# Hijacking Bitcoin

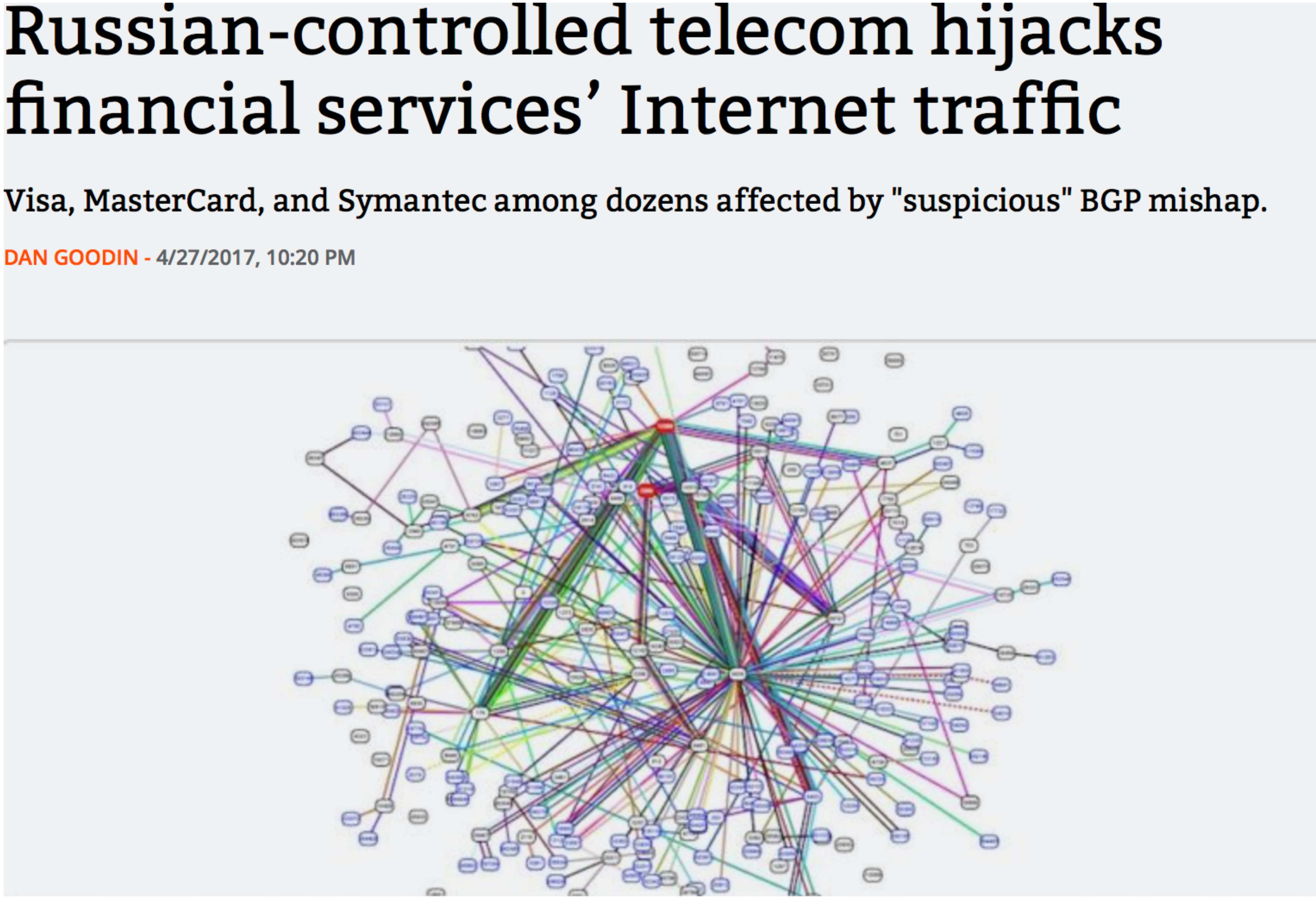
## Routing Attacks on Cryptocurrencies

**Maria Apostolaki**

ETH Zürich

Joint work with Aviv Zohar and Laurent Vanbever

# Routing attacks quite often make the news



# Russian-controlled telecom hijacks financial services’ Internet traffic

Visa, MasterCard, and Symantec among dozens affected by "suspicious" BGP mishap.

DAN GOODIN - 4/27/2017, 10:20 PM

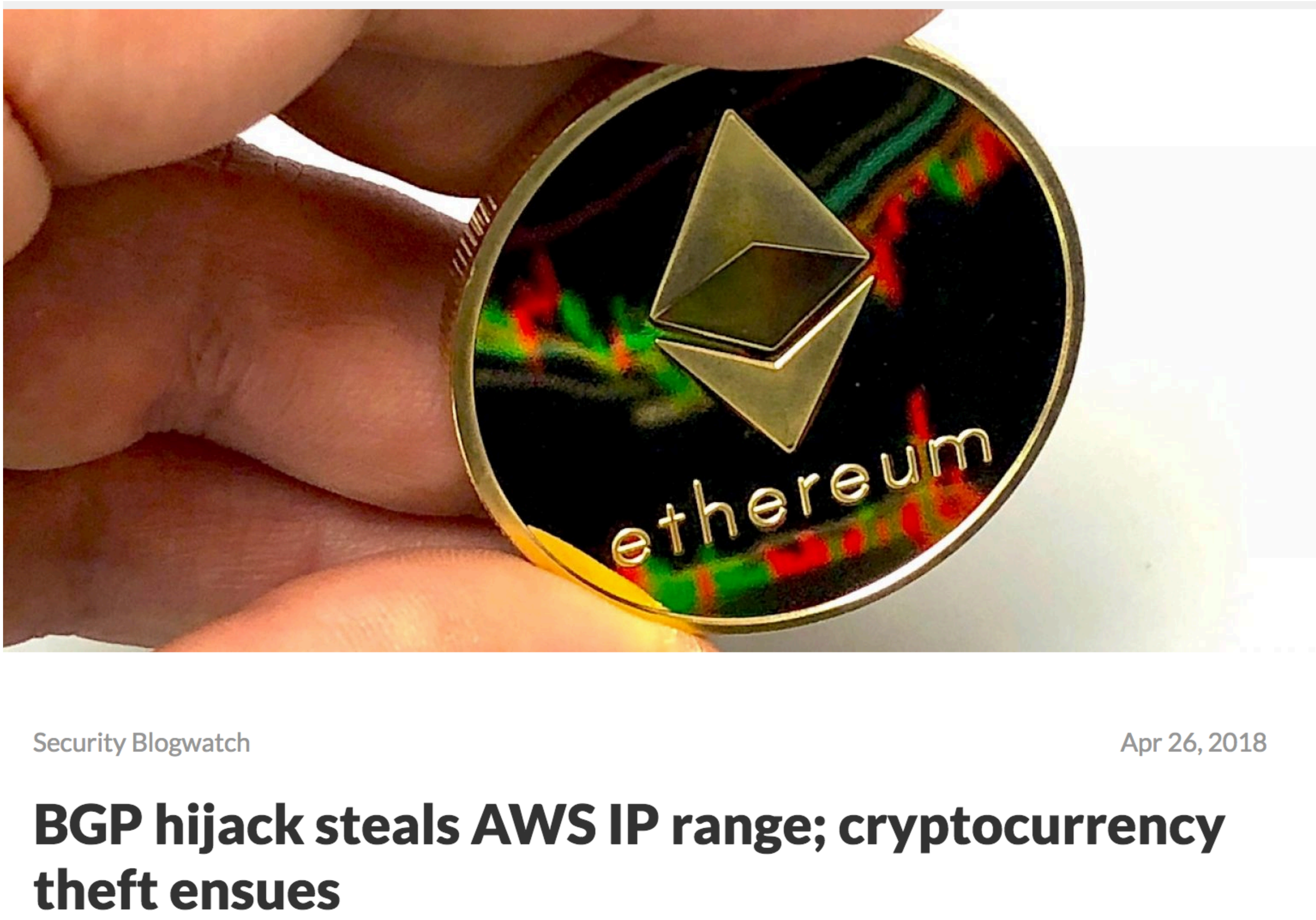
source: arstechnica.com

THREAT LEVEL

# Hacker Redirects Traffic From 19 Internet Providers to Steal Bitcoins

BY ANDY GREENBERG 08.07.14 | 1:00 PM | PERMALINK

source: wired.com



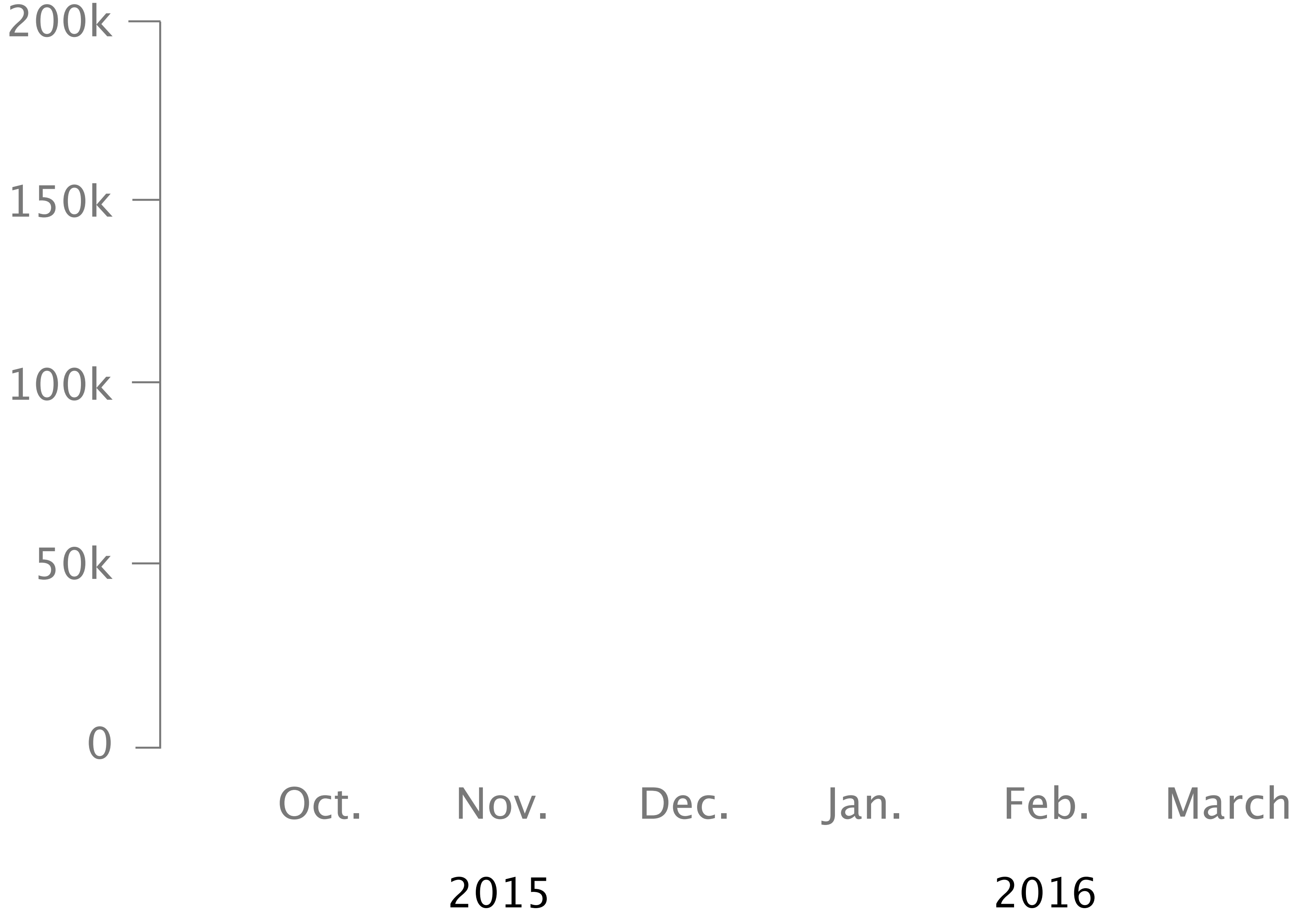
Security Blogwatch

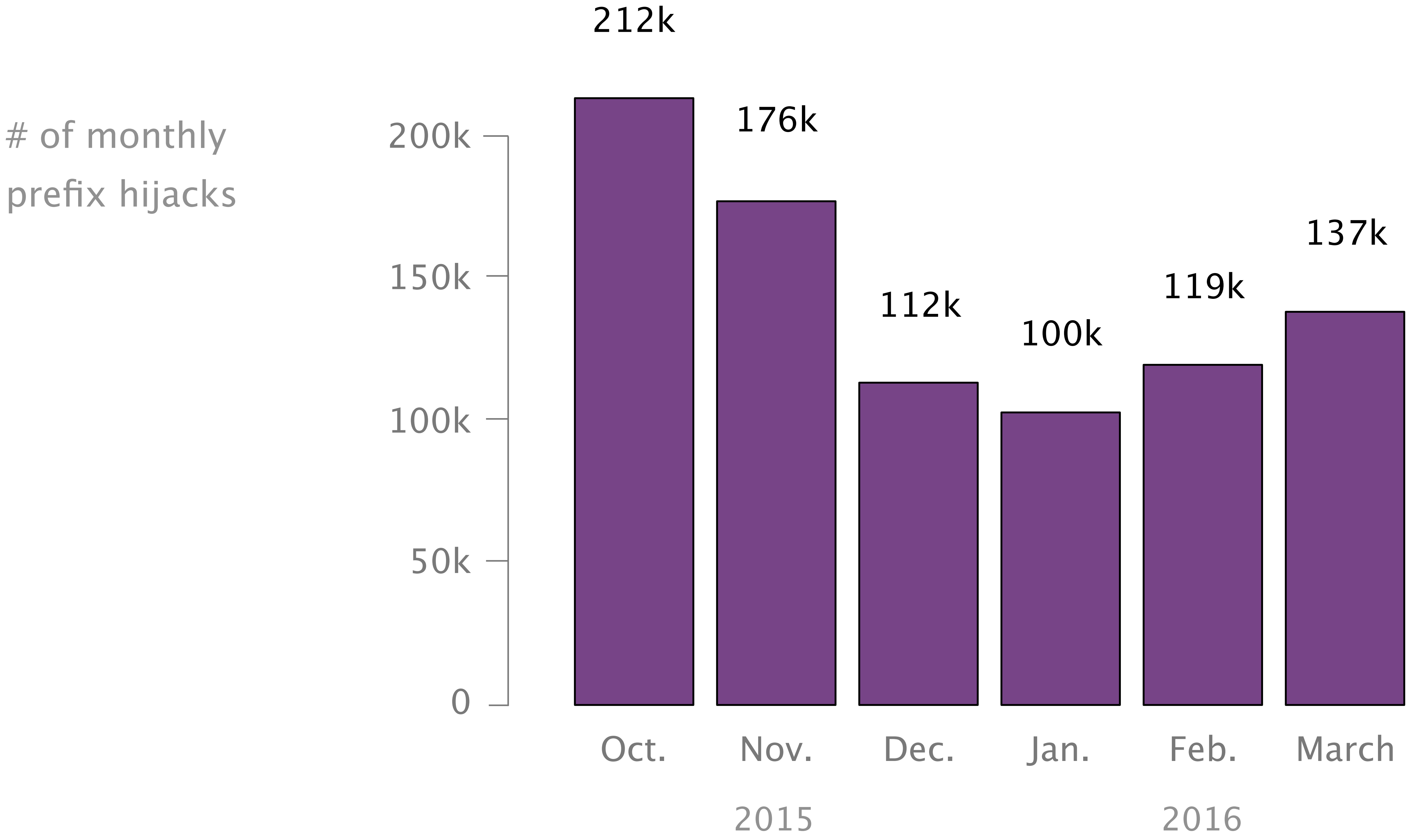
Apr 26, 2018

# BGP hijack steals AWS IP range; cryptocurrency theft ensues

# That is only the tip of the iceberg of routing manipulations

# of monthly
prefix hijacks




# of monthly
prefix hijacks
200k
150k
100k
50k
0
212k
176k
112k
100k
119k
137k
Oct.
Nov.
Dec.
Jan.
Feb.
March
2015
2016

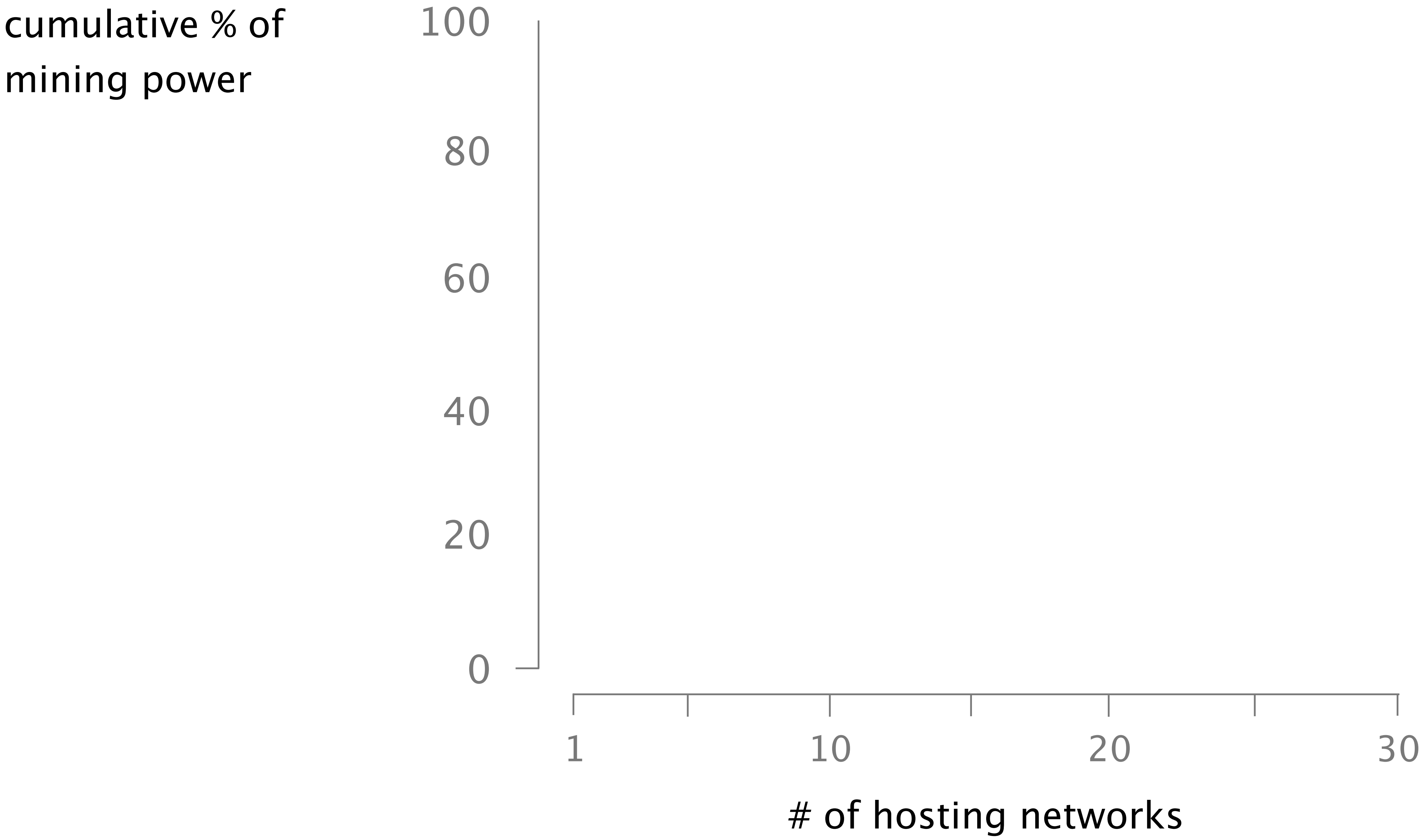
# Can routing attacks impact Bitcoin?

# Bitcoin is highly decentralized
# making it robust to routing attacks, in theory…

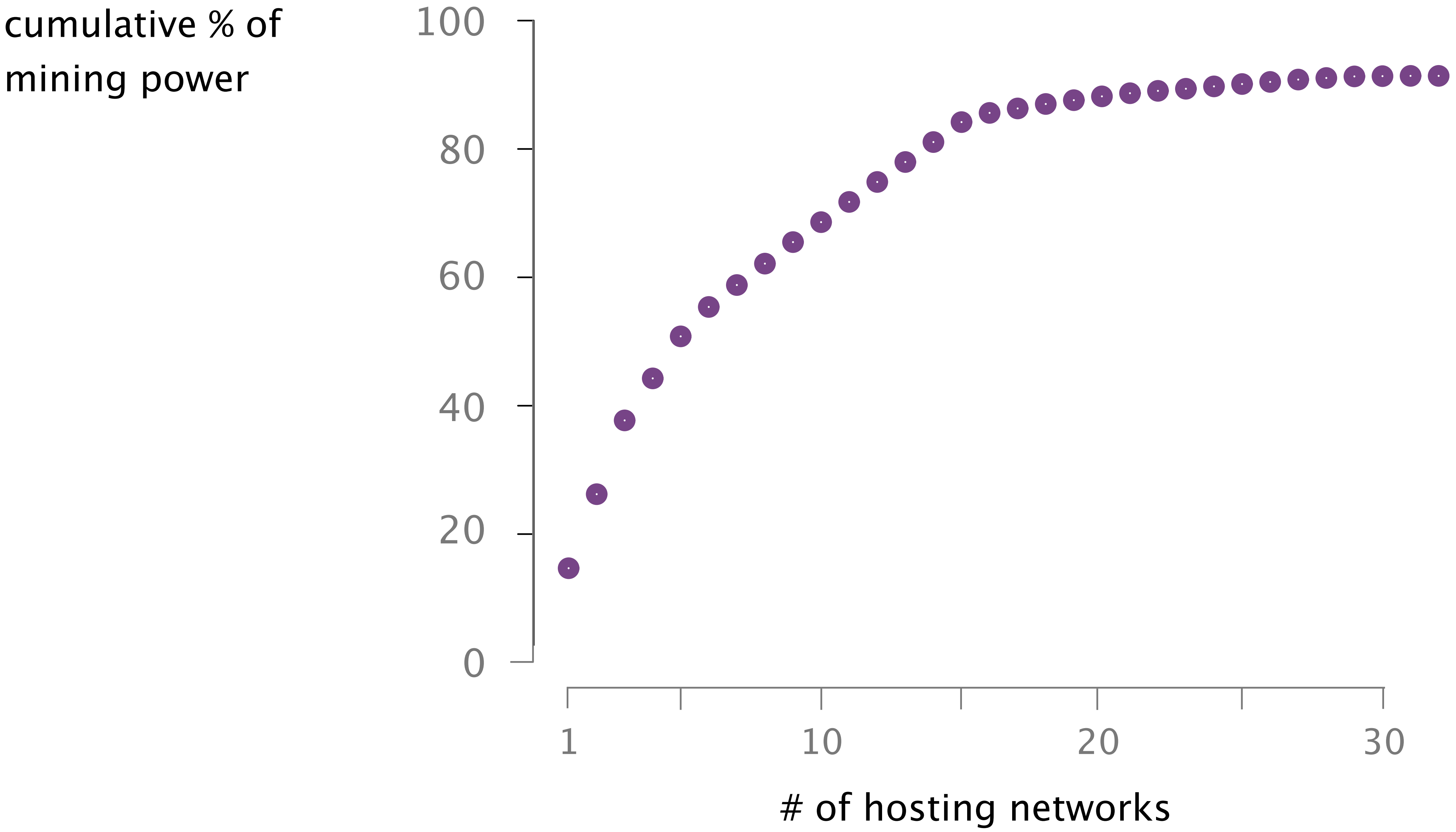
Bitcoin nodes …

- are scattered all around the globe
- establish random connections
- use multihoming and extra relay networks

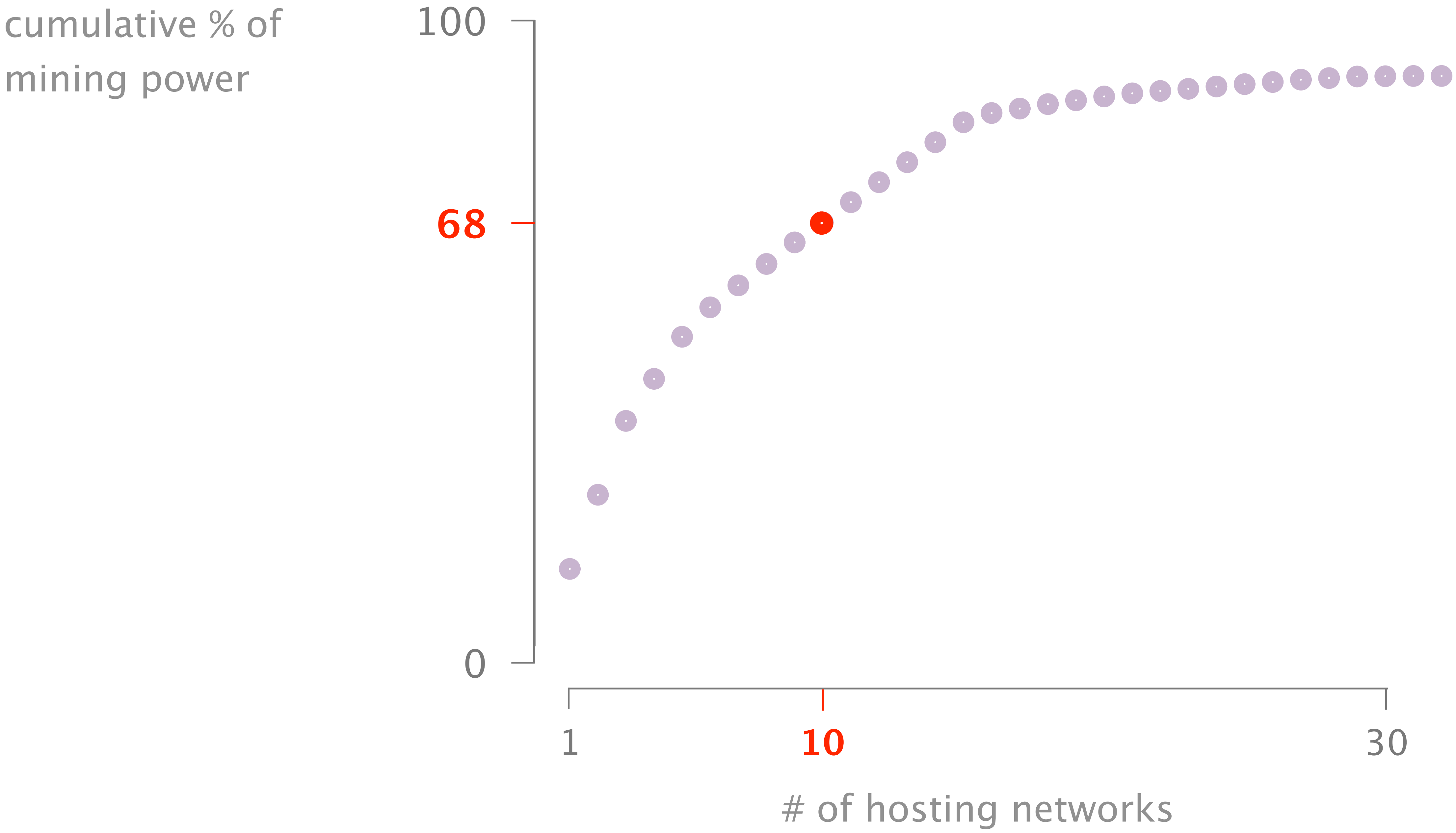
In practice, Bitcoin is highly centralized,
both from a routing and mining viewpoint

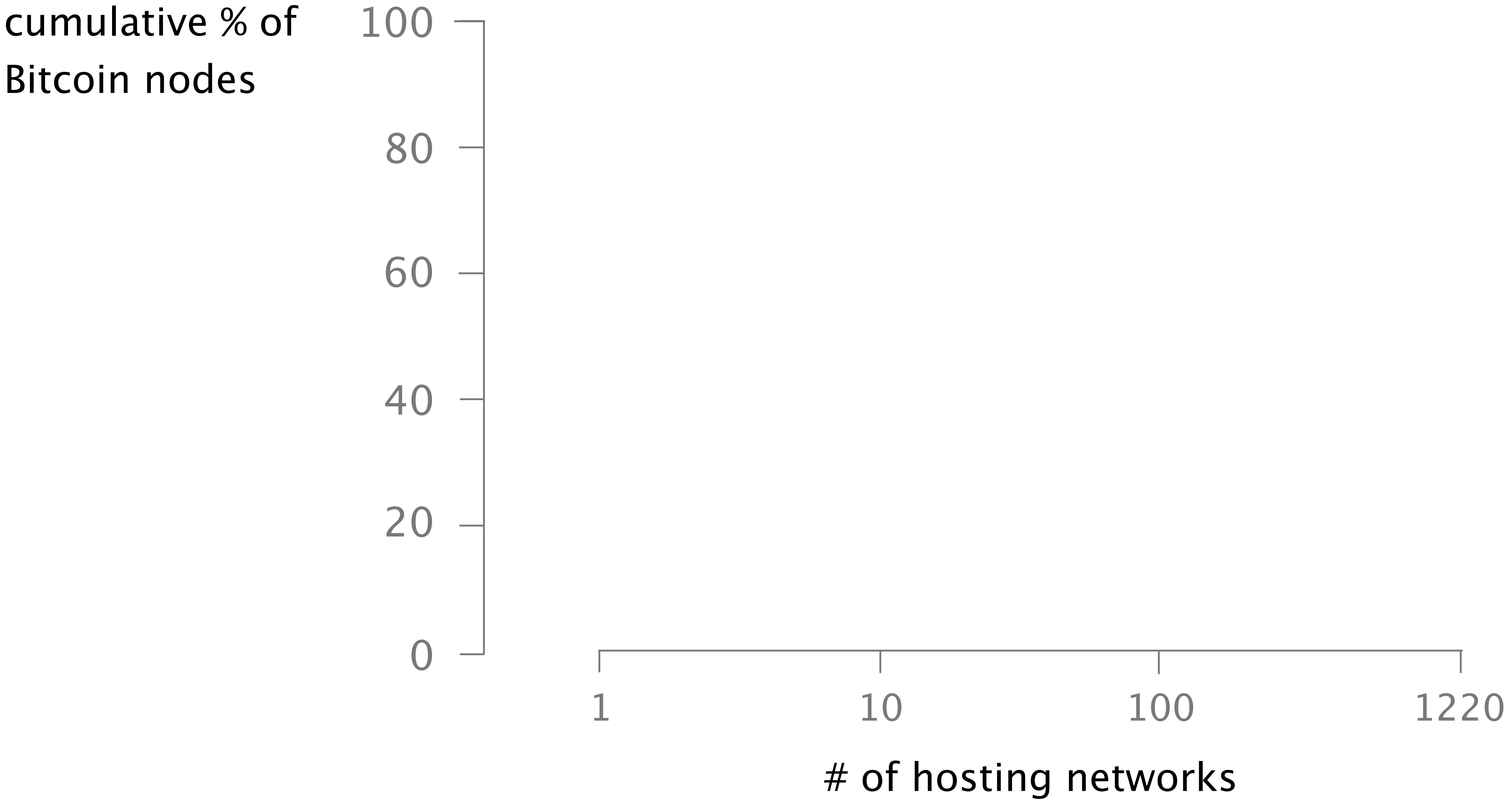

cumulative % of
mining power
100
80
60
40
20
0
1
10
20
30
# of hosting networks

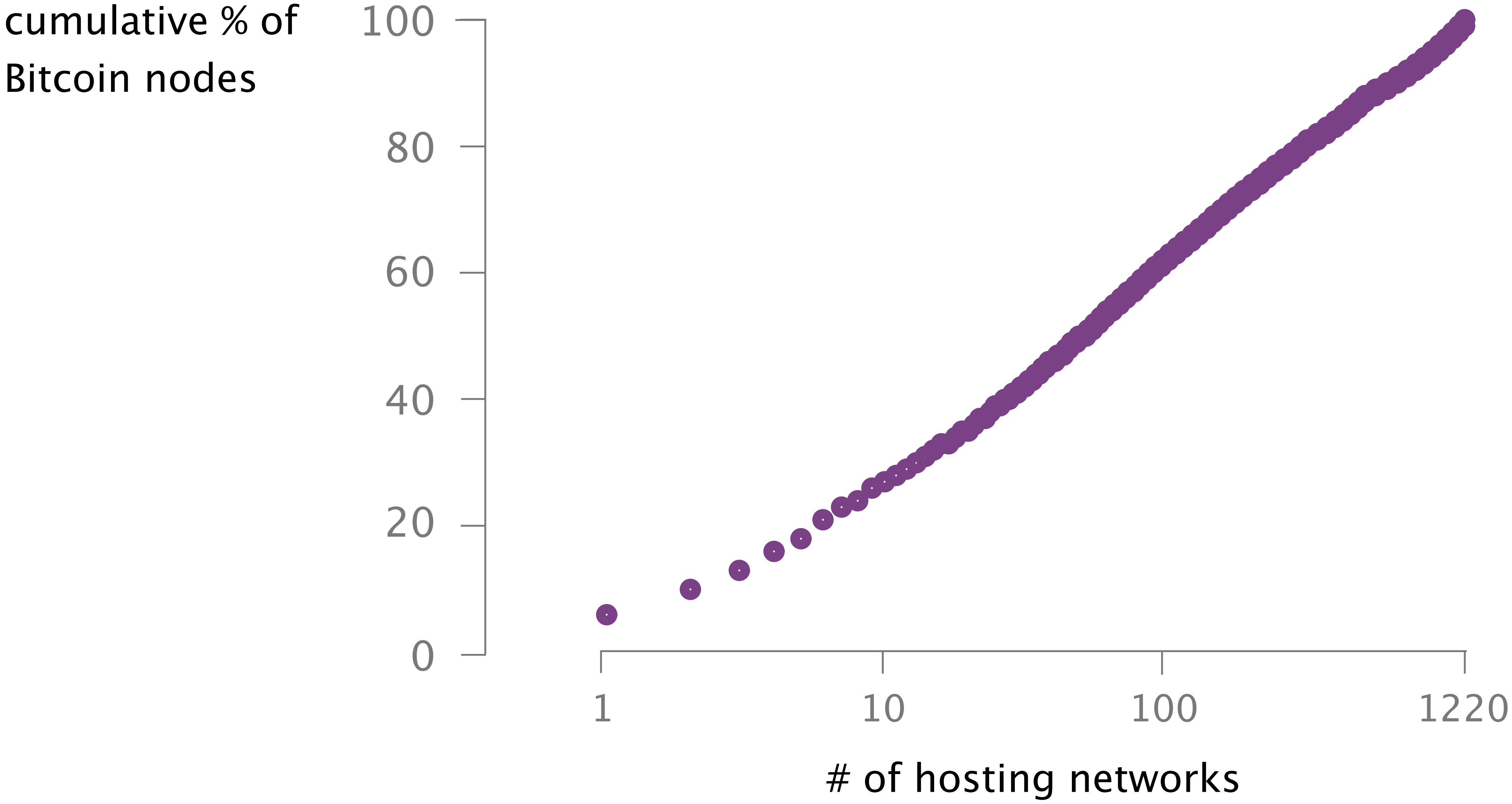
# Mining power is centralized to few hosting networks



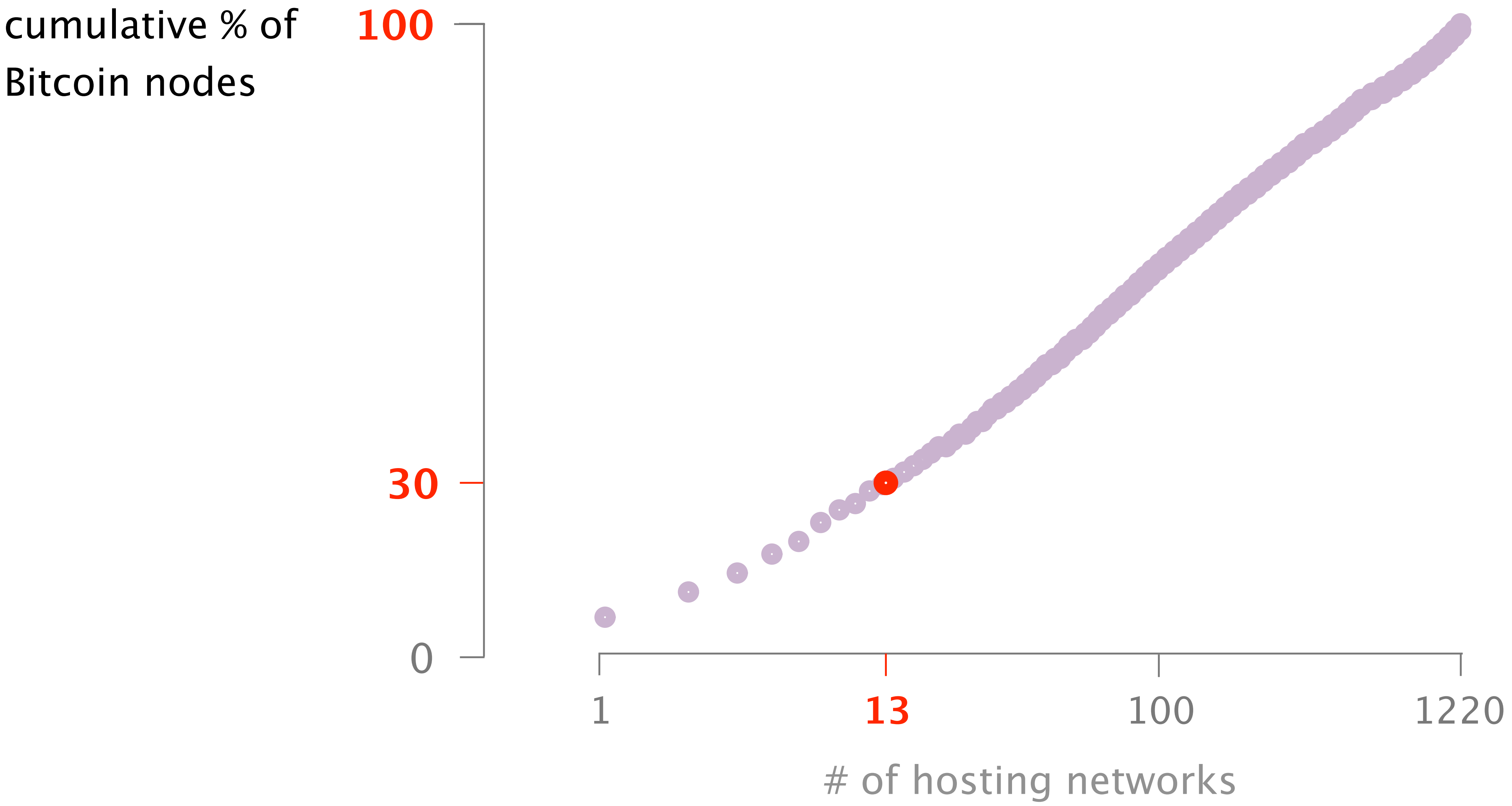
# 68% of the mining power is hosted in 10 networks only

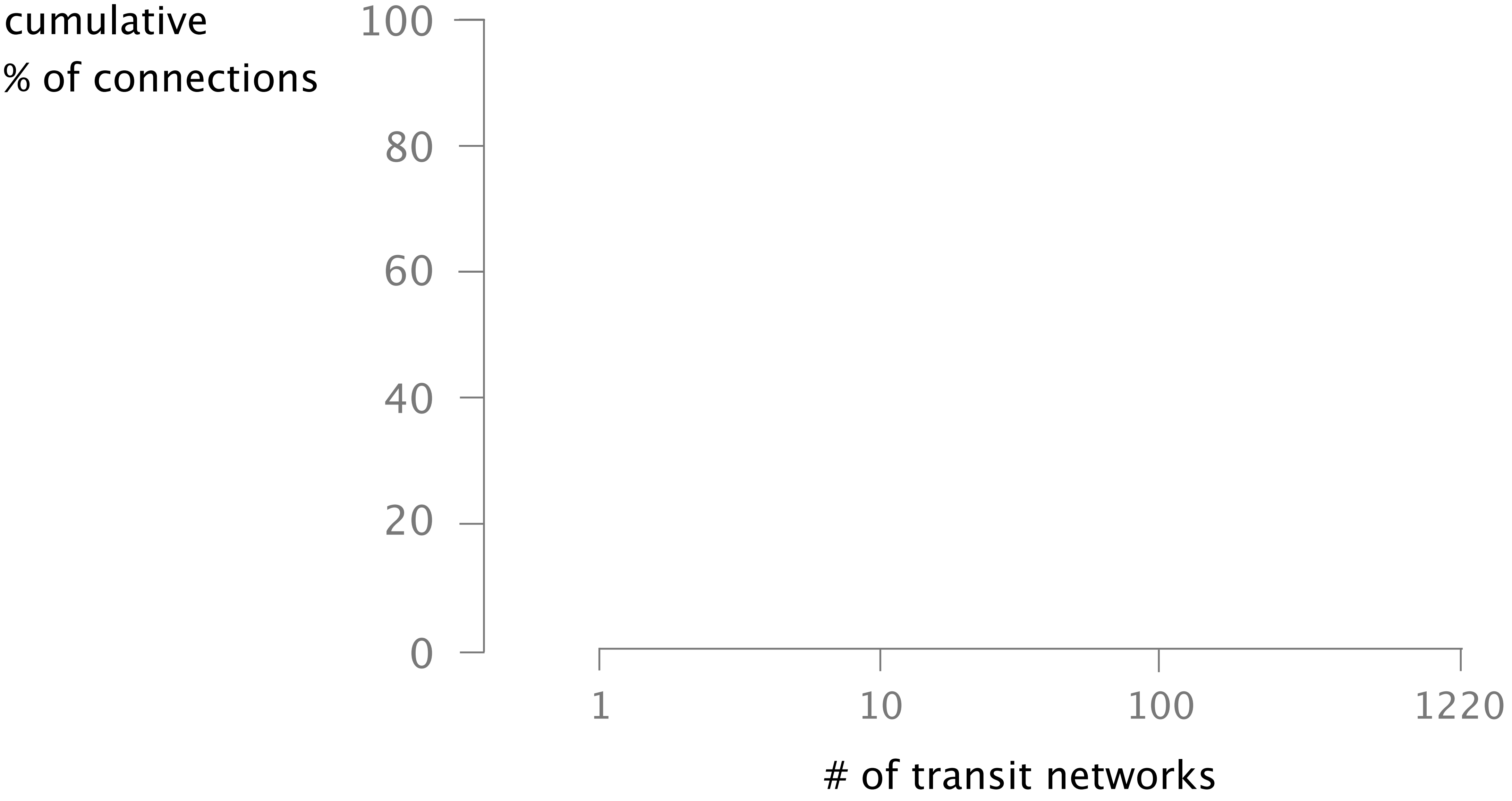



cumulative % of
Bitcoin nodes
100
80
60
40
20
0
1
10
100
1220
# of hosting networks

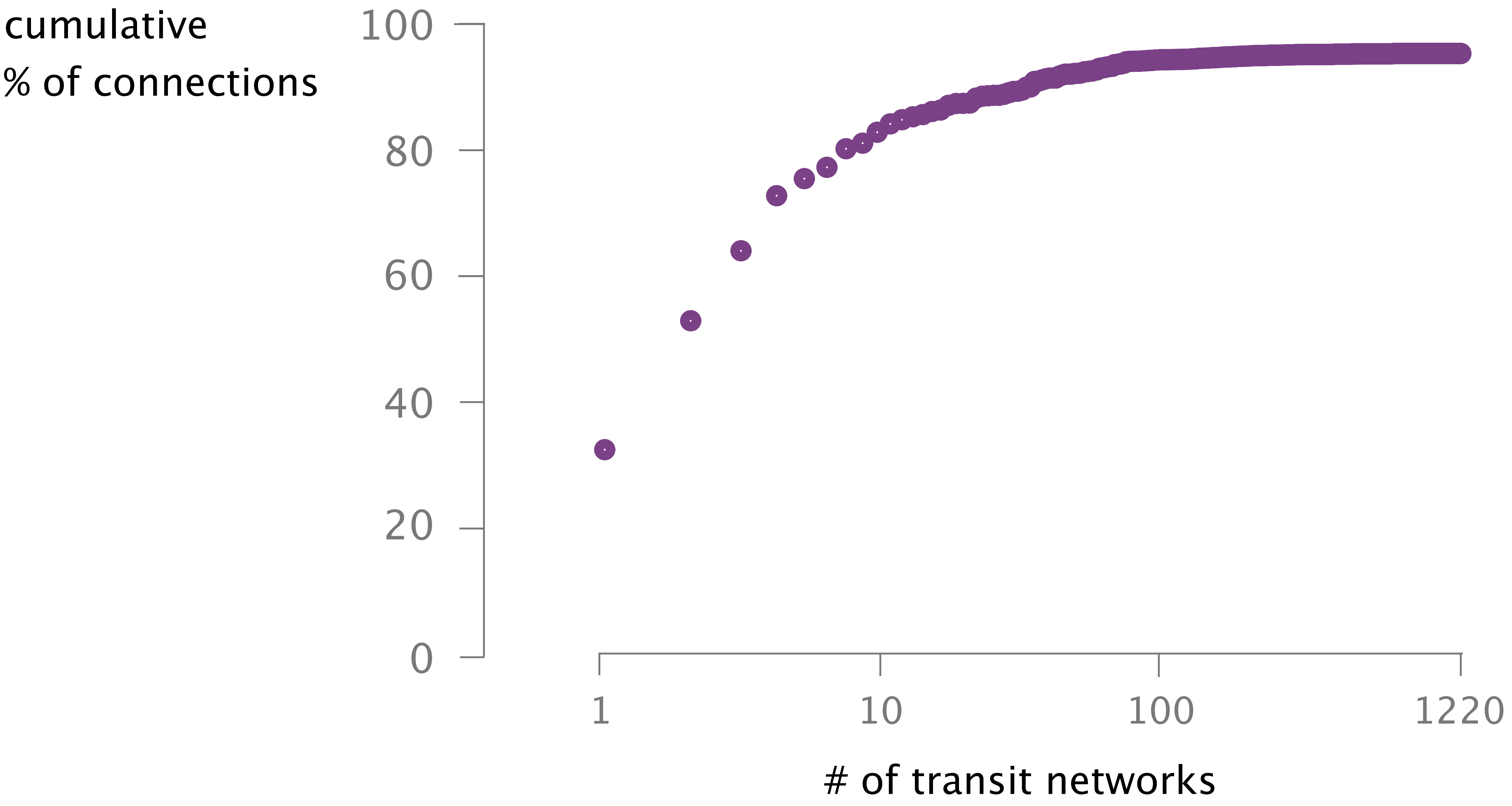

cumulative % of
Bitcoin nodes
100
80
60
40
20
0
1
10
100
1220
# of hosting networks

