Network Attacks I

Yongdae Kim

KAIST

Two Planes

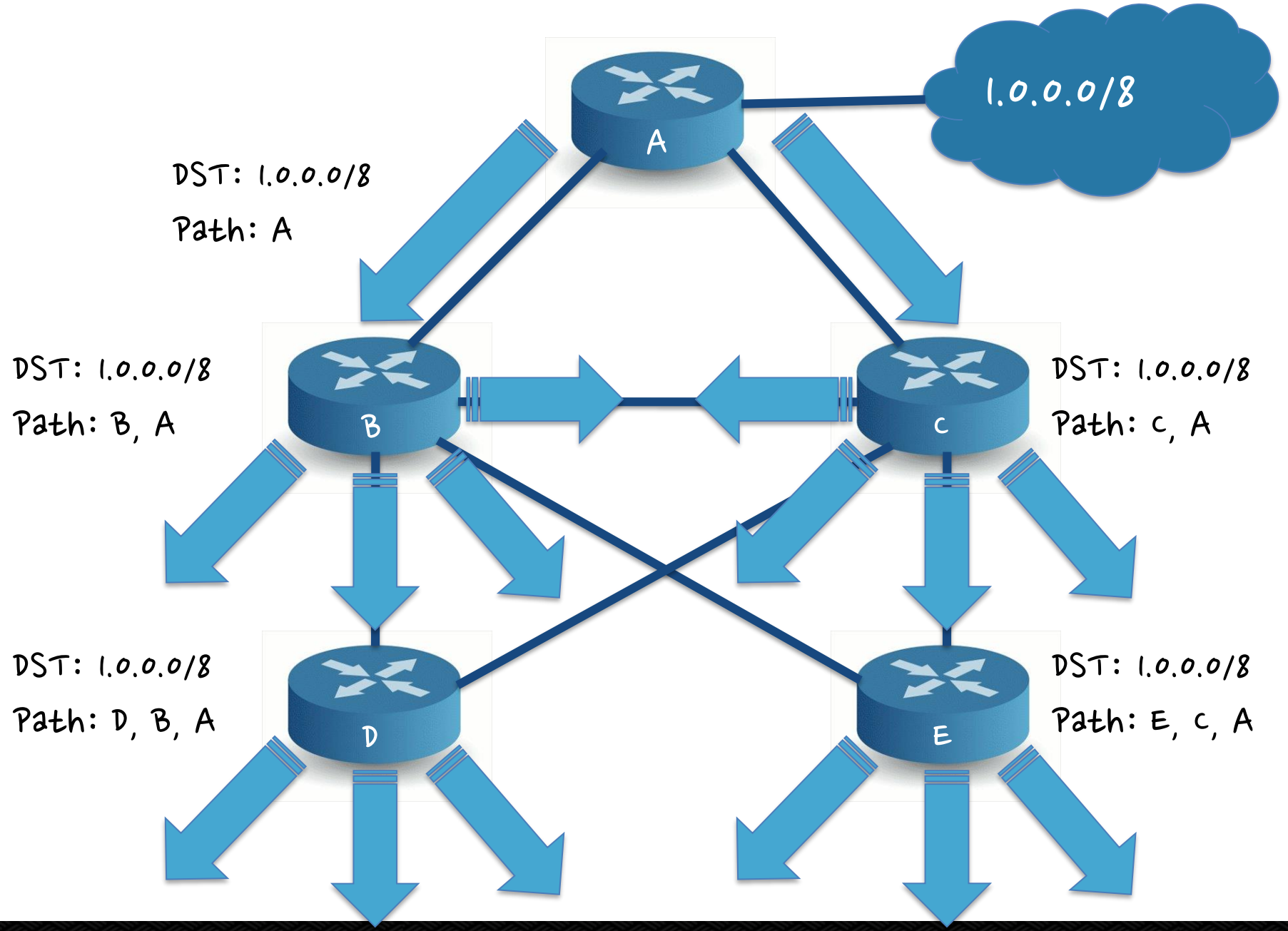
- Data Plane: Actual data delivery
- Control Plane
  - To support data delivery (efficiently, reliably, and etc.)
  - Routing information exchange
  - In some sense, every protocol except data delivery is considered to be control plane protocols
- Example network
  - Peer-to-peer network, Cellular network, Internet, …

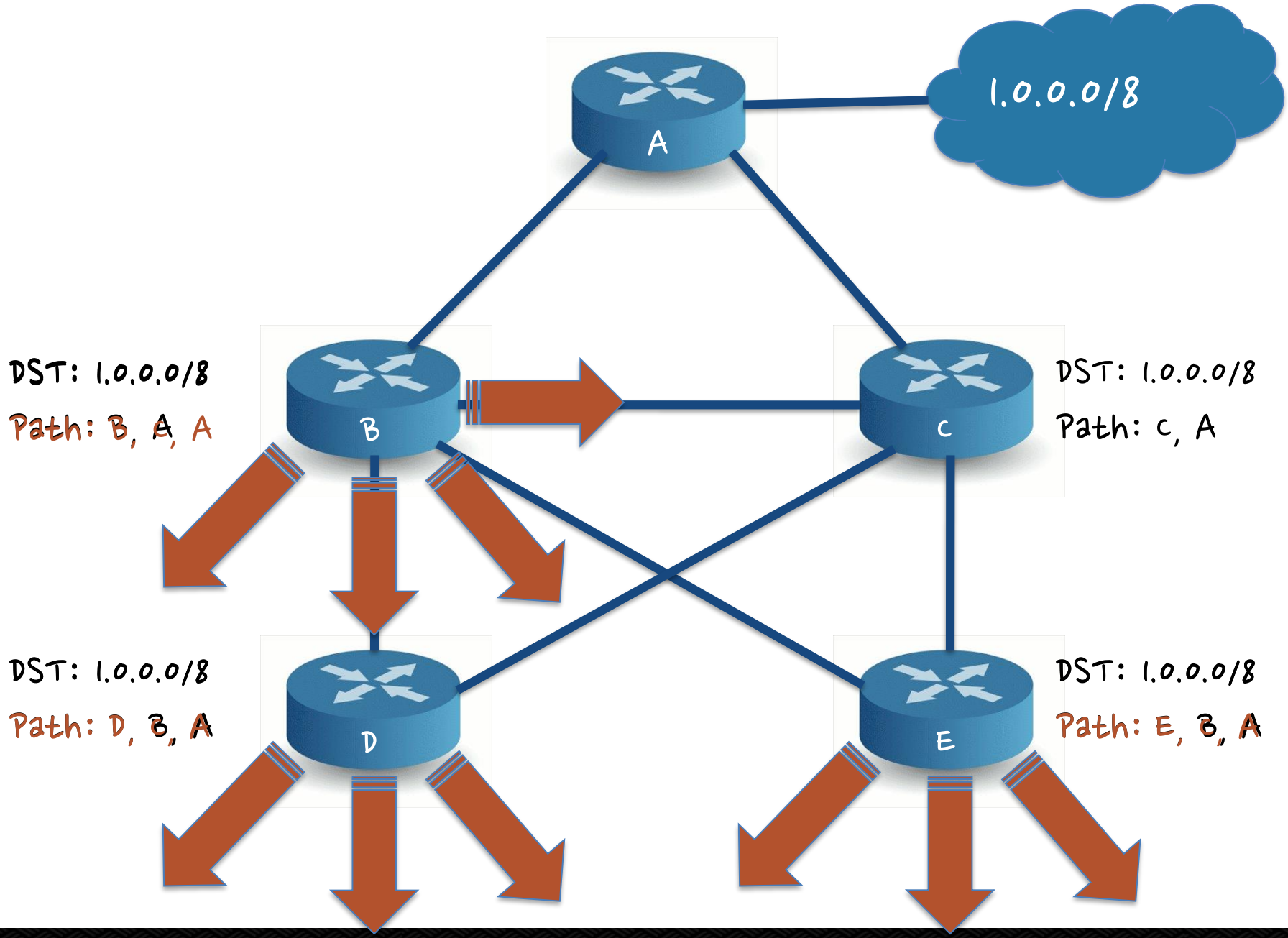
Misconfigurations and Redirection

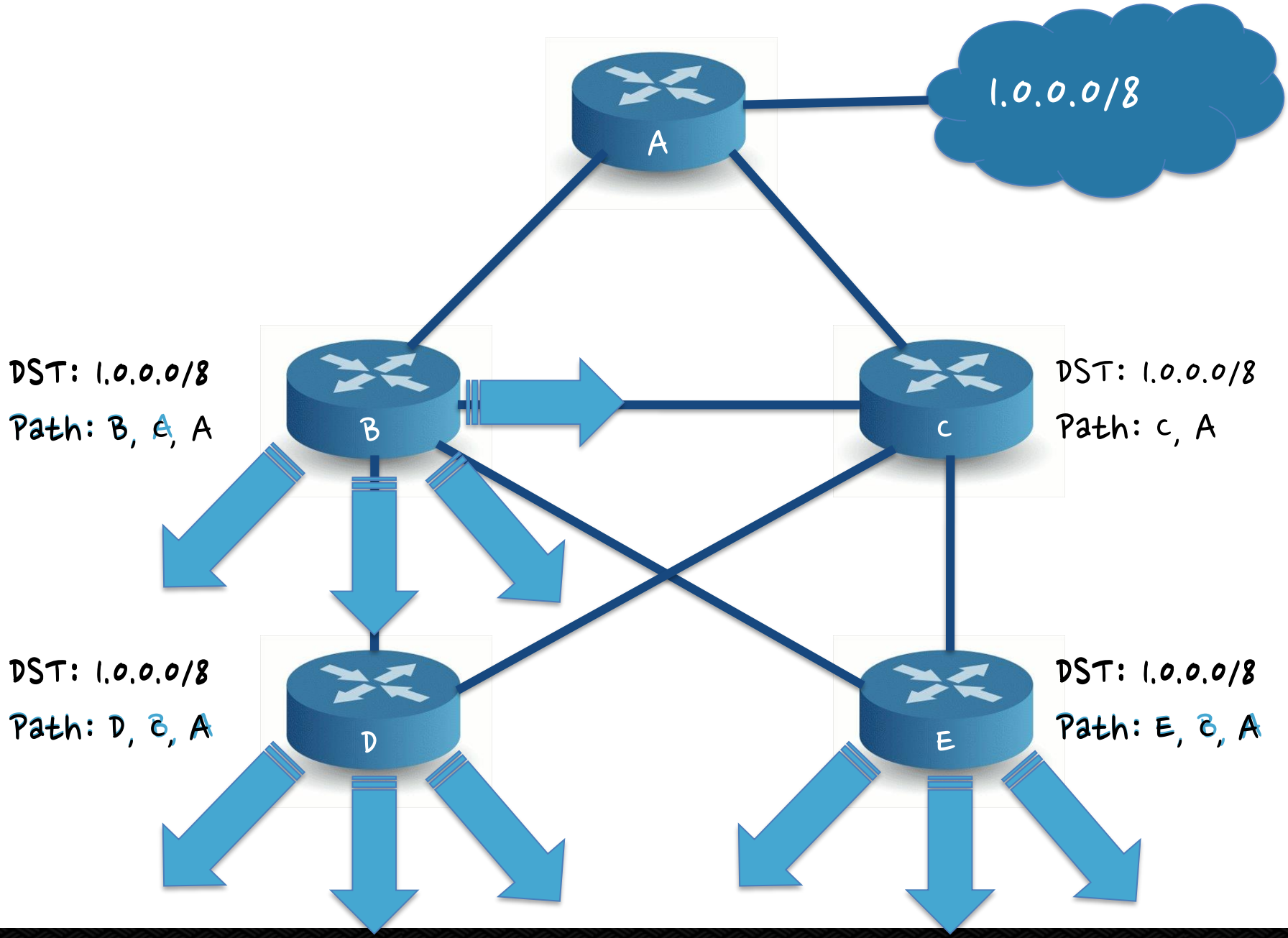
- 1997: AS7007
  - Claimed shortest path to the whole Internet
  - Causing Internet Black hole
- 2004: TTNet (AS9121)
  - Claimed shortest path to the whole Internet
  - Lasted for several hours
- 2006: AS27056
  - "stole" several important prefixes on the Internet
  - From Martha Stewart Living to The New York Daily News
- 2008: Pakistan Youtube
  - decided to block Youtube
  - One ISP advertised a small part of YouTube's (AS 36561) network
- 2010: China
  - 15% of whole Internet traffic was routed through China for 18 minutes
  - including .mil and .gov domain
- 2011: China
  - All traffic from US iPhone to Facebook
  - routed through China and Korea

AS, BGP and the Internet

- AS (Autonomous System)
  - Core AS: High degree of connectivity
  - Fringe AS: very low degrees of connectivity, sitting at the outskirts of the Internet
  - Transit AS: core ASes, which agree to forward traffic to and from other Ases
- BGP (Border Gateway Protocol)
  - the de facto standard routing protocol spoken by routers connecting different ASes.
  - BGP is a path vector routing algorithm, allowing routers to maintain a table of AS paths to every destination.
  - uses policies to preferentially use certain AS paths in favor.


1.0.0.0/8
A
DST: 1.0.0.0/8
Path: A
B
DST: 1.0.0.0/8
Path: B, A
C
DST: 1.0.0.0/8
Path: C, A
D
DST: 1.0.0.0/8
Path: D, B, A
E
DST: 1.0.0.0/8
Path: E, C, A


1.0.0.0/8
A
B
C
D
E
DST: 1.0.0.0/8
Path: B, A, A
DST: 1.0.0.0/8
Path: C, A
DST: 1.0.0.0/8
Path: D, B, A
DST: 1.0.0.0/8
Path: E, B, A


1.0.0.0/8
A
B
C
D
E
DST: 1.0.0.0/8
Path: B, A
DST: 1.0.0.0/8
Path: C, A
DST: 1.0.0.0/8
Path: D, B, A
DST: 1.0.0.0/8
Path: E, B, A

Hijacking Bitcoin: Routing Attacks on Cryptocurrencies

Maria Apostolaki, Aviv Zohar, Laurent Vanbever
ETH Zurich, The Hebrew University, ETH Zurich

Geunwoo Lim
KAIST

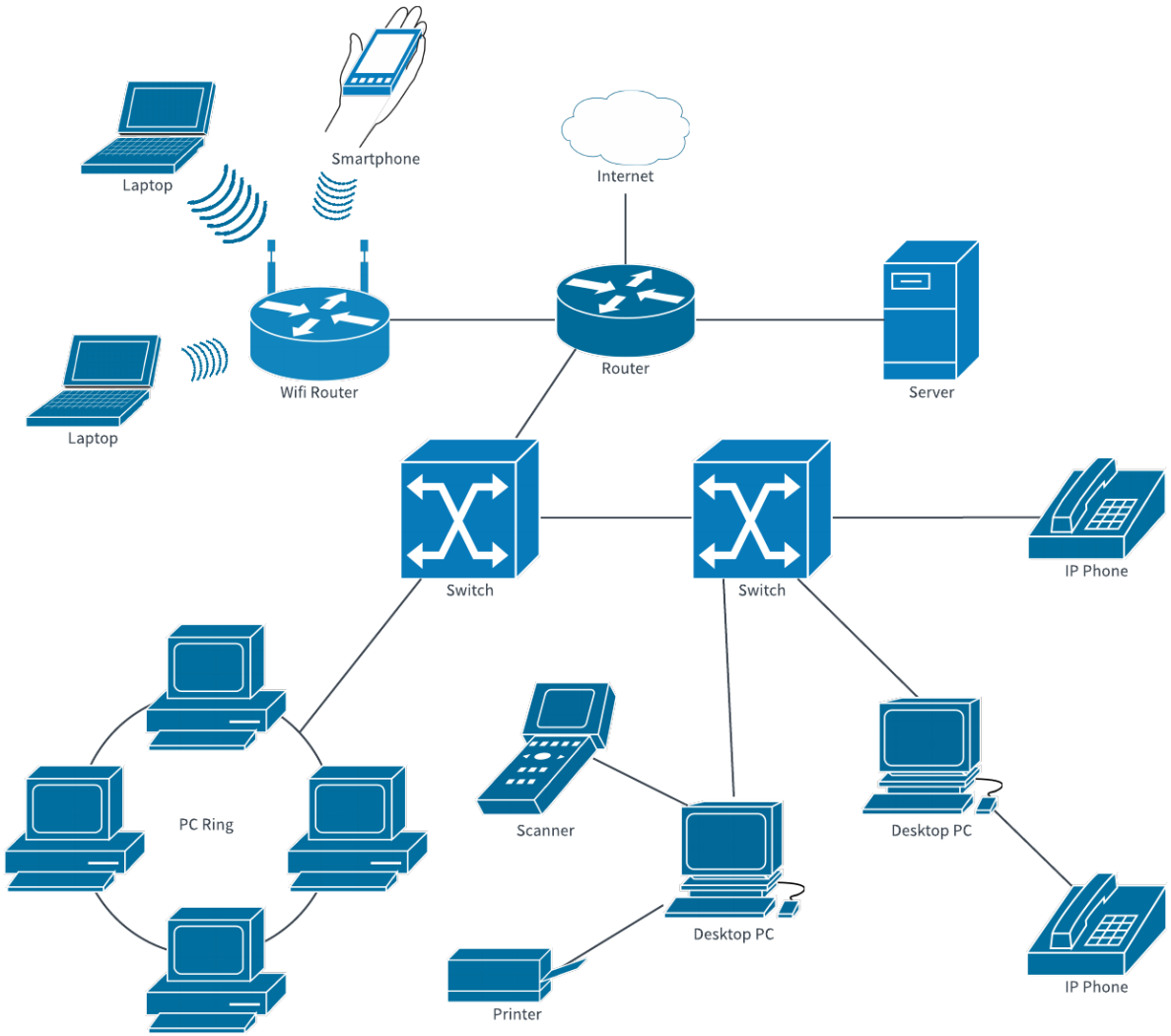
Various Attacks

- Many attacks are discovered belonging to consensus and mining pool
  - Double spending
  - Selfish mining
  - BWH attack
  - FAW attack

- But consensus and mining pool is only a fraction of blockchain system

- One of the major part of blockchain is network, easily think about P2P system.

