A Better Method to Analyze Blockchain Consistency

Lucianna Kiffer Rajmohan Rajaraman abhi shelat

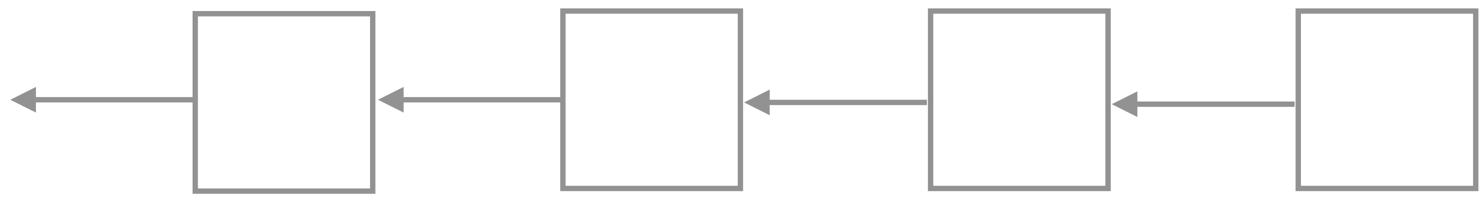
Northeastern University

AFT 2019

BLOCKCHAIN 101

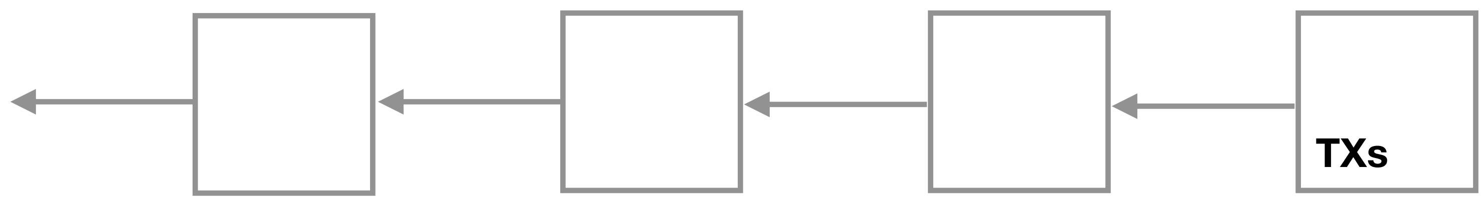
BLOCKCHAIN 101

(i) Chain of blocks



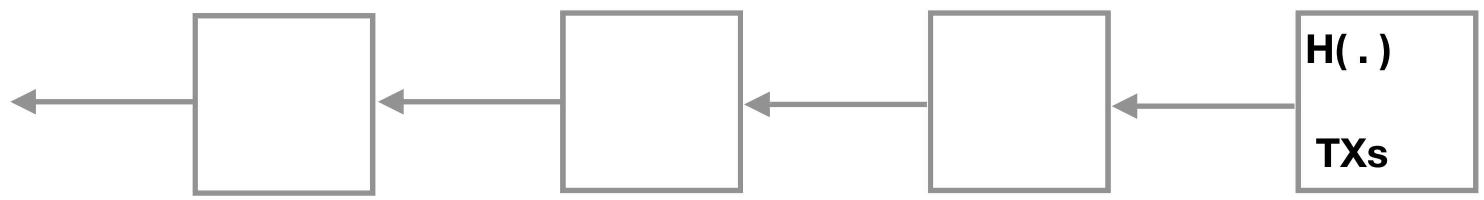
BLOCKCHAIN 101

(i) Chain of blocks



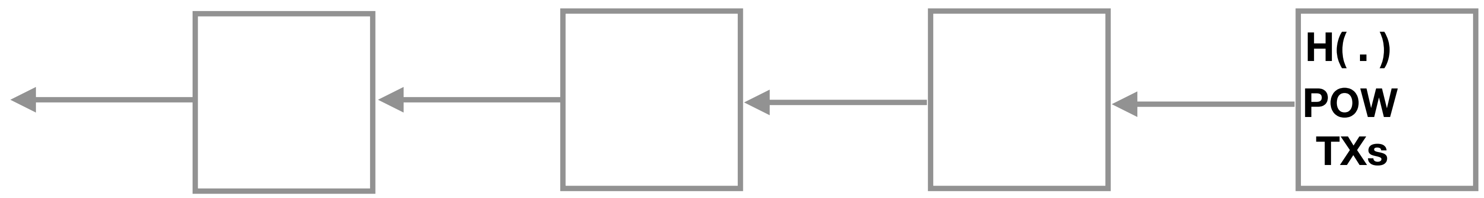
BLOCKCHAIN 101

(i) Chain of blocks



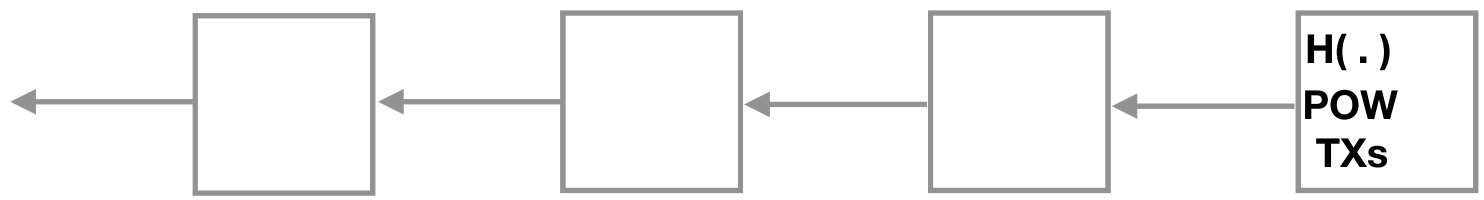
BLOCKCHAIN 101

(i) Chain of blocks

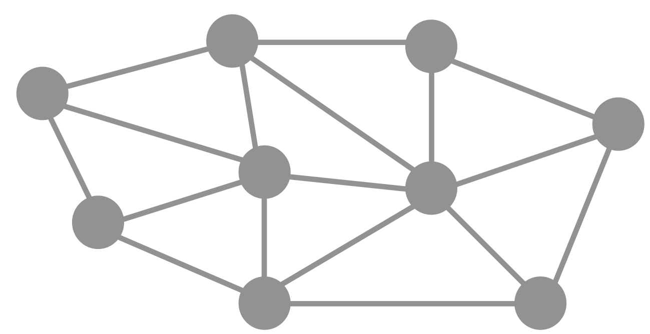


BLOCKCHAIN 101

(i) Chain of blocks

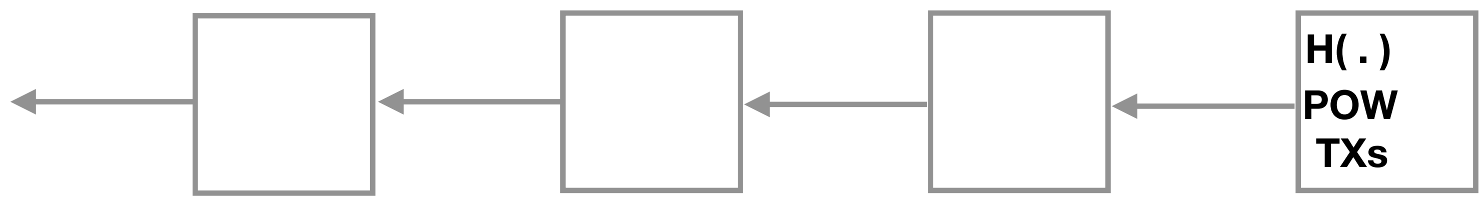


(ii) Peer-to-peer network

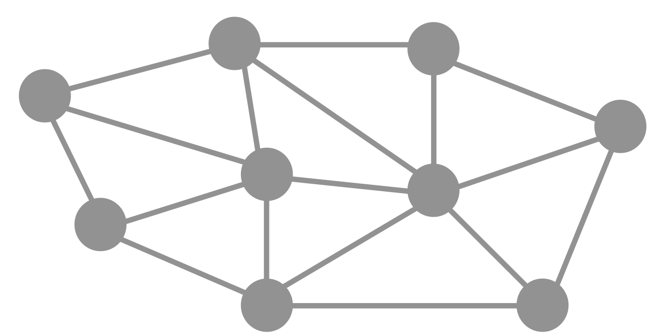


BLOCKCHAIN 101

(i) Chain of blocks



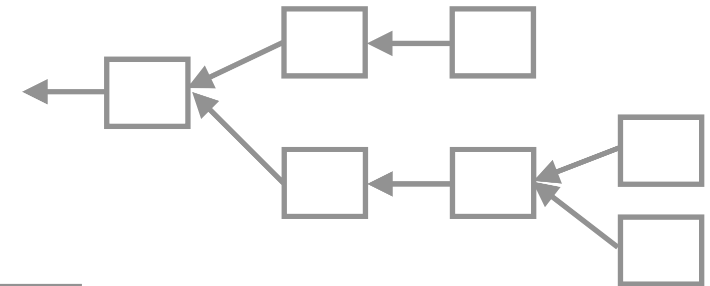
(ii) Peer-to-peer network



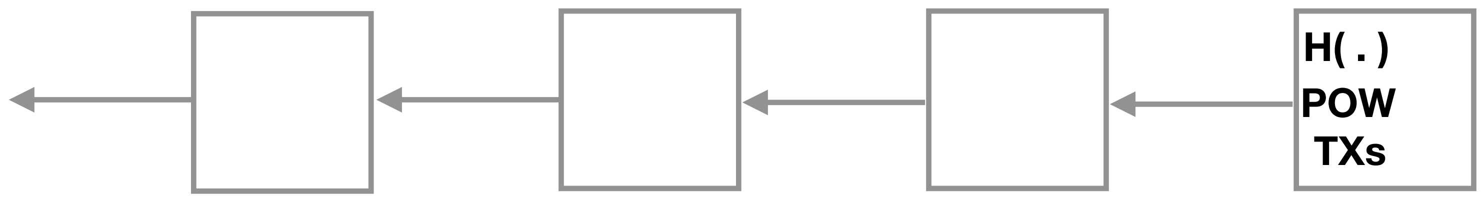
(iii) Mining rule

e.g. longest chain

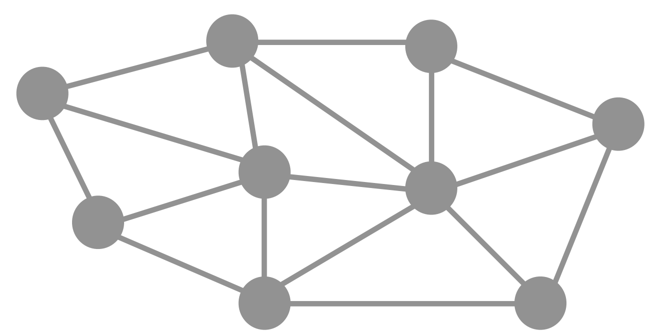
BLOCKCHAIN 101



(i) Chain of blocks



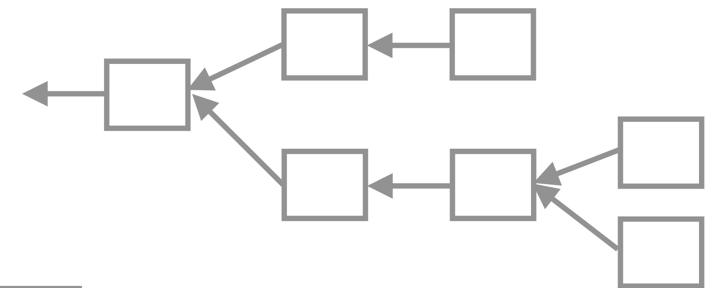
(ii) Peer-to-peer network



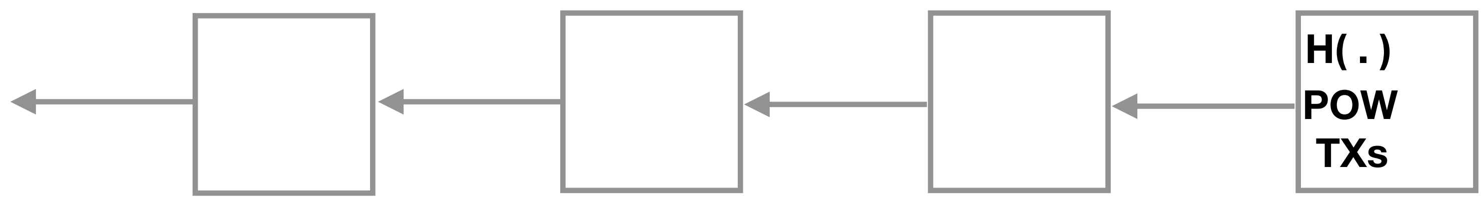
(iii) Mining rule

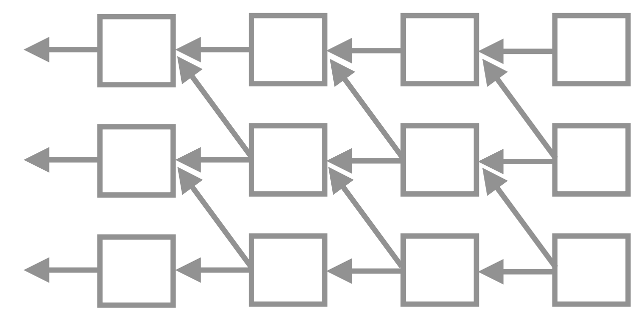
e.g. longest chain

BLOCKCHAIN 101

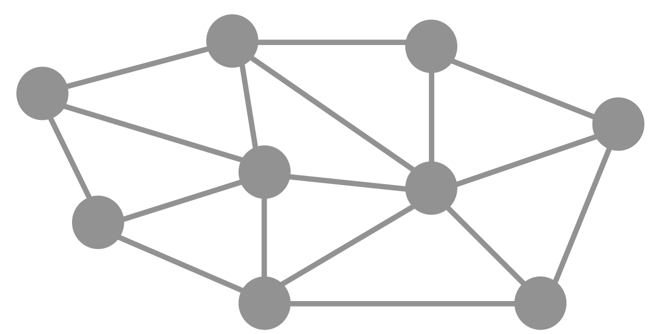


(i) Chain of blocks





(ii) Peer-to-peer network



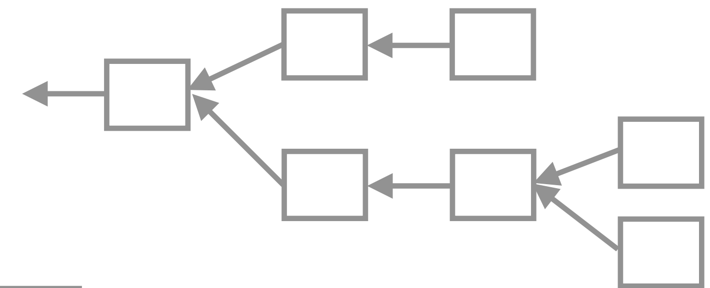
(iii) Mining rule

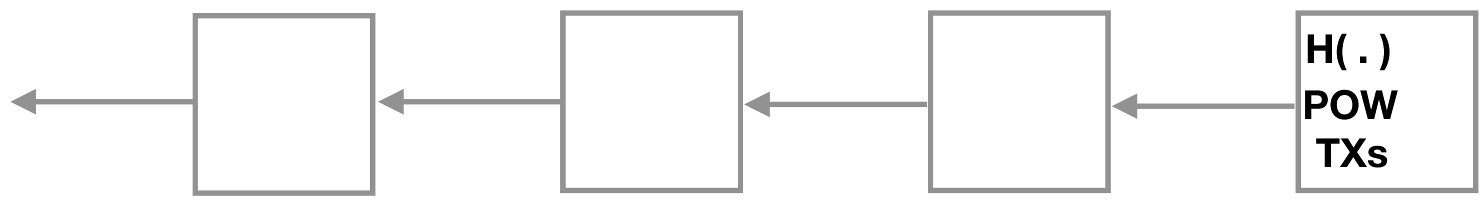
e.g. longest chain

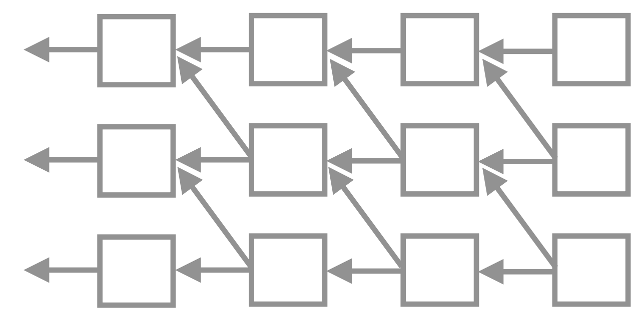
BLOCKCHAIN 101

DAG

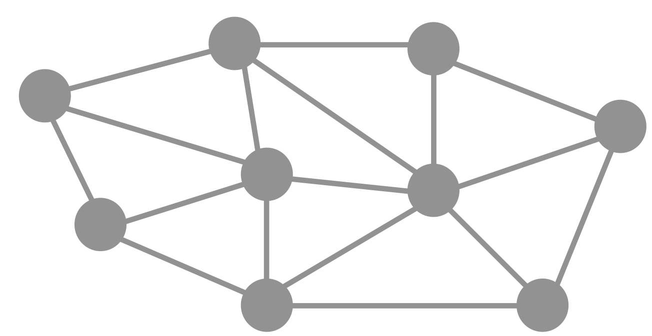
(i) ~~Chain~~ of blocks







(ii) Peer-to-peer network



(iii) Mining rule

e.g. longest chain

Related Work

- Formal framework for analyzing blockchain protocols:
    - Consistency(common prefix/persistence)
    - Chainquality
    - Chaingrowth (Liveness)

[*Pass, Seeman, shelat 2016], [Garay, Kiayias, Leonardos 2016, 2017*], [*Kiayias, Panagiotakos 2015]*

- DAG-based blockchain models
    - eg. GHOST, Spectre, Chainweb …

[Lewenberg, Sompolinsky, Zohar 2015,2016], [Sompolinsky, Zohar 2015], [Martino, Quaintance,Popejoy 2018]

- Attacks on chainquality, growth and consistency

[*Nakamoto 2009], [Eyal, Sirer 2013],[Kiayias, Panagiotakos 2015,2016], [Pass,Seeman, shelat 2016]*

Blockchain Definition

$(\Pi, \mathcal{C})$ Both algorithms use a security parameter k

$\Pi^V(k)$ Maintains a local variable state

V predicate defines the semantics of the blockchain

$\mathcal{C}(k, \text{state})$ On input (k, state), outputs a sequence of records $\vec{m}$

BLOCKCHAIN MODEL

Execution

$Z(1^\kappa)$ **Environment**, initializes players, either honest or corrupt. Provides inputs to all parties.

Execution

$Z(1^\kappa)$ **Environment**, initializes players, either honest or corrupt. Provides inputs to all parties.

Honest parties execute protocol Π^V

$\Pi^V(k)$

state_1

$\Pi^V(k)$

state_2

$\Pi^V(k)$

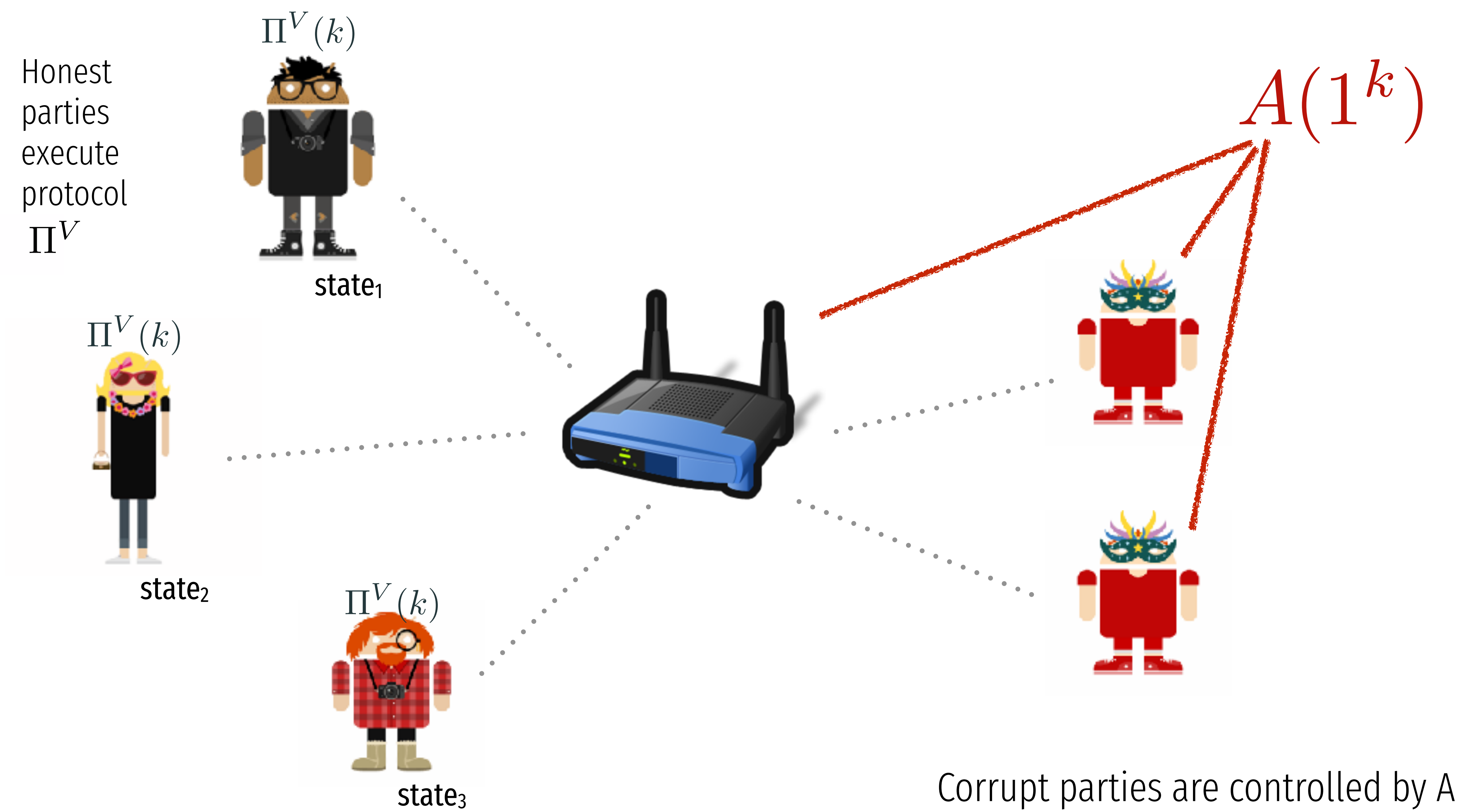
state_3

$A(1^k)$

Corrupt parties are controlled by A

Execution

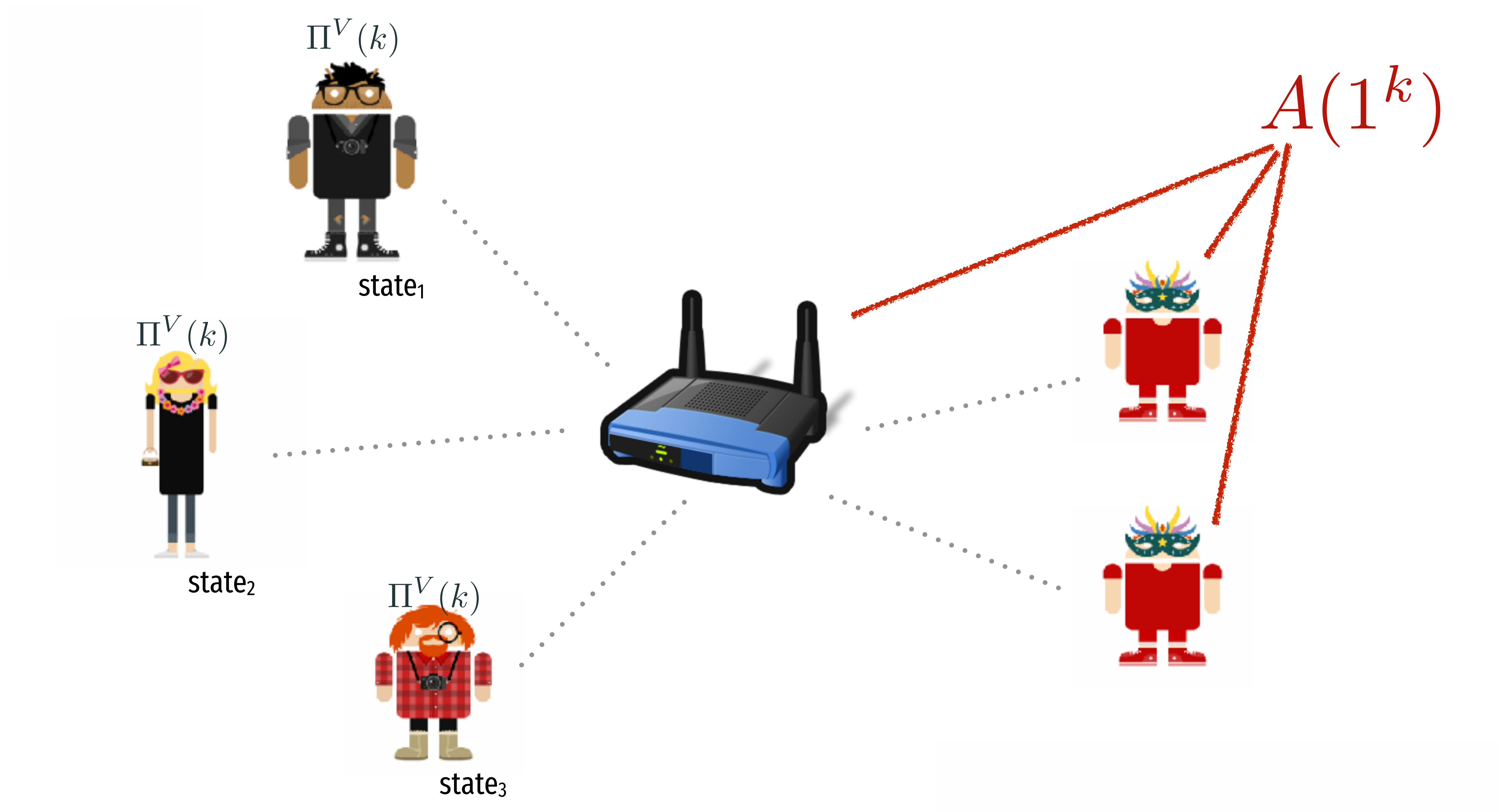
$Z(1^\kappa)$ **Environment**, initializes players, either honest or corrupt. Provides inputs to all parties.