# **13** networks host **30% of all the nodes**

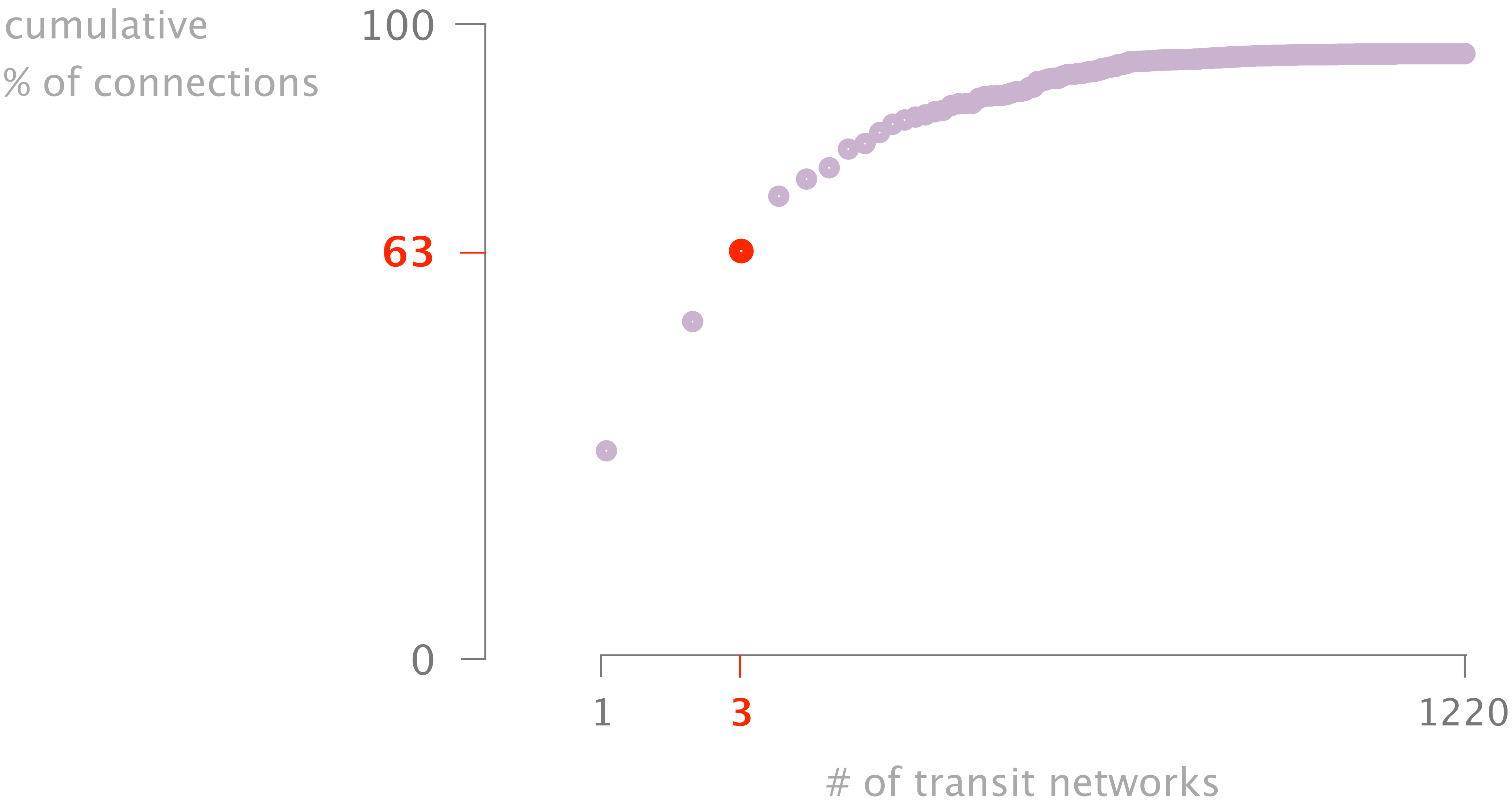



cumulative
% of connections
100
80
60
40
20
0
1
10
100
1220
# of transit networks

# Likewise, a few transit networks can intercept a large fraction of the Bitcoin connections



# 3 transit networks see more than 60% of all connections



# Because of these characteristics two routing attacks practical and effective today

Attack 1

Partitioning

Split the network in half

Attack 2

Delay

Delay block propagation

# Each attack differs in terms of its visibility, impact, and targets

Attack 1

Partitioning

visible

network-wide attack

Attack 2

Delay

invisible

targeted attack (set of nodes)

# Each attack differs in terms of its visibility, impact, and targets

| Attack 1 | Attack 2 |
|---|---|
| Partitioning | Delay |
| visible | invisible |
| network-wide attack | targeted attack (set of nodes) |

# Hijacking Bitcoin

## Routing Attacks on Cryptocurrencies

1 **Background**

BGP & Bitcoin

2 **Partitioning attack**

splitting the network

3 **Delay attack**

slowing the network down

4 **Countermeasures**

short-term & long-term

# Hijacking Bitcoin

## Routing Attacks on Cryptocurrencies

1 Background

BGP & Bitcoin

Partitioning attack

splitting the network

Delay attack

slowing the network down

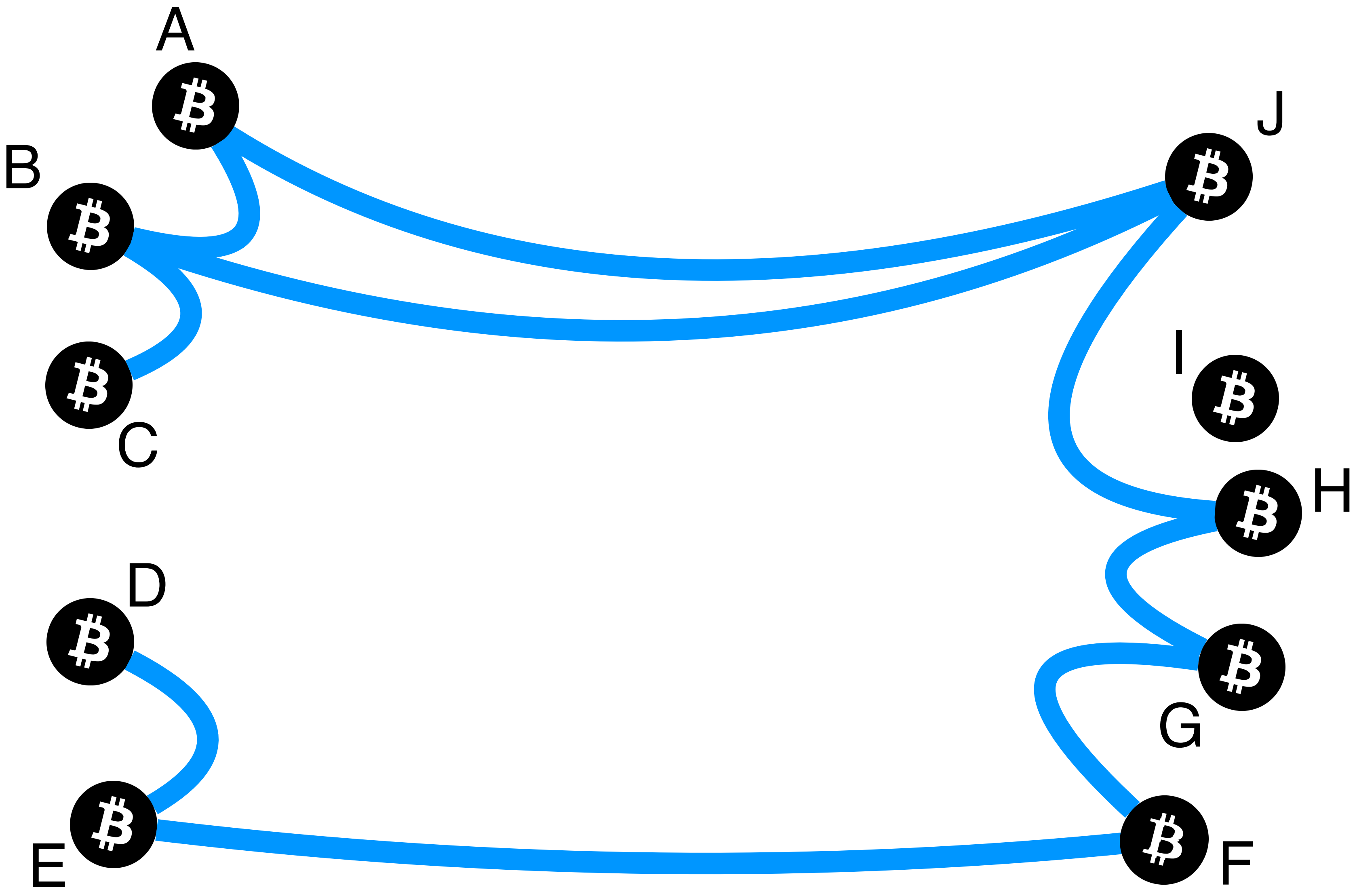
Countermeasures

short-term & long-term

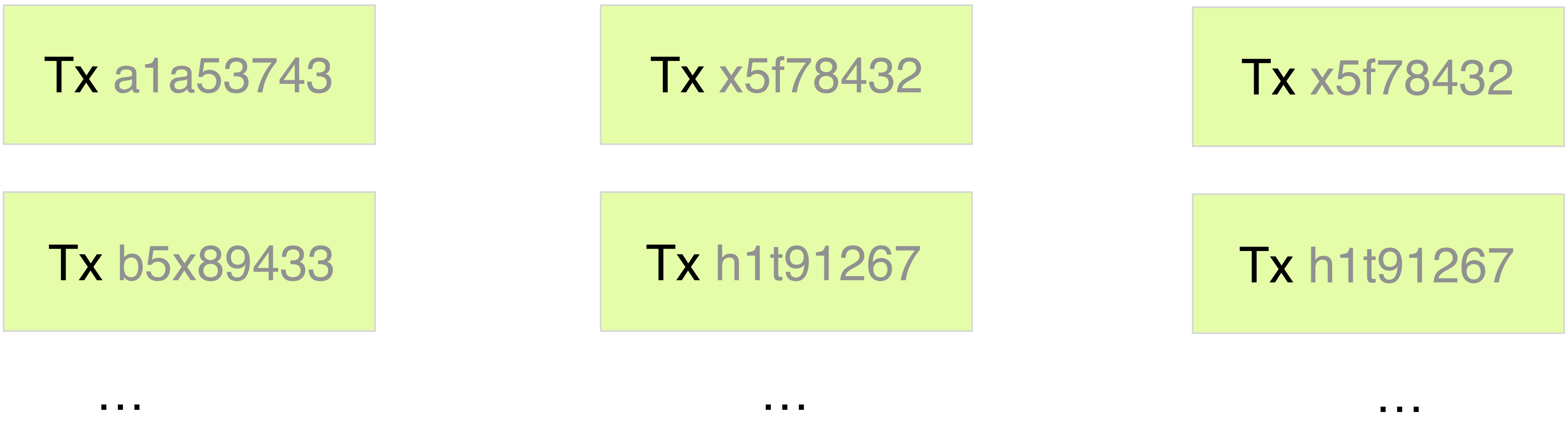
# Bitcoin is a distributed network of nodes

A
B
C
D
E
J
I
H
G
F

# Bitcoin nodes establish random connections between each other



Each node keeps a ledger of all transactions ever performed: “the blockchain”



Tx a1a53743

Tx b5x89433

…

Tx x5f78432

Tx h1t91267

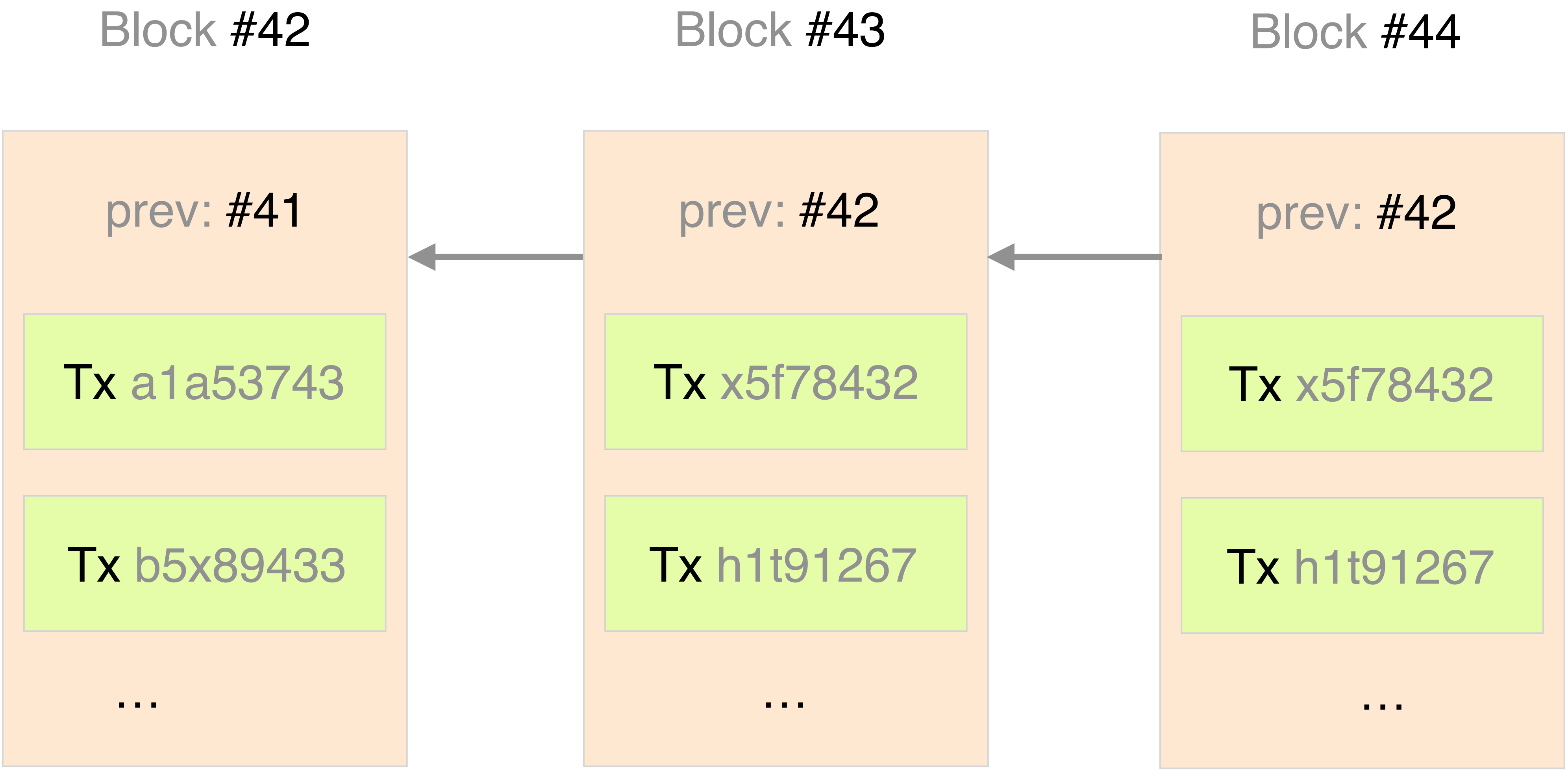
…

Tx x5f78432

Tx h1t91267

…

# The Blockchain is a chain of Blocks



Block #42

prev: #41

Tx a1a53743

Tx b5x89433

…

Block #43

prev: #42

Tx x5f78432

Tx h1t91267

…

Block #44

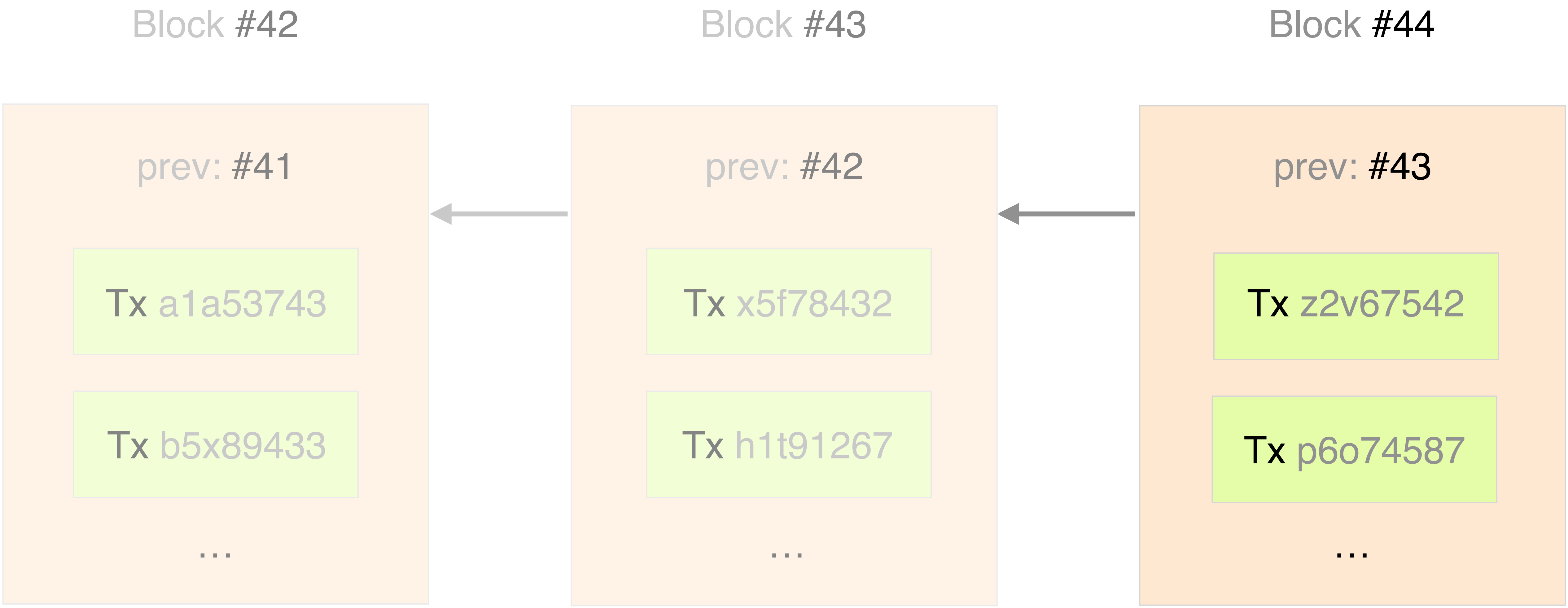
prev: #42

Tx x5f78432

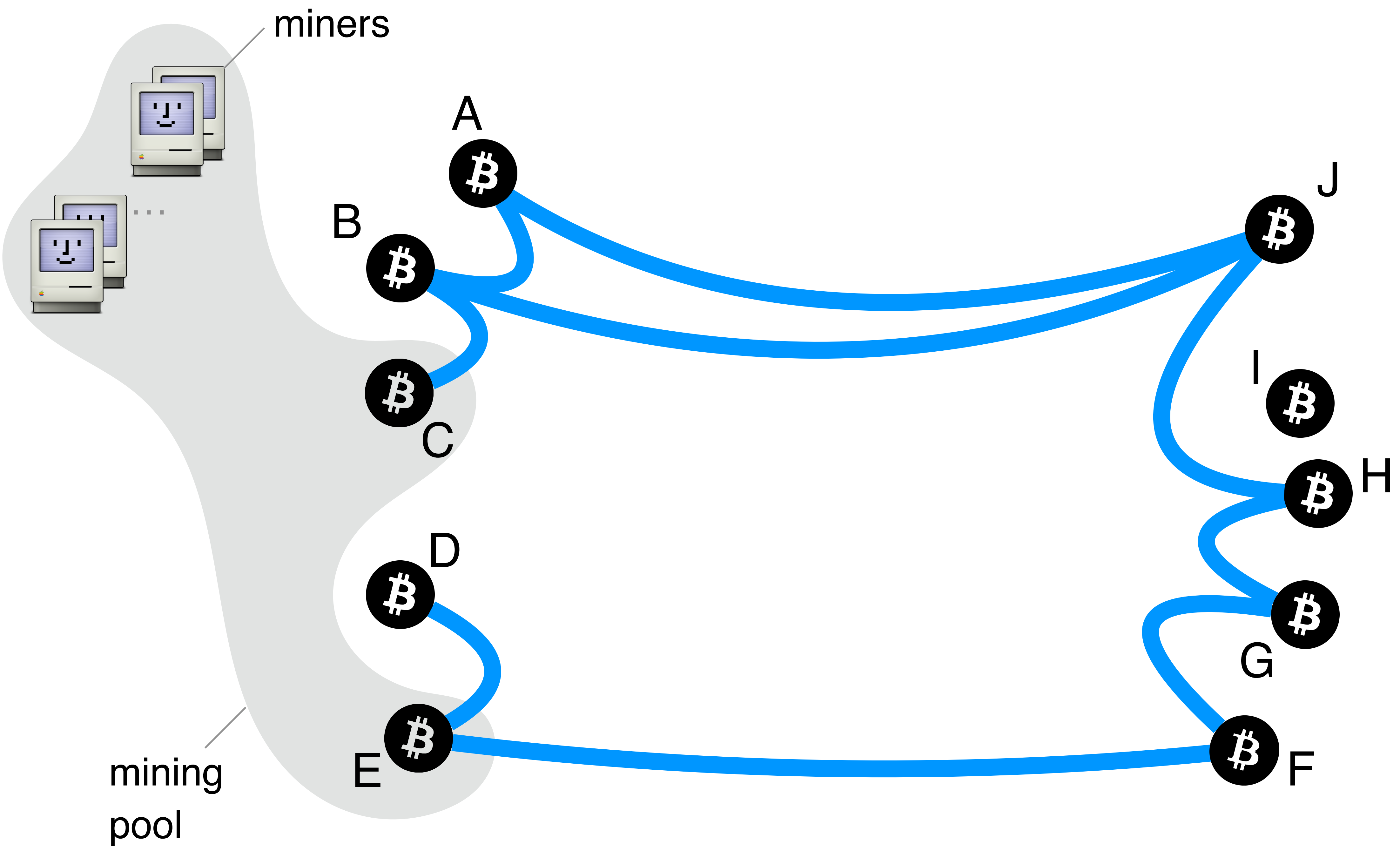
Tx h1t91267

…

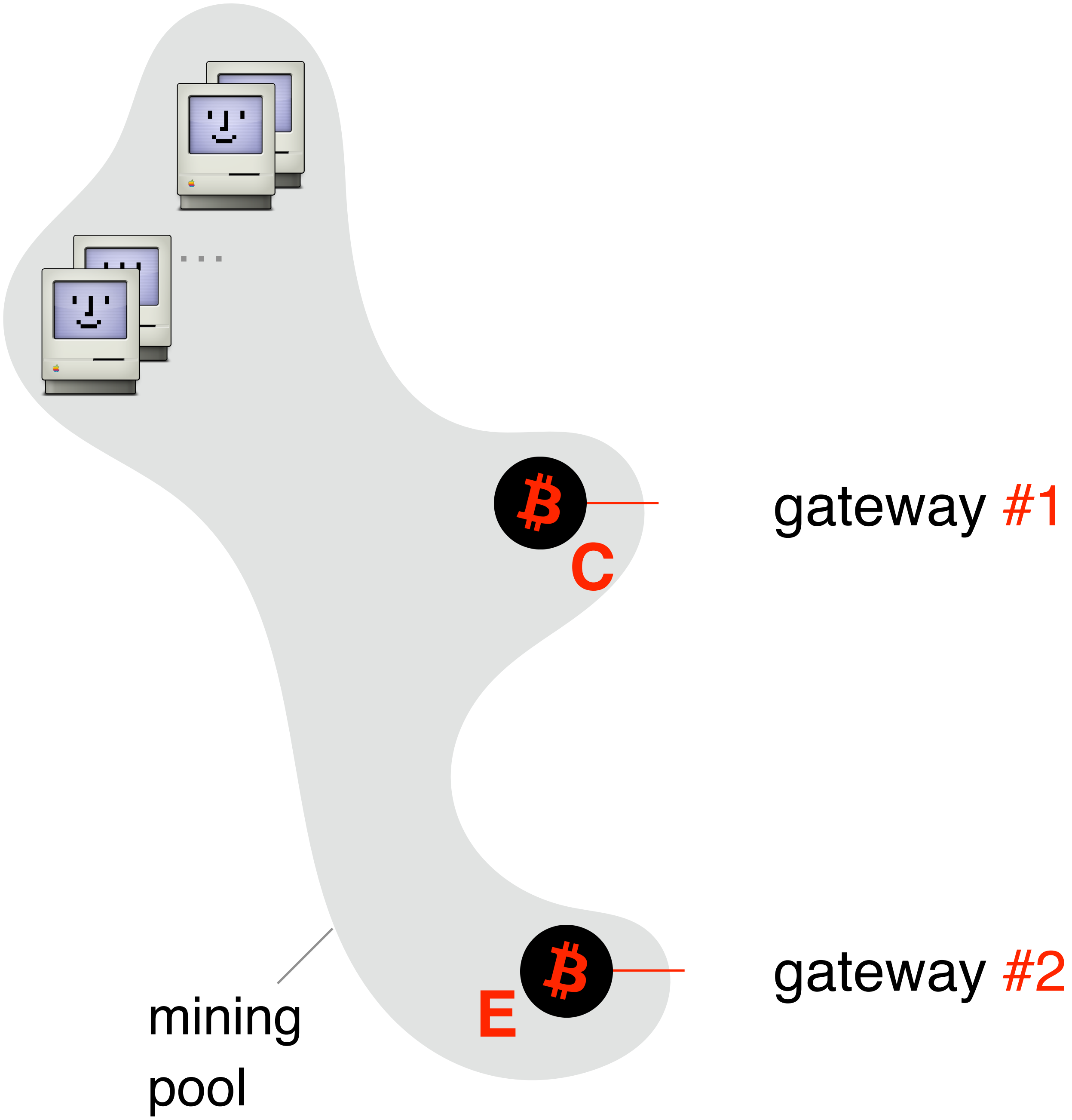
# The Blockchain is extended by miners



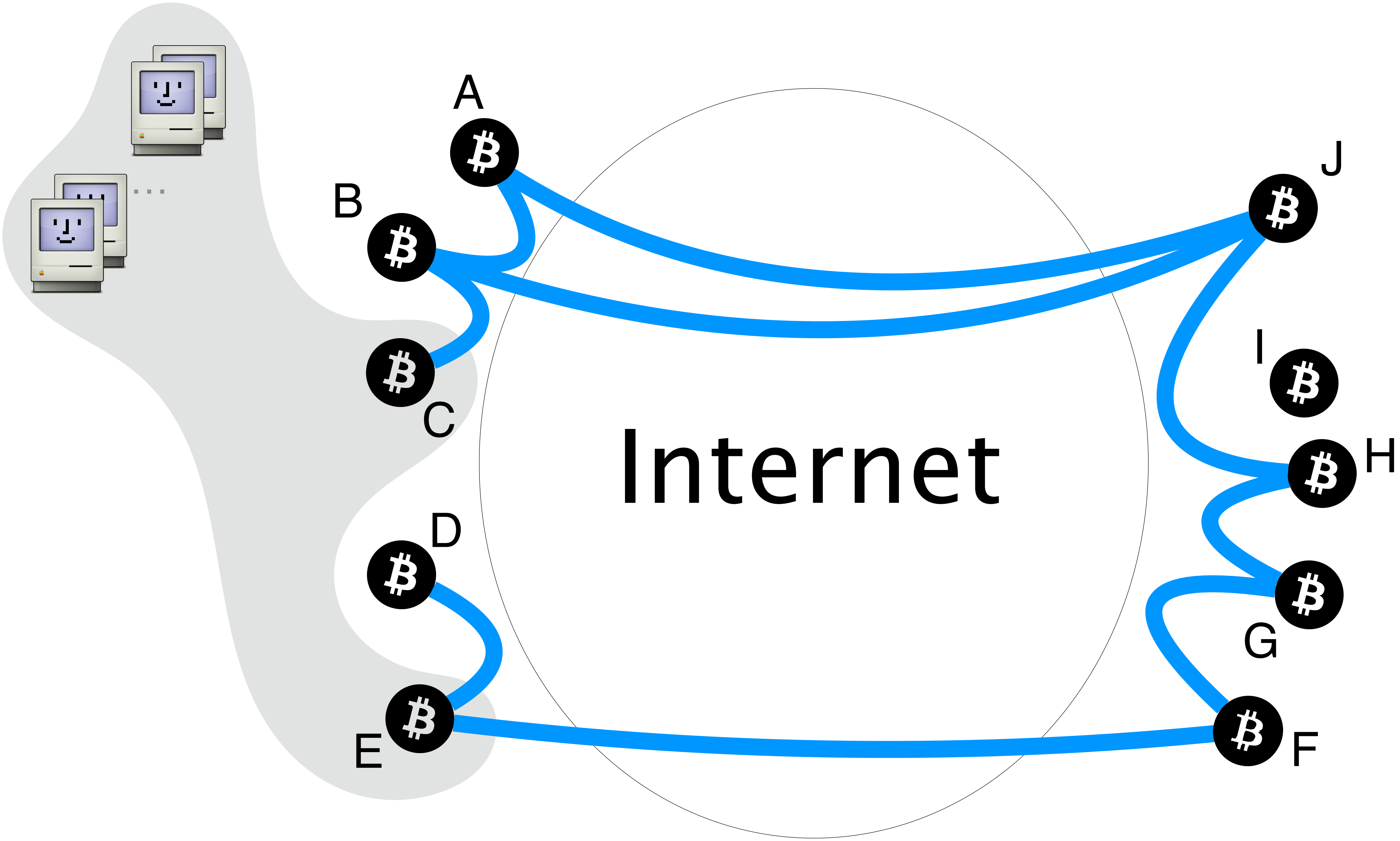
# Miners are grouped in mining pools



# Mining pools connect to the Bitcoin network through multiple gateways

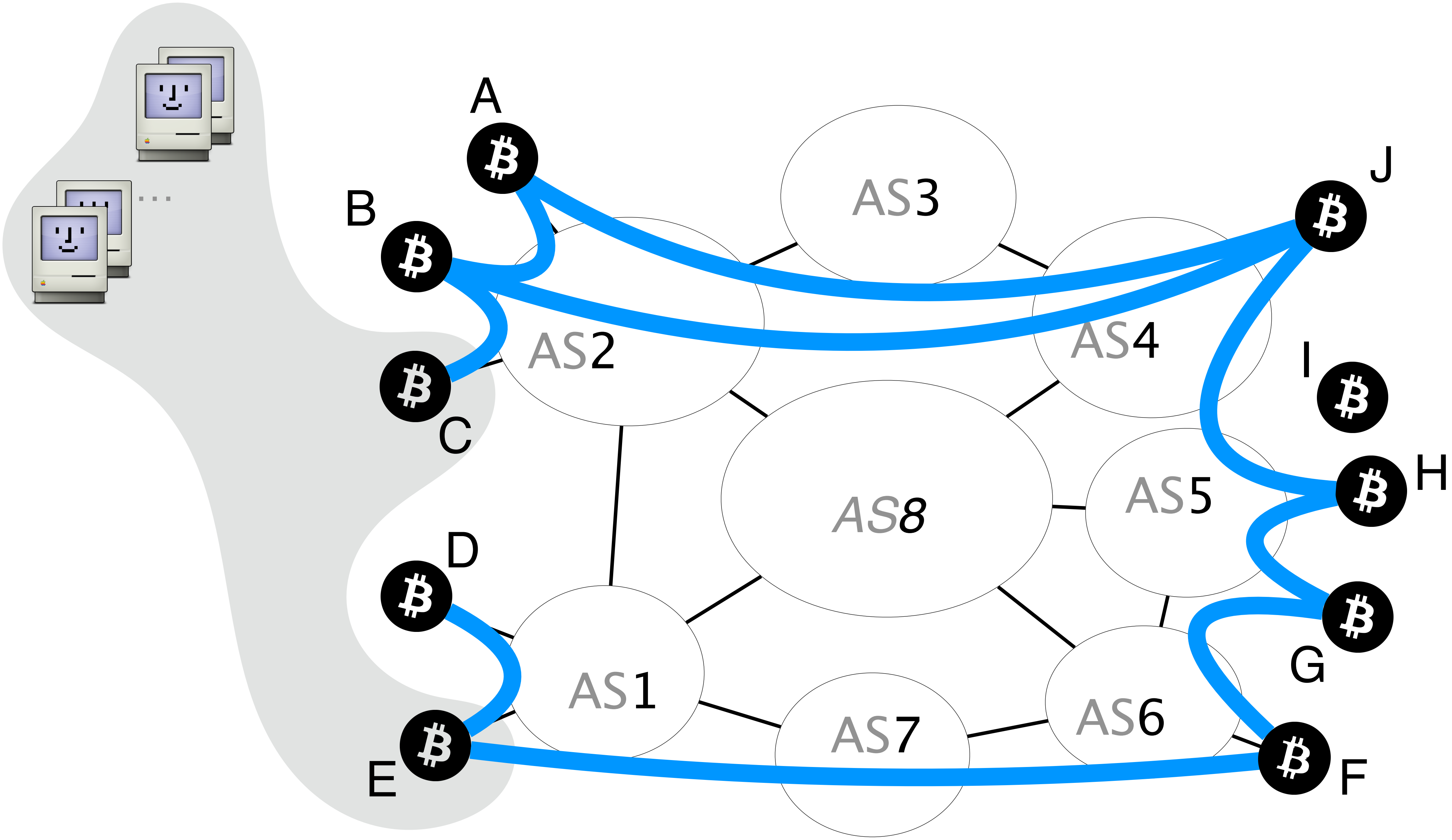


# Bitcoin connections are routed over the Internet

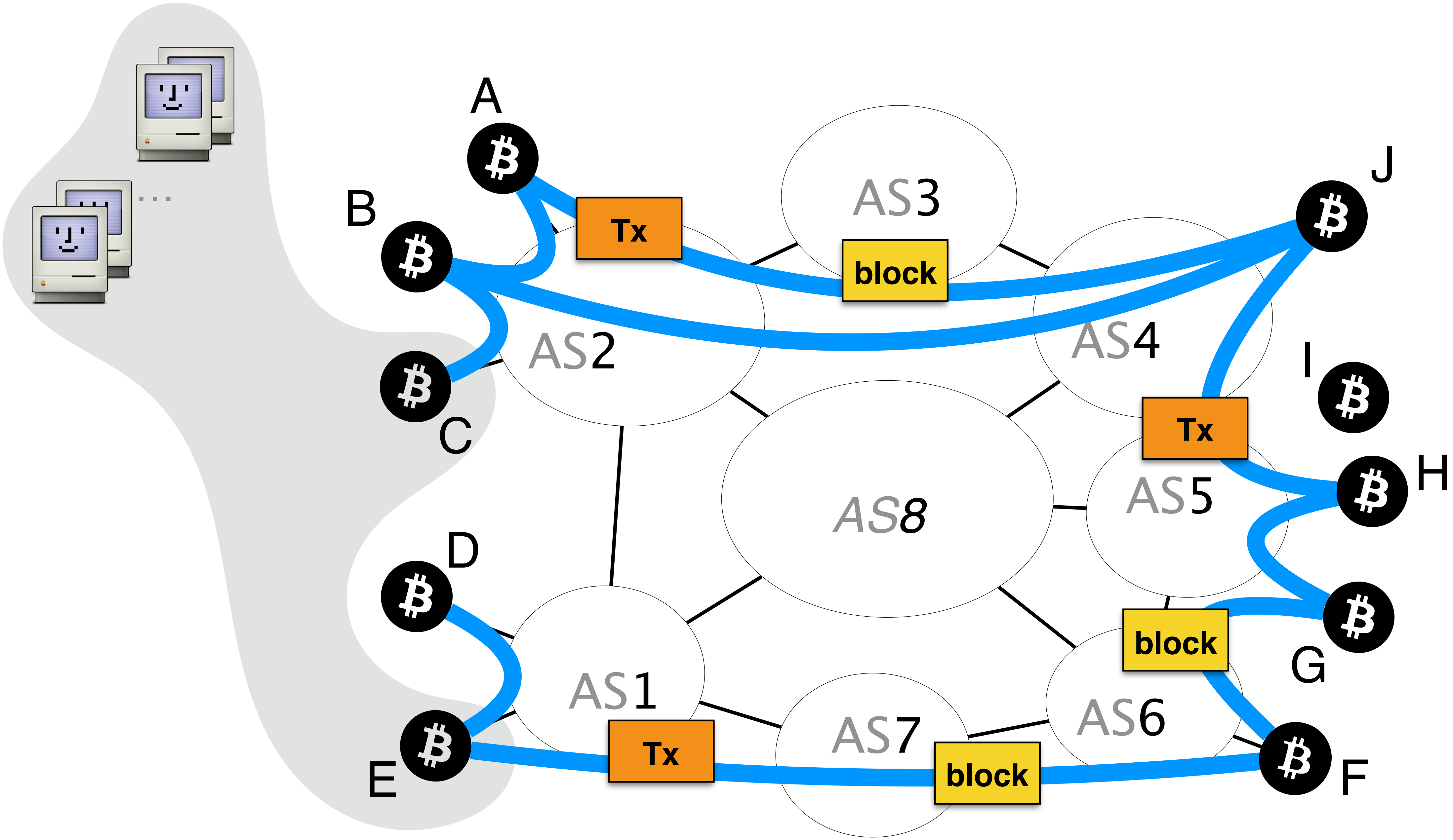


The Internet is composed of Autonomous Systems (ASes).

BGP computes the forwarding path across them



## Bitcoin messages are propagated unencrypted and without any integrity guarantees



# Hijacking Bitcoin

Routing Attacks on Cryptocurrencies

Background

BGP & Bitcoin

2 Partitioning attack

splitting the network

Delay attack

slowing the network down

Countermeasures

short-term & long-term

The goal of a partitioning attack is to split the Bitcoin network into two disjoint components

# The impact of such an attack is worrying

Denial of Service

Revenue Loss

Double spending

# The impact of such an attack is worrying

Denial of Service

Bitcoin clients and wallets cannot secure or propagate transactions

Revenue Loss

Double spending

# The impact of such an attack is worrying

Denial of Service

Revenue Loss

Blocks in component with less mining power are discarded

Double spending

# The impact of such an attack is worrying

Denial of Service

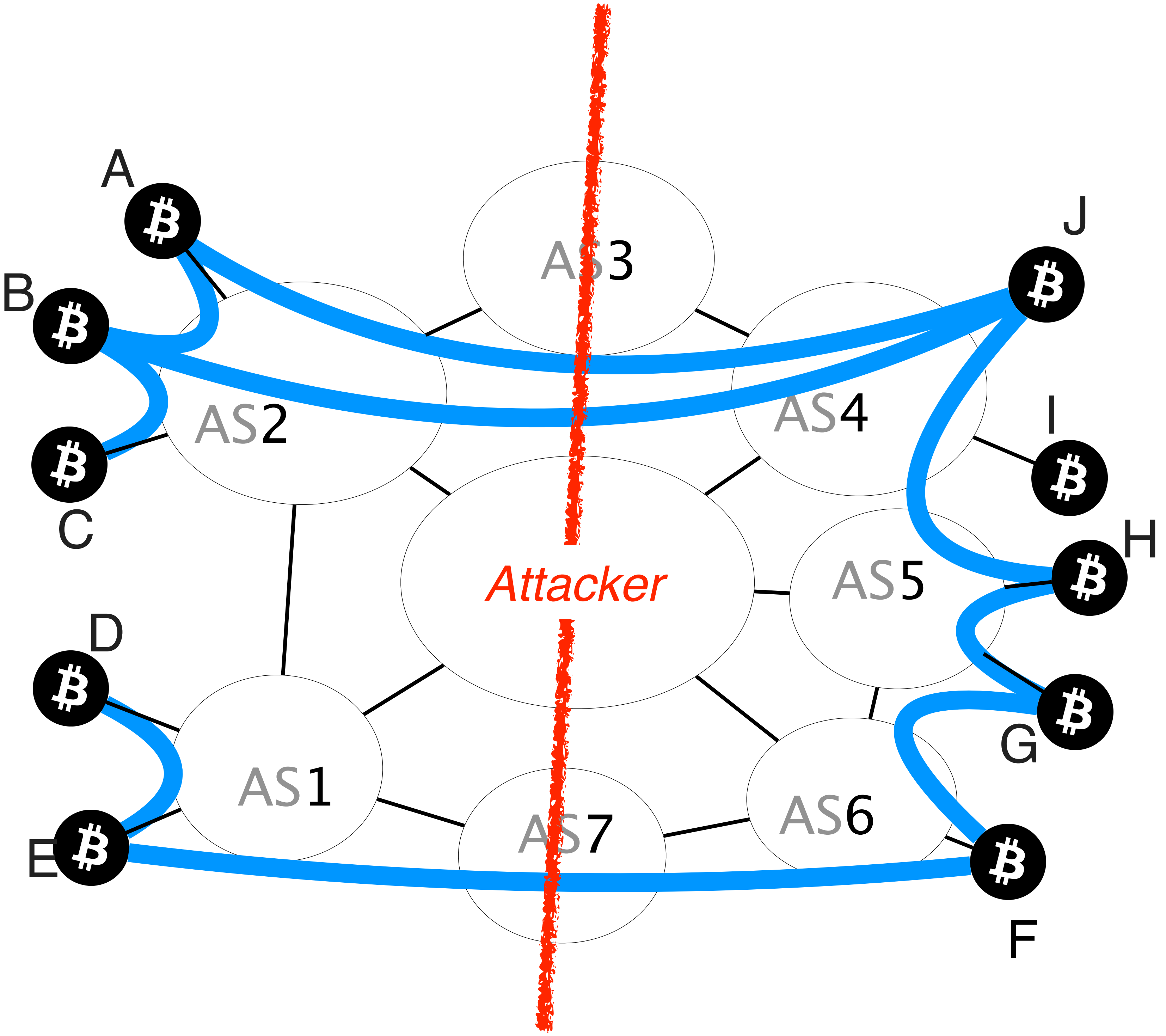
Revenue Loss

Double spending

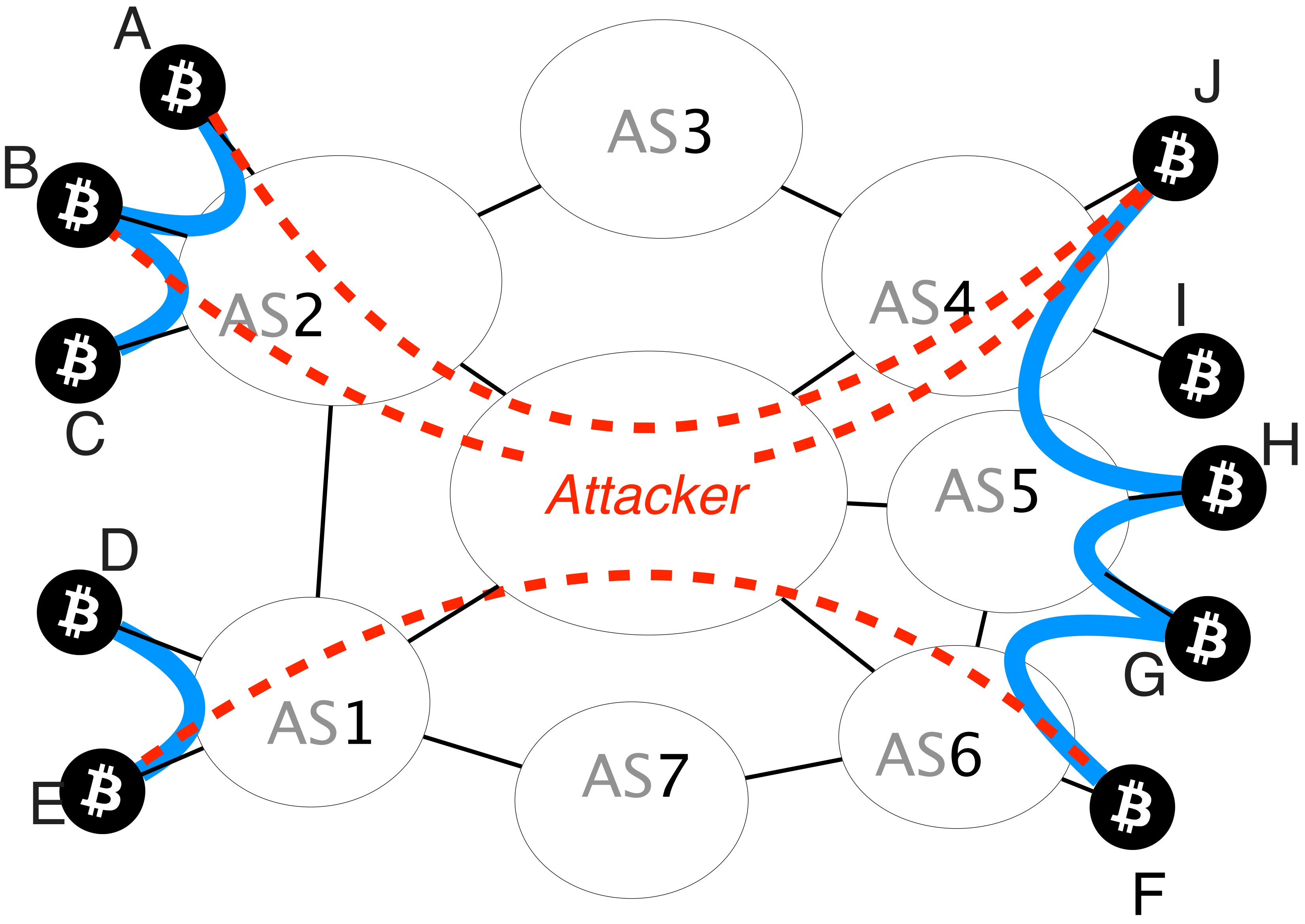
Transactions in components with less mining power can be reverted