Network component



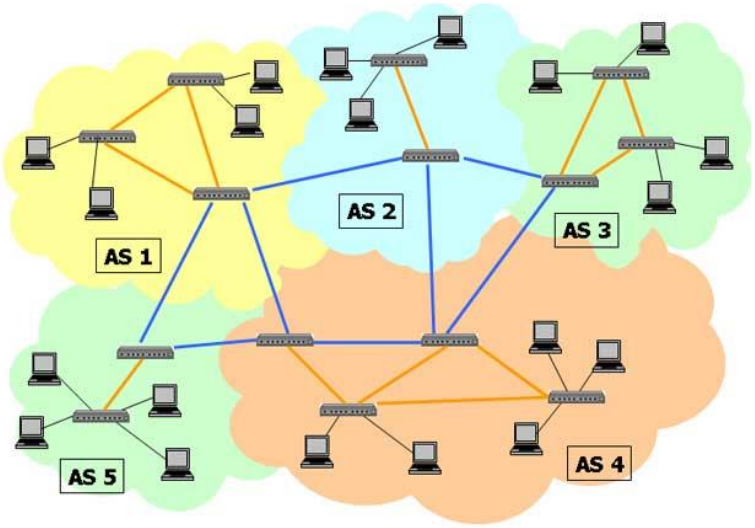
AS and ISP

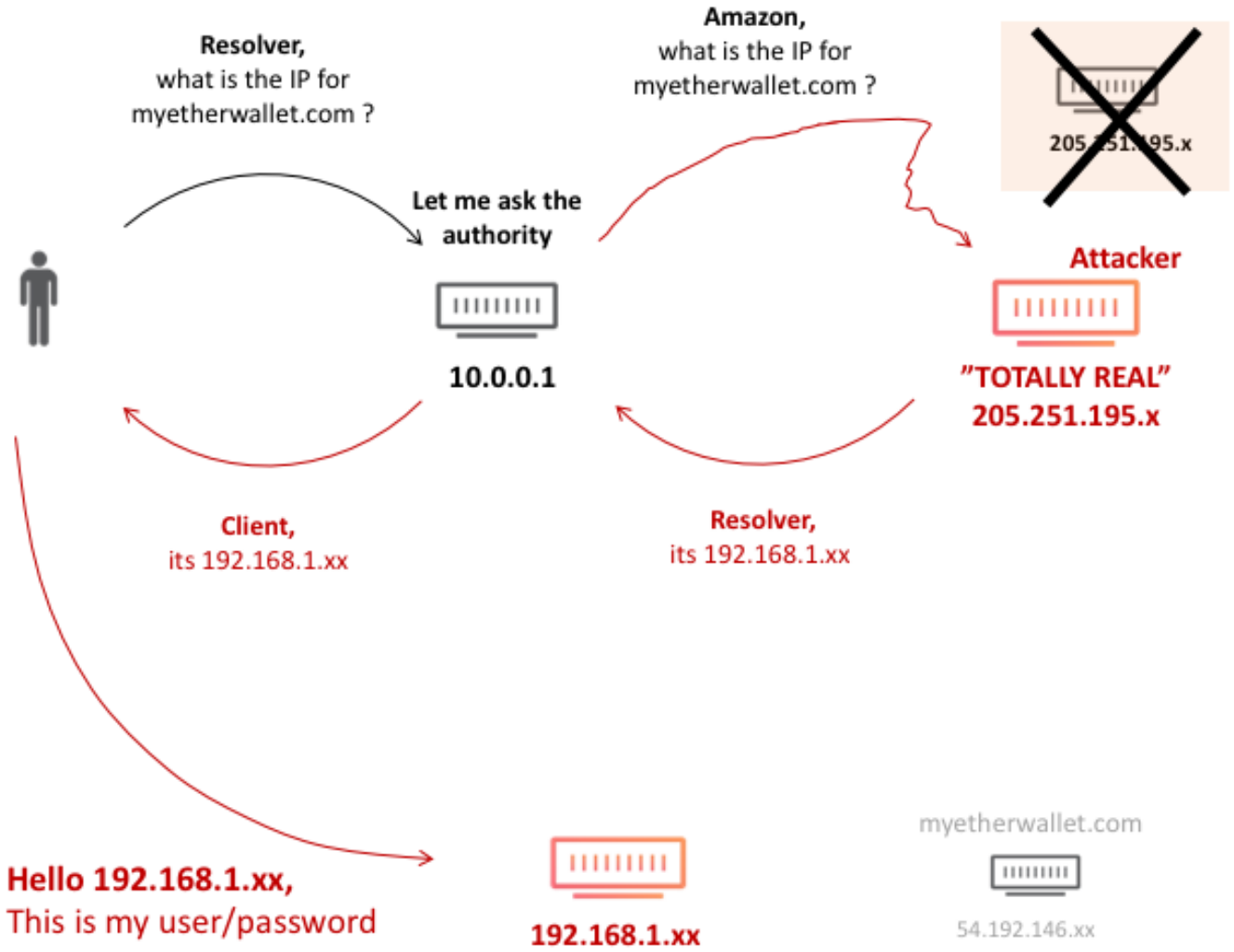
- Autonomous System
  - Set of same routing policy with same administrator
  - Distinguished by ASN
  - The reason why we use AS is many
    - » Independence of routing policy
    - » Security issue
    - » Minimization of routing traffic

- Internet Service Provider
  - Company which provide internet service
  - SKT, KT, LG U+

BGP

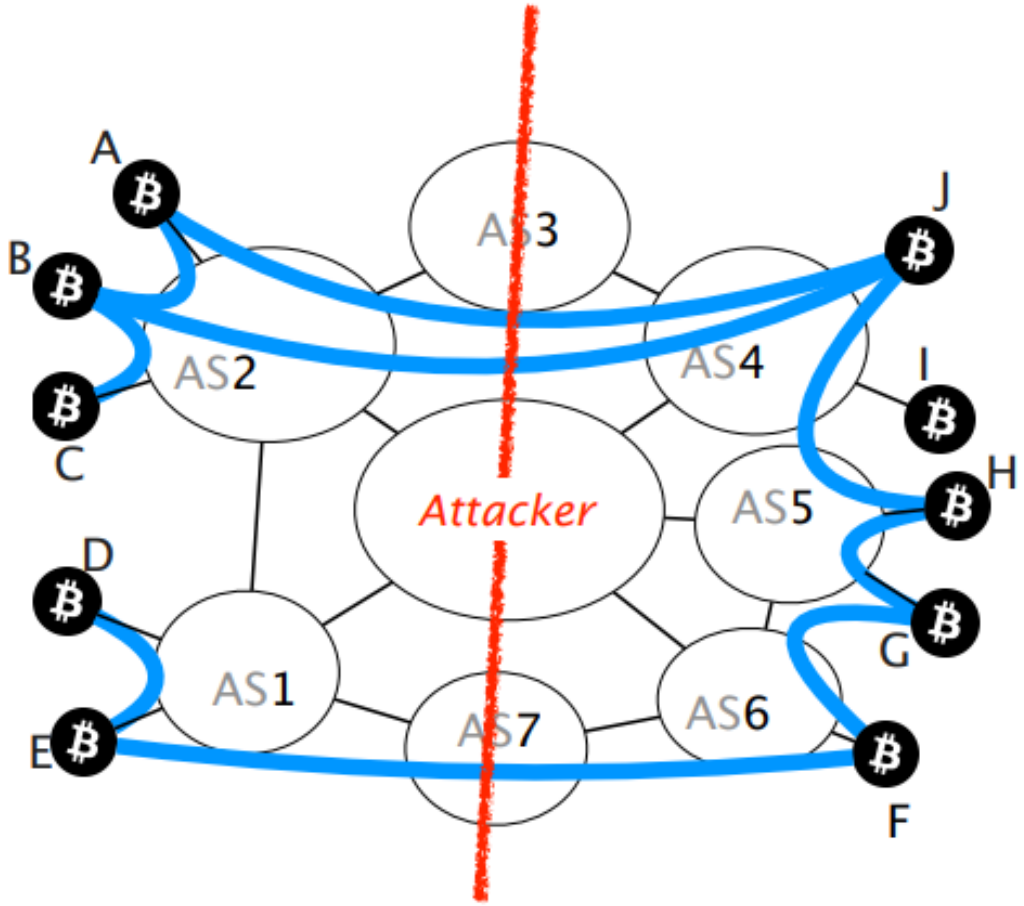
- Border Gateway Protocol
  - Standardized exterior gateway protocol (EGP)
  - Path vector protocol
  - BGP have many security issue because of these three vulnerabilities
    - » Do not have enough mechanism for message integrity, freshness, authentication, etc
    - » Do not have any authority about Network Layer Reachability Information announcement
    - » Do not have authentication process about path announced by other ASes




Resolver,
what is the IP for
myetherwallet.com ?
Amazon,
what is the IP for
myetherwallet.com ?
205.251.195.x
Let me ask the
authority
Attacker
10.0.0.1
"TOTALLY REAL"
205.251.195.x
Client,
its 192.168.1.xx
Resolver,
its 192.168.1.xx
Hello 192.168.1.xx,
This is my user/password
192.168.1.xx
myetherwallet.com
54.192.146.xx

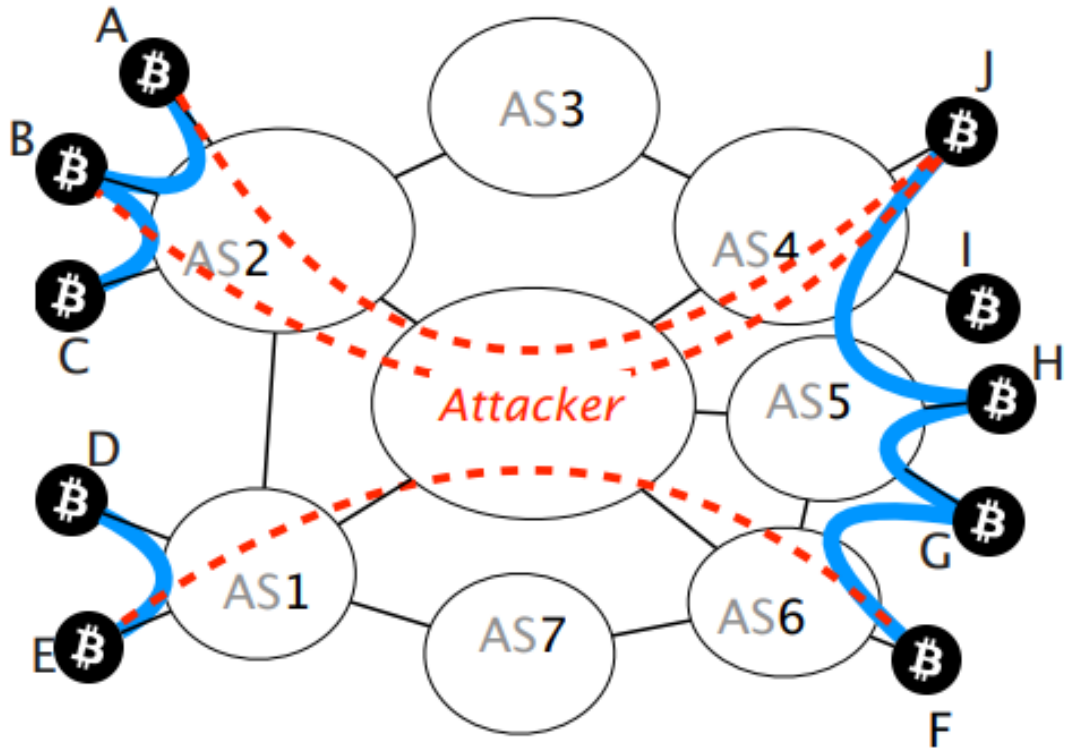
Attack Scenario (partition)

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



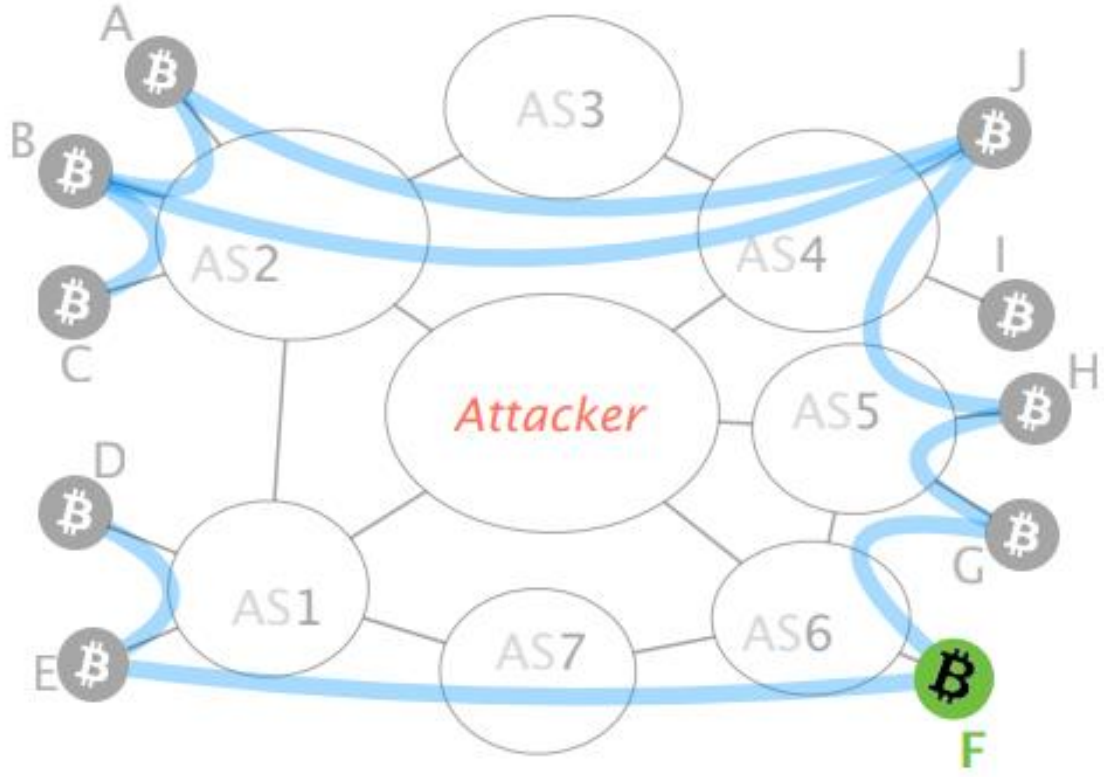
Attack Scenario (partition)