Execution

$Z(1^\kappa)$ **Environment**, initializes players, either honest or corrupt. Provides inputs to all parties.

Round r

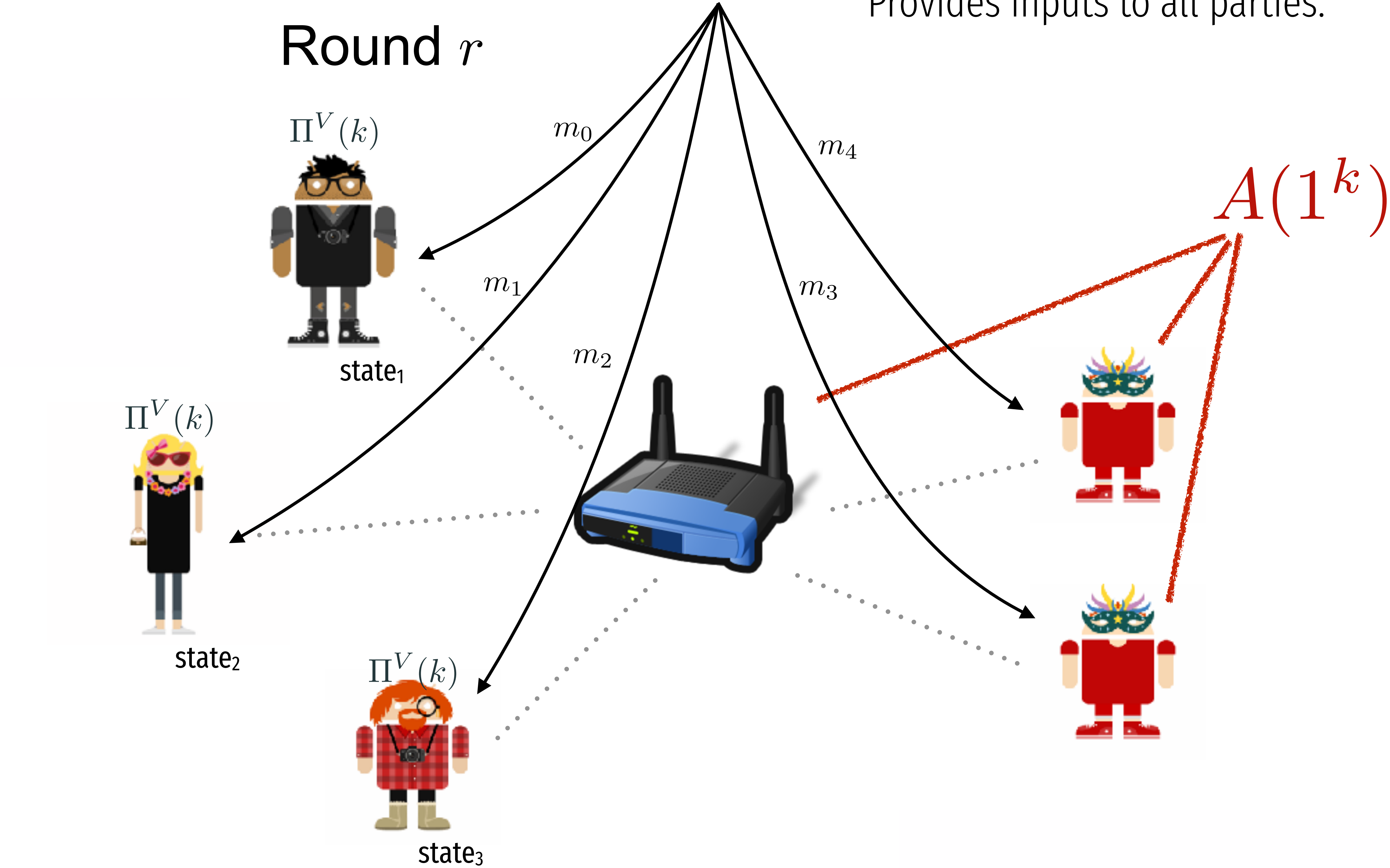


Execution

$Z(1^\kappa)$

Environment, initializes players, either honest or corrupt. Provides inputs to all parties.

Round r

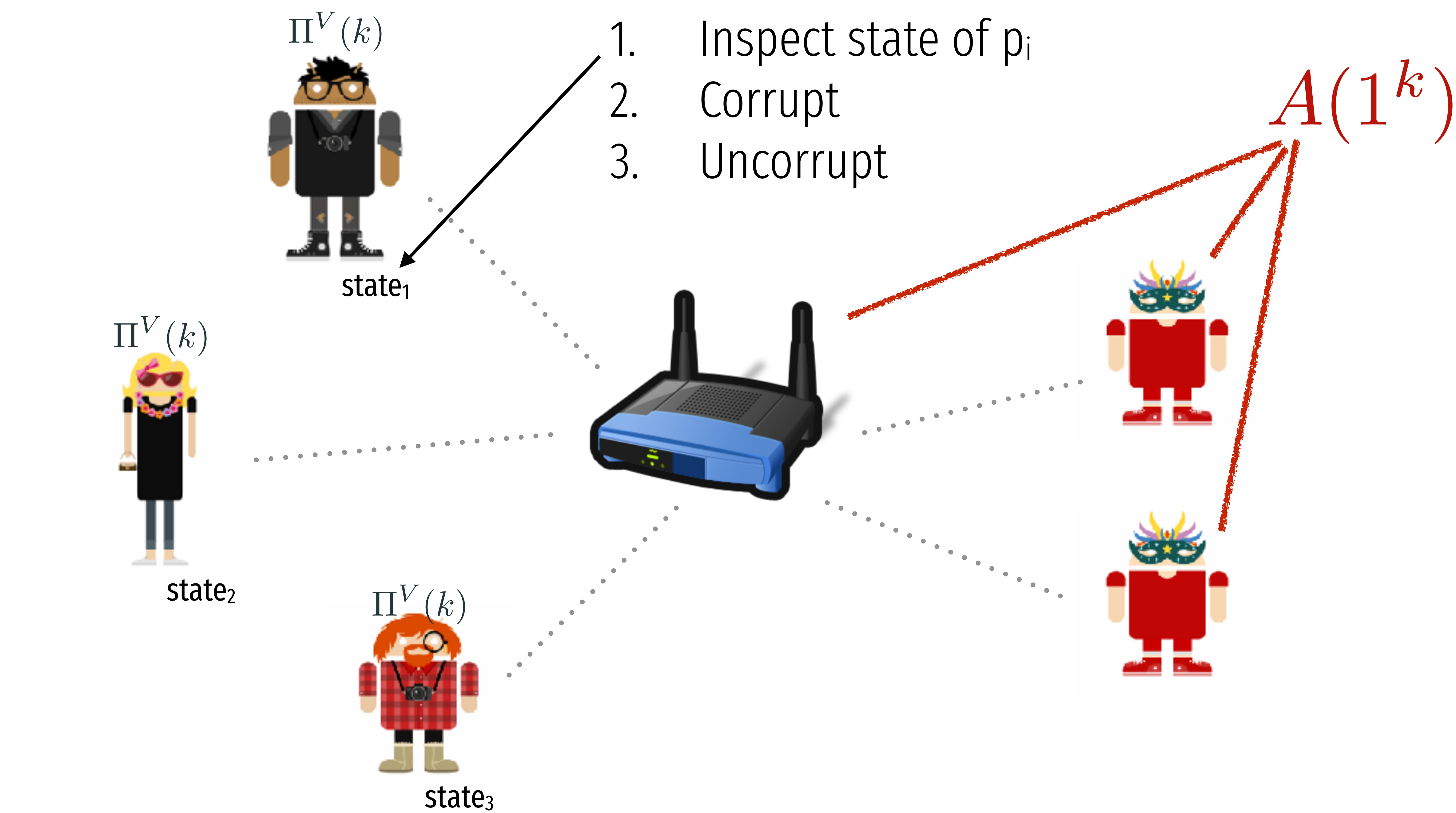


Execution

$Z(1^\kappa)$ **Environment**, initializes players, either honest or corrupt. Provides inputs to all parties.

Round r

1. Inspect state of p_i
2. Corrupt
3. Uncorrupt

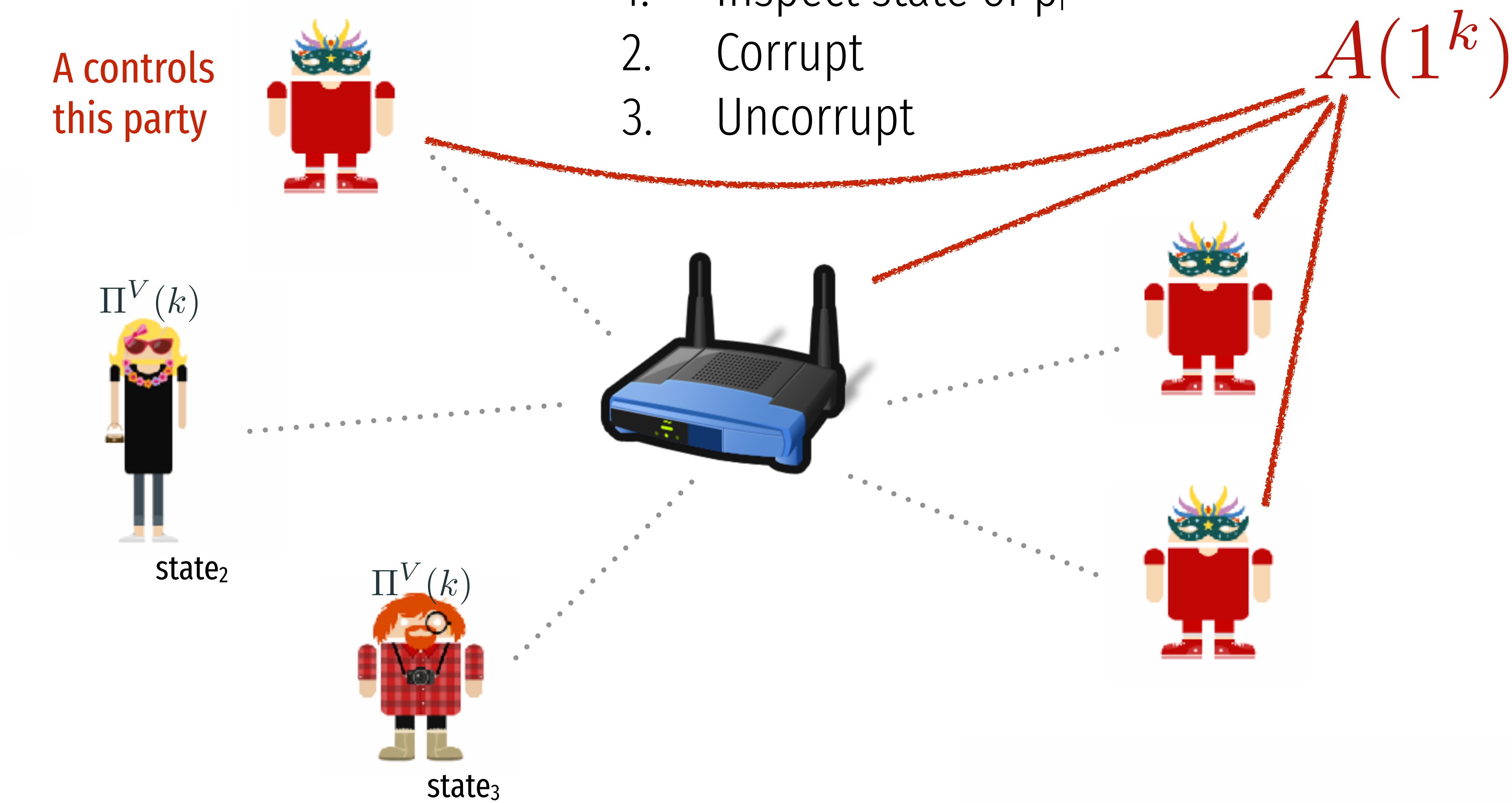


Execution

Round r

$Z(1^\kappa)$ **Environment**, initializes players, either honest or corrupt. Provides inputs to all parties.

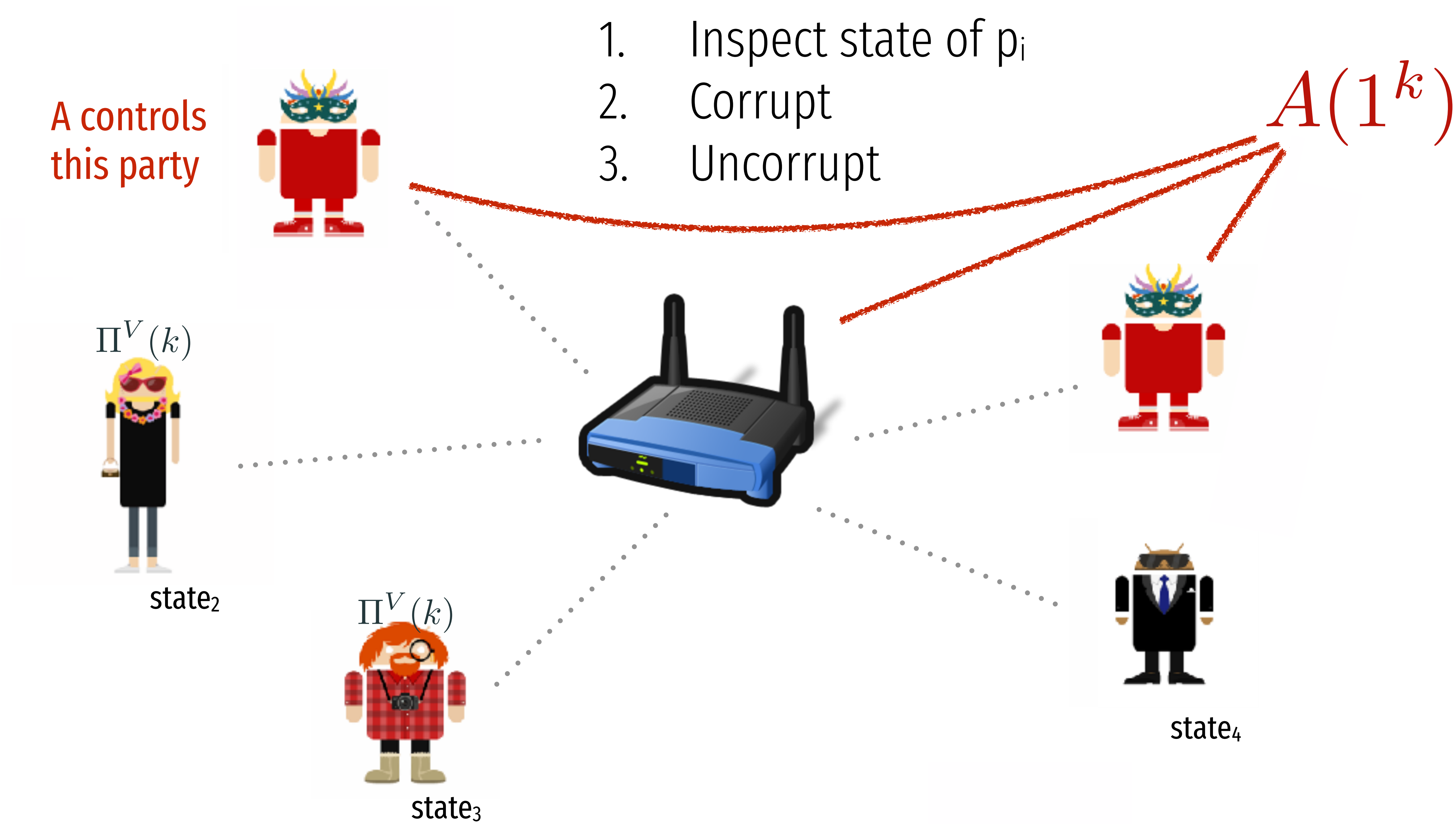
1. Inspect state of p_i
2. Corrupt
3. Uncorrupt



Execution

Round r

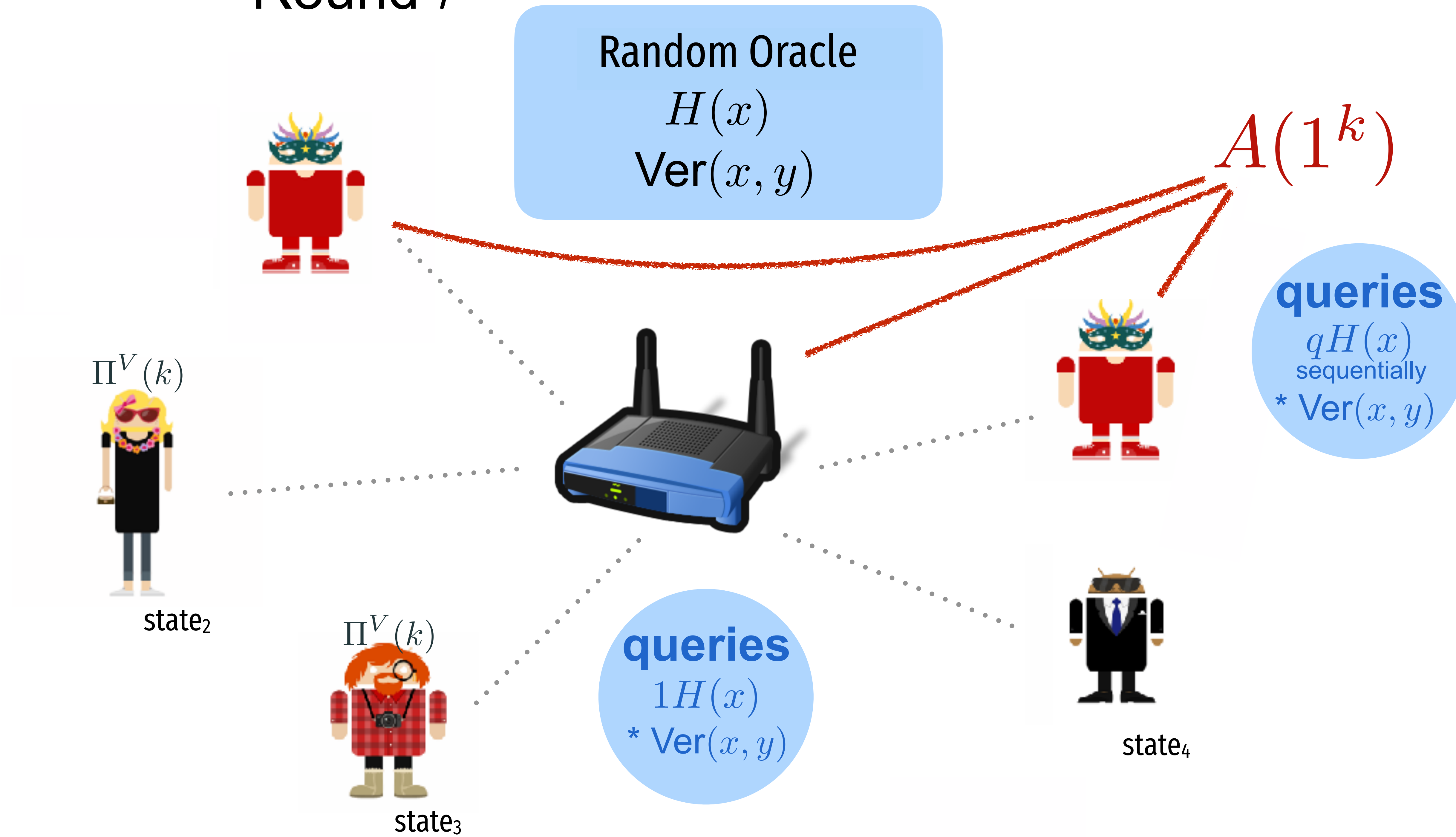
$Z(1^\kappa)$ **Environment**, initializes players, either honest or corrupt. Provides inputs to all parties.



Execution

$Z(1^\kappa)$ **Environment**, initializes players, either honest or corrupt. Provides inputs to all parties.

Round r



Random Oracle

$H(x) : \{0,1\}^* \rightarrow \{0,1\}^k$

"Best way to mine a block is to hash-and-check. Only 1 hash per round."

$H.Ver(x,y): \{0,1\}^* \times \{0,1\}^k \rightarrow \{0,1\}$

verify a hash

"Players can verify blocks without having to use their hash query."

Adversarial Model

- Dynamic control of who to corrupt/uncorrupt
- Full view of all honest states
- q sequential queries to $H(x)$ at every round
- Reorder receipt of honest blocks
- Delay receipt of honest blocks up to some amount
- Withhold adversarial blocks

MAIN PARAMETERS (with a round being the smallest unit of time)

Δ the network delay bound

MAIN PARAMETERS (with a round being the smallest unit of time)

Δ	the network delay bound
$p = \frac{1}{c \cdot n\Delta}$	the mining hardness is expressed in terms of parameter c, roughly the expected number of network delays before some block is mined

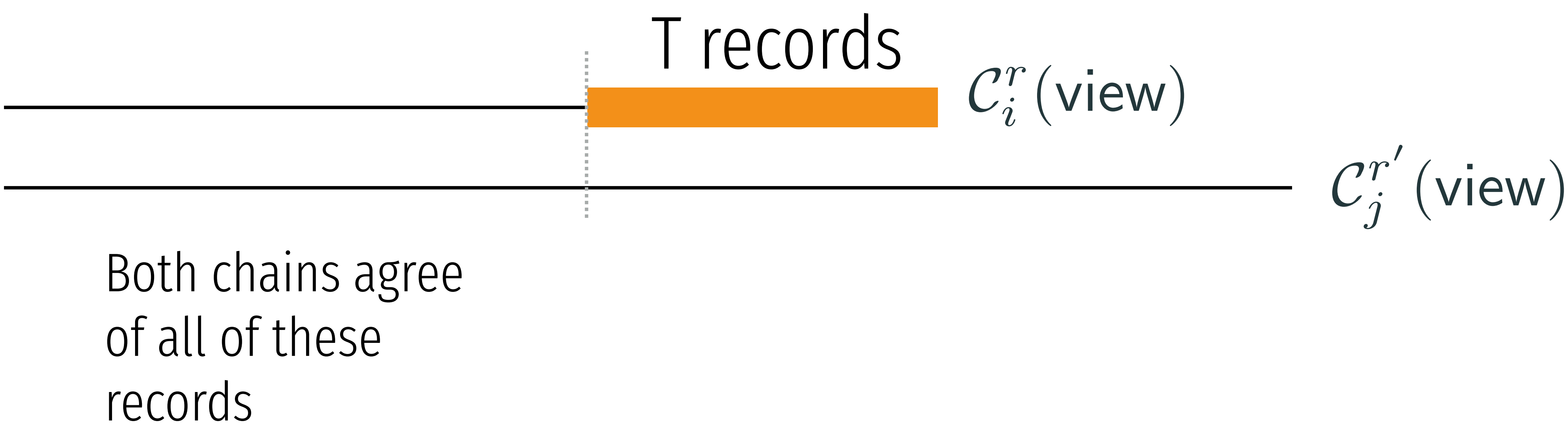
MAIN PARAMETERS (with a round being the smallest unit of time)

Δ	the network delay bound
$p = \frac{1}{c \cdot n\Delta}$	the mining hardness is expressed in terms of parameter c, roughly the expected number of network delays before some block is mined
ρ	the adversarial fraction of parties
$\mu = 1 - \rho$	the fraction of honest parties