## How does the attack work?

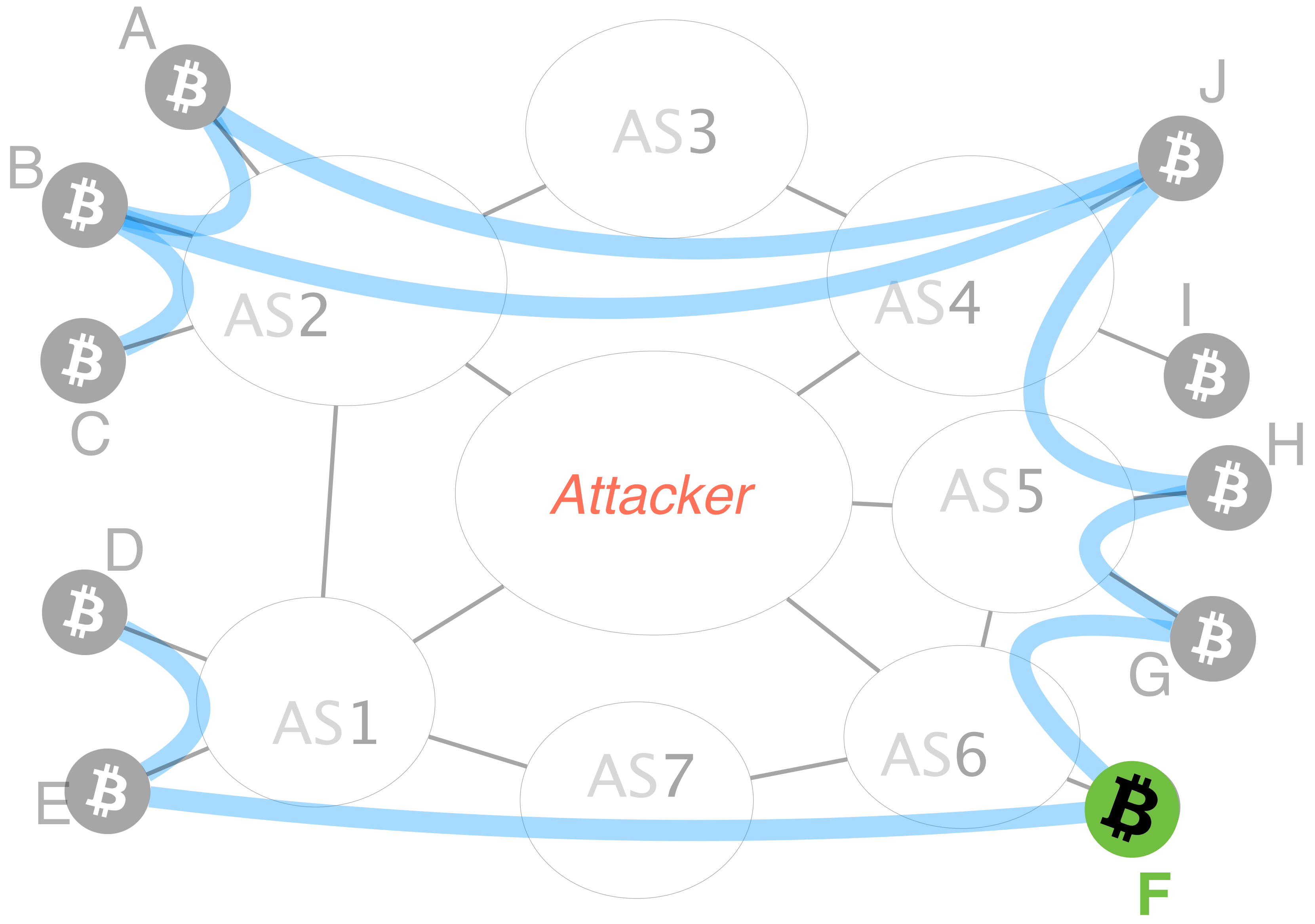
Let’s say an attacker wants to partition the network into the left and right side



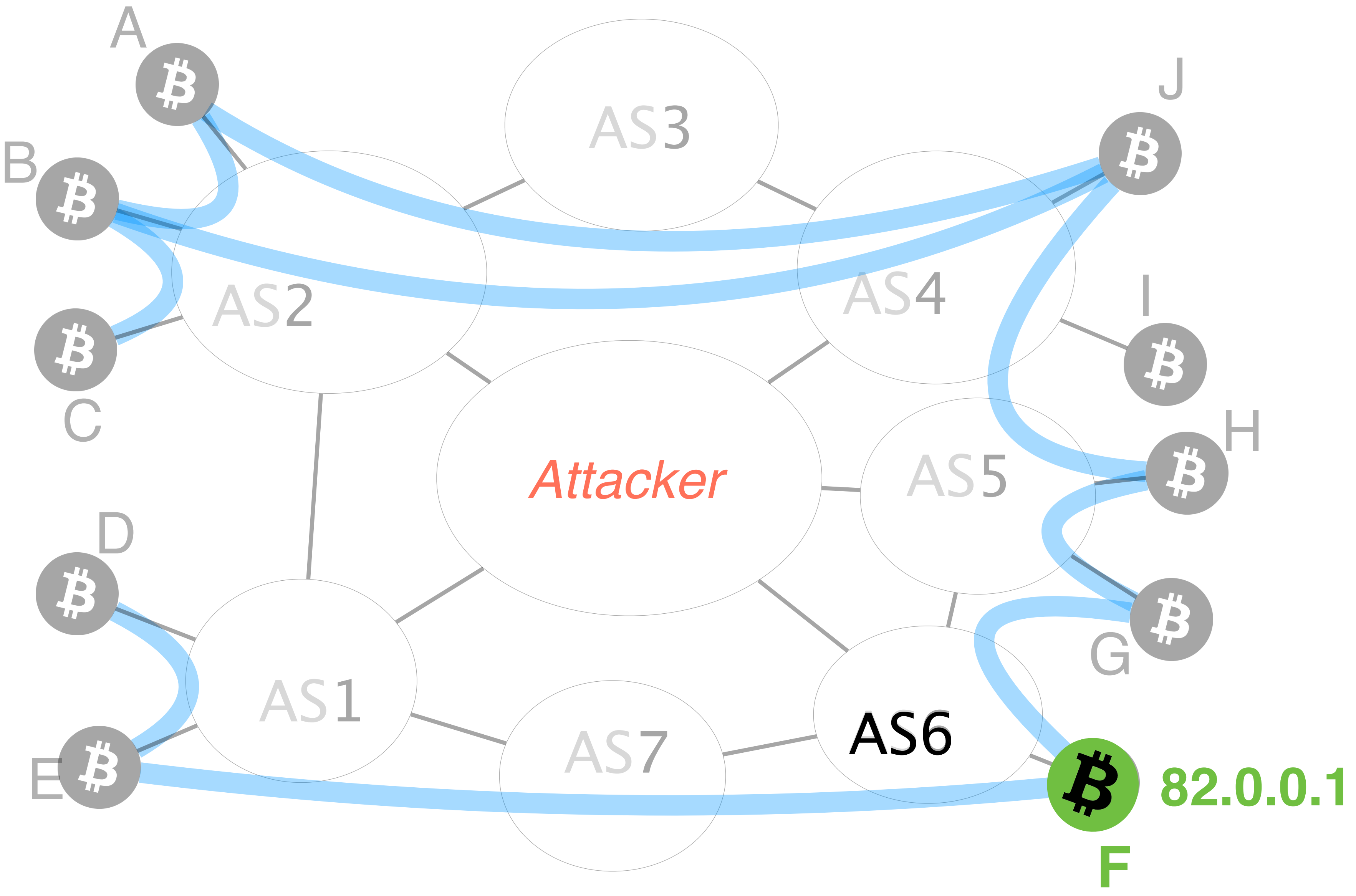
For doing so, the attacker will manipulate BGP routes to intercept any traffic to the nodes in the right



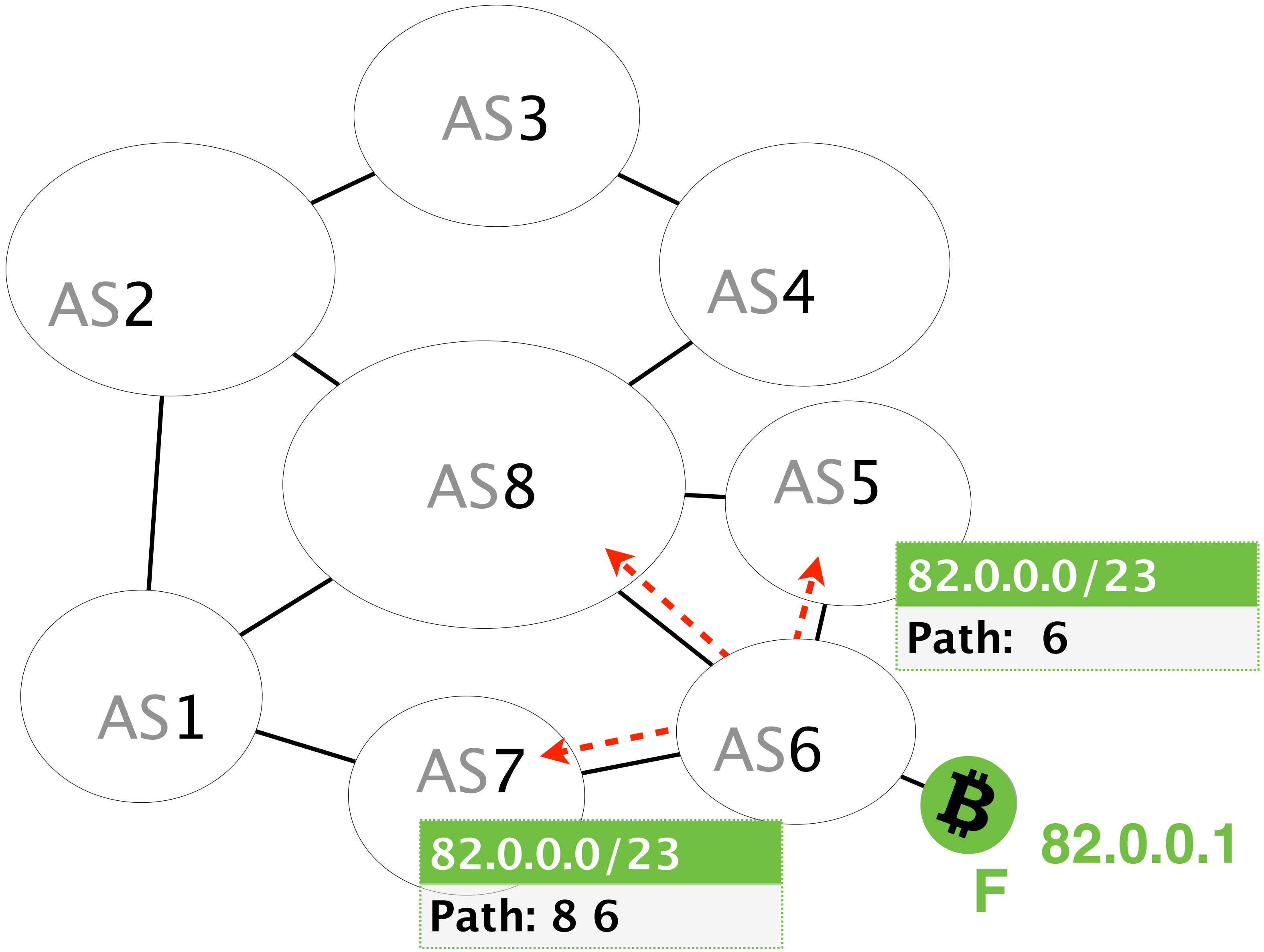
# Let us focus on node F



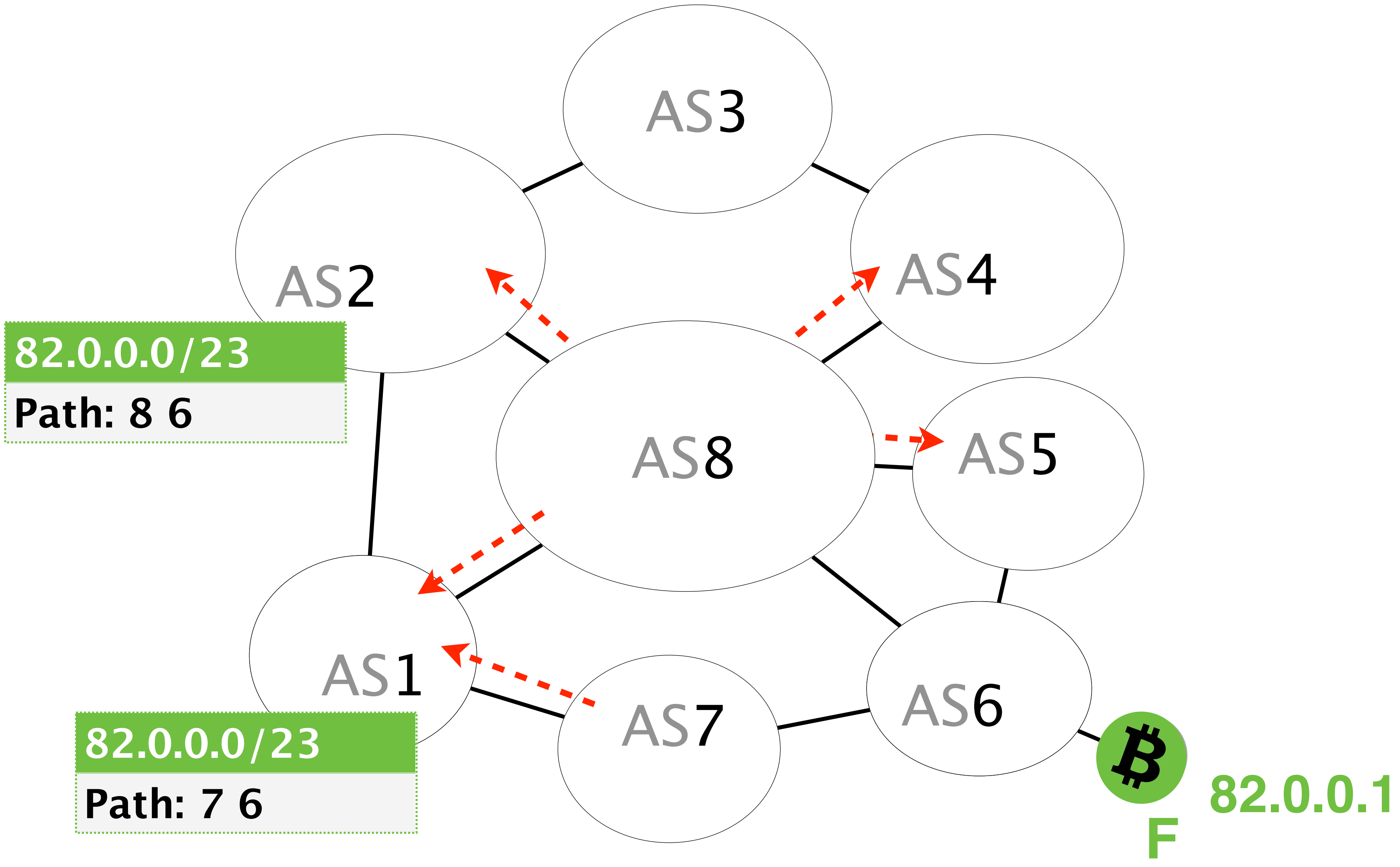
# F’s provider (AS6) is responsible for IP prefix



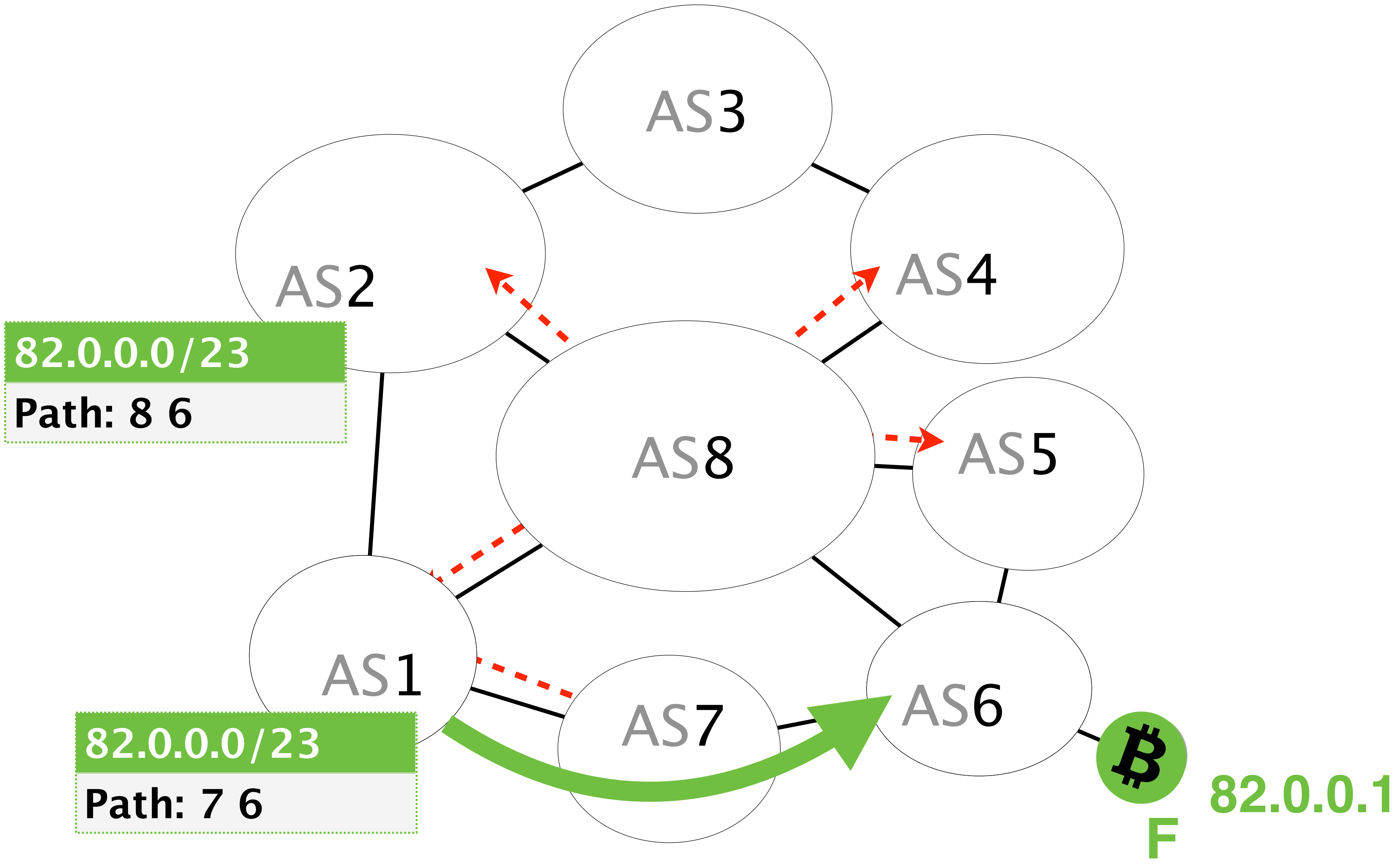
# AS6 will create a BGP advertisement



# AS6's advertisement is propagated AS-by-AS until all ASes in the Internet learn about it

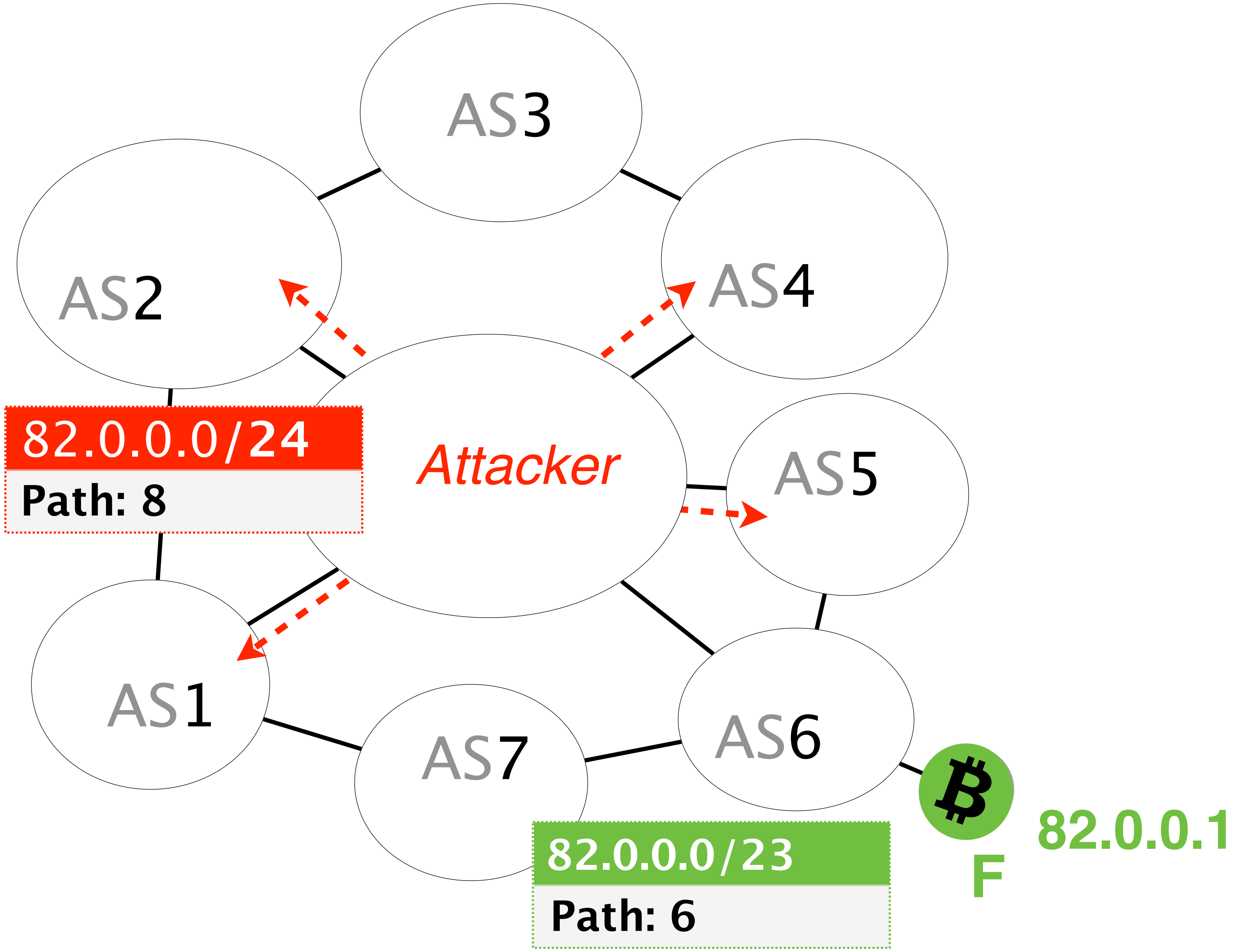


# AS6's advertisement is propagated AS-by-AS until all ASes in the Internet learn about it

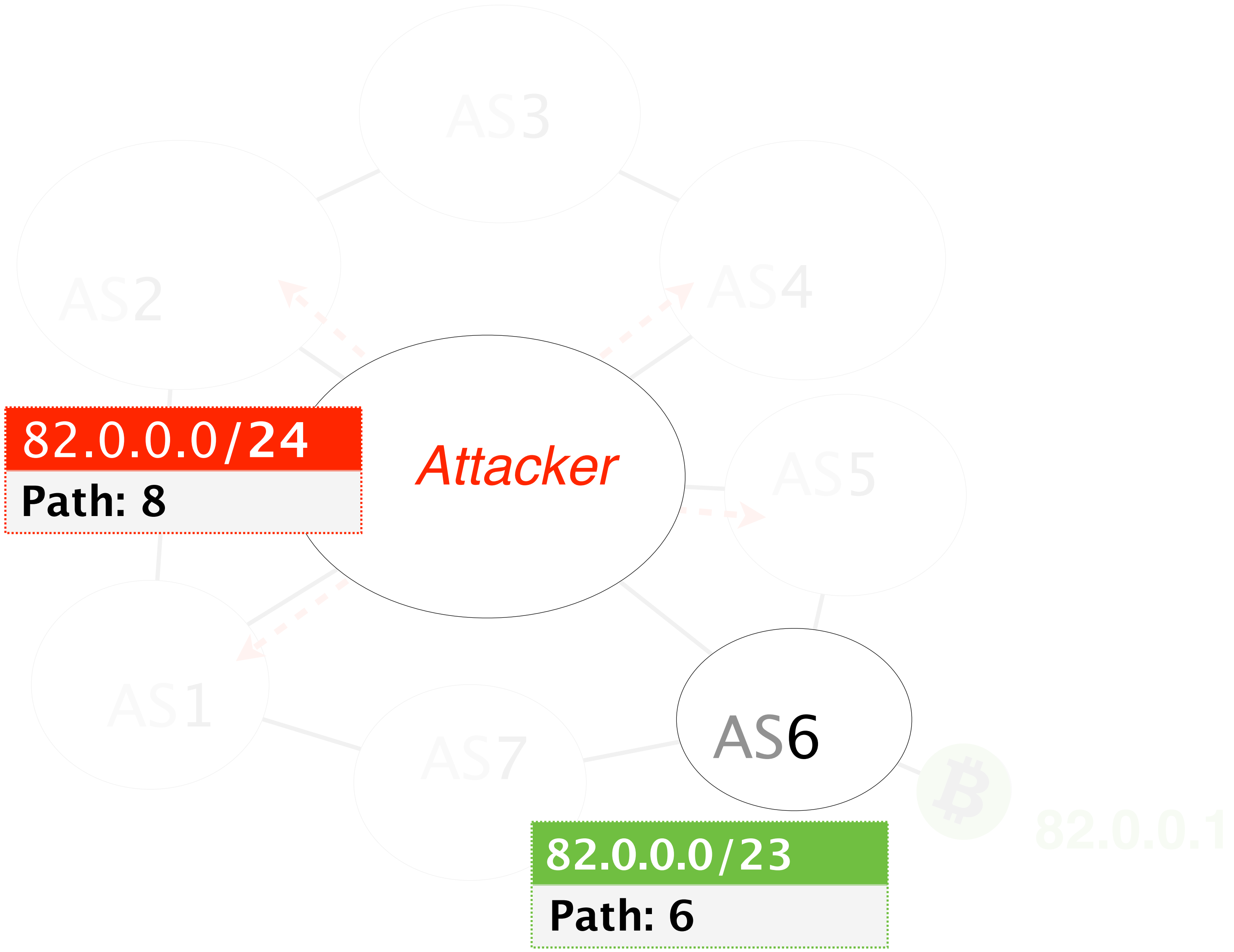


BGP does not check the validity of advertisements, meaning any AS can announce any prefix

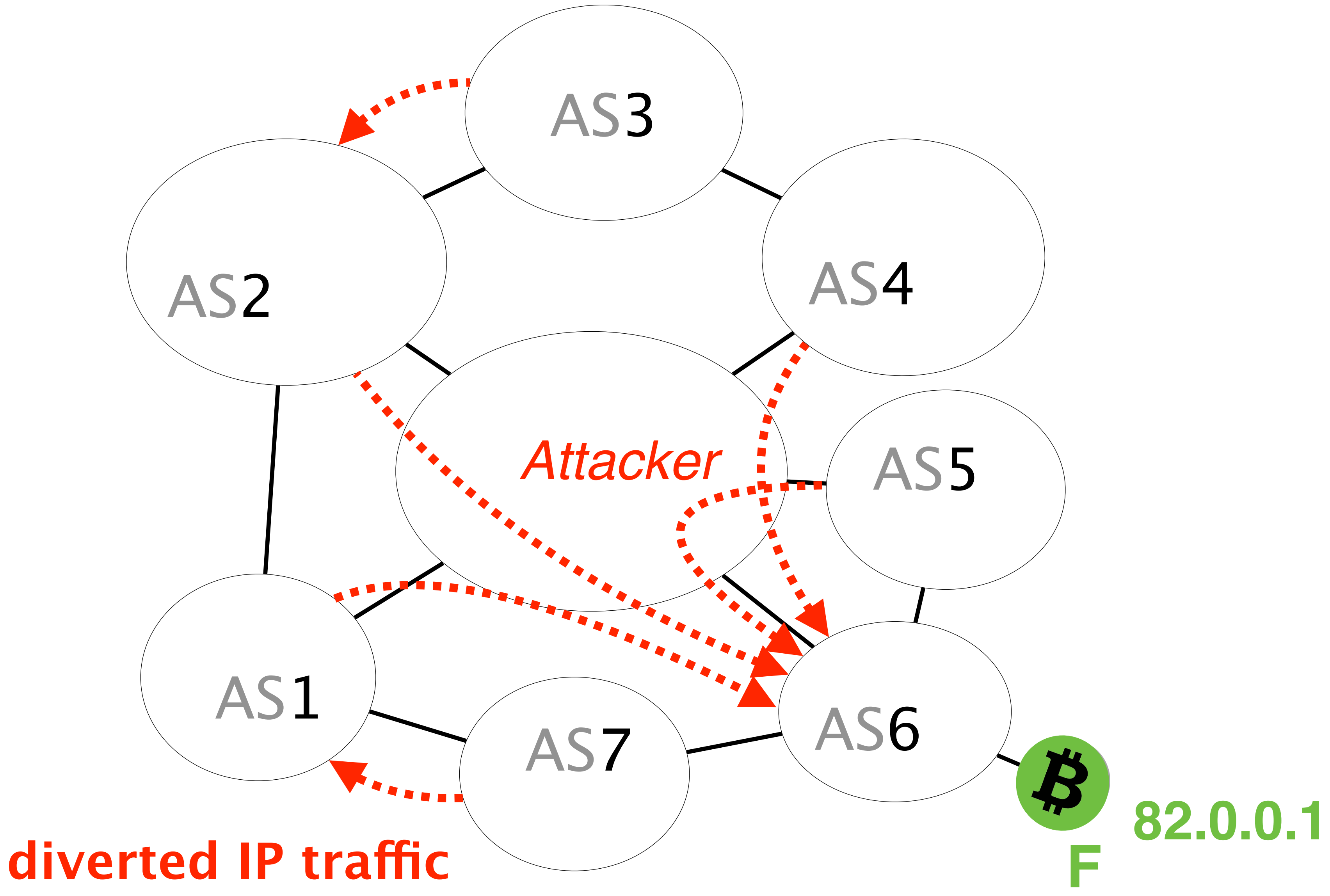
Consider that the attacker advertises a more-specific prefix covering F’s IP address



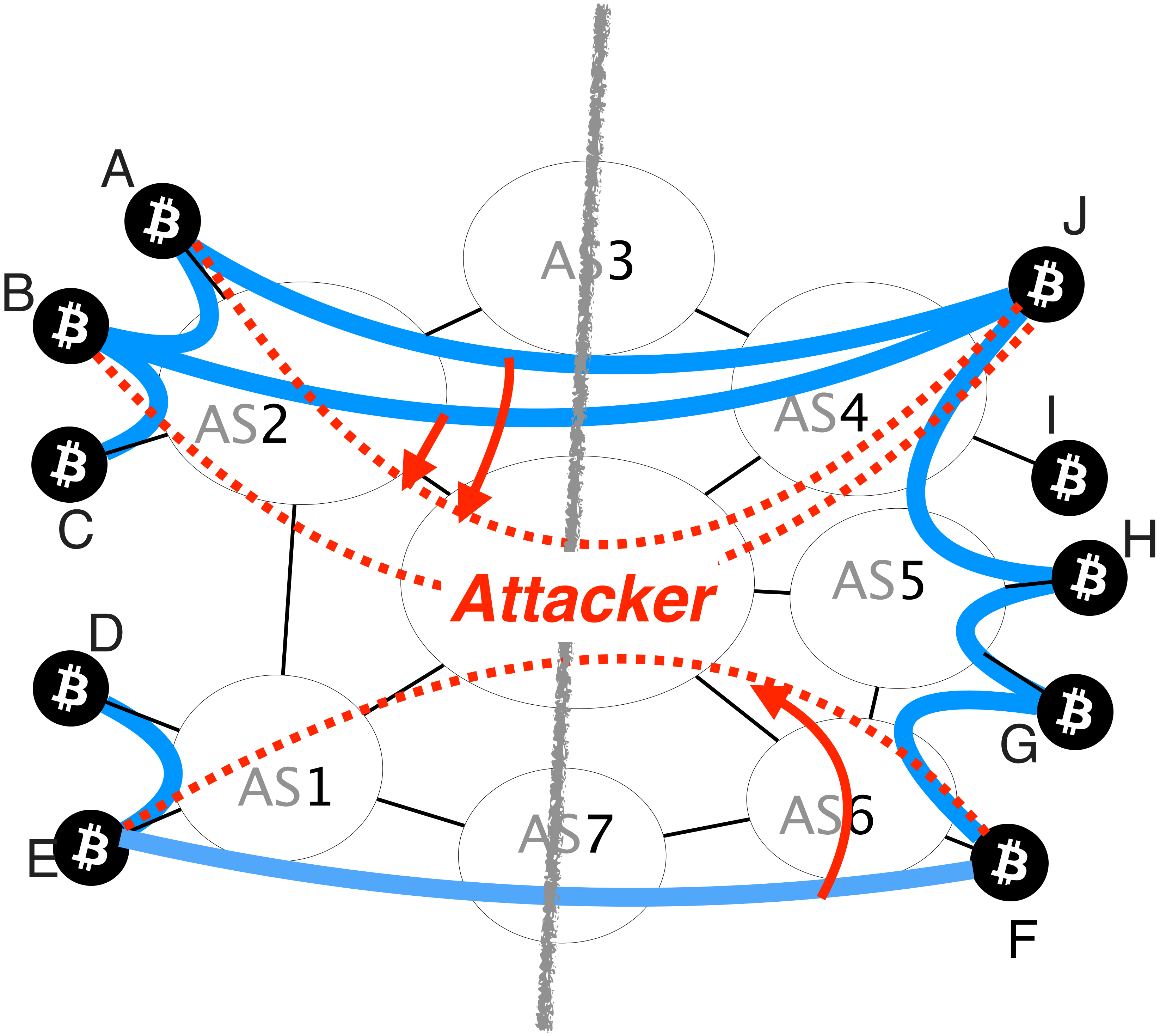
As IP routers prefer more-specific prefixes, the attacker route will be preferred



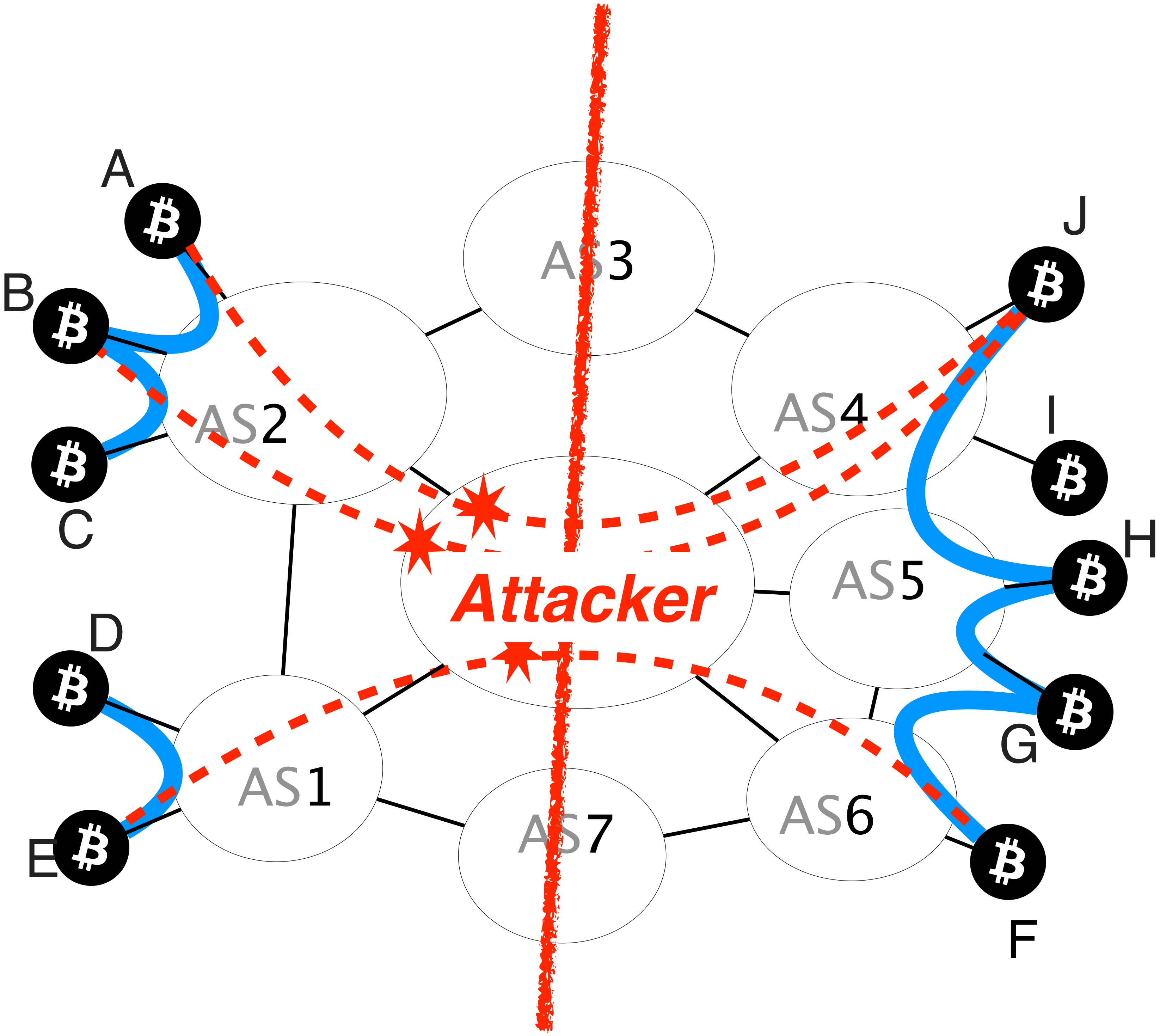
## Traffic to node F is hijacked



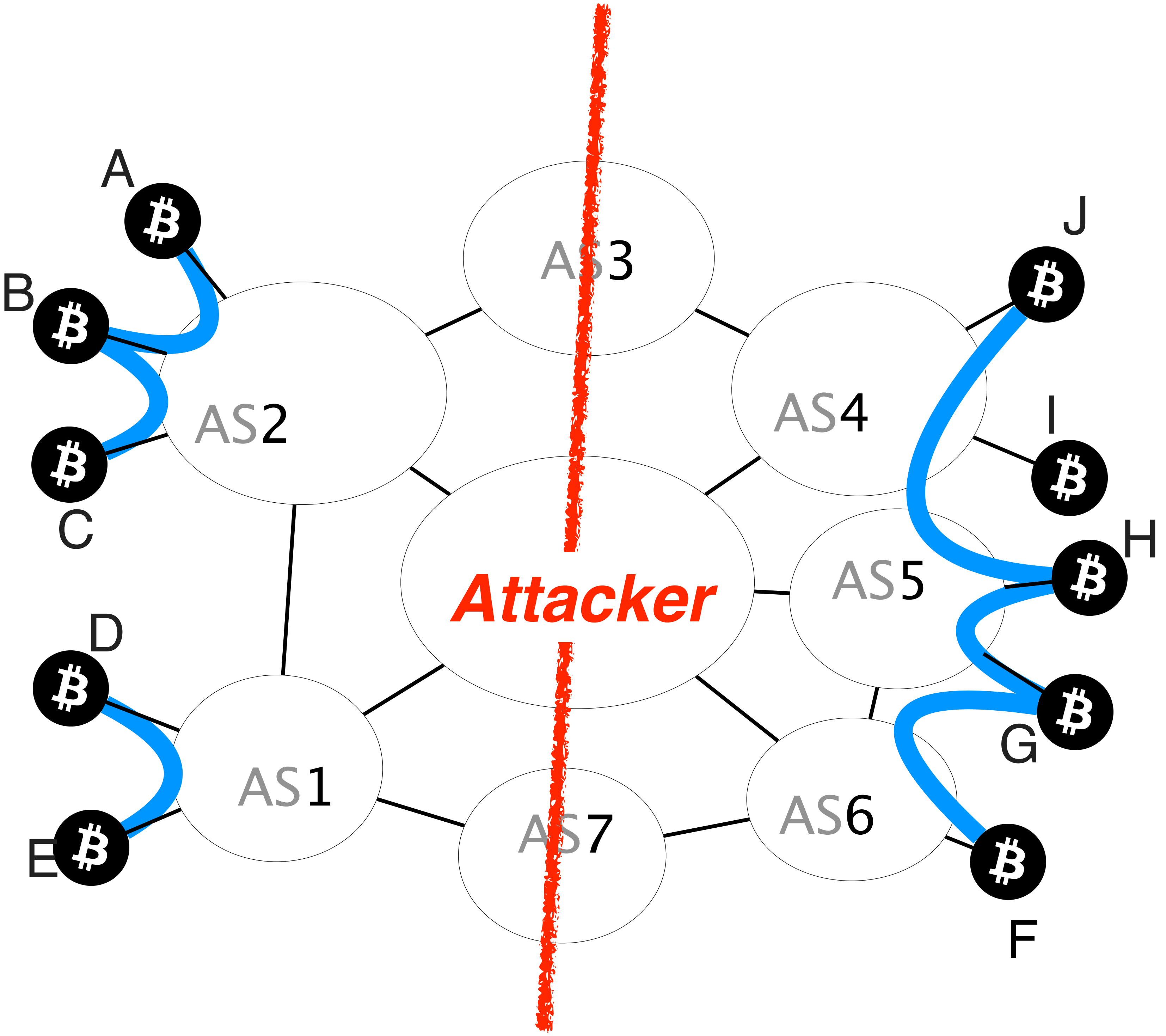
By hijacking the IP prefixes pertaining to the right nodes, the attacker can intercept all their connections



## Once on-path, the attacker can drop all connections crossing the partition



# The partition is created



Not all partition are feasible in practice:

some connections cannot be intercepted

Bitcoin connections established…

- within a mining pool
- within an AS
- between mining pools with private agreements

cannot be hijacked (usually)

Bitcoin connections established…

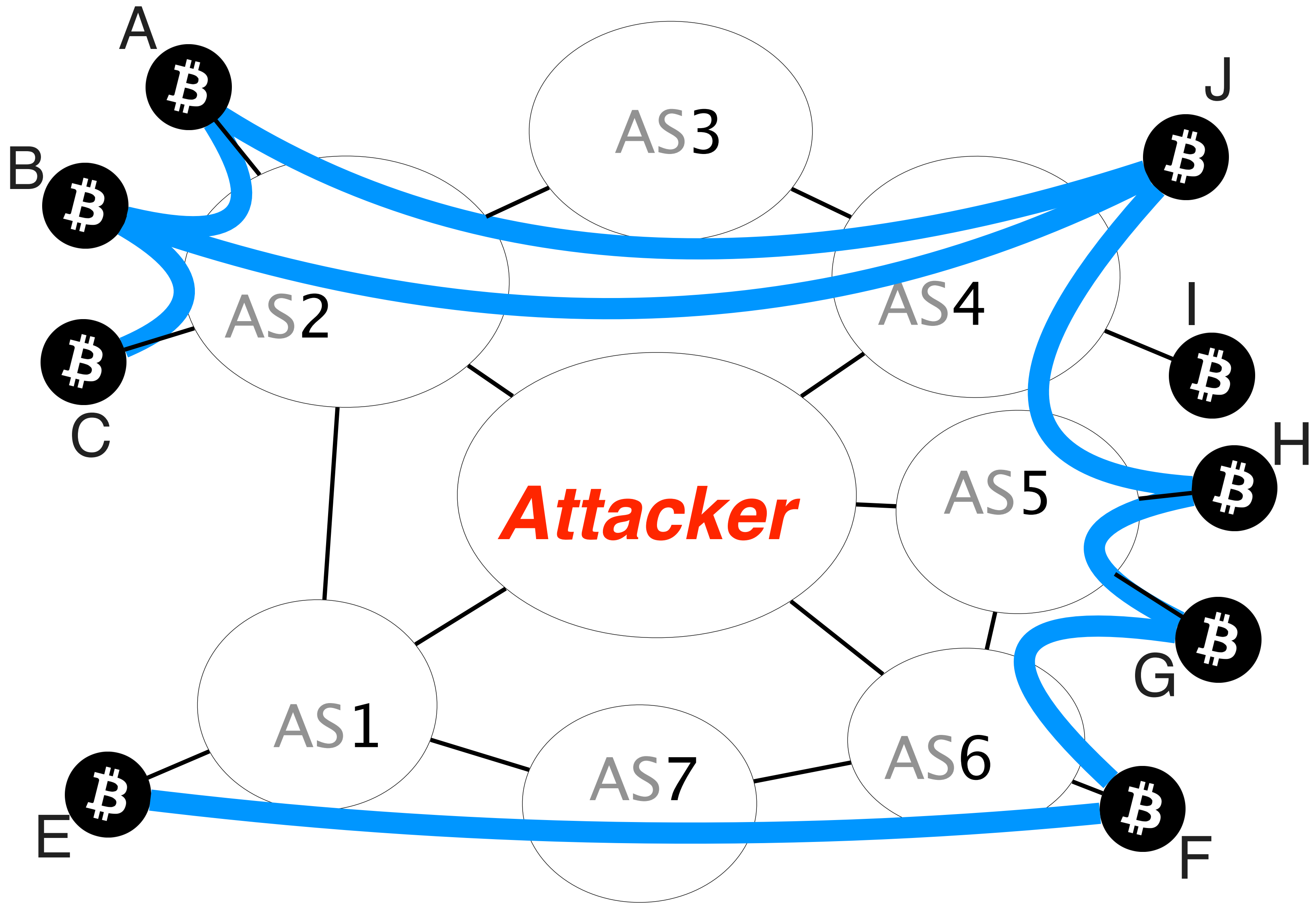
- within a mining pool
- within an AS
- between mining pools

cannot be hijacked (usually)

