK

1. Semantics for δ-delayed adversaries within UC.

2. Two instantiations of flooding networks secure against an adaptive adversary that is delayed for "*the time it takes to send + the time it takes time to resend*":

Formalizing Delayed Adaptive Corruptions and the Security of Flooding Networks

Christian Matt[1], Jesper Buus Nielsen*[2], and Søren Eller Thomsen[2]

[1]Concordium, Zurich, Switzerland
cm@concordium.com
[2]Concordium Blockchain Research Center, Aarhus University, Denmark
{jbn, sethomsen}@cs.au.dk

June 27, 2022

Abstract

Many decentralized systems rely on flooding protocols for message dissemination. In such a protocol, the sender of a message sends it to a randomly selected set of peers. These peers again send the message to their randomly selected peers, until every network participant has received the message. This type of protocols clearly fail in face of an adaptive adversary who can simply corrupt all peers of the sender and thereby prevent the message from being delivered. Nevertheless, flooding protocols are commonly used within protocols that aim to be cryptographically secure, most notably in blockchain protocols. While it is possible to revert to static corruptions, this gives unsatisfactory security guarantees, especially in the setting of a blockchain that is supposed to run for an extended period of time.

To be able to provide meaningful security guarantees in such settings, we give precise semantics to what we call *δ-delayed adversaries* in the Universal Composability (UC) framework. Such adversaries can adaptively corrupt parties, but there is a delay of time δ from when an adversary decides to corrupt a party until they succeed in overtaking control of the party. Within this model, we formally prove the intuitive result that flooding protocols are secure against δ-delayed adversaries when δ is at least the time it takes to send a message from one peer to another plus the time it takes the recipient to resend the message. To this end, we show how to reduce the adaptive setting with a δ-delayed adversary to a static experiment with an Erdős–Rényi graph. Using the established theory of Erdős–Rényi graphs, we provide upper bounds on the propagation time of the flooding functionality for different neighborhood sizes of the gossip network. More concretely, we show the following for security parameter κ, point-to-point channels with delay at most Δ, and n parties in total, with a sufficiently delayed adversary that can corrupt any constant fraction of the parties: If all parties send to $\Omega(\kappa)$ parties on average, then we can realize a flooding functionality with maximal delay $\mathcal{O}(\Delta \cdot \log(n))$; and if all parties send to $\Omega(\sqrt{\kappa n})$ parties on average, we can realize a flooding functionality with maximal delay $\mathcal{O}(\Delta)$.

*Partially funded by The Concordium Foundation; The Danish Independent Research Council under Grant-ID DFF-8021-00366B (BETHE); The Carlsberg Foundation under the Semper Ardens Research Project CF18-112 (BCM).

OUR WORK

1. Semantics for δ-delayed adversaries within UC.

2. Two instantiations of flooding networks secure against an adaptive adversary that is delayed for "*the time it takes to send + the time it takes time to resend*":

   - $\Omega(\kappa)$ neighbors with $\mathcal{O}\big(\log(n)\big)$ diameter.

Formalizing Delayed Adaptive Corruptions and the Security of Flooding Networks

Christian Matt[1], Jesper Buus Nielsen*[2], and Søren Eller Thomsen[2]

[1]Concordium, Zurich, Switzerland
cm@concordium.com
[2]Concordium Blockchain Research Center, Aarhus University, Denmark
{jbn, sethomsen}@cs.au.dk

June 27, 2022

Abstract

Many decentralized systems rely on flooding protocols for message dissemination. In such a protocol, the sender of a message sends it to a randomly selected set of peers. These peers again send the message to their randomly selected peers, until every network participant has received the message. This type of protocols clearly fail in face of an adaptive adversary who can simply corrupt all peers of the sender and thereby prevent the message from being delivered. Nevertheless, flooding protocols are commonly used within protocols that aim to be cryptographically secure, most notably in blockchain protocols. While it is possible to revert to static corruptions, this gives unsatisfactory security guarantees, especially in the setting of a blockchain that is supposed to run for an extended period of time.

To be able to provide meaningful security guarantees in such settings, we give precise semantics to what we call δ*-delayed adversaries* in the Universal Composability (UC) framework. Such adversaries can adaptively corrupt parties, but there is a delay of time δ from when an adversary decides to corrupt a party until they succeed in overtaking control of the party. Within this model, we formally prove the intuitive result that flooding protocols are secure against δ-delayed adversaries when δ is at least the time it takes to send a message from one peer to another plus the time it takes the recipient to resend the message. To this end, we show how to reduce the adaptive setting with a δ-delayed adversary to a static experiment with an Erdős–Rényi graph. Using the established theory of Erdős–Rényi graphs, we provide upper bounds on the propagation time of the flooding functionality for different neighborhood sizes of the gossip network. More concretely, we show the following for security parameter κ, point-to-point channels with delay at most Δ, and n parties in total, with a sufficiently delayed adversary that can corrupt any constant fraction of the parties: If all parties send to $\Omega(\kappa)$ parties on average, then we can realize a flooding functionality with maximal delay $\mathcal{O}(\Delta \cdot \log(n))$; and if all parties send to $\Omega(\sqrt{\kappa n})$ parties on average, we can realize a flooding functionality with maximal delay $\mathcal{O}(\Delta)$.

*Partially funded by The Concordium Foundation; The Danish Independent Research Council under Grant-ID DFF-8021-00366B (BETHE); The Carlsberg Foundation under the Semper Ardens Research Project CF18-112 (BCM).

OUR WORK

1. Semantics for δ-delayed adversaries within UC.

2. Two instantiations of flooding networks secure against an adaptive adversary that is delayed for "*the time it takes to send + the time it takes time to resend*":
   - $\Omega(\kappa)$ neighbors with $\mathcal{O}\big(\log(n)\big)$ diameter.
   - $\Omega\left(\sqrt{\kappa \cdot n}\right)$ neighbors with $\mathcal{O}(1)$ diameter.

Formalizing Delayed Adaptive Corruptions and the Security of Flooding Networks

Christian Matt[1], Jesper Buus Nielsen*[2], and Søren Eller Thomsen[2]

[1]Concordium, Zurich, Switzerland
cm@concordium.com
[2]Concordium Blockchain Research Center, Aarhus University, Denmark
{jbn, sethomsen}@cs.au.dk

June 27, 2022

Abstract

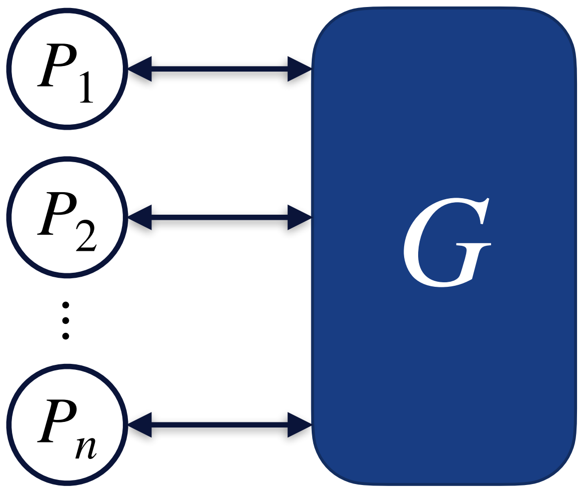
Many decentralized systems rely on flooding protocols for message dissemination. In such a protocol, the sender of a message sends it to a randomly selected set of peers. These peers again send the message to their randomly selected peers, until every network participant has received the message. This type of protocols clearly fail in face of an adaptive adversary who can simply corrupt all peers of the sender and thereby prevent the message from being delivered. Nevertheless, flooding protocols are commonly used within protocols that aim to be cryptographically secure, most notably in blockchain protocols. While it is possible to revert to static corruptions, this gives unsatisfactory security guarantees, especially in the setting of a blockchain that is supposed to run for an extended period of time.

To be able to provide meaningful security guarantees in such settings, we give precise semantics to what we call *δ-delayed adversaries* in the Universal Composability (UC) framework. Such adversaries can adaptively corrupt parties, but there is a delay of time δ from when an adversary decides to corrupt a party until they succeed in overtaking control of the party. Within this model, we formally prove the intuitive result that flooding protocols are secure against δ-delayed adversaries when δ is at least the time it takes to send a message from one peer to another plus the time it takes the recipient to resend the message. To this end, we show how to reduce the adaptive setting with a δ-delayed adversary to a static experiment with an Erdős–Rényi graph. Using the established theory of Erdős–Rényi graphs, we provide upper bounds on the propagation time of the flooding functionality for different neighborhood sizes of the gossip network. More concretely, we show the following for security parameter κ, point-to-point channels with delay at most Δ, and n parties in total, with a sufficiently delayed adversary that can corrupt any constant fraction of the parties: If all parties send to $\Omega(\kappa)$ parties on average, then we can realize a flooding functionality with maximal delay $\mathcal{O}(\Delta \cdot \log(n))$; and if all parties send to $\Omega(\sqrt{\kappa n})$ parties on average, we can realize a flooding functionality with maximal delay $\mathcal{O}(\Delta)$.

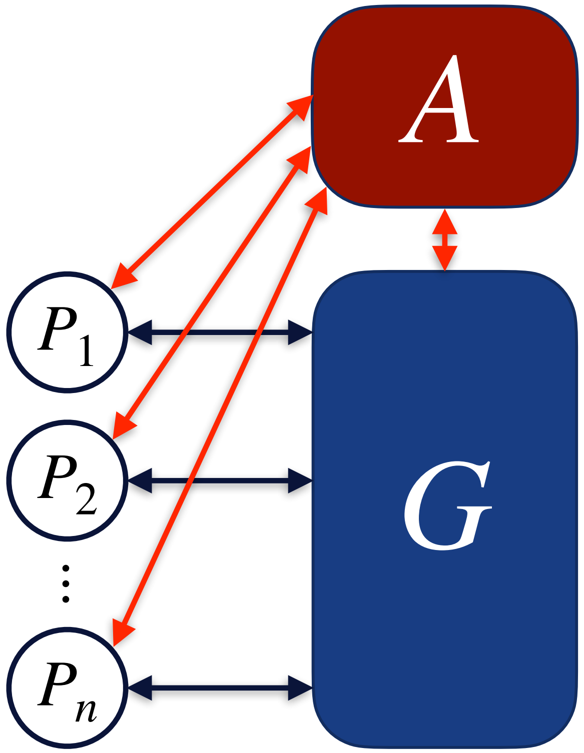
*Partially funded by The Concordium Foundation; The Danish Independent Research Council under Grant-ID DFF-8021-00366B (BETHE); The Carlsberg Foundation under the Semper Ardens Research Project CF18-112 (BCM).

