EqualNet: A Secure and Practical Defense for Long-term Network Topology Obfuscation

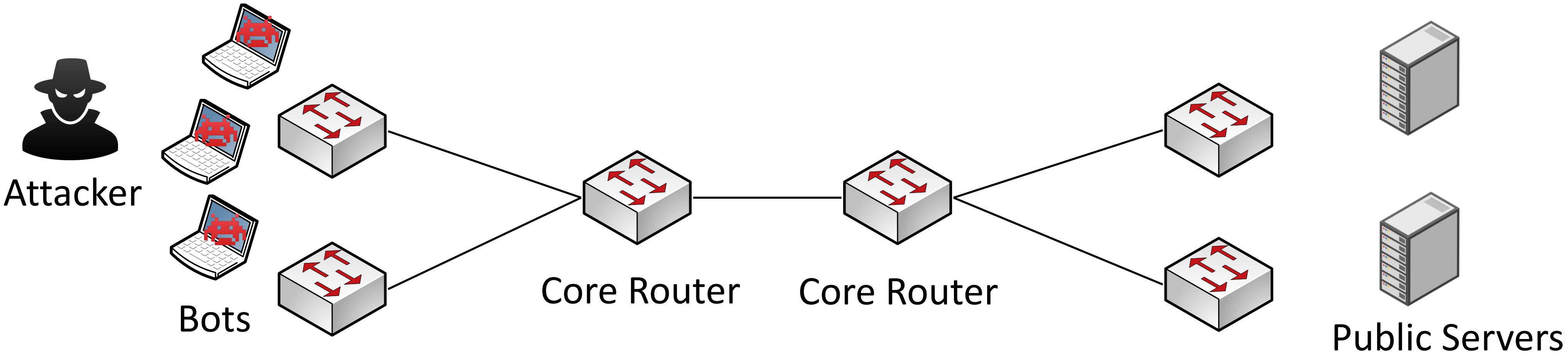
Jinwoo Kim[*], Eduard Marin[+], Mauro Conti[‡], and Seungwon Shin[*]

[*]KAIST, [+]Telefonica Research, [‡]University of Padua

KAIST Telefónica UNIVERSITÀ DEGLI STUDI DI PADOVA

Link Flooding Attacks (LFA)

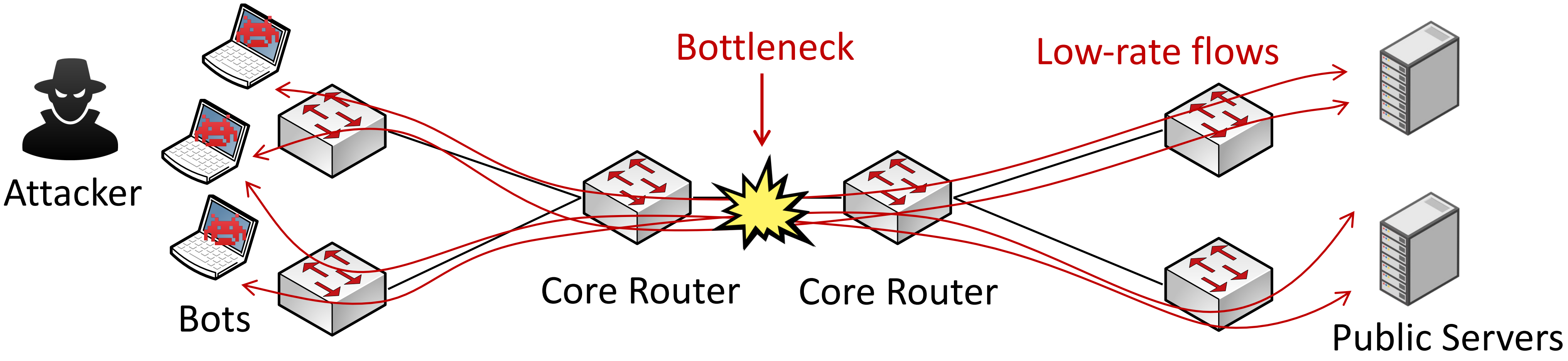
- Stealthy but powerful DDoS attacks
  - Target network infrastructure



Link Flooding Attacks (LFA)