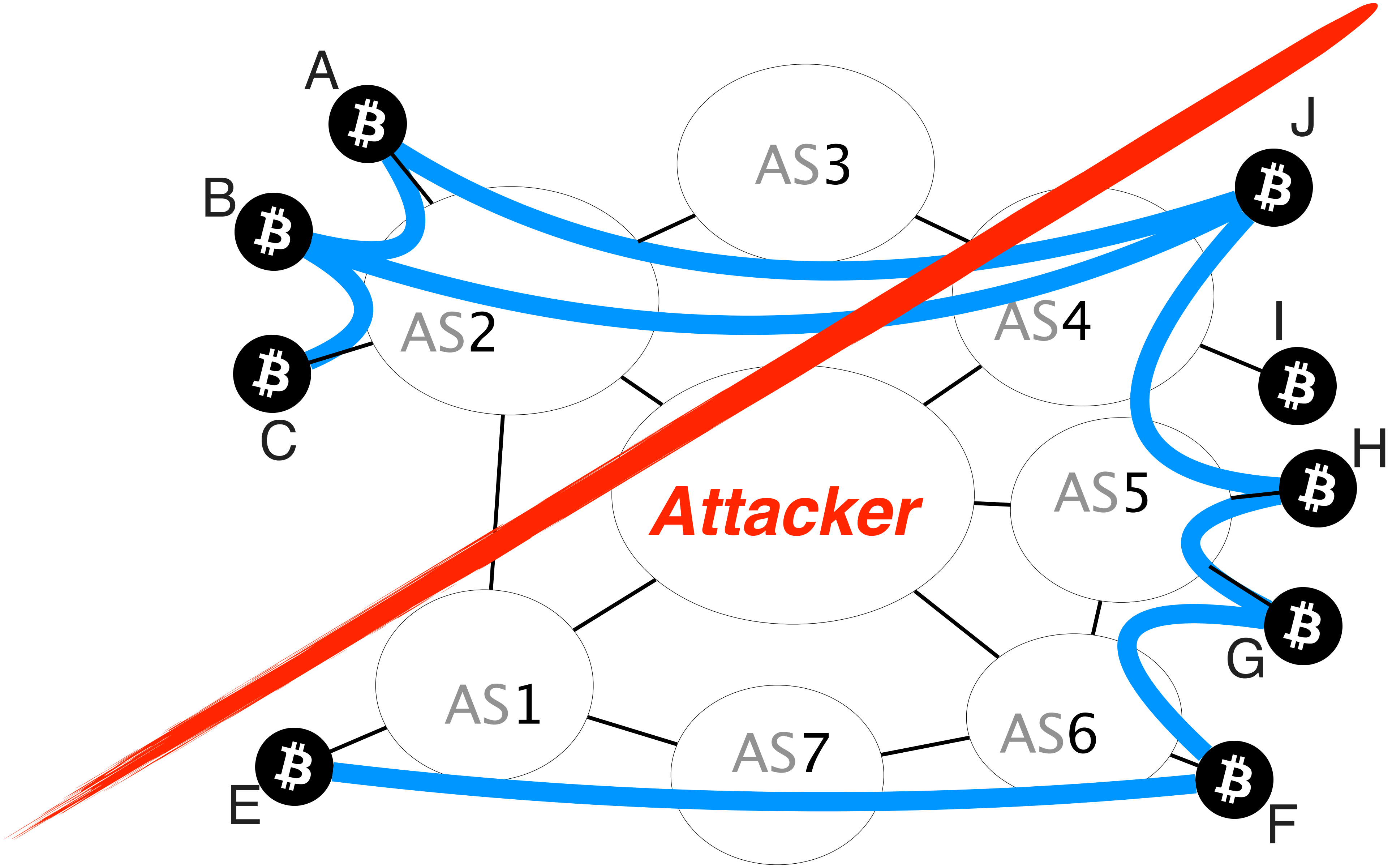
*but* can be *detected* and *located* by the attacker

enabling her to build a similar but feasible partition

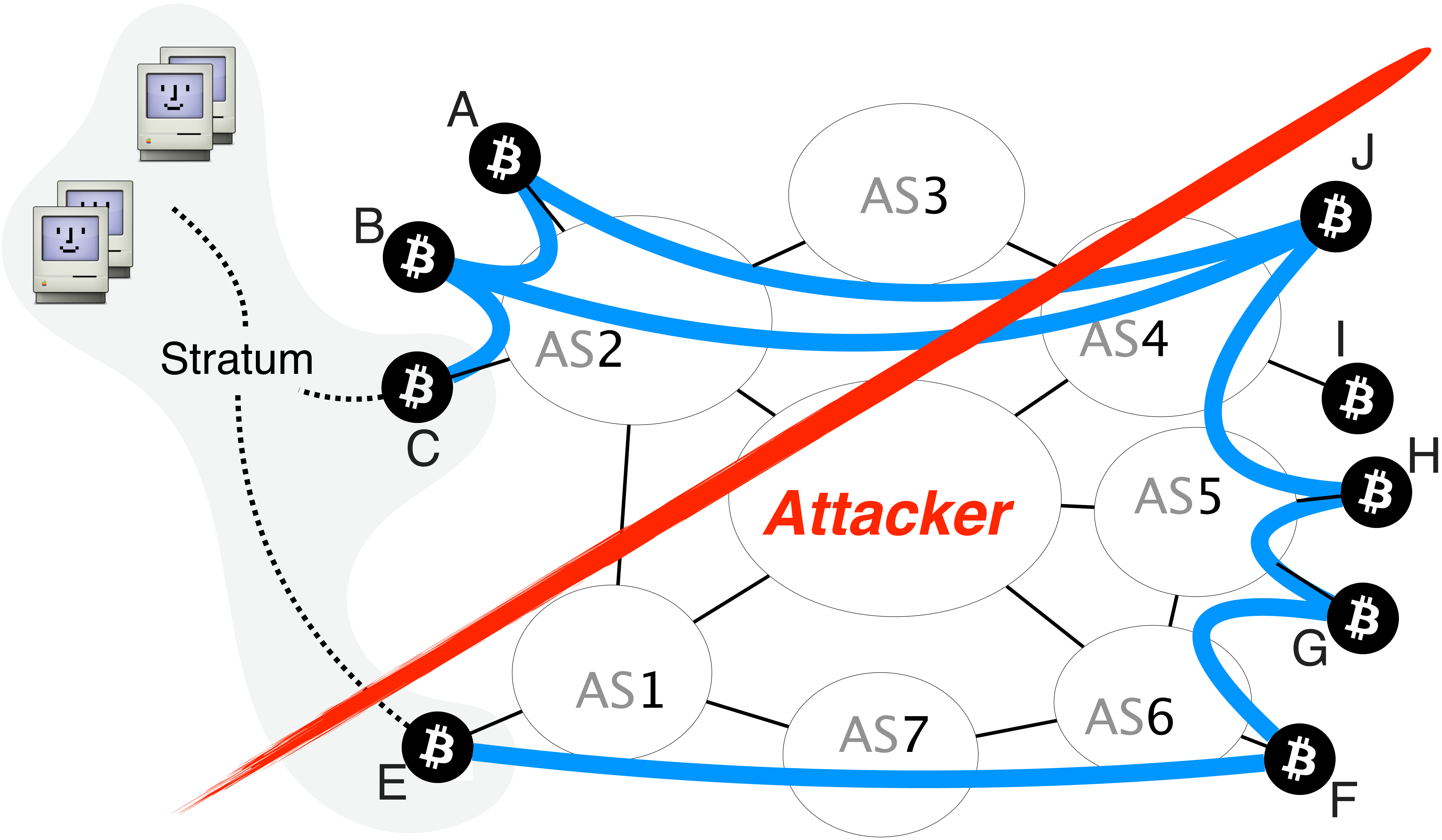
# Same attacker wants to create a different partition



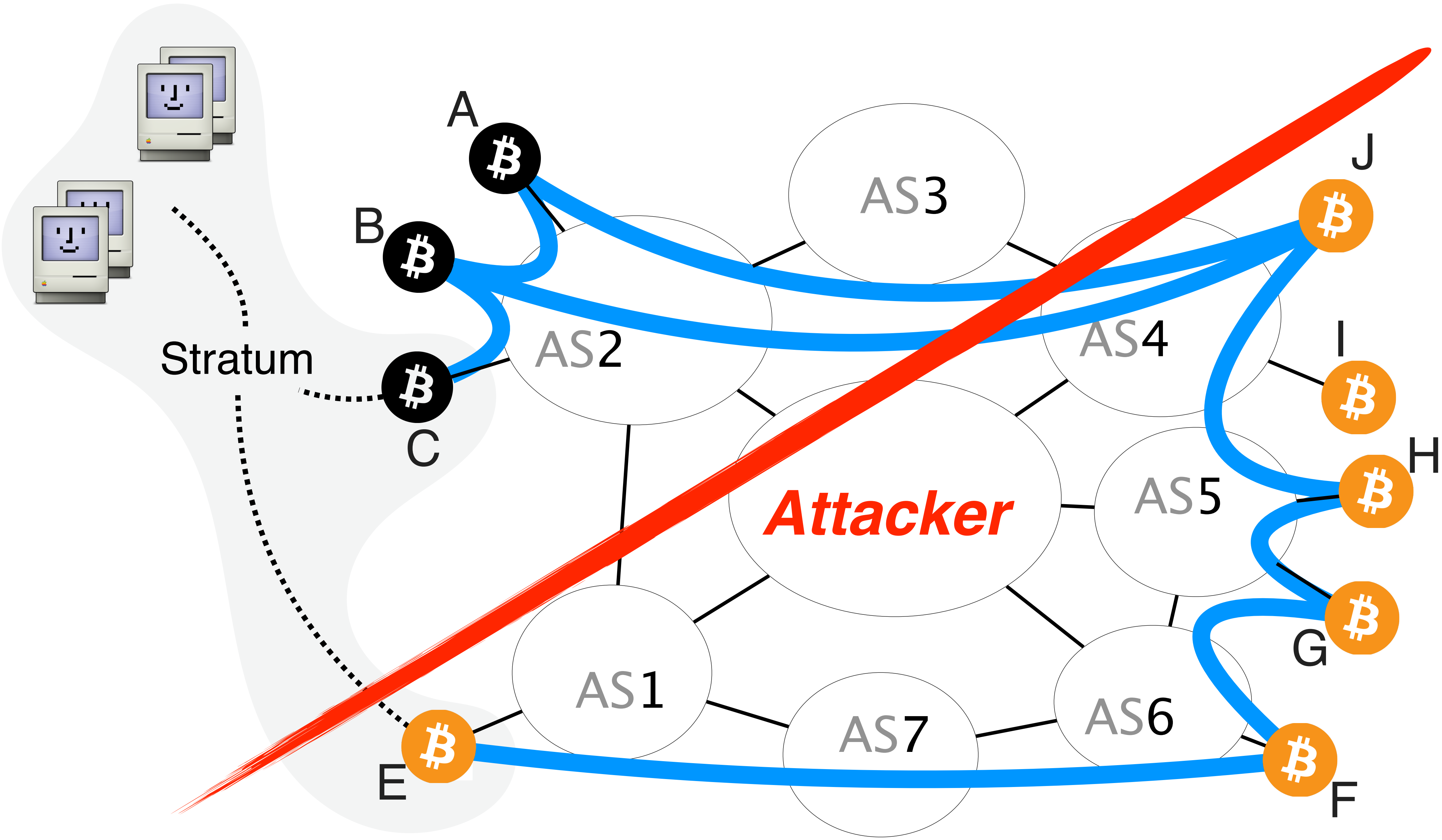
# Same attacker wants to create a different partition



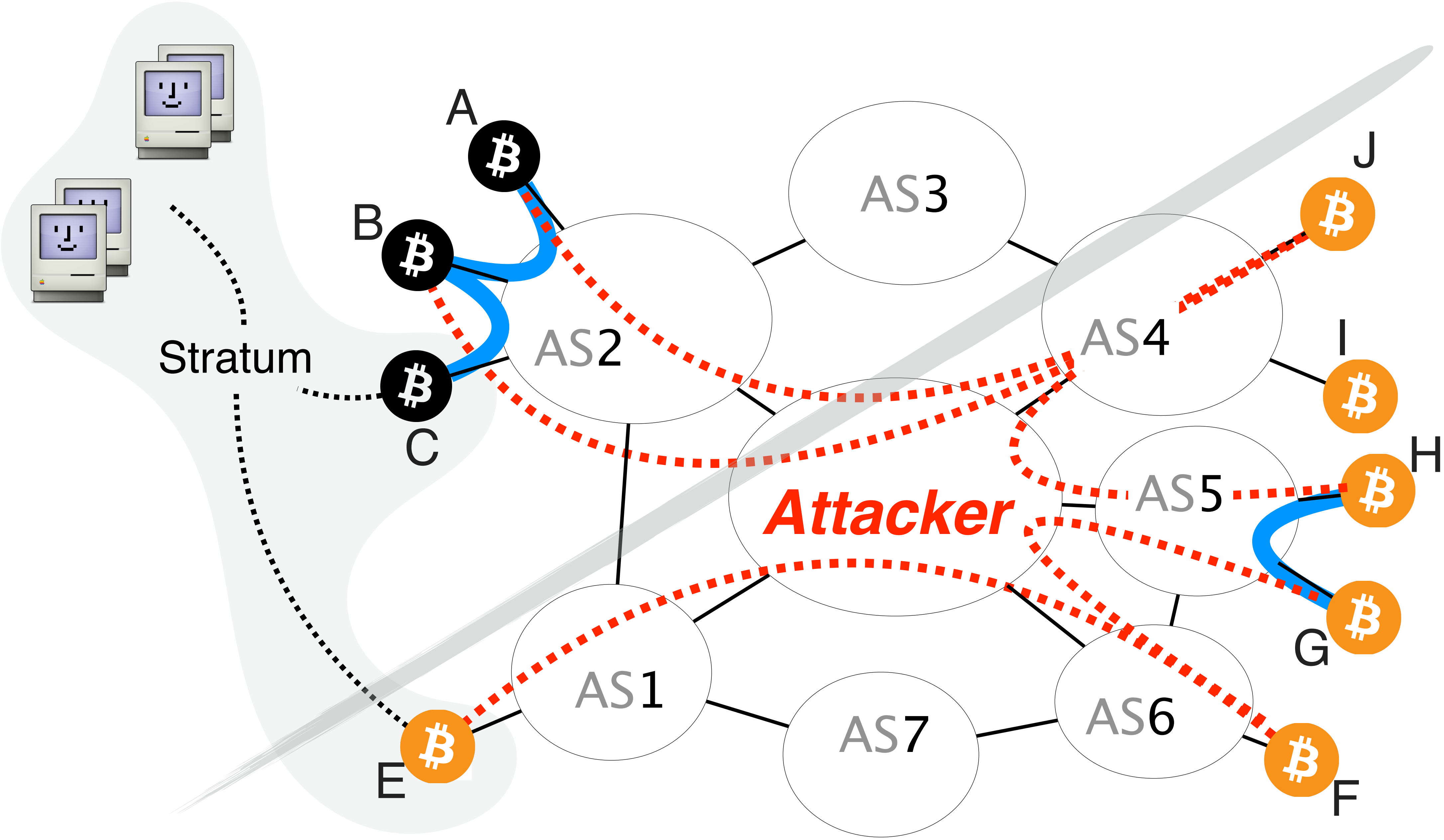
# There is a mining pool in the topology



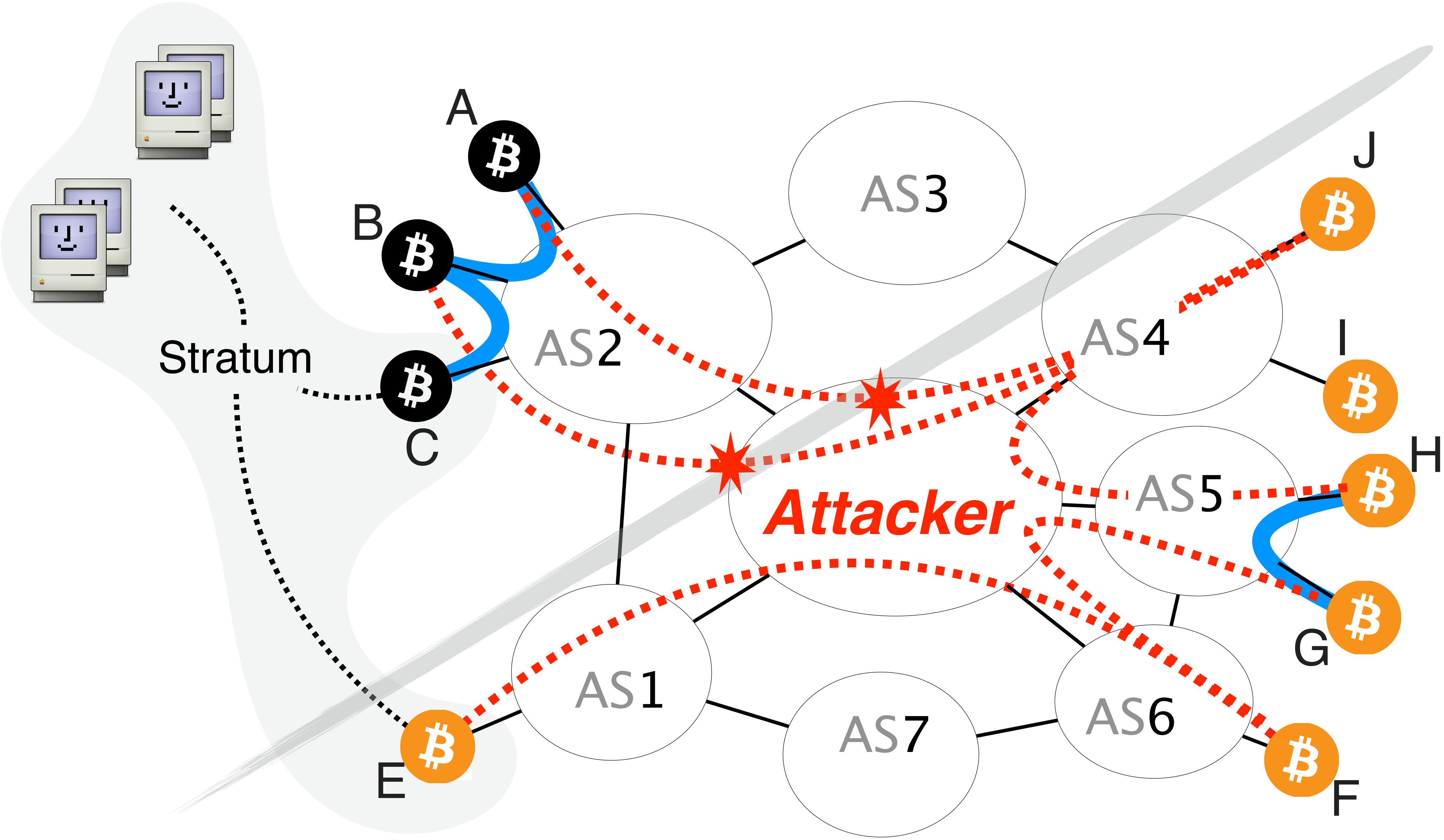
# Attacker hijacks all prefixes pertaining to nodes in the orange side



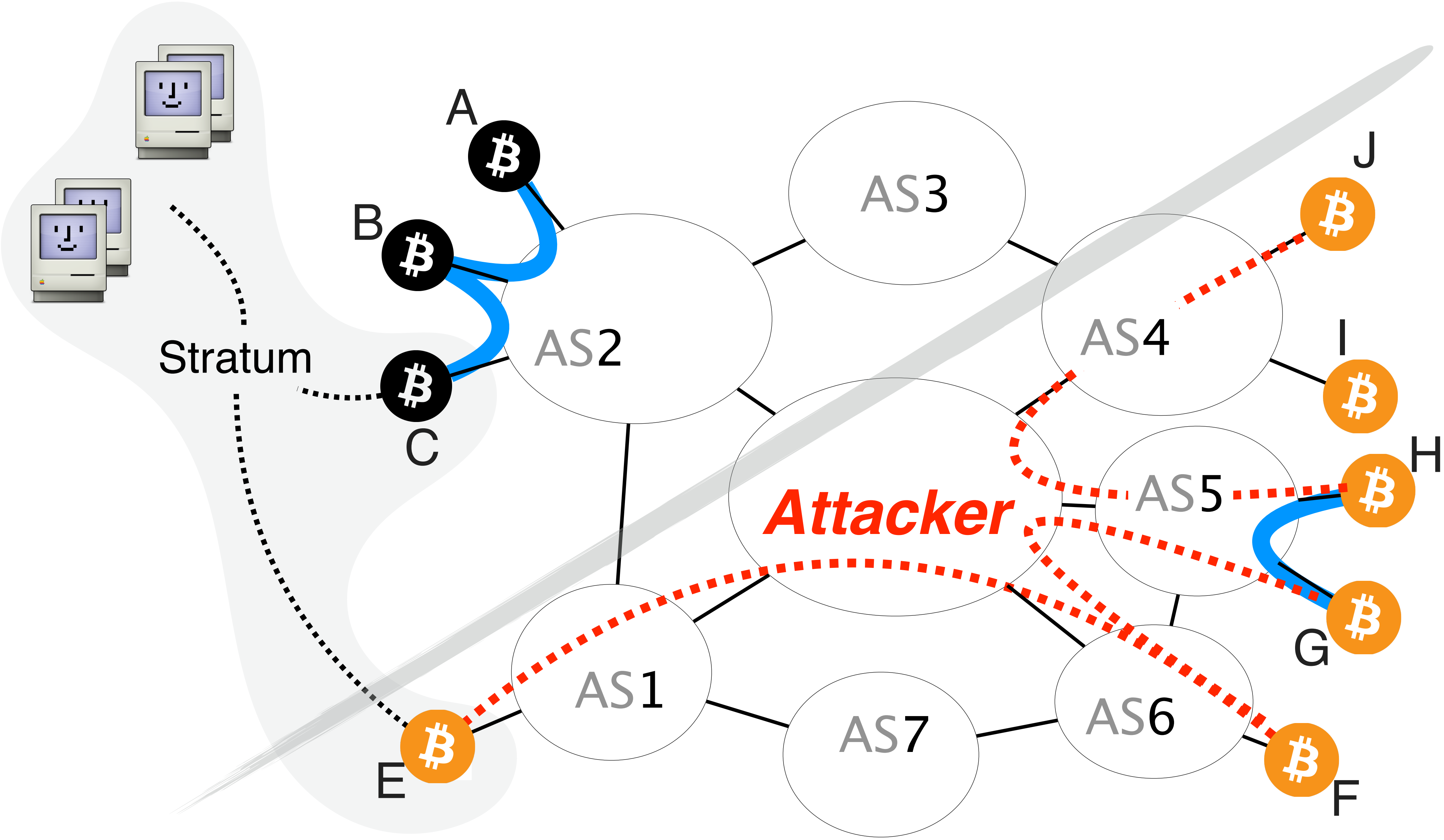
# Attacker hijacks all prefixes pertaining to nodes in the orange side



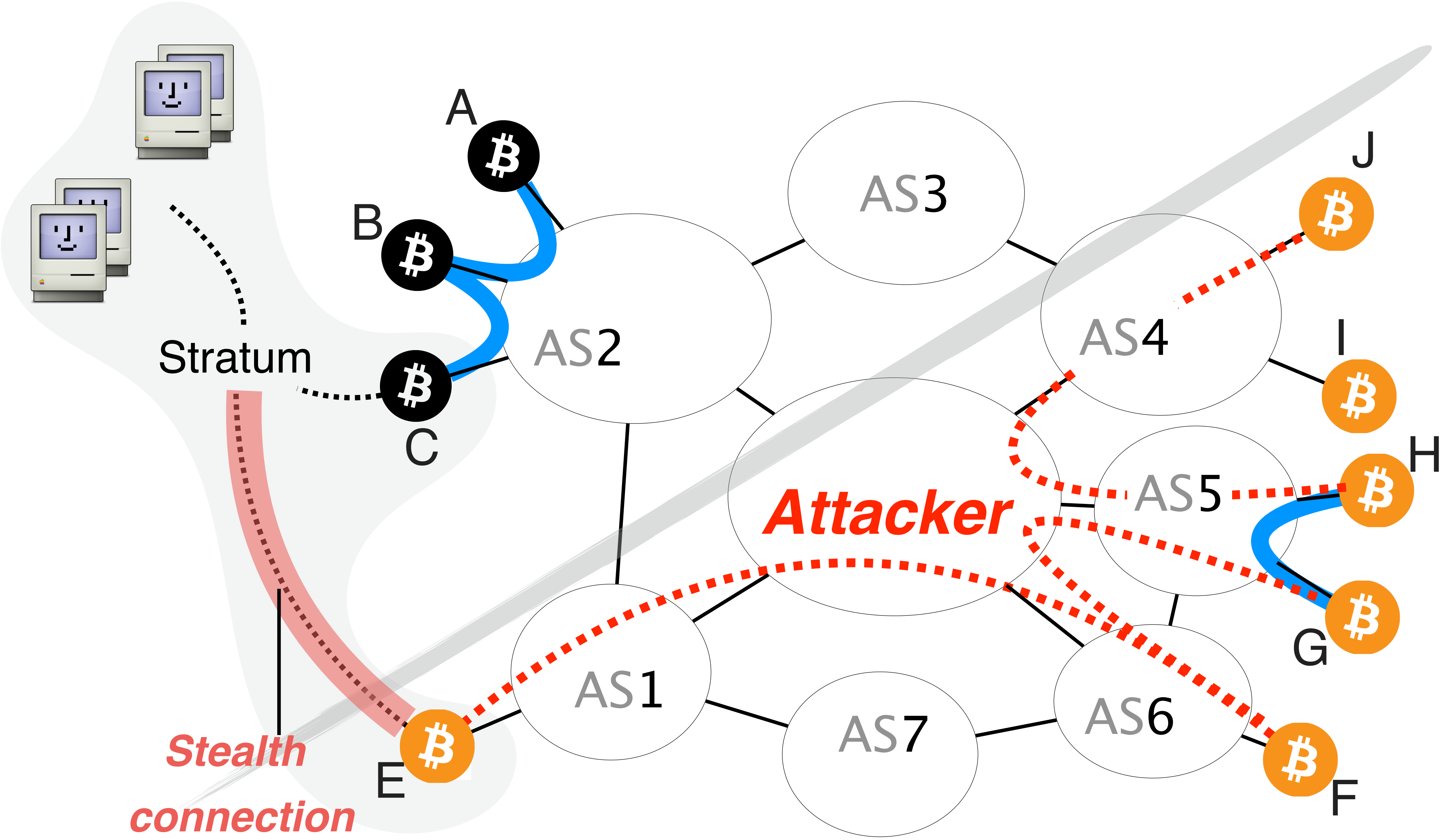
# The attacker drops connections



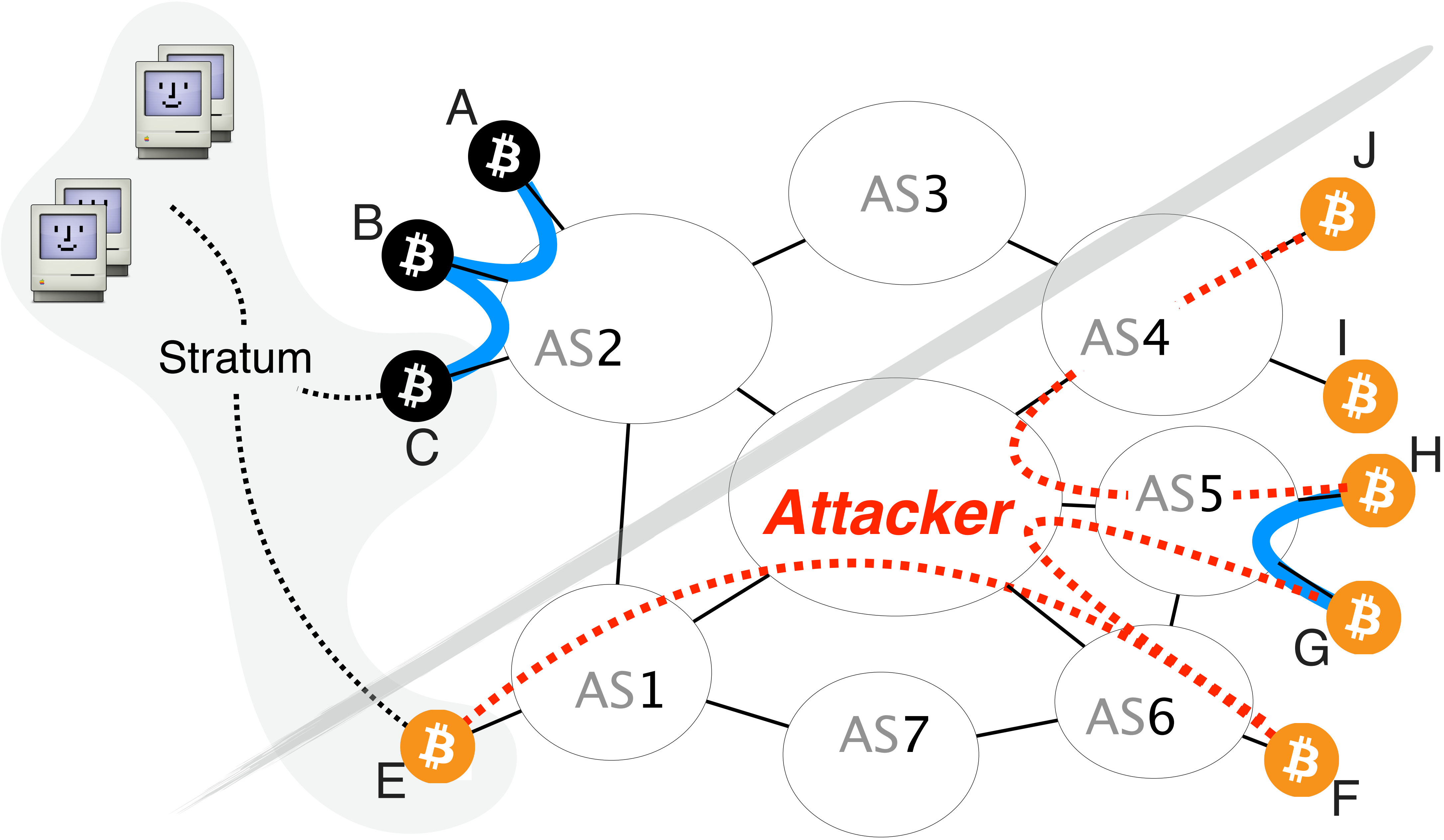
# The partition is created but is ineffective



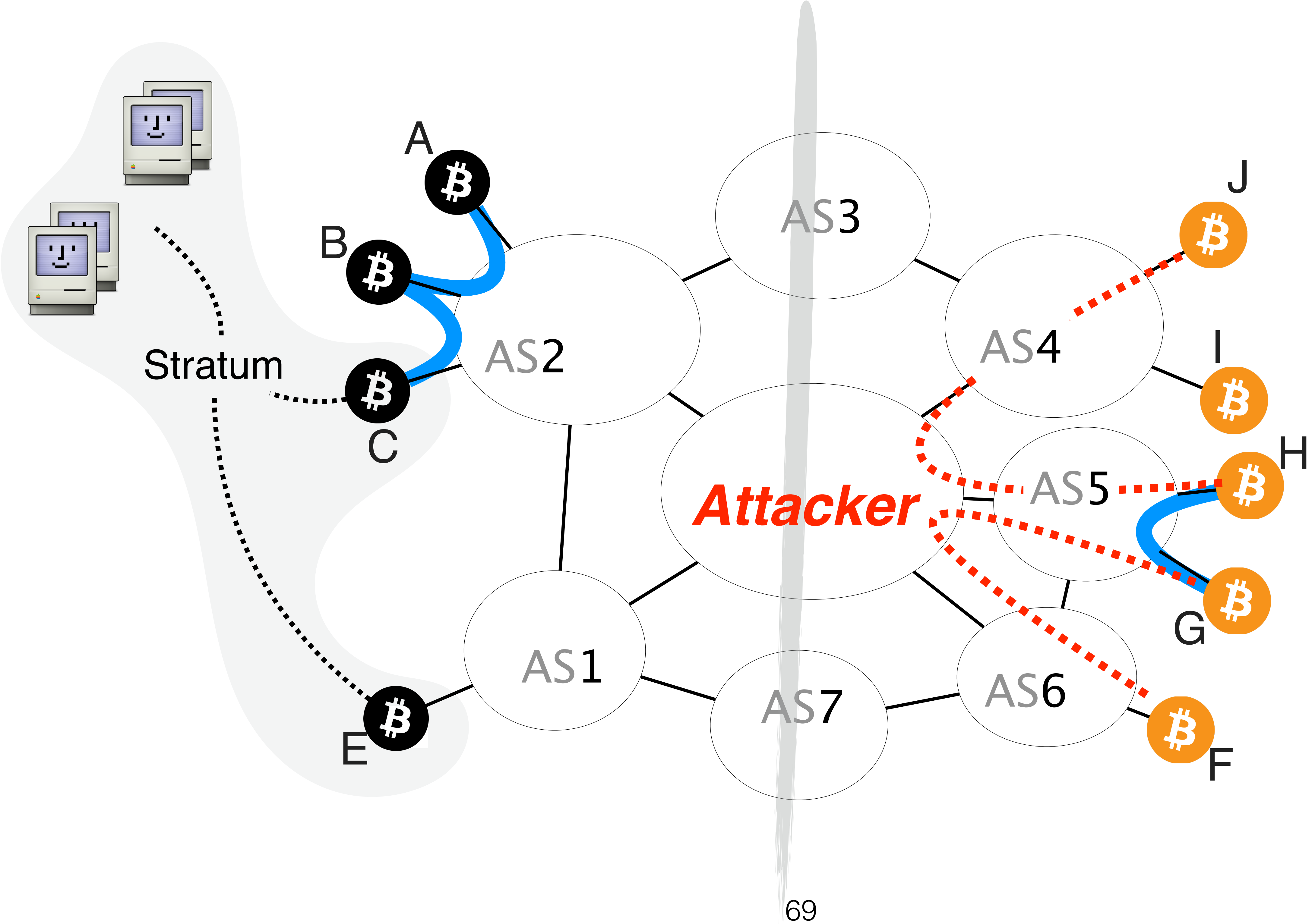
# The partition is infeasible



# The attacker monitors the connections and detects leakage



# The attacker monitors the connections



Theorem

Given a set of nodes to disconnect from the network, there exist a unique maximal subset that can be isolated and that the attacker will isolate.

see paper for proof

# We evaluated the partition attack in terms of practicality and time efficiency

Practicality

Can it actually happen?

Time efficiency

How long does it take?

# We evaluated the partition attack in terms of practicality and time efficiency

Practicality

Time efficiency

Can it actually happen?

Splitting the mining power even to half can be done by hijacking less than 100 prefixes

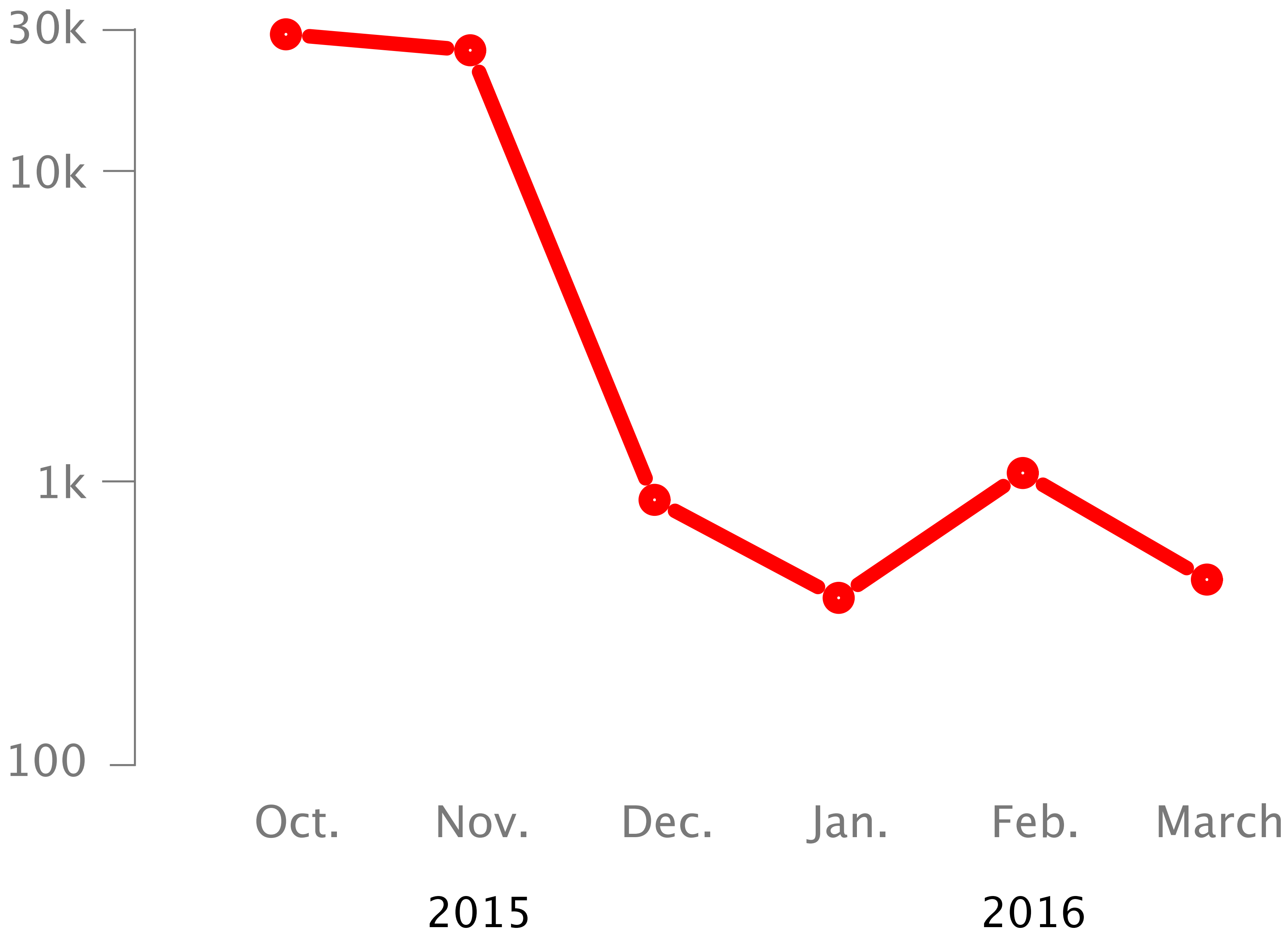
Splitting the mining power even to half can be done by hijacking less than 100 prefixes

*negligible* with respect to
routinely observed hijacks

# Hijacks involving up to 1k of prefixes are frequently seen in the Internet today

max # of prefixes hijacked at once

log scale



# We also evaluated the partition in terms of time efficiency

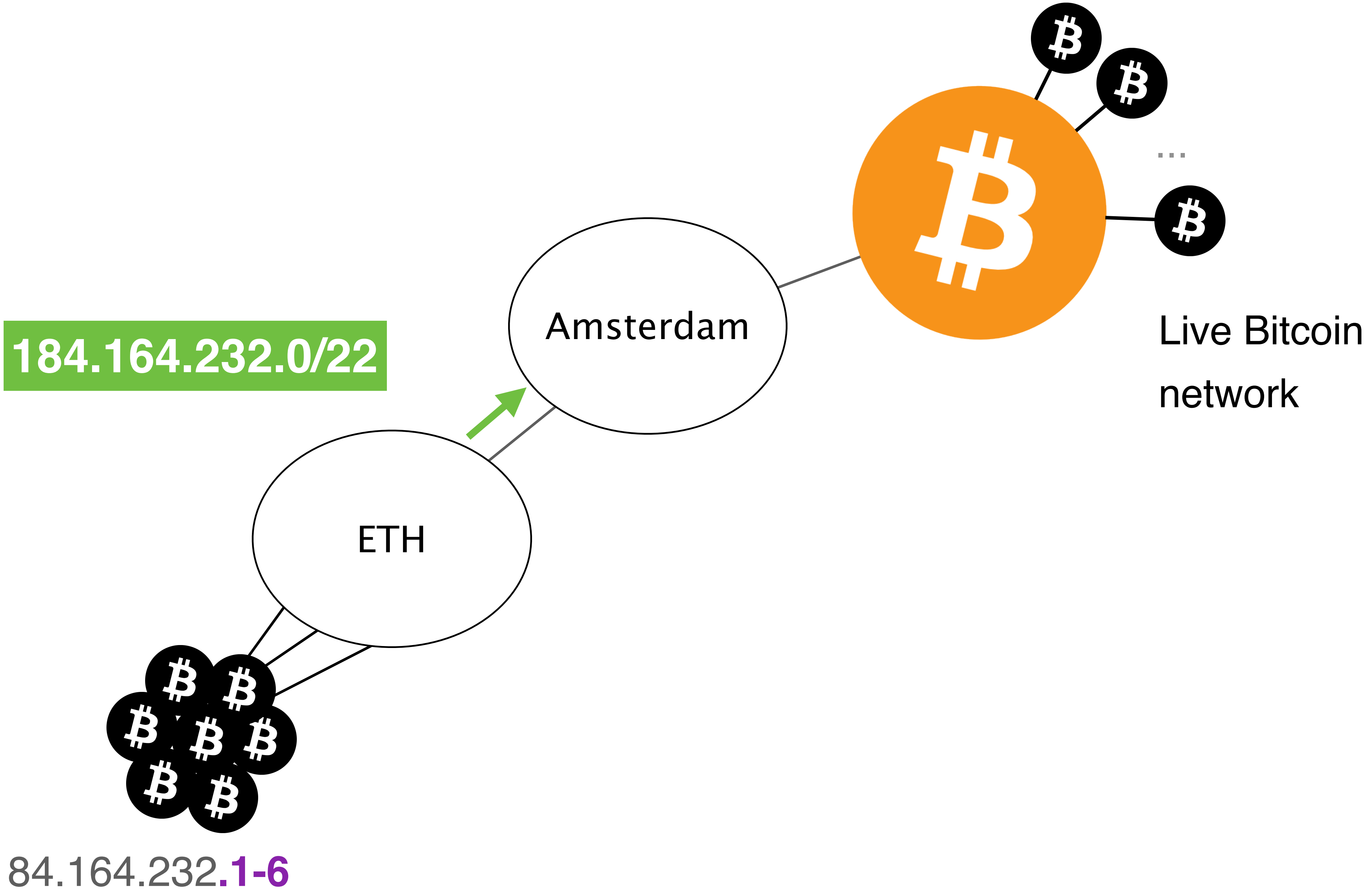
Practicality

Time efficiency

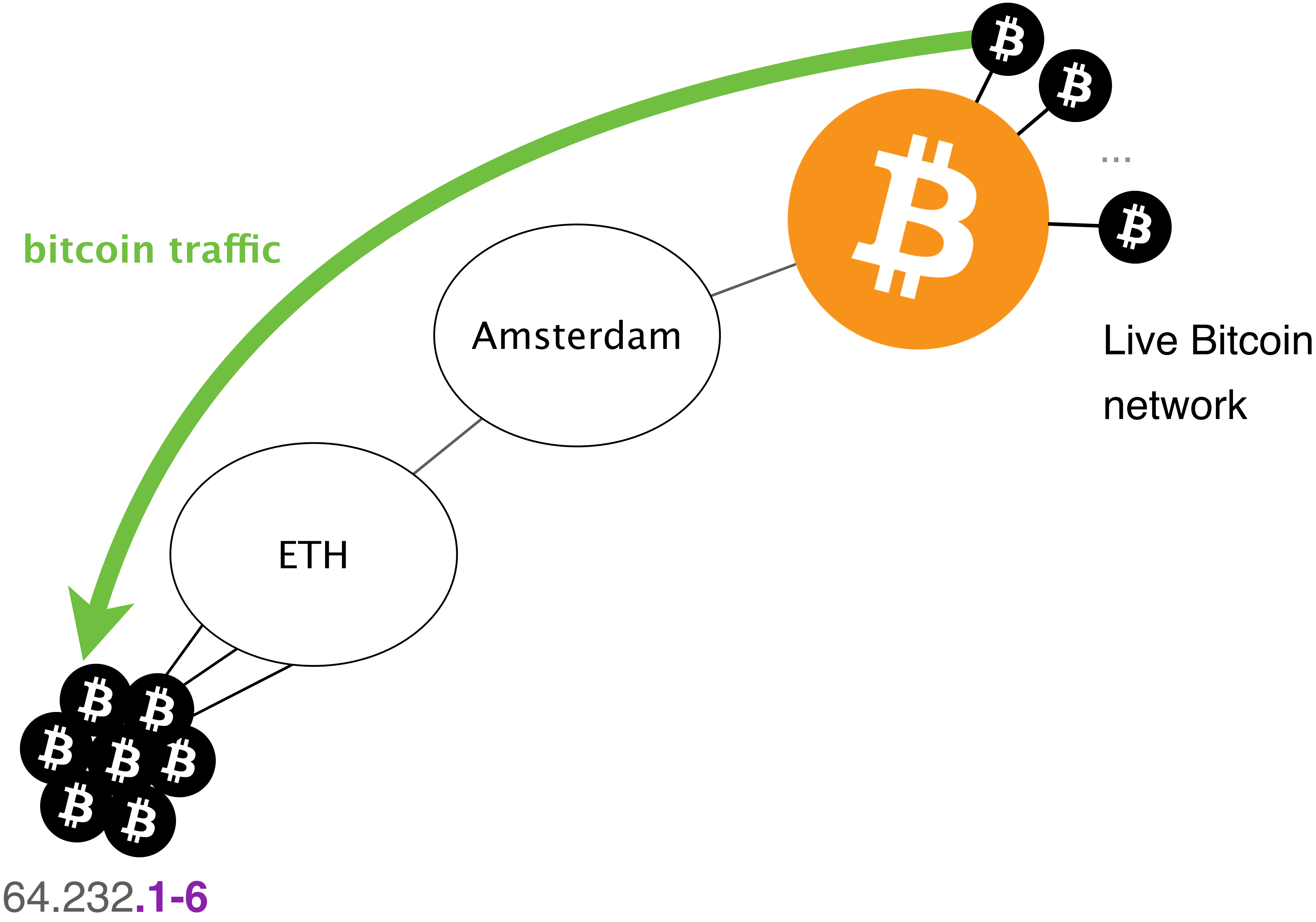
How long does it take?