DELAYED ADVERSARIES IN UC

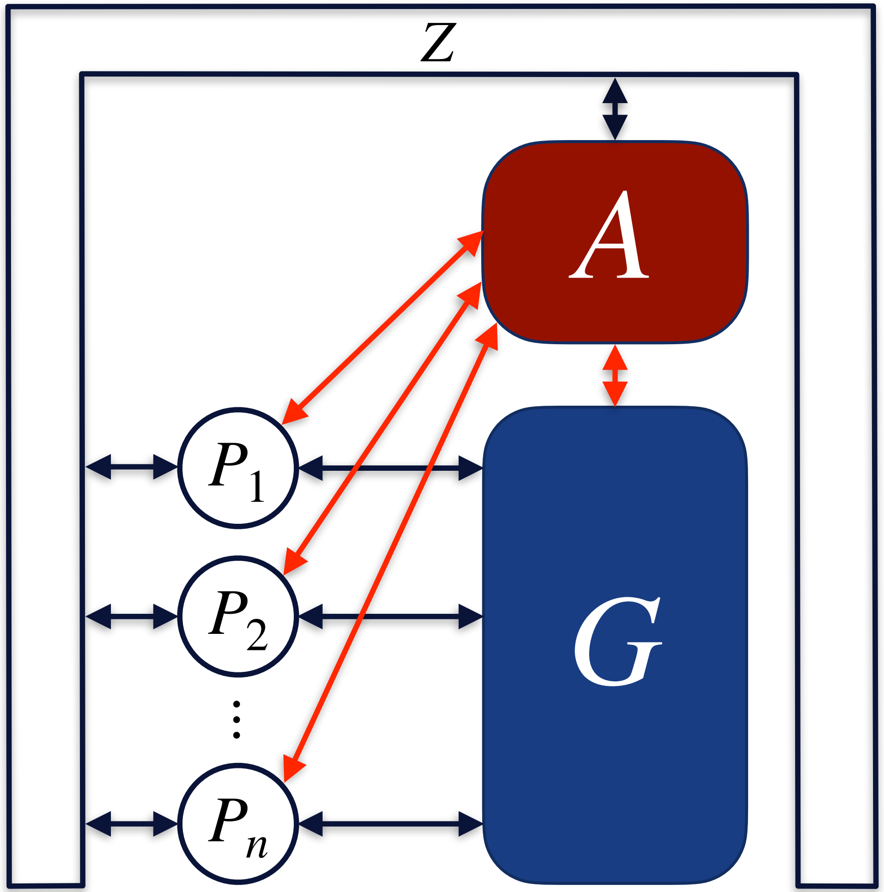
DELAYED ADVERSARIES IN UC



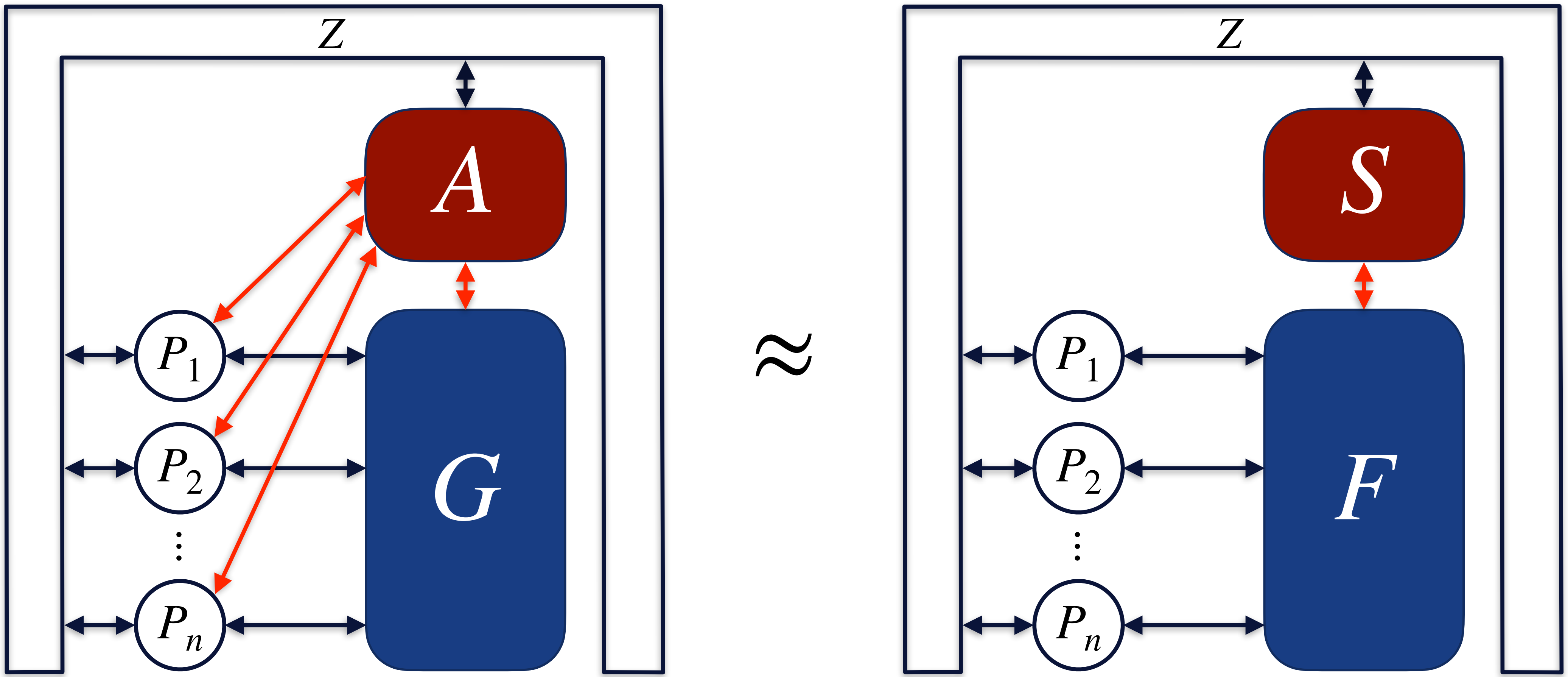
DELAYED ADVERSARIES IN UC



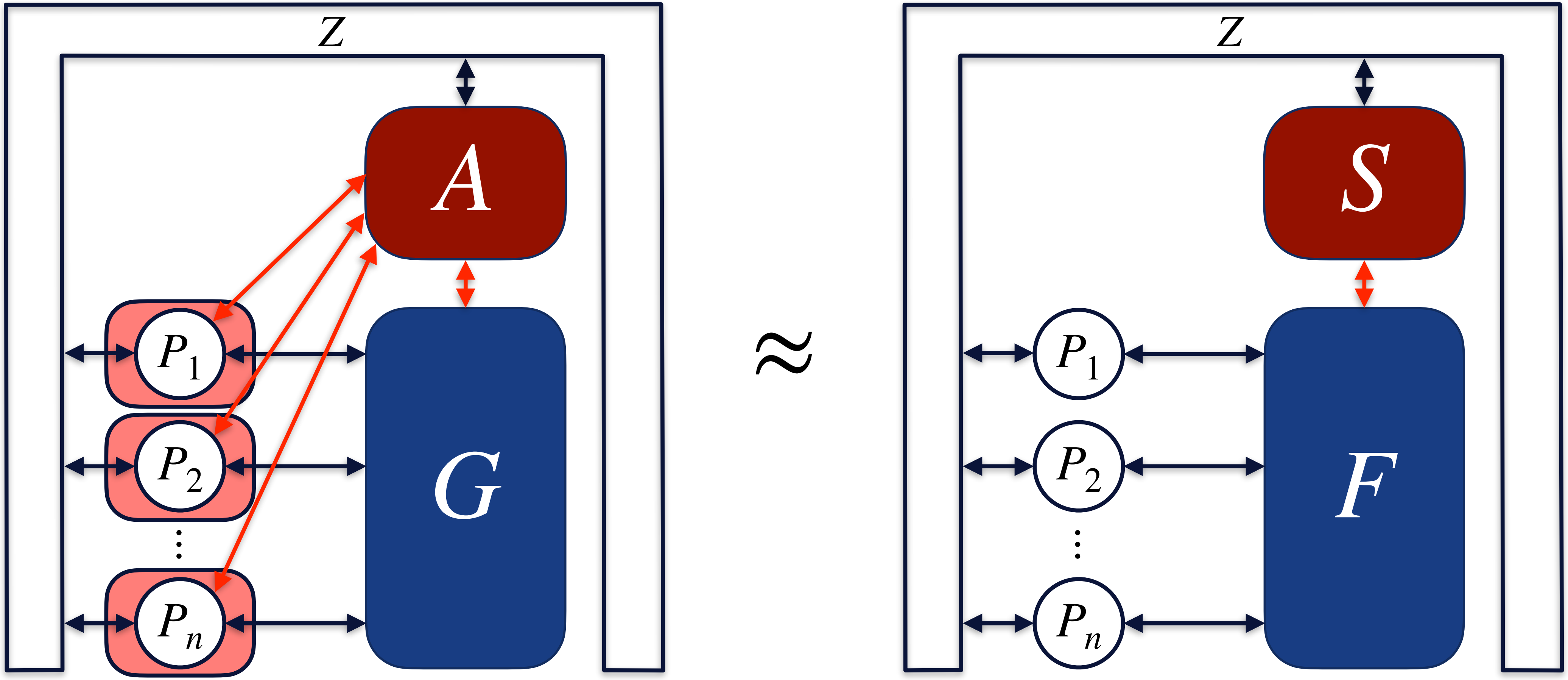
DELAYED ADVERSARIES IN UC



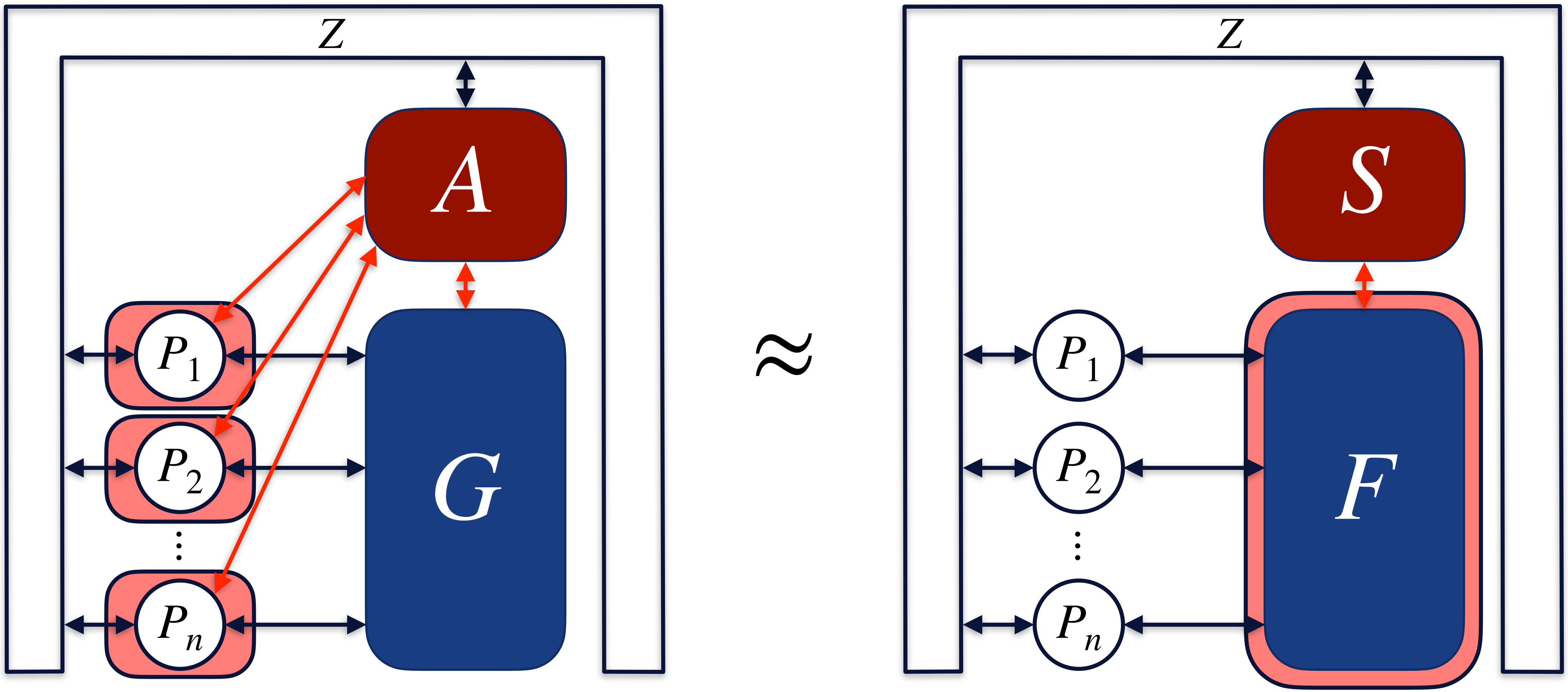
DELAYED ADVERSARIES IN UC



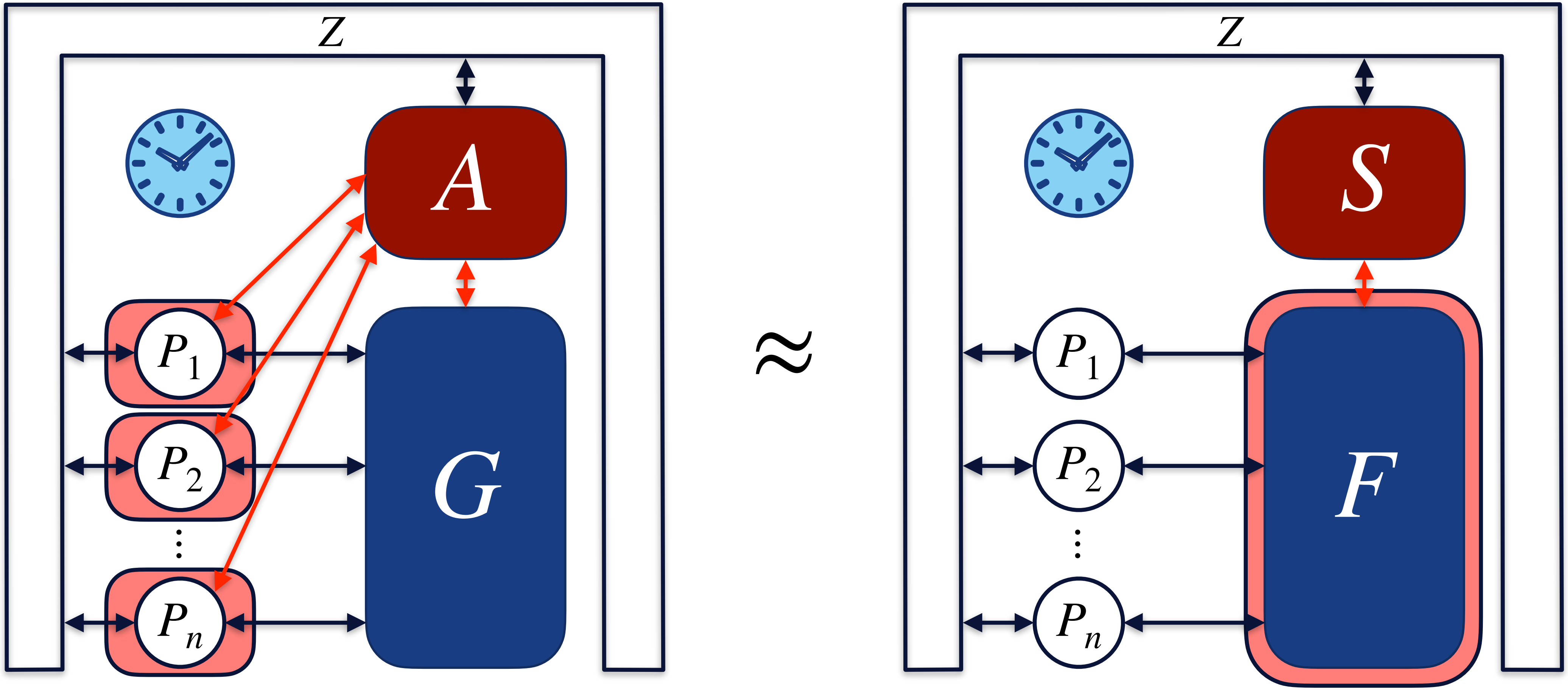
DELAYED ADVERSARIES IN UC



DELAYED ADVERSARIES IN UC

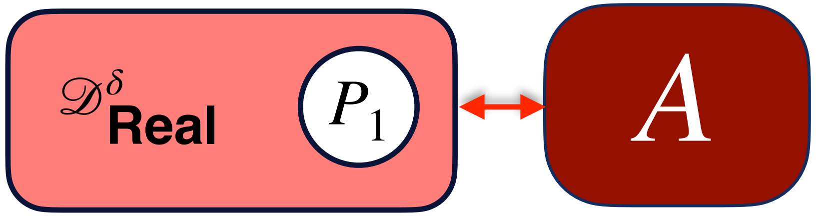


DELAYED ADVERSARIES IN UC

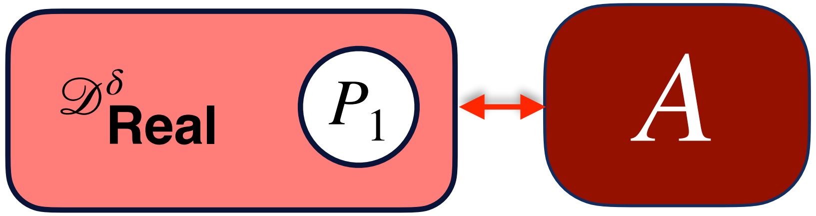


*We use the notion of time from TARDIS [BDDNO21].