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



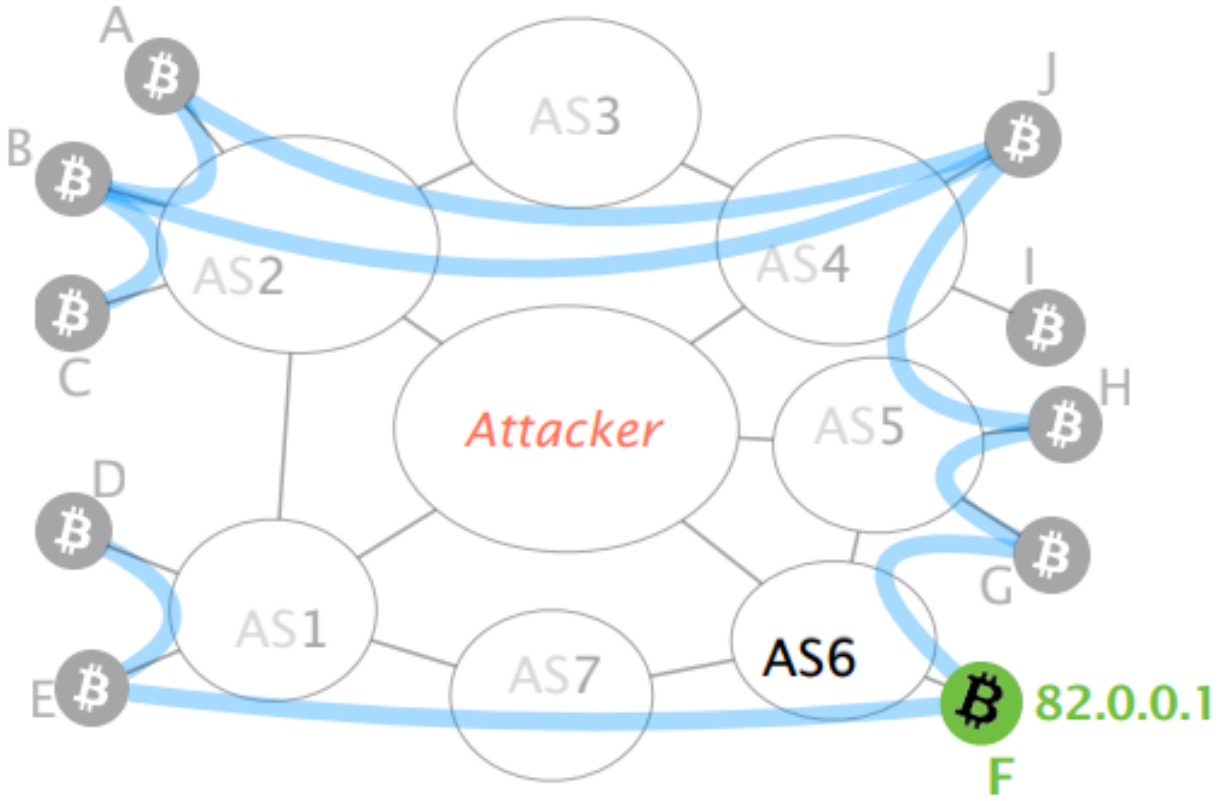
Attack Scenario (partition)

Let us focus on node F



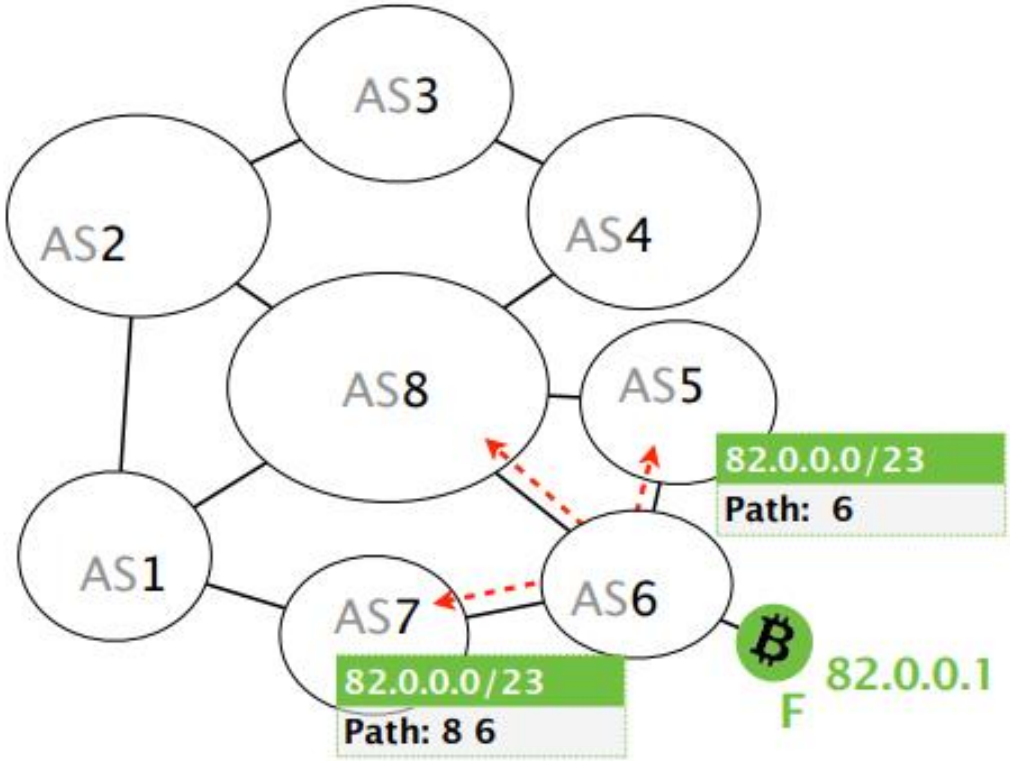
Attack Scenario (partition)

F's provider (AS6) is responsible for IP prefix



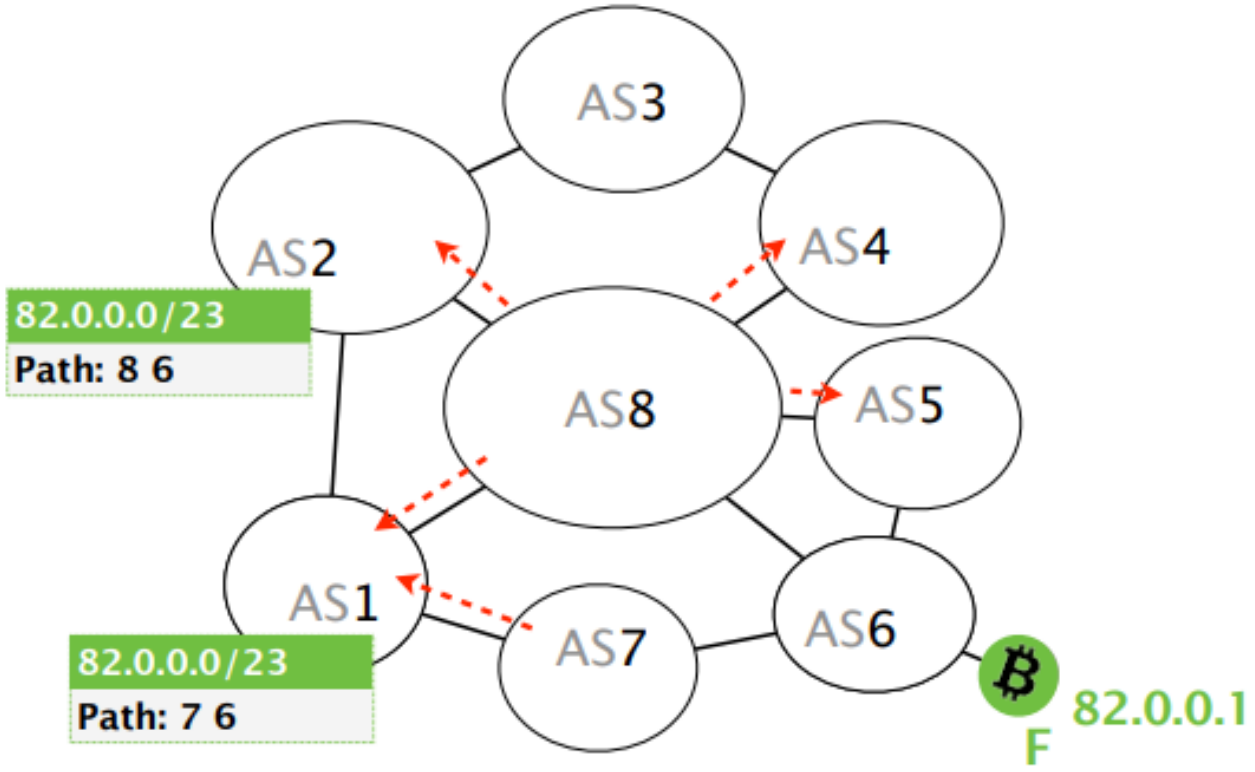
Attack Scenario (partition)

AS6 will create a BGP advertisement



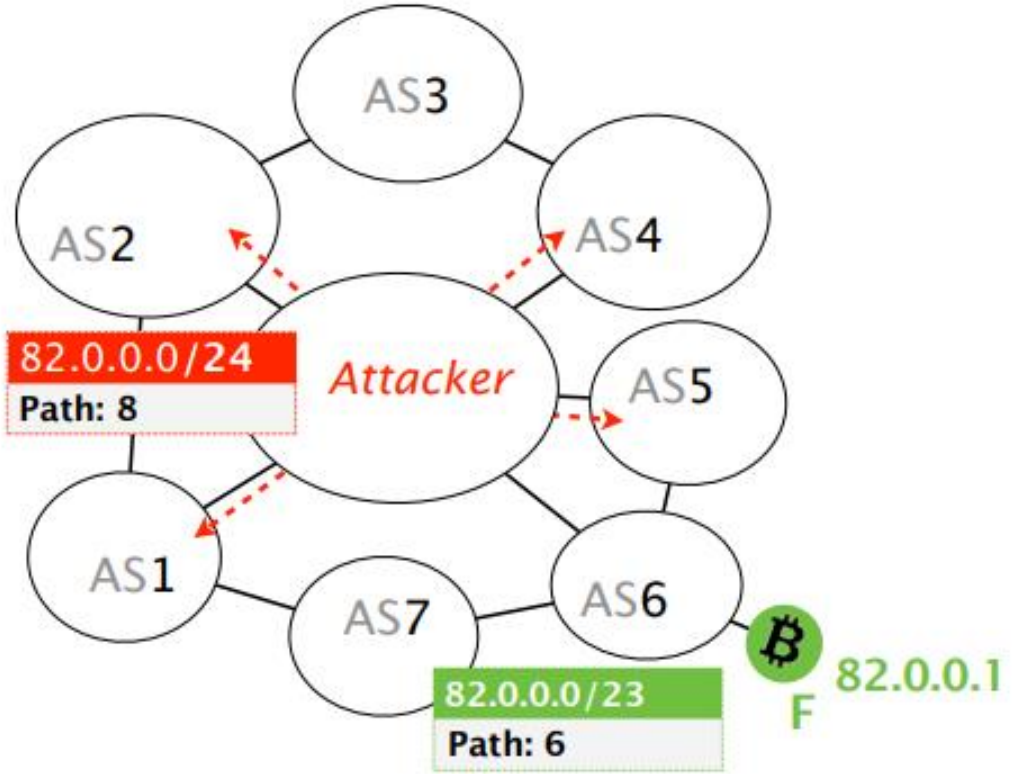
Attack Scenario (partition)

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



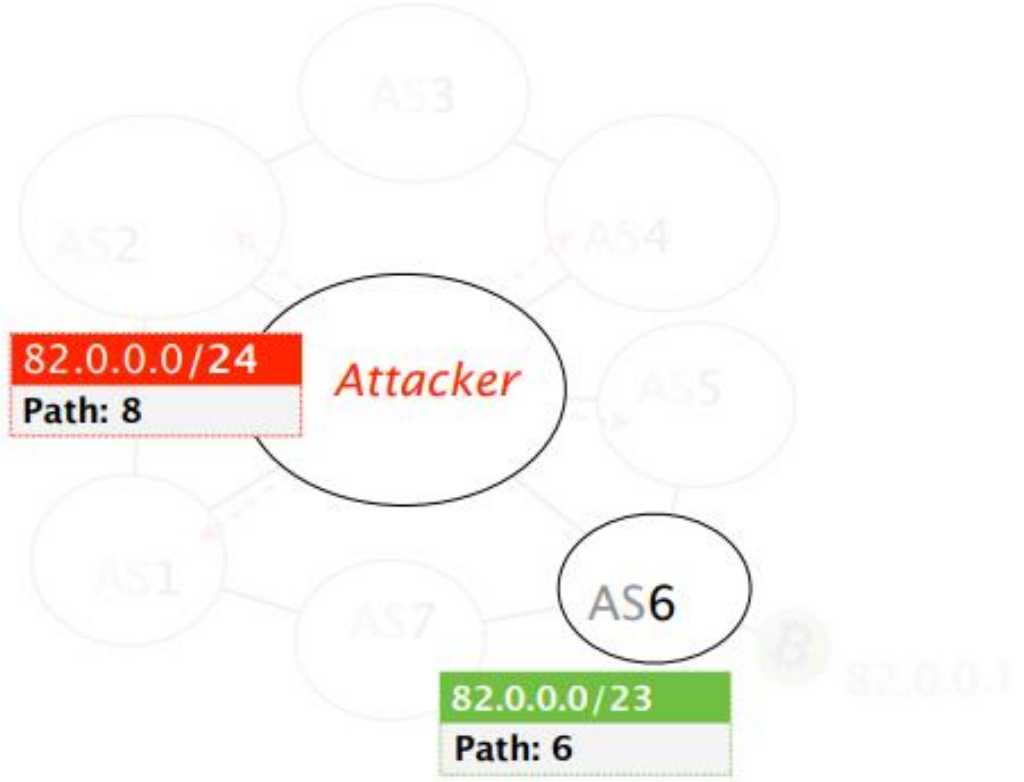
Attack Scenario (partition)

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



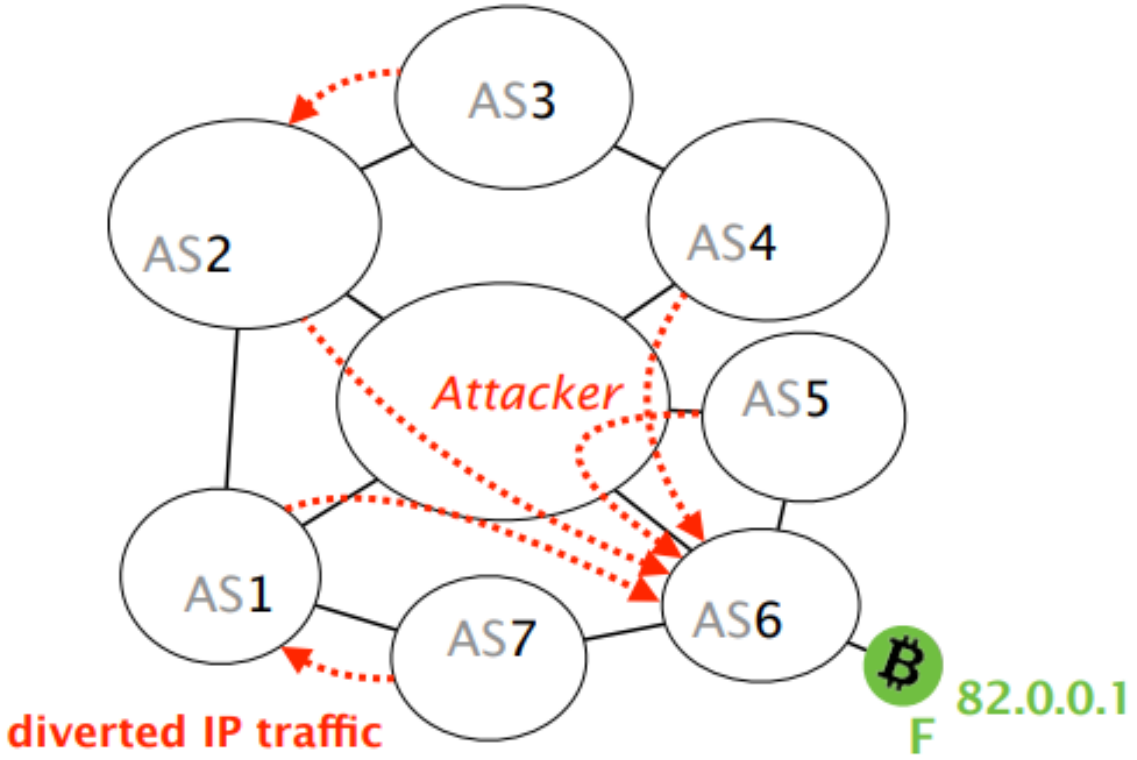
Attack Scenario (partition)