We measured the time required to perform a partition attack by attacking our own nodes

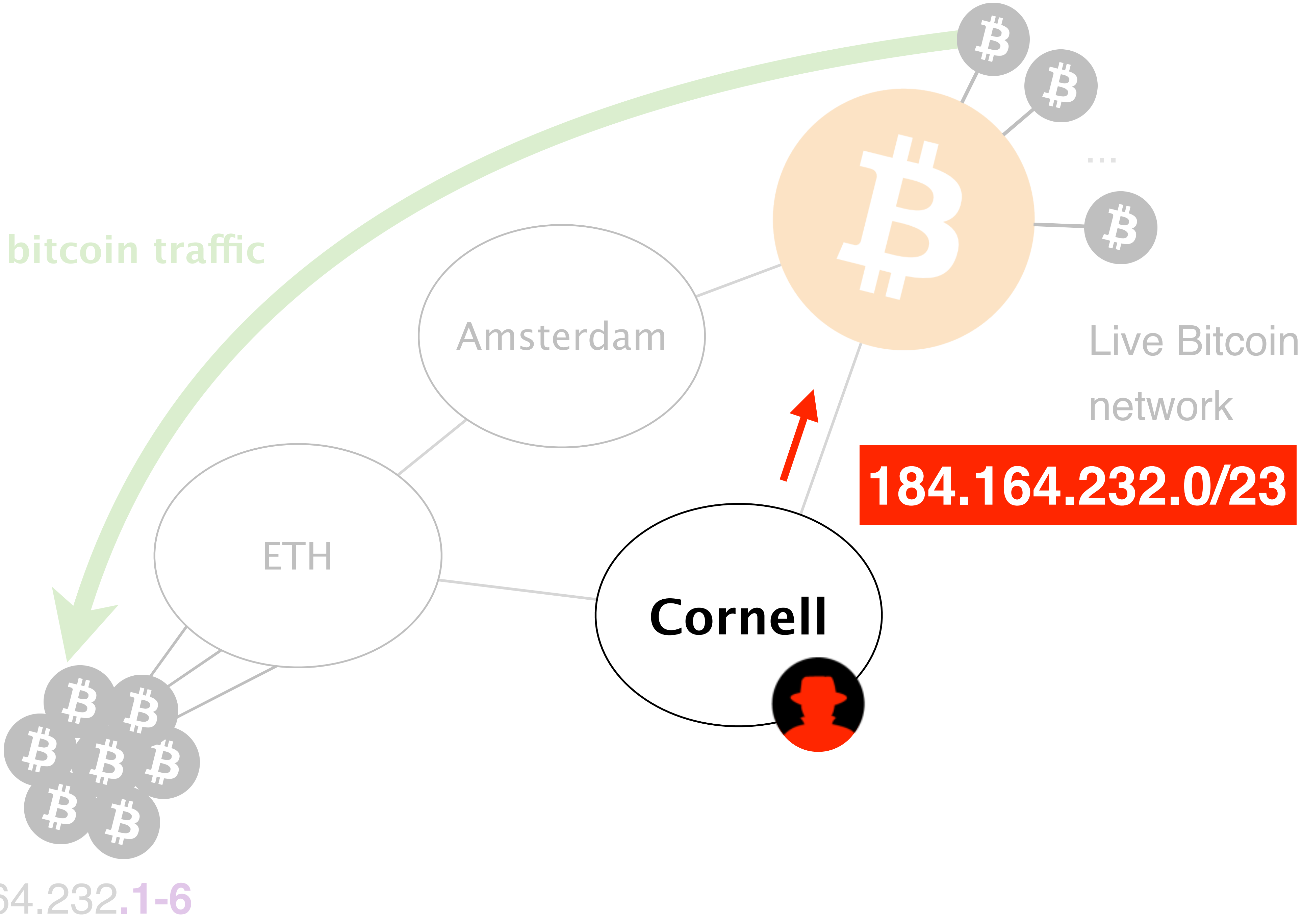
# We hosted a few Bitcoin nodes at ETH and advertised a covering prefix via Amsterdam



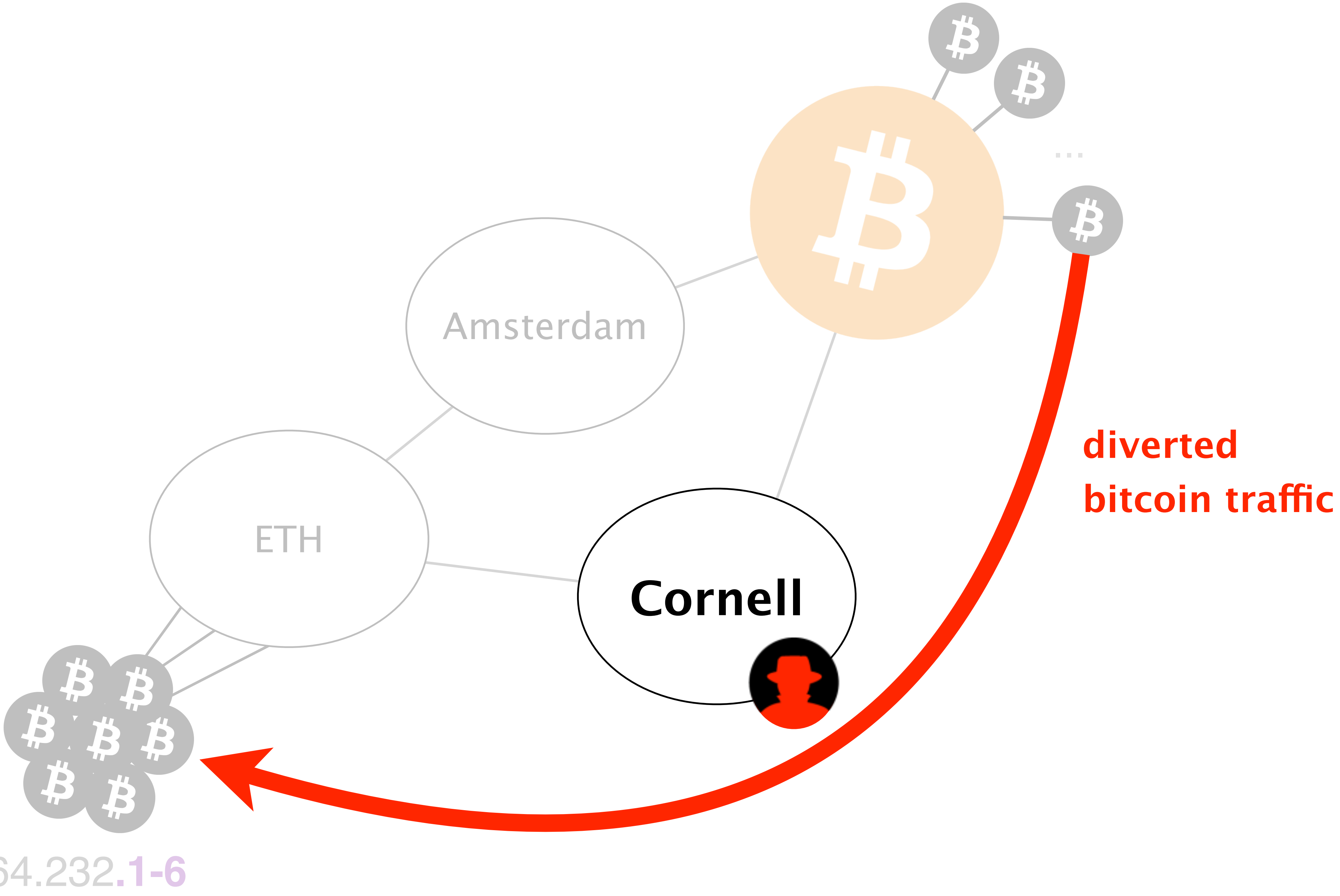
# Initially, all the traffic to our nodes transits via Amsterdam

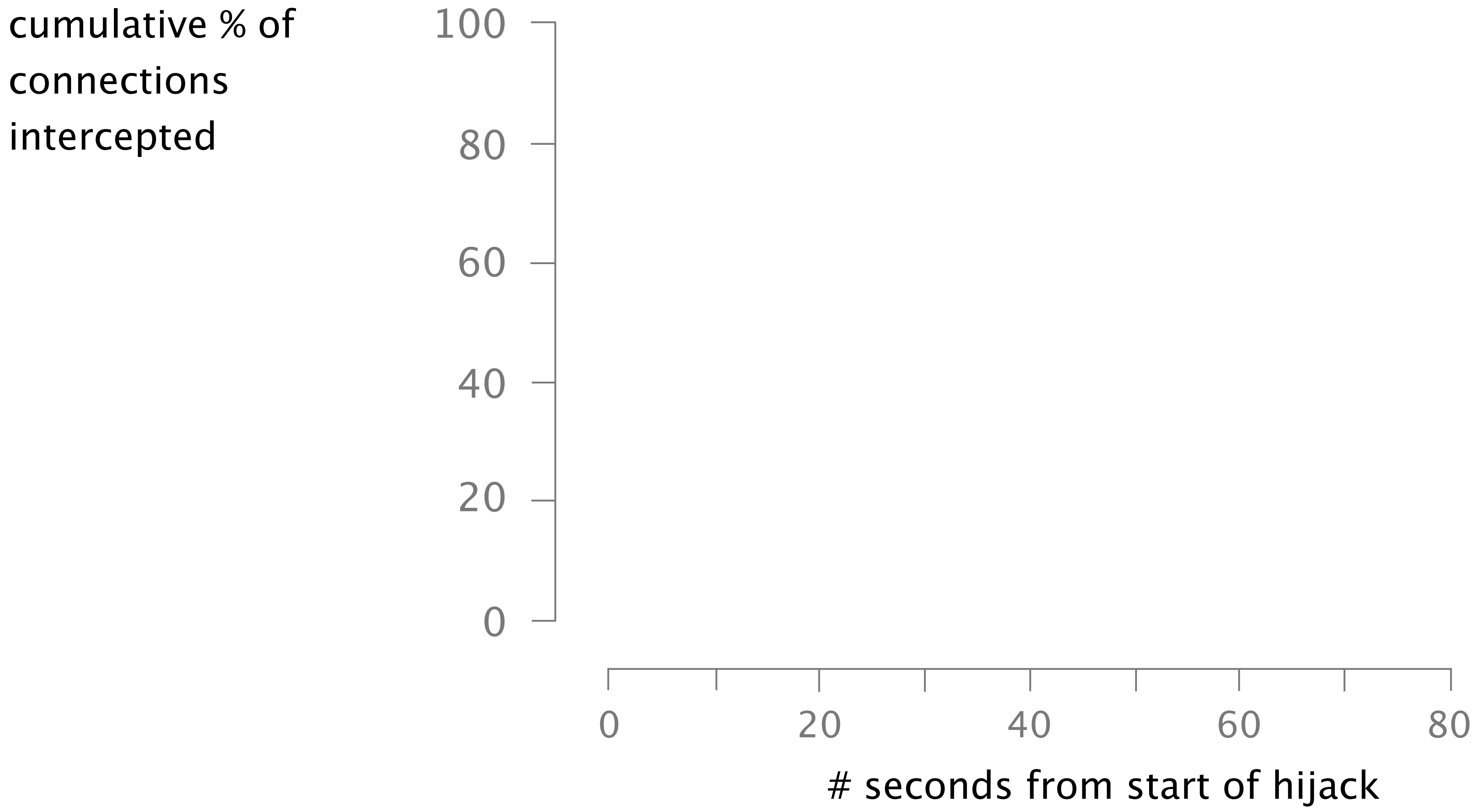


# We hijacked our nodes

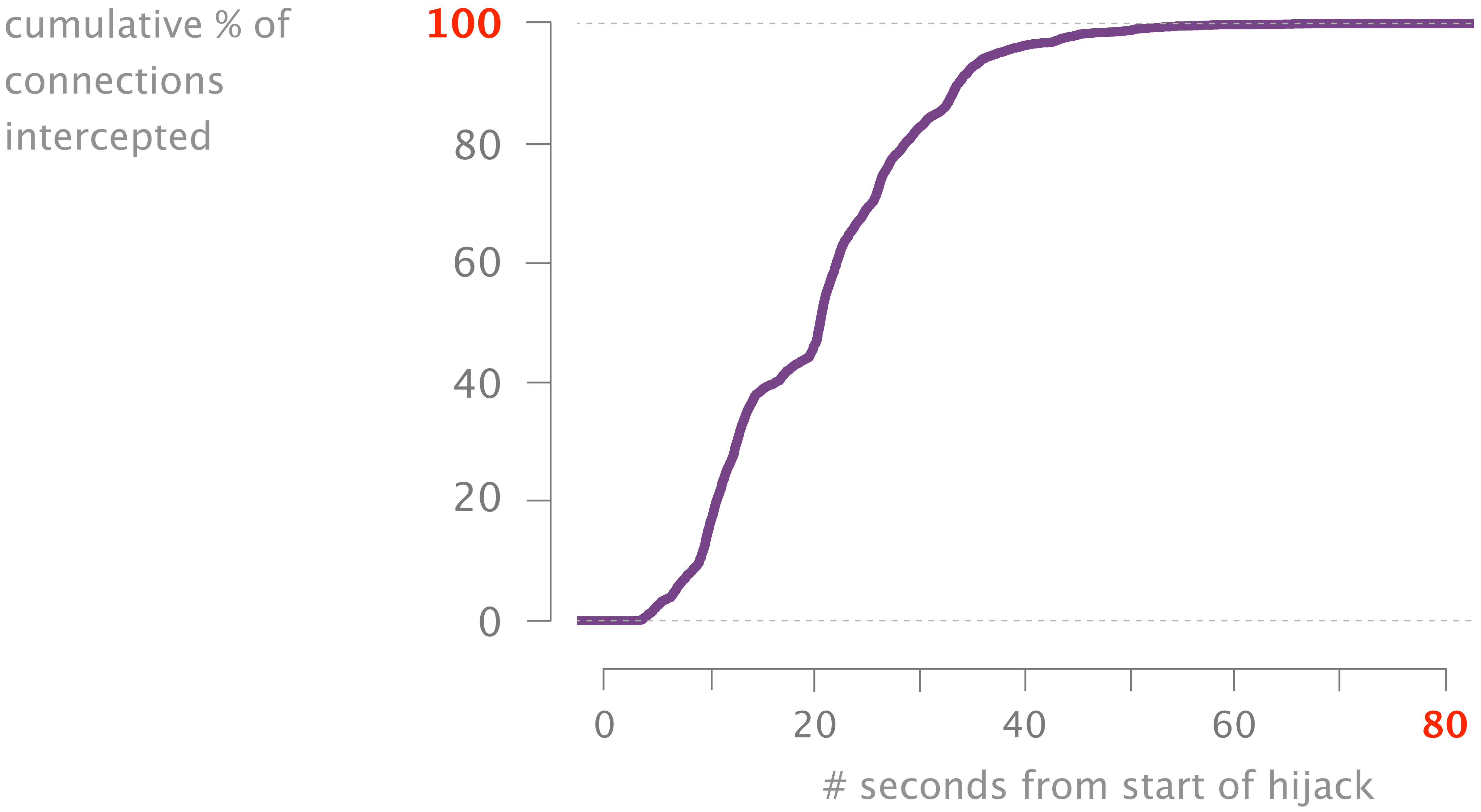


# We measured the time required for a rogue AS to divert all the traffic to our nodes




cumulative % of connections intercepted
100
80
60
40
20
0
0
20
40
60
80
# seconds from start of hijack

# It takes less than 2 minutes for the attacker to intercept all the connections



Mitigating a hijack is a human-driven process,
as such it often takes hours to be resolved

Mitigating a hijack is a human-driven process, as such it often takes hours to be resolved

It took Google close to 3h
to mitigate a large hijack in 2008 [6]

(same hold for more recent hijacks)

# We measured the healing time of the partition in a testbed of 1050 Bitcoind clients

The Bitcoin network will regain connectivity in seconds after the hijack stops

The two components will be loosely connected for hours

# We measured the healing time of the partition in a testbed of 1050 Bitcoind clients

The Bitcoin network will regain connectivity in seconds after the hijack stops

The two components will be loosely connected for hours

# We measured the healing time of the partition in a testbed of 1050 Bitcoind clients

The Bitcoin network will regain connectivity in seconds after the hijack stops

The two components will be loosely connected for hours

# Hijacking Bitcoin

Routing Attacks on Cryptocurrencies

1 **Background**

BGP & Bitcoin

2 **Partitioning attack**

splitting the network

3 **Delay attack**

slowing the network down

4 **Countermeasures**

short-term & long-term

The goal of a delay attack is to keep the victim uninformed of the latest Block

# The impact of delay attacks is worrying and depends on the victim

Merchant

Mining pool

Regular node

# The impact of delay attacks is worrying and depends on the victim

Merchant

susceptible to be the victim of double-spending attacks

Mining pool

Regular node

# The impact of delay attacks is worrying and depends on the victim

Merchant

Mining pool

waste their mining power by mining on an obsolete chain

Regular node

# The impact of delay attacks is worrying and depends on the victim

Merchant

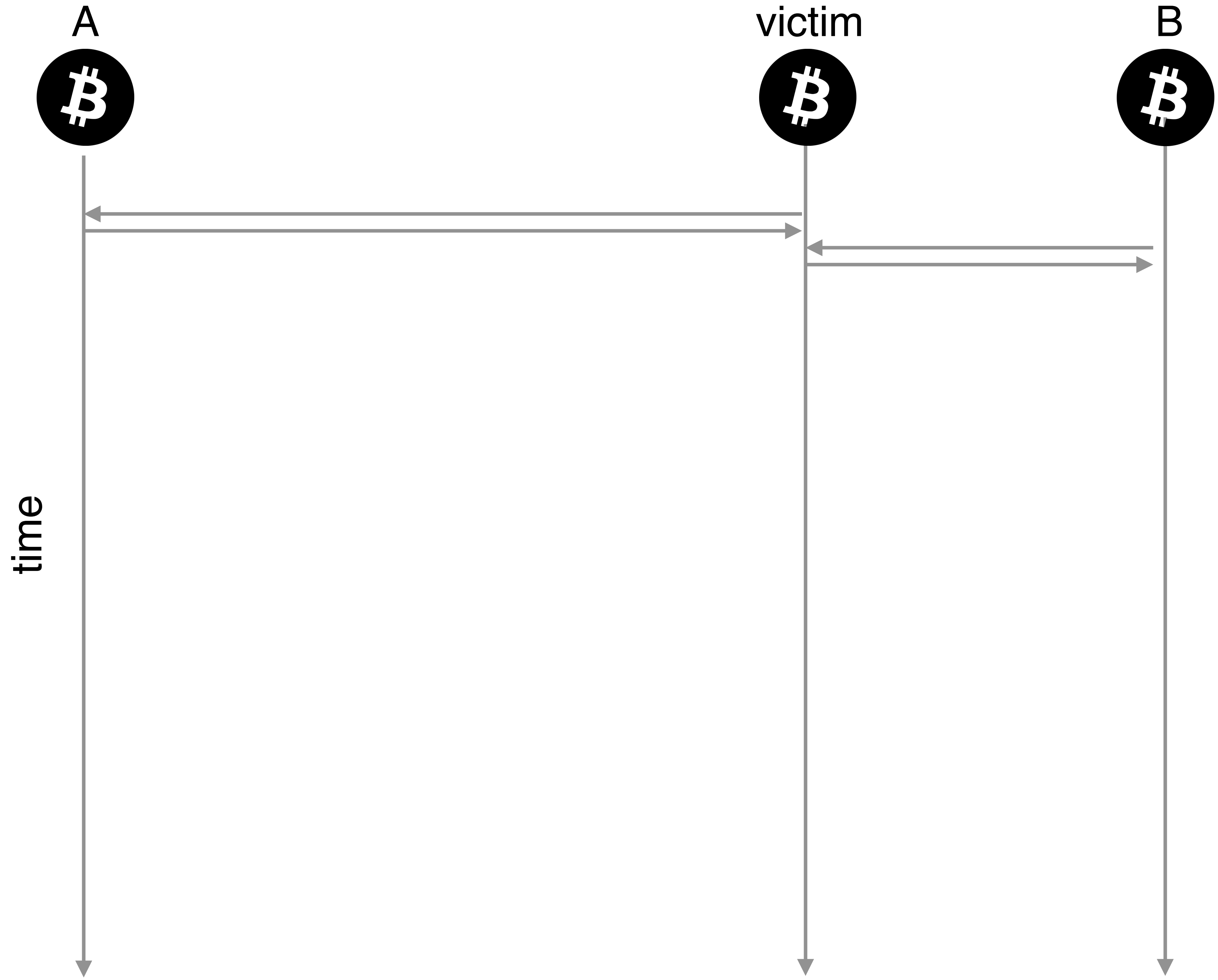
Mining pool

Regular node

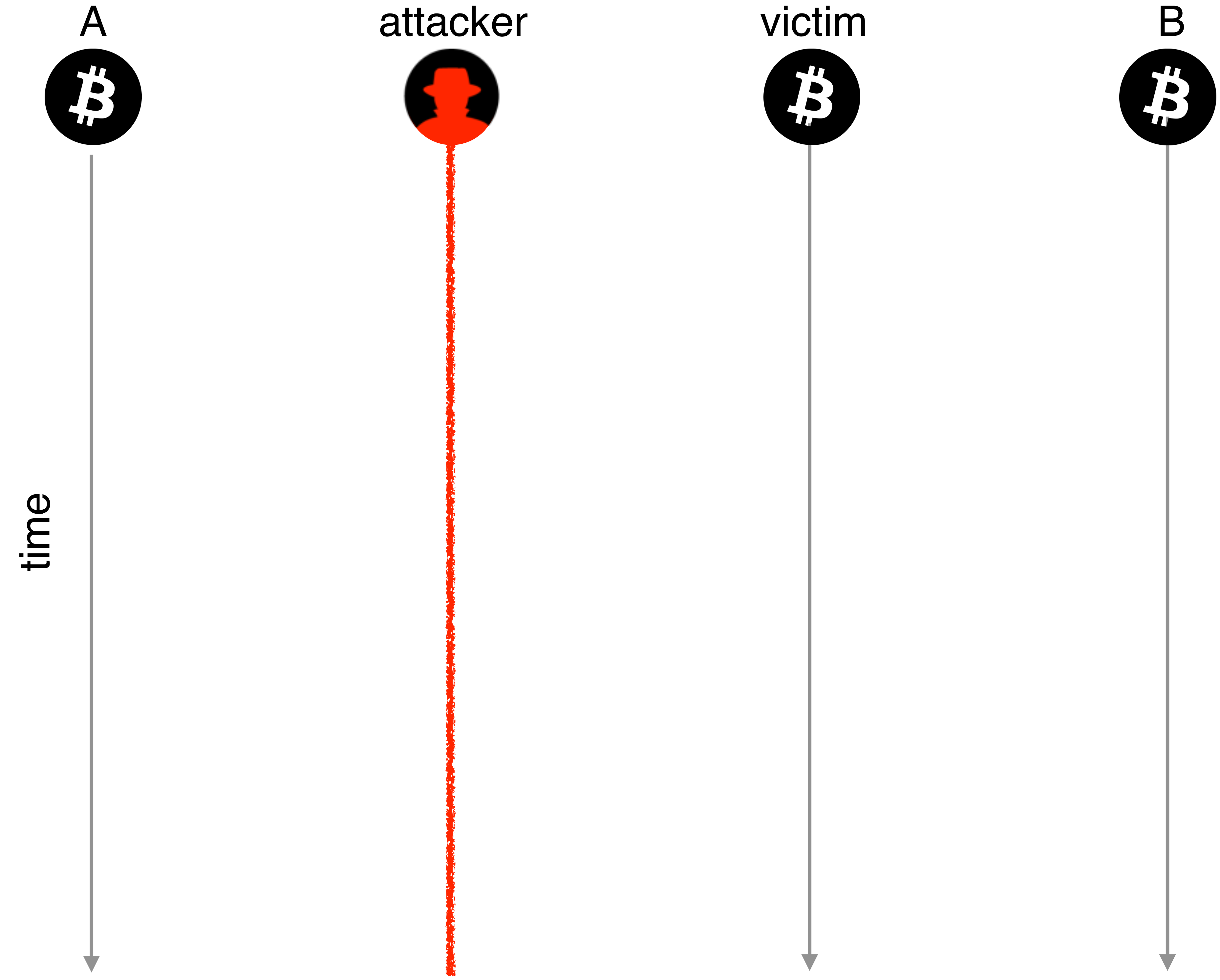
unable to collaborate to
the peer-to-peer network

## How does a delay attack work?

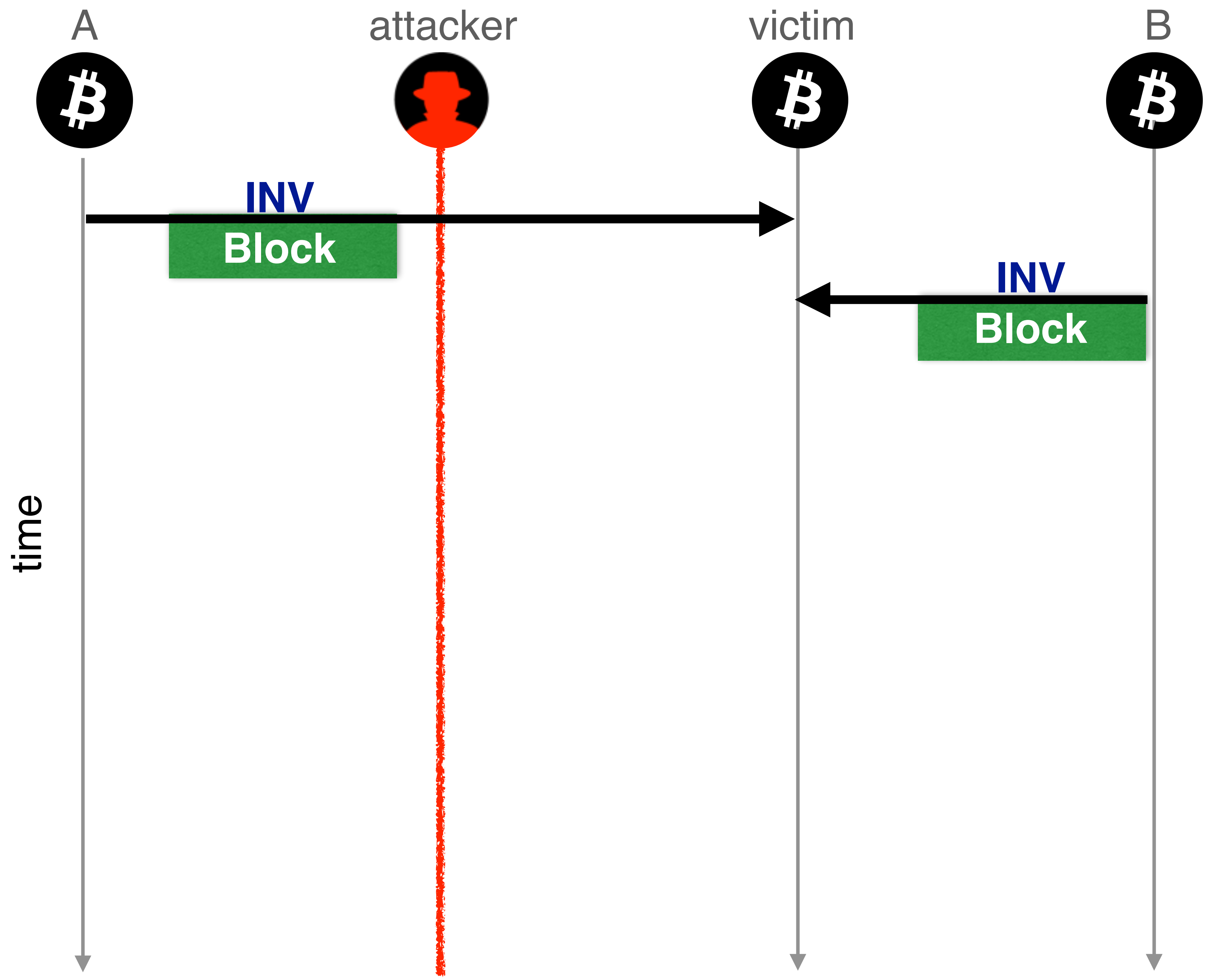
# Consider these three Bitcoin nodes



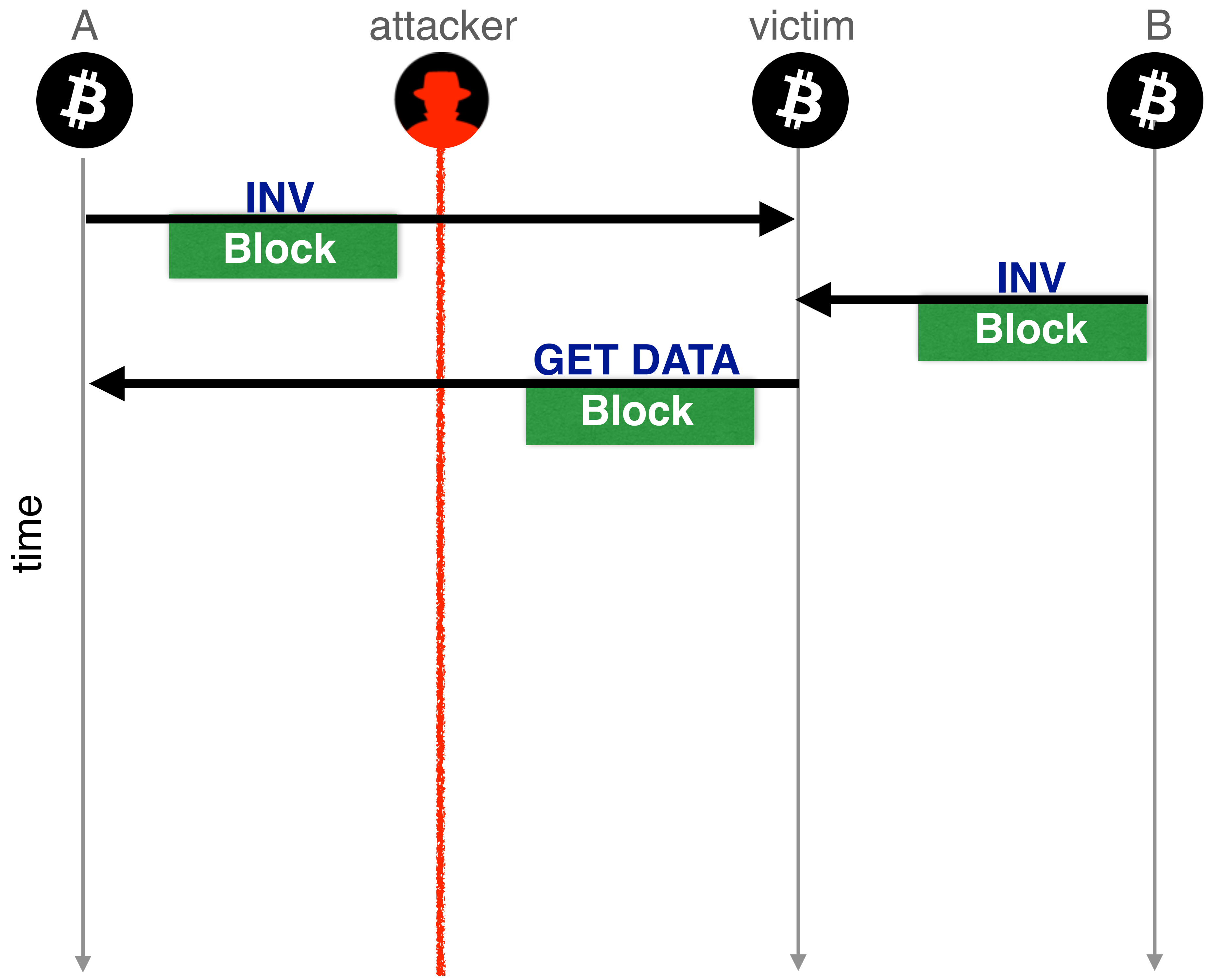
# An attacker wishes to delay the block propagation towards the victim



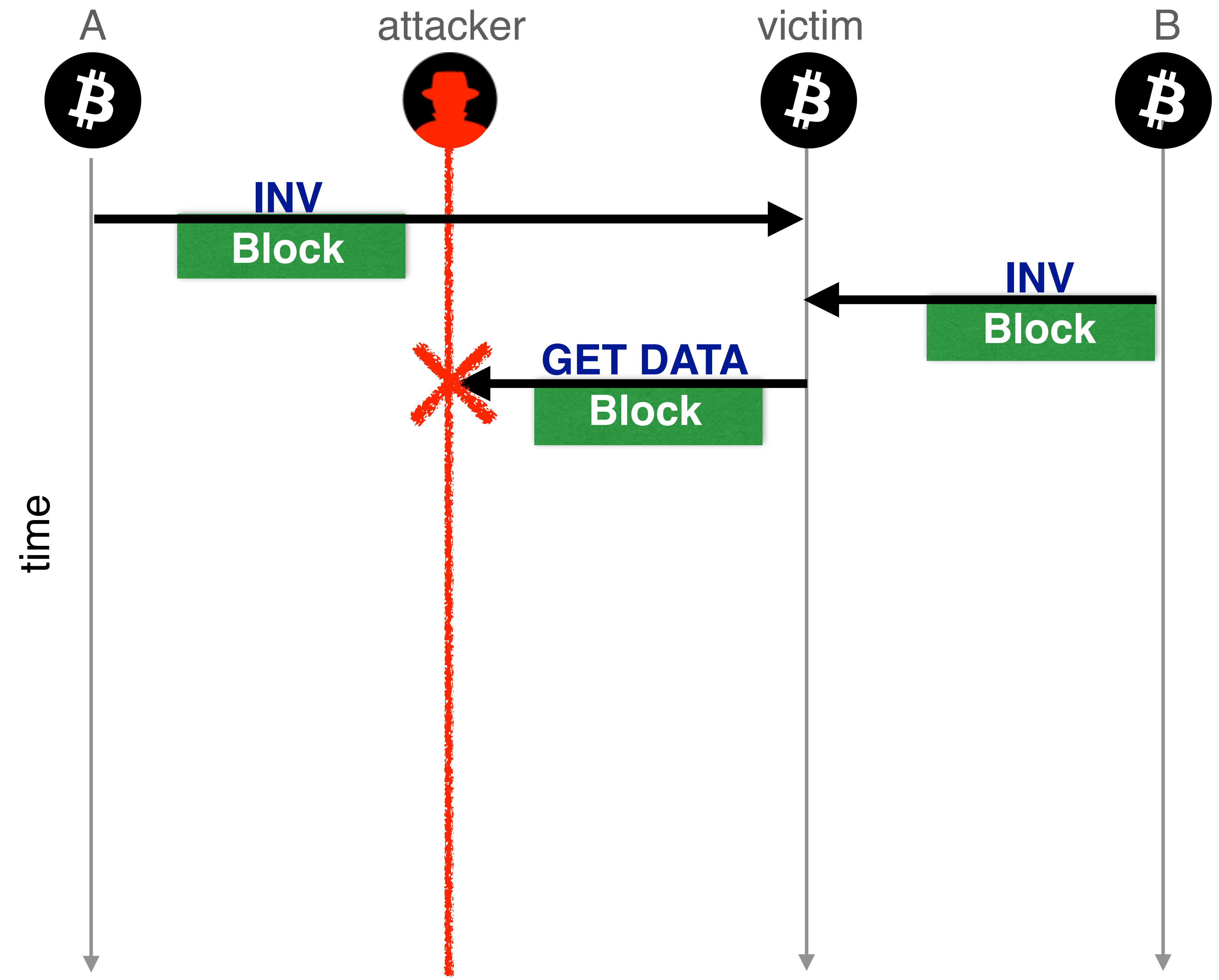
# The victim receives two advertisement for the **block**



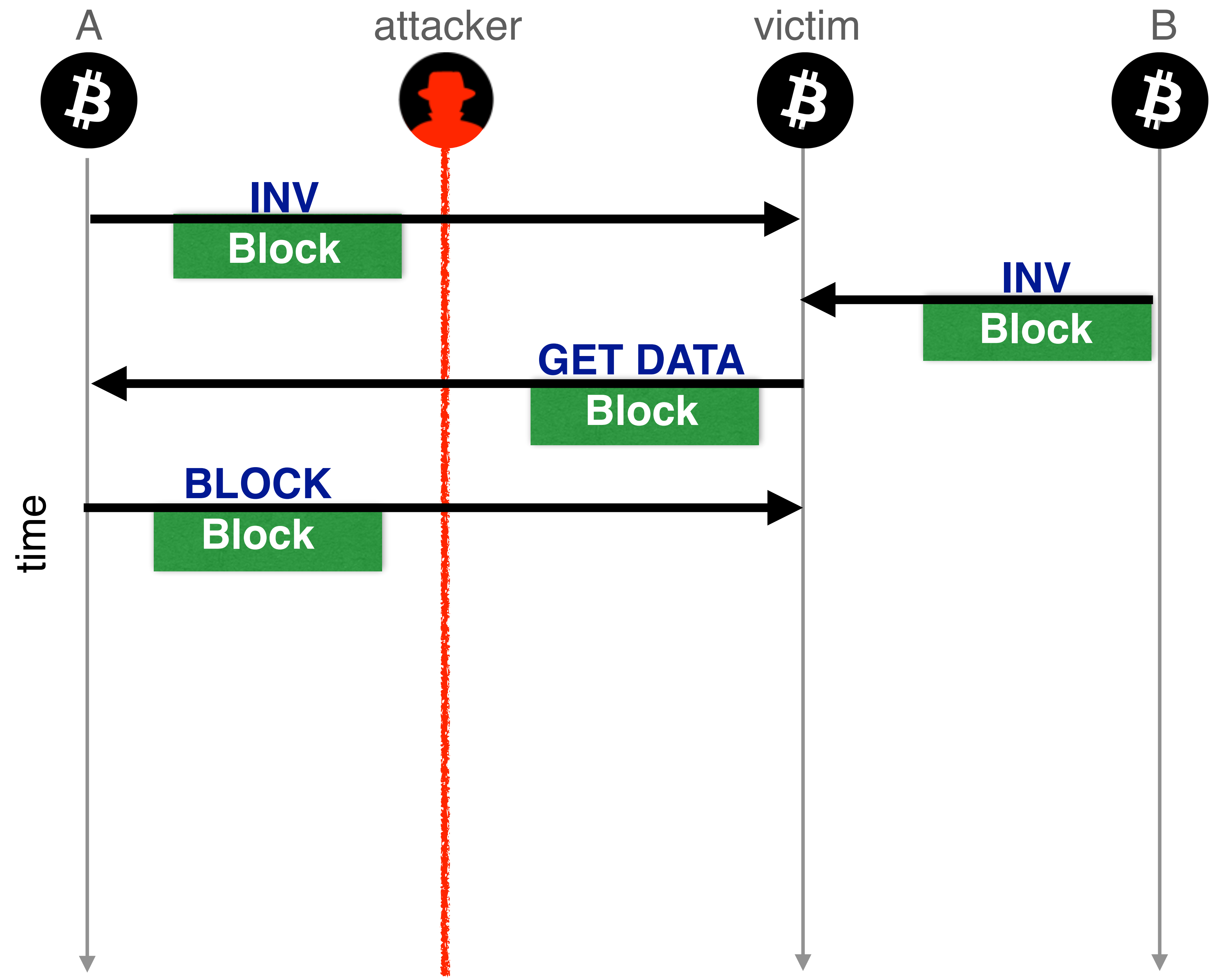
The victim requests the **block** to one of its peer, say A



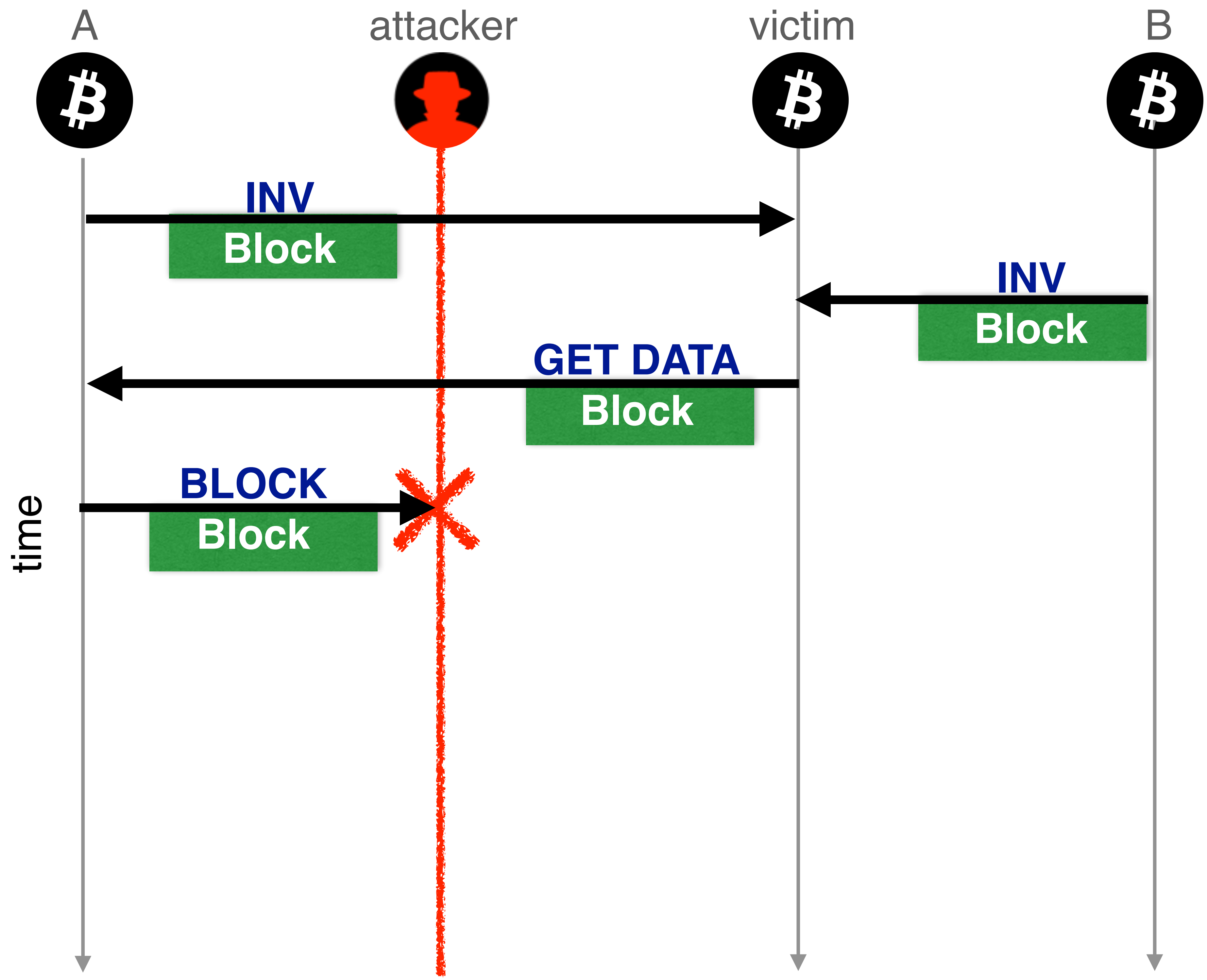
# As a MITM, the attacker could drop the **GETDATA** message



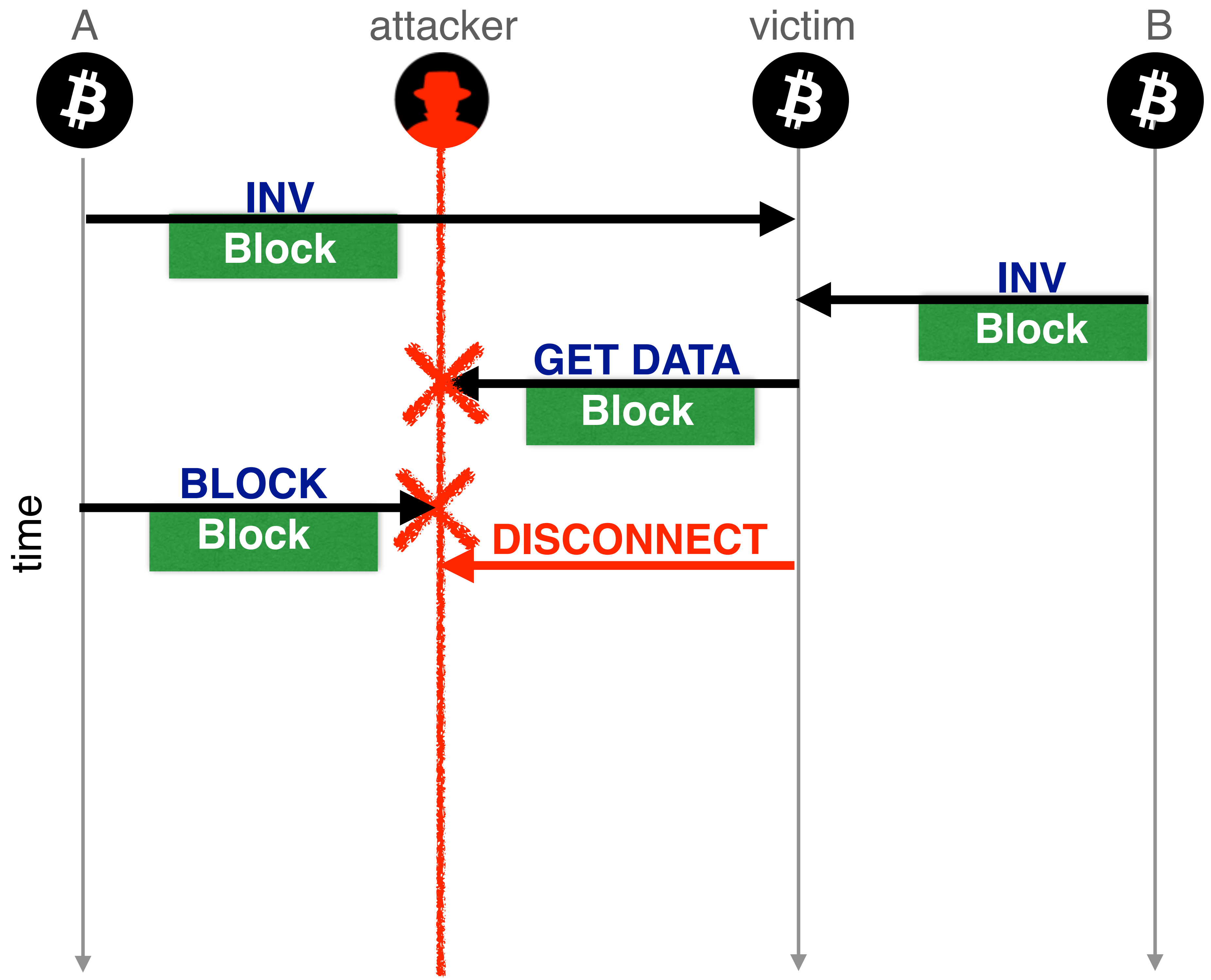
Similarly, the attacker could drop the delivery of the **block** message