DELAY SHELLS

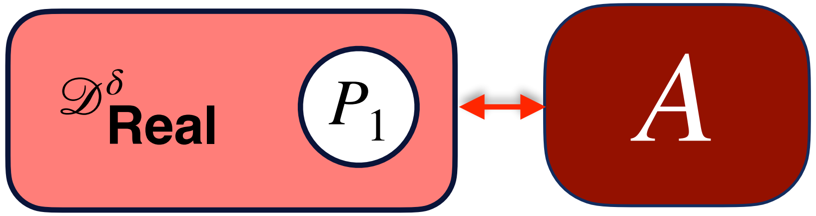


DELAY SHELLS



For each party, three additional commands:

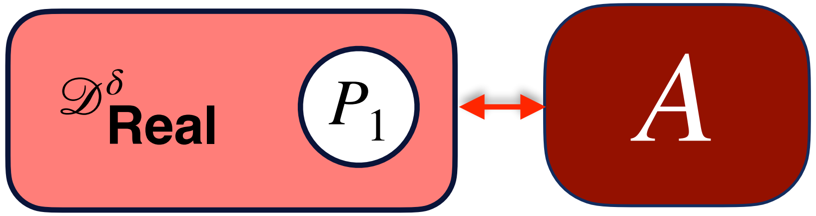
DELAY SHELLS



For each party, three additional commands:

- $\textsc{Precorrupt}$: Time for precorruption is noted.

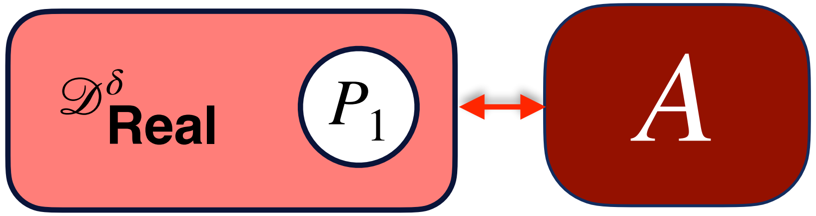
DELAY SHELLS



For each party, three additional commands:

- PRECORRUPT: Time for precorruption is noted.
- CORRUPT: Checks if current time is at least δ time after precorruption and otherwise ignores input.

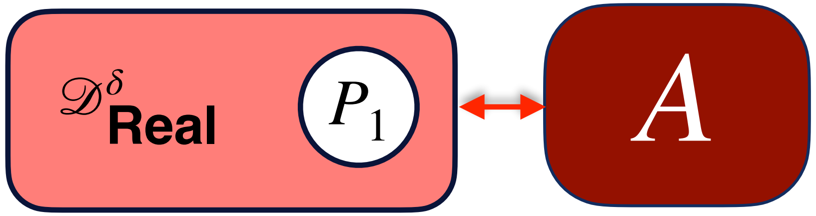
DELAY SHELLS

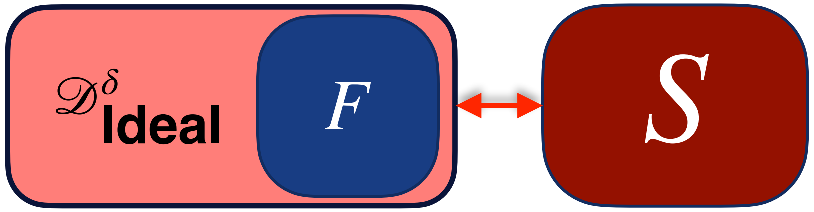


For each party, three additional commands:

- $\textsc{Precorrupt}$: Time for precorruption is noted.
- $\textsc{Corrupt}$: Checks if current time is at least δ time after precorruption and otherwise ignores input.
- $(\textsc{Initialize}, \tau)$: If given as the first command, the precorruption time is updated to be τ.

DELAY SHELLS





For each party, three additional commands:

- PRECORRUPT: Time for precorruption is noted.
- CORRUPT: Checks if current time is at least δ time after precorruption and otherwise ignores input.
- (INITIALIZE, τ): If given as the first command, the precorruption time is updated to be τ.

THEOREMS

Theorem 1. *Security against a byzantine adversary implies security against a 0-delayed adversary.*

THEOREMS

Theorem 1. *Security against a byzantine adversary implies security against a 0-delayed adversary.*

Theorem 2. *Security against a fast adversary implies security against a slow adversary.*

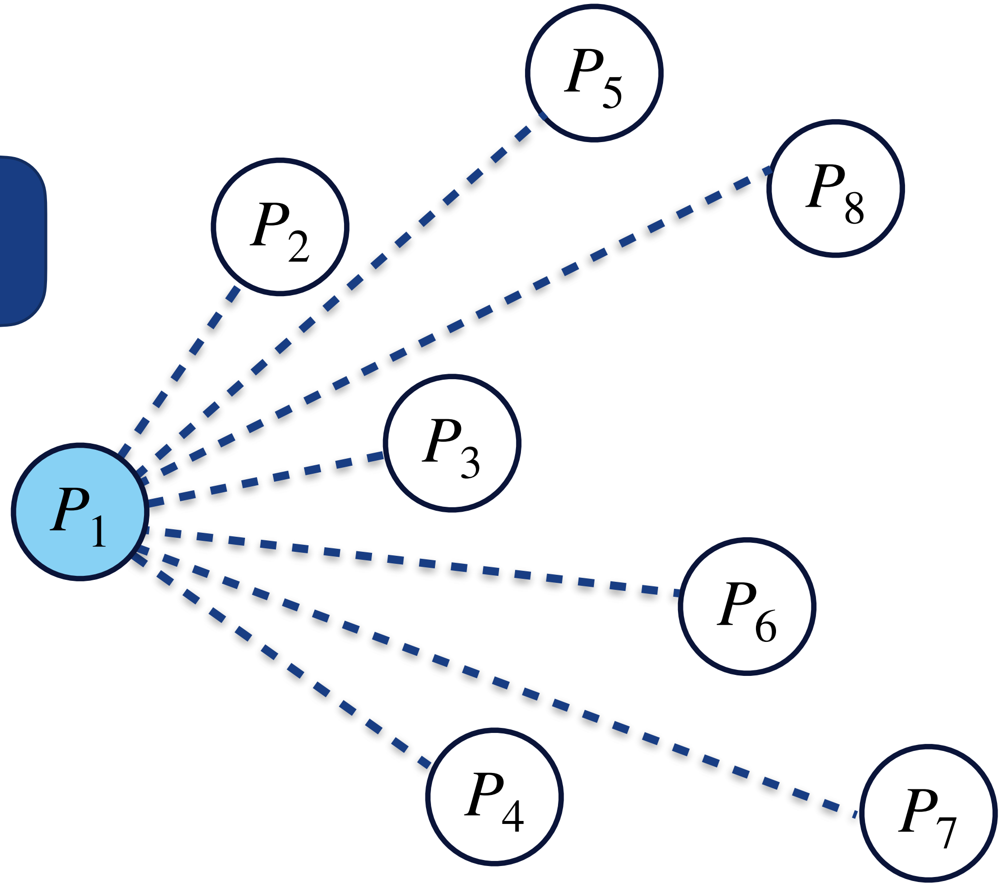
FLOODING PROTOCOL: $\pi_{\text{ERFlood}}(\rho)$

FLOODING PROTOCOL: $\pi_{\textbf{ERFlood}}(\rho)$

Forward each message to each party with probability ρ.

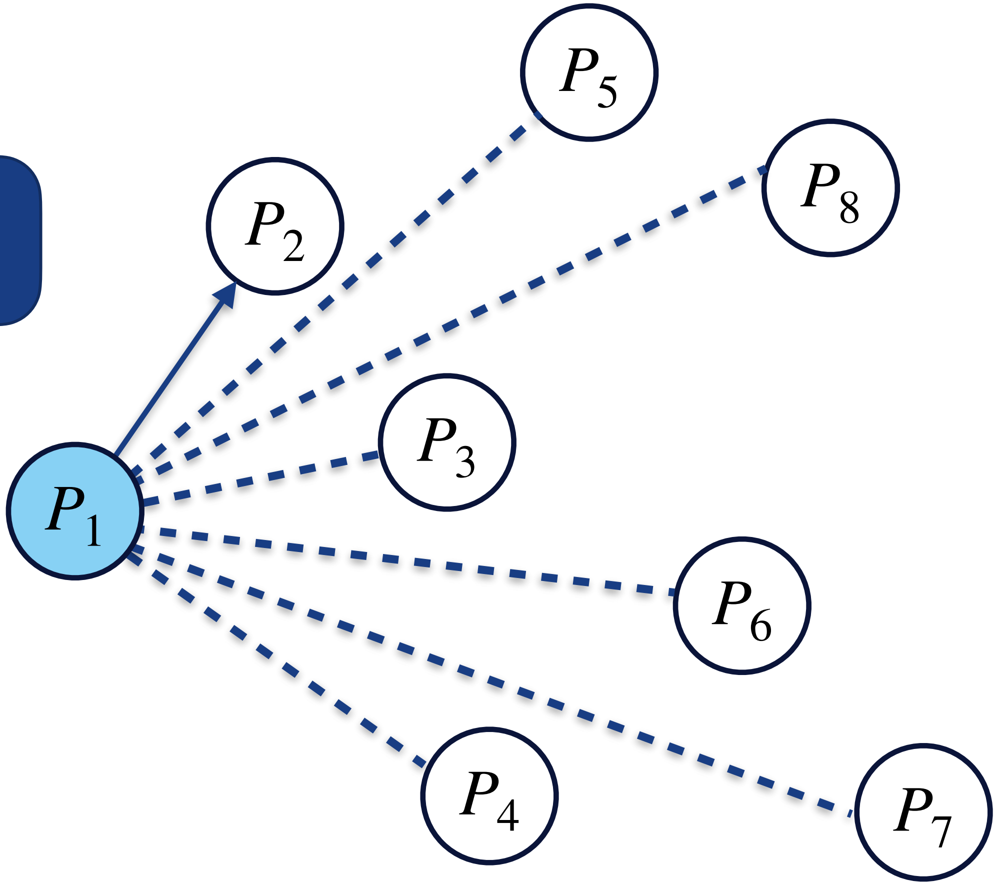
FLOODING PROTOCOL: $\pi_{\text{ERFlood}}(\rho)$

Forward each message to each party with probability ρ.



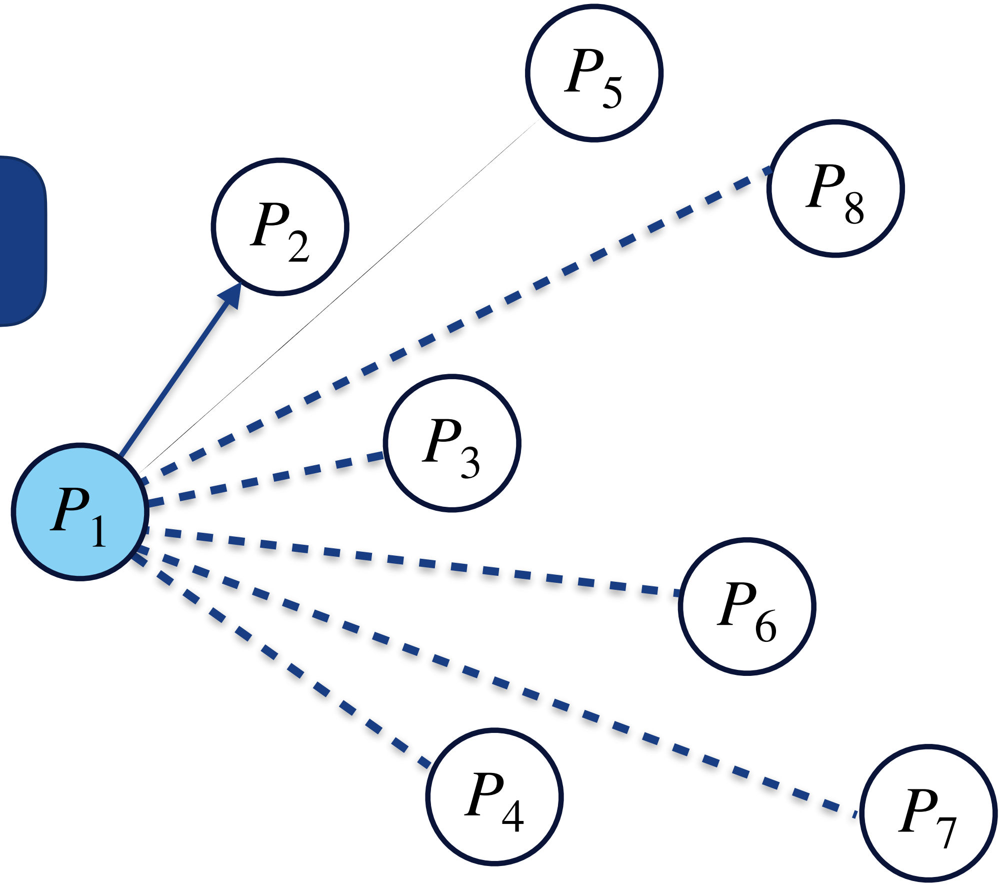
FLOODING PROTOCOL: $\pi_{\mathbf{ERFlood}}(\rho)$

Forward each message to each party with probability ρ.