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



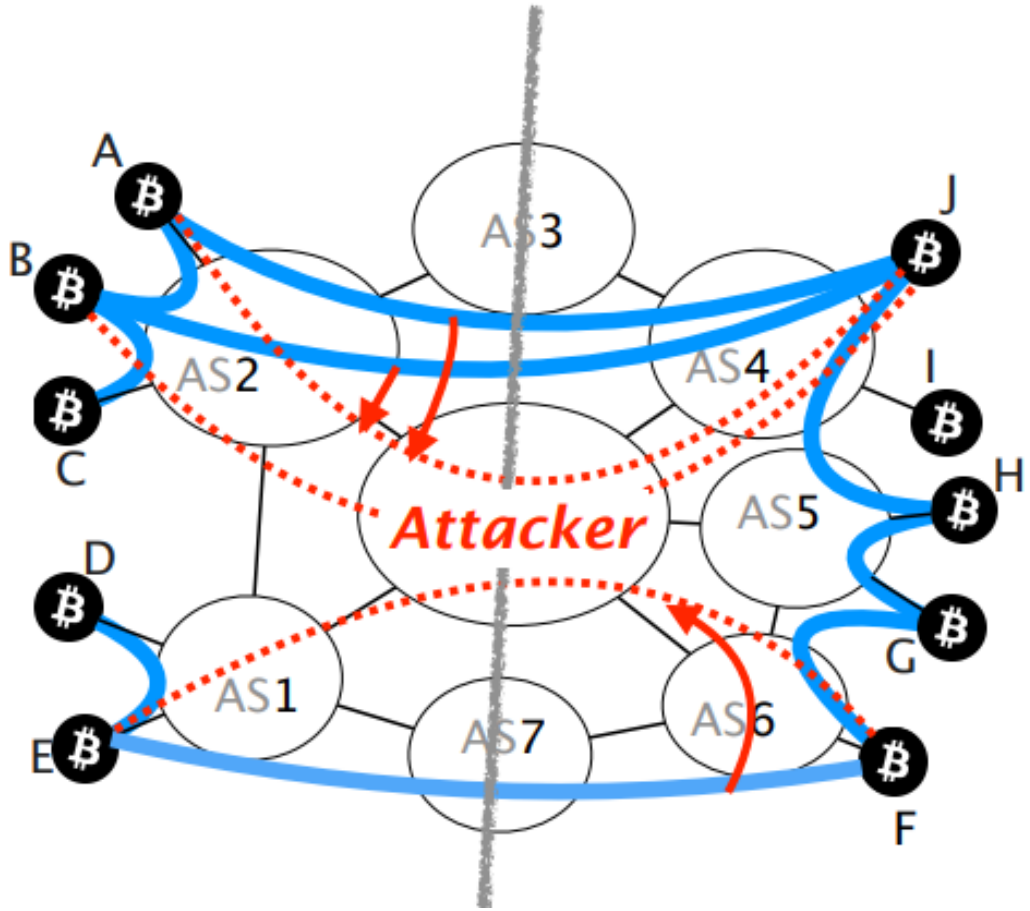
Attack Scenario (partition)

Traffic to node F is hijacked



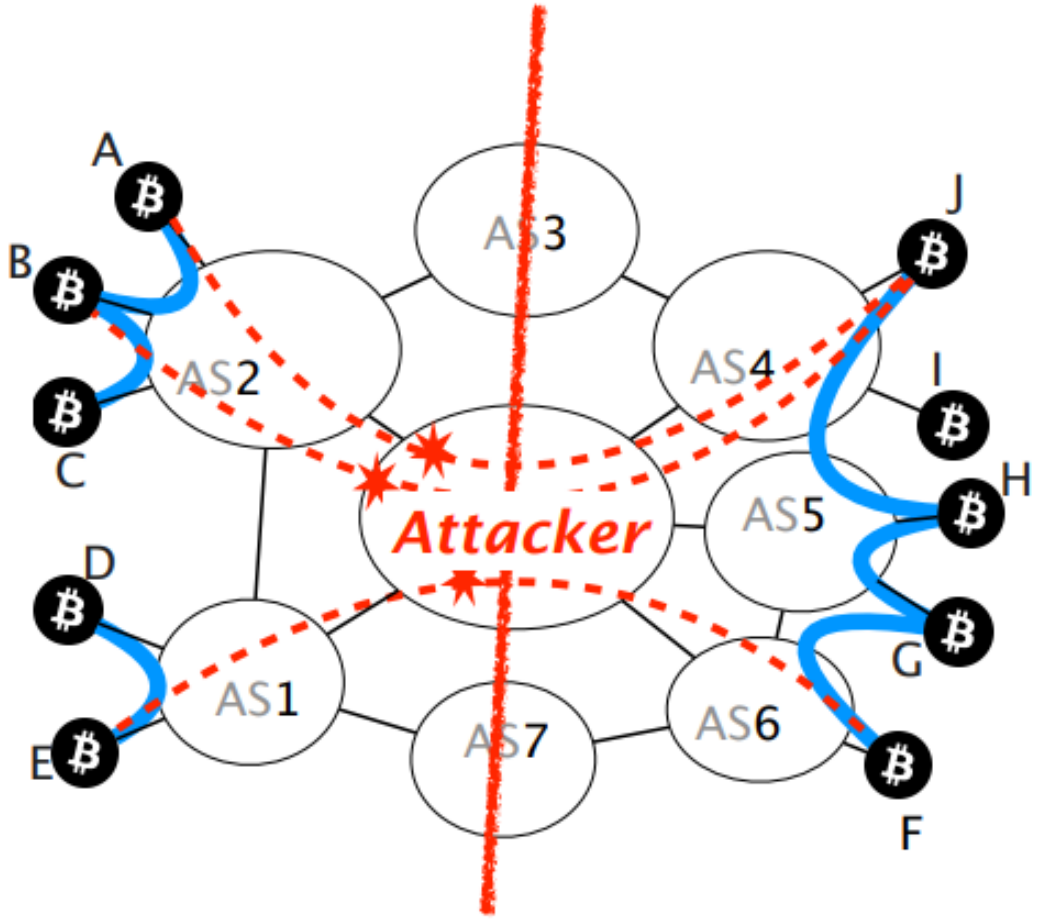
Attack Scenario (partition)

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



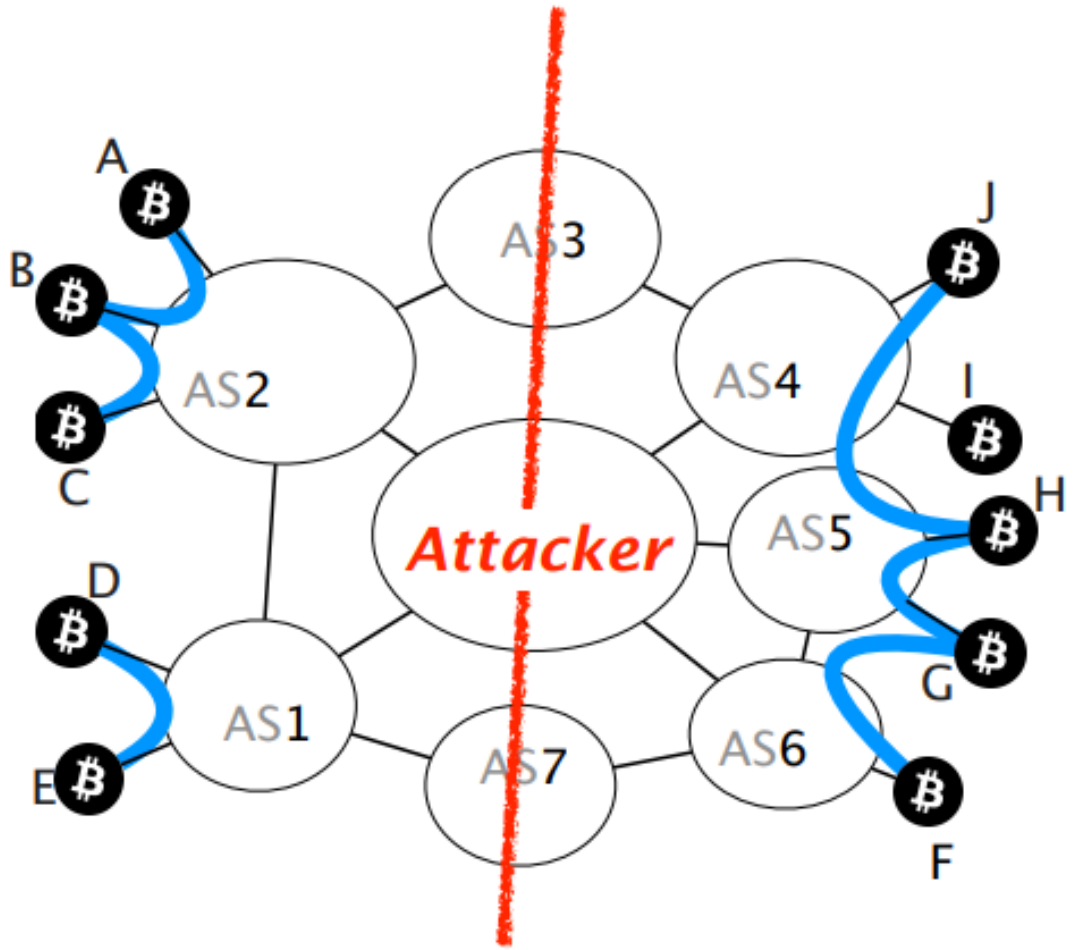
Attack Scenario (partition)

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



Attack Scenario (partition)

The partition is created



Attack Scenario (partition)

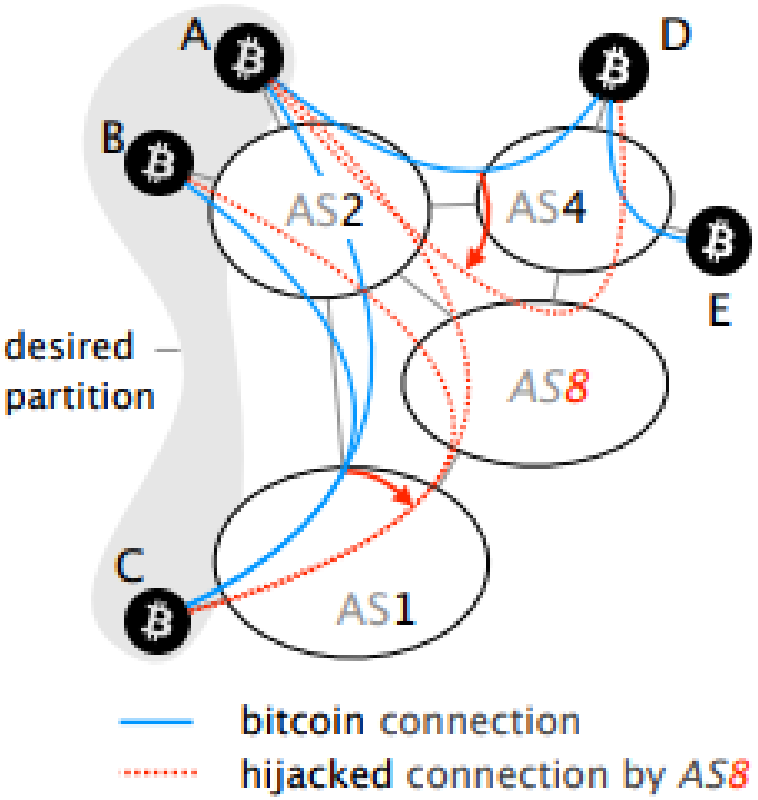
Not all partition are feasible in practice:
some connections cannot be intercepted

Bitcoin connections established…

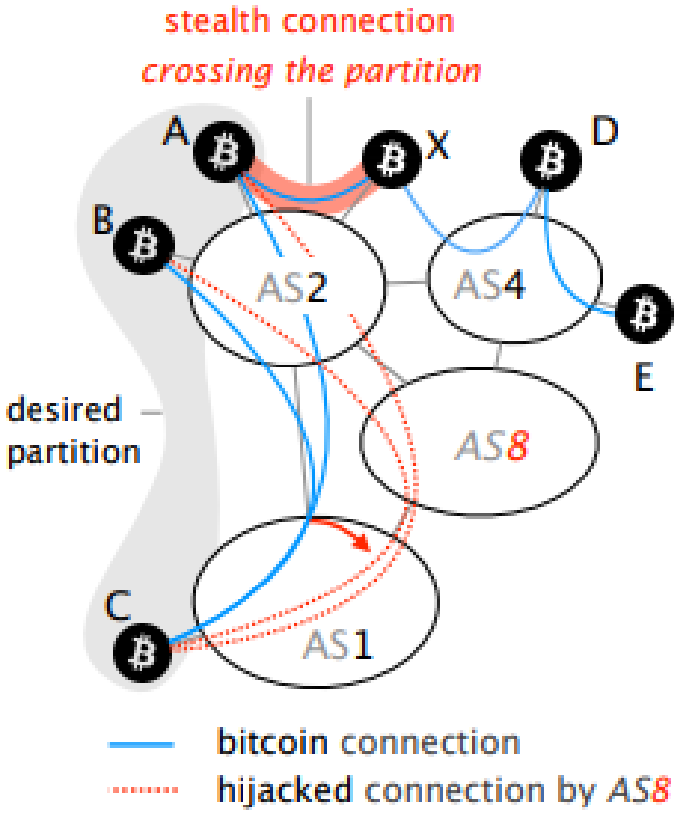
- within a mining pool
- within an AS
- between mining pools with private agreements

cannot be hijacked (usually)

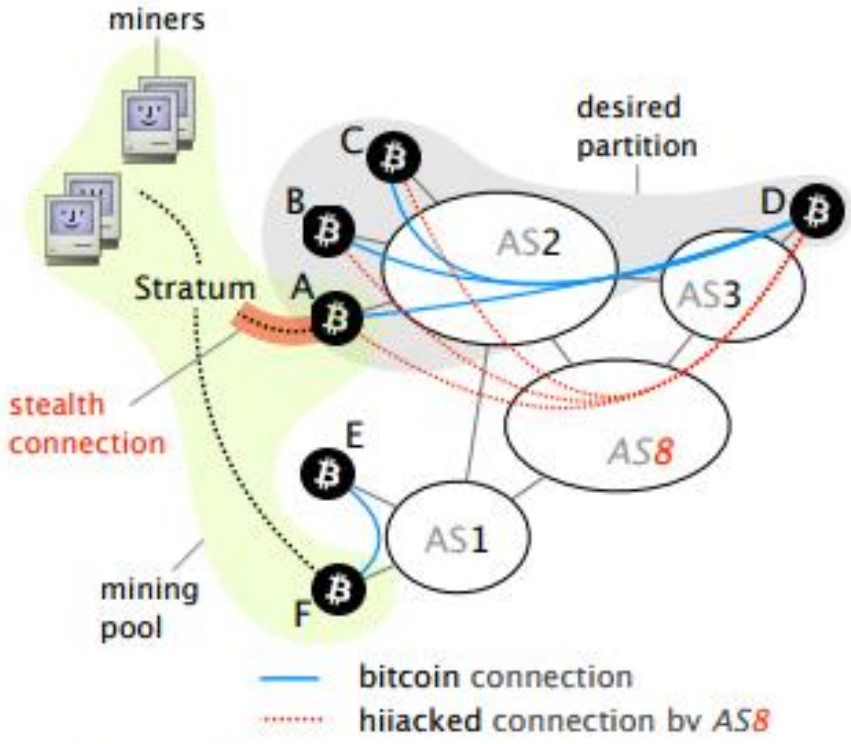
Attack Scenario (partition)



(a) Feasible partition



(b) Infeasible partition because of intra-AS connections



(c) Infeasible partition because of intra-pool connections

Attack Scenario (partition)