BLOCKCHAIN CONSISTENCY

Chain Consistency

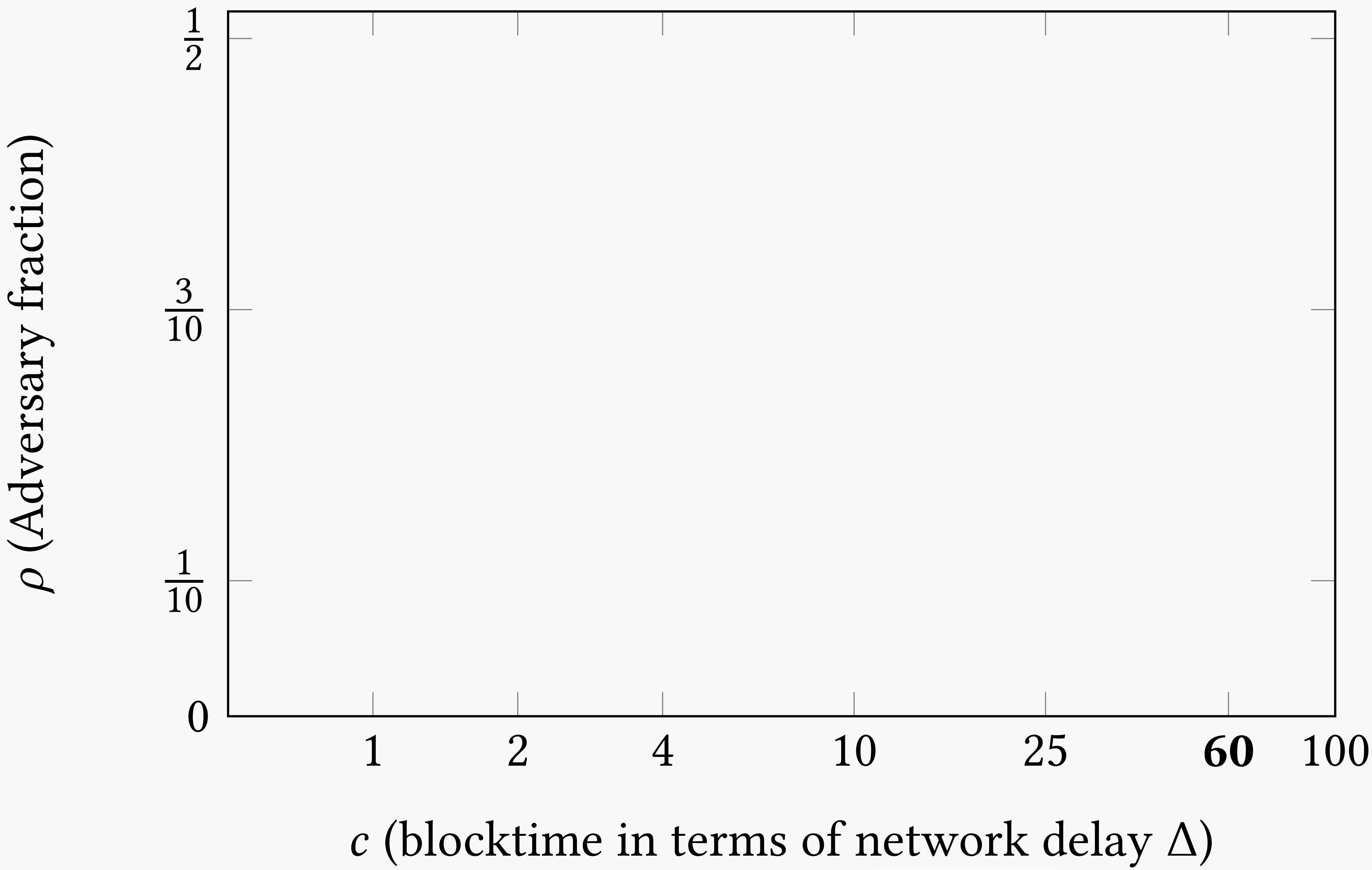
“for any two rounds r < r’,
for any two players i,j, i is honest @ r, j honest @ r’
chains of i,j agree on all but last T records of i”



[*Pass, Seeman, shelat 2016], [Common prefix: Garay, Kiayias, Leonardos 2016, 2017*]

Previous Work on Nakamoto Consistency

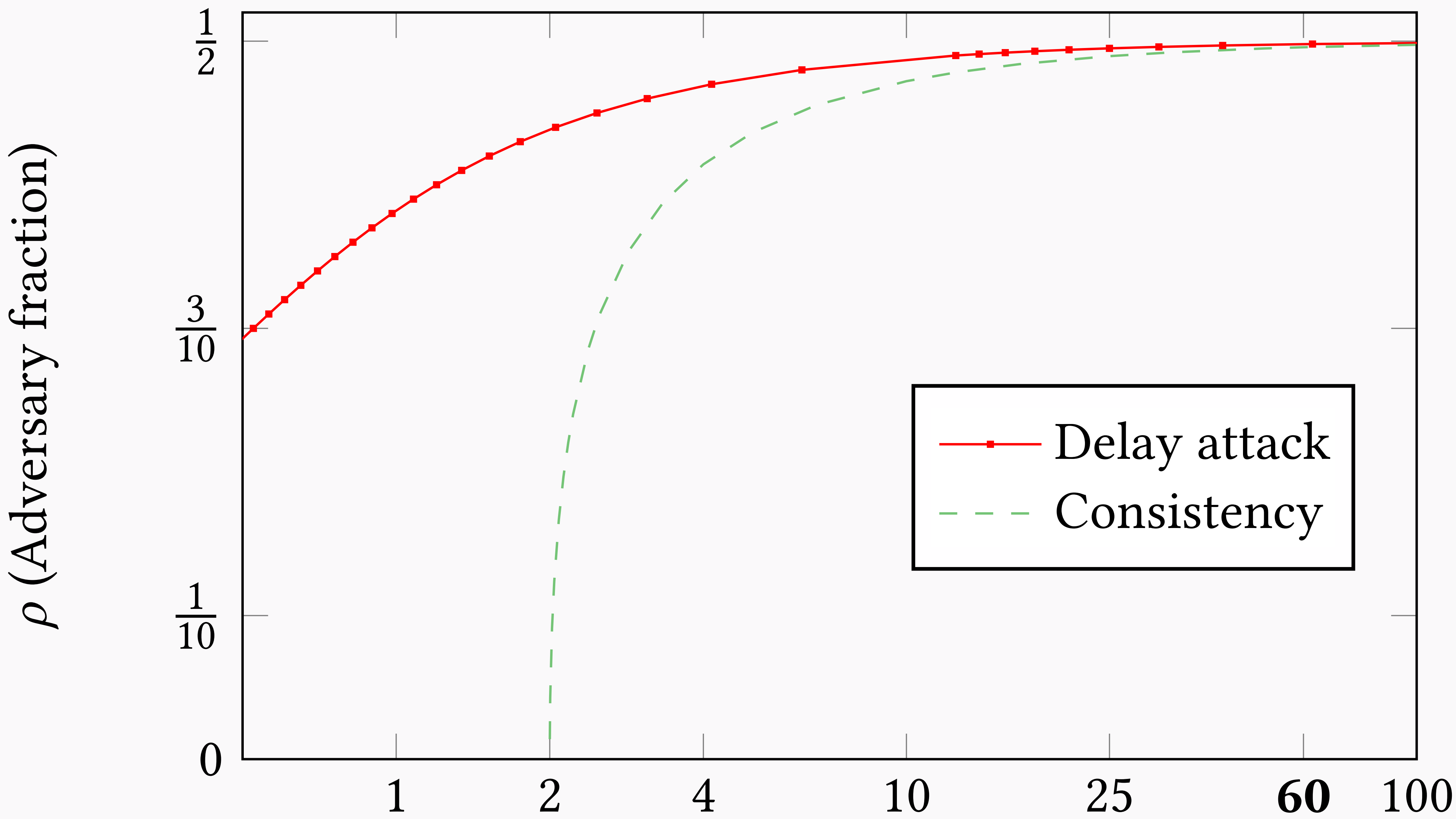
[Pass, Seeman and shelat 2016]



Recall mining hardness is $p = \frac{1}{c \cdot n\Delta}$

Previous Work on Nakamoto Consistency

[Pass, Seeman and shelat 2016]

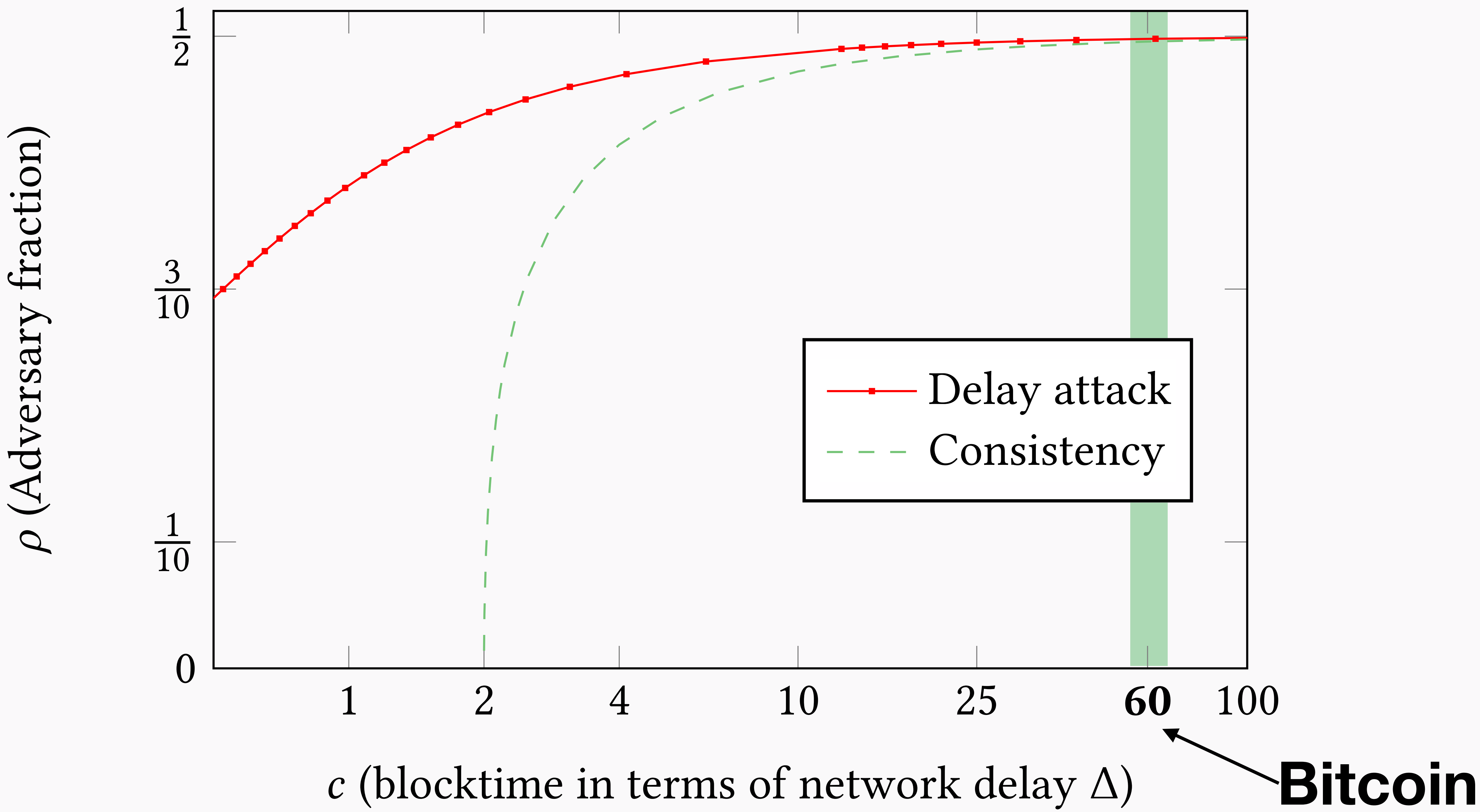


c (blocktime in terms of network delay Δ)

Recall mining hardness is $p = \frac{1}{c \cdot n\Delta}$

Previous Work on Nakamoto Consistency

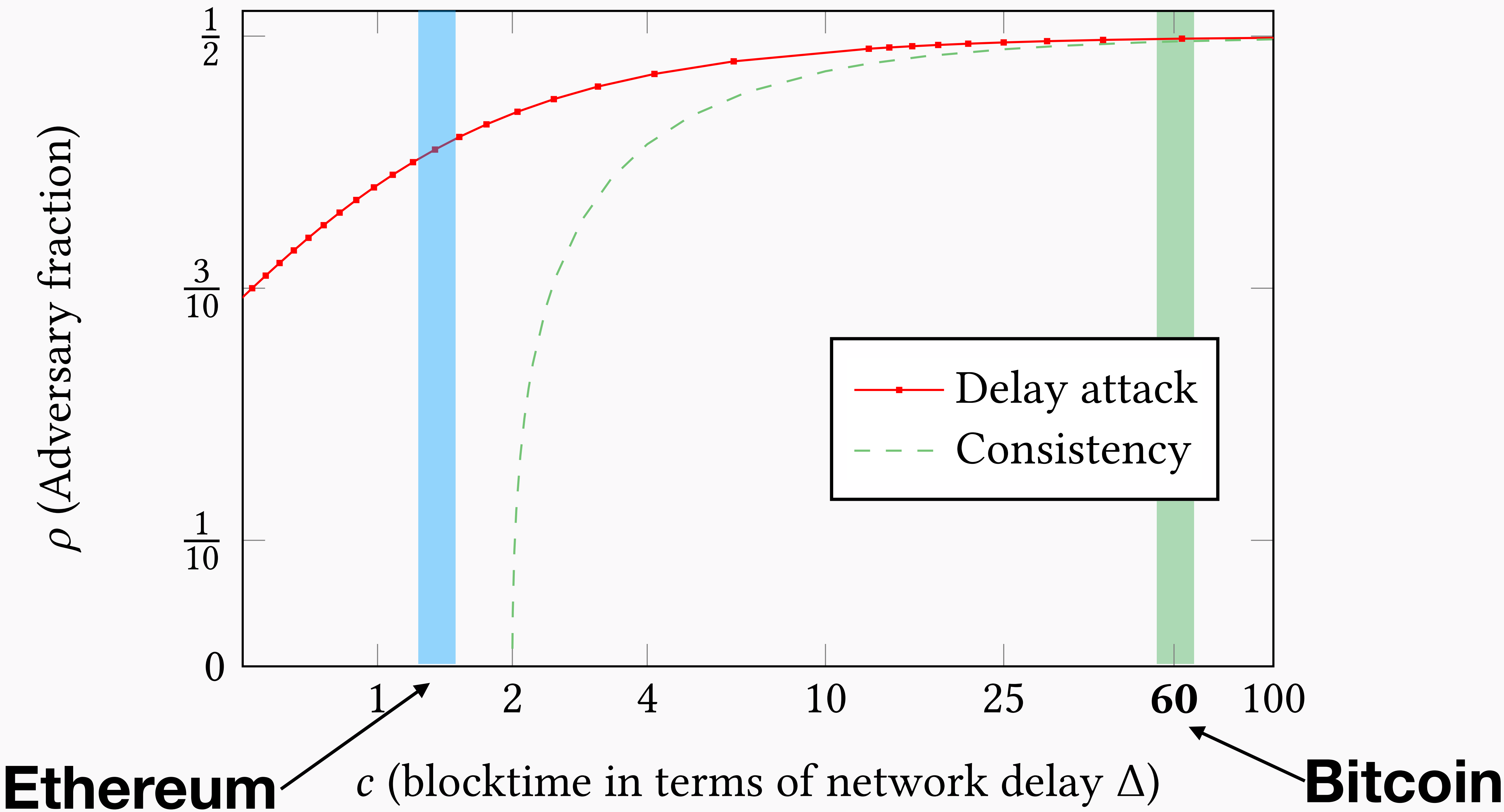
[Pass, Seeman and shelat 2016]



Recall mining hardness is $p = \frac{1}{c \cdot n\Delta}$

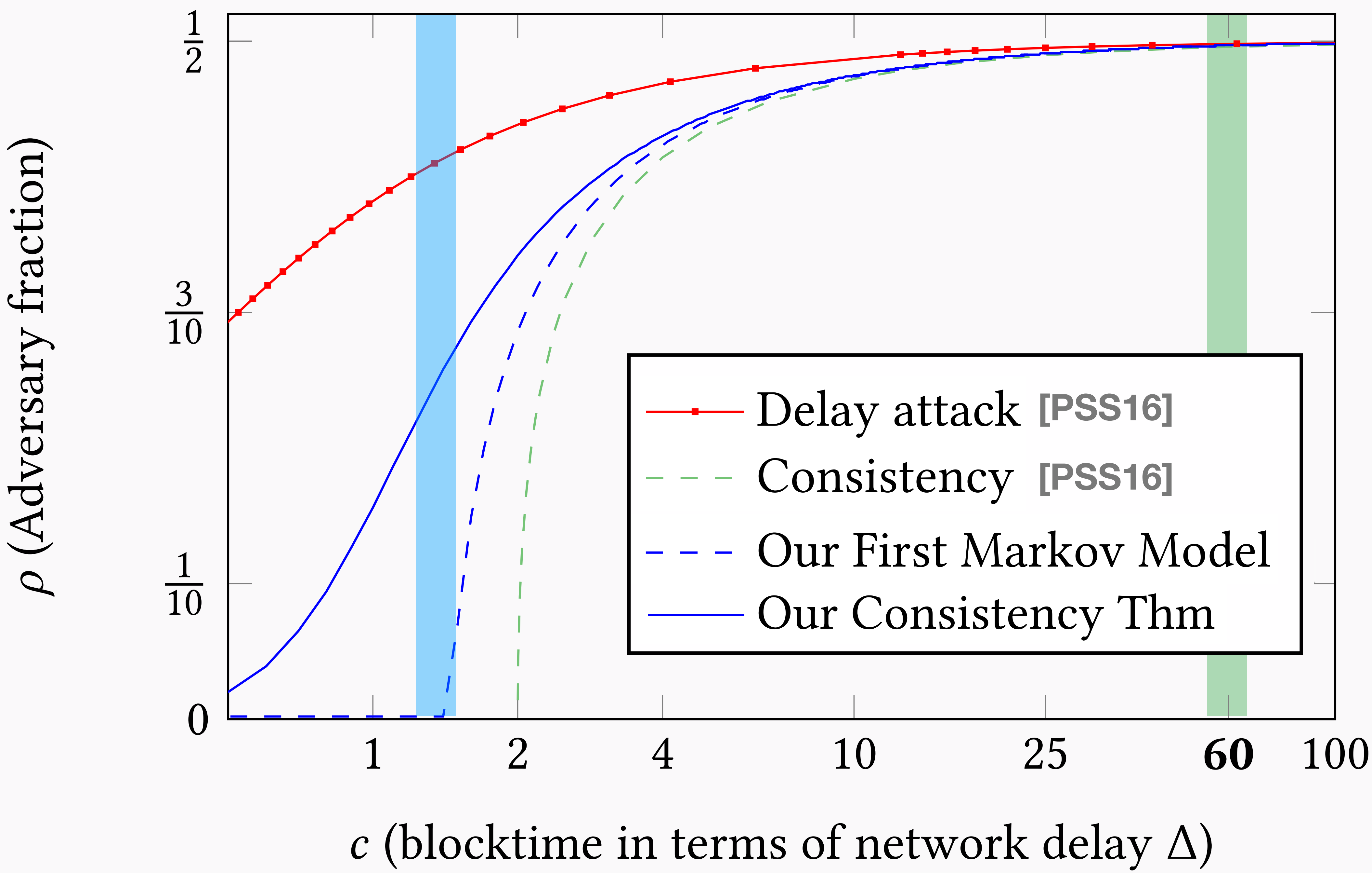
Previous Work on Nakamoto Consistency

[Pass, Seeman and shelat 2016]



Recall mining hardness is $p = \frac{1}{c \cdot n\Delta}$

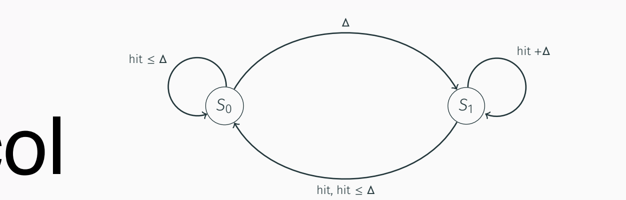
Our Improved Analysis of Nakamoto Consistency



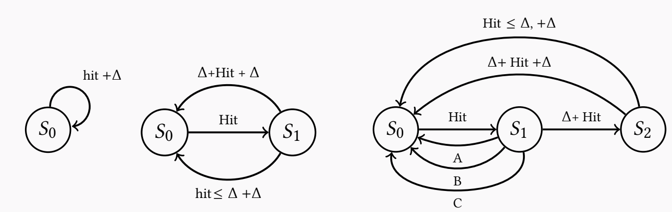
Recall mining hardness is $p = \frac{1}{c \cdot n\Delta}$

Summary of Main Results

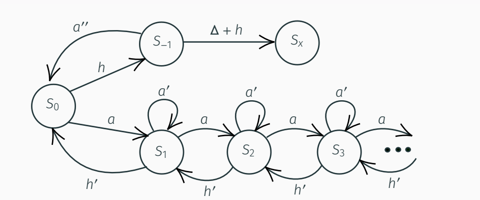
- A Markov-based method for analyzing consistency
- Better consistency bound for Nakamoto Protocol



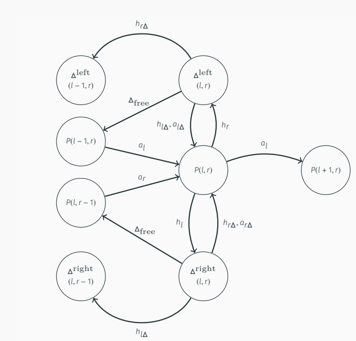
- Analysis of a family of Delay Attacks



- Analysis of *confirmation time* for transactions



- Analysis of consistency for Cliquechain and GHOST
- Balancing attack for GHOST

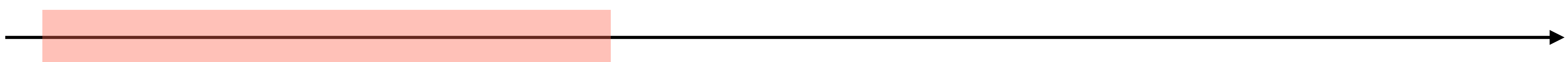


Roadmap

1. How to analyze consistency
2. Our analysis on Nakamoto consistency
3. An attack on Nakamoto consistency
4. Cliquechain consistency and attack
5. GHOST consistency and attack

HOW TO ANALYZE CONSISTENCY

Why does it work?



Period when nobody succeeds in mining. Everyone has same blockchain with **block A** at the top.

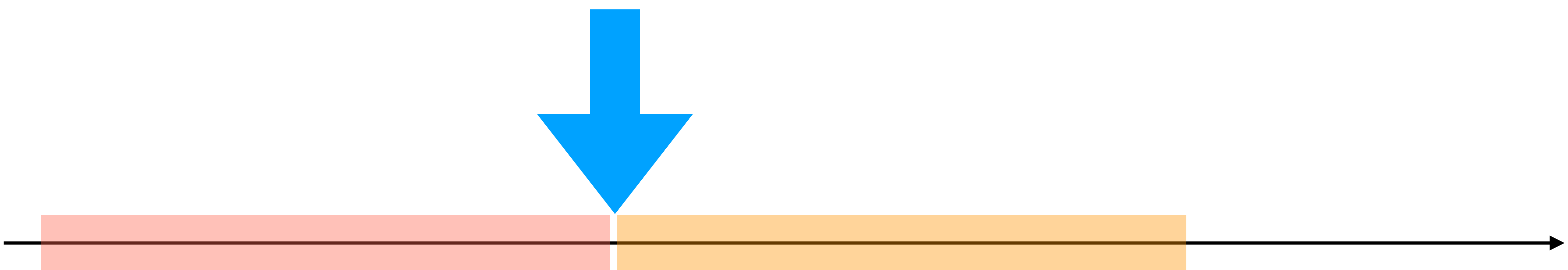
Why does it work?

Eureka! BOB finds a block **B** & broadcasts it.

Period when nobody
succeeds in mining.
Everyone has same
blockchain with
block A at the top.

Why does it work?

Eureka! BOB finds a block **B** & broadcasts it.

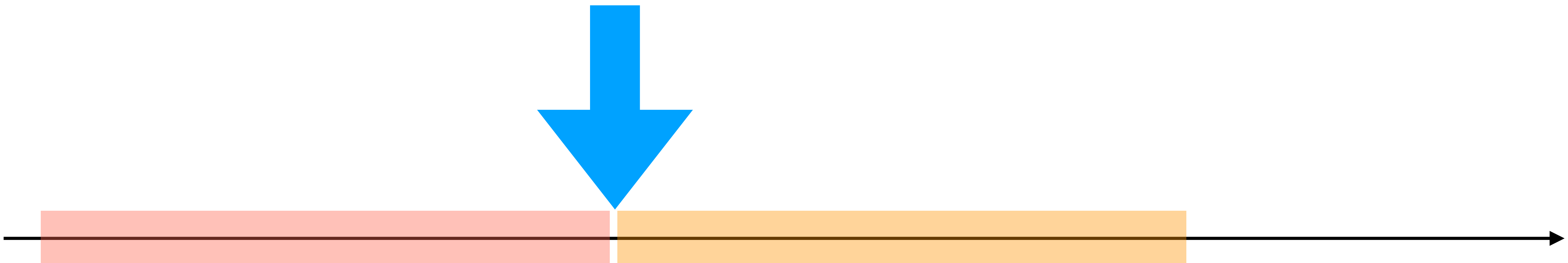


Period when nobody succeeds in mining. Everyone has same blockchain with **block A** at the top.

Block **B** is being transmitted over the network to all other miners.

Why does it work?

Eureka! BOB finds a block **B** & broadcasts it.