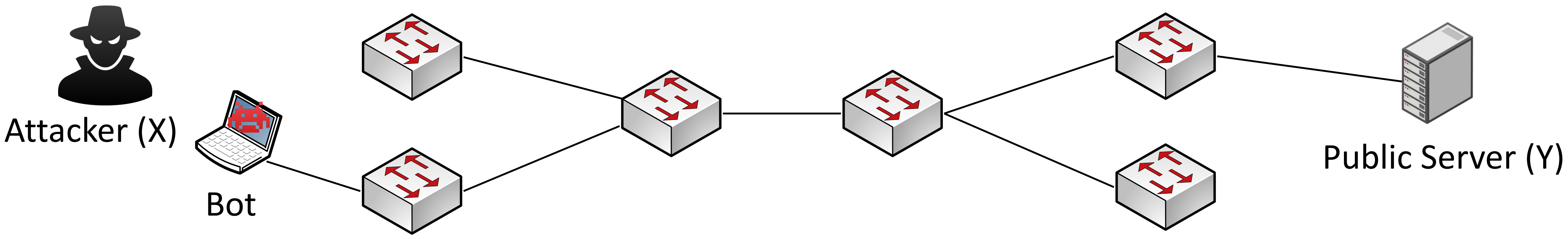
- Stealthy but powerful DDoS attacks
  - Target network infrastructure

- Cause congestion on core routers or links → ***bottlenecks***
  - "Able to cut off 53% of Internet connections in some US states" [1]



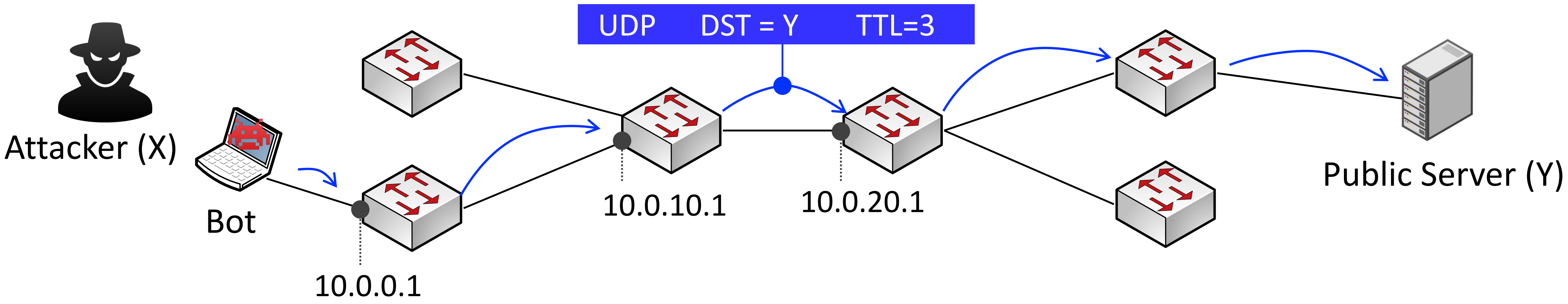
[1] The Crossfire attacks, S&P '13

How attackers find bottlenecks?



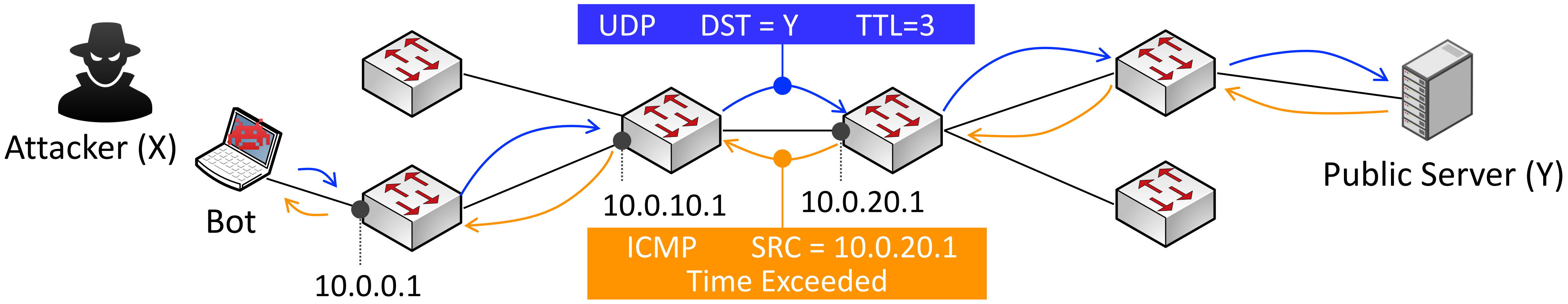
How attackers find bottlenecks?