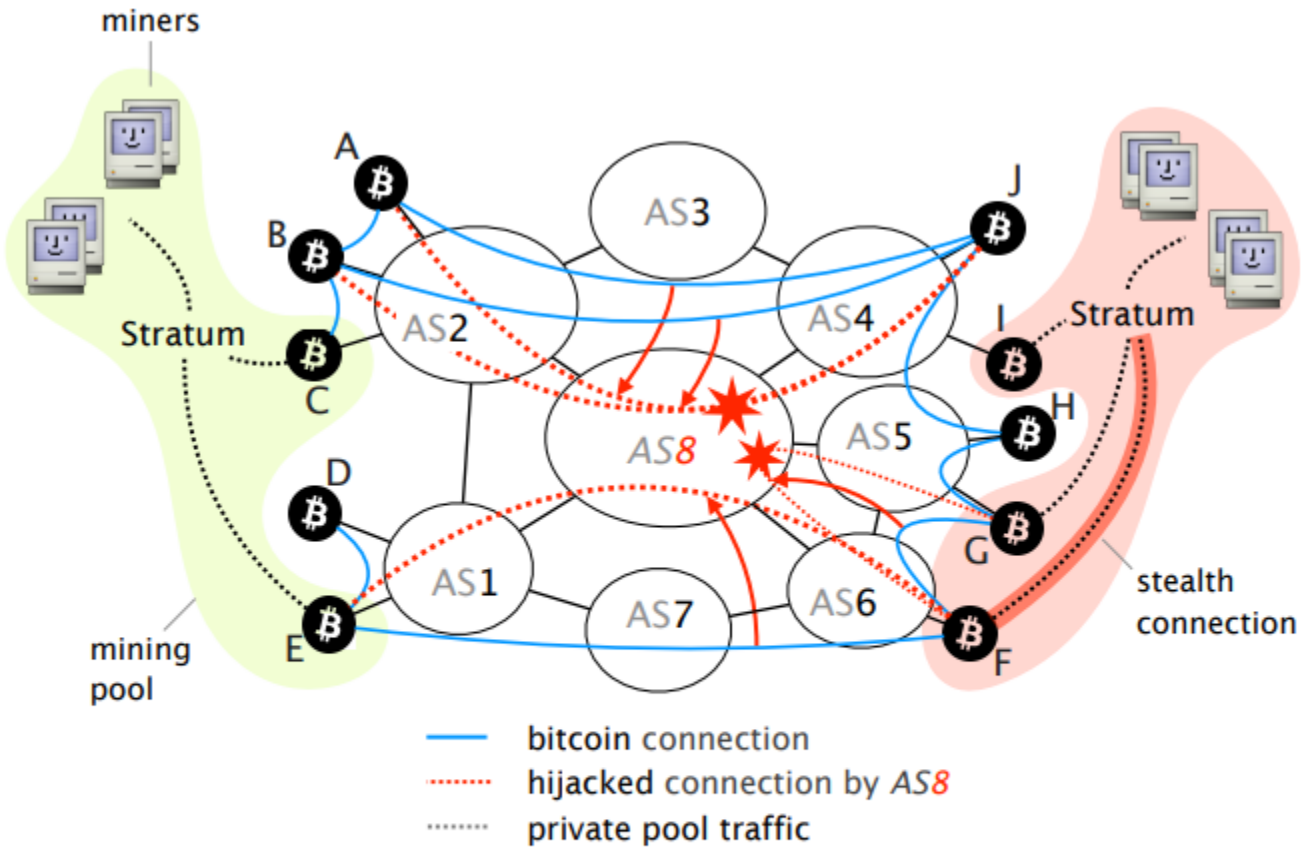


Fig. 1: Illustration of how an AS-level adversary (AS8) can intercept Bitcoin traffic by hijacking prefixes to isolate the set of nodes $P = (A, B, C, D, E, F)$.

Attack Scenario (partition)

Algorithm 1: Partitioning algorithm.

Input: - P, a set of Bitcoin IP addresses to disconnect from the rest of the Bitcoin network; and
- $S = [pkt_1, \cdots]$, an infinite packet stream of diverted Bitcoin traffic resulting from the hijack of the prefixes pertaining to P.

Output: False if there is no node $\in P$ that can be verifiably isolated;

```
1  enforce_partition(P, S):
2  begin
3      U ← ∅ ;
4      L ← ∅ ;
5      while P \ (L ∪ U) ≠ ∅ do
6          for pkt ∈ S do
7              if pkt.ip_src ∈ P ∧ pkt.ip_src ∉ L then
8                  last_seen[pkt.ip_dst] = now() ;
9                  U ← U \ {pkt.ip_src} ;
10                 detect_leakage(U, pkt) ;
11             else
12                 drop(pkt) ;
13             for src ∈ P ∧ src ∉ L do
14                 if last_seen[src] > now() − threshold then
15                     U ← U ∪ {src}
16     return false ;
```

Algorithm 2: Leakage detection algorithm.

Input: - U, a set of Bitcoin IP addresses the attacker cannot monitor; and
- pkt, a (parsed) diverted Bitcoin packet.

```
1  detect_leakage(U, pkt):
2  begin
3      if contains_block(pkt) ∨ contains_inv(pkt) then
4          if hash(pkt) ∈ Blocks(¬(P \ L)) then
5              L ← L ∪ {pkt.ip_src} ;
6              drop(pkt) ;
```

Attack Scenario (delay)

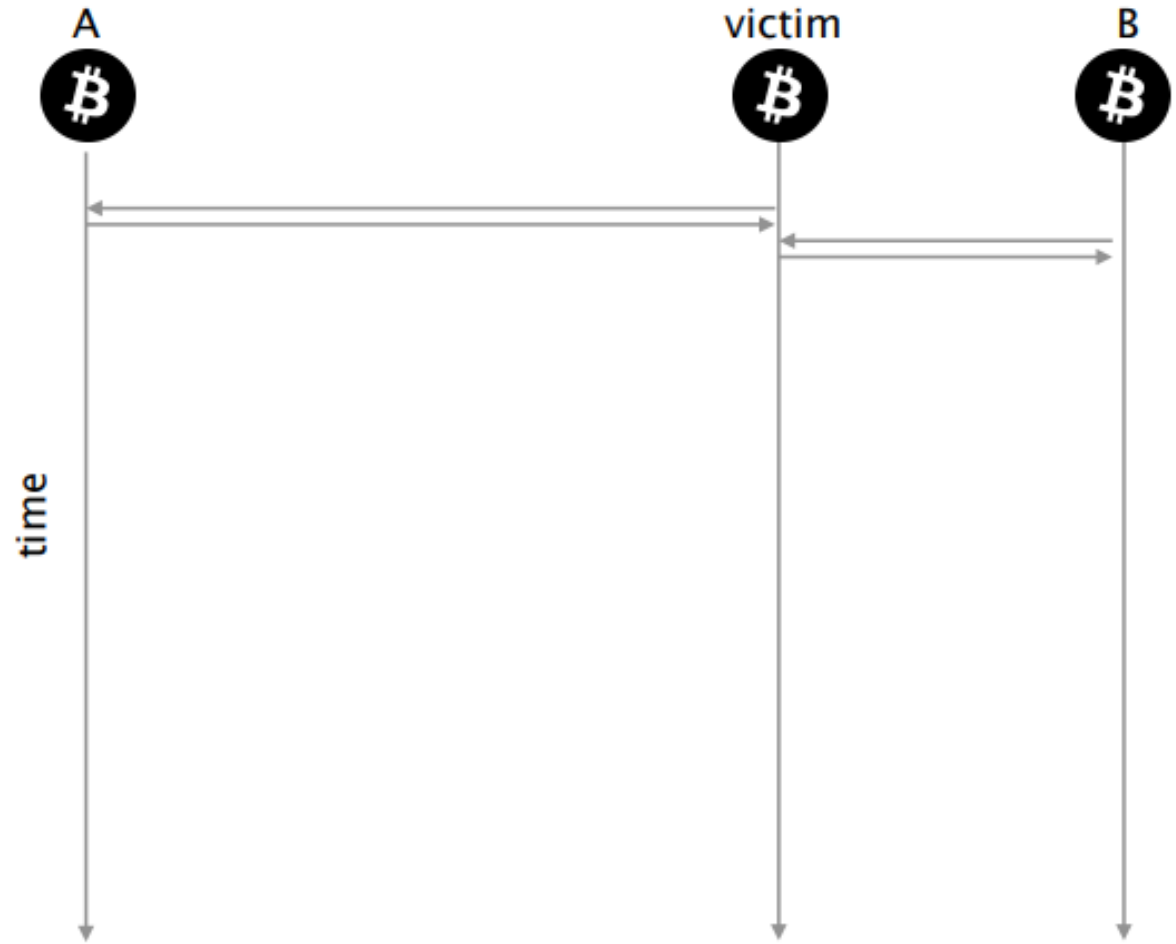
- Before describe delay attack, there are three phase of gossiping blocks
  - INV: Initiate message which containing the hash of the announced block
  - GETDATA: If the hash value is appropriate, requesting message of block data
  - BLOCK: Response message of GETDATA which contains every information of whole block

- After GETDATA message, the requester waits 20 minutes for arriving BLOCK message

- The delay attack has two type
  - Intercepting outgoing traffic
  - Intercepting incoming traffic

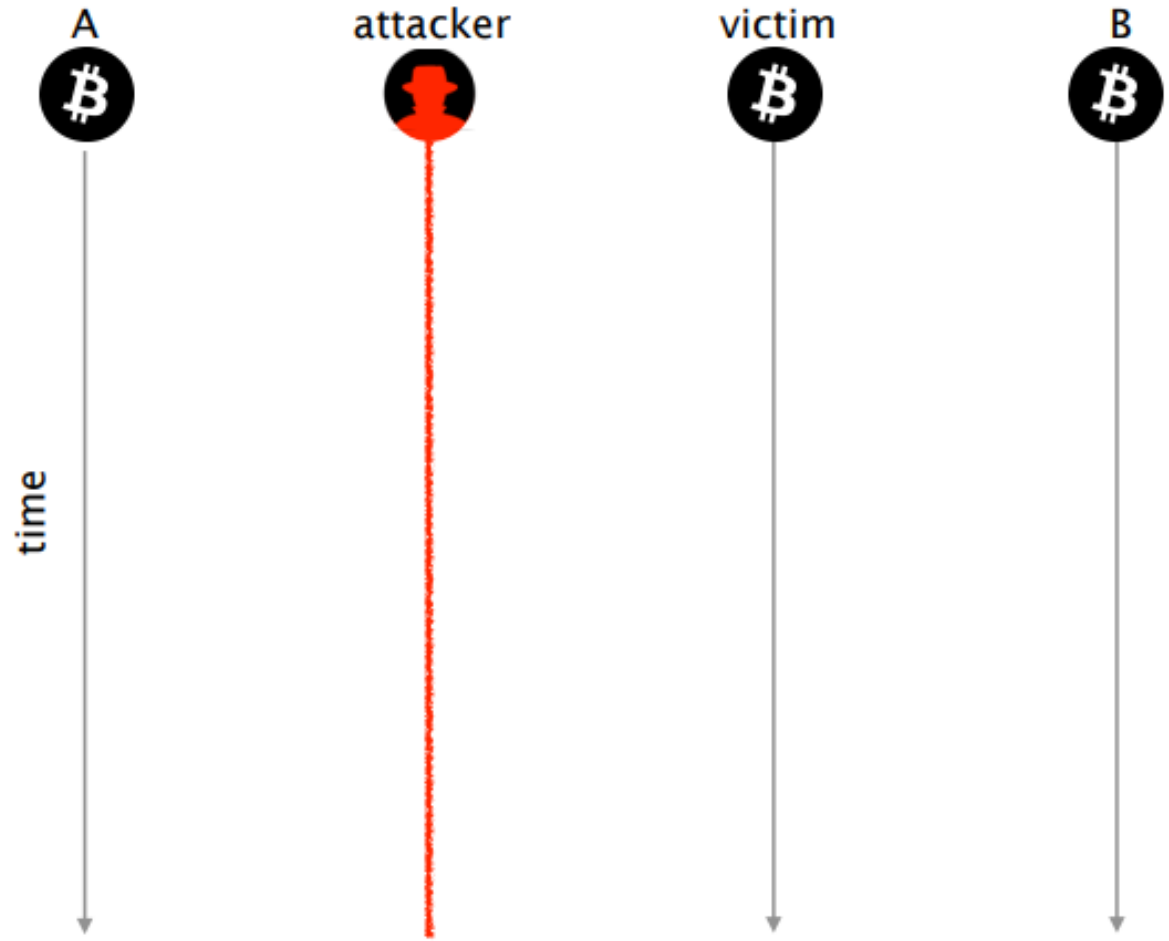
Attack Scenario (delay)

Consider these three Bitcoin nodes



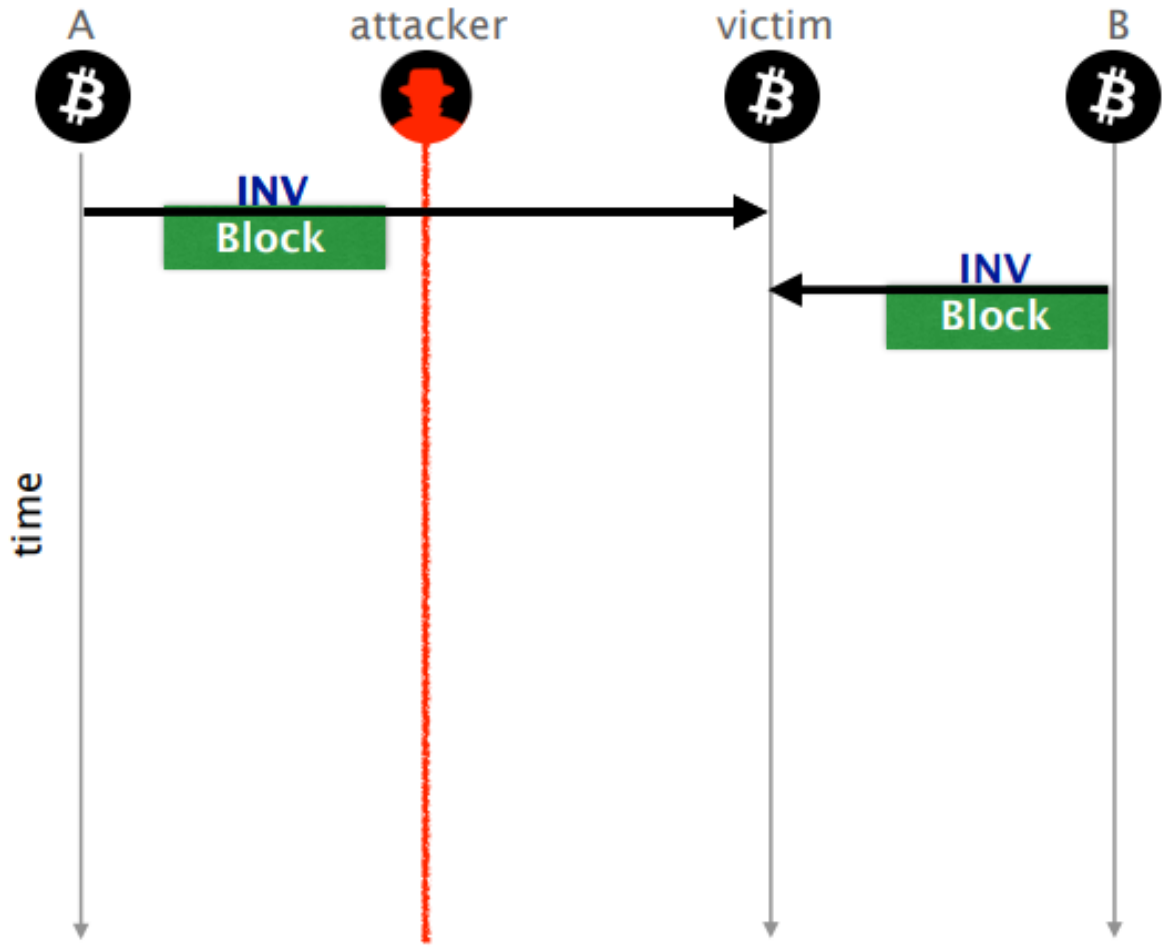
Attack Scenario (delay)