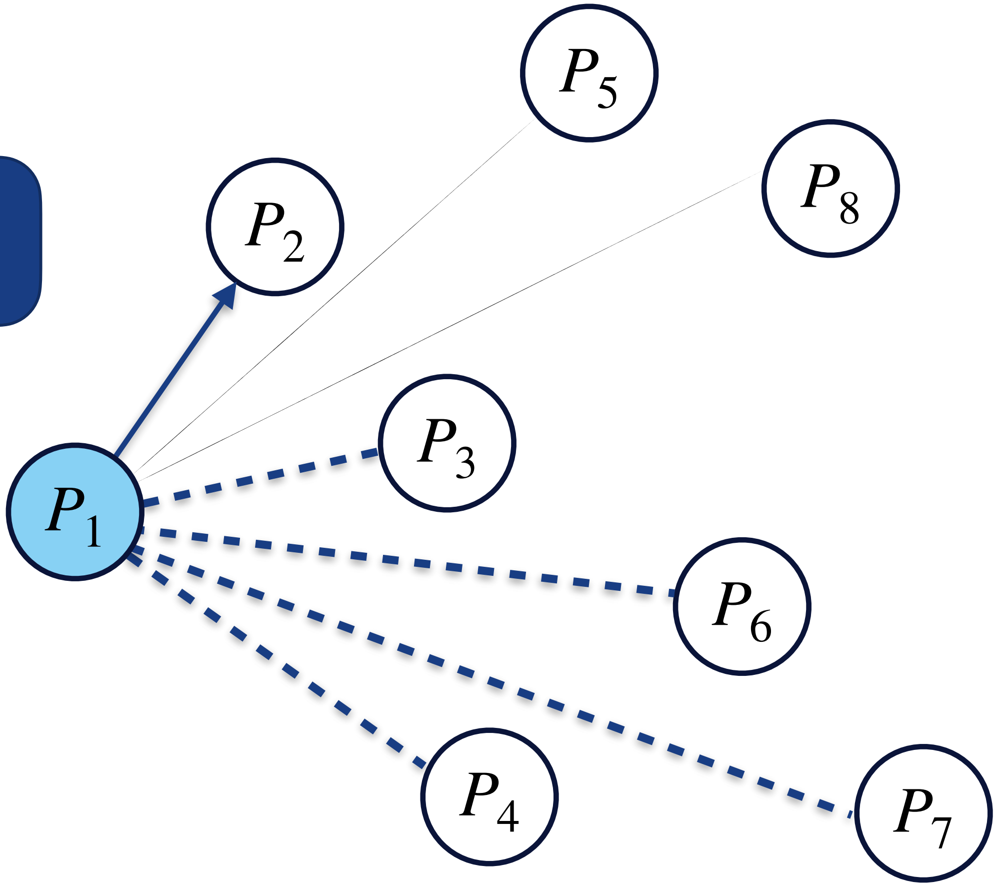
FLOODING PROTOCOL: $\pi_{\text{ERFlood}}(\rho)$

Forward each message to each party with probability ρ.



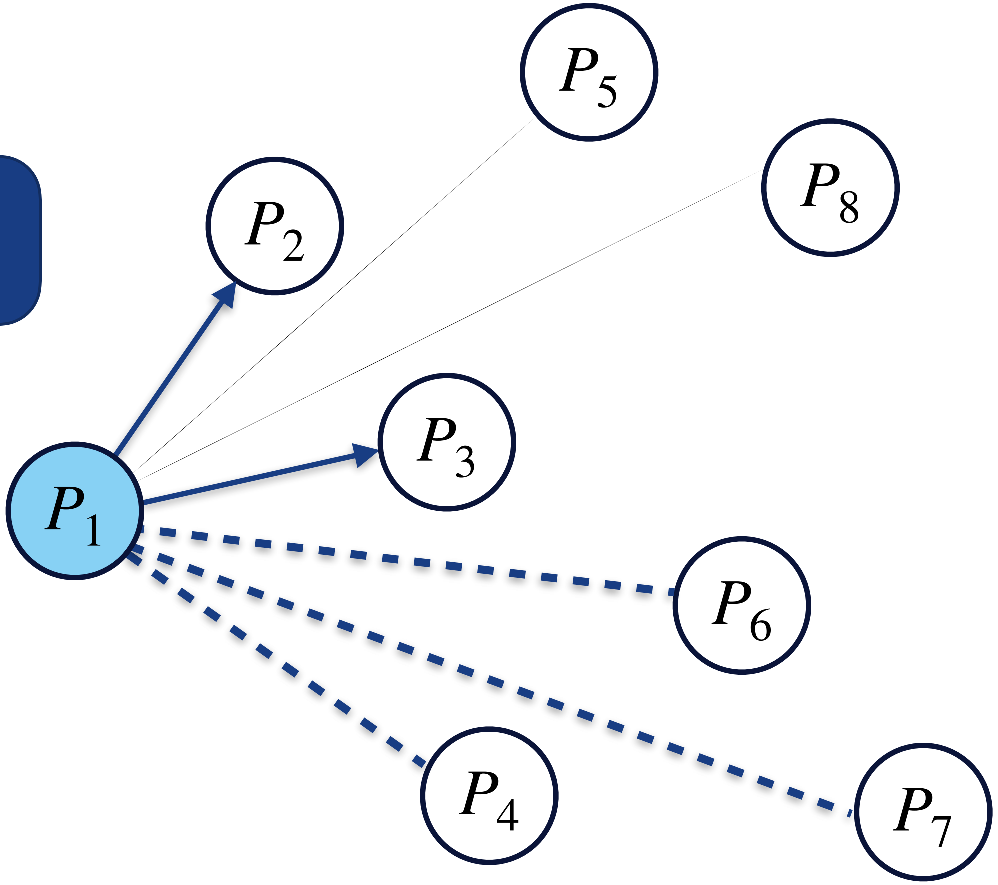
FLOODING PROTOCOL: $\pi_{\mathbf{ERFlood}}(\rho)$

Forward each message to each party with probability ρ.



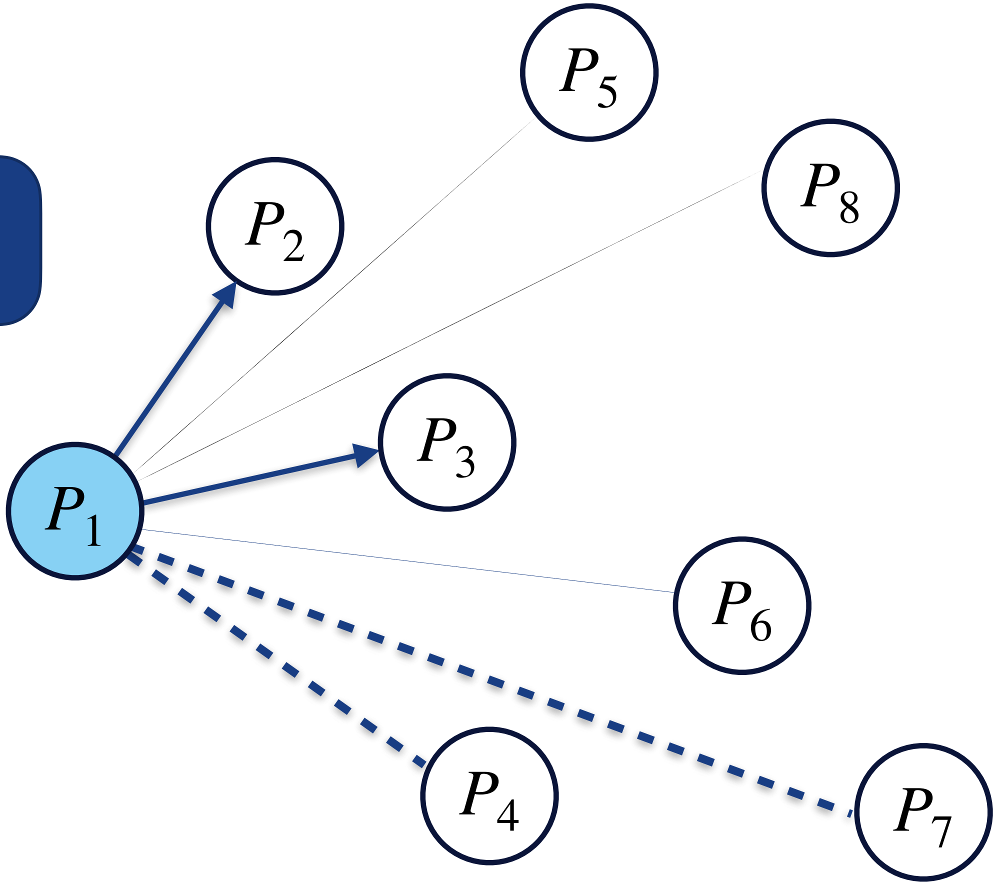
FLOODING PROTOCOL: $\pi_{\textbf{ERFlood}}(\rho)$

Forward each message to each party with probability ρ.



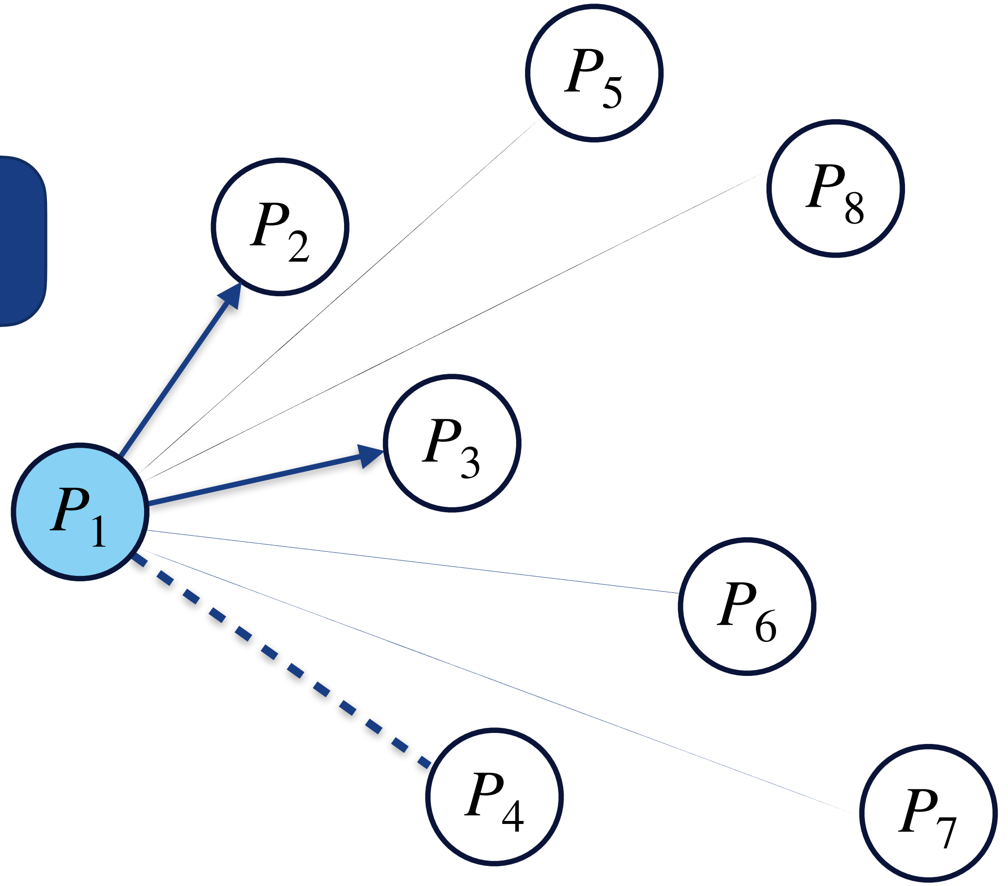
FLOODING PROTOCOL: $\pi_{\textbf{ERFlood}}(\rho)$

Forward each message to each party with probability ρ.



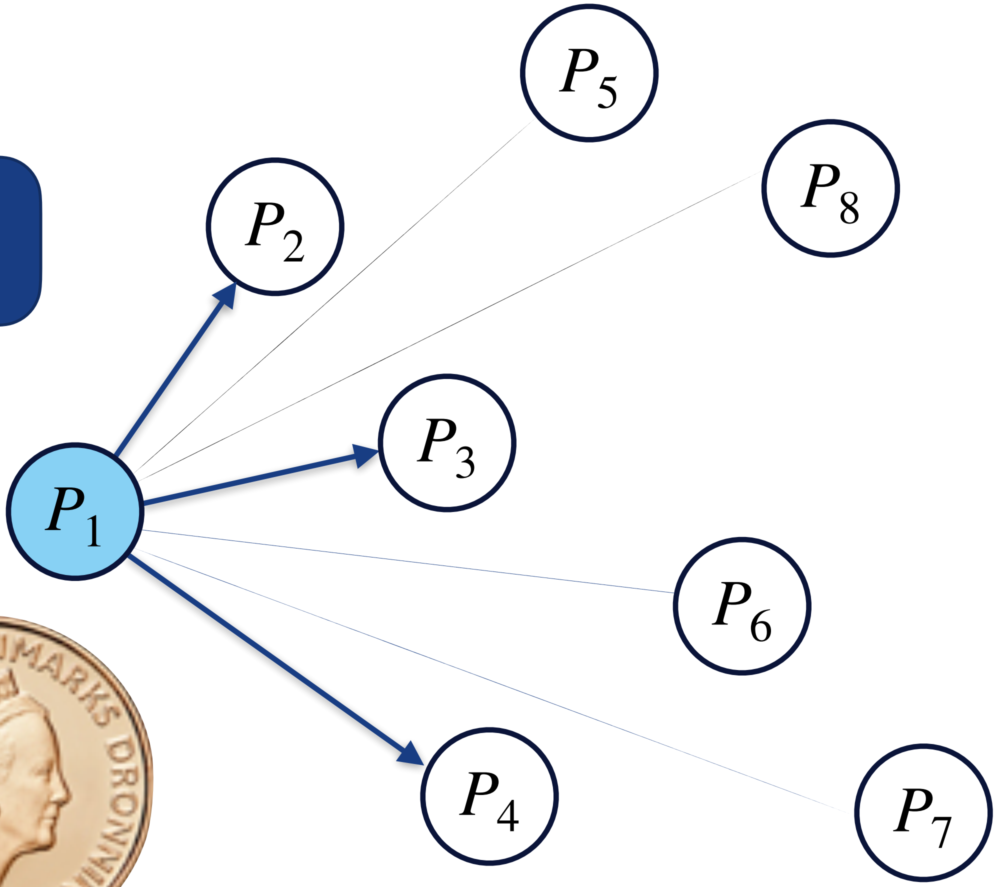
FLOODING PROTOCOL: $\pi_{\mathbf{ERFlood}}(\rho)$

Forward each message to each party with probability ρ.