1. Scan a target network via path tracing tools (e.g., traceroute)
   - By sending probing packets (i.e., low TTL packets) to public servers



How attackers find bottlenecks?

1. Scan a target network via path tracing tools (e.g., traceroute)
   - By sending probing packets (i.e., low TTL packets) to public servers

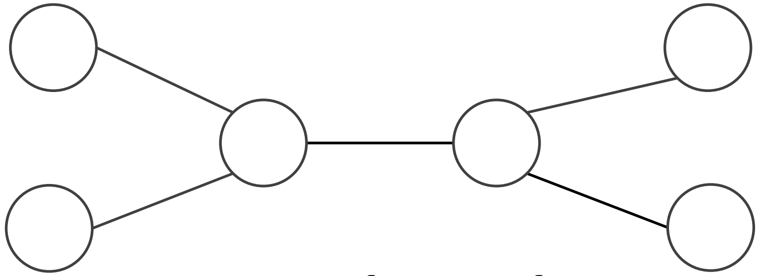


How attackers find bottlenecks?

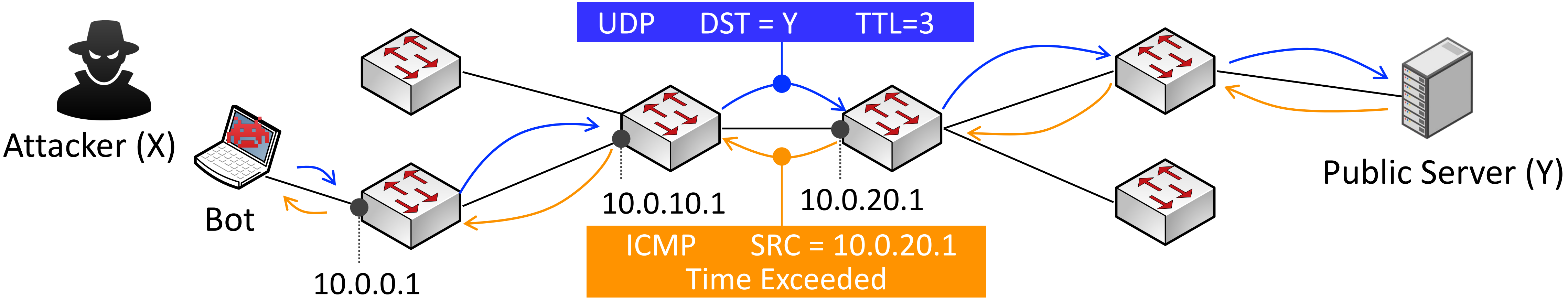
1. Scan a target network via path tracing tools (e.g., traceroute)
   – By sending probing packets (i.e., low TTL packets) to public servers

```
$ traceroute X → Y
1    10.0.0.1    2.367ms
2    10.0.10.1   1.977ms
3    10.0.20.1   2.042ms
4    ...
```

Topology Inferring

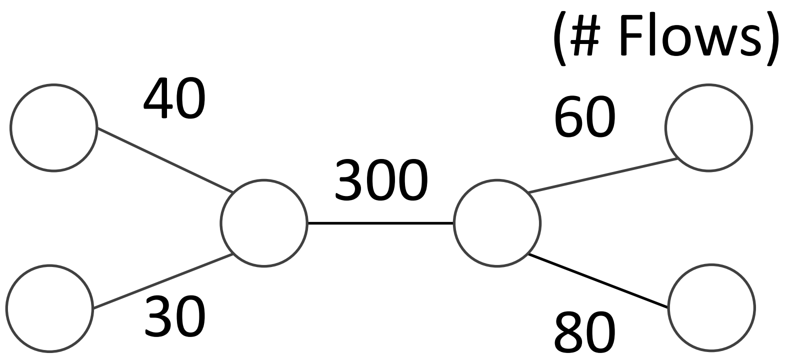


Network topology



How attackers find bottlenecks?