An attacker wishes to delay the block propagation towards the victim



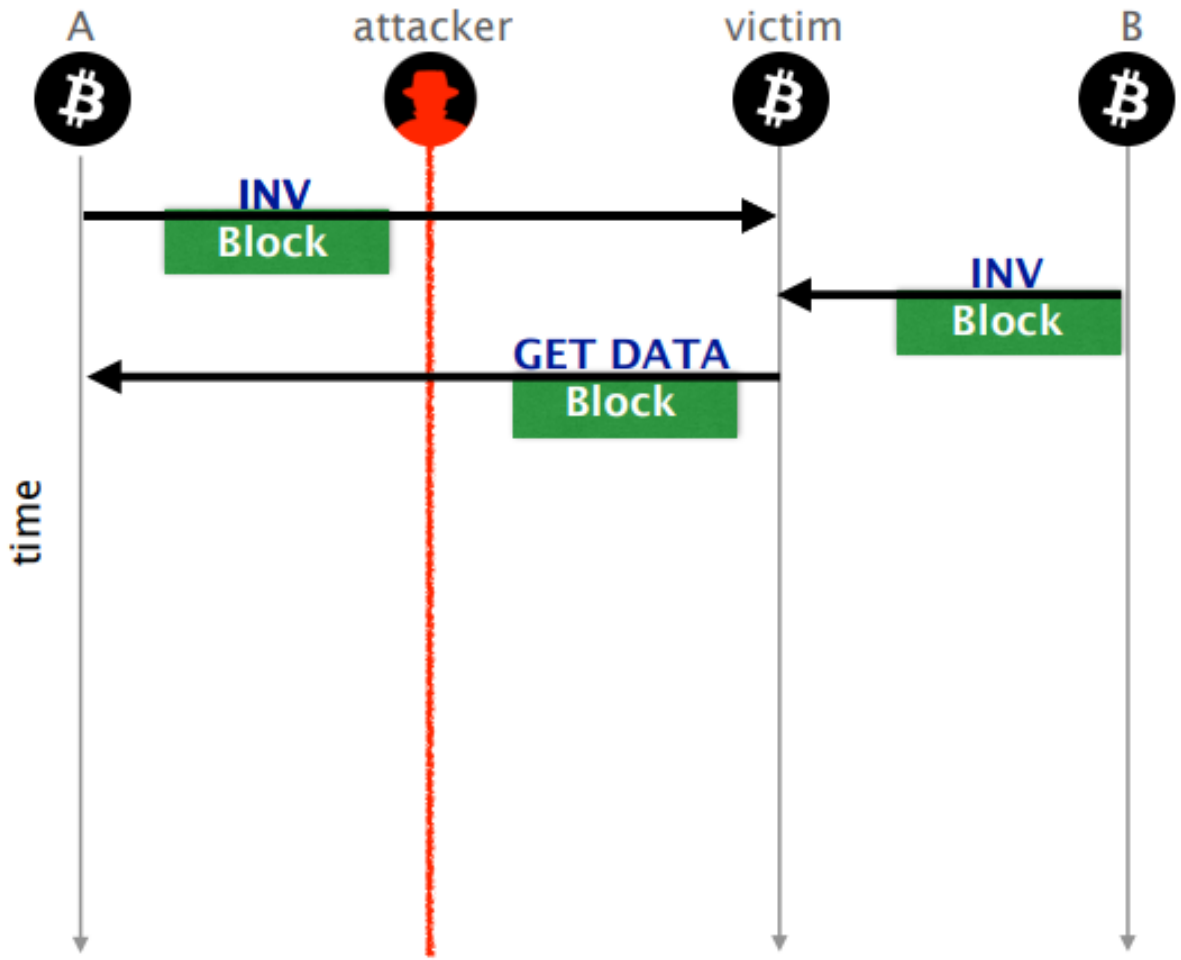
Attack Scenario (delay)

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



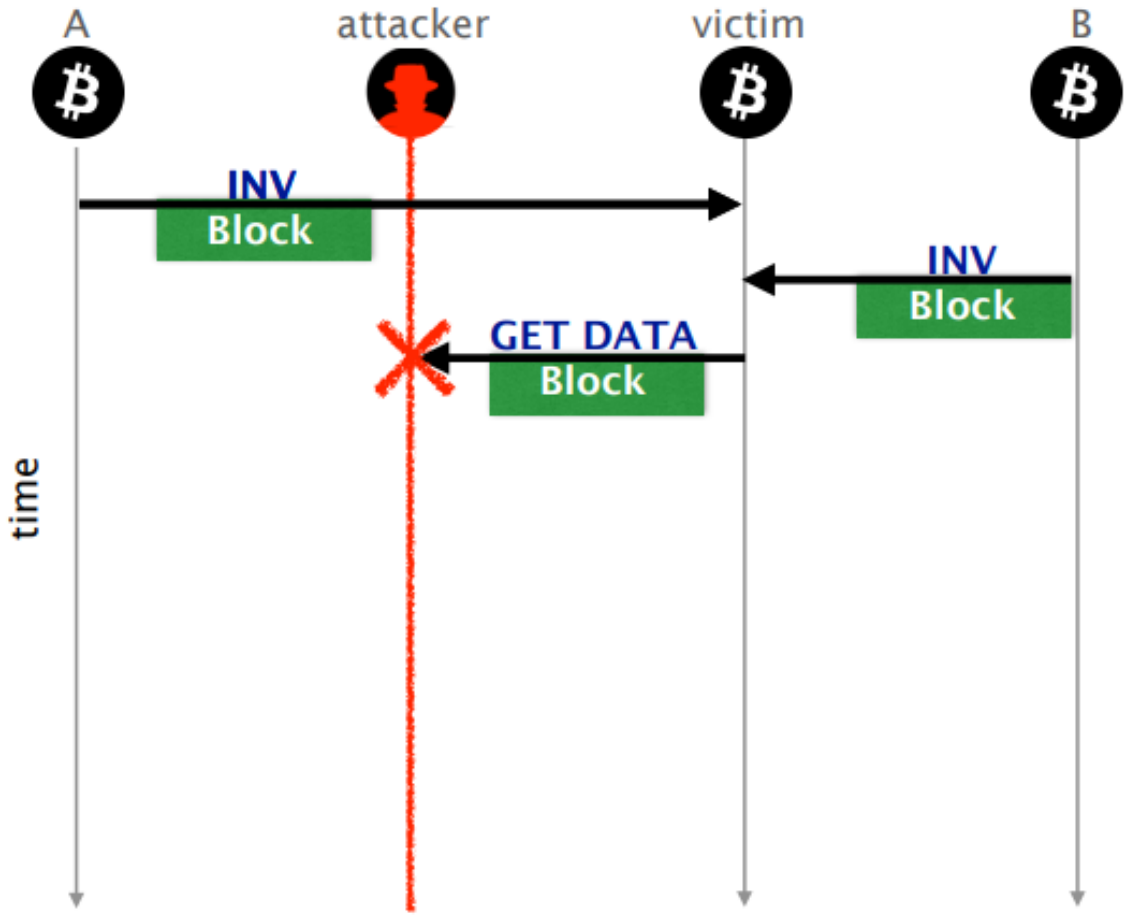
Attack Scenario (delay)

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



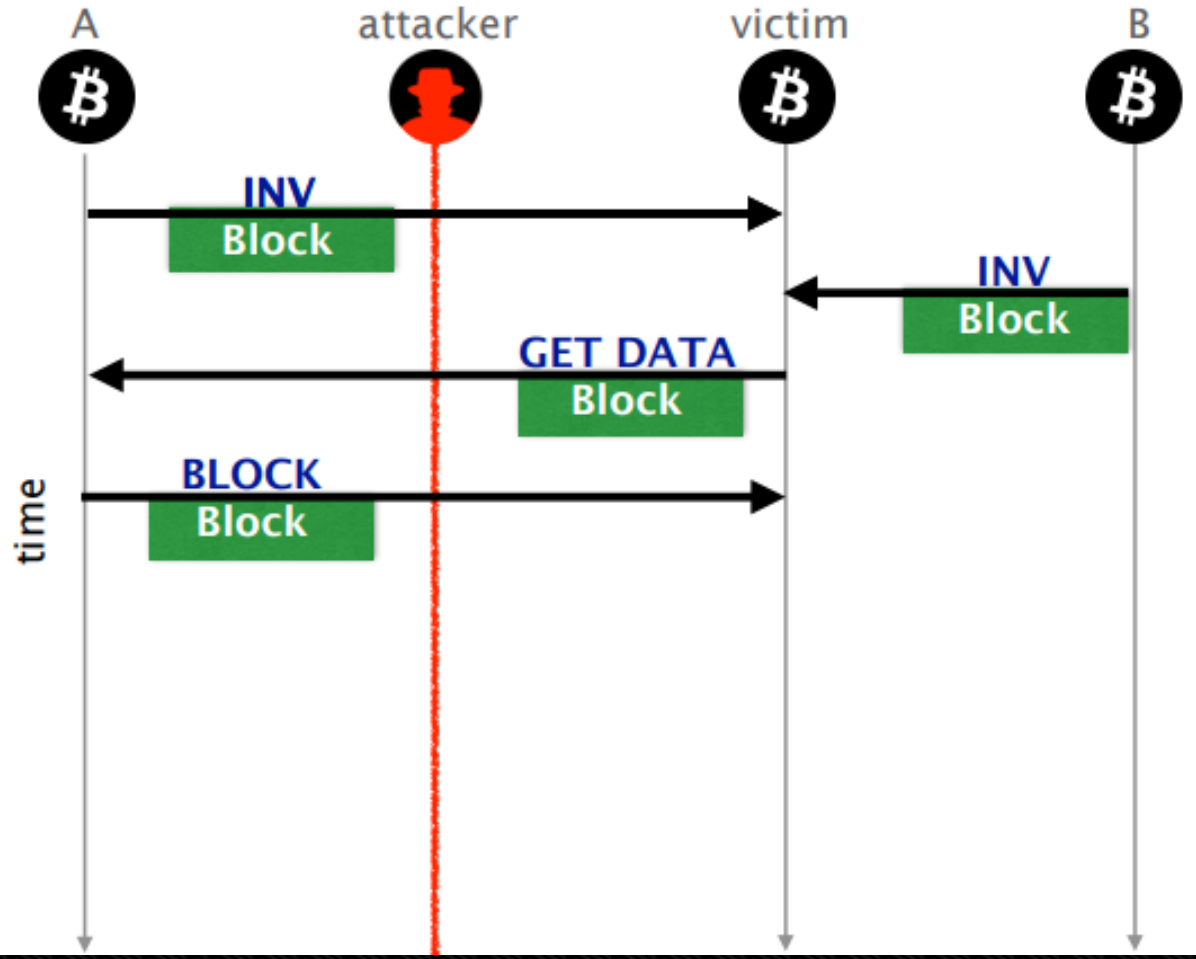
Attack Scenario (delay)

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



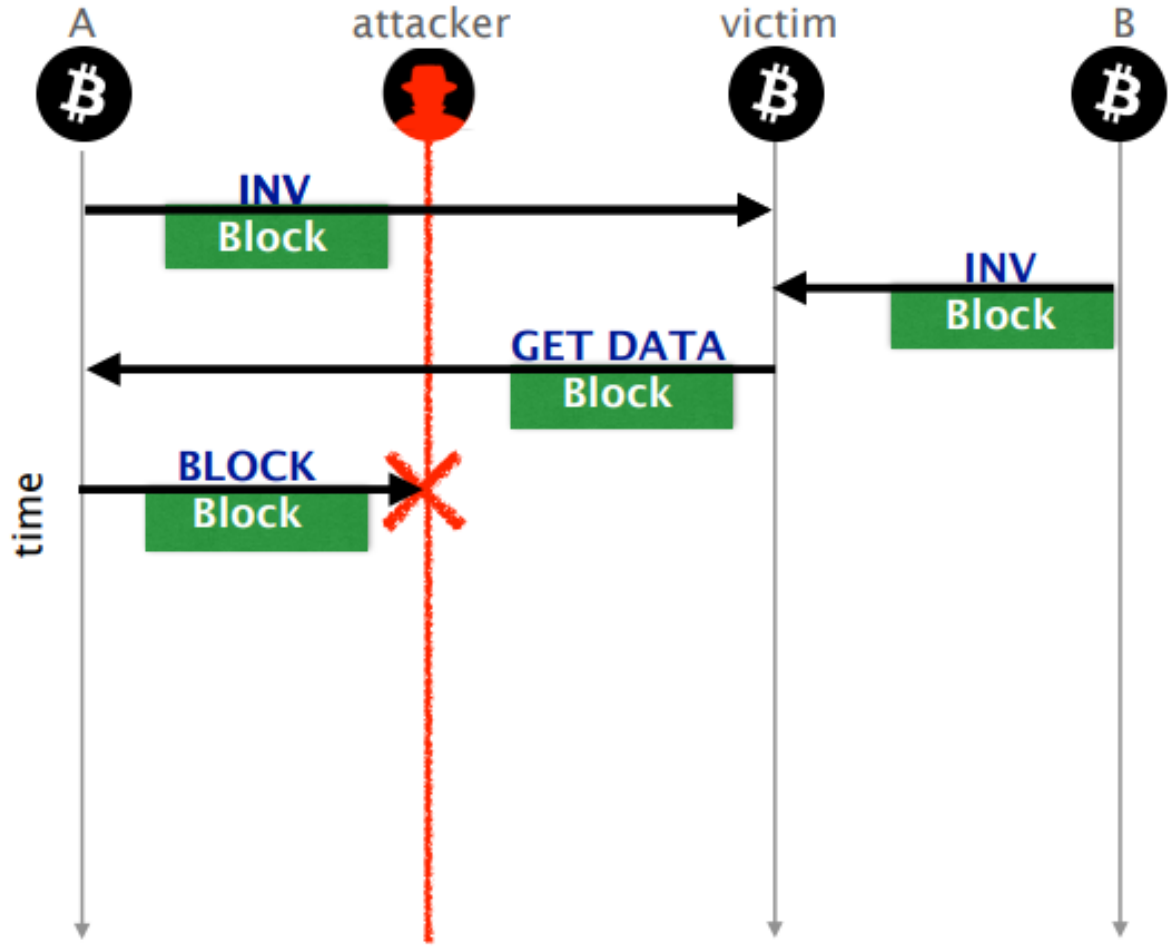
Attack Scenario (delay)

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



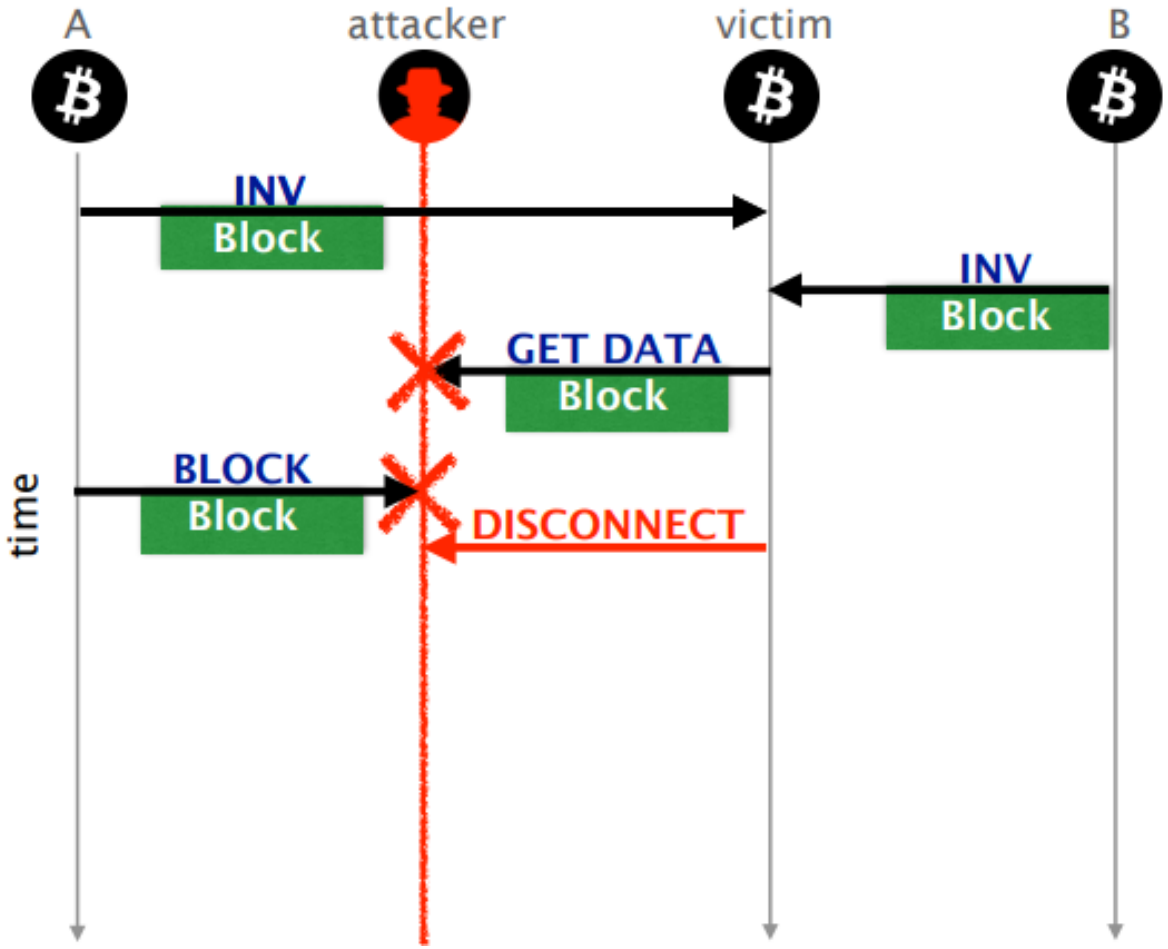
Attack Scenario (delay)