FLOODING PROTOCOL: $\pi_{\text{ERFlood}}(\rho)$

Forward each message to each party with probability ρ.



MAIN RESULT (INFORMAL)

Theorem 3. *The protocol* π_{ERFlood} *implements a flooding network against an adversary that is delayed for the time it takes to send plus the time it takes to resend with either:*

1. $\Omega\left(\sqrt{\kappa \cdot n}\right)$ *neighborhood and a diameter of 2.*
2. $\Omega(\kappa)$ *neighborhood and a logarithmic diameter.*

ASSUMPTIONS

ASSUMPTIONS

- **Point-to-point Channels:** $F^{\sigma,\Delta}_{\textbf{MT}}$

ASSUMPTIONS

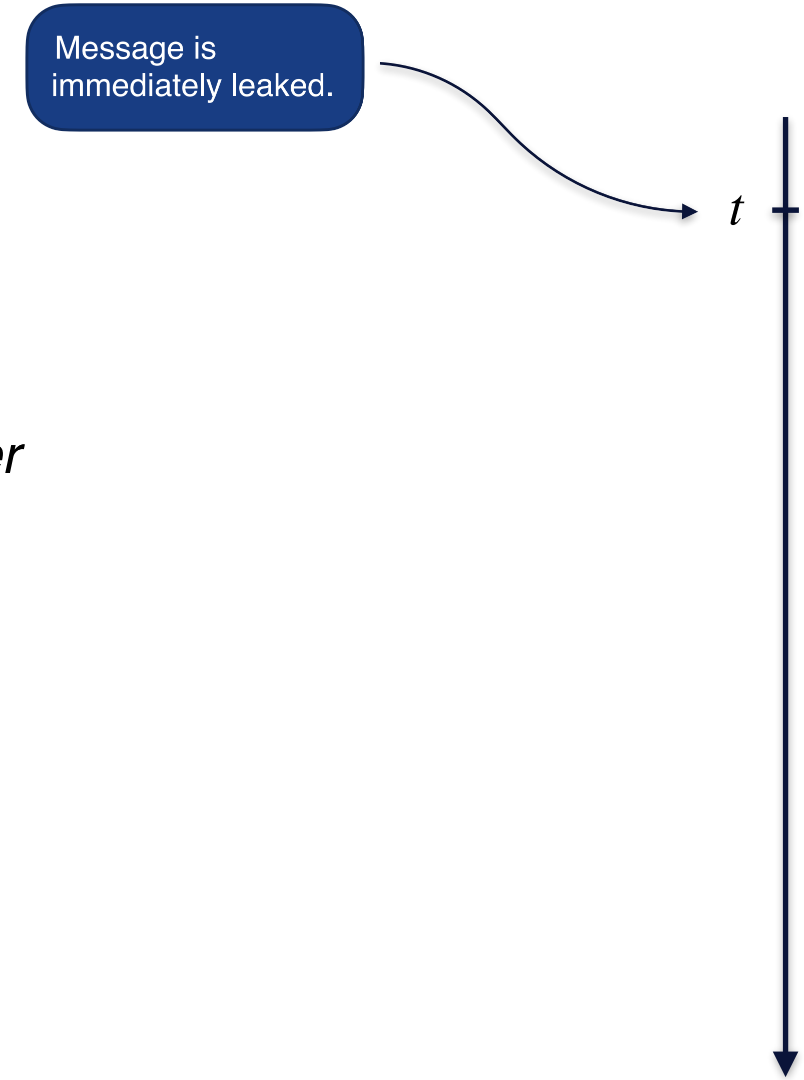
- **Point-to-point Channels:** $F^{\sigma,\Delta}_{\mathbf{MT}}$
  - *Messages input at time t must be delivered before time $t + \Delta$ if sender stays honest until time $t + \sigma$.*

ASSUMPTIONS

- **Point-to-point Channels:** $F_{\mathbf{MT}}^{\sigma,\Delta}$
  - *Messages input at time t must be delivered before time $t + \Delta$ if sender stays honest until time $t + \sigma$.*

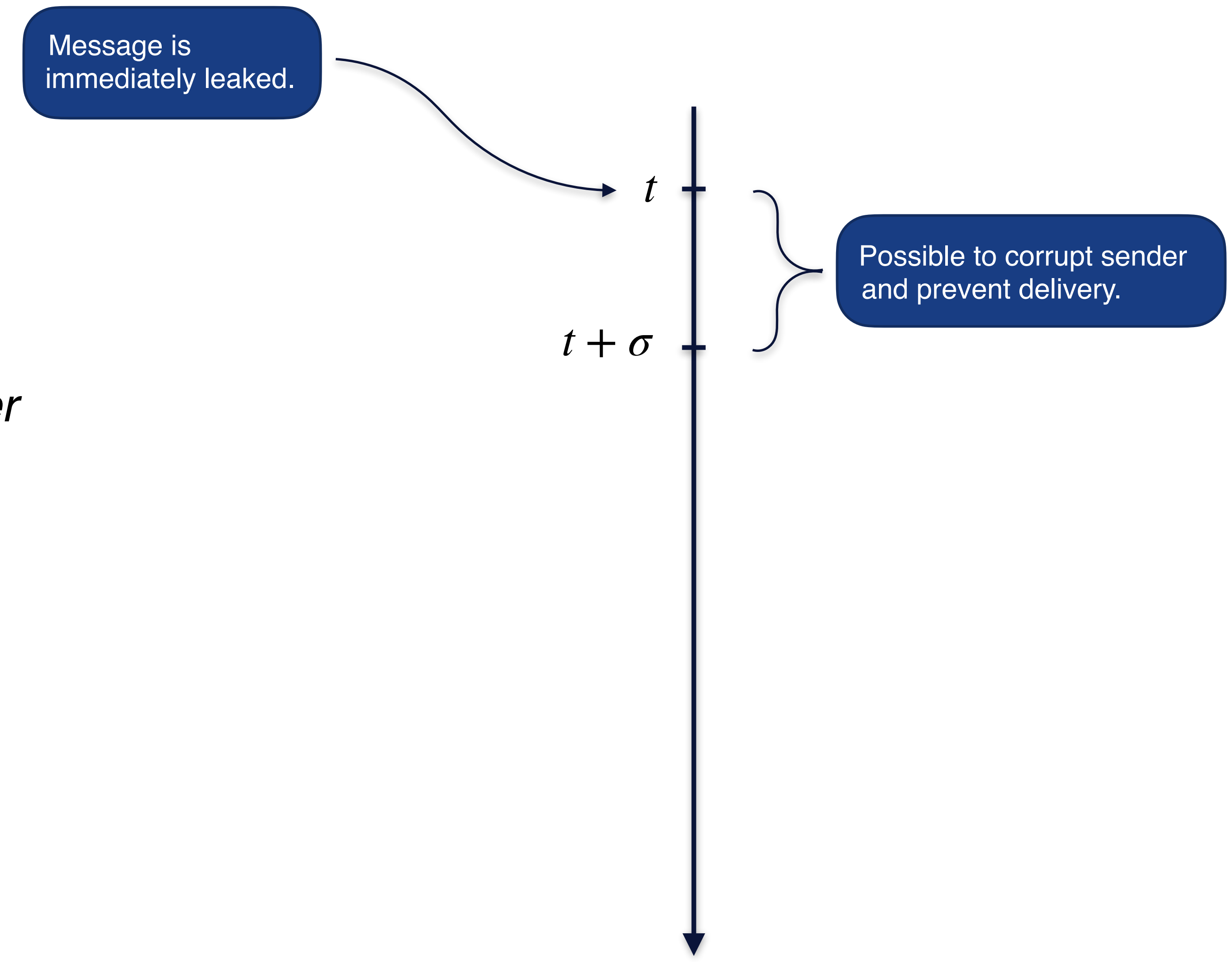
ASSUMPTIONS

- **Point-to-point Channels:** $F^{\sigma,\Delta}_{\mathbf{MT}}$
  - *Messages input at time t must be delivered before time $t+\Delta$ if sender stays honest until time $t+\sigma$.*



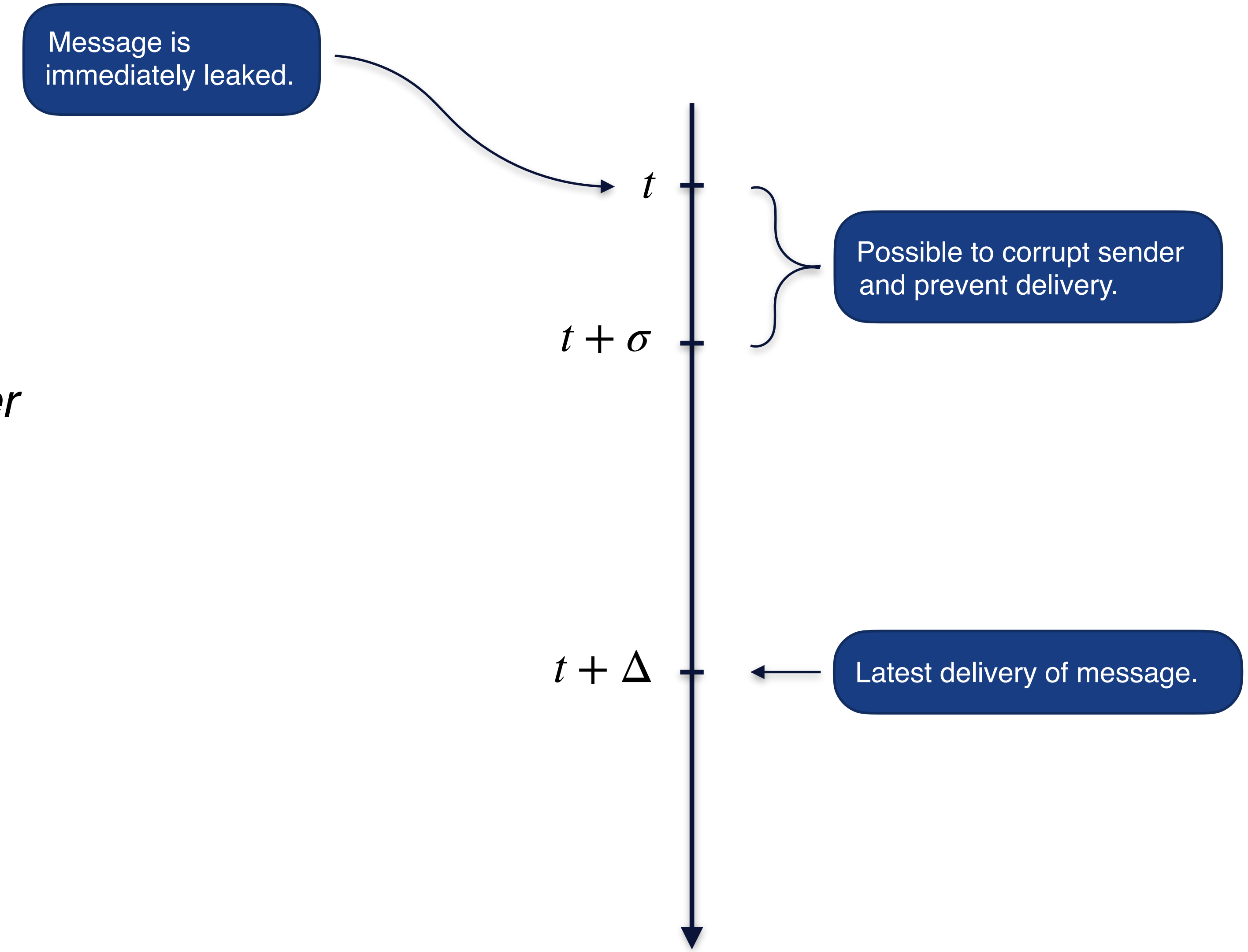
ASSUMPTIONS

- **Point-to-point Channels:** $F_{\mathbf{MT}}^{\sigma,\Delta}$
  - *Messages input at time t must be delivered before time $t+\Delta$ if sender stays honest until time $t+\sigma$.*