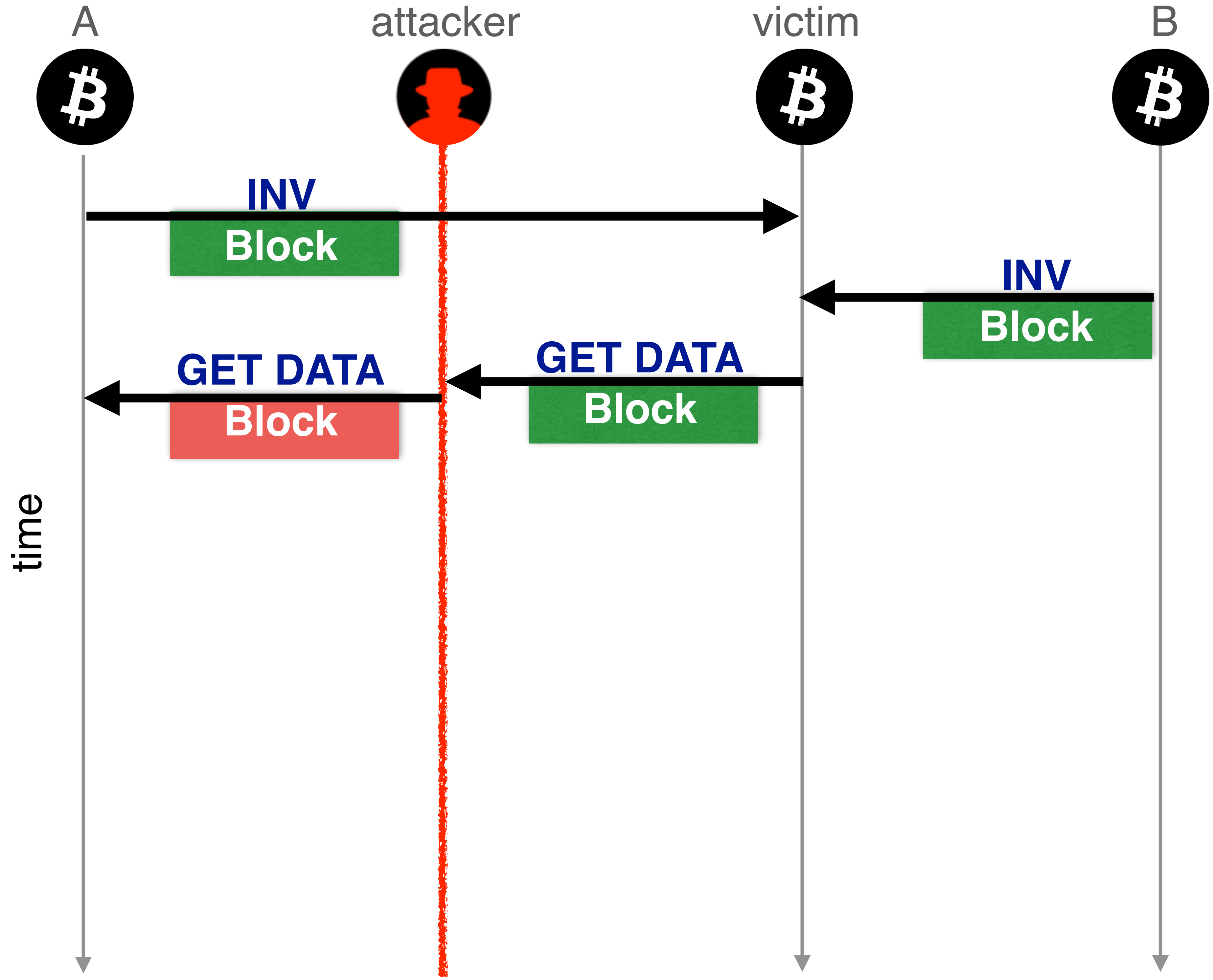
# Similarly, the attacker could drop the delivery of the **block** message



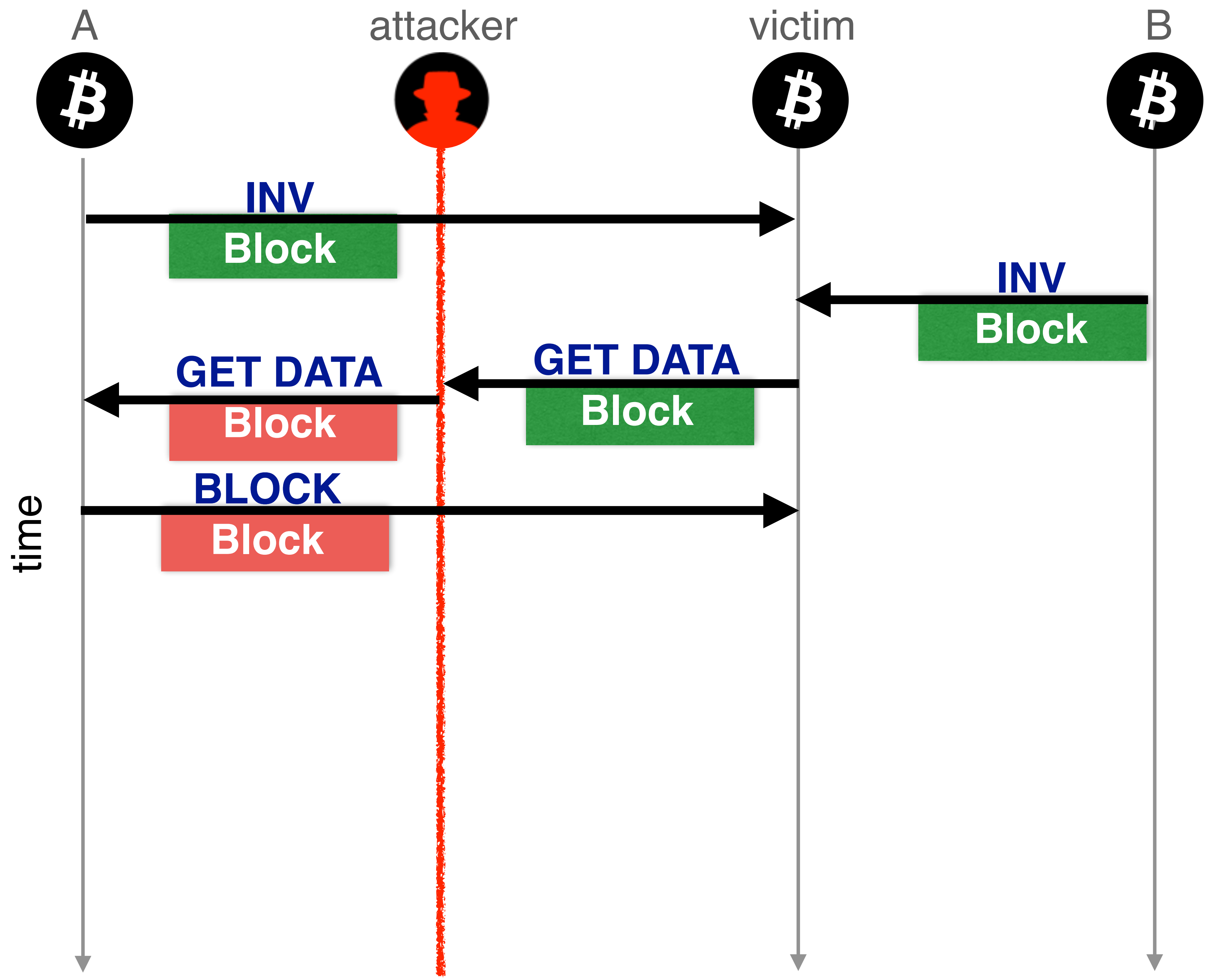
Yet, both cases will lead to the victim killing the connection (by the TCP stack on the victim)



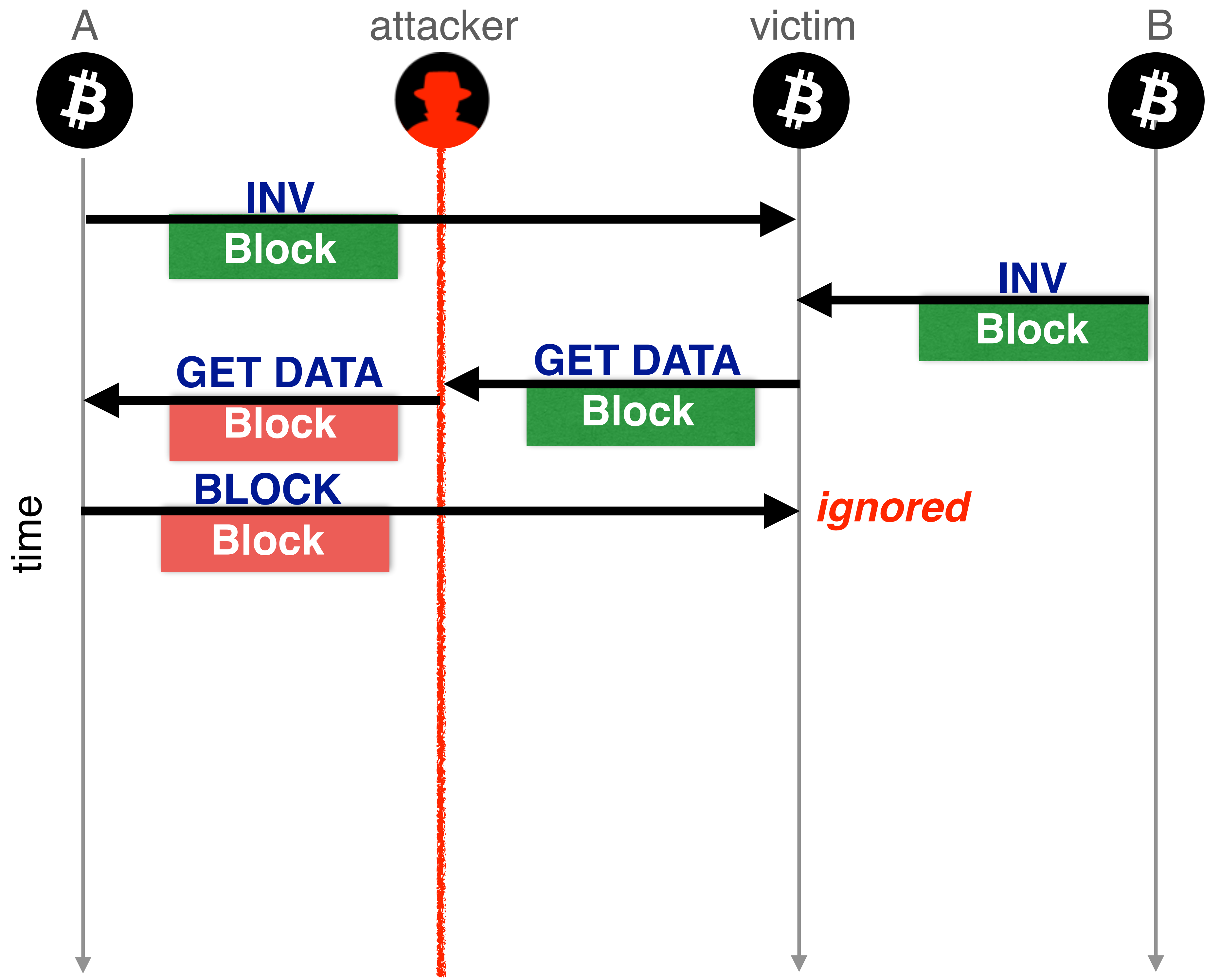
Instead, the attacker could intercept the **GETDATA** and modifies its content



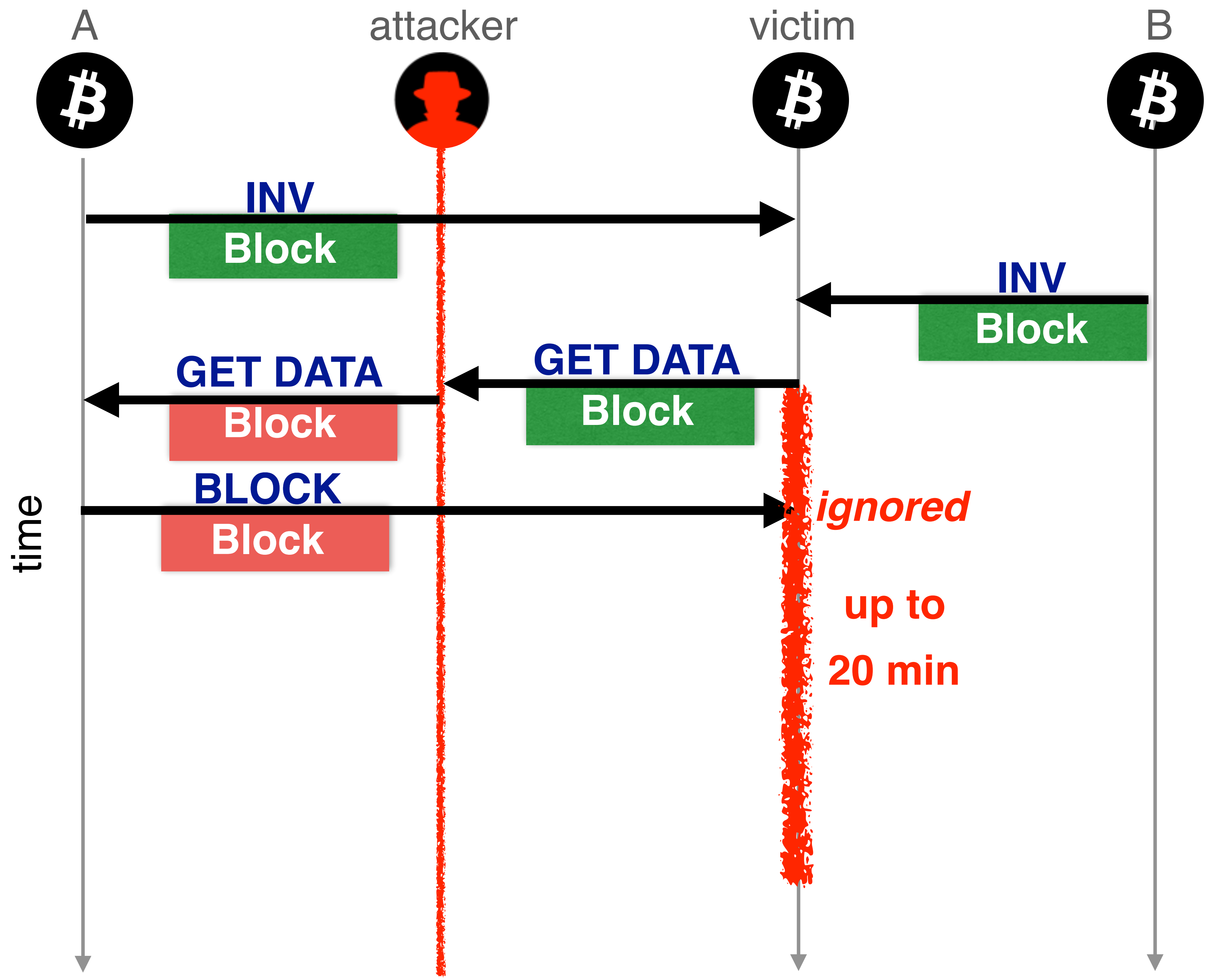
By modifying the ID of the requested block,
the attacker triggers the delivery of an older **block**



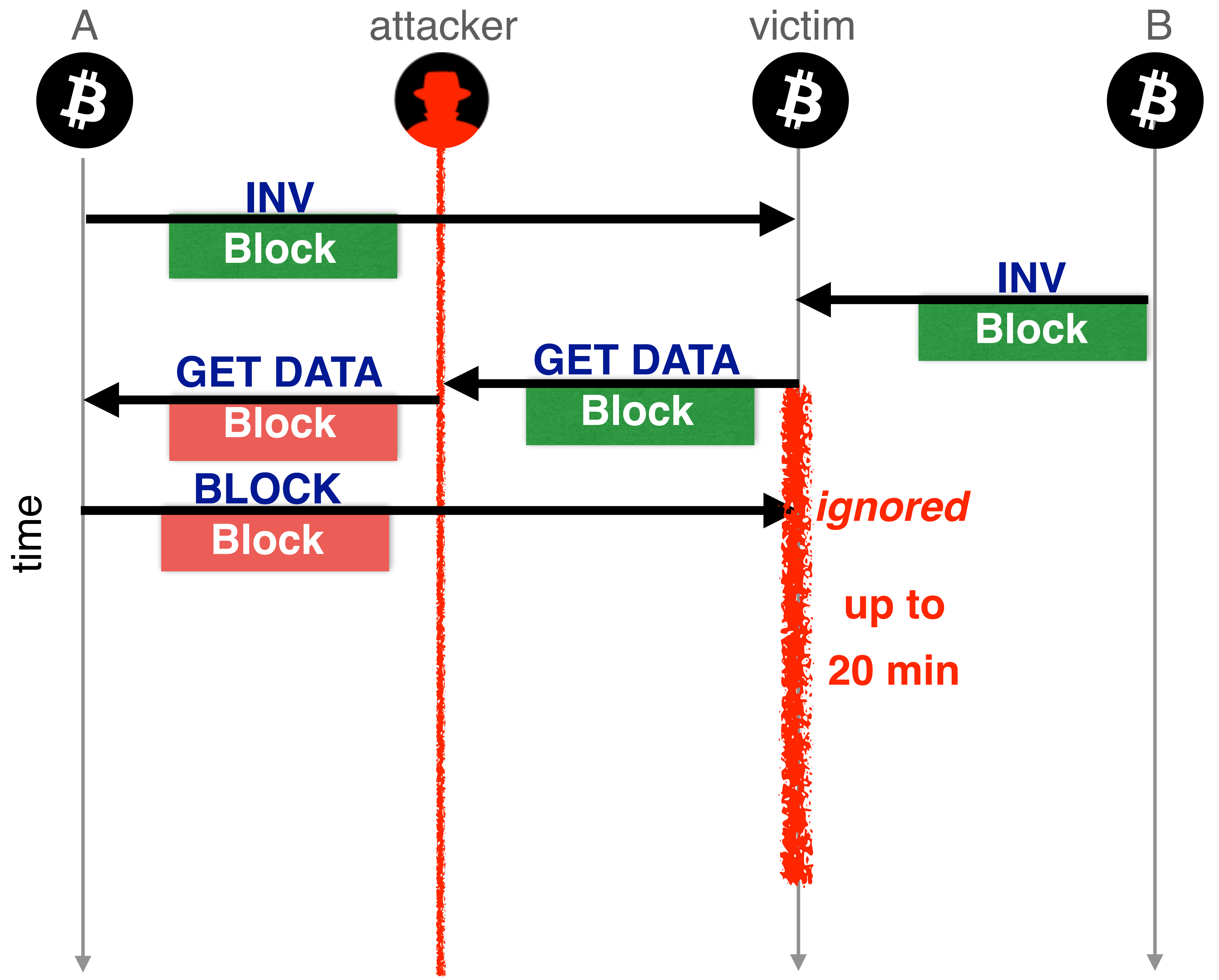
# The delivery of an older block triggers no error message at the victim



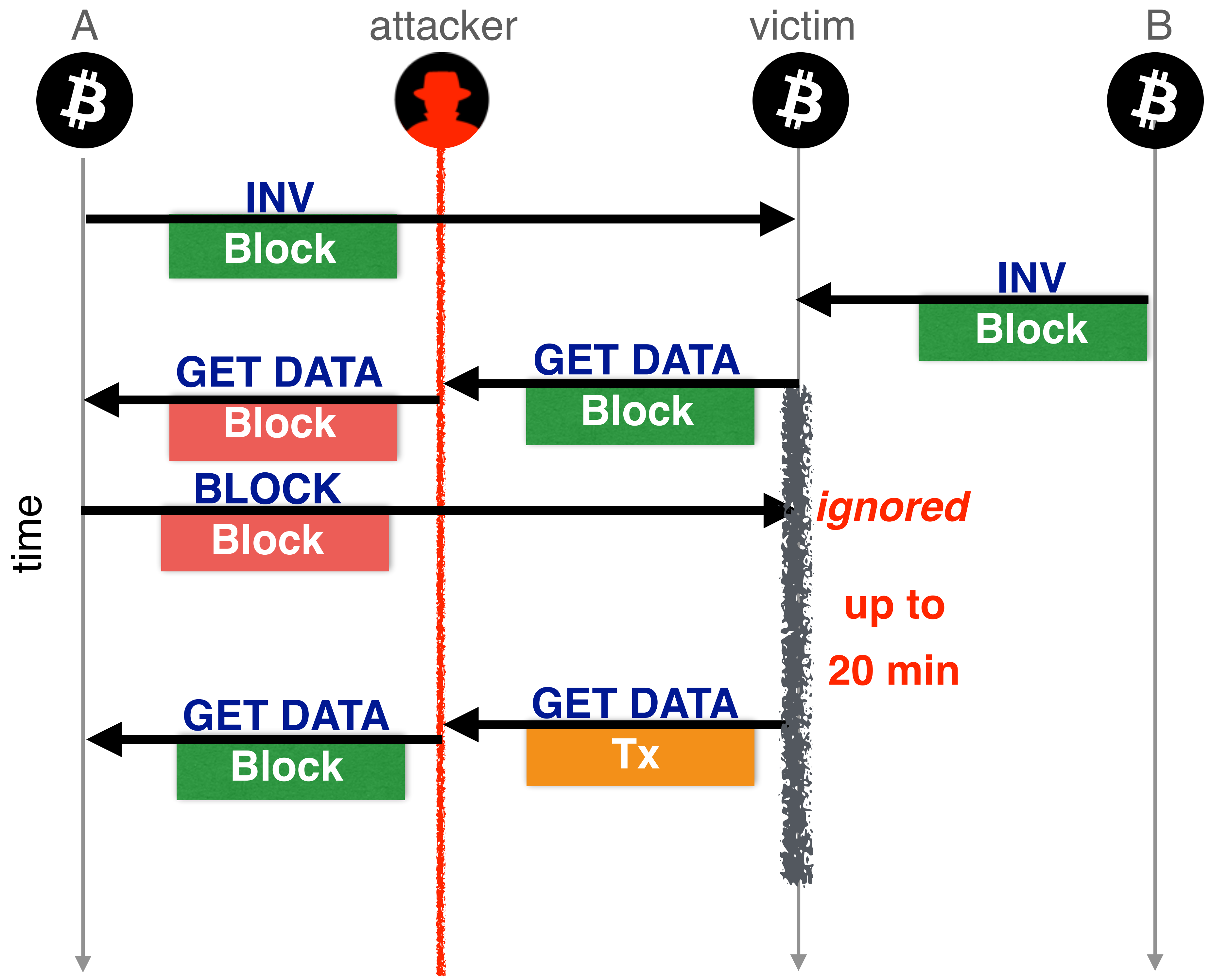
From there on, the victim will wait for 20 minutes for the actual block to be delivered



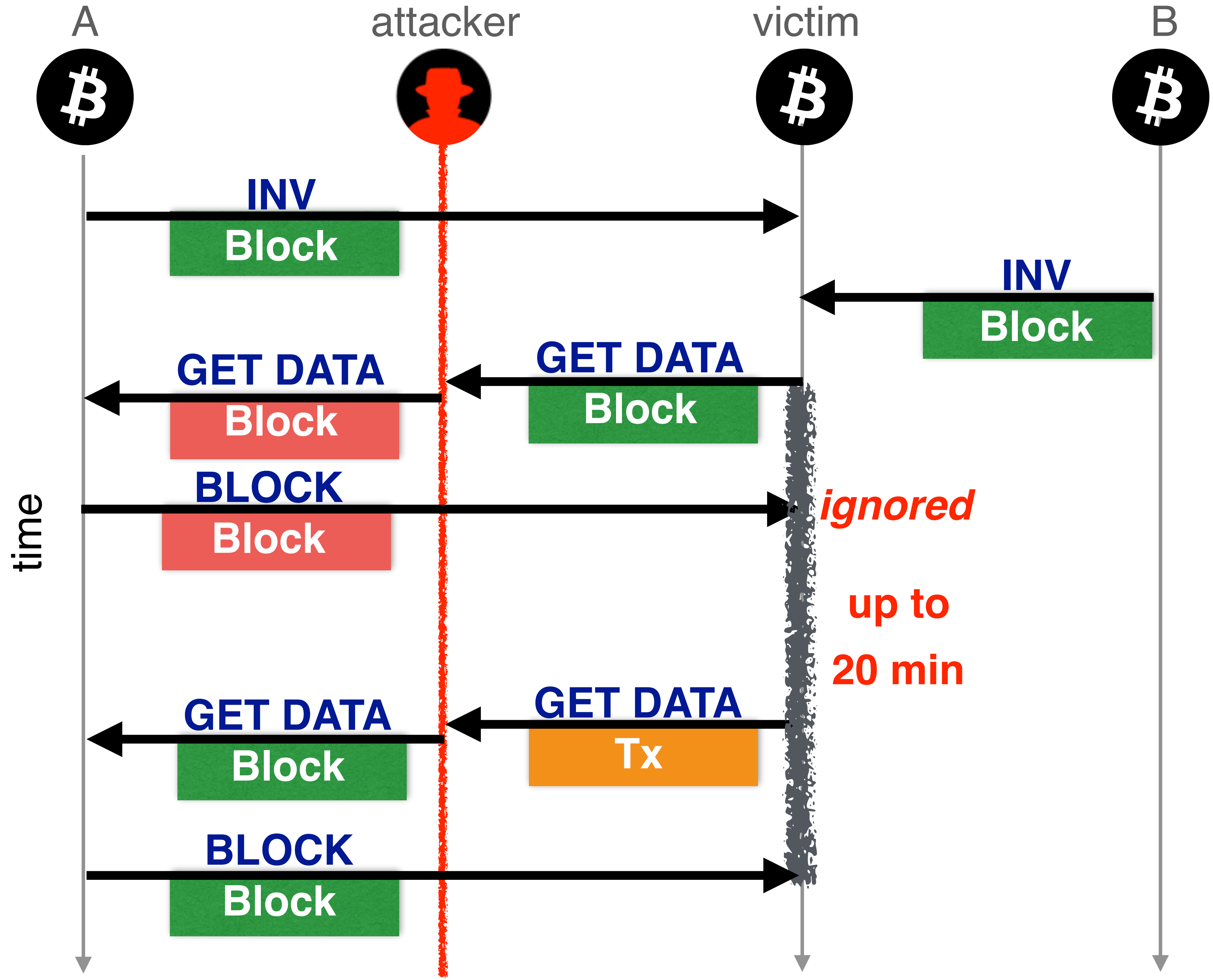
From there on, the victim will wait for 20 minutes for the actual block to be delivered



To keep the connection alive, the attacker can trigger the block delivery by modifying another **GETDATA** message



# Doing so, the block is delivered before the timeout and the attack goes undetected (and could be resumed)



# We evaluated the delay attack in terms of effectiveness and practicality

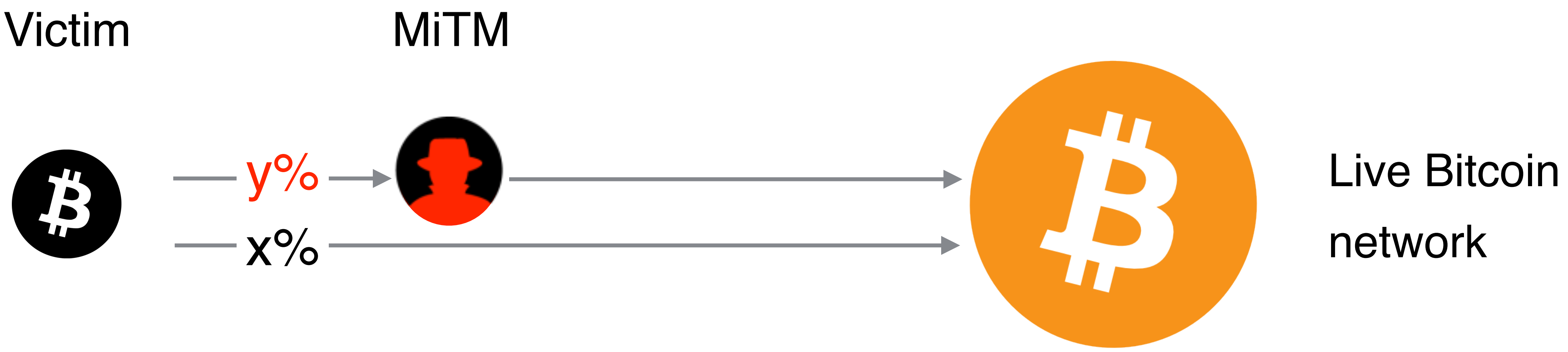
Effectiveness

How much time does
the victim stay uniformed?

Practicality

Is it likely to happen?

# We performed the attack on a percentage of a node's connections (*)



(*) software available online: https://btc-hijack.ethz.ch/

The attacker can keep the victim uninformed
for most of its uptime  while staying under the radar

The attacker can keep the victim uninformed for most of its uptime while staying under the radar

even if the attacker intercepts
a fraction of the node connection

| | |
|---|---|
| % intercepted connections | 50% |

| | |
|---|---|
| % intercepted connections | 50% |
| % time victim does not have the most recent block | 63.2% |

# The vast majority of the Bitcoin network is at risk

| | |
|---|---|
| % intercepted connections | 50% |
| % time victim does not have the most recent block | 63.2% |
| % nodes vulnerable to attack | 67.9% |

# Hijacking Bitcoin

Routing Attacks on Cryptocurrencies

1 **Background**

BGP & Bitcoin

2 **Partitioning attack**

splitting the network

3 **Delay attack**

slowing the network down

4 **Countermeasures**

short-term & long-term

# Countermeasures exist for both types of attacks

# Delay attacks could be prevented with short and long-term countermeasures

| | |
|---|---|
| Long-term | Use end-to-end encryption or MAC<br>prevent delay attacks (not partition attacks) |

# Delay attacks could be prevented with short and long-term countermeasures

| | |
|---|---|
| Long-term | Use end-to-end encryption or MAC<br>prevent delay attacks (not partition attacks) |
| Short-term | Routing-aware peer selection<br>reduce risk of having one ISP seeing all connections |

# Countermeasures against partition attacks exist

Short-term

Host all Bitcoin clients in /24 prefixes

reduce of a successful hijack

# Countermeasures against partition attacks exist

| | |
|---|---|
| Short-term | Host all Bitcoin clients in /24 prefixes<br>reduce chances of a successful hijack |
| Long-term | Deploy secure routing protocols<br>prevent partition attacks |

# Countermeasures against partition attacks exist

But are impractical

Host all Bitcoin clients in /24 prefixes

Deploy secure routing protocols

# Countermeasures against partition attacks exist

But are impractical

Host all Bitcoin clients in /24 prefixes

increase BGP routing tables

Deploy secure routing protocols

# Countermeasures against partition attacks exist

But are impractical

Host all Bitcoin clients in /24 prefixes

increase BGP routing tables

Deploy secure routing protocols

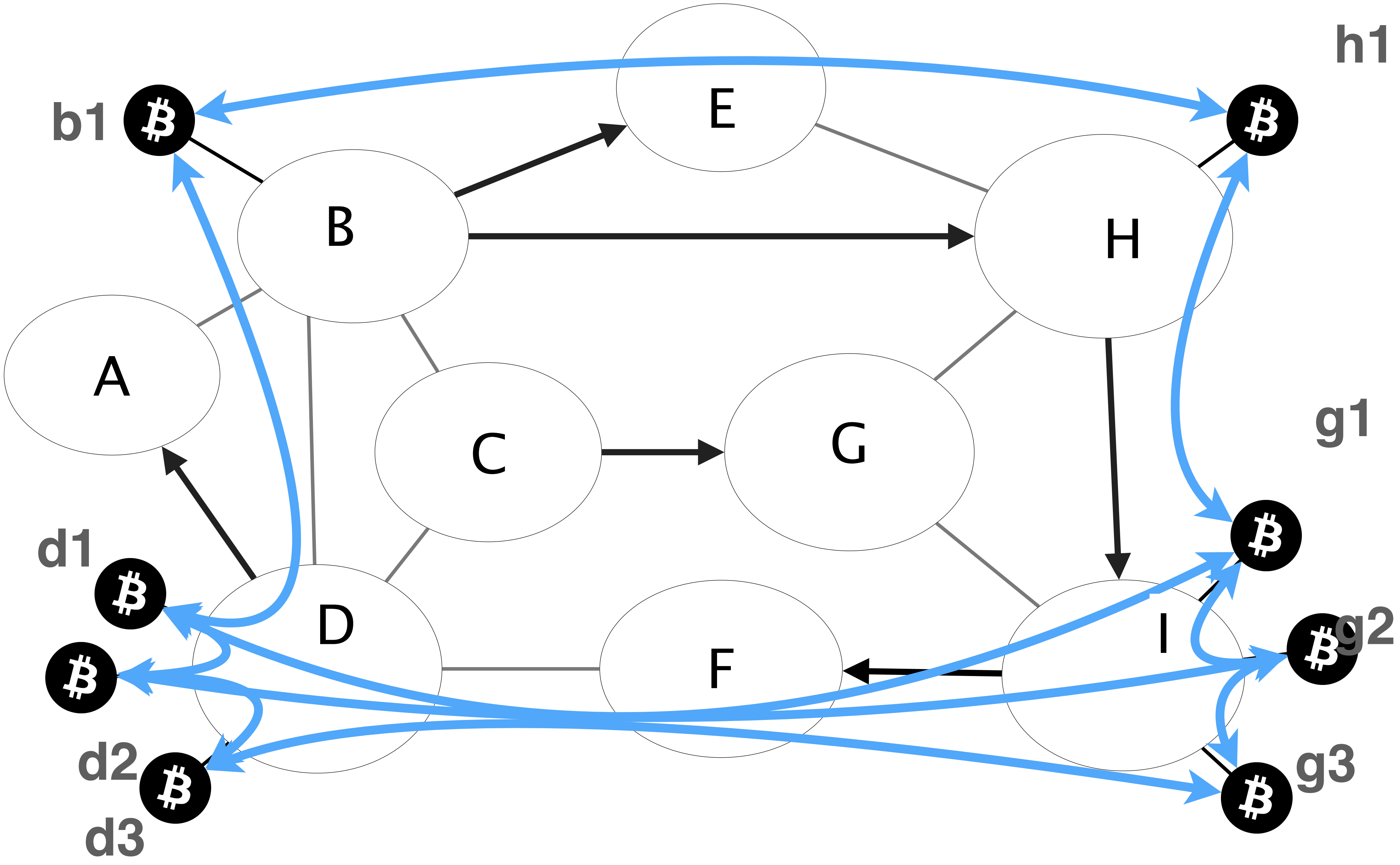
ISP collaboration required

Build additional secure channel to allow communication even if  the Bitcoin network is partitioned

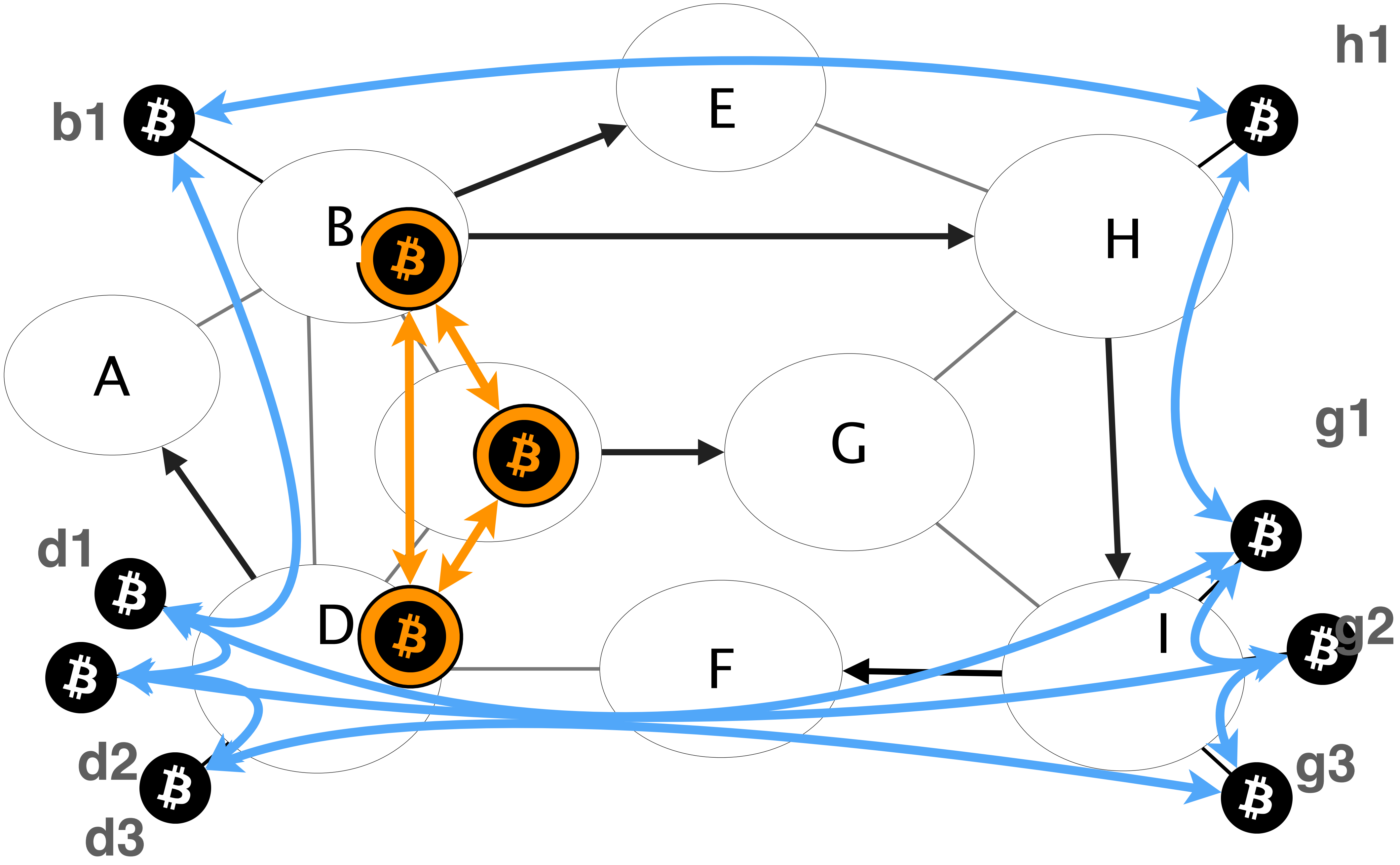
SABRE = Secure Relay Location + Robust Design

SABRE = Secure Relay Location + Robust Design

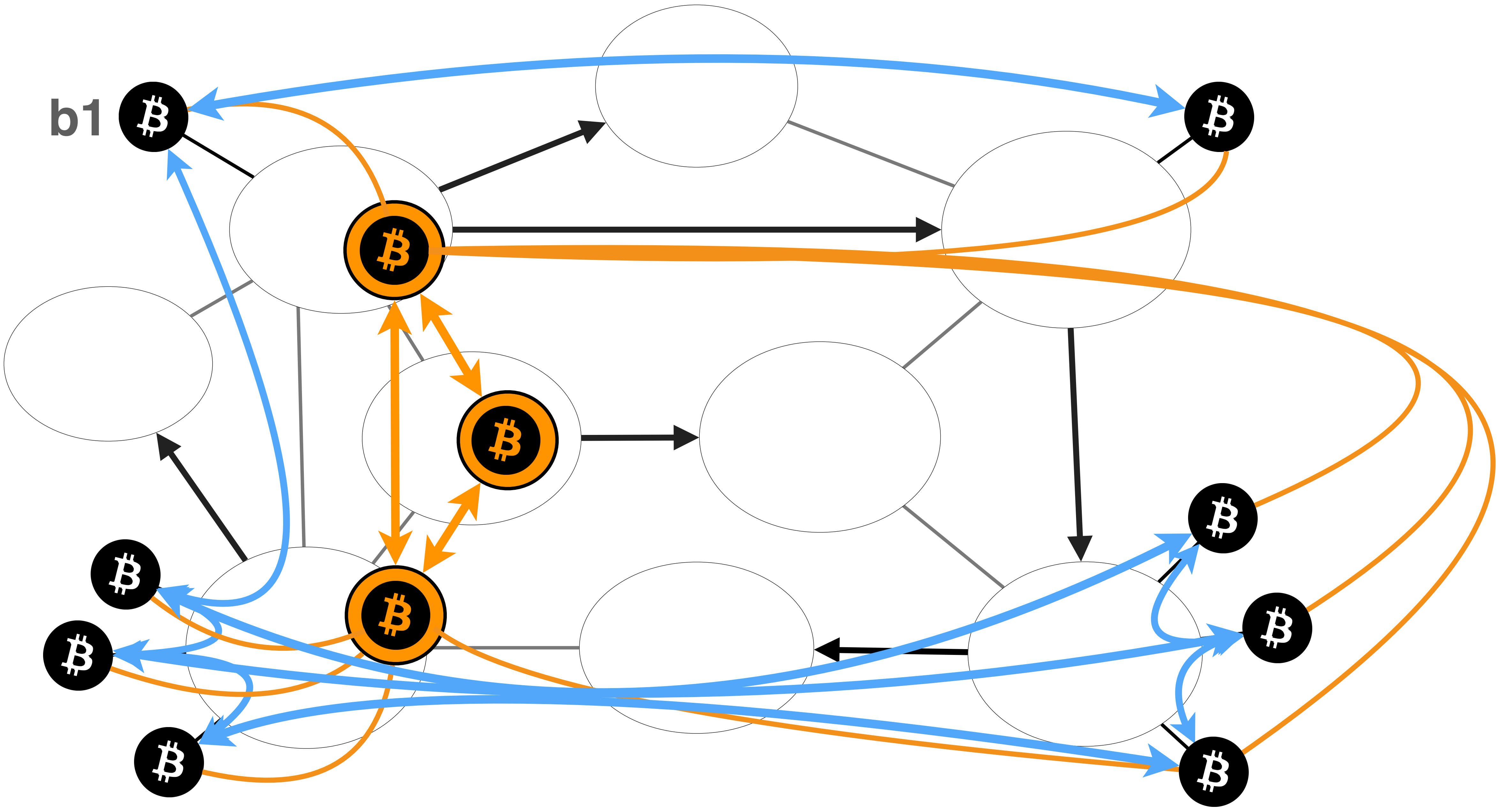
add few clients that connect to each other and to all other clients


h1
b1
E
B
H
A
C
G
g1
d1
D
F
I
g2
d2
g3
d3

# SABRE: Additional relay network of relay nodes



# Clients connect to at least one relay node



SABRE = Secure Relay Location + Robust Design

SABRE = Secure Relay Location + Robust Design

additional nodes protected
against hijacking attacks

SABRE = Secure Relay Location + **Robust Design**

Open and Resilient against DDoS attacks

SABRE = Secure Relay Location + Robust Design

# Secure Relay Placement

nodes in /24 prefix

peering ASes with no customers

k-connected graph of relays

relays cover most clients

# Secure Relay Placement

nodes in /24 prefix

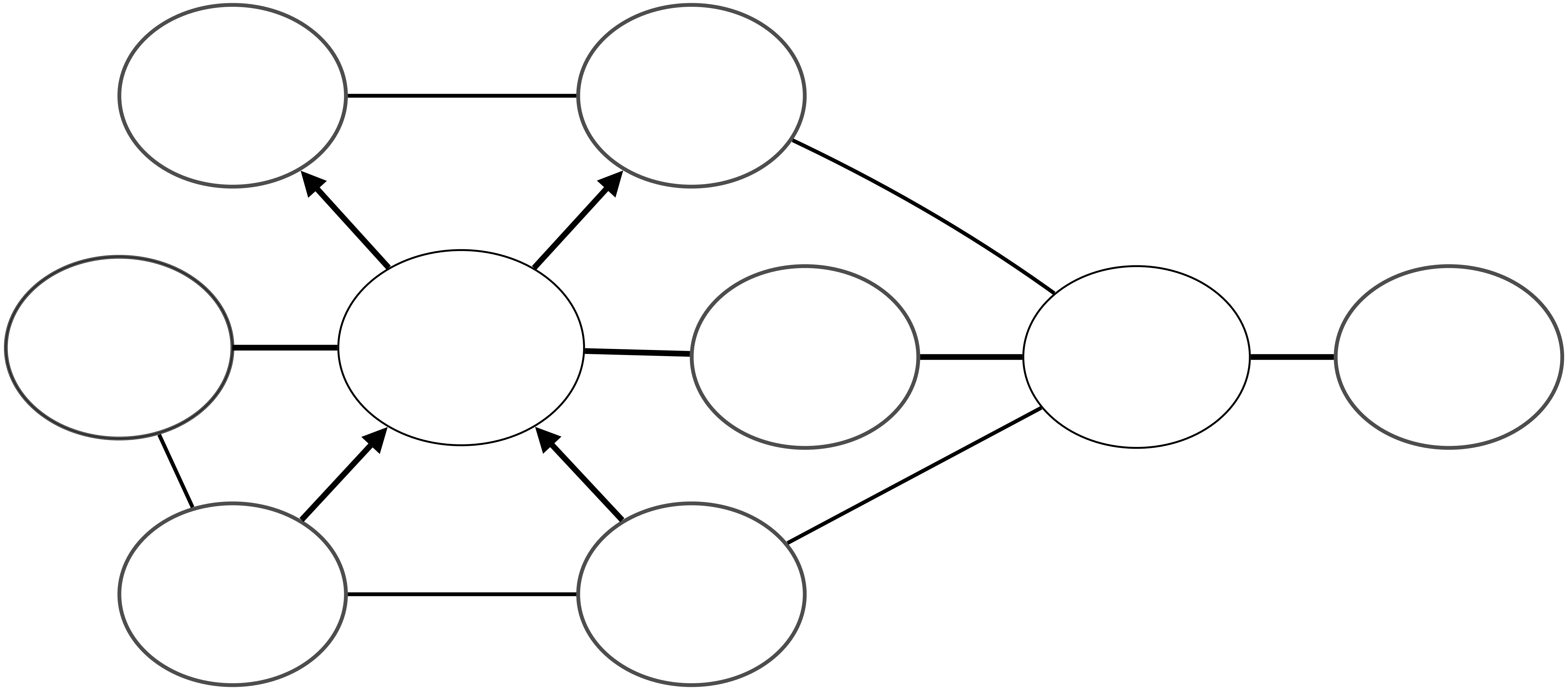
malicious prefix in competition with legitimate ones