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



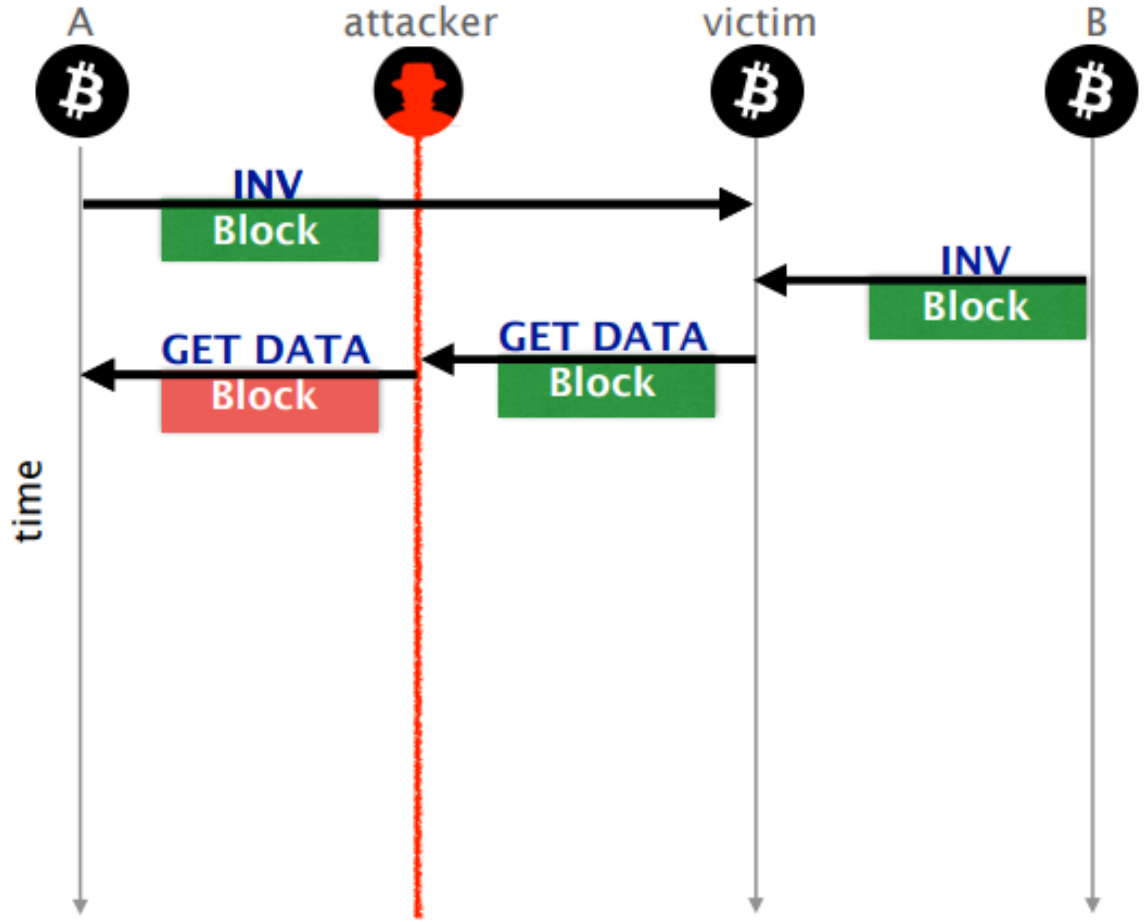
Attack Scenario (delay)

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



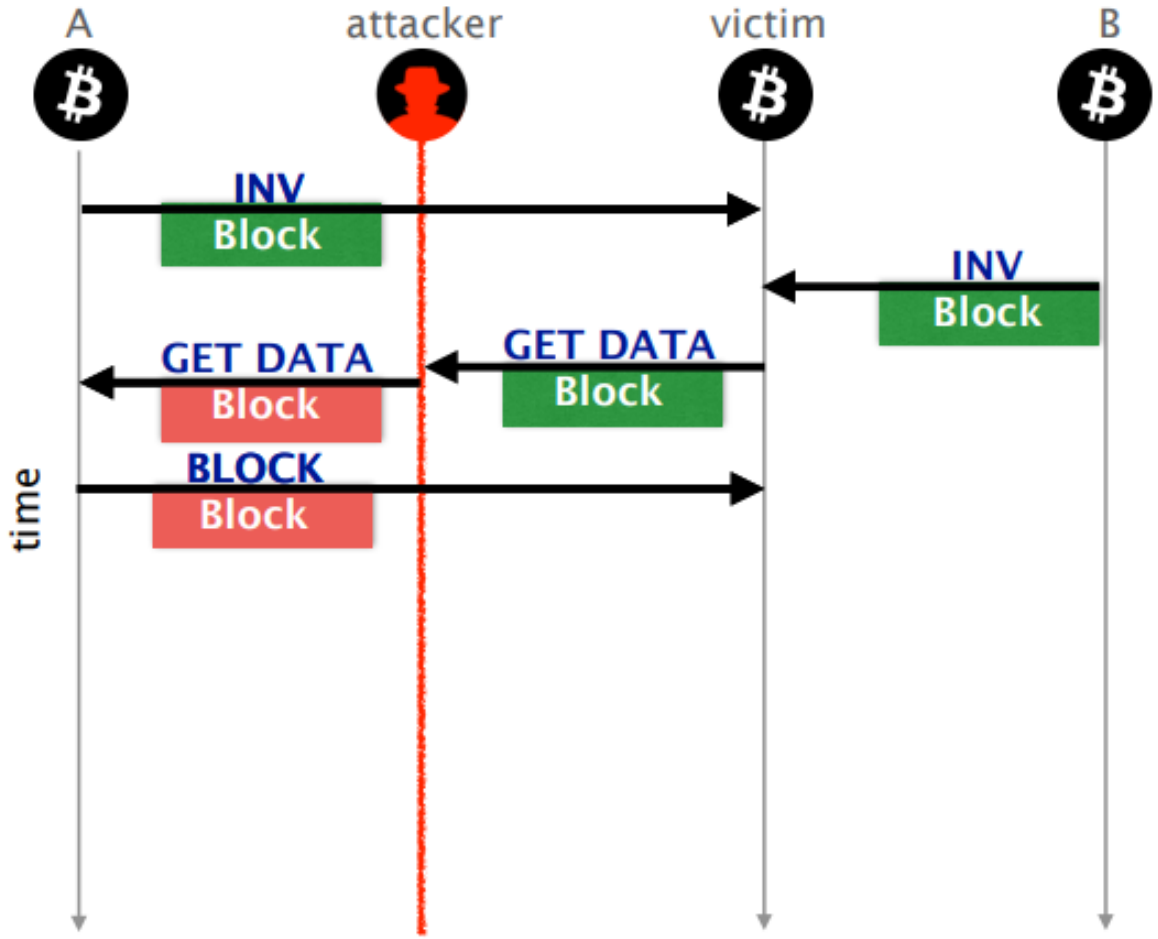
Attack Scenario (delay)

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



Attack Scenario (delay)

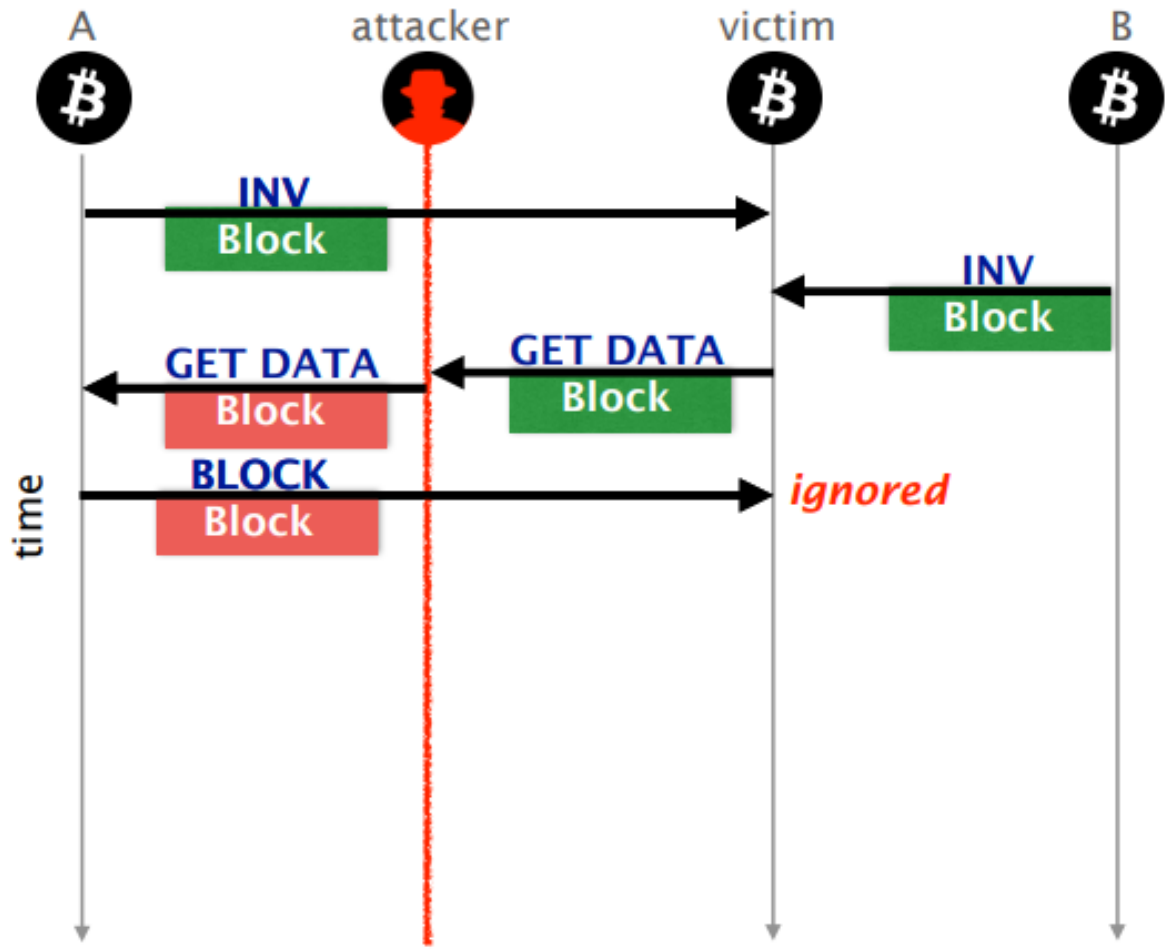
By modifying the ID of the requested block,
the attacker triggers the delivery of an older **block**



Attack Scenario (delay)

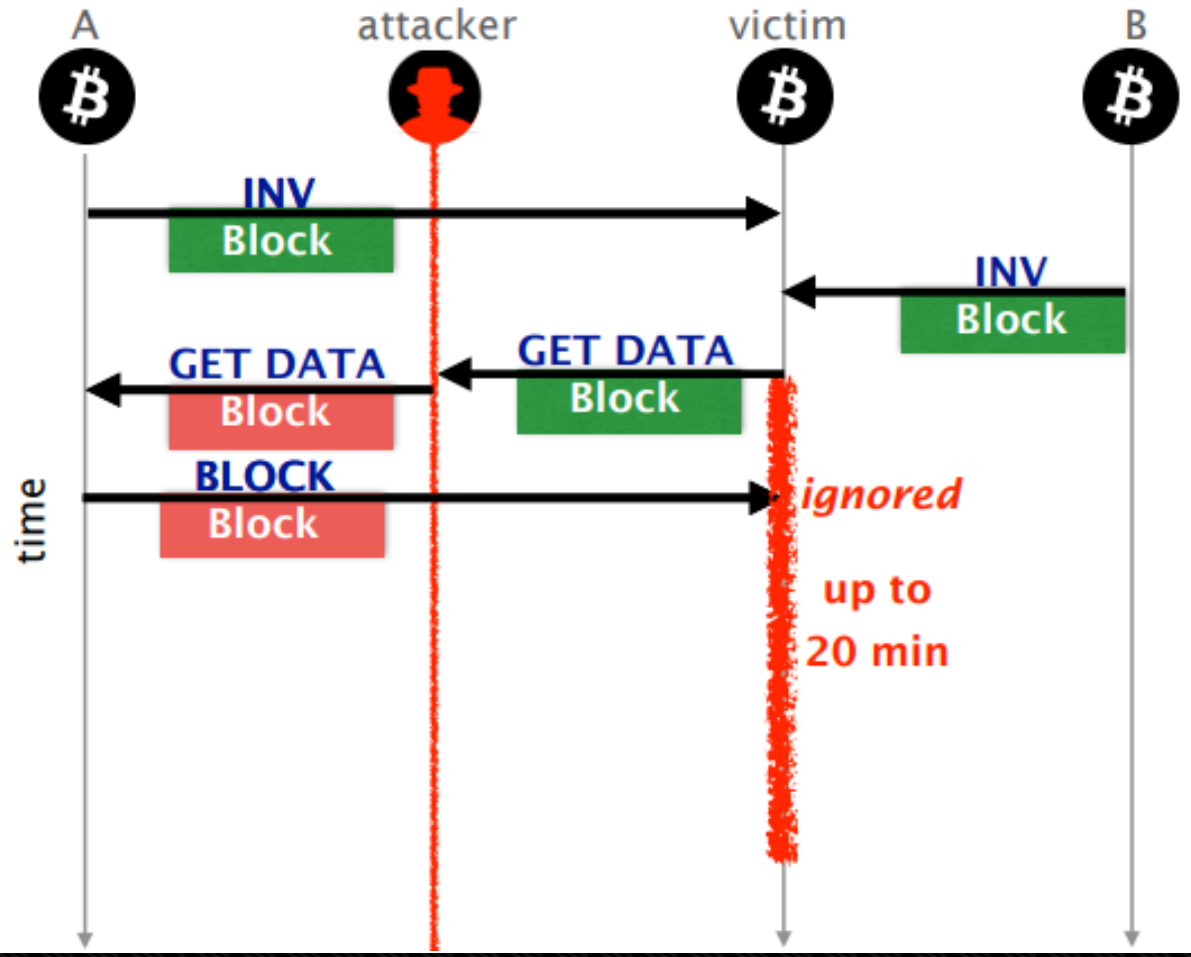
The delivery of an older block triggers

no error message at the victim



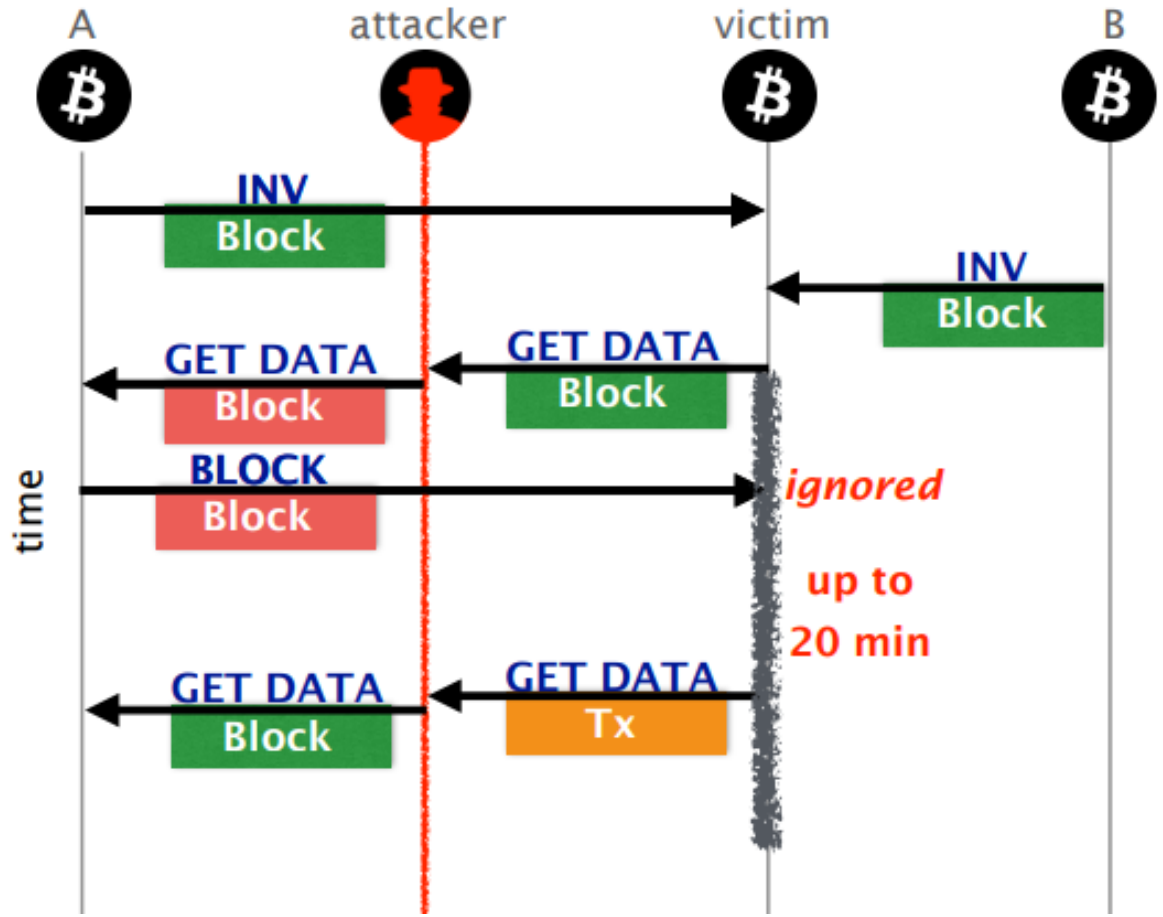
Attack Scenario (delay)