ASSUMPTIONS

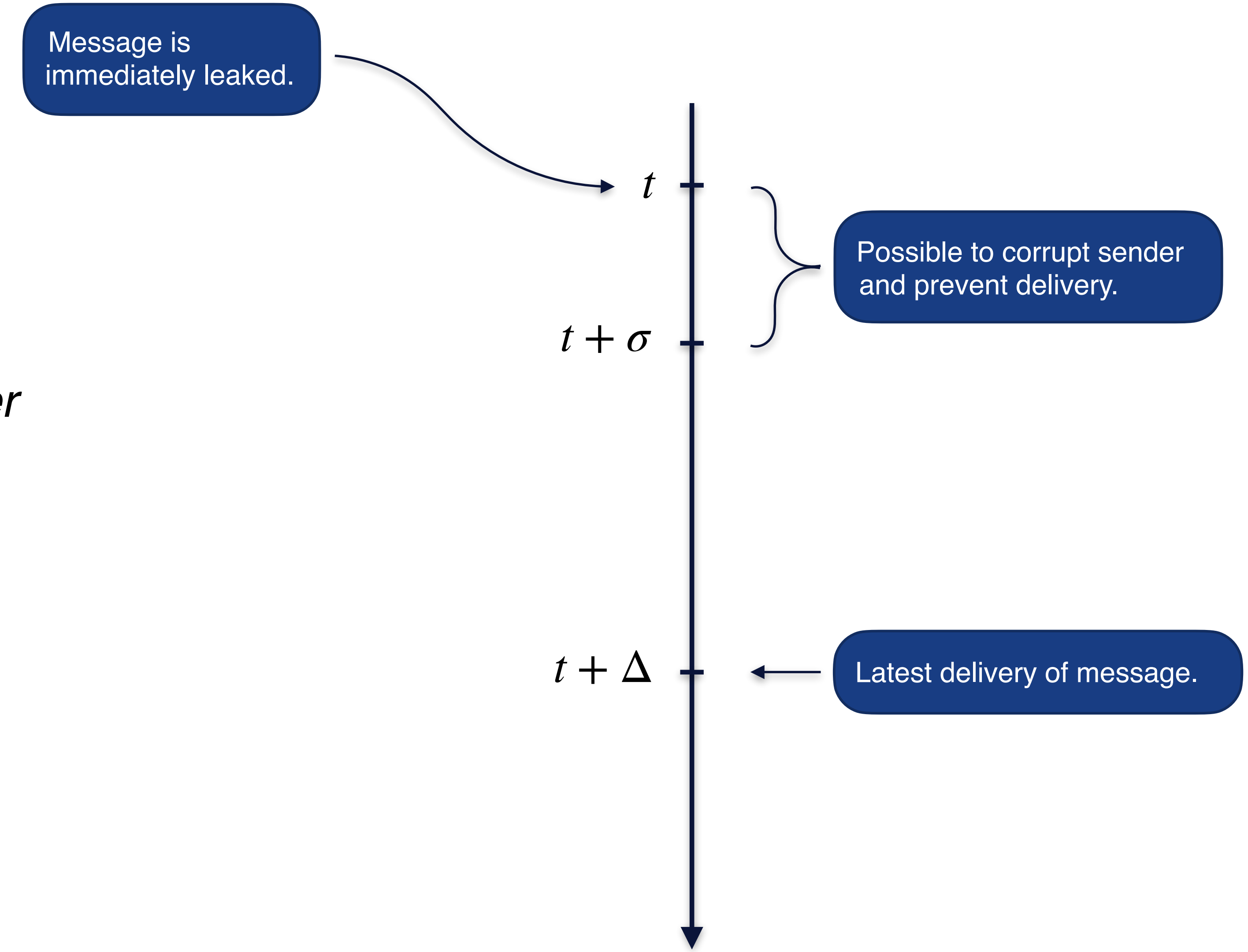
- **Point-to-point Channels:** $F^{\sigma,\Delta}_{\textbf{MT}}$
  - *Messages input at time t must be delivered before time $t+\Delta$ if sender stays honest until time $t+\sigma$.*



ASSUMPTIONS

- **Point-to-point Channels:** $F^{\sigma,\Delta}_{\textbf{MT}}$
  - *Messages input at time t must be delivered before time $t+\Delta$ if sender stays honest until time $t+\sigma$.*

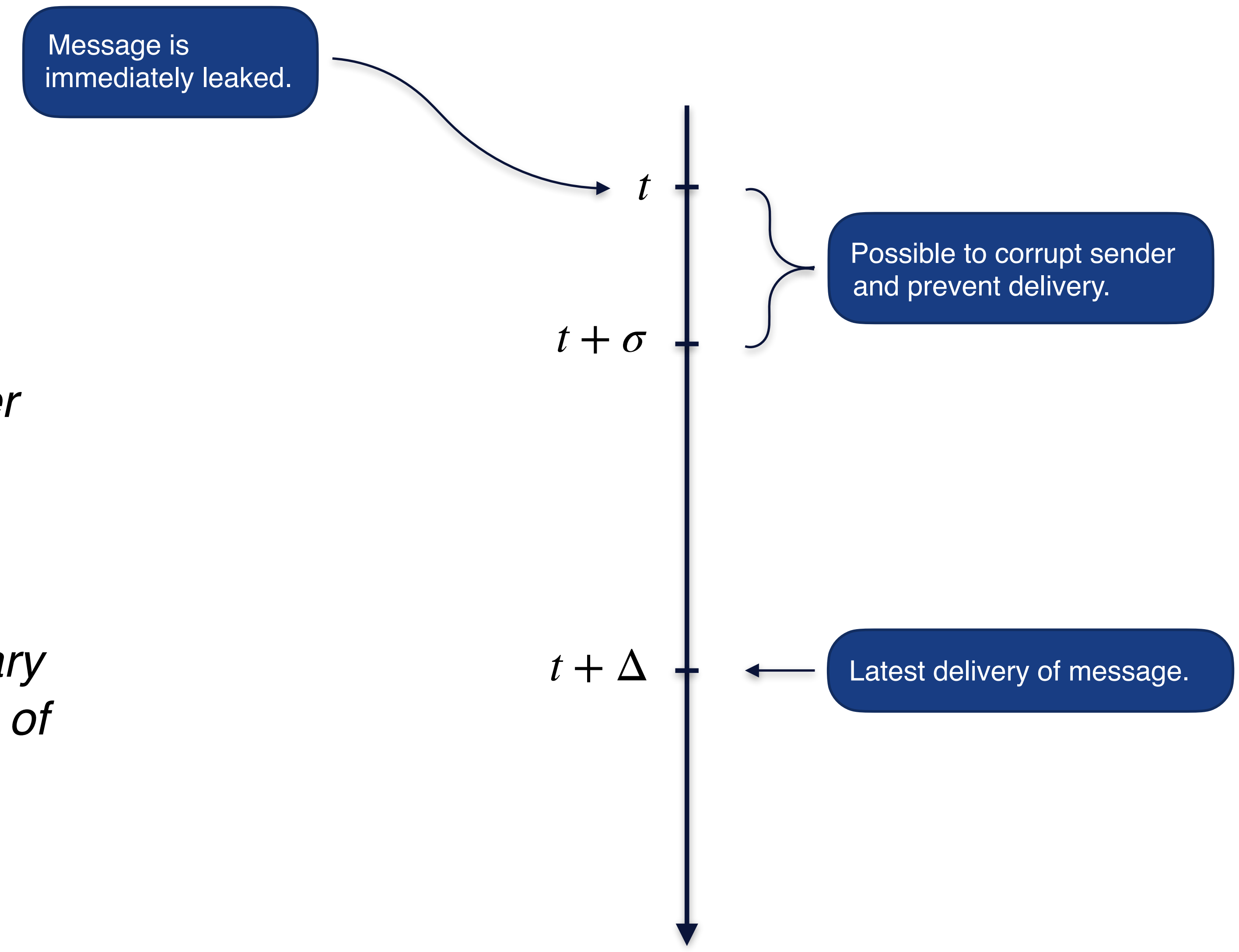
- **Delayed Adversary:**



ASSUMPTIONS

- **Point-to-point Channels:** $F^{\sigma,\Delta}_{\textsf{MT}}$
  - *Messages input at time t must be delivered before time $t+\Delta$ if sender stays honest until time $t+\sigma$.*

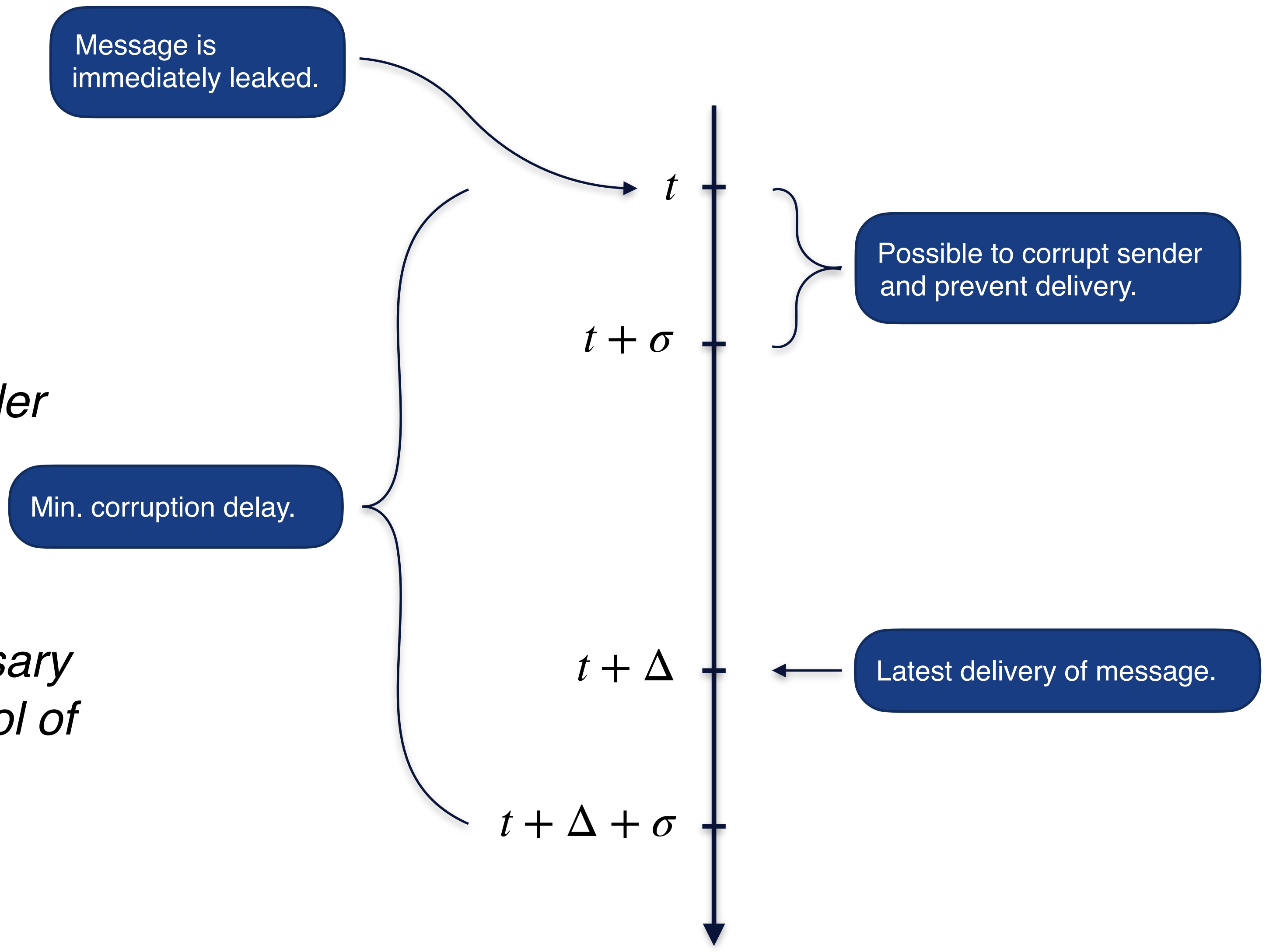
- **Delayed Adversary:**
  - *It takes $\Delta+\sigma$ time from an adversary decides to corrupt a party, to control of the party is given to the adversary.*



ASSUMPTIONS

- **Point-to-point Channels:** $F^{\sigma,\Delta}_{\textbf{MT}}$
  - *Messages input at time t must be delivered before time $t + \Delta$ if sender stays honest until time $t + \sigma$.*

- **Delayed Adversary:**
  - *It takes $\Delta + \sigma$ time from an adversary decides to corrupt a party, to control of the party is given to the adversary.*



FUNCTIONALITY: $F^{\Delta'}_{\textbf{Flood}}$

FUNCTIONALITY: $F^{\Delta'}_{\textbf{Flood}}$

Any message input by an honest party at time t must be delivered to all other honest parties before time $t + \Delta'$.

FUNCTIONALITY: $F^{\Delta'}_{\textbf{Flood}}$

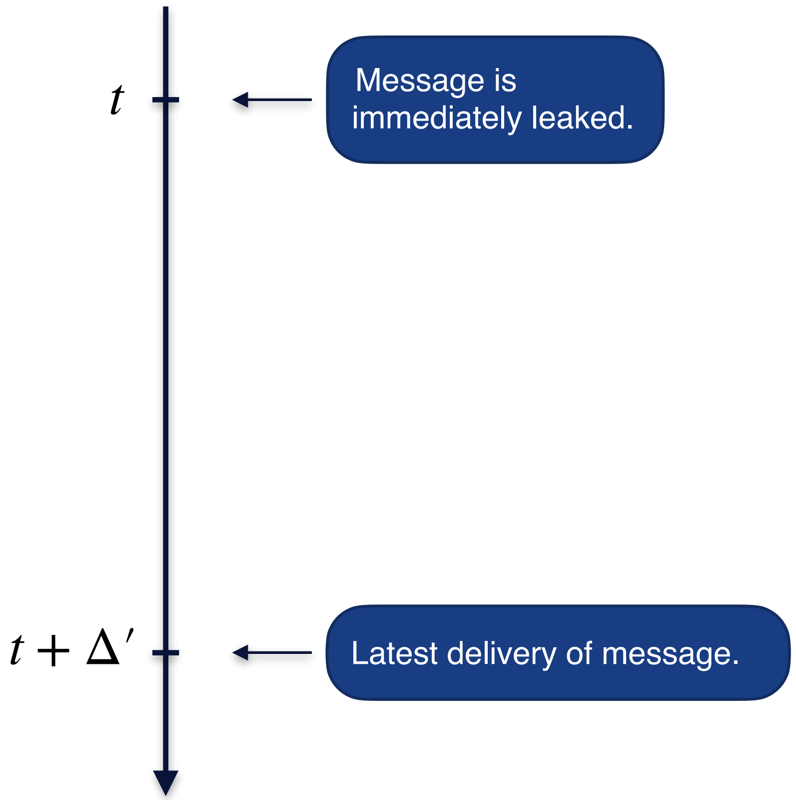
Any message input by an honest party at time t must be delivered to all other honest parties before time $t + \Delta'$.