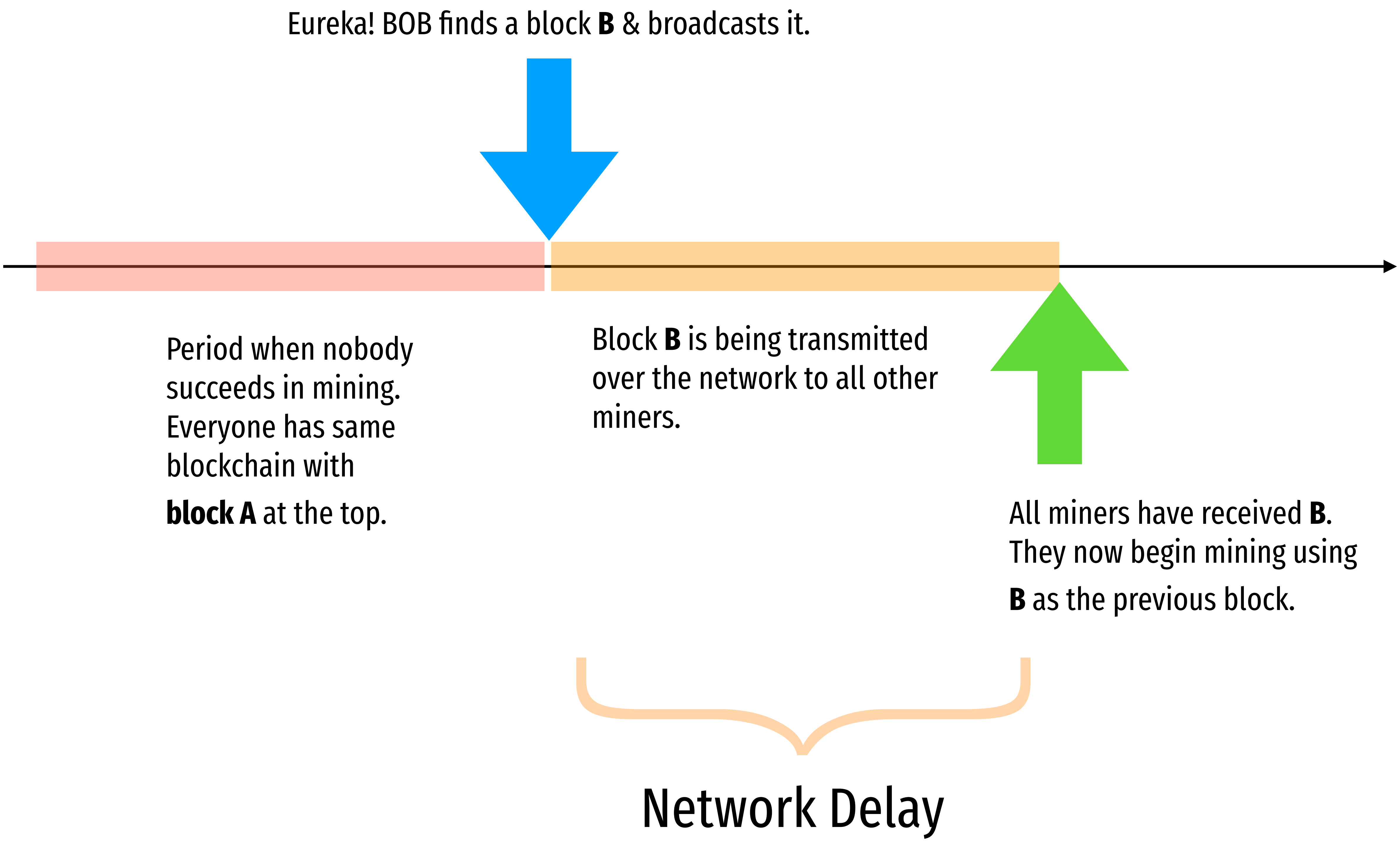


Period when nobody succeeds in mining. Everyone has same blockchain with **block A** at the top.

Block **B** is being transmitted over the network to all other miners.

Network Delay

Why does it work?

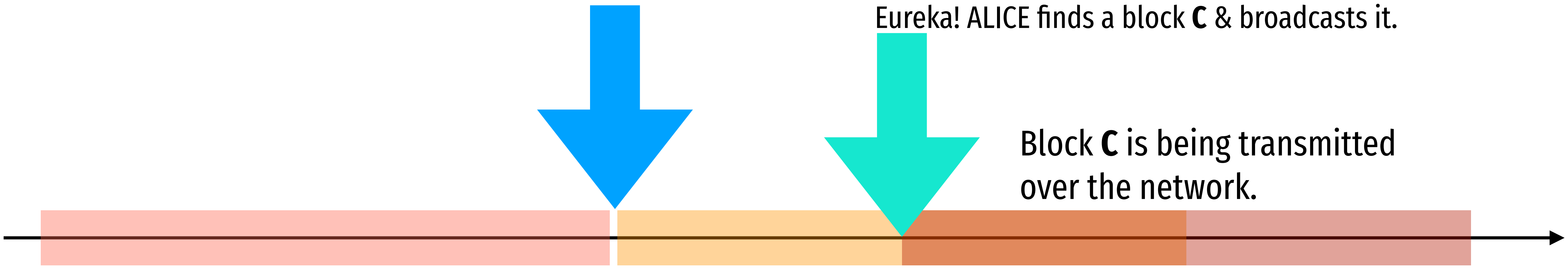


Why does it work?

Eureka! BOB finds a block **B** & broadcasts it.

Eureka! ALICE finds a block **C** & broadcasts it.

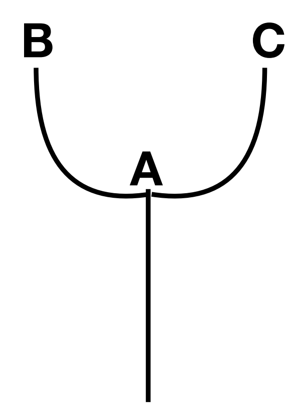
Block **C** is being transmitted over the network.



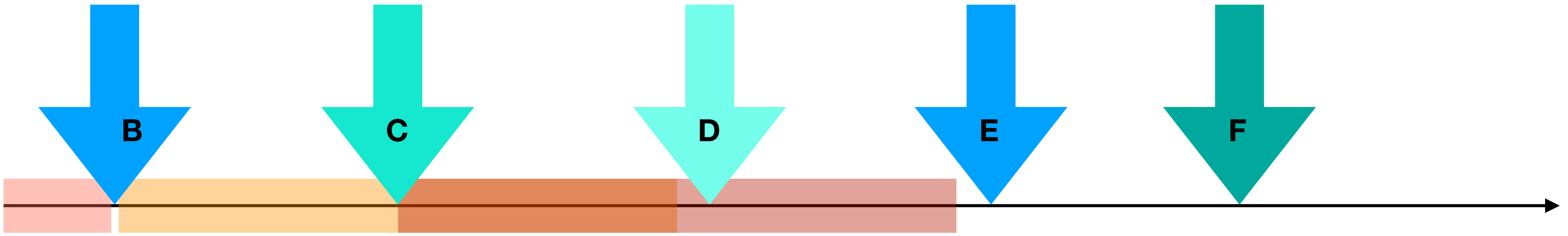
Period when nobody succeeds in mining. Everyone has same blockchain with **block A** at the top.

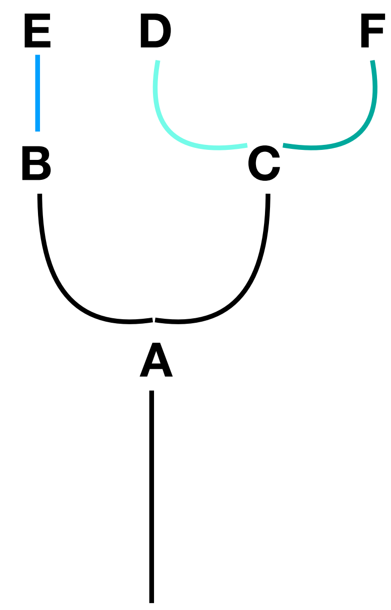
Block **B** is being transmitted over the network to all other miners.

Some miners received **B** first, some received **C** first. Network is trying to extend both **B** and **C**.



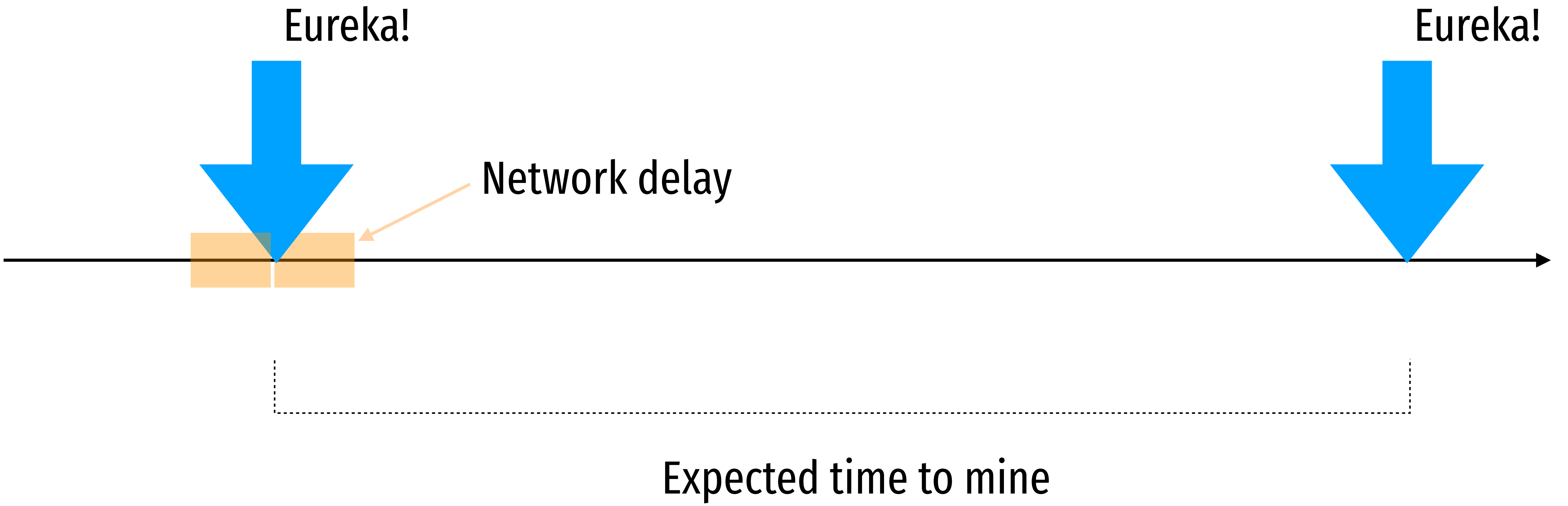
It could happen again



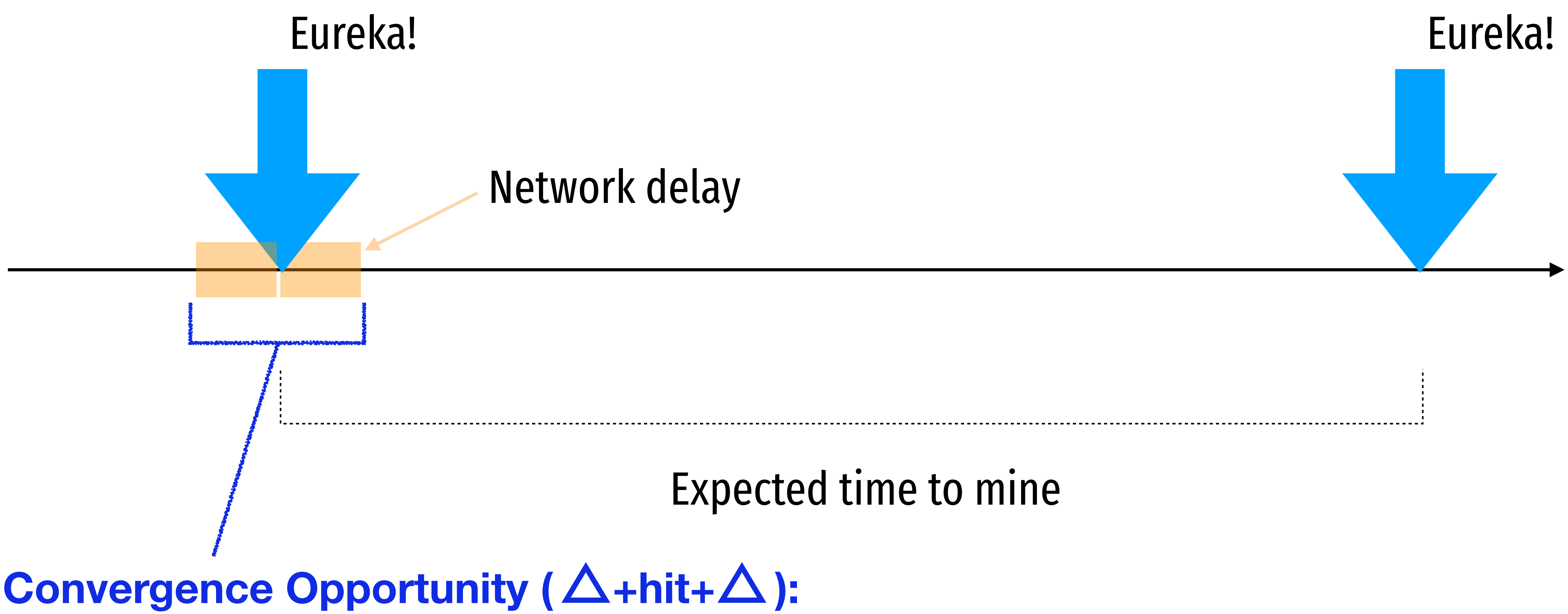


What prevents forking ad nauseum?

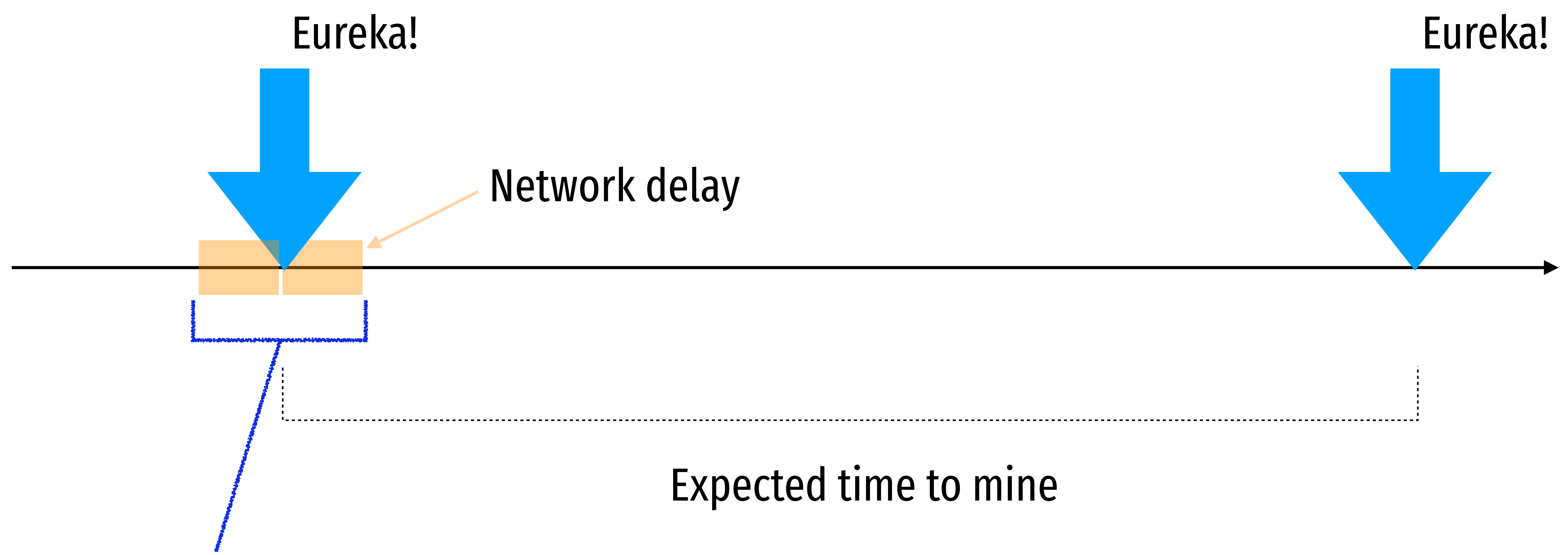
Network delay vs mining hardness



Network delay vs mining hardness



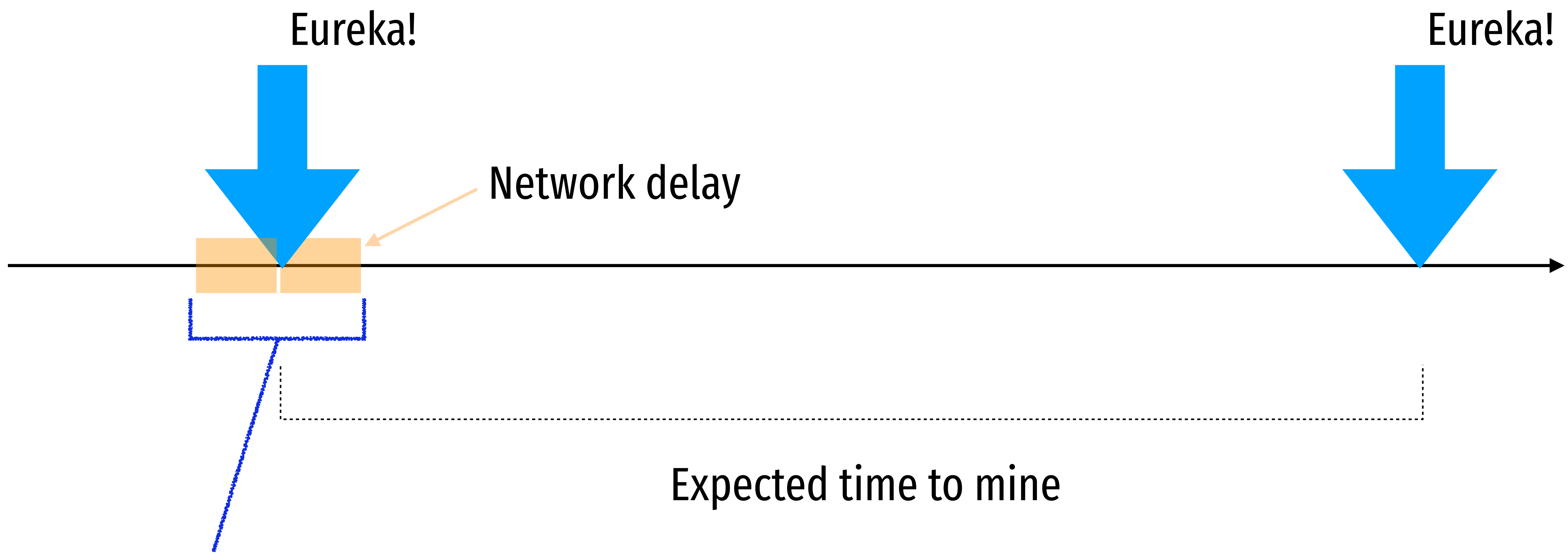
Network delay vs mining hardness



Convergence Opportunity (△+hit+△):

1. **△ rounds where no one mines** (everyone hears about all blocks)

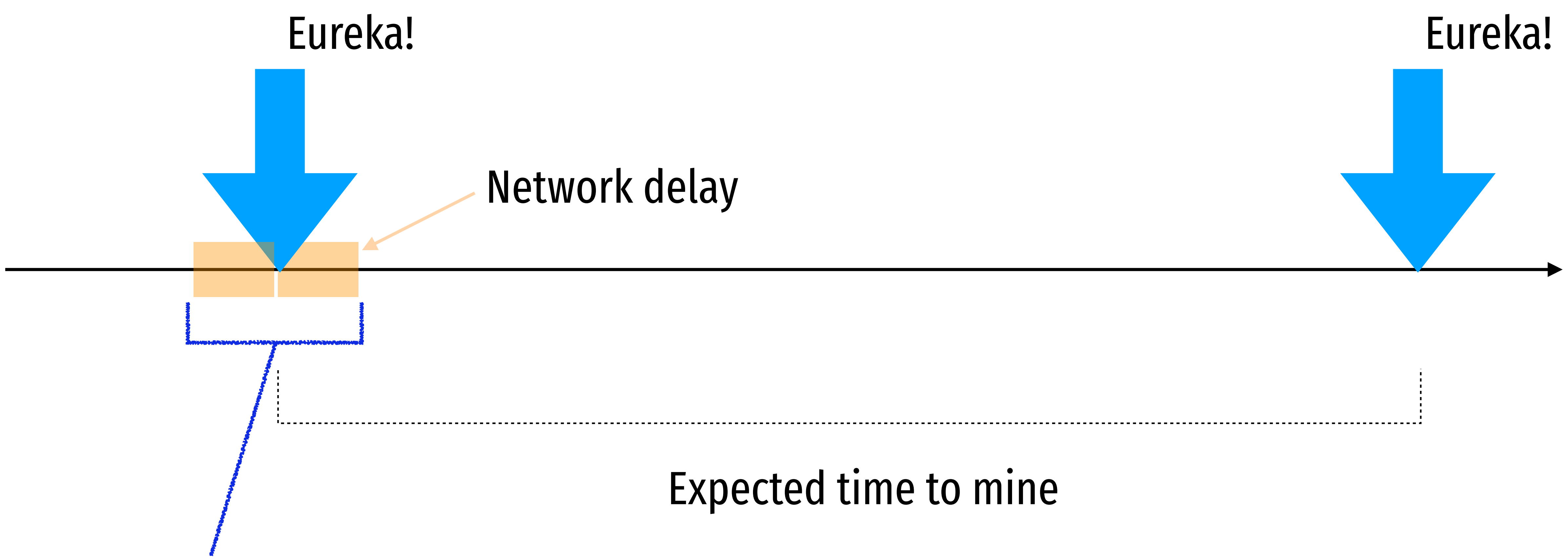
Network delay vs mining hardness



Convergence Opportunity ($\triangle$+hit+$\triangle$):

1. **$\triangle$ rounds where no one mines** (everyone hears about all blocks)
2. **One player mines a block** (single longest chain)

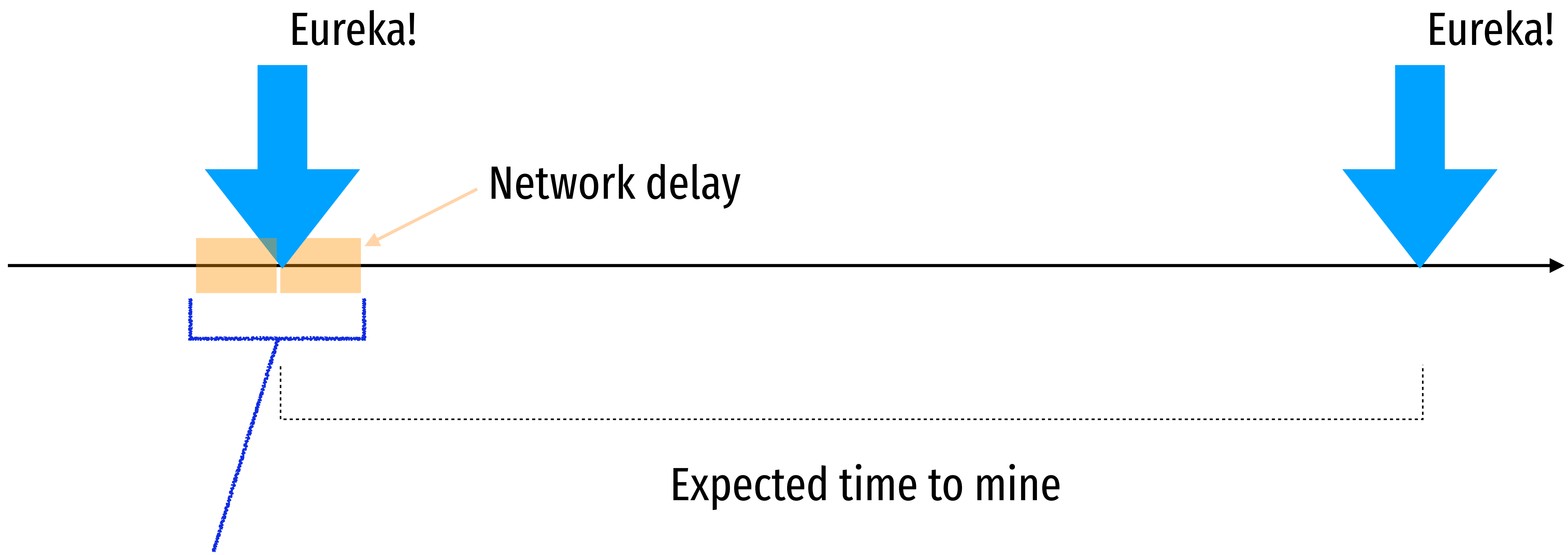
Network delay vs mining hardness



Convergence Opportunity (△+hit+△):

1. **△ rounds where no one mines** (everyone hears about all blocks)
2. **One player mines a block** (single longest chain)
3. **Another △ rounds where no one mines** (everyone hears about that block)