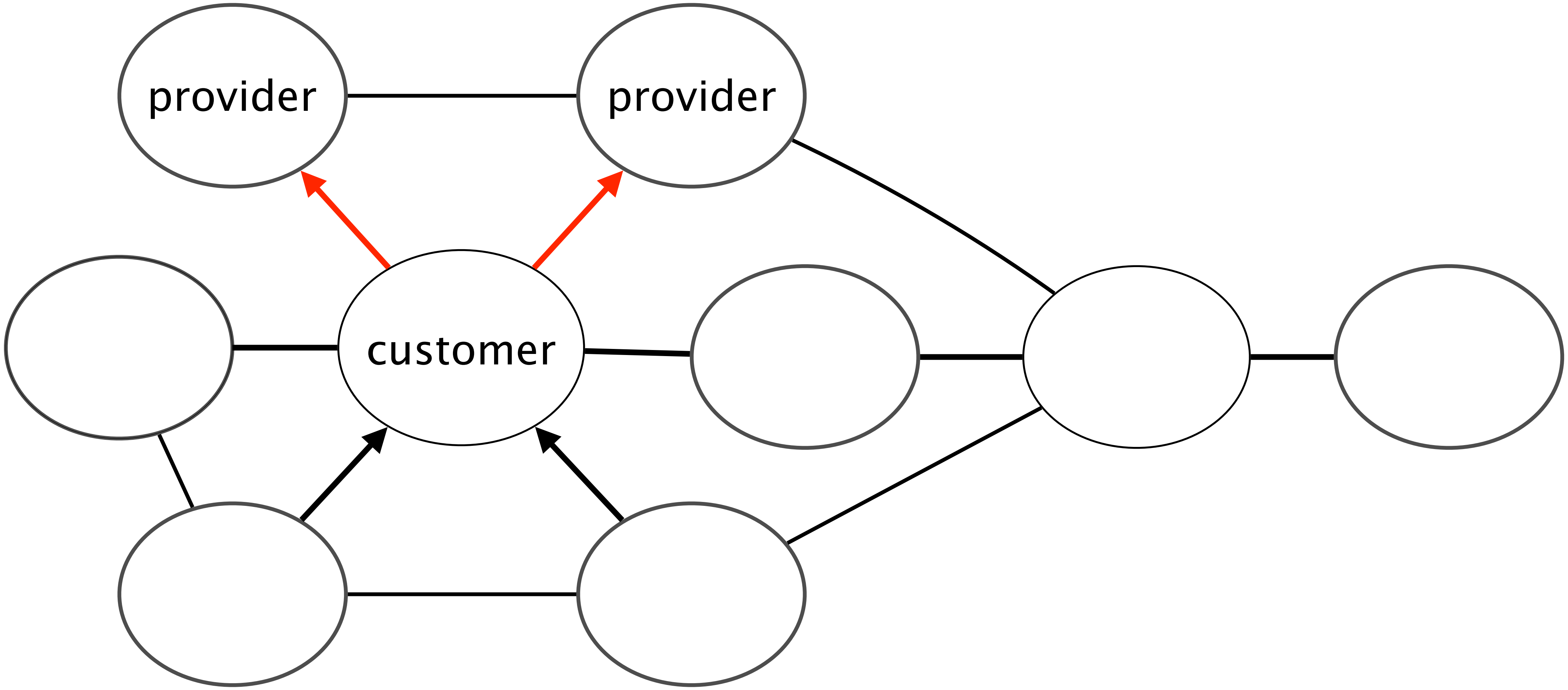
peering ASes with no customers

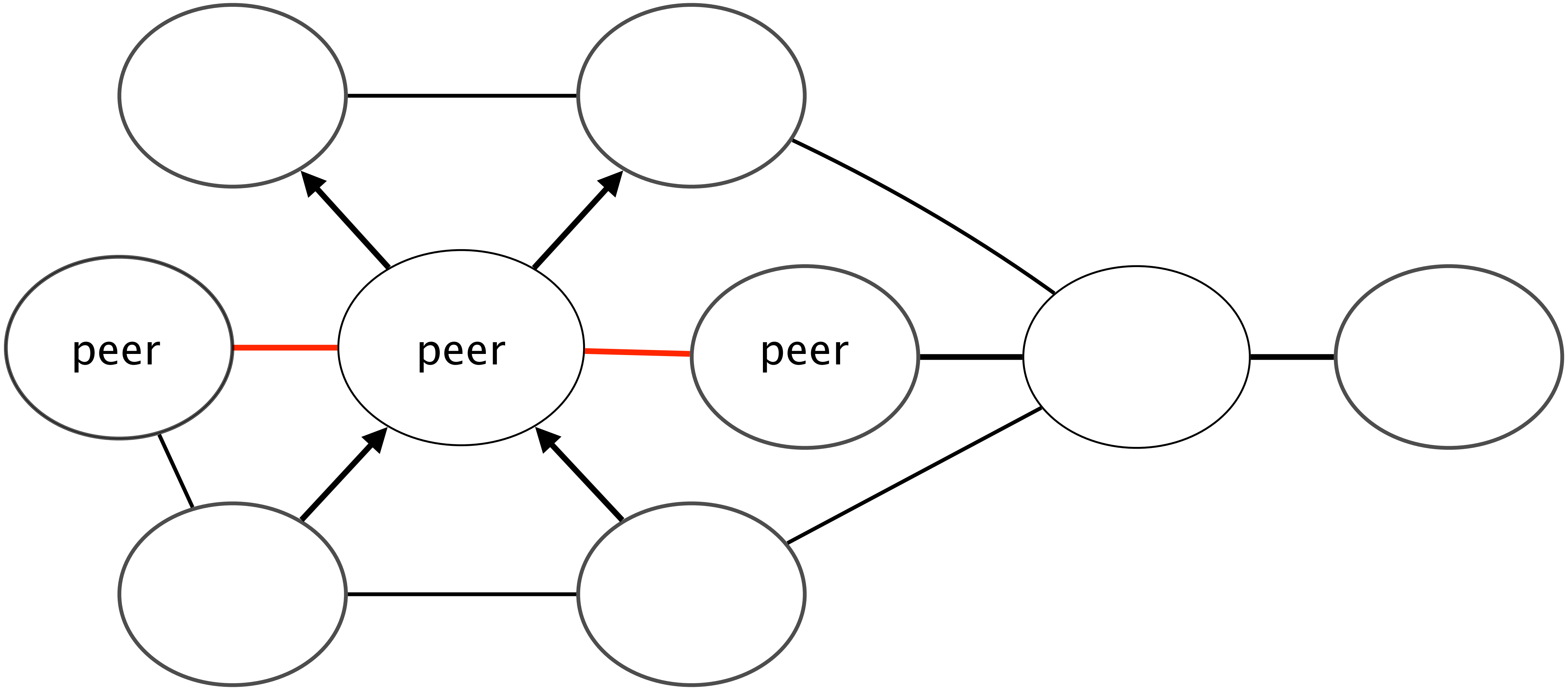
k-connected graph of relays

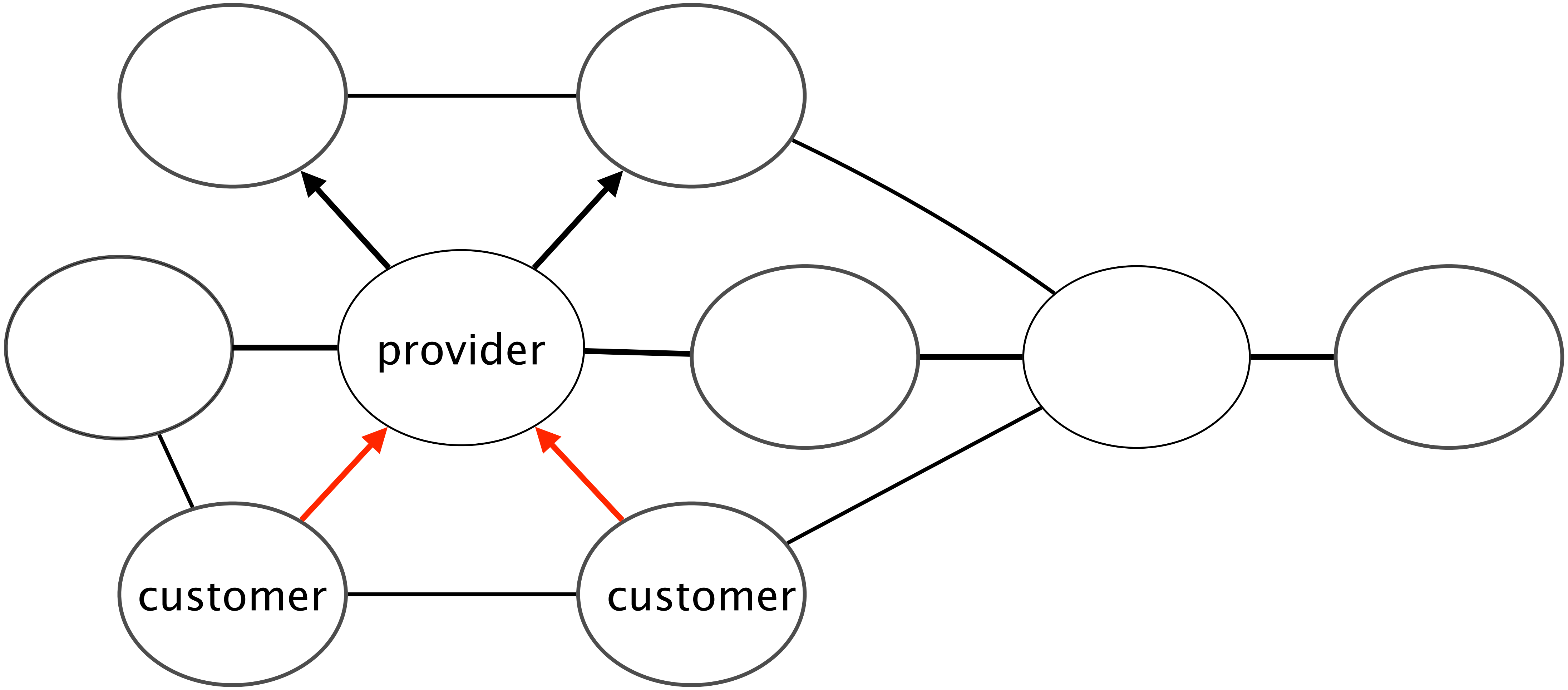
relays cover most clients

# Arrows show the money flow

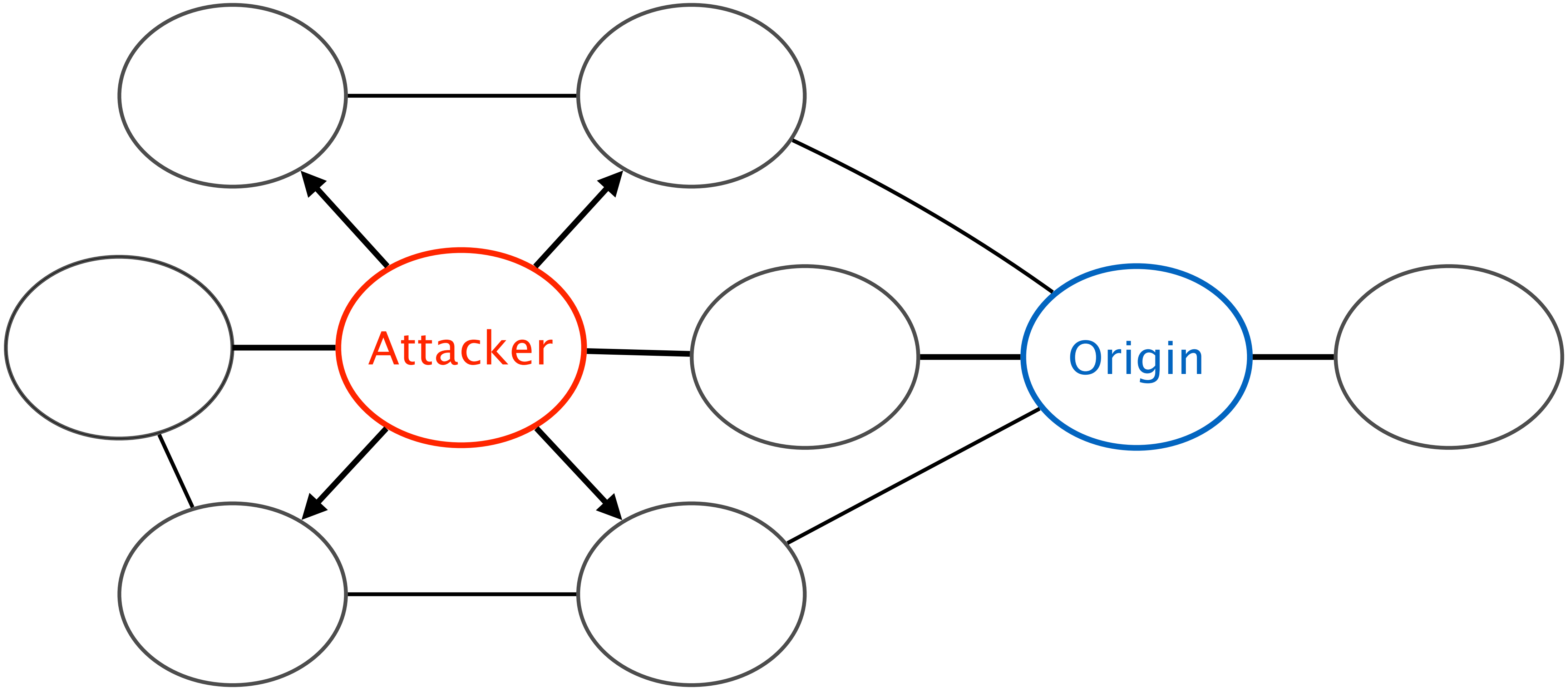



provider
provider
customer

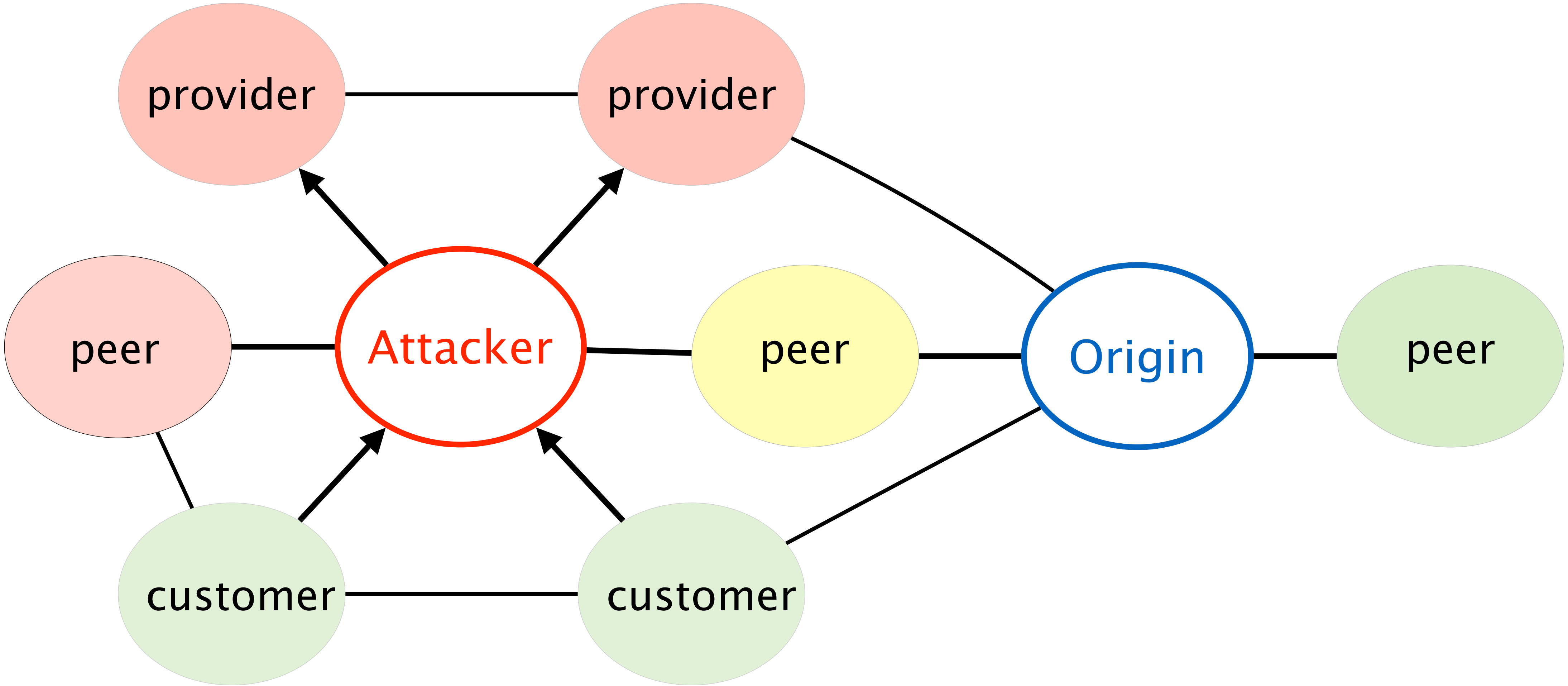

peer
peer
peer


provider
customer
customer

# The attacker advertises same length prefix as the origin



# ~50% ASes would  follow the attacker’s advertisement



# Secure Relay Placement

nodes in /24 prefix

peering ASes with no customers

k-connected graph of relays

relays cover most clients

# Secure Relay Placement

nodes in /24 prefix

peering ASes with no customers

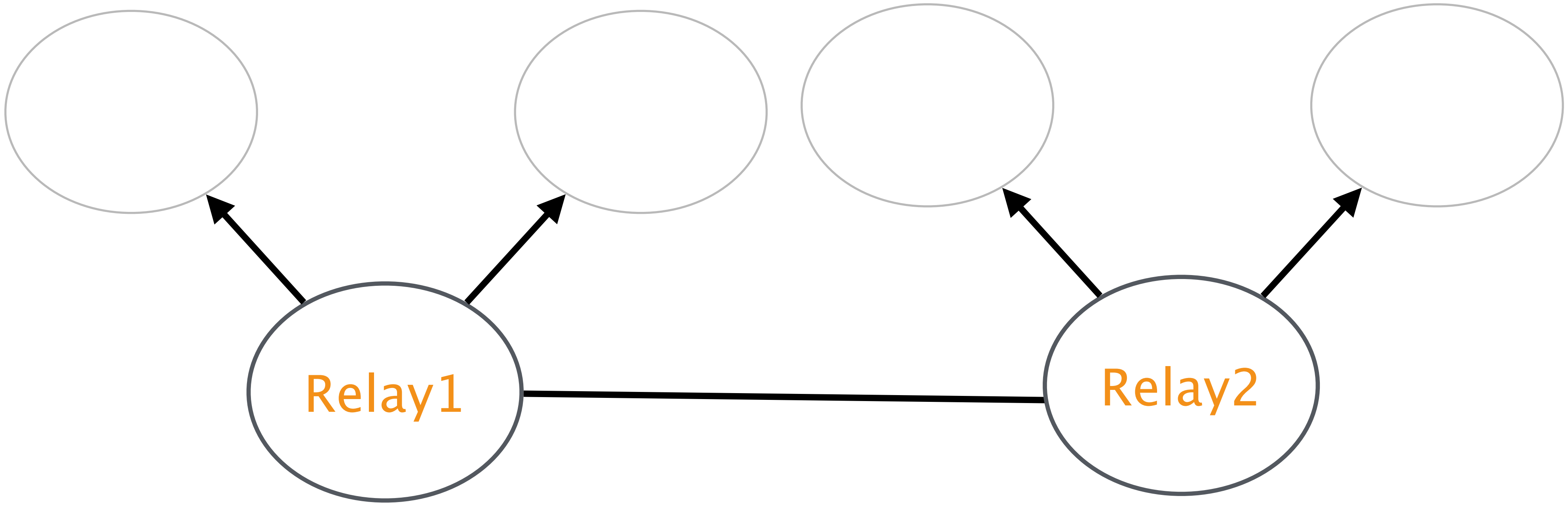
no strictly better prefix

advertisement exists

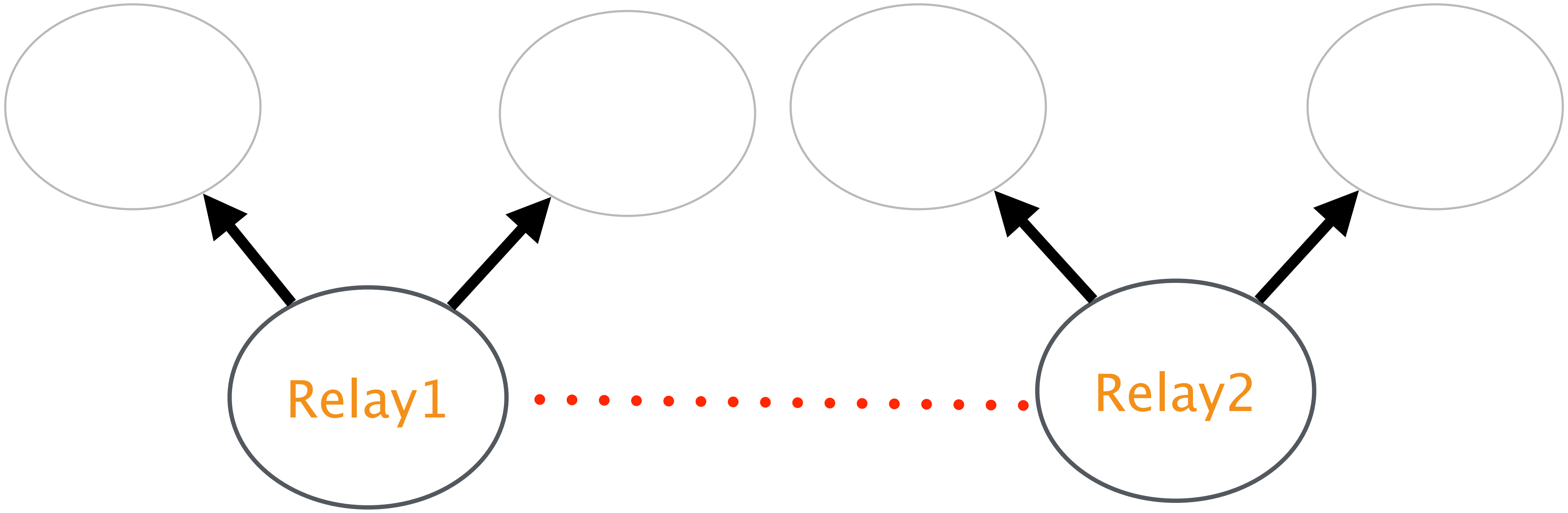
k-connected graph of relays

relays cover most clients

No strictly better advertisement exist



# Peering agreement can be revoked



# Secure Relay Placement

nodes in /24 prefix

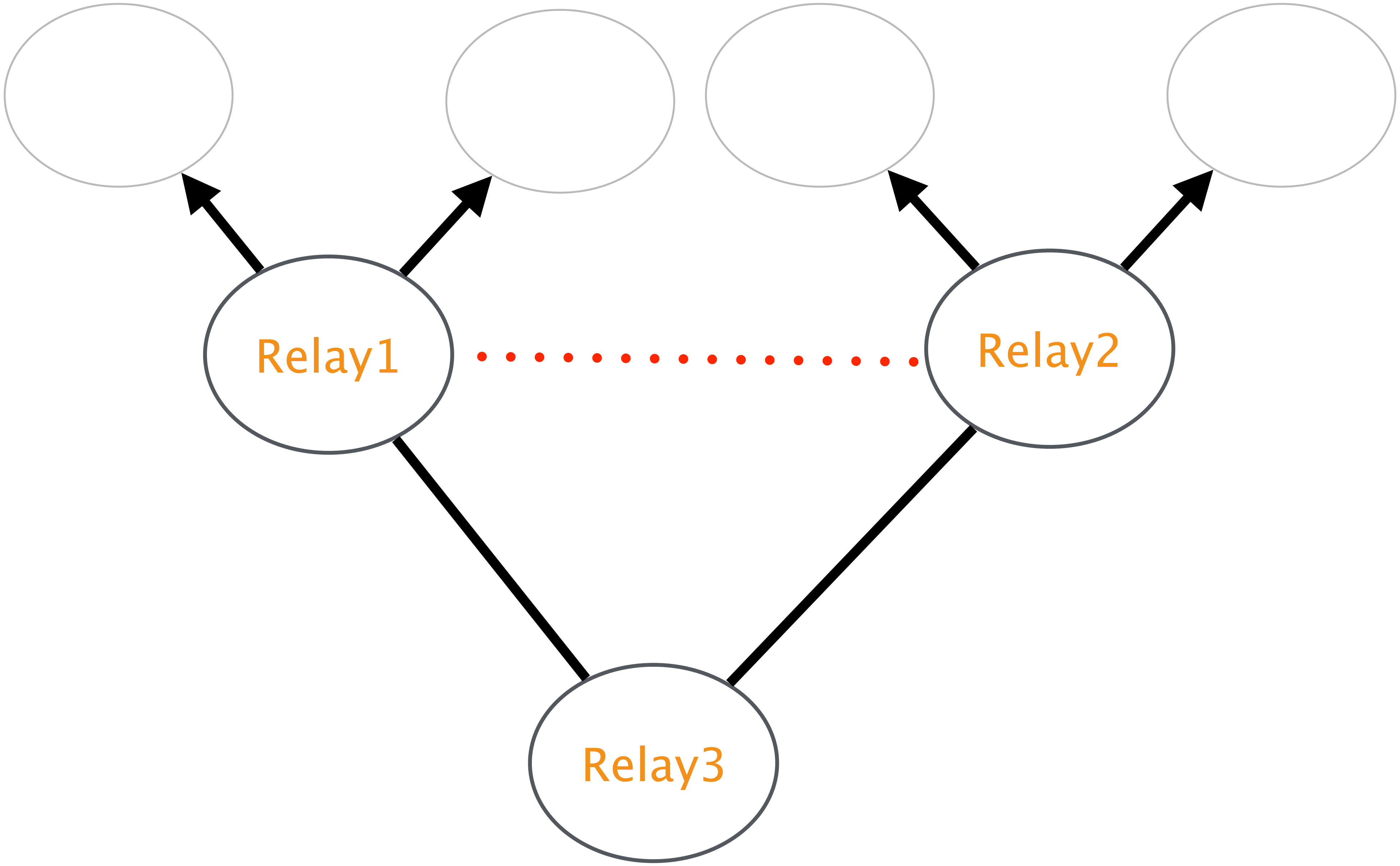
peering ASes with no customers

k-connected graph of relays

relays cover most clients

relay connectivity
is not affected by  any k cuts

# 2-connected graph retains connectivity



# Secure Relay Placement

nodes in /24 prefix

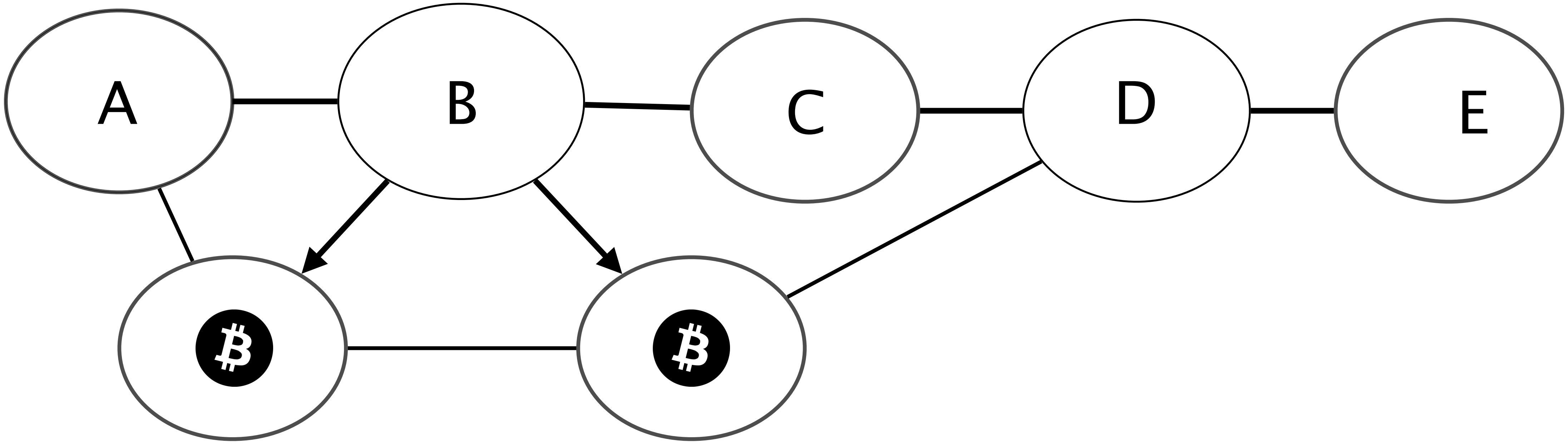
peering ASes with no customers

k-connected graph of relays

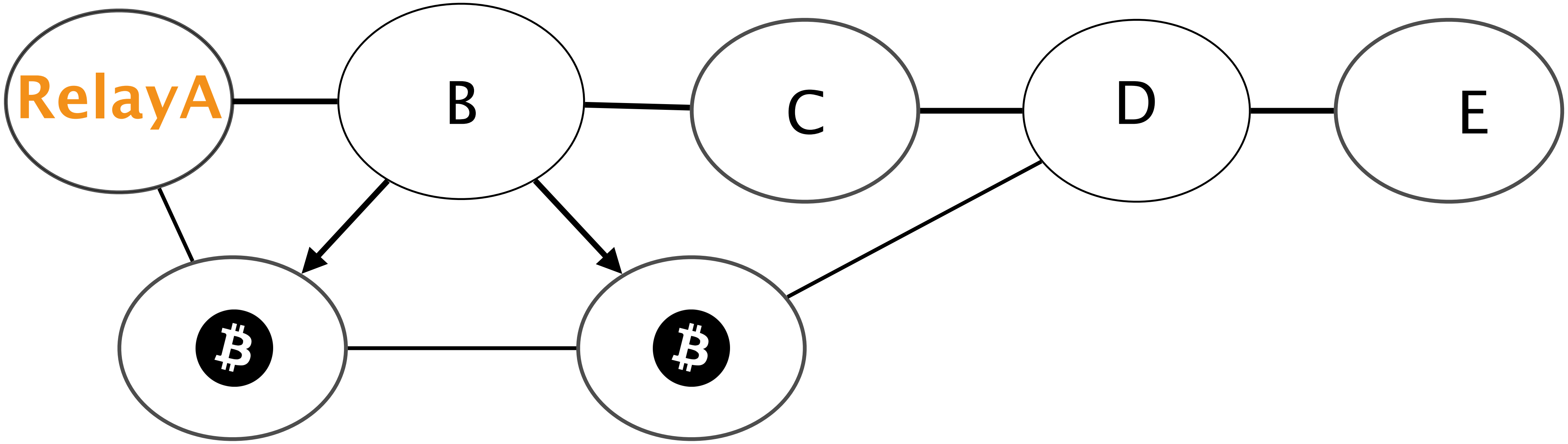
relays cover most clients

relays are in path that are more preferred than any alternative

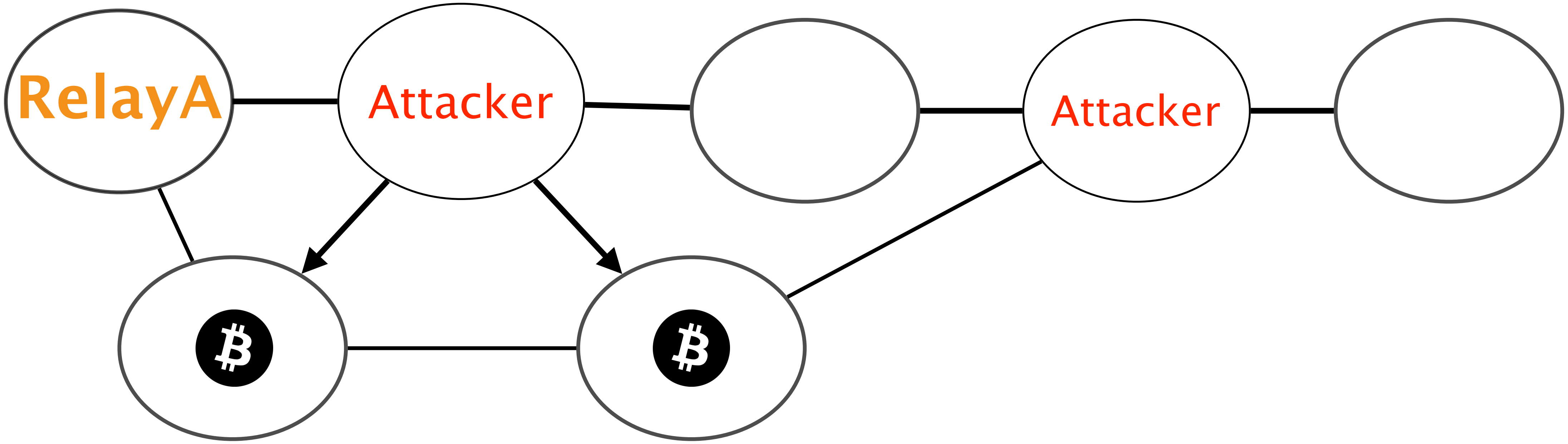
Where should we place a relay node to avoid interception of traffic from Bitcoin clients to this relay node?



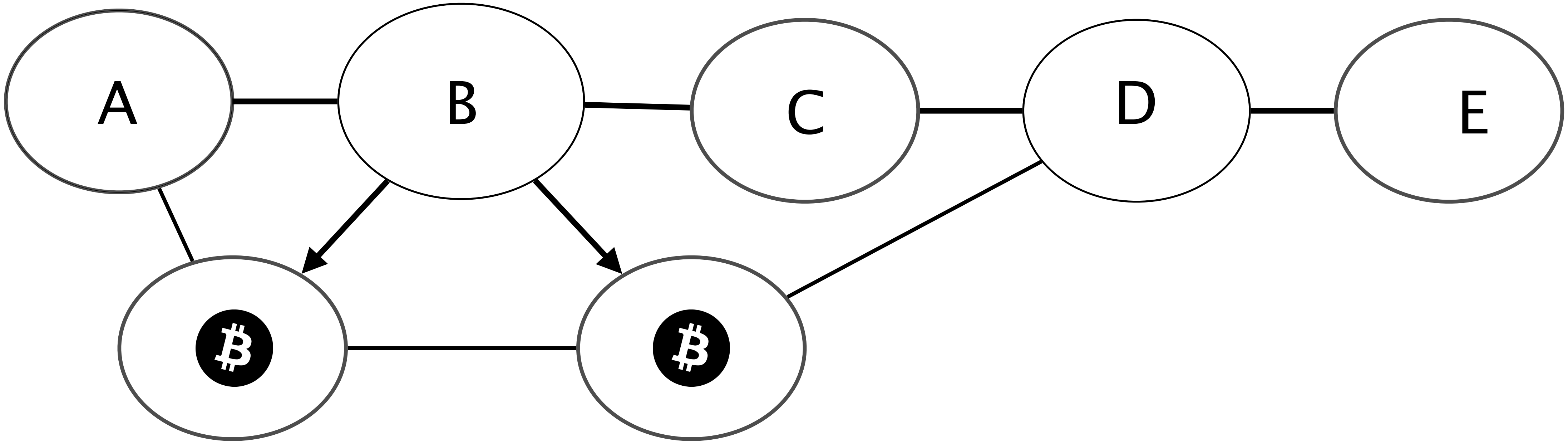
If Relay is hosted in ASA,



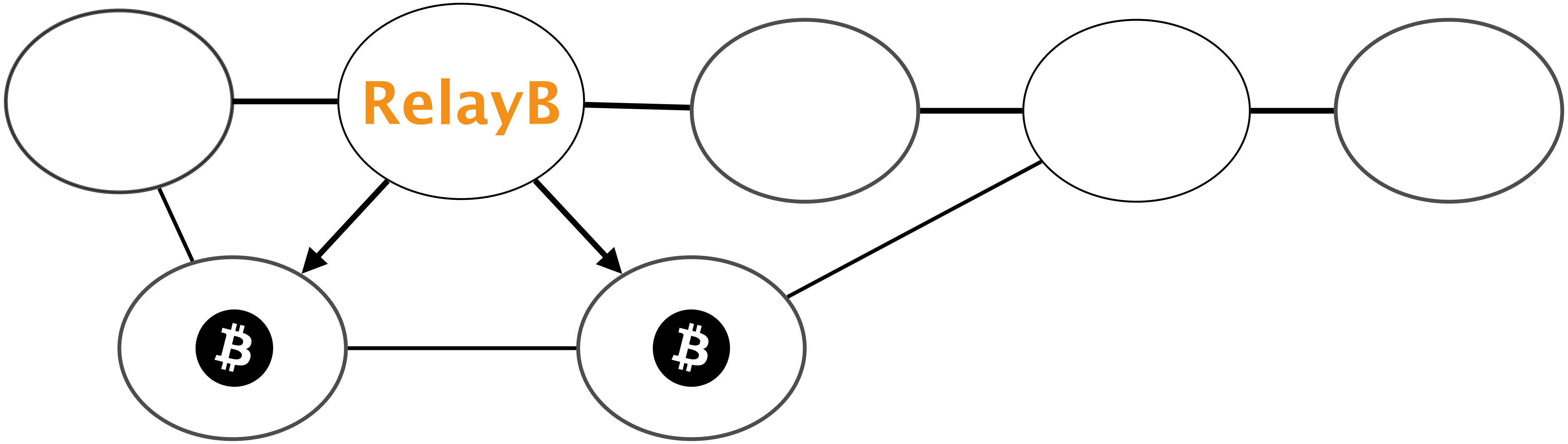
If Relay is hosted in ASA, there are two effective attackers



Where should we place a relay node to avoid interception of traffic from Bitcoin clients to this relay node?