FUNCTIONALITY: $F^{\Delta'}_{\textbf{Flood}}$

Any message input by an honest party at time t must be delivered to all other honest parties before time $t + \Delta'$.

FUNCTIONALITY: $F^{\Delta'}_{\textbf{Flood}}$

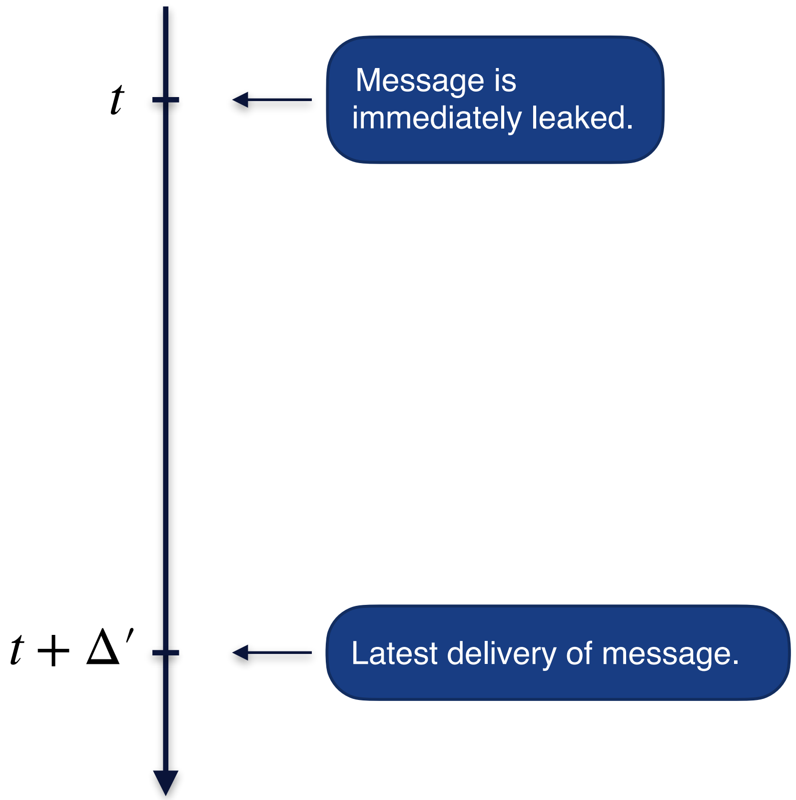
Any message input by an honest party at time t must be delivered to all other honest parties before time $t + \Delta'$.



FUNCTIONALITY: $F^{\Delta'}_{\textbf{Flood}}$

Adversary has not initiated the corruption.

*Any message input by an **honest** party at time t must be delivered to all other honest parties before time $t + \Delta'$.*



MAIN RESULT

Theorem 3. *The protocol* $\pi_{\mathsf{ERFlood}}(\rho)$ *UC-realises the functionality* $F^{\Delta'}_{\mathsf{Flood}}$ *in the* $F^{\sigma,\Delta}_{\mathsf{MT}}$*-hybrid world against a* $(\sigma + \Delta)$*-delayed adversary if either:*

MAIN RESULT

Theorem 3. *The protocol $\pi_{\textit{ERFlood}}(\rho)$ UC-realises the functionality $F^{\Delta'}_{\textit{Flood}}$ in the $F^{\sigma,\Delta}_{\textit{MT}}$-hybrid world against a $(\sigma + \Delta)$-delayed adversary if either:*

1. $\rho = \sqrt{\dfrac{\kappa}{h}}$ *and* $\Delta' = 2 \cdot \Delta$;

κ = security parameter.
h = number of parties guaranteed to be honest.
n = number of parties in total.

MAIN RESULT

Theorem 3. *The protocol* $\pi_{\textit{ERFlood}}(\rho)$ *UC-realises the functionality* $F^{\Delta'}_{\textit{Flood}}$ *in the* $F^{\sigma,\Delta}_{\textit{MT}}$*-hybrid world against a* $(\sigma + \Delta)$*-delayed adversary if either:*

1. $\rho = \sqrt{\dfrac{\kappa}{h}}$ *and* $\Delta' = 2 \cdot \Delta$;

κ = security parameter.
h = number of parties guaranteed to be honest.
n = number of parties in total.

MAIN RESULT

Theorem 3. *The protocol* $\pi_{\textsf{ERFlood}}(\rho)$ *UC-realises the functionality* $F^{\Delta'}_{\textsf{Flood}}$ *in the* $F^{\sigma,\Delta}_{\textsf{MT}}$*-hybrid world against a* $(\sigma + \Delta)$*-delayed adversary if either:*

1. $\rho = \sqrt{\dfrac{\kappa}{h}}$ *and* $\Delta' = 2 \cdot \Delta$.

2. $\rho = \dfrac{\kappa}{h}$ *and* $\Delta' = \mathcal{O}\left(\Delta \cdot \log\left(\dfrac{n}{\kappa}\right)\right)$.

κ = security parameter.
h = number of parties guaranteed to be honest.
n = number of parties in total.

CONCLUSION

- Formal model for δ-delayed adversaries within UC.

CONCLUSION

- Formal model for δ-delayed adversaries within UC.
- Two instantiations of a flooding network secure against adaptive adversaries:

CONCLUSION

- Formal model for δ-delayed adversaries within UC.
- Two instantiations of a flooding network secure against adaptive adversaries:
  - One with a constant neighborhood and logarithmic diameter.

CONCLUSION

- Formal model for δ-delayed adversaries within UC.

- Two instantiations of a flooding network secure against adaptive adversaries:
  - One with a constant neighborhood and logarithmic diameter.
  - One with a squareroot neighborhood and constant diameter.

CONCLUSION

- Formal model for δ-delayed adversaries within UC.

- Two instantiations of a flooding network secure against adaptive adversaries:
  - One with a constant neighborhood and logarithmic diameter.
  - One with a squareroot neighborhood and constant diameter.

- Details: https://eprint.iacr.org/2022/010.

CONCLUSION

- Formal model for δ-delayed adversaries within UC.

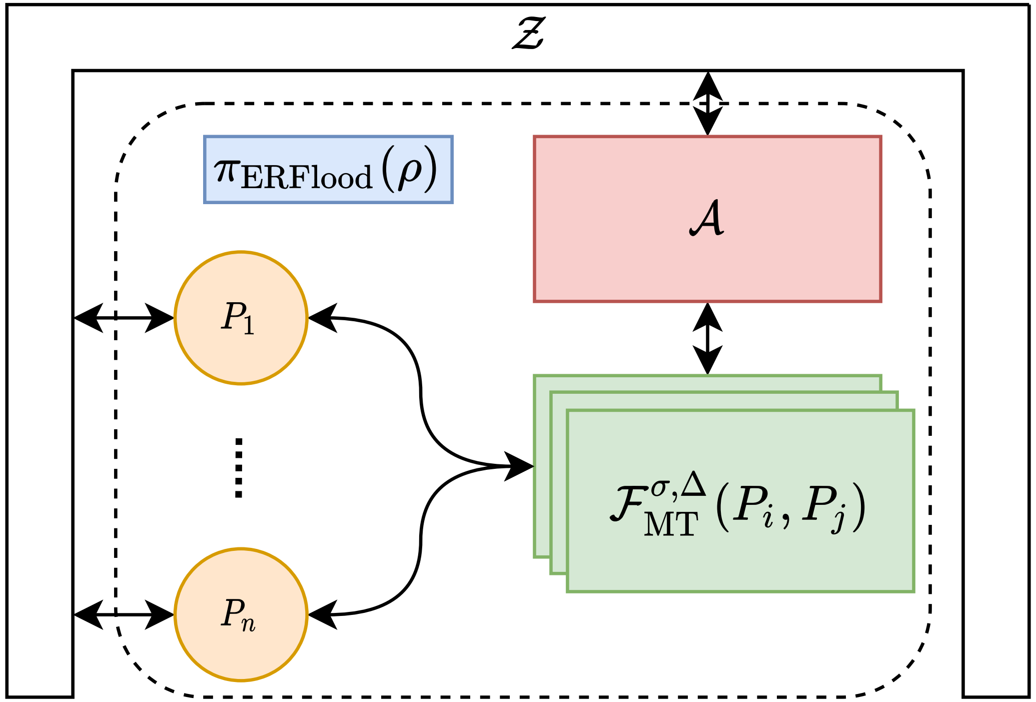
- Two instantiations of a flooding network secure against adaptive adversaries:
  - One with a constant neighborhood and logarithmic diameter.
  - One with a squareroot neighborhood and constant diameter.

- Details: https://eprint.iacr.org/2022/010.

- Contact: sethomsen@cs.au.dk.