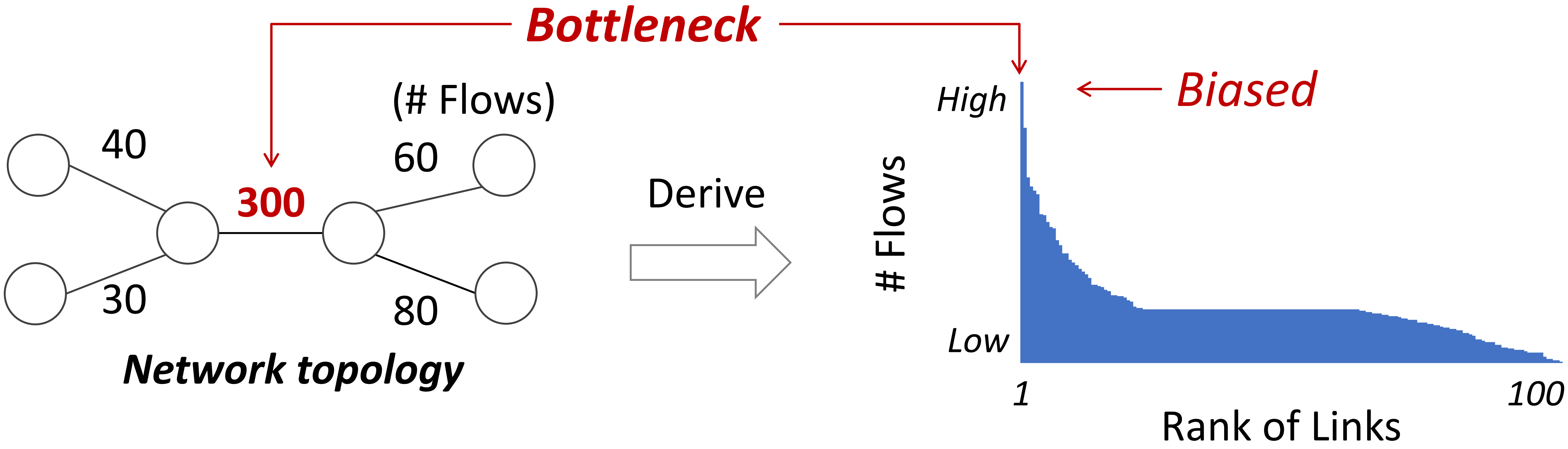
2. Analyze the network topology



Network topology

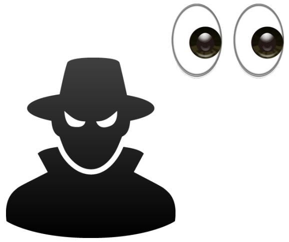
How attackers find bottlenecks?

2. Analyze the network topology
   - Derive a flow distribution to see which links are ***popular***
   - Choose the links whose # flows are higher than others

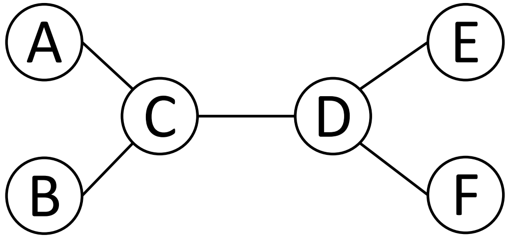


Network Topology Obfuscation

- Adopt the cyber deception strategy
  - To mitigate LFAs ***proactively*** by deceiving attackers