If we place the relay to ASB, there is no effective attacker



# Secure Relay Placement

nodes in /24 prefix

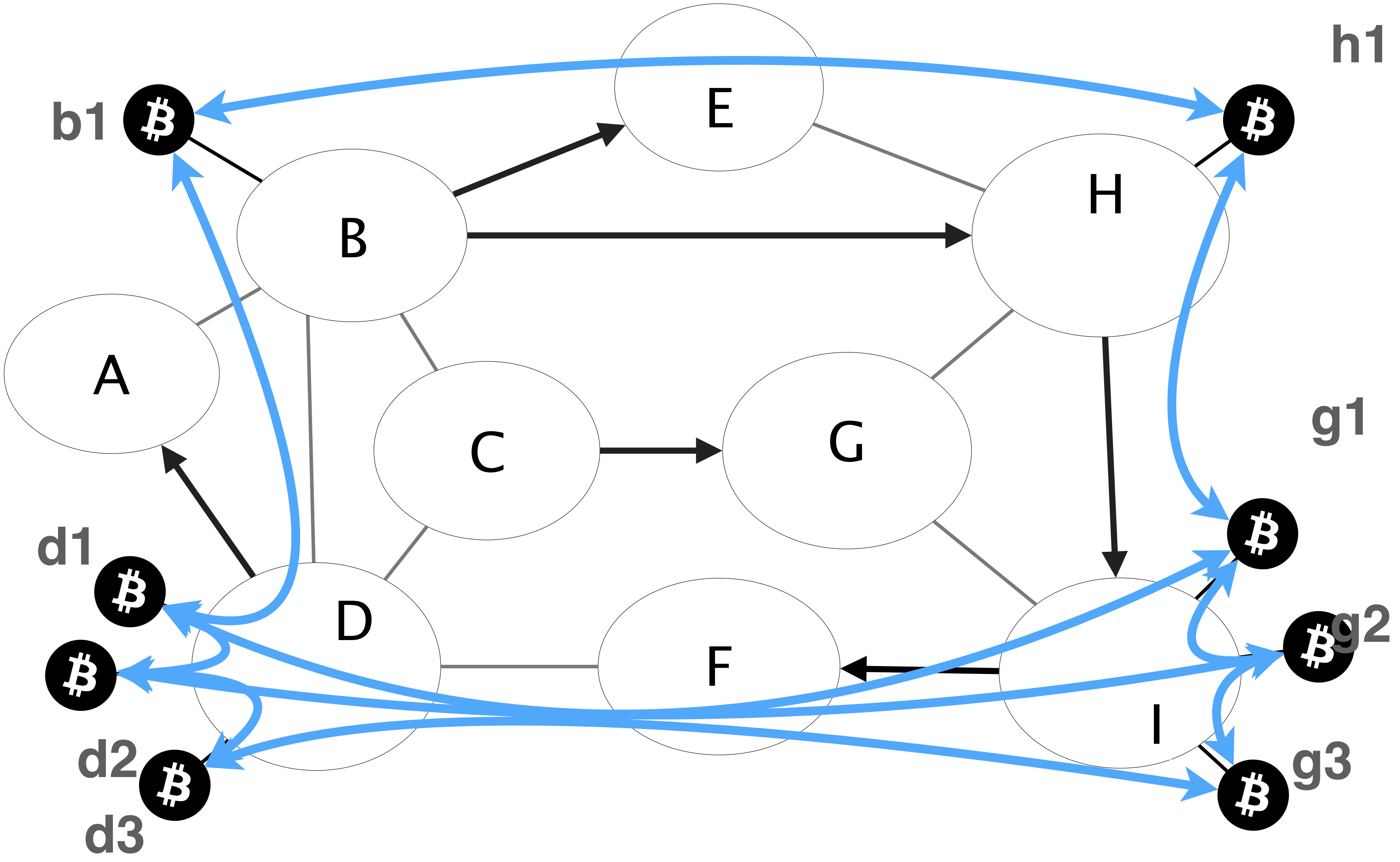
peering ASes with no customers

k-connected graph of relays

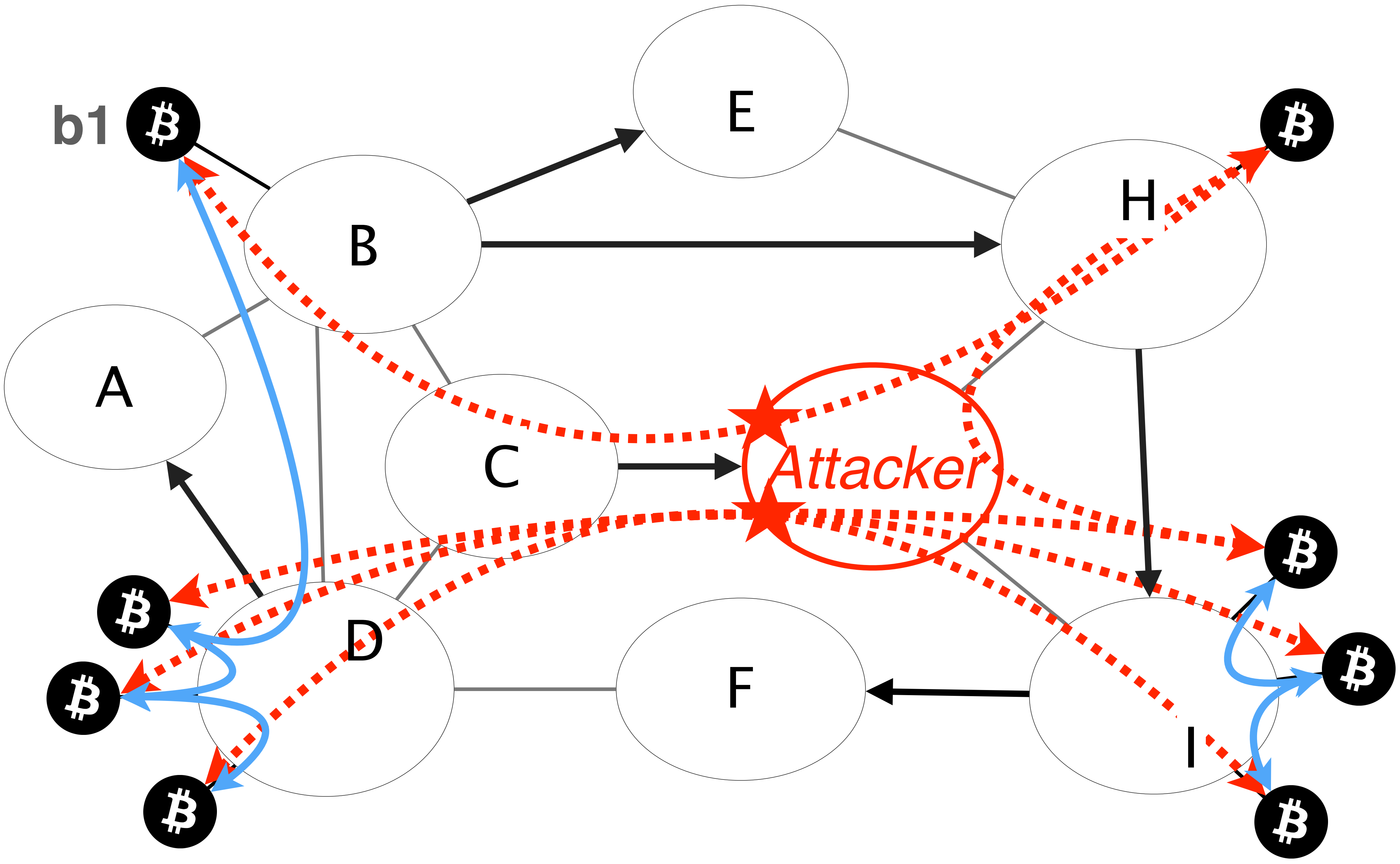
relays cover most clients

# How SABRE helps in case of an attack?

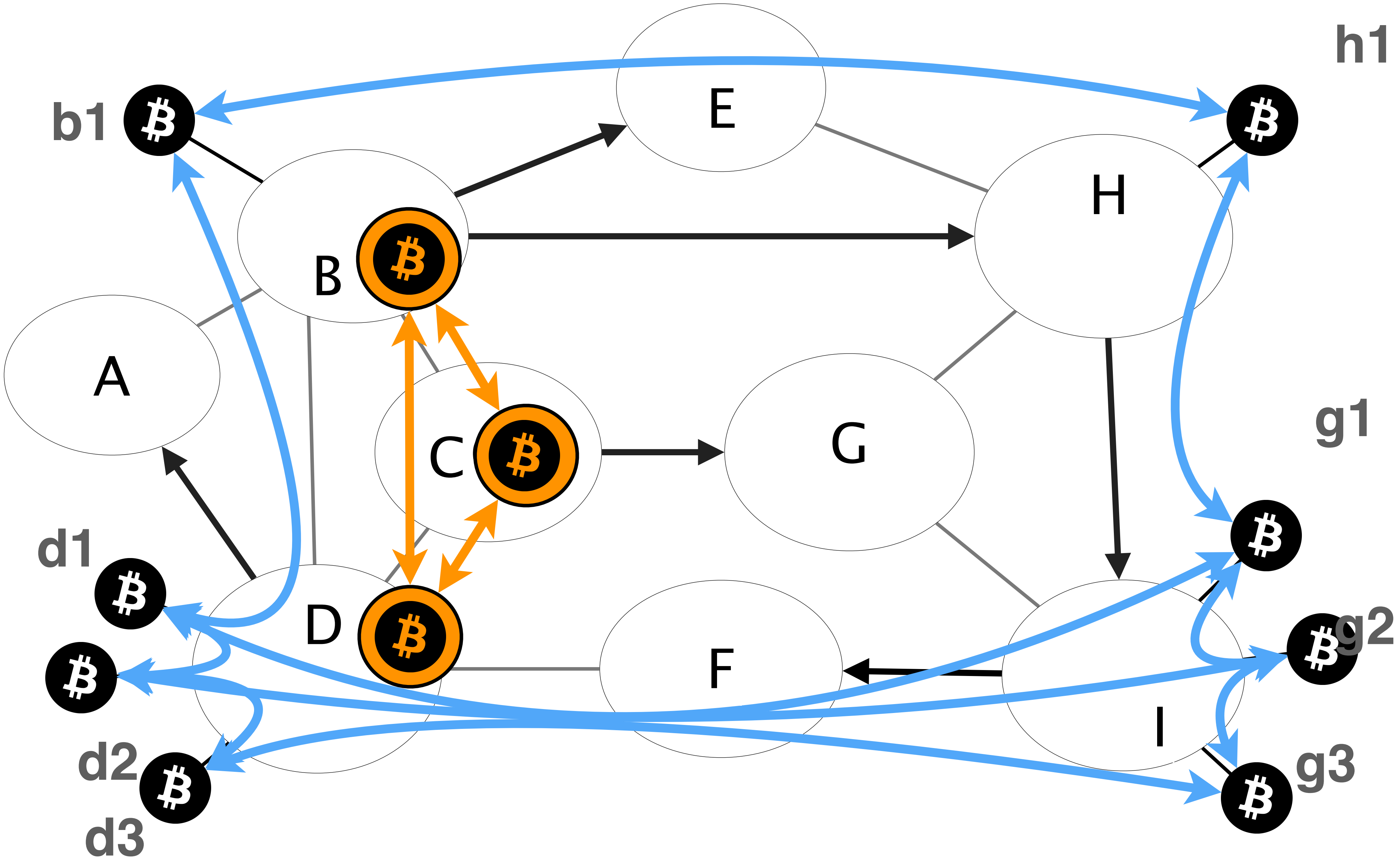
# Let’s see SABRE in practice



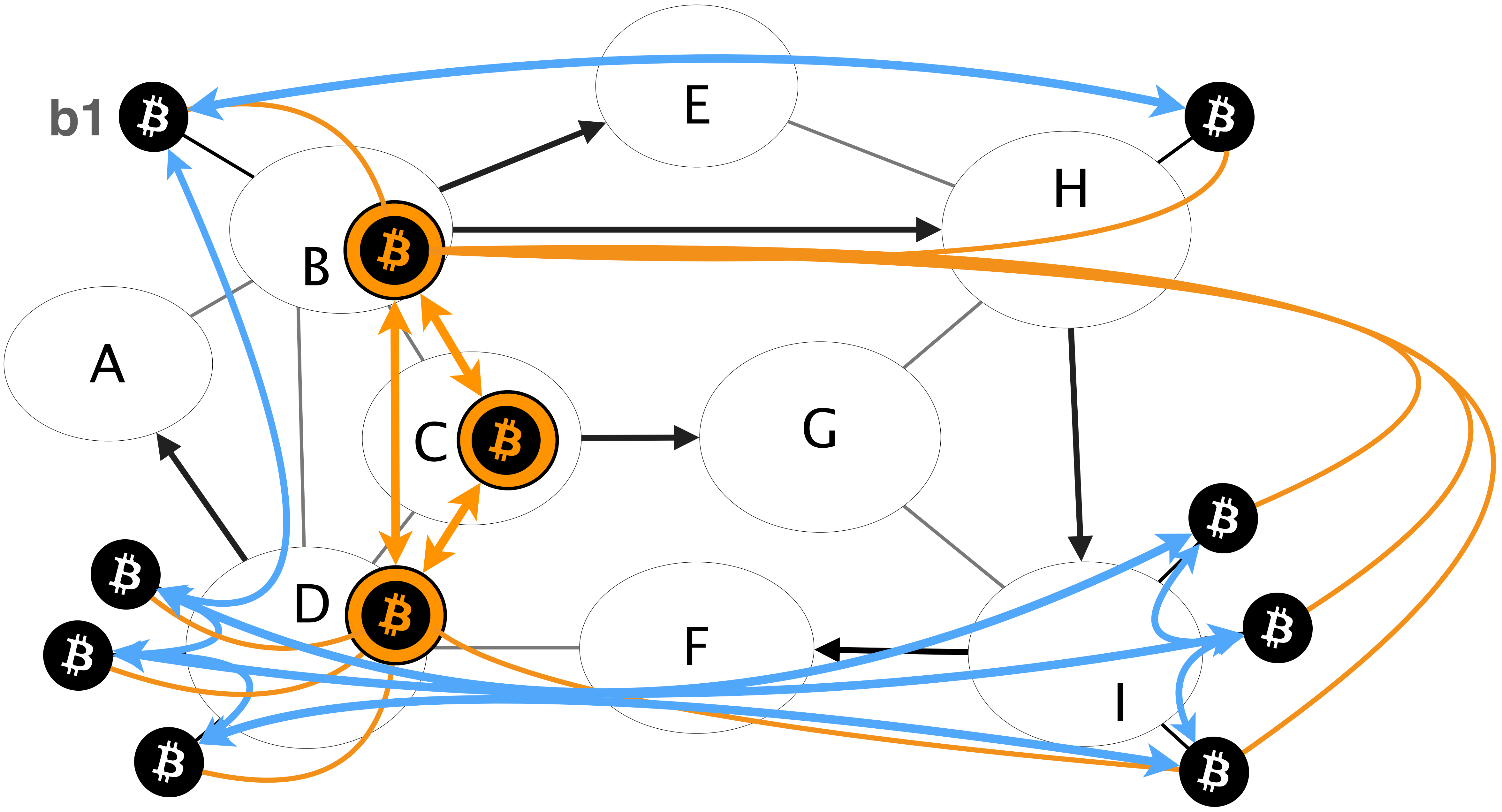
Attacker hijacks and drops connection between components



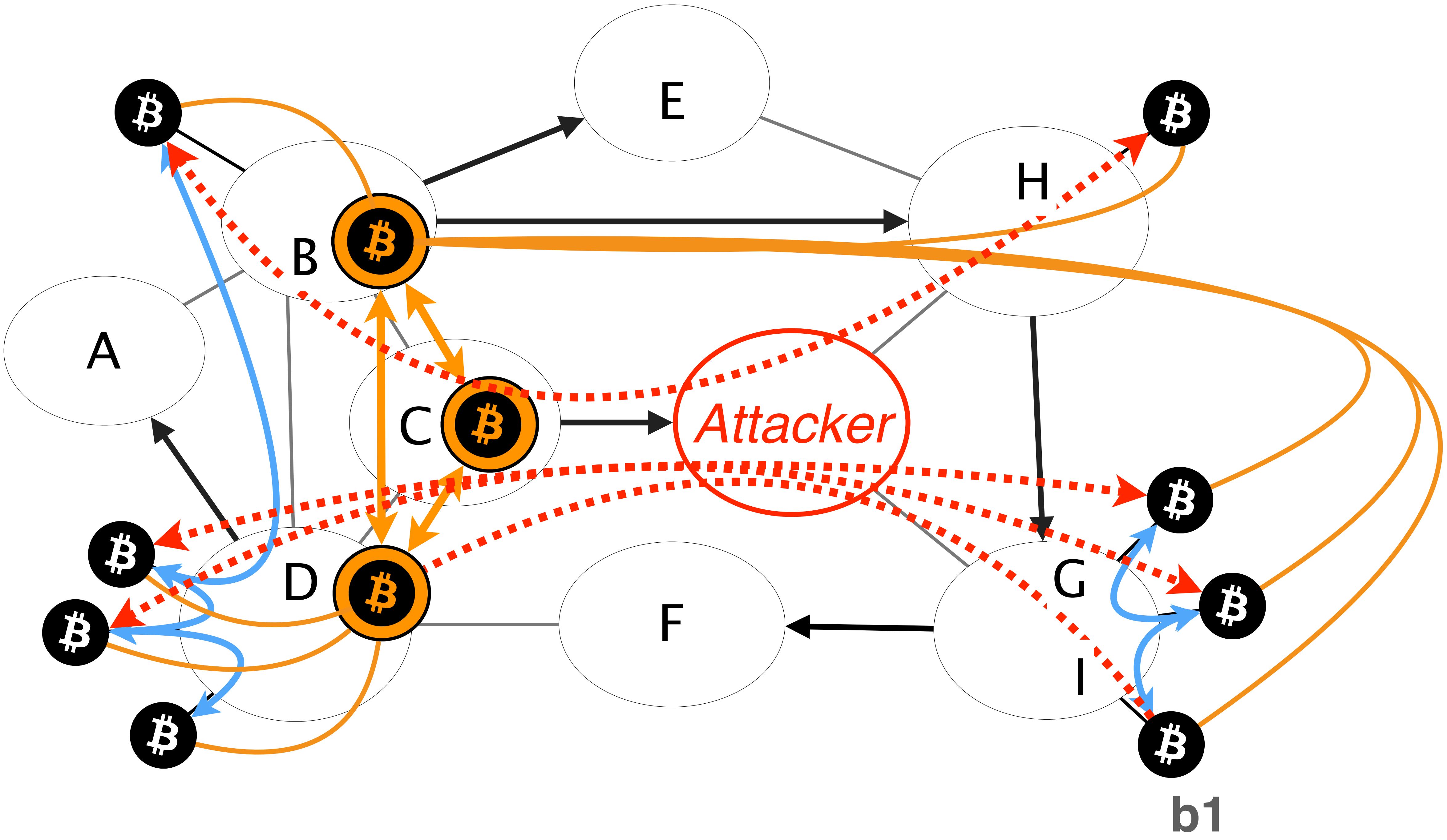
# SABRE: Additional relay network of relay nodes



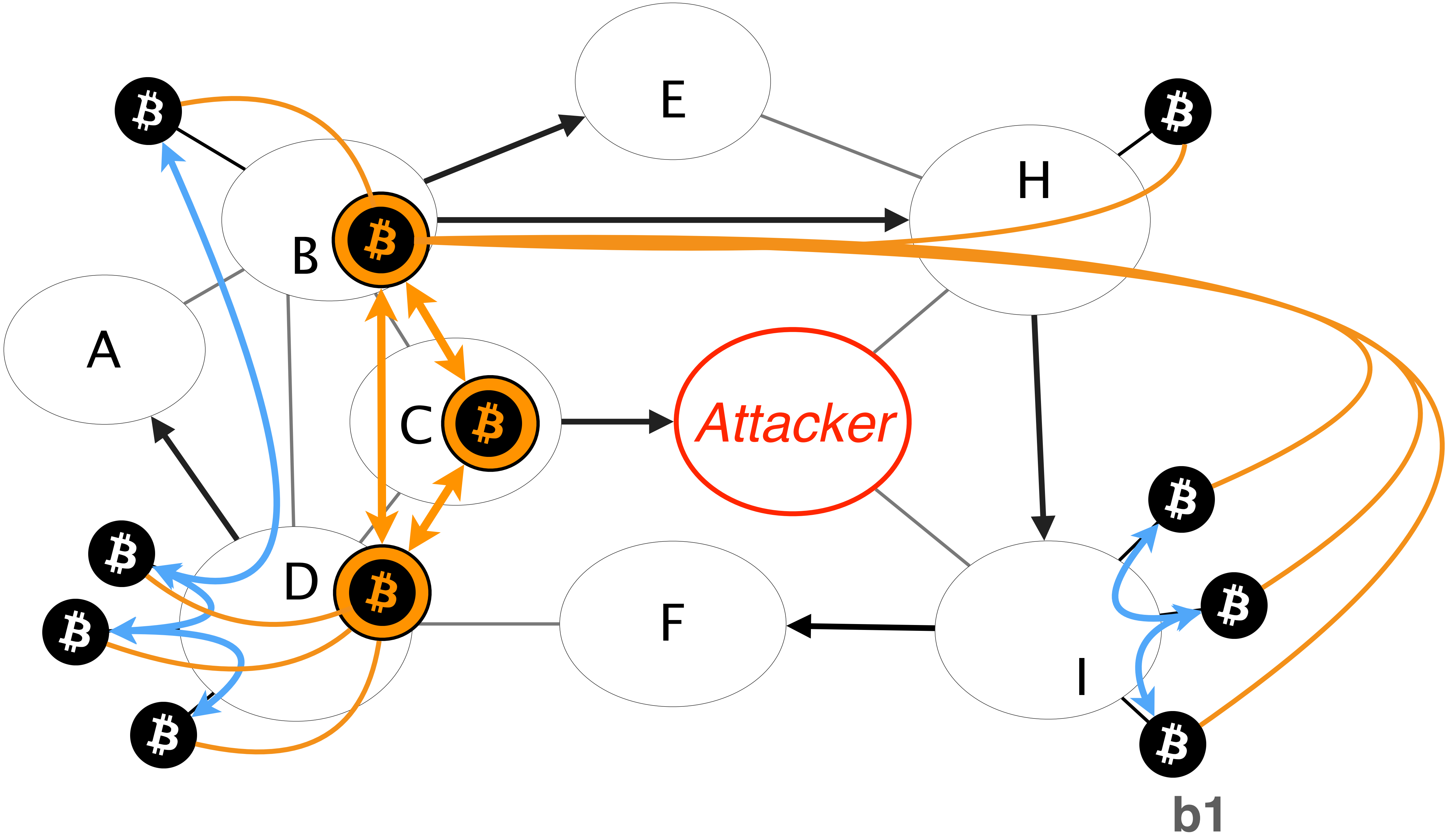
# Clients connect to at least one relay node



Intra-relay & some inter-relay connection survive



Partition attack failed

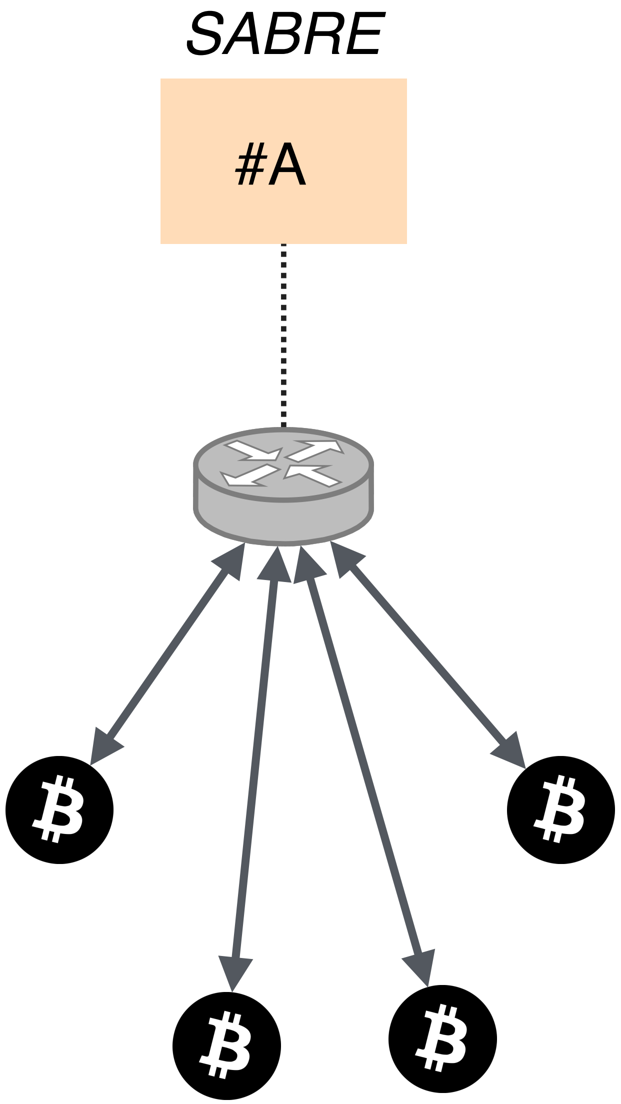


SABRE = Secure Relay Location + Robust Design

# Software/Hardware co-design

control plane
software

data plane
hardware



*SABRE*

#A

# Software/Hardware co-design is suitable because…

keep up with high demand

dynamic network defenses

# Software/Hardware co-design is suitable because…

keep up with high demand

- Tbps of traffic at line rate
- sustain DDoS attacks

dynamic network defenses

# Software/Hardware co-design is suitable because…

keep up with high demand

dynamic network defenses

Whitelists, BlackLists.

Spoofing Detection,

Amplification mitigation

# Software/Hardware co-design is possible because…

communication heavy protocol

rarely updated state

# Software/Hardware co-design is possible because…

communication heavy protocol

simple computations,

many message exchanges

rarely updated state

# Software/Hardware co-design is possible because…

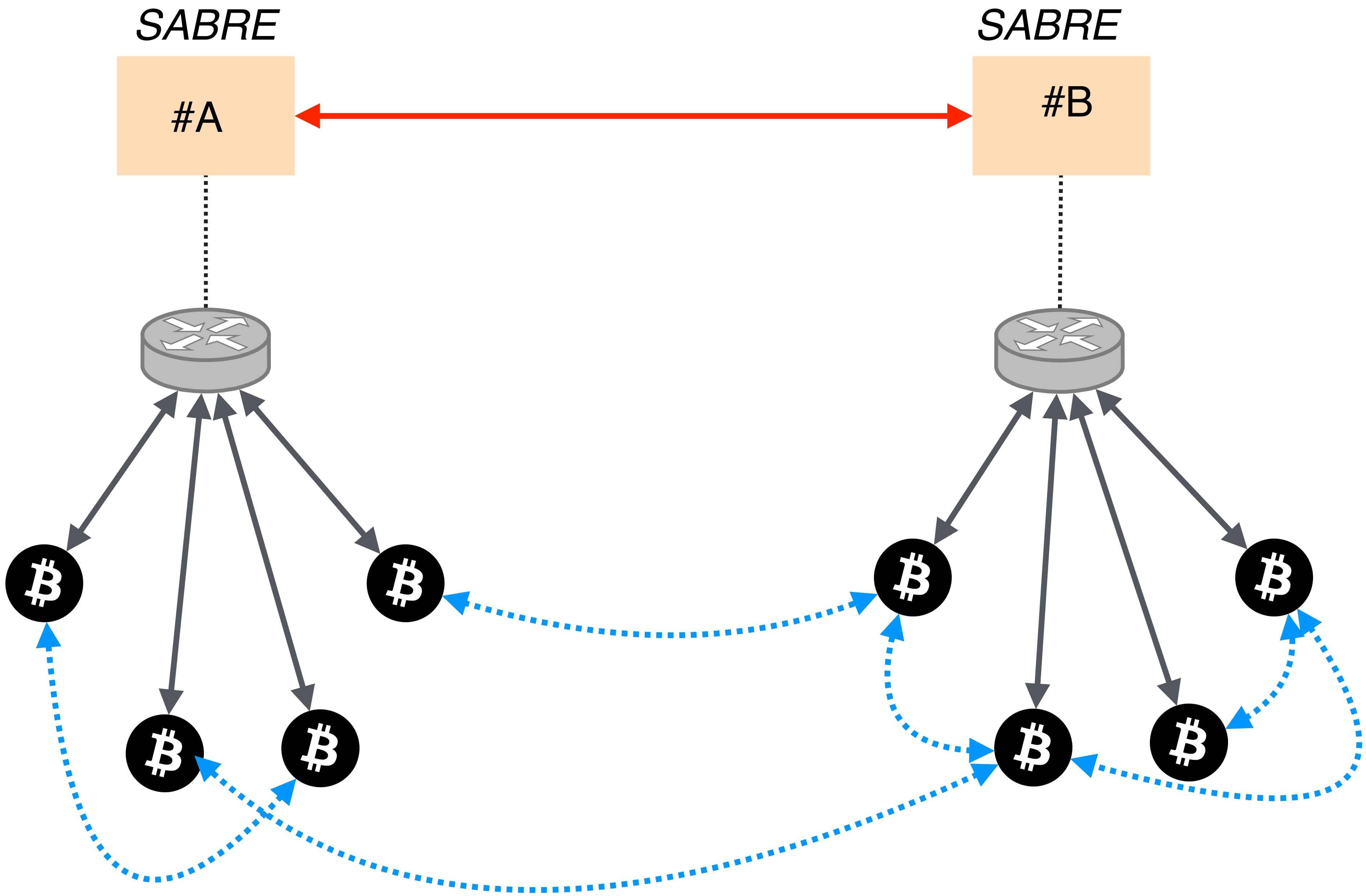
communication heavy protocol

rarely updated state

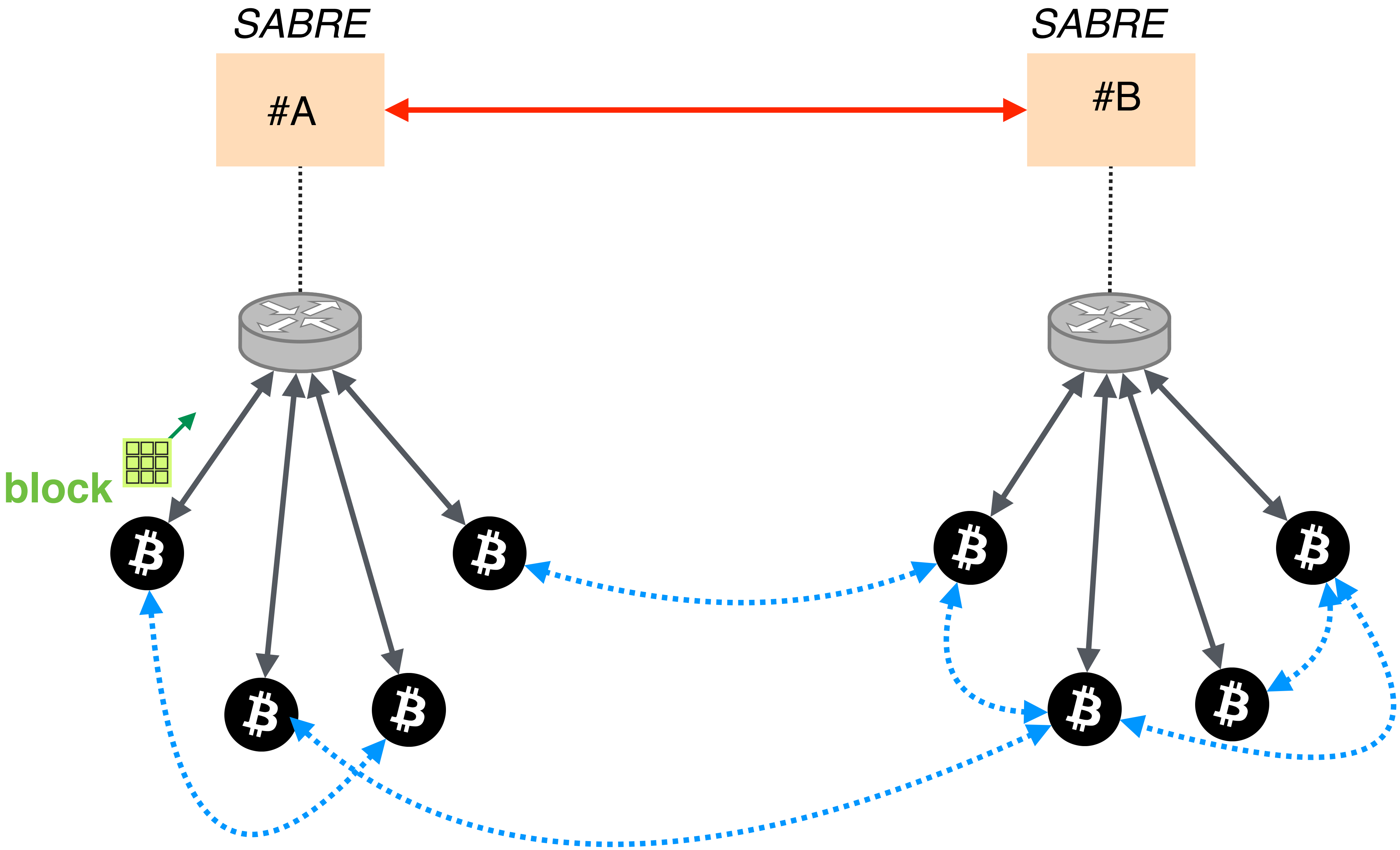
New Blocks are mined
every 10 minutes

## What is the life-cycle of a new Block?

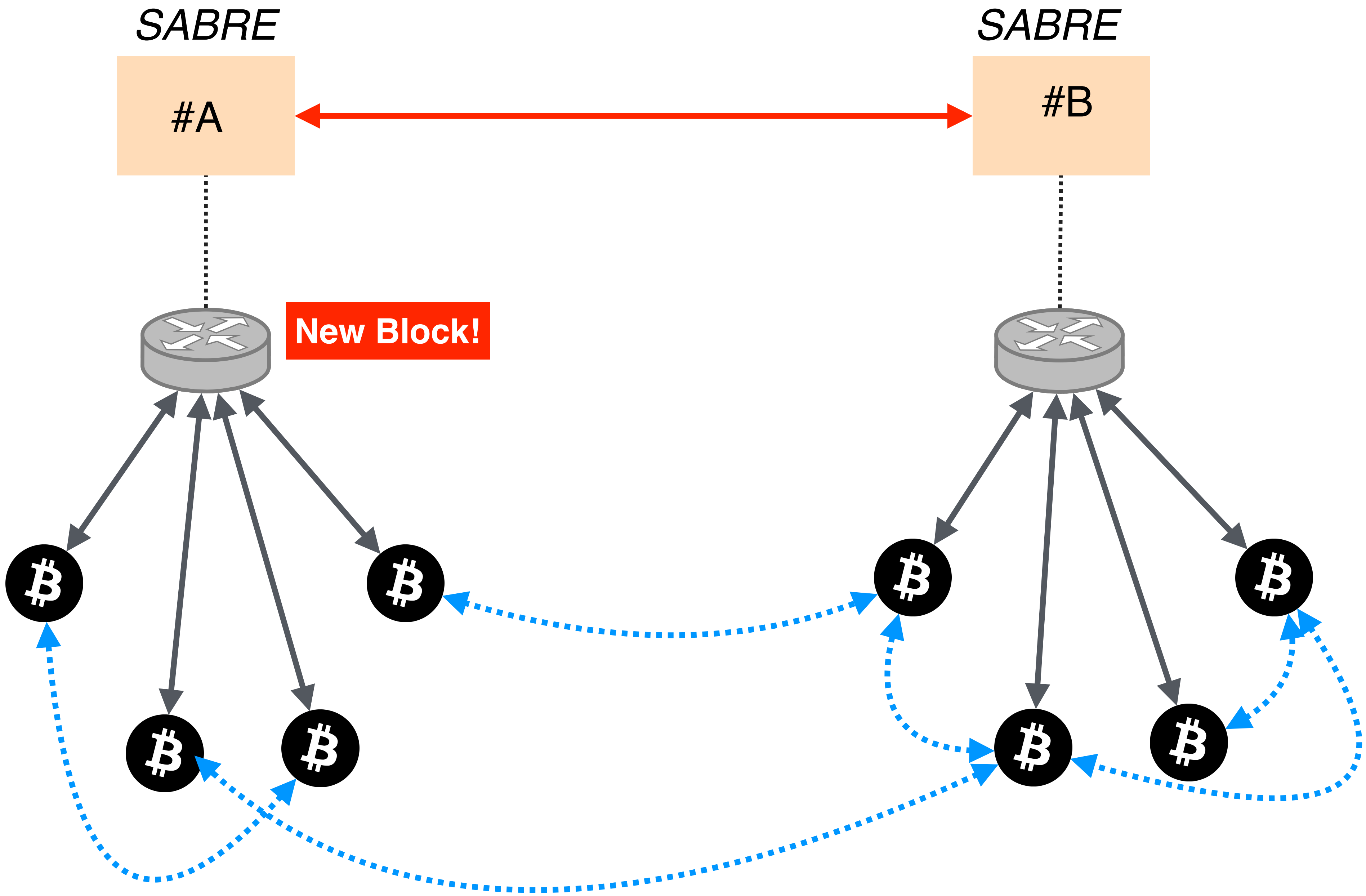
# Let’s see how it works in practice



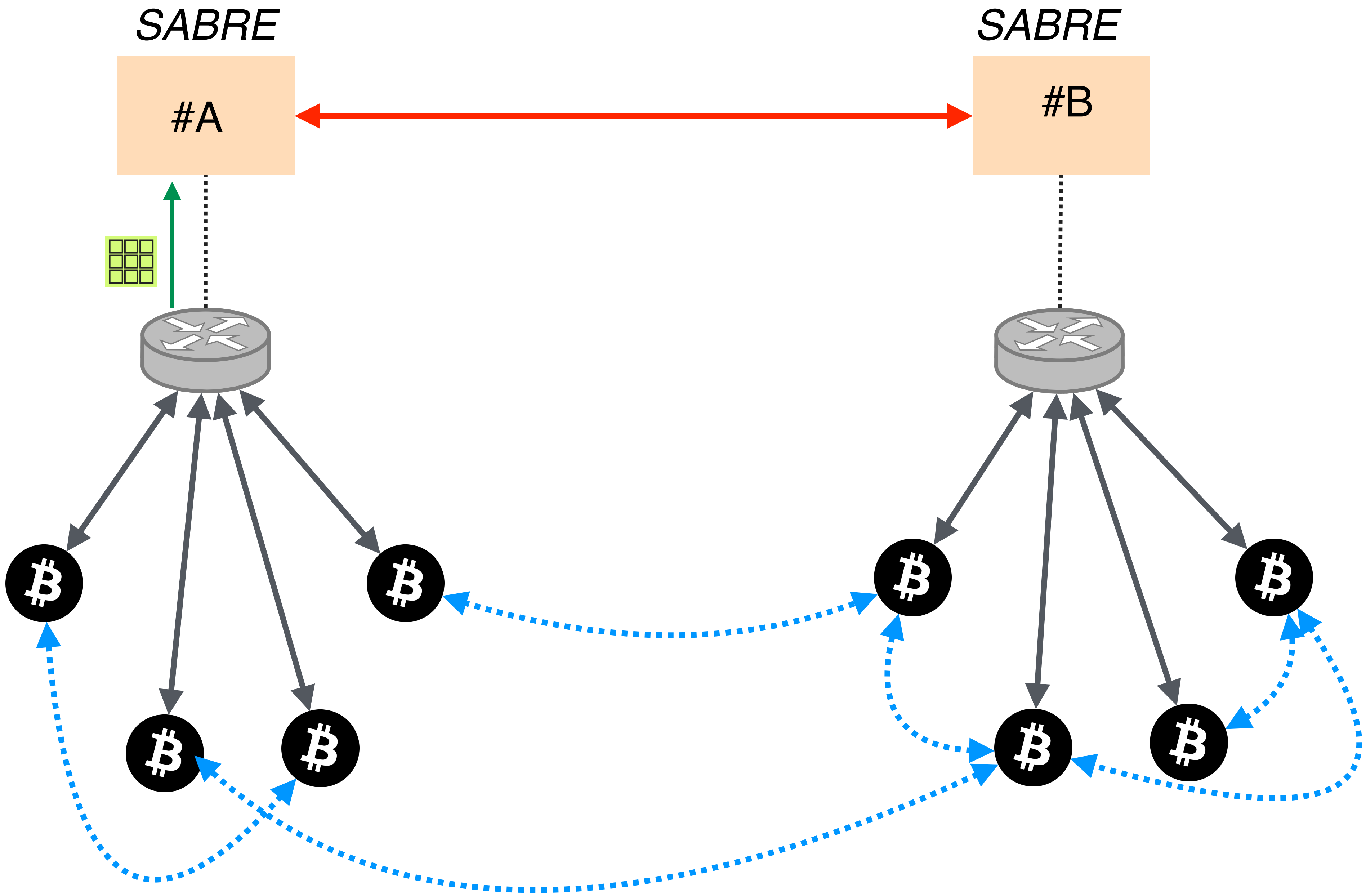
# New block sent to SABRE node



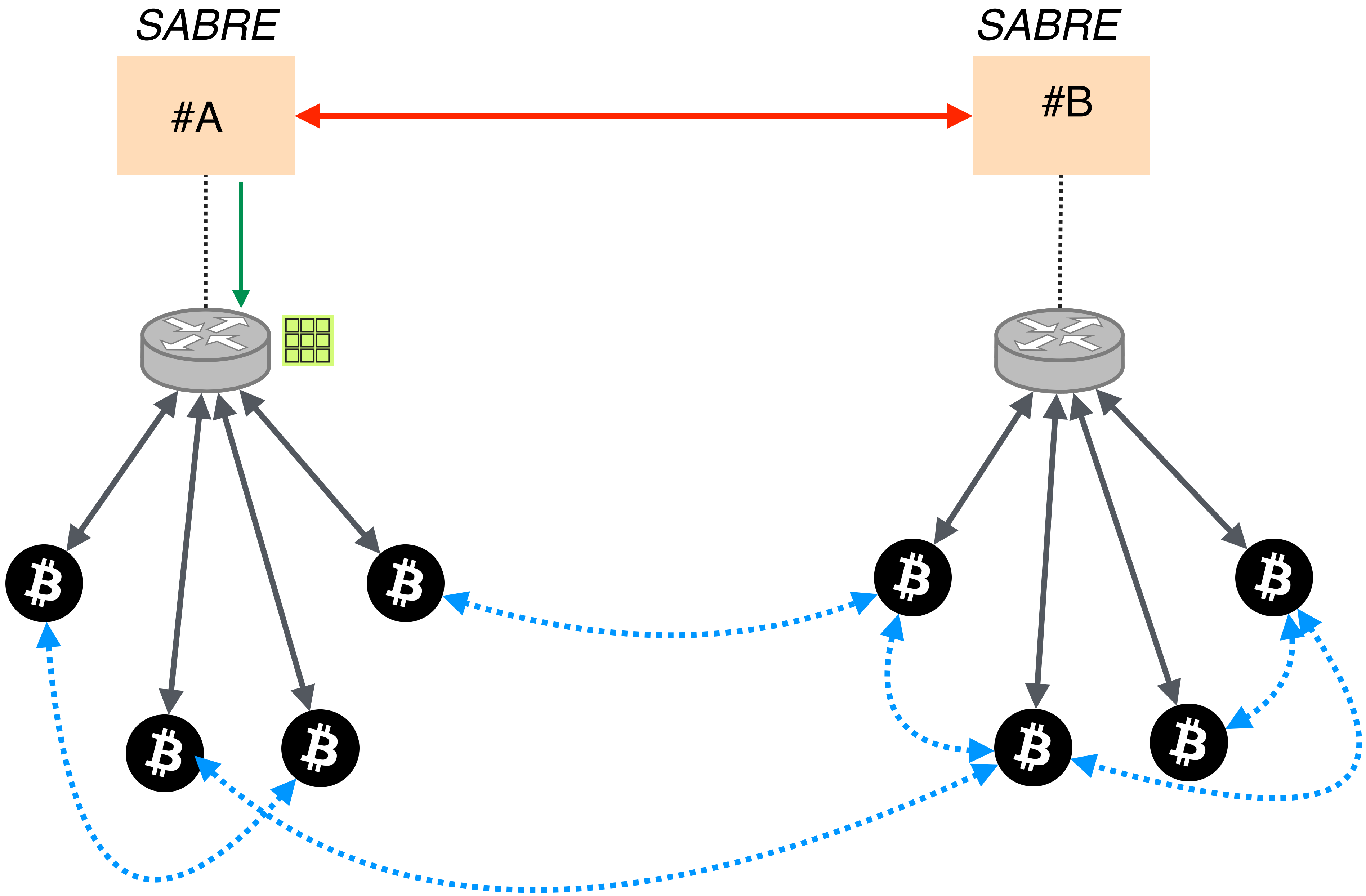
Switch detects it is a new block and forwards it to controller



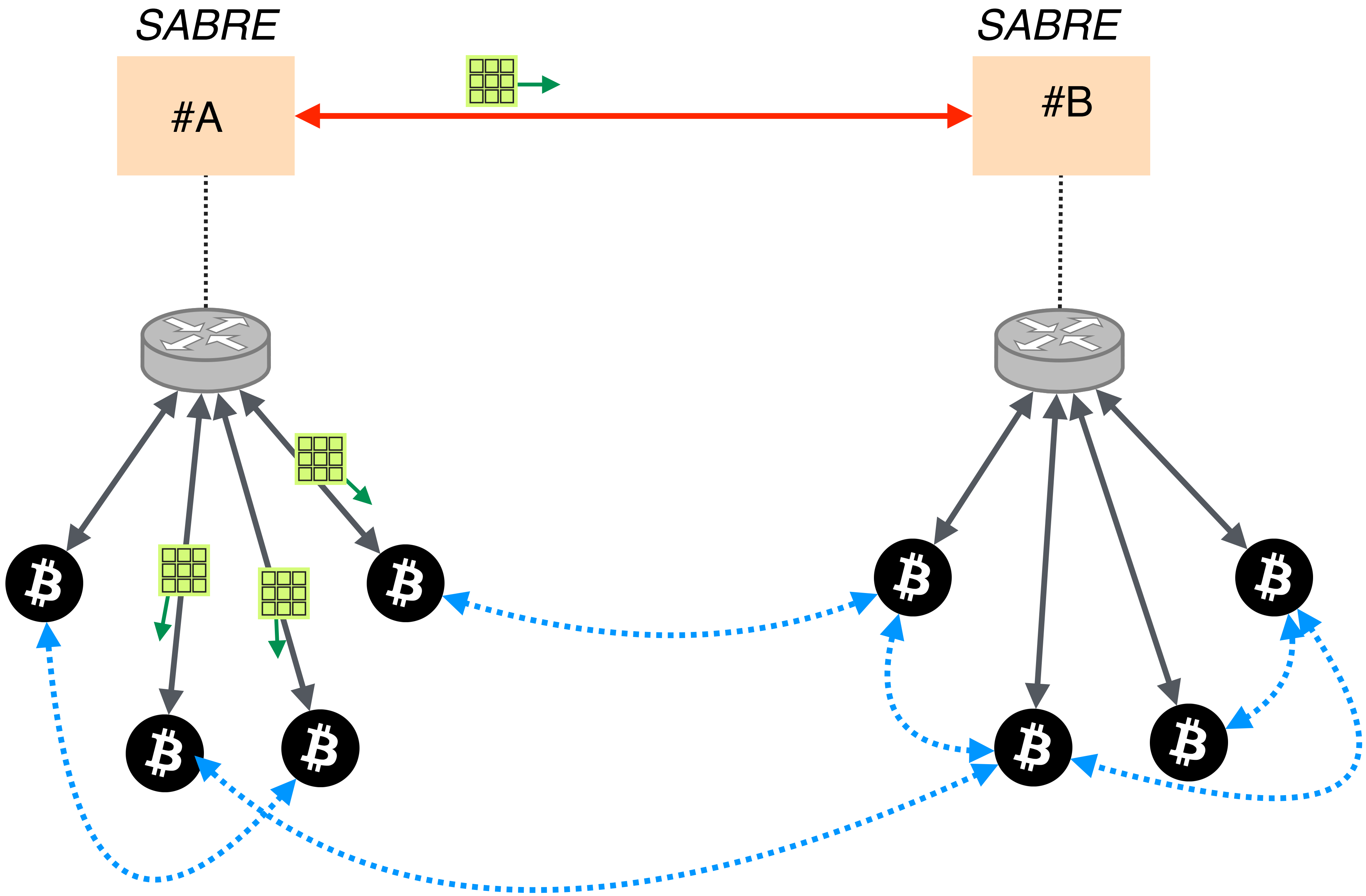
# Block is forwarded to the controller for validation



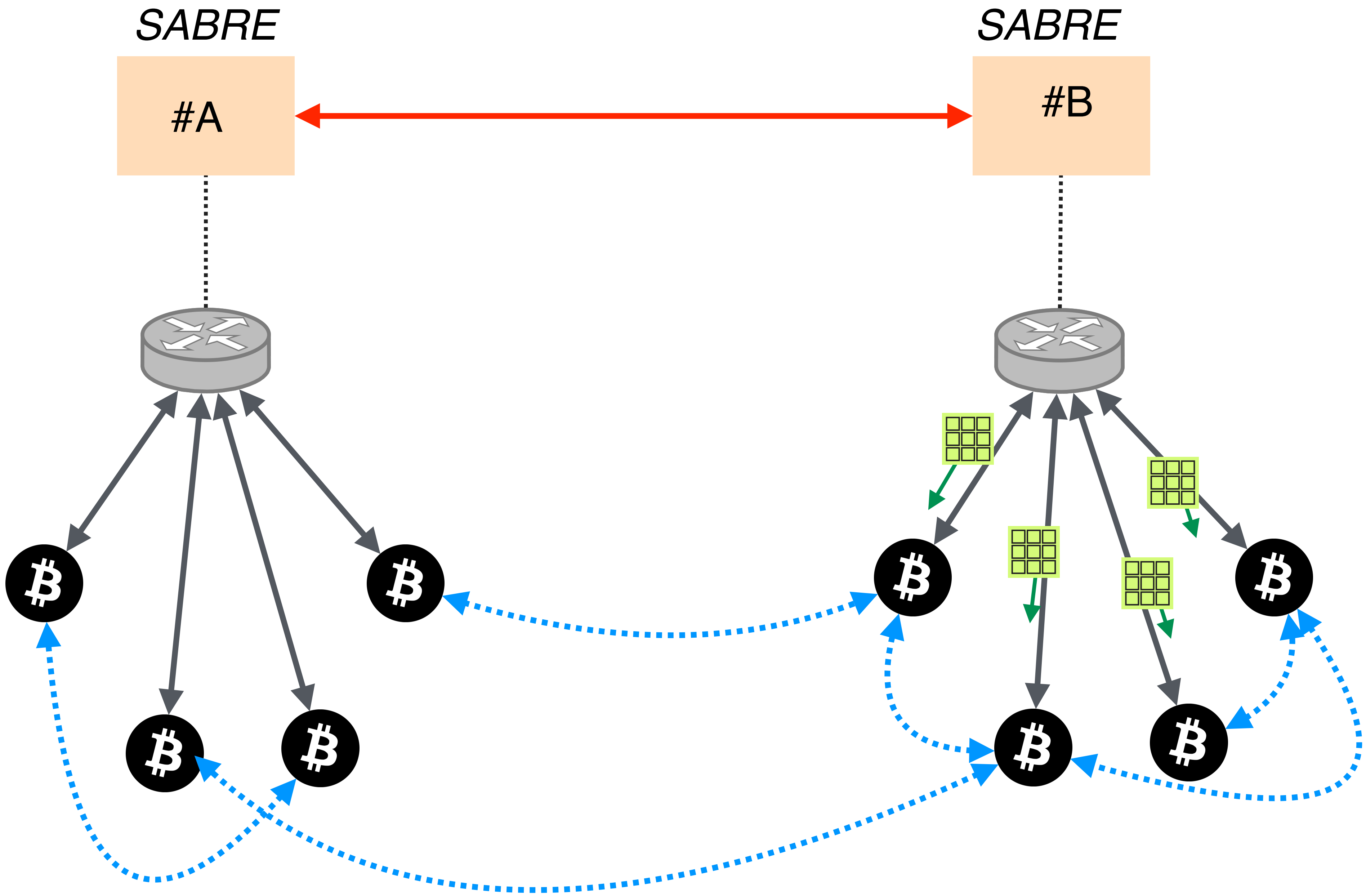
# Controller updates the memory of the switch



# Block is propagated upon request



# Block is propagated upon request



# Hijacking Bitcoin

Routing Attacks on Cryptocurrencies

Background

BGP & Bitcoin

Partitioning attack

splitting the network

Delay attack

slowing the network down

Countermeasures

short-term & long-term

https://btc-hijack.ethz.ch

# Hijacking Bitcoin

## Routing Attacks on Cryptocurrencies

Bitcoin is vulnerable to routing attacks

both at the network and at the node level

The potential impact on the currency is worrying

DoS, double spending, loss of revenues, etc.

Countermeasures exist

Secure routing is best; SABRE is a good alternative

https://btc-hijack.ethz.ch