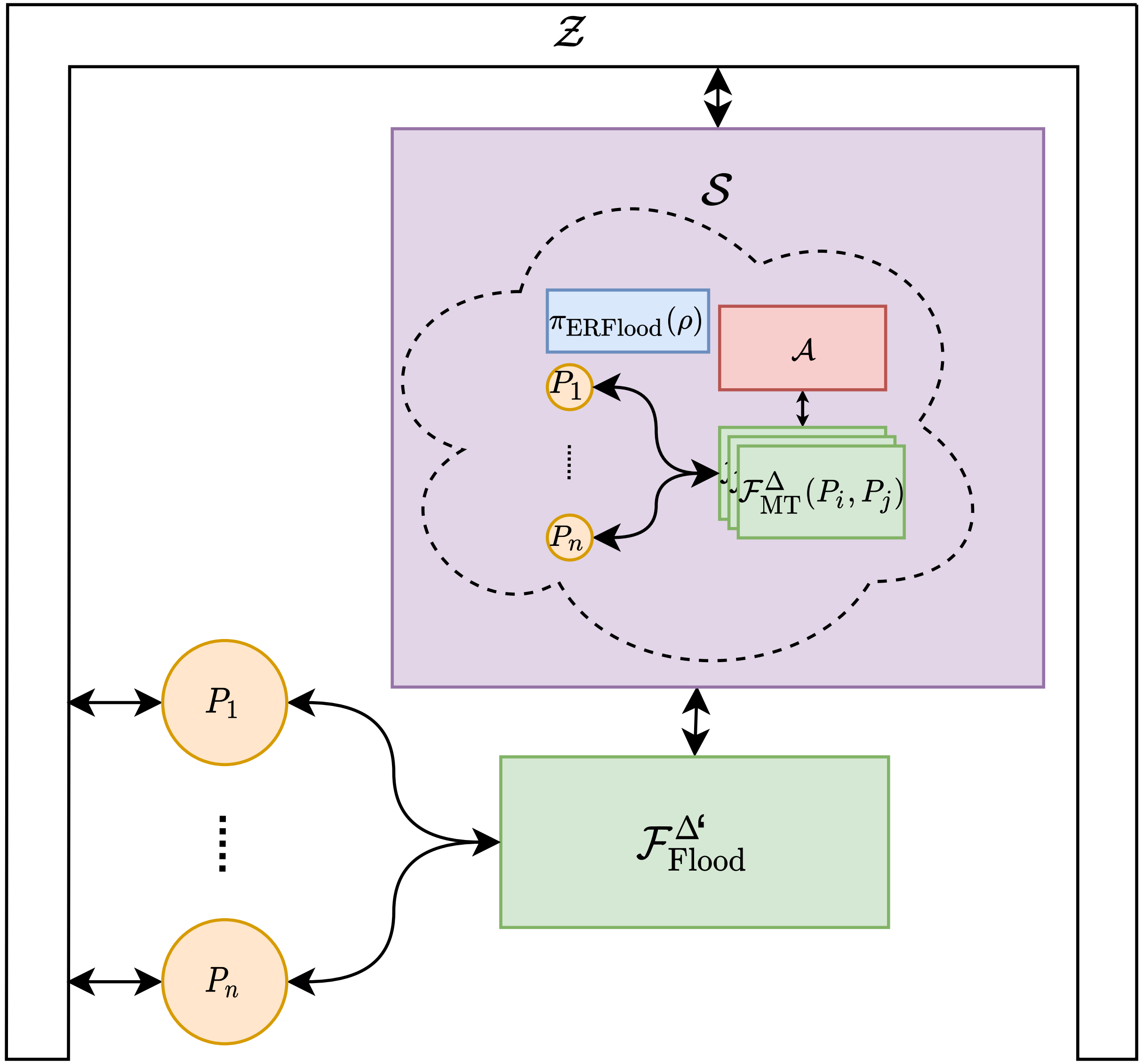
PROOF IDEA UC

The Hybrid World



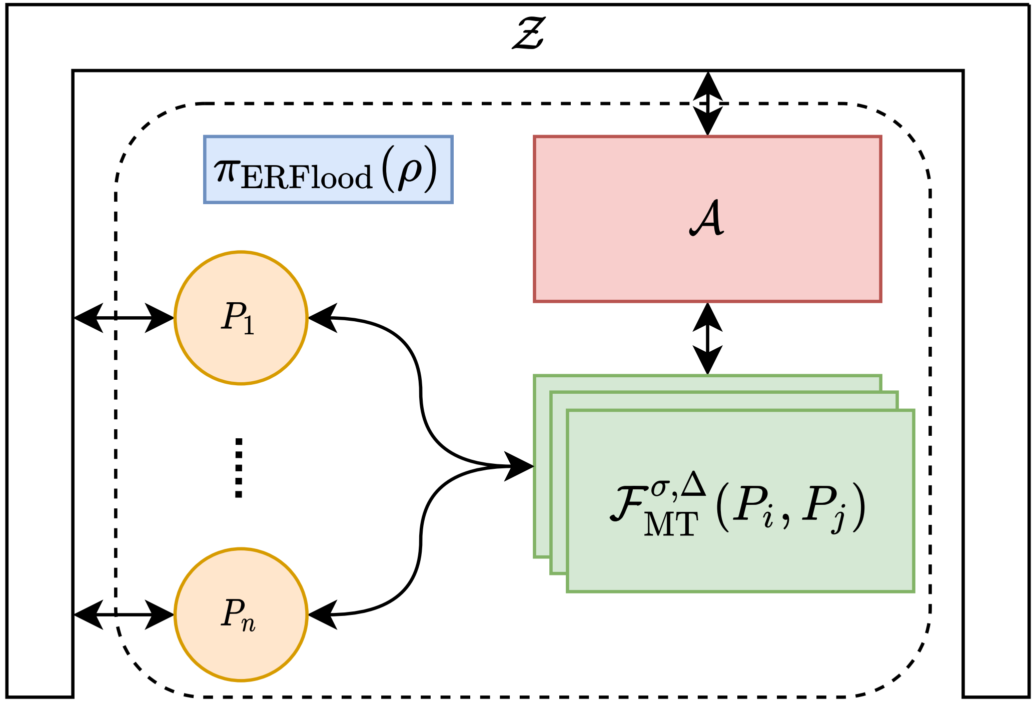
$\approx$

The Ideal World



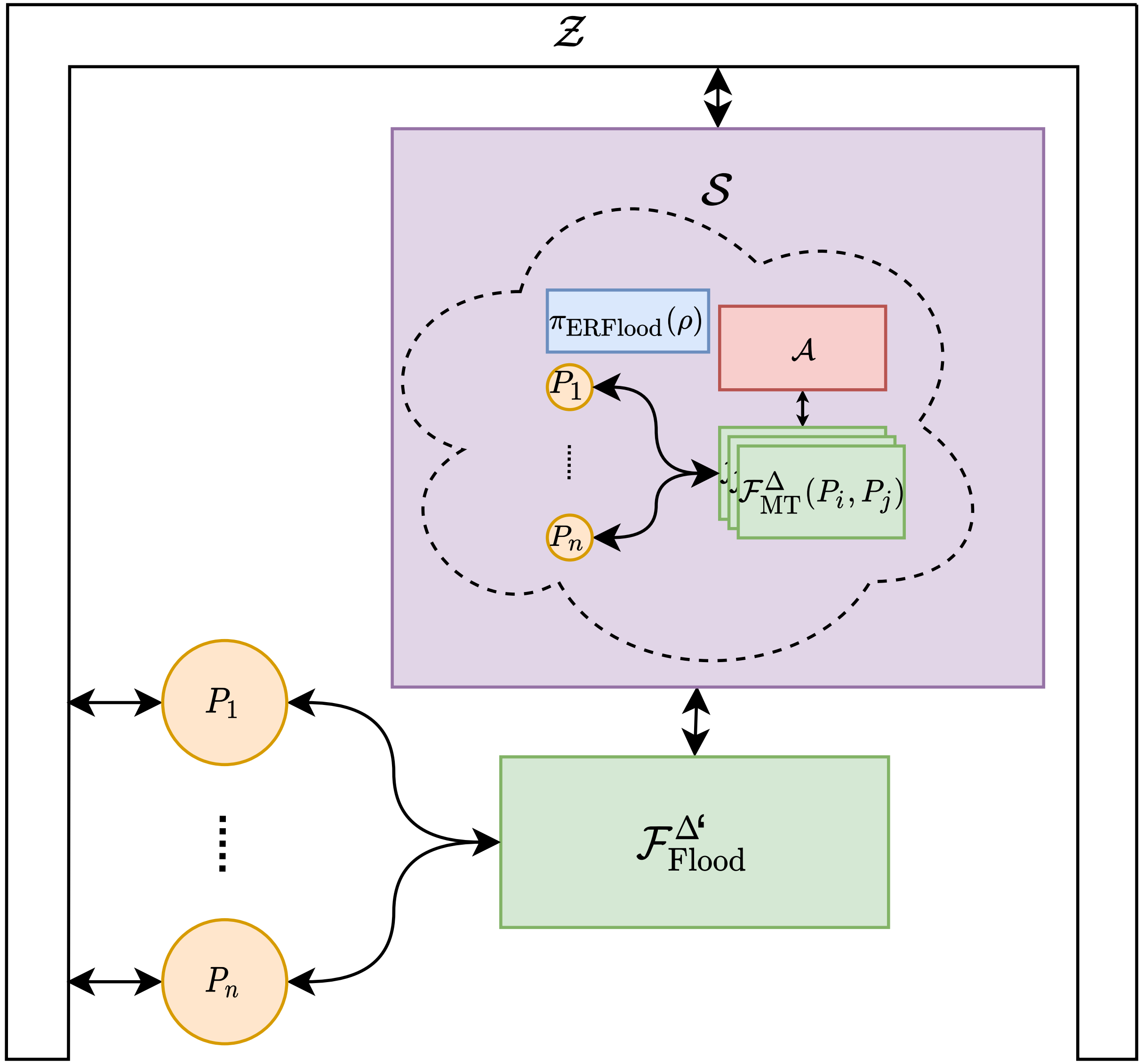
PROOF IDEA UC

The Hybrid World



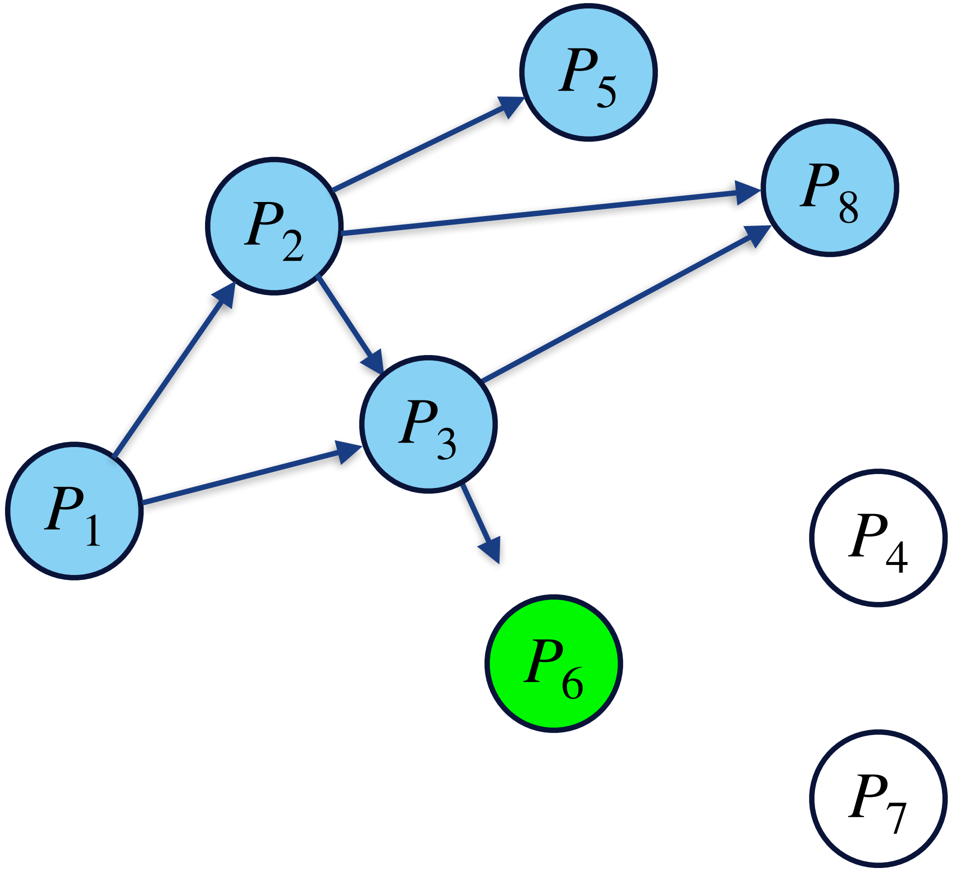
$\approx$

The Ideal World

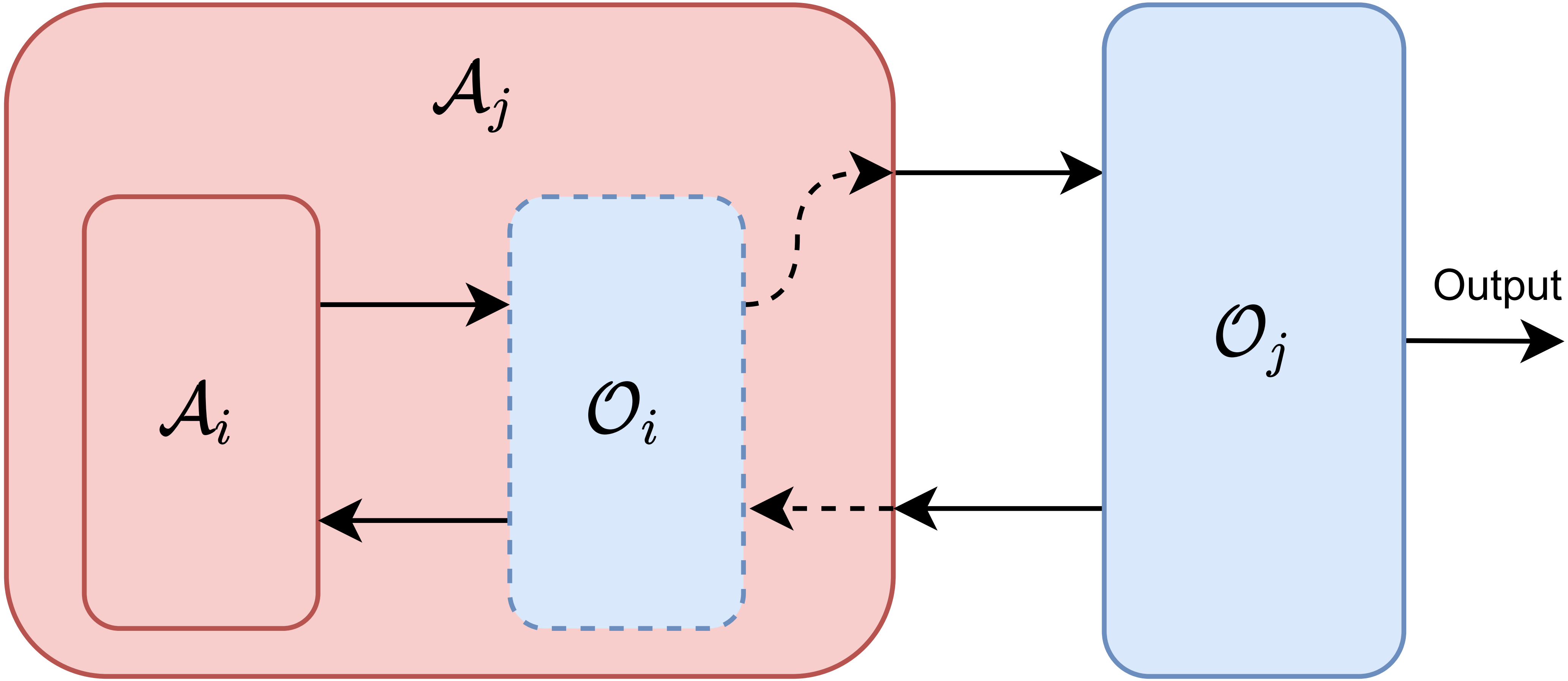


Sufficient to bound probability for late delivery!

DYNAMIC SIZE IS AN ADVANTAGE



GAME-HOPS



REFERENCES

[GKL15]: Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. In Elisabeth Oswald and Marc Fischlin, editors, EUROCRYPT 2015, Part II, volume 9057 of LNCS, pages 281–310, Sofia, Bulgaria, April 26–30, 2015. Springer, Heidelberg, Germany.

[GKL17]: Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol with chains of variable difficulty. In Jonathan Katz and Hovav Shacham, editors, CRYPTO 2017, Part I, volume 10401 of LNCS, pages 291–323, Santa Barbara, CA, USA, August 20–24, 2017. Springer, Heidelberg, Germany.

[PSs17]: Rafael Pass, Lior Seeman, and abhi shelat. Analysis of the blockchain protocol in asynchronous networks. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, EUROCRYPT 2017, Part II, volume 10211 of LNCS, pages 643–673, Paris, France, April 30 – May 4, 2017. Springer, Heidelberg, Germany.

[PS17]: Rafael Pass and Elaine Shi. Hybrid consensus: Efficient consensus in the permissionless model. In Andréa W. Richa, editor, 31st International Symposium on Distributed Computing, DISC 2017, October 16-20, 2017, Vienna, Austria, volume 91 of LIPIcs, pages 39:1–39:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[DGKR18]: Bernardo David, Peter Gazi, Aggelos Kiayias, and Alexander Russell. Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain. In Jes- per Buus Nielsen and Vincent Rijmen, editors, EUROCRYPT 2018, Part II, volume 10821 of LNCS, pages 66–98, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Heidelberg, Germany.

[BDDNO21]: Carsten Baum, Bernardo David, Rafael Dowsley, Jesper Buus Nielsen, and Sabine Oechsner. TARDIS: A foundation of time-lock puzzles in UC. In EUROCRYPT (3), volume 12698 of Lecture Notes in Computer Science, pages 429–459. Springer, 2021.