Network delay vs mining hardness

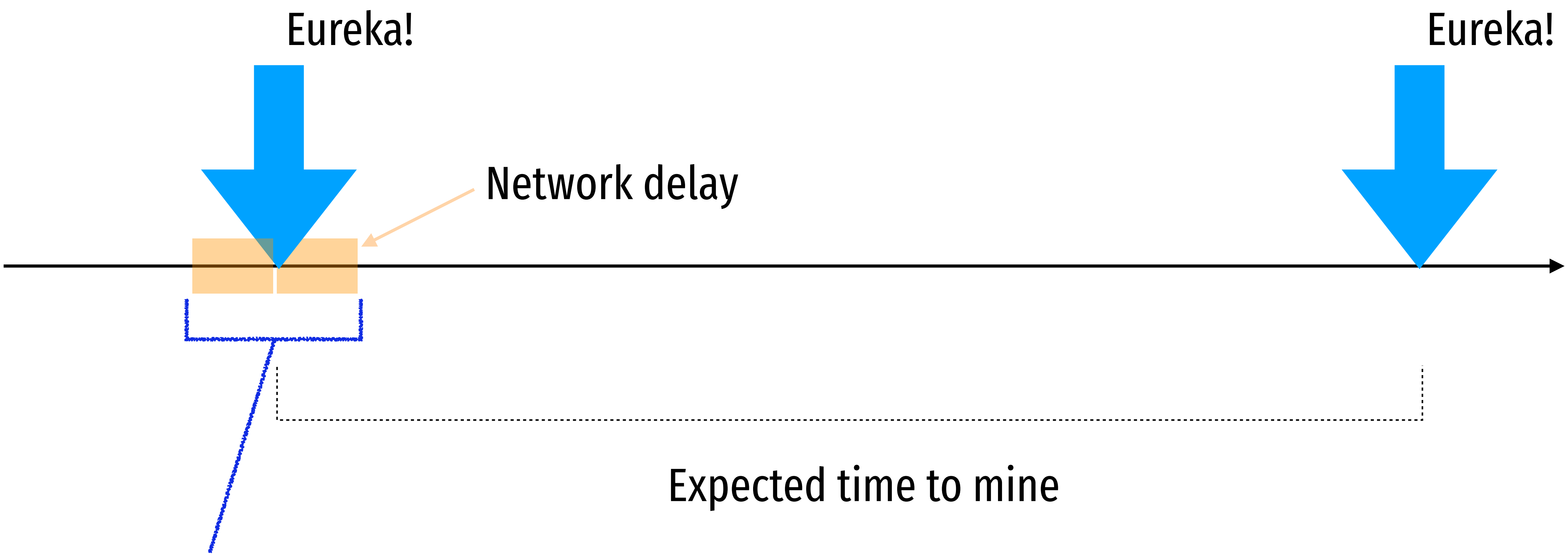


Convergence Opportunity (△+hit+△):

1. **△ rounds where no one mines** (everyone hears about all blocks)
2. **One player mines a block** (single longest chain)
3. **Another △ rounds where no one mines** (everyone hears about that block)

—> everyone agrees (2.) is the longest chain

Network delay vs mining hardness



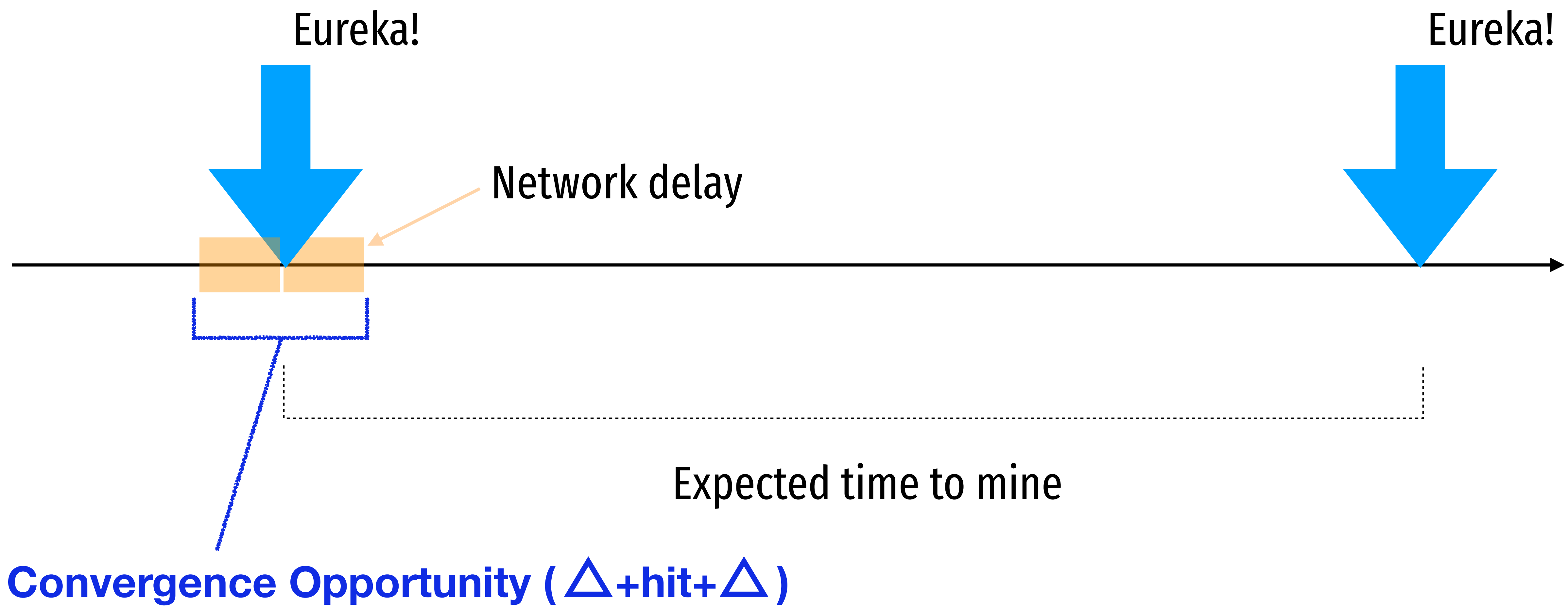
Convergence Opportunity (△+hit+△):

1. **△ rounds where no honest player mines** (everyone hears about all blocks)
2. **One honest player mines a block** (single longest chain)
3. **Another △ rounds where no honest player mines**

   (everyone hears about that block)

—> every honest player agrees (2.) is the longest chain

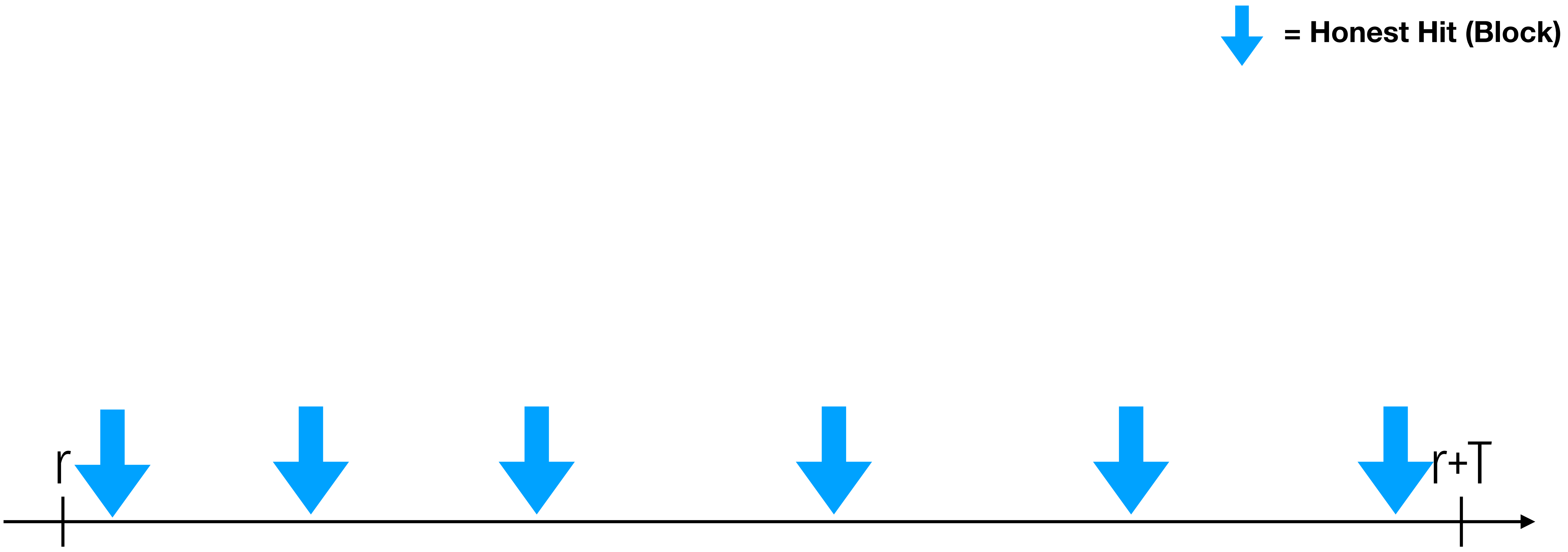
Network delay vs mining hardness

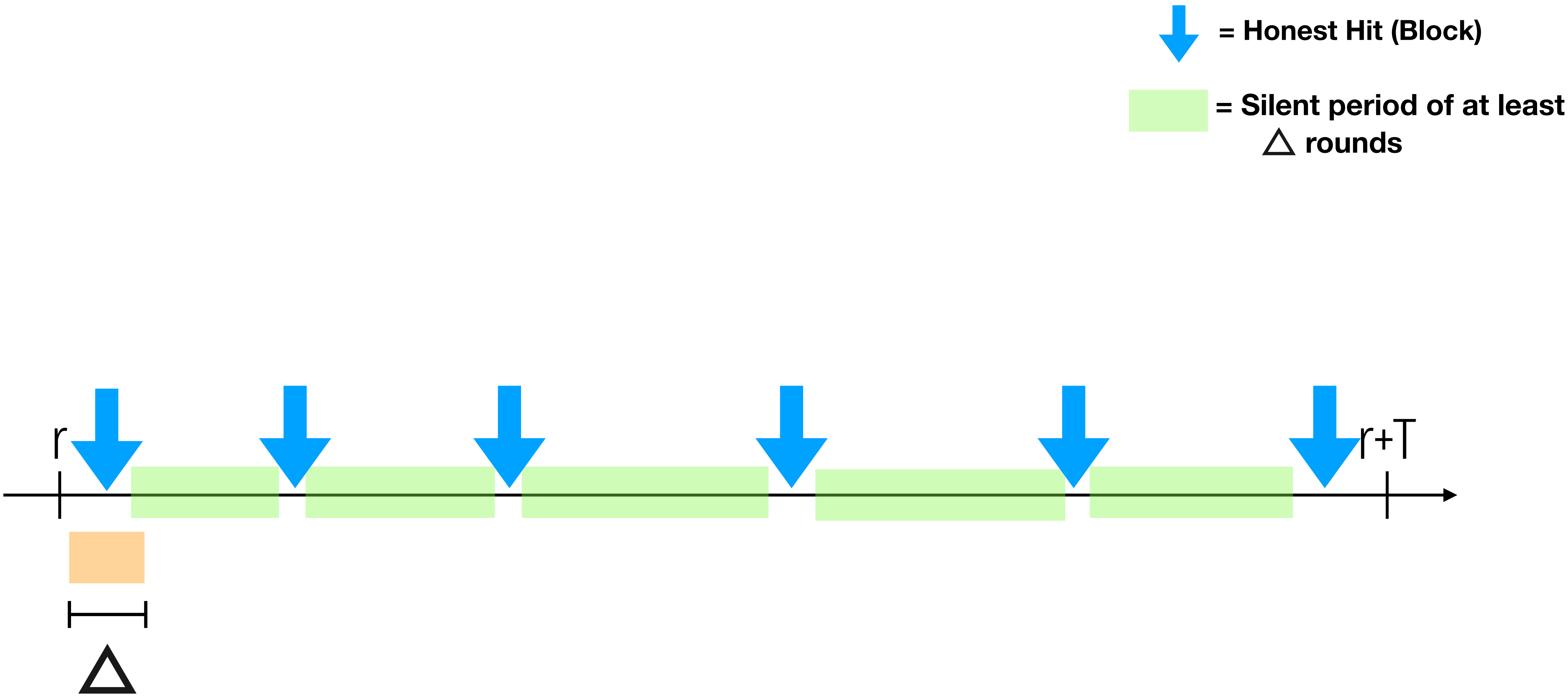


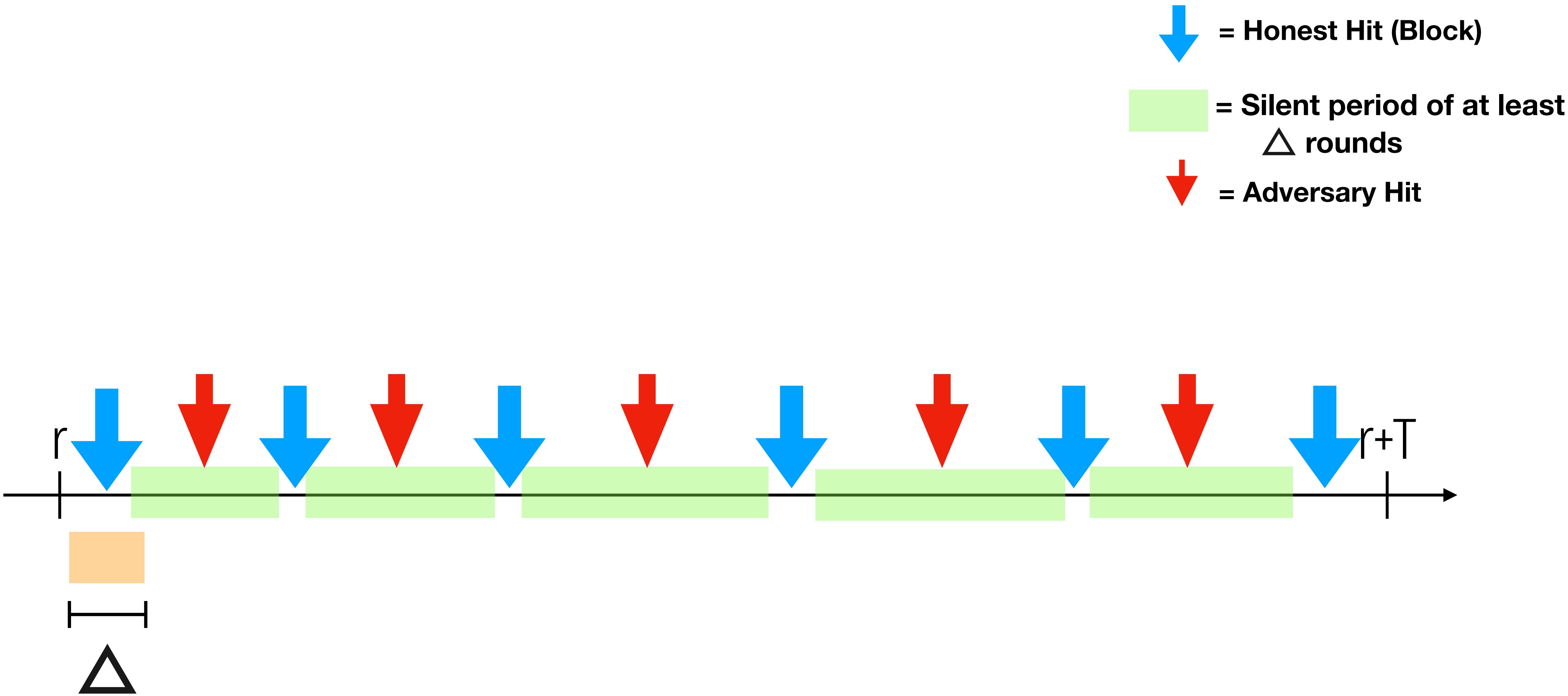
Analysis: In order to break consistency, adversary must *<u>break</u>* all COs

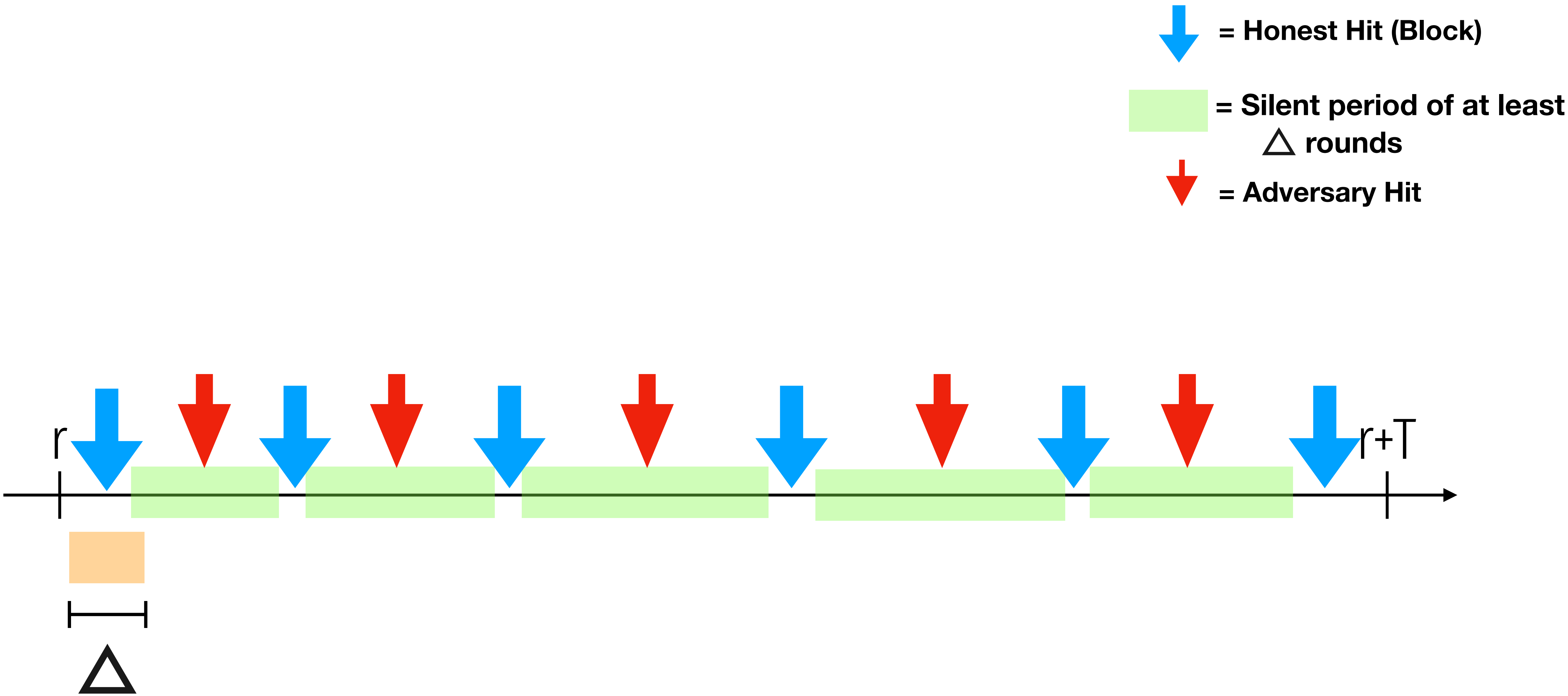
COUNTING CONVERGENCE OPPORTUNITIES


r
r+T


= Honest Hit (Block)
r
r+T


= Honest Hit (Block)
= Silent period of at least △ rounds
r
r+T
△

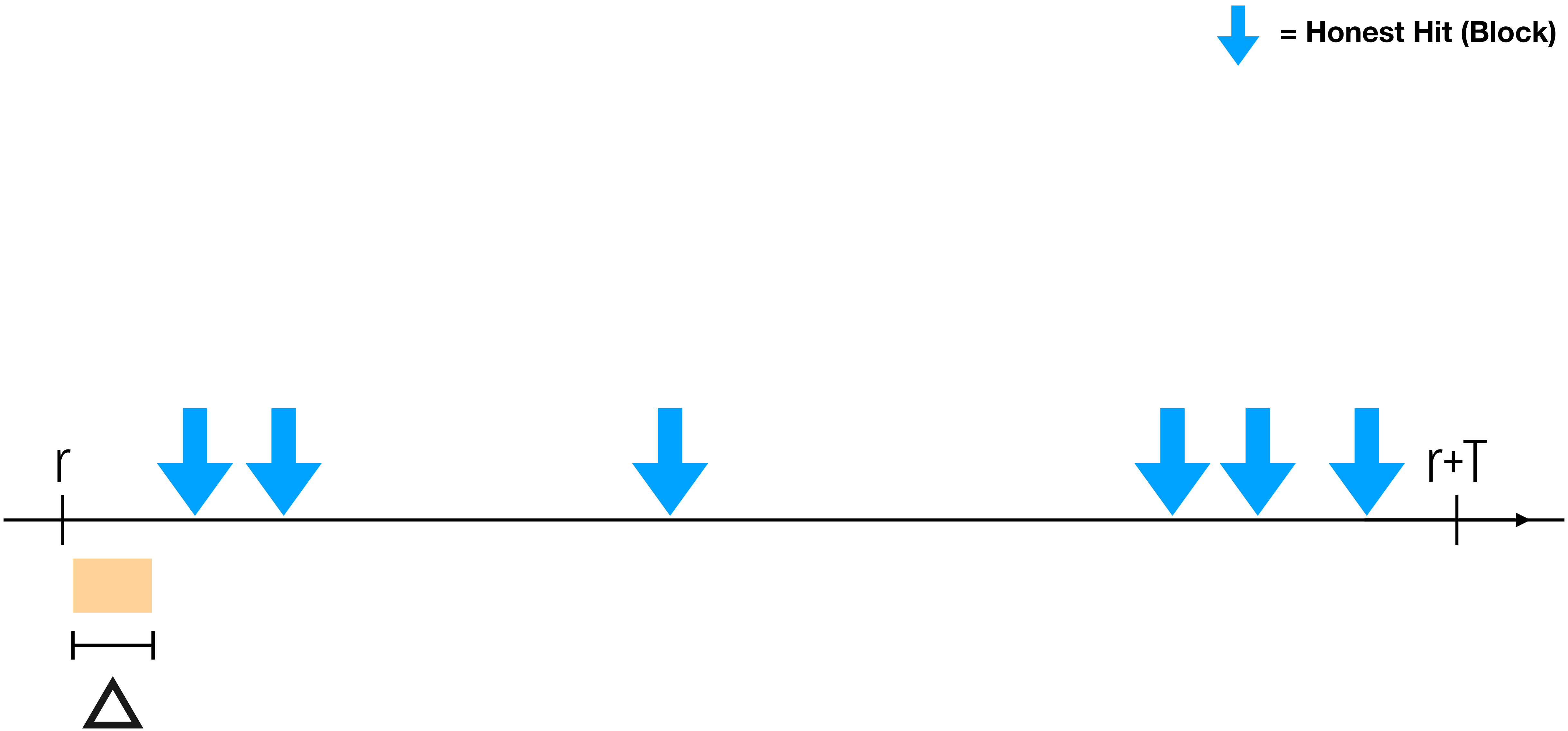

= Honest Hit (Block)
= Silent period of at least △ rounds
= Adversary Hit
r
r+T
△



Analysis: In order to break consistency, adversary must *break* all COs

Goal:

- **Count expected number of COs**
- **Compare with expected number of blocks the adversary can mine**



Analysis: In order to break consistency, adversary must *__break__* all COs

Goal:

- **Count expected number of COs**
- **Compare with expected number of blocks the adversary can mine**

Roadmap

1. How to analyze consistency
2. **Our analysis on Nakamoto consistency**
3. An attack on Nakamoto consistency
4. Cliquechain consistency and attack
5. GHOST consistency and attack

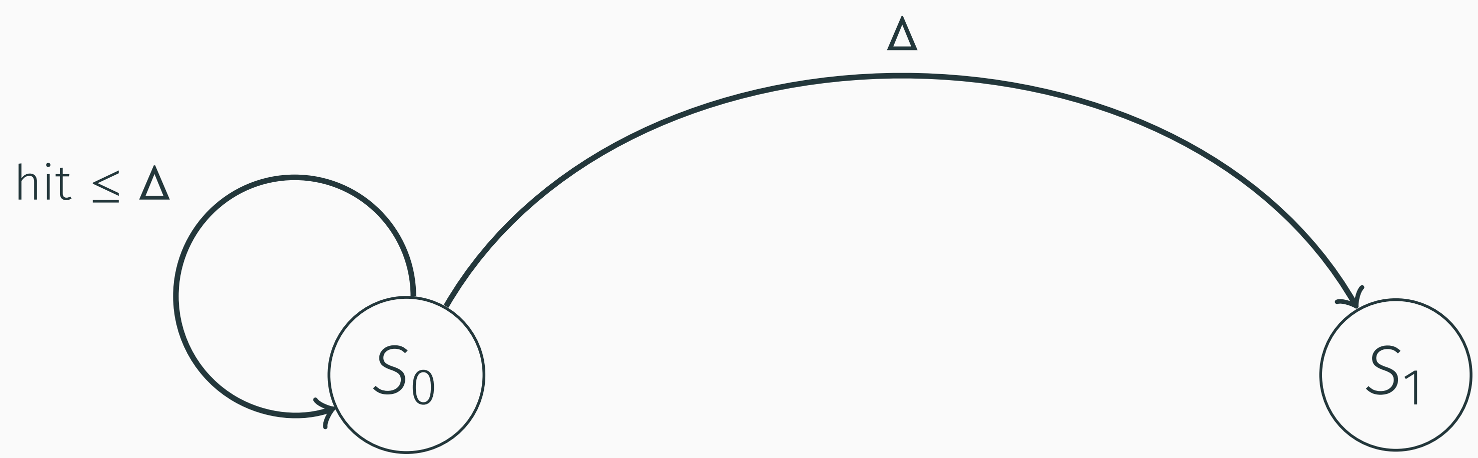
SIMPLE MARKOV MODEL OF CONVERGENCE OPPORTUNITY