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



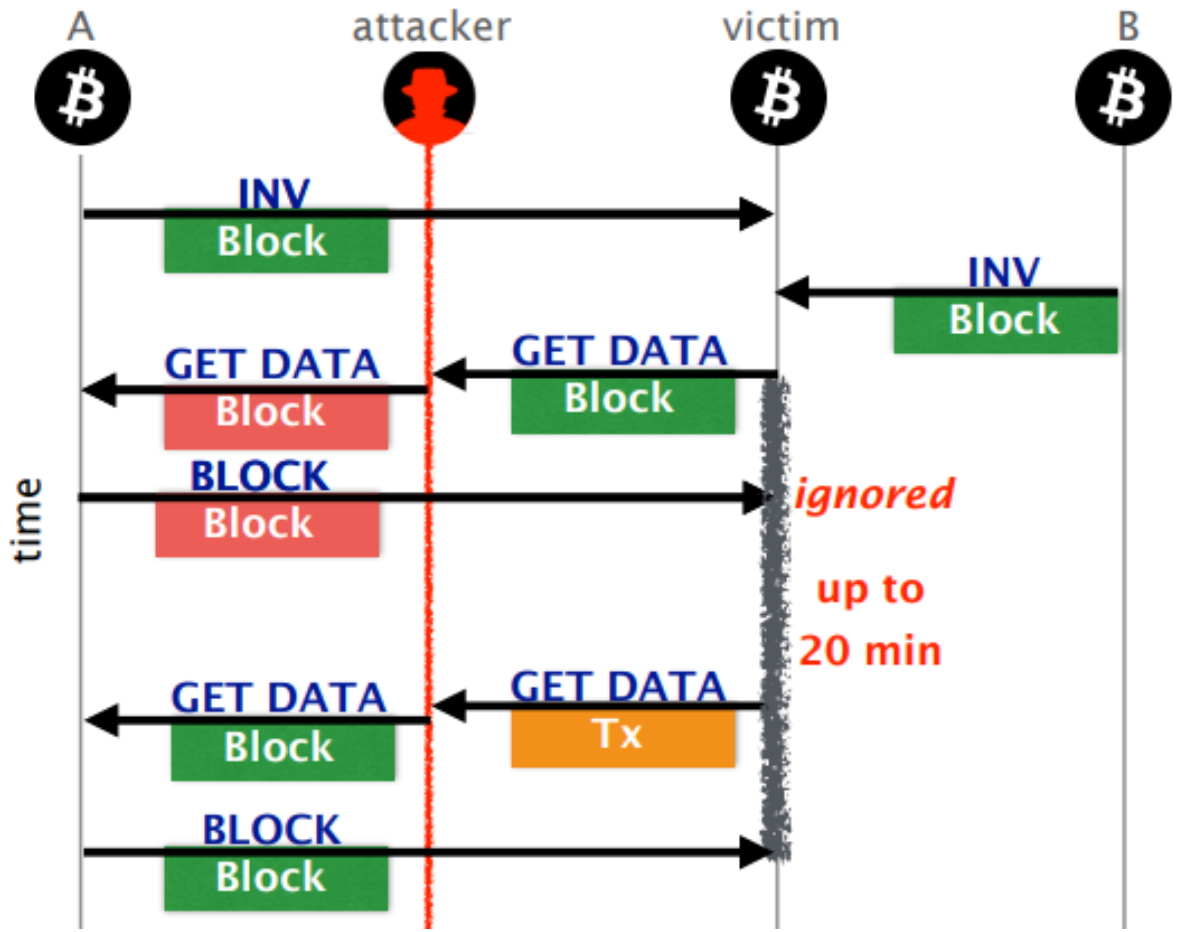
Attack Scenario(delay)

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

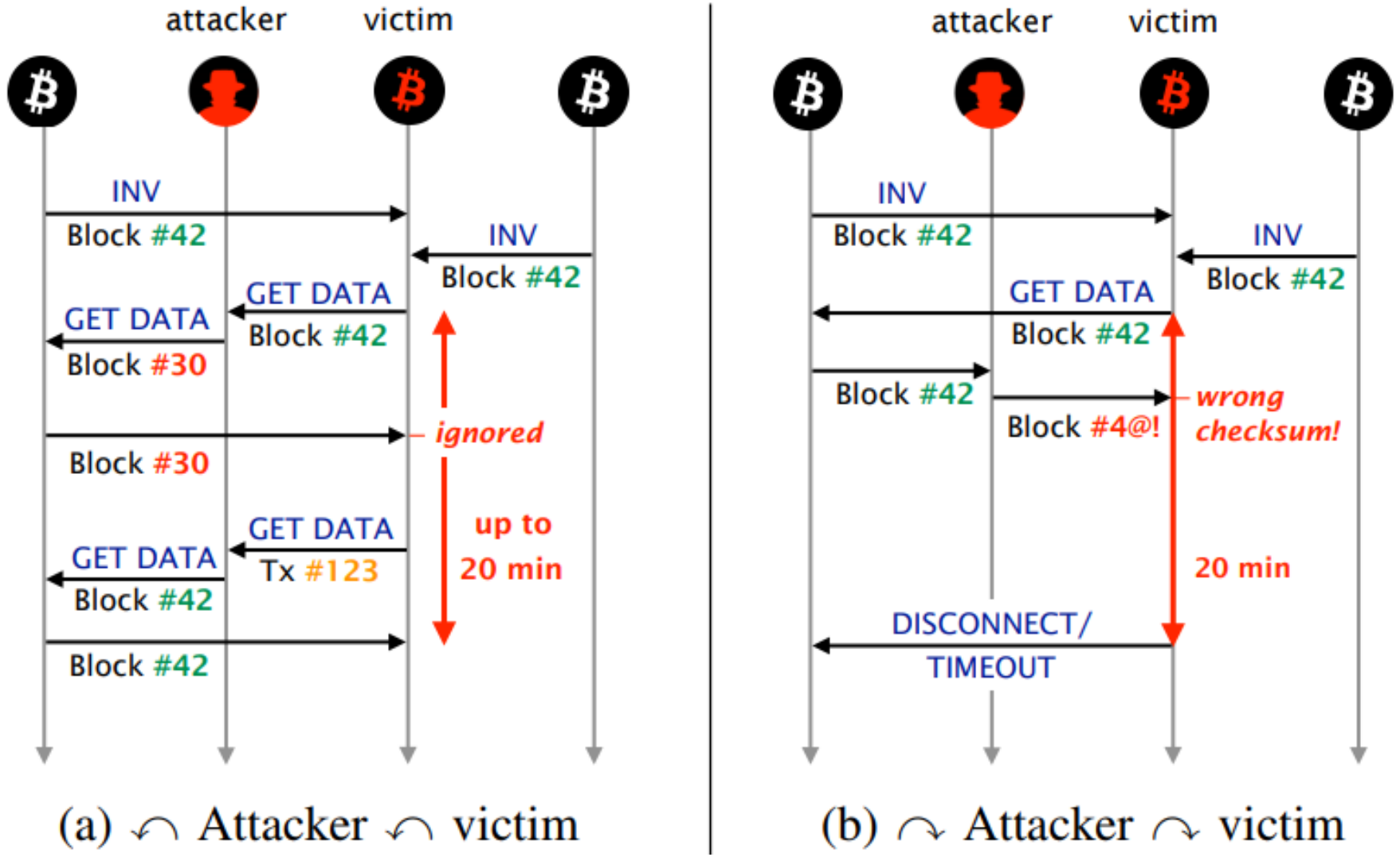


Attack Scenario(delay)

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

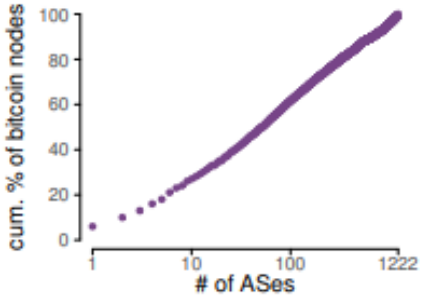


Attack Scenario(delay)

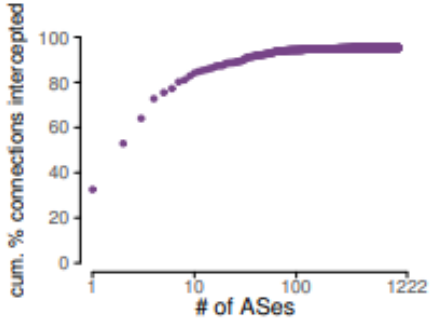


How Vulnerable Is Bitcoin To Routing Attacks

- A few ASes host most of the Bitcoin nodes
- A few ASes naturally intercept the majority of the Bitcoin traffic



(a) Only 13 ASes host 30% of the entire network, while 50 ASes host 50% of the Bitcoin network.



(b) Few ASes intercept large percentages of Bitcoin traffic: 3 of them intercept 60% of all possible Bitcoin connections.

- >90% of Bitcoin nodes are vulnerable to BGP hijacking
  - 93% of all prefixes hosting Bitcoin nodes are shorter than /24

How Vulnerable Is Bitcoin To Routing Attacks

- Diverting Bitcoin traffic via BGP is fast (takes < 2 minutes)
- Hijacking < 100 prefixes is enough to isolate ~50% of Bitcoin mining power

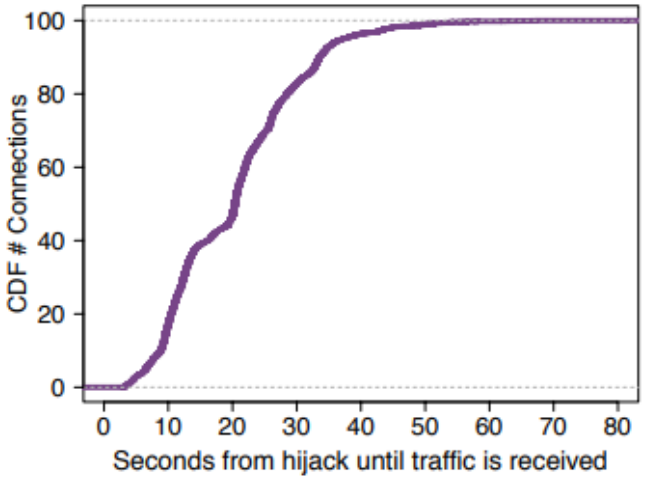


Fig. 6: Intercepting Bitcoin traffic using BGP hijack is fast and effective: all the traffic was flowing through the attacker within 90 seconds. Results computed while performing an *actual* BGP hijack against our own Bitcoin nodes.

Isolated mining power	*min. # pfxes to hijack*	*median # pfxes to hijack*	*# feasible partitions*
8%	32	70	14
30%	83	83	1
40%	37	80	8
47%	39	39	1

TABLE I: Hijacking <100 prefixes is enough to feasibly partition ~50% of the mining power. Complete table in Appendix B.

Short-term Countermeasures

- Increase the diversity of node connections
  - More connected, harder to attack like multihomed
- Monitor round-trip time (RTT)
  - The RTT towards hijacked destinations increases during the attack
- Embrace churn
  - To refresh their connections
- Prefer peers hosted in the same AS and in /24 prefixes
  - Note that network ignores about more than /24 prefix matching connection

Long-term Countermeasures

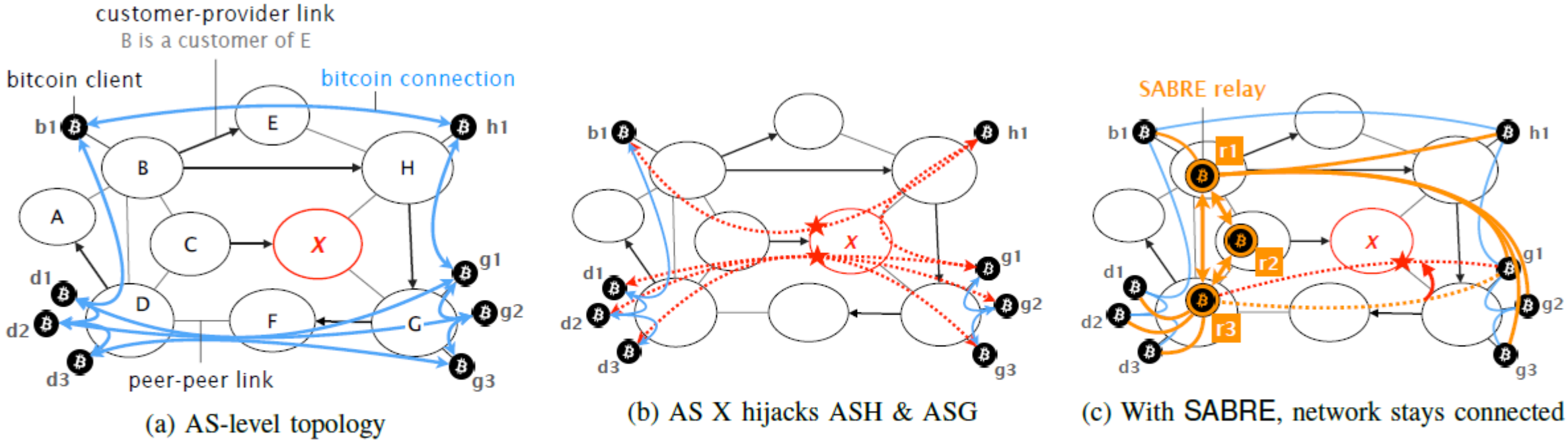
- Encrypt Bitcoin Communication and/or adopt MAC
  - Cannot modify the contents and authenticate sender

- Use distinct control and data channels
  - Currently, Bitcoin traffic is easily identifiable by filtering on the default port(8333)
  - Using randomized TCP port, it will force the AS-level adversary to maintain state to keep track of these ports.

- Request a block on multiple connections

Follow-up Paper

- SABRE: Protecting Bitcoin against Routing Attacks
  - Make transparent relay network protecting Bitcoin client from routing attacks by providing them with an extra secure channel



(a) AS-level topology

(b) AS X hijacks ASH & ASG

(c) With SABRE, network stays connected

Fig. 2: SABRE protects the Bitcoin network from AS-level adversaries aiming to partition it. Without SABRE, AS X can split the network in half by first diverting traffic destined to AS H and AS G using a BGP hijack and then dropping the corresponding connections (Fig. 2b). With SABRE, the network stays connected (Fig. 2c).

Questions?