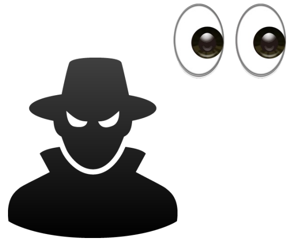
Attacker



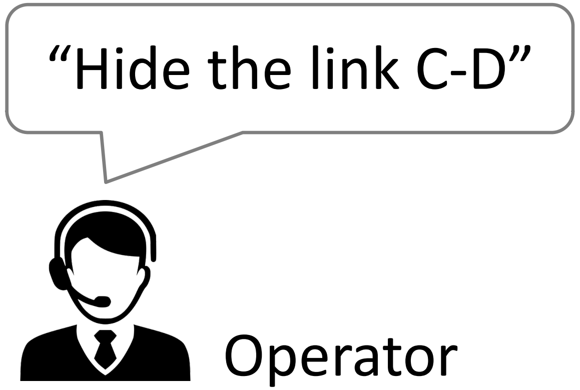
Network Topology

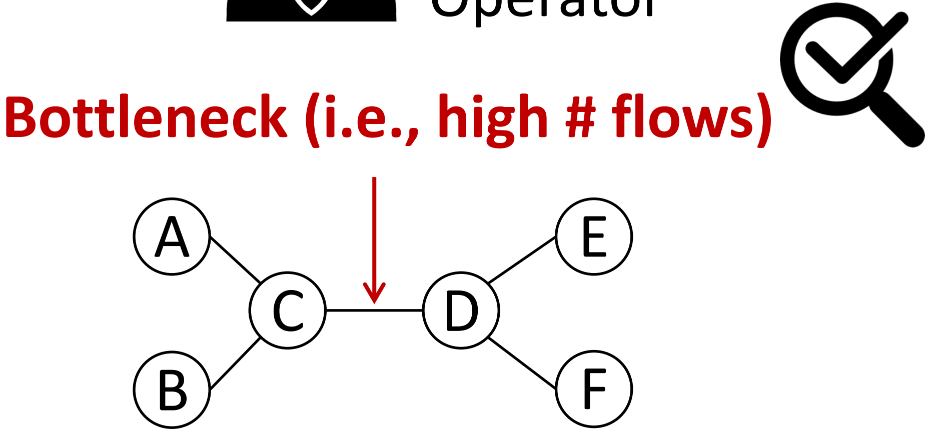
Network Topology Obfuscation

- Adopt the cyber deception strategy
  - To mitigate LFAs ***proactively*** by deceiving attackers
- Pinpoint potential bottlenecks
  - By simulating attacker flows in a network