A simple Markov model with two states

SIMPLE MARKOV MODEL OF CONVERGENCE OPPORTUNITY

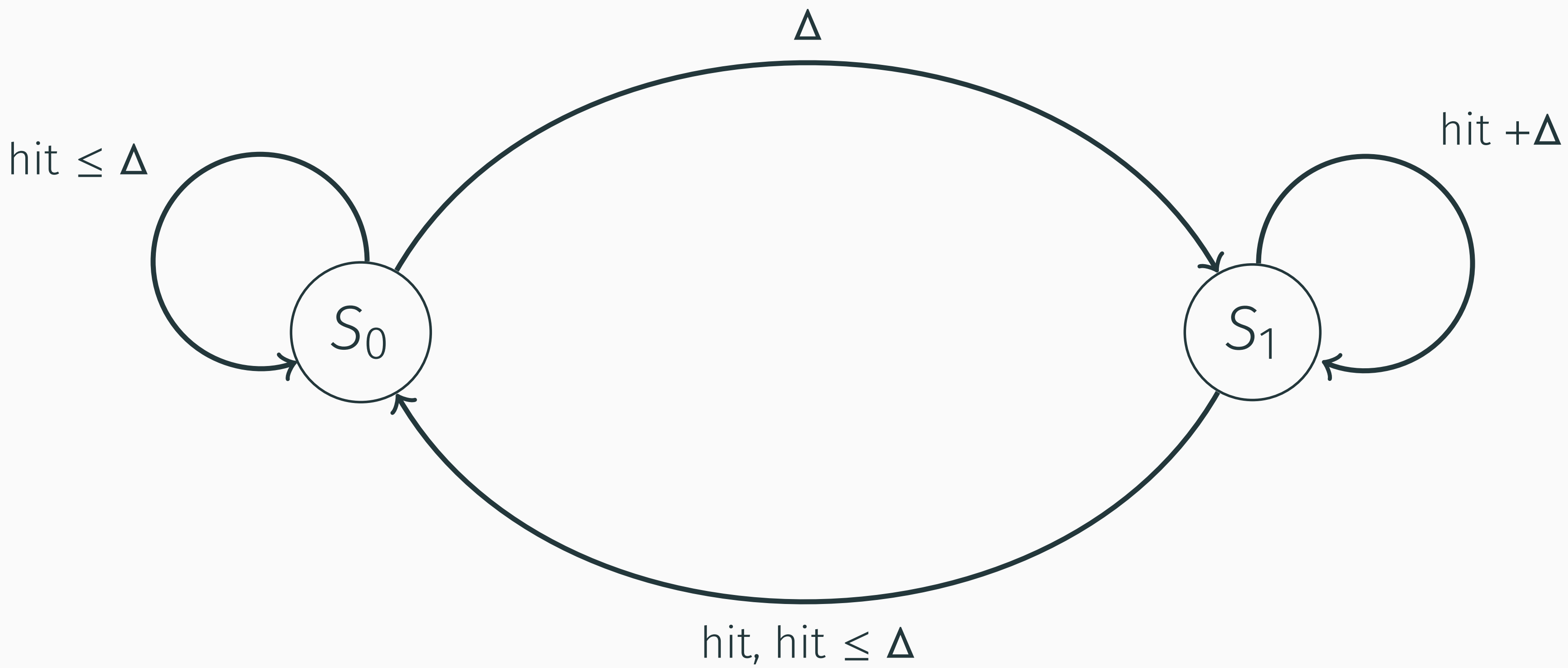
"Messy" state S_0 and back if hit within Δ

SIMPLE MARKOV MODEL OF CONVERGENCE OPPORTUNITY



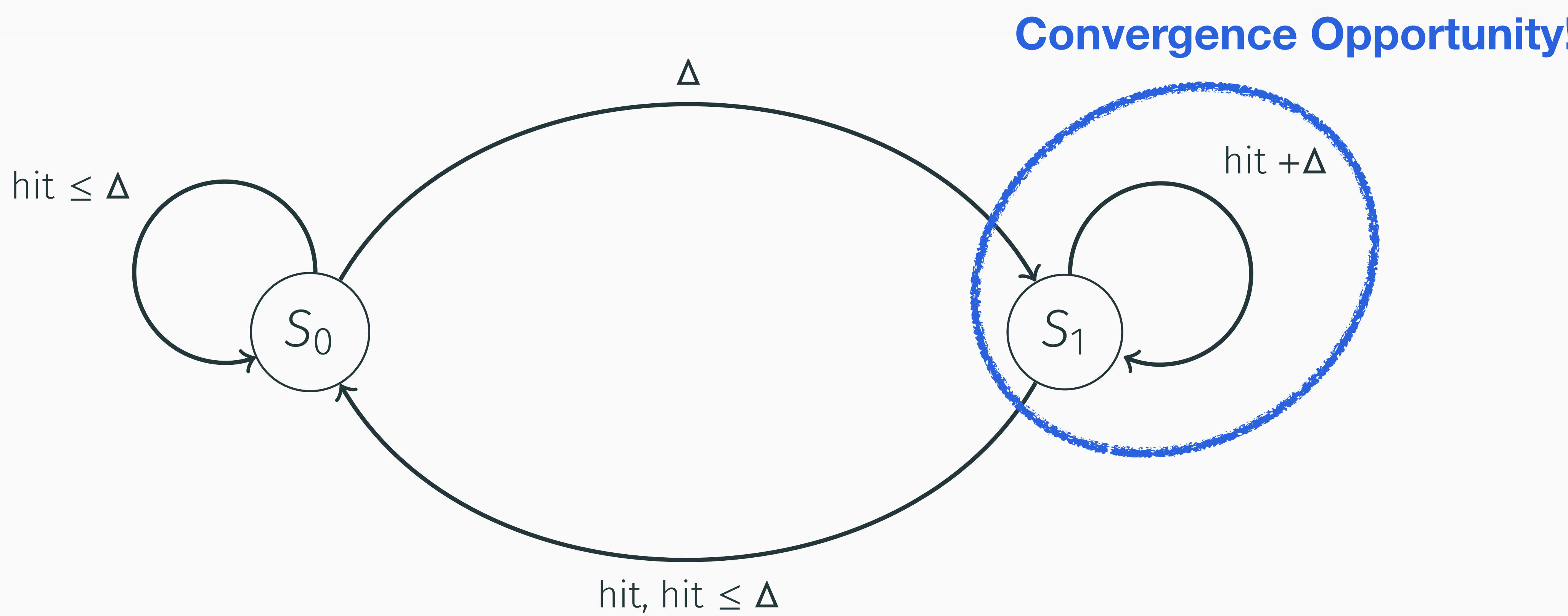
Transition to S_1 after quiet period

SIMPLE MARKOV MODEL OF CONVERGENCE OPPORTUNITY



Complete model

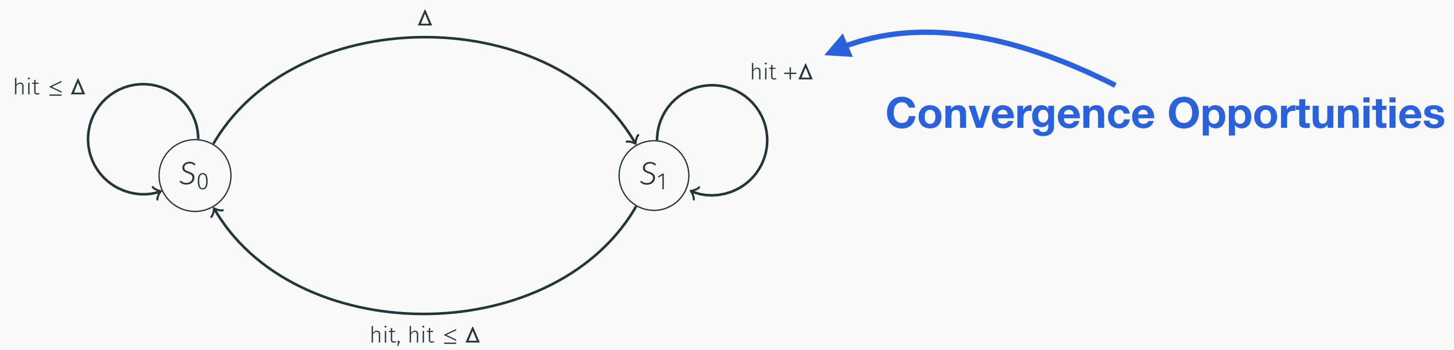
SIMPLE MARKOV MODEL OF CONVERGENCE OPPORTUNITY



Complete model

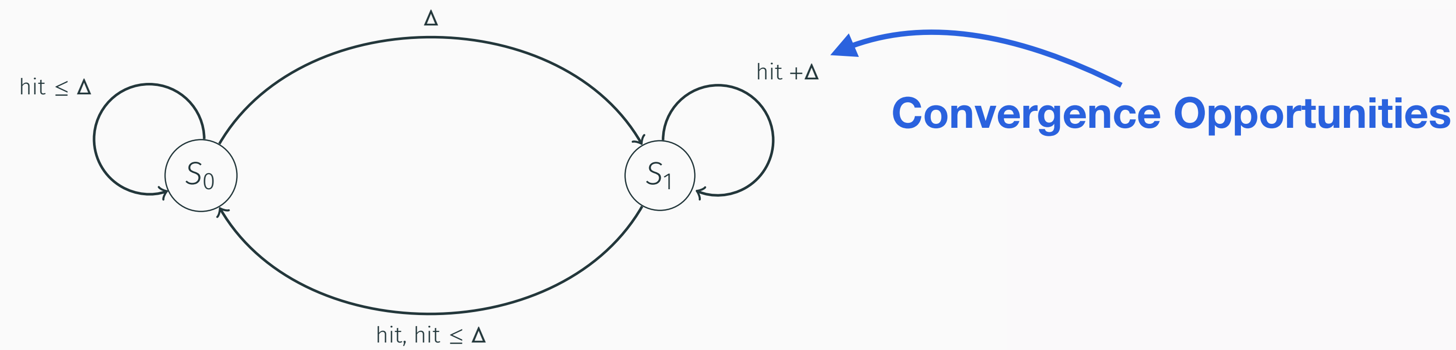
OUR PROGRAM

1. Define a Markov model with states & events of interest



OUR PROGRAM

1. Define a Markov model with states & events of interest



2. Compute stationary distribution for states and edges

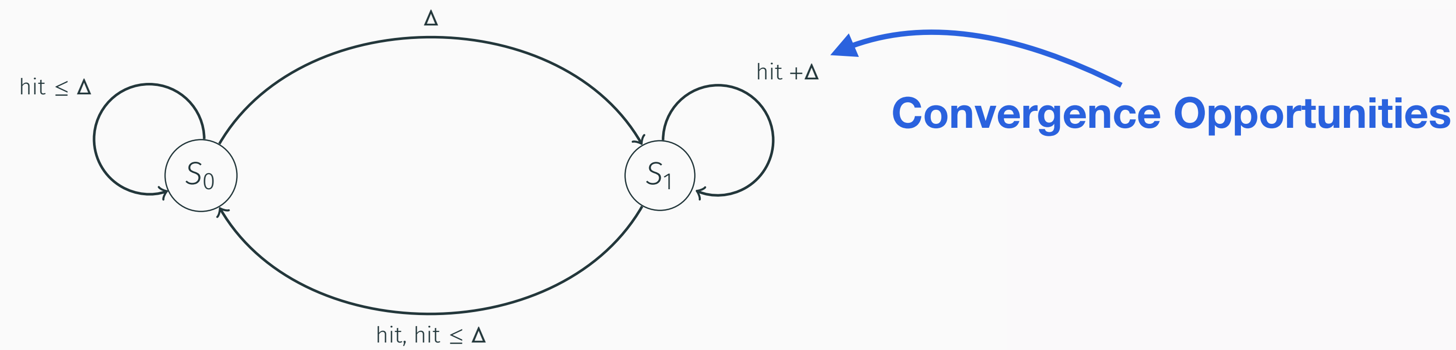
$$P_\Delta = (1 - \mu p)^\Delta$$

$$\Pr[e_{00}] = \Pr[e_{10}] = 1 - P_\Delta$$
$$\Pr[e_{01}] = \Pr[e_{11}] = P_\Delta$$

$$\pi_0 = \Pr[S_0] = (1 - P_\Delta)\pi_0 + (1 - P_\Delta)\pi_1$$
$$\pi_1 = \Pr[S_1] = P_\Delta \pi_1 + P_\Delta \pi_0$$

OUR PROGRAM

1. Define a Markov model with states & events of interest



2. Compute stationary distribution for states and edges

$$P_\Delta = (1 - \mu p)^\Delta$$

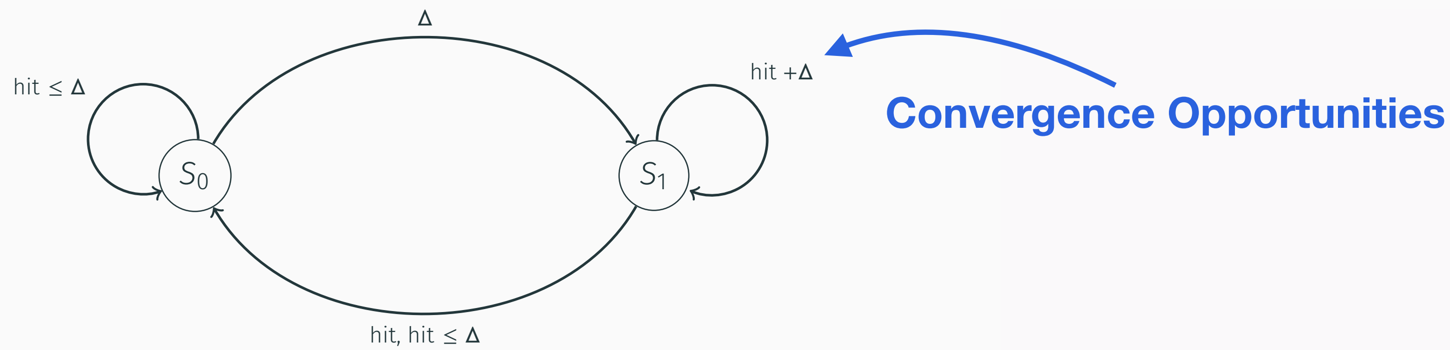
$$\Pr[e_{00}] = \Pr[e_{10}] = 1 - P_\Delta$$
$$\Pr[e_{01}] = \Pr[e_{11}] = P_\Delta$$

$$\pi_0 = \Pr[S_0] = (1 - P_\Delta)\pi_0 + (1 - P_\Delta)\pi_1$$
$$\pi_1 = \Pr[S_1] = P_\Delta \pi_1 + P_\Delta \pi_0$$

3. Derive expectations for events of interest

Expected number of C.O.s in T rounds is $T * \frac{P_\Delta^2}{\sum_{i,j} \Pr[e_{ij}]\pi_{ij} l_{ij}}$

1. Define a Markov model with states & events of interest



2. Compute stationary distribution for states and edges

$$P_\Delta = (1-\mu p)^\Delta$$

$$\Pr[e_{00}] = \Pr[e_{10}] = 1 - P_\Delta$$
$$\Pr[e_{01}] = \Pr[e_{11}] = P_\Delta$$

$$\pi_0 = \Pr[S_0] = (1-P_\Delta)\pi_0 + (1-P_\Delta)\pi_1$$
$$\pi_1 = \Pr[S_1] = P_\Delta \pi_1 + P_\Delta \pi_0$$

3. Derive expectations for events of interest

Expected number of C.O.s in T rounds is $T * \frac{P_\Delta^2}{\sum_{i,j} \Pr[e_{ij}]\pi_{ij} l_{ij}}$

4. Apply Concentration theorems
   - Number of C.O.s is concentrated around the expectation

Theorem (Our Nakamoto Consistency)

Nakamoto's protocol satisfies consistency if there exists $\delta > 0$ *such that*

$\beta = \Pr[\text{adversary mines a block}]$

$$\frac{P_\Delta^2}{\sum_{i,j} \Pr[e_{i,j}]\pi_{i,j} l_{i,j}} \geq (1+\delta)\beta$$

Theorem (Our Nakamoto Consistency)

Nakamoto's protocol satisfies consistency if there exists $\delta > 0$ such that

$\beta = \Pr[\text{adversary mines a block}]$

$$\underbrace{\frac{P_\Delta^2}{\sum_{i,j} \Pr[e_{i,j}]\pi_{i,j} l_{i,j}}}_{\text{Fraction of events that are convergence opportunities}} \geq (1+\delta)\beta$$

Fraction of events that
are convergence opportunities

Theorem (Our Nakamoto Consistency)

Nakamoto's protocol satisfies consistency if there exists $\delta > 0$ such that

$\beta = \Pr[$adversary mines a block $]$

$$\underbrace{\frac{P_\Delta^2}{\sum_{i,j} \Pr[e_{i,j}]\pi_{i,j} l_{i,j}}}_{\text{Fraction of events that are convergence opportunities}} \geq \underbrace{(1+\delta)\beta}_{\text{Fraction of events that are adversarial mined blocks}}$$

Theorem (Our Nakamoto Consistency)

Nakamoto's protocol satisfies consistency if there exists $\delta > 0$ such that

In T rounds, with probability $\geq 1 - \epsilon_1(k) - \epsilon_2(T)$

$$\underbrace{T * \frac{P_{\Delta}^2}{\sum_{i,j} \Pr[e_{i,j}]\pi_{i,j} l_{i,j}}}_{\text{Number of blocks adversary needs}} \geq \underbrace{(1+\delta)\beta * T}_{\text{Number of blocks adversary can mine}}$$

Roadmap

1. How to analyze consistency
2. Our analysis on Nakamoto consistency
3. **An attack on Nakamoto consistency**
4. Cliquechain consistency and attack
5. GHOST consistency and attack

Nakamoto Delay Attack

Attack:

- delay receipt of all honest blocks by Δ
- adversary mines secret chain efficiently

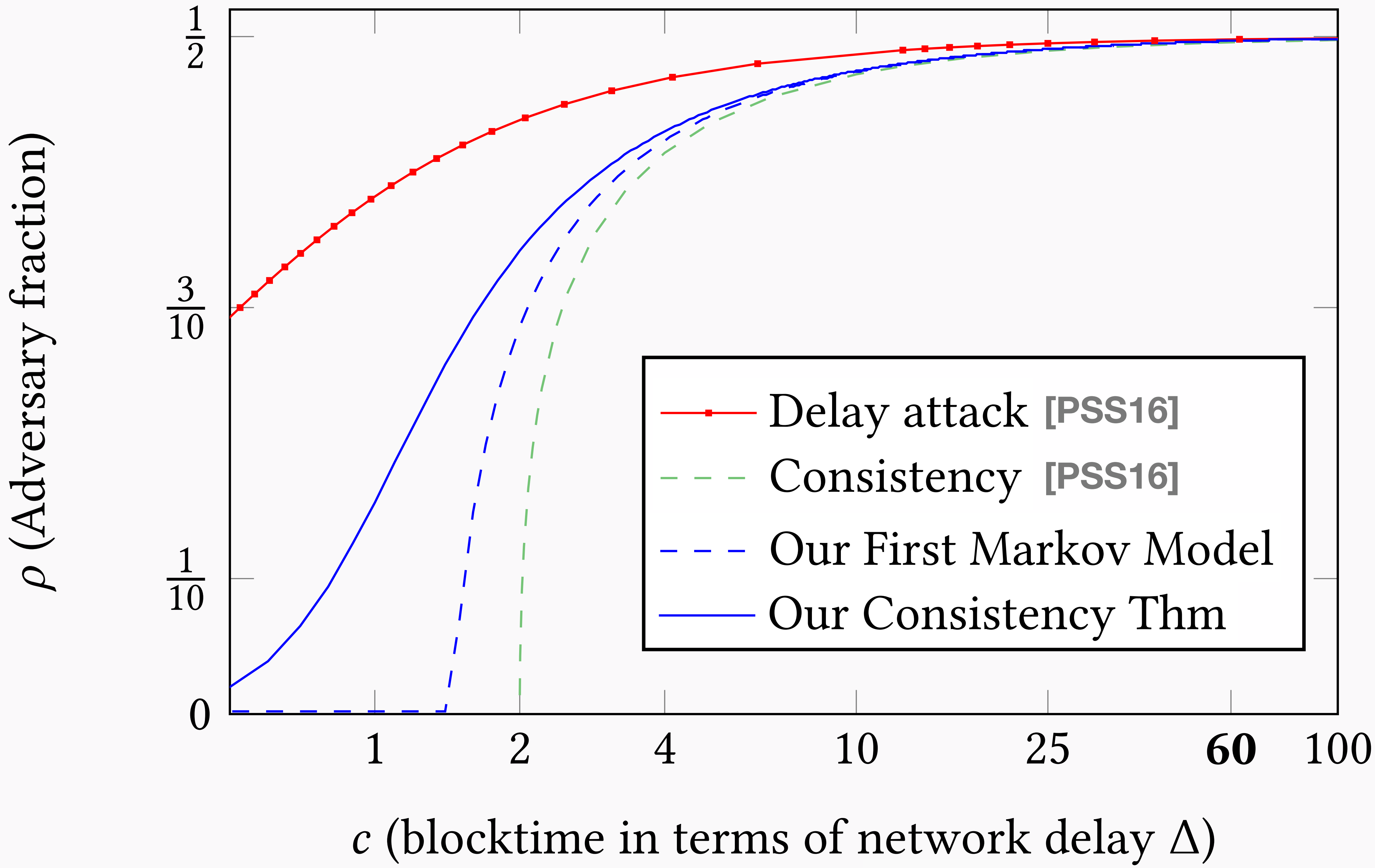
Nakamoto Delay Attack

Attack:

- delay receipt of all honest blocks by Δ
- adversary mines secret chain efficiently

Goal: Thwart the rate of growth of the honest chain so adversary's secret chain is longer

Our Result: An Improved Analysis



Recall mining hardness is $p = \frac{1}{c \cdot n\Delta}$

DELAY ATTACK ON NAKAMOTO
HOW LONG TO WAIT

Height	593468
Time	2019-09-05 23:30:05
Difficulty	10,771,996,663,680.39648438
Bits	387588414
Version	1073733632
Nonce	4183169508
Block Reward	12.5 BTC
Days Destroyed	84,508,971

Hash	00000000000000000014fcb29e6e3b0ead3bd2e307d7f619a935f1d5323e9013
Previous Block	00000000000000000011bd5a659d7eb86644579e3da466812f3d39ce46d62fb5
Next Block(s)	0000000000000000000096932531c7e92094cccc6e260aec6271eb406a1de9c2b
Merkle Root	5c42d325609babad08921ce5bffa508a673e34af5c20ef7d552ff4199ace7243

tx:118732872f9172d85fbed0918ed8b7ca428f6f0c3f60ccb788e40379124c2056 1.646209 BTC Fee: 0.05176597 BTC

Input	Amount	Output	Amount
prev tx 14tMizKCDy2937HiBTp22nBoyiYo6ZEcFr wallet: 54183648	-0.89895996 BTC	13YeXYBbvwd4nSF8xWzEp6KvTUzUm7qaks wallet: 49851654	0.646209 BTC
prev tx 1QAmJsuejUb6bieSCVukNxqHV8zYVqD61v wallet: 54183648	-0.79901501 BTC	37XuVSEpWW4trkfmvWzegTHQt7BdktSKUs	1 BTC

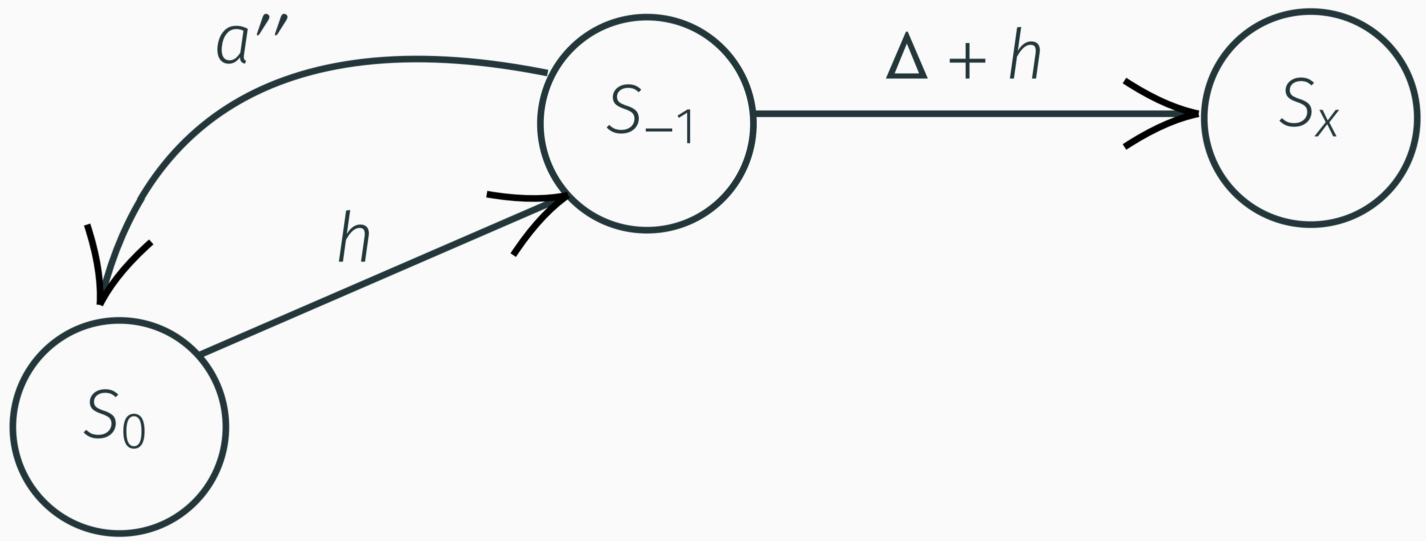
tx:4410c8d14ff9f87ceeed1d65cb58e7c7b2422b2d7529afc675208ce2ce09ed7d 94,504.03465148 BTC Fee: 0.06534852 BTC

Input	Amount	Output	Amount
prev tx 1JCe8z4jJVNXSjohjM4i9Hh813dLCNx2Sy wallet: 49851654	-20,000 BTC	37XuVSEpWW4trkfmvWzegTHQt7BdktSKUs	94,504.03465148 BTC
prev tx 1JCe8z4jJVNXSjohjM4i9Hh813dLCNx2Sy wallet: 49851654	-18,000 BTC		
prev tx 1JCe8z4jJVNXSjohjM4i9Hh813dLCNx2Sy wallet: 49851654	-15,000 BTC		
prev tx 1PfceCKGraSPEvx6nfjw5ZCLLy8Ct23Qd5 wallet: 49851654	-14,999.89950753 BTC		
prev tx 1KKiEAkpnQR2FH5kpkGP6442ZDkd6ZdrRS wallet: 49851654	-12,799.99950753 BTC		
prev tx 15NQthxeLSwMtEaXJFM7YUCf59LzmFjkeH wallet: 49851654	-11,799.99950753 BTC		
prev tx 18b3BfortqFEPHx8vRHVz3LJU7gBECuP51 wallet: 49851654	-1,103.99895996 BTC		

NAKAMOTO DELAY ATTACK

State S_0 represents two chains of equal length

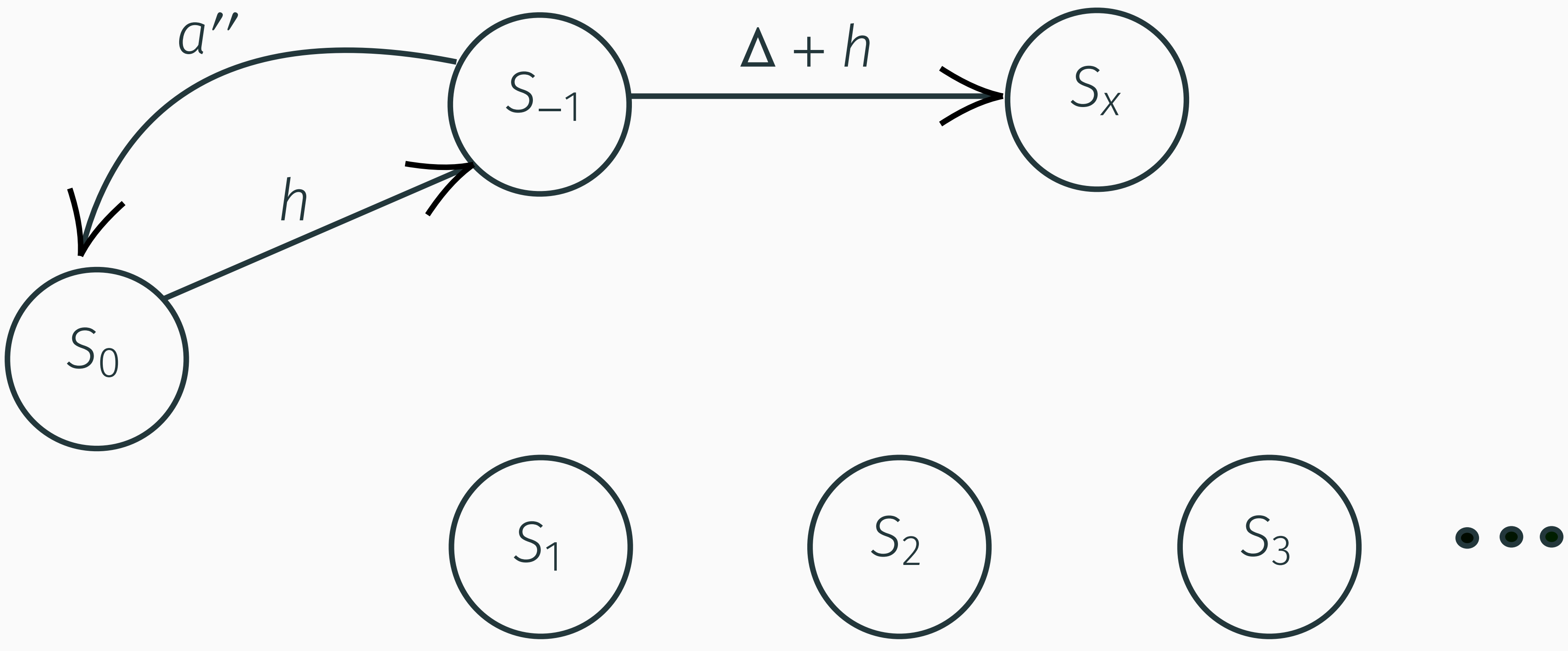
NAKAMOTO DELAY ATTACK



State S_{-1}: honest chain ahead by one block.

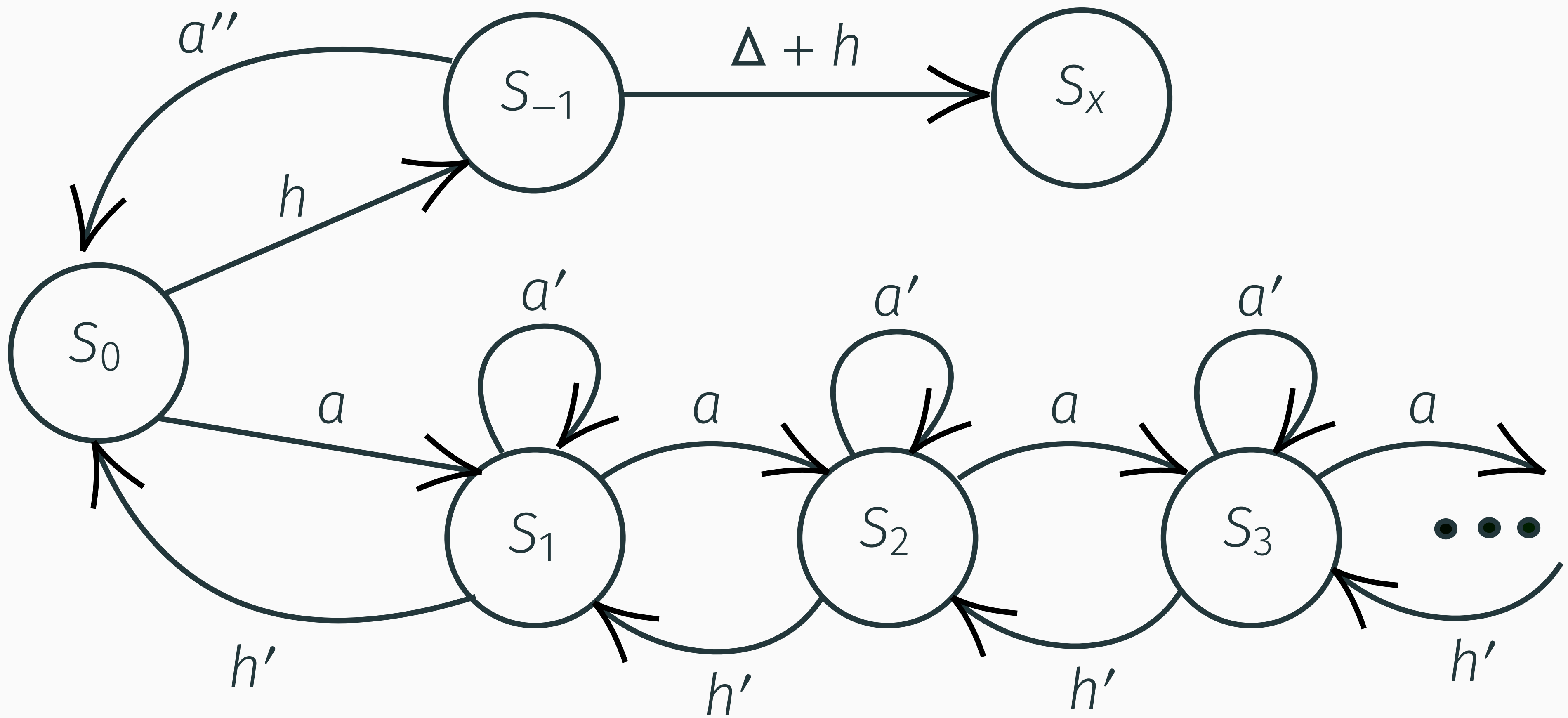
S_X: attack has failed

NAKAMOTO DELAY ATTACK

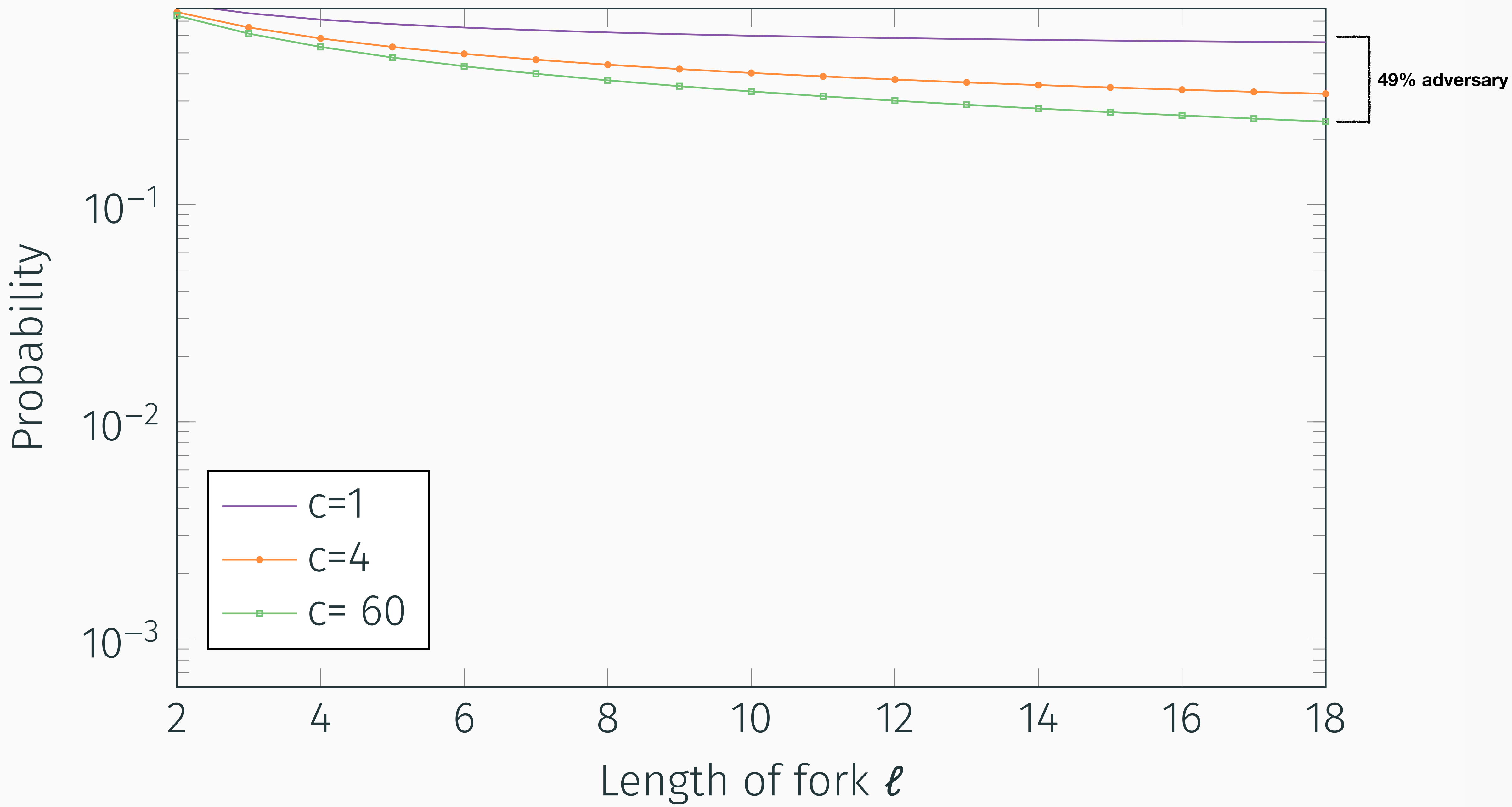


State S_i: adversary chain is ahead by i blocks

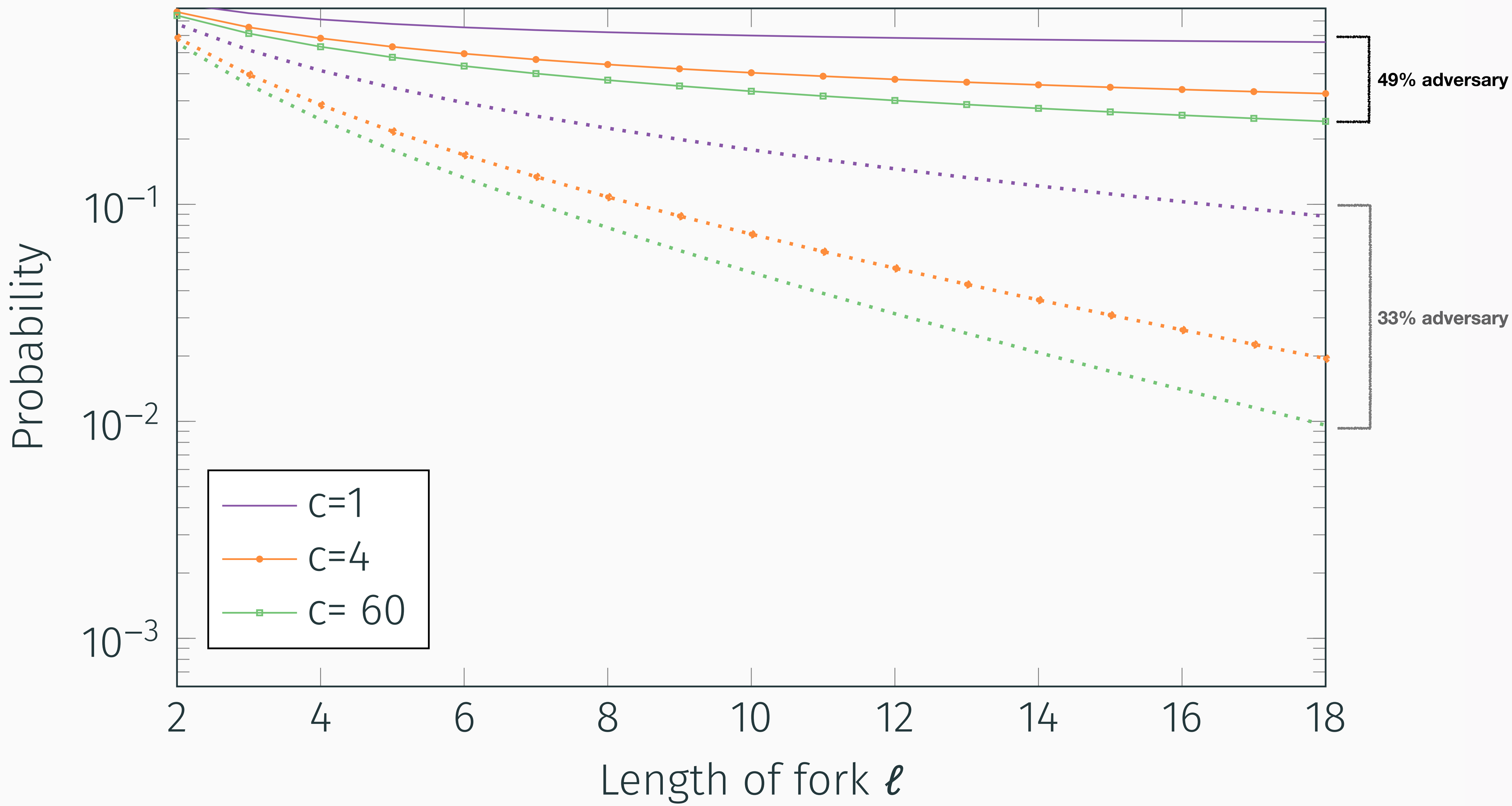
NAKAMOTO DELAY ATTACK



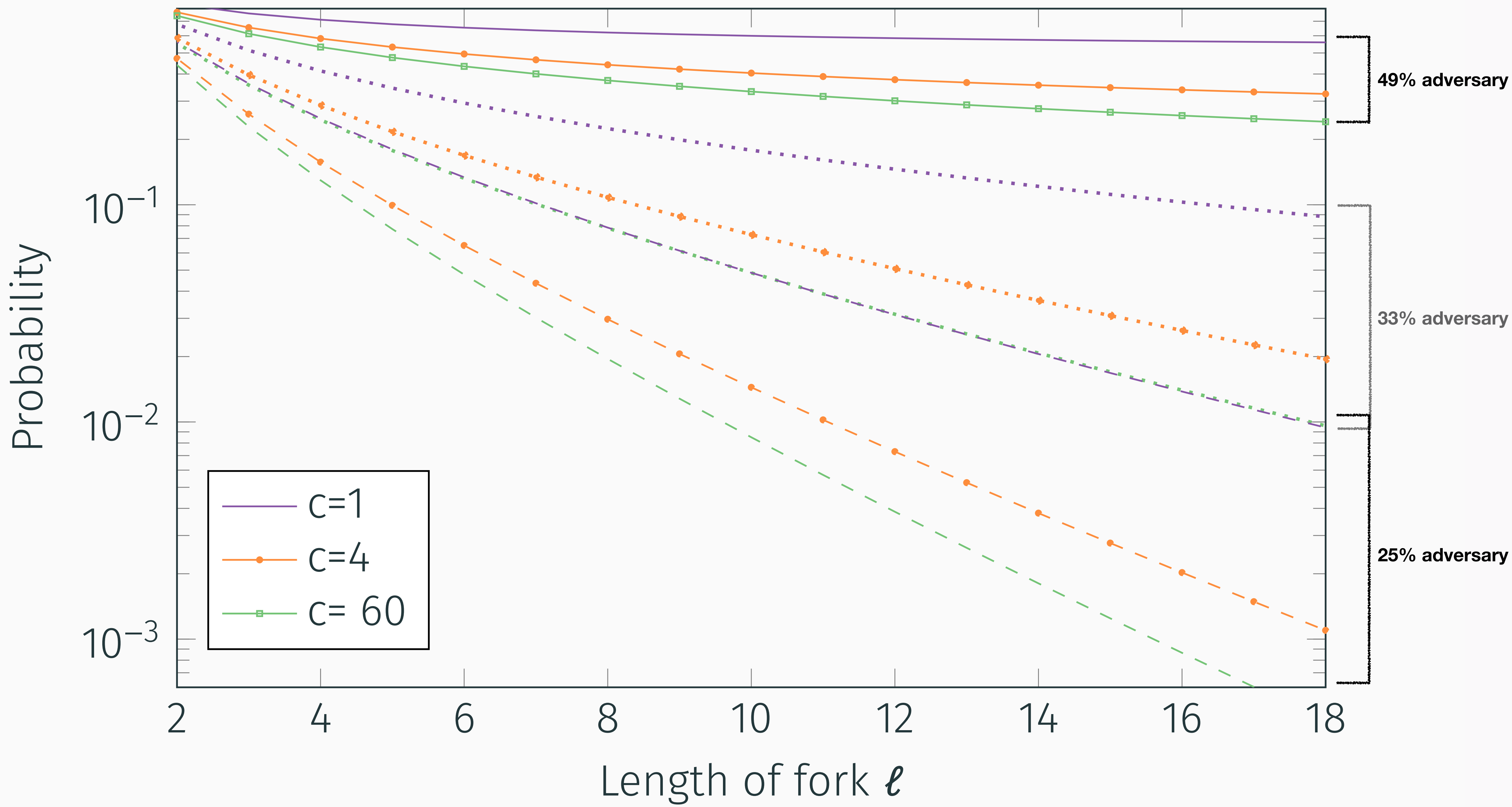
$P_0(k)$: Prob of passing through k bad edges before state S_X



Pr for Nakamoto to sustain a fork of length ℓ at 49% Adversary.



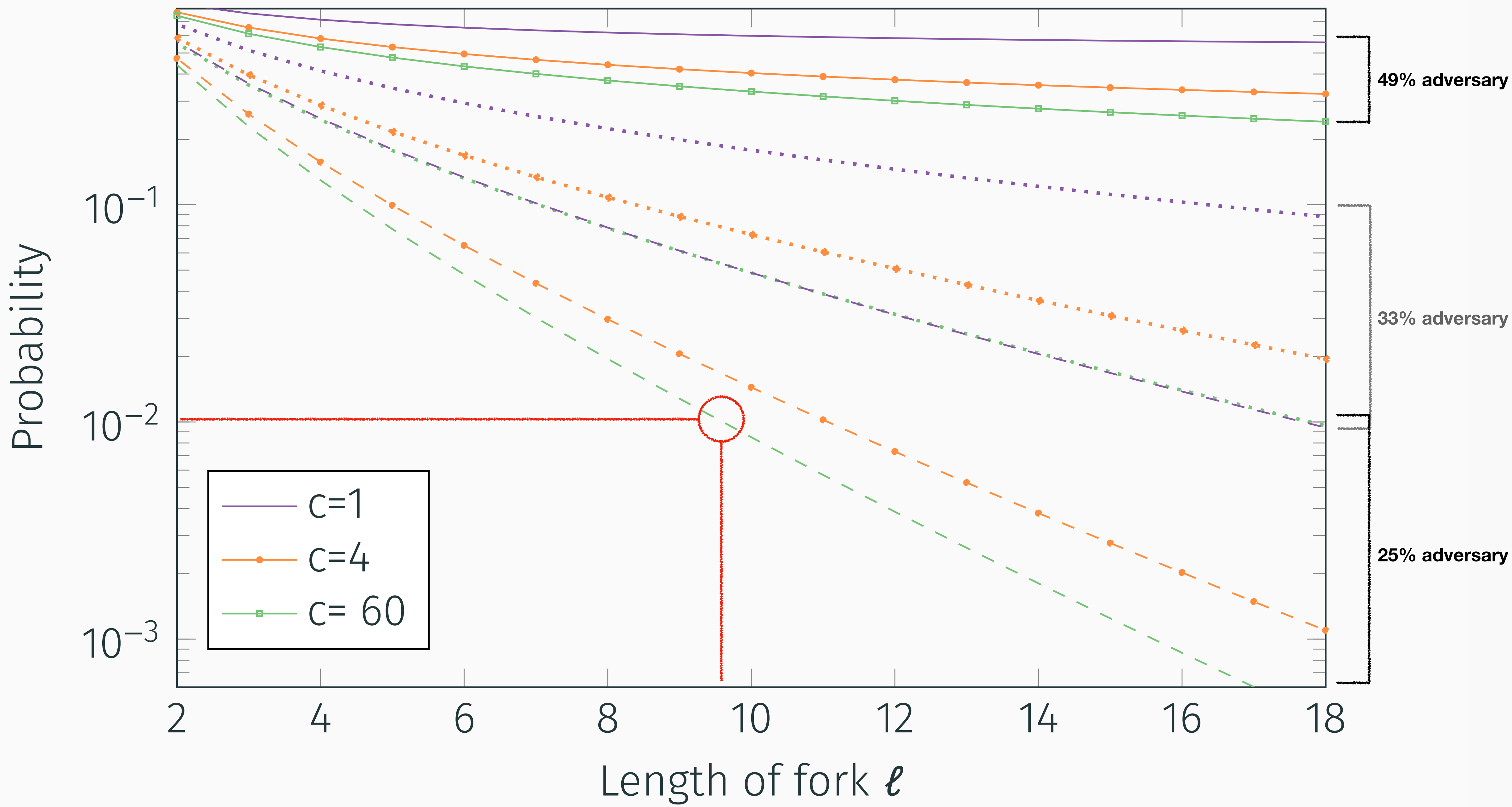
Pr for Nakamoto to sustain a fork of length ℓ at 49% Adversary.
Now 33% adversary



Pr for Nakamoto to sustain a fork of length ℓ at 49% Adversary.

Now 33% adversary

Now 25% adversary



Pr for Nakamoto to sustain a fork of length ℓ at 49% Adversary.

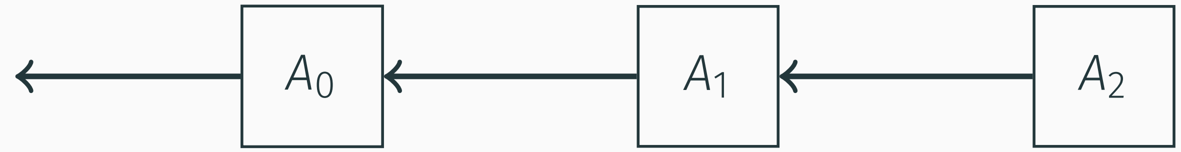
Now 33% adversary

Now 25% adversary

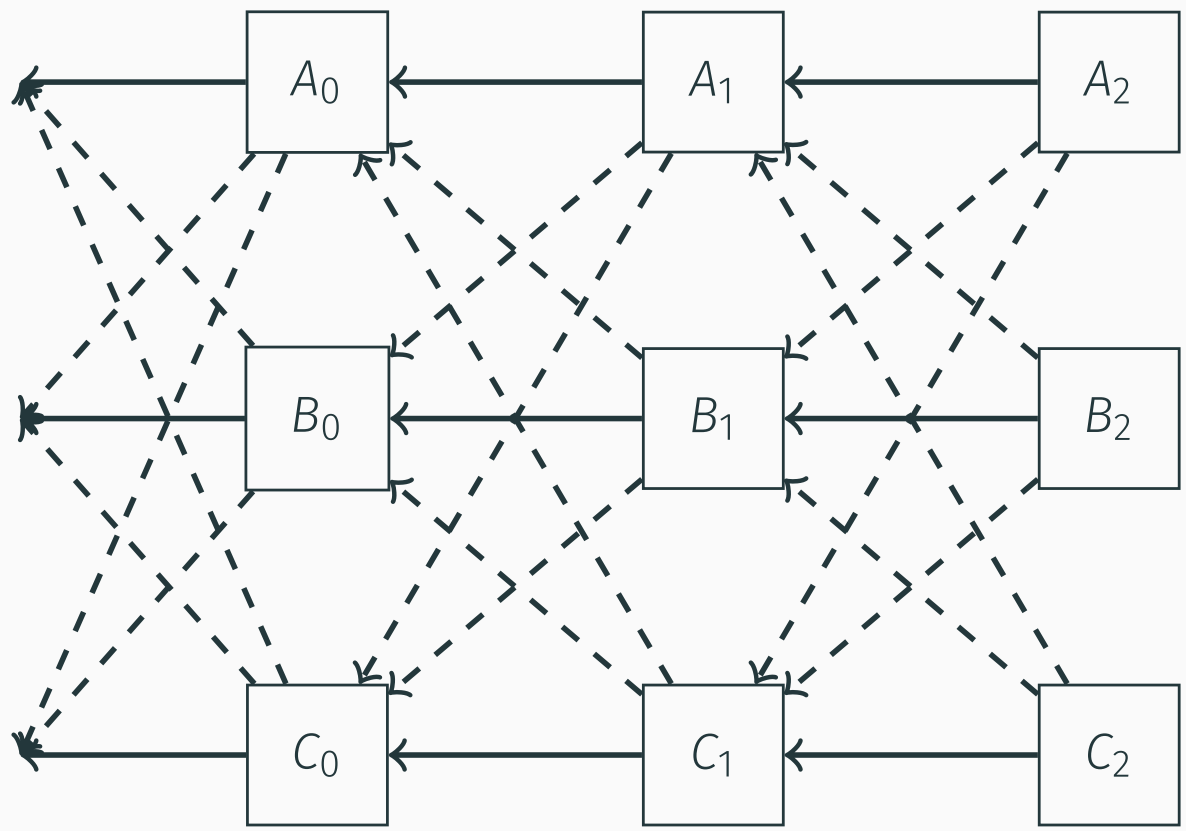
Roadmap

1. How to analyze consistency
2. Our analysis on Nakamoto consistency
3. An attack on Nakamoto consistency
4. **Cliquechain consistency and attack**
5. **GHOST consistency and attack**

ANALYZING CLIQUECHAIN

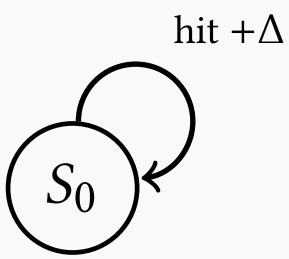


ANALYZING CLIQUECHAIN

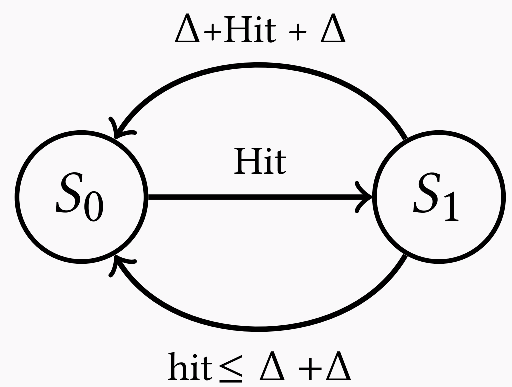


ANALYZING CLIQUECHAIN

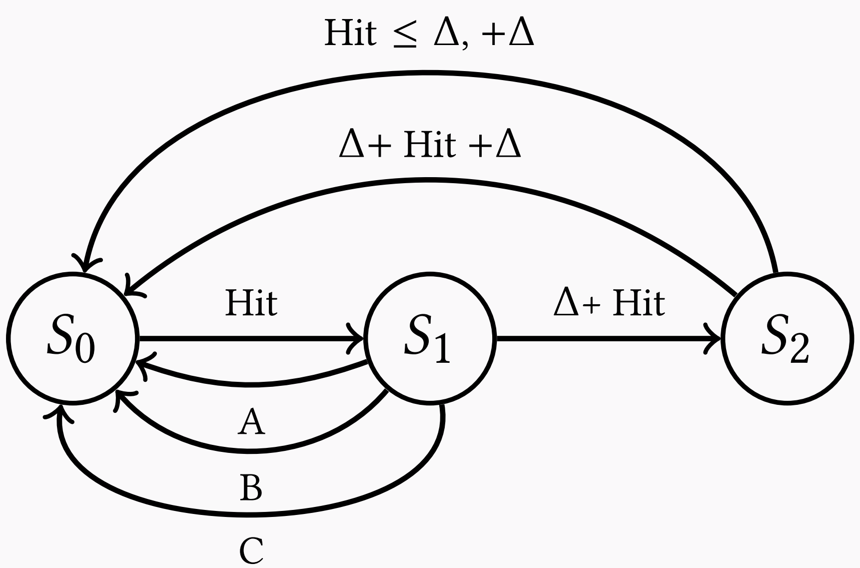
1-chain

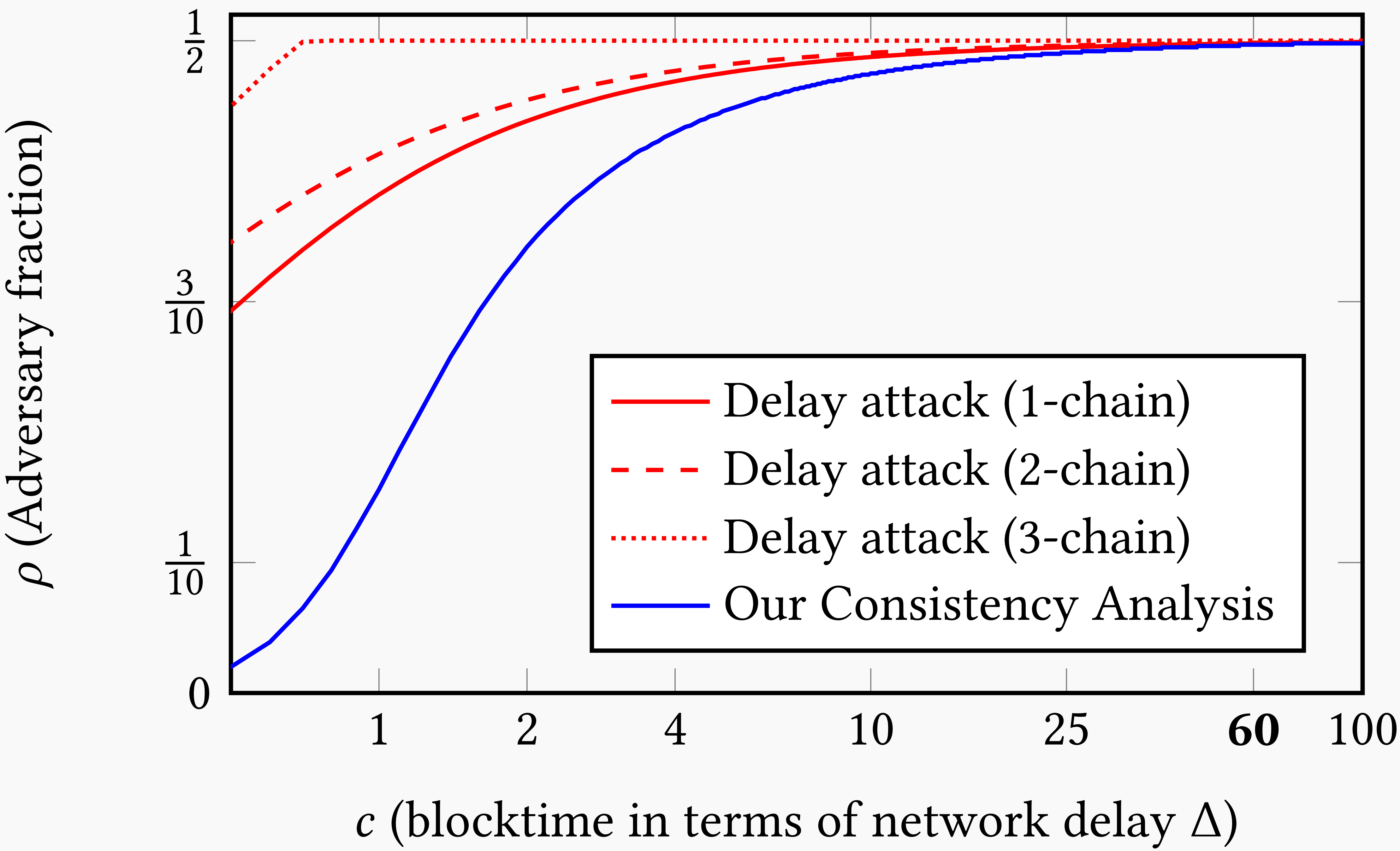


2-chain



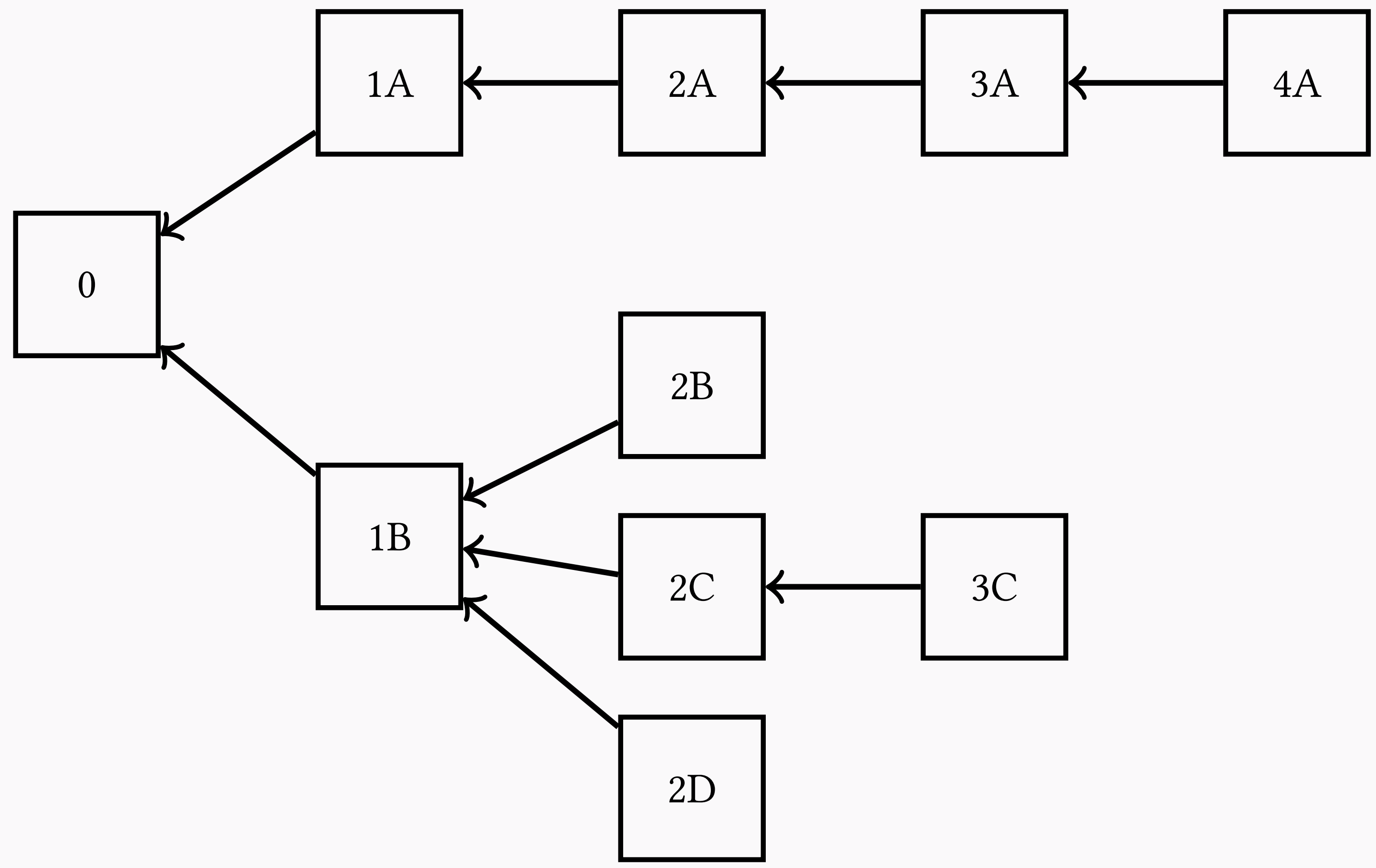
3-chain



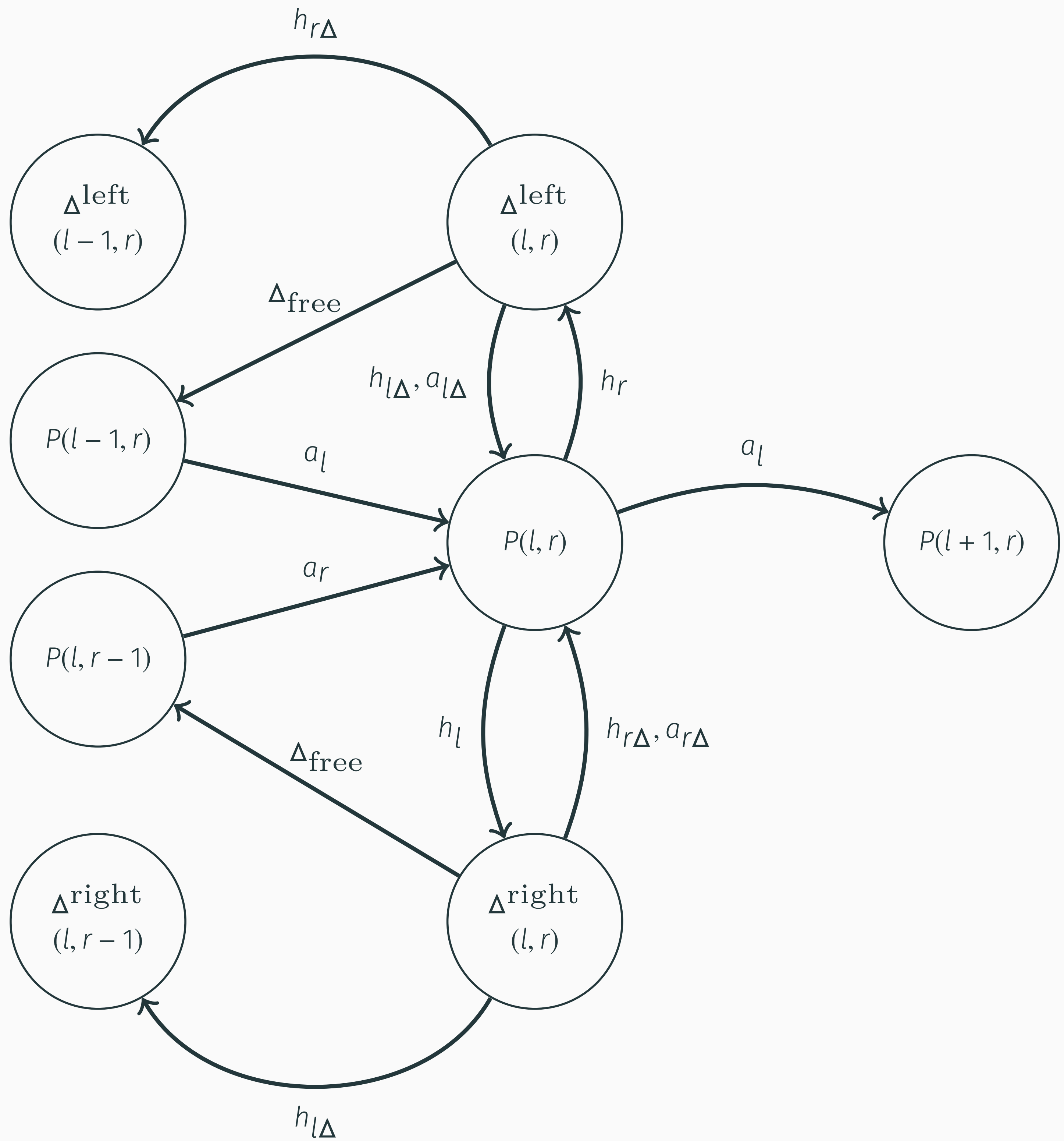


$\frac{1}{2}$
$\frac{3}{10}$
$\frac{1}{10}$
0
ρ (Adversary fraction)
1
2
4
10
25
60
100
c (blocktime in terms of network delay Δ)
Delay attack (1-chain)
Delay attack (2-chain)
Delay attack (3-chain)
Our Consistency Analysis

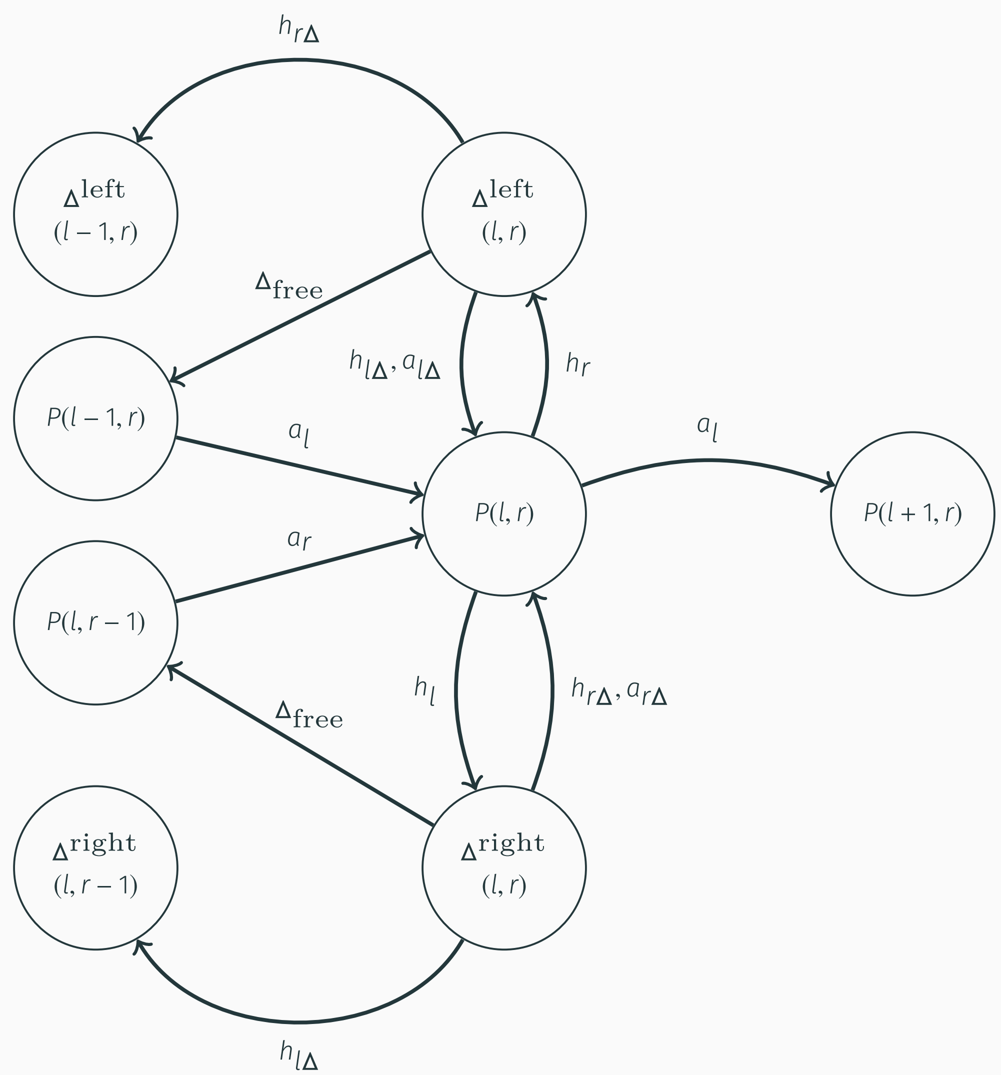
ANALYZING GHOST



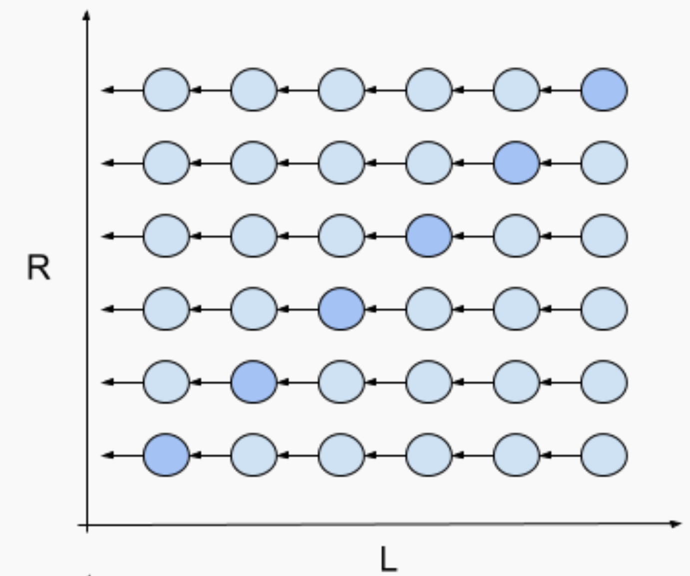
ANALYZING GHOST



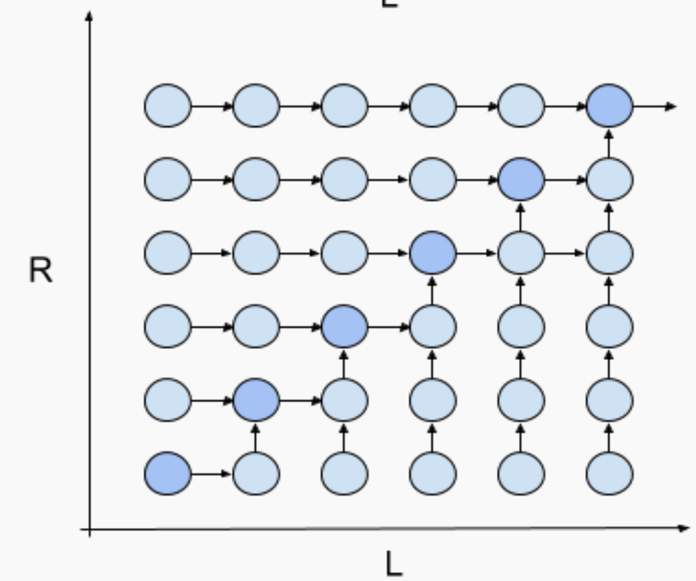
ANALYZING GHOST



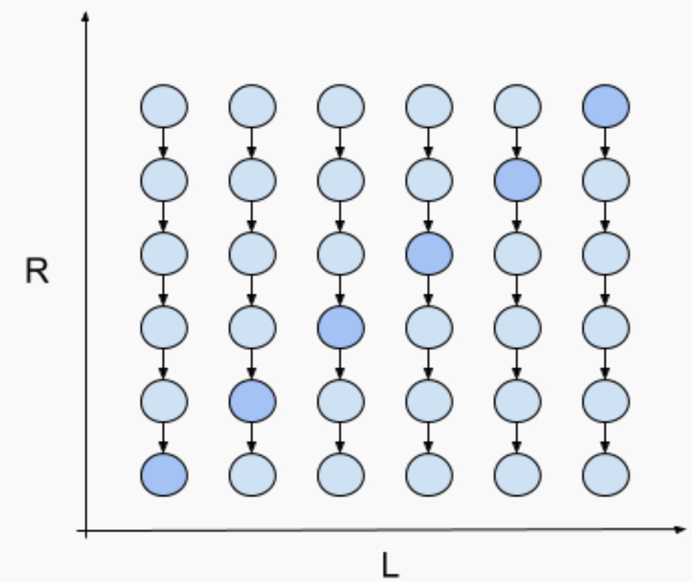
Δleft



Parity



Δright



Summary

Our Markov model framework provides a flexible model to reason about the consistency property of different blockchain protocols, allowing us to:

- Reduce three different blockchain protocols to the same consistency lower bound
- Reason about the success of attacks on consistency

Future work: continue to extend our analysis to different protocols and attacks.

THX

H,q,H,Q,H,Q,H,q,H,q,H,q,...,