Attacker

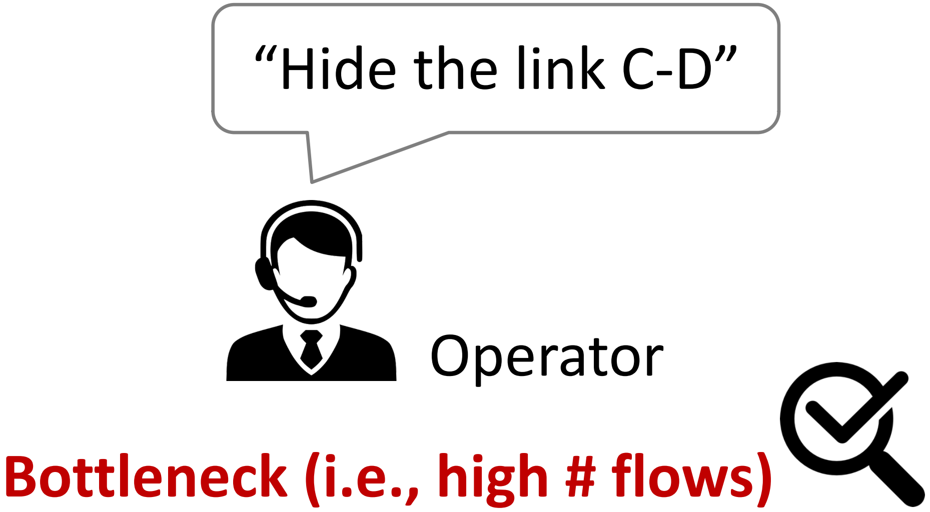


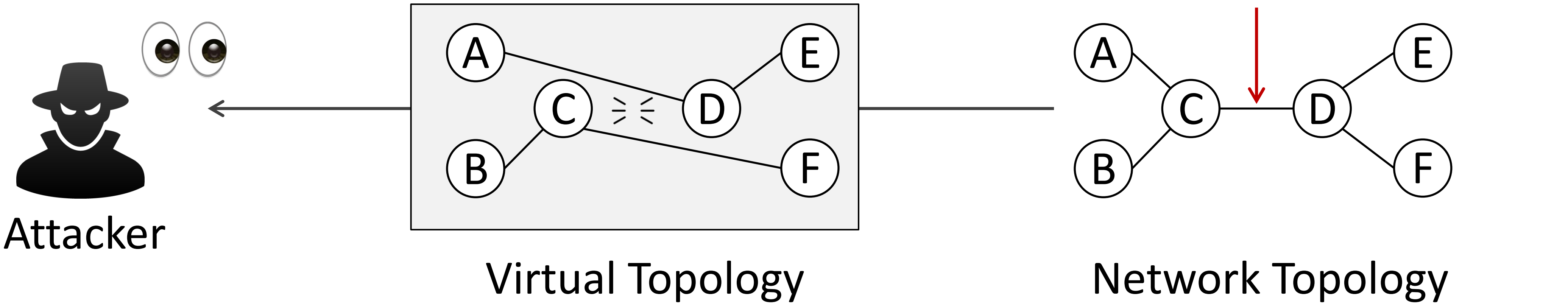


Network Topology

Network Topology Obfuscation

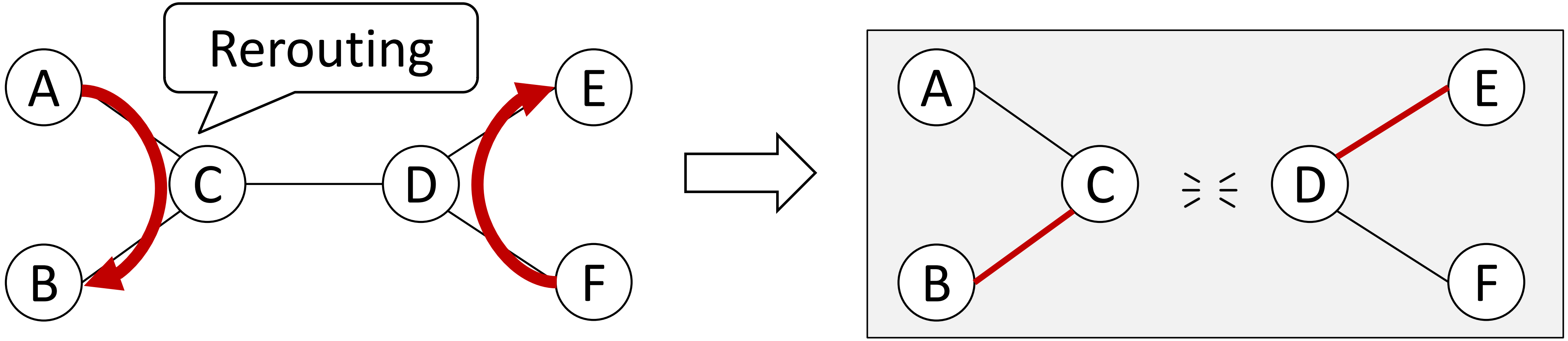
- Adopt the cyber deception strategy
  - To mitigate LFAs ***proactively*** by deceiving attackers
- Pinpoint potential bottlenecks
  - By simulating attacker flows in a network
- Create a ***virtual network topology***
  - Hide potential bottlenecks of a network





Prior Solutions

- Rerouting probing packets to nearby links
  - e.g., LinkBait



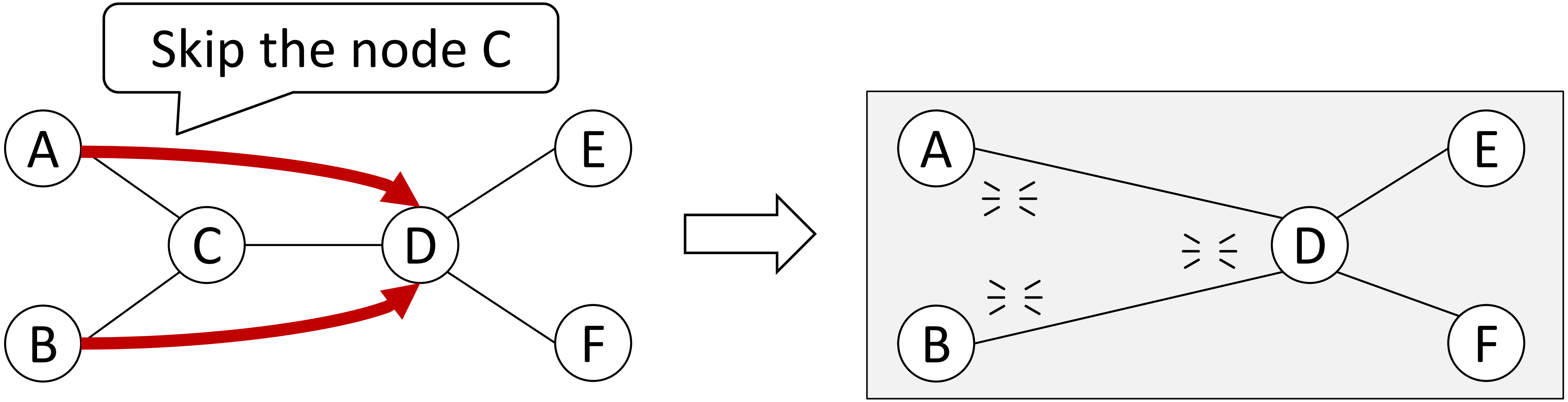
Network Topology

Virtual Topology

Attacker

Prior Solutions

- Skipping probing packets partially
  - e.g., NetHide, Trassare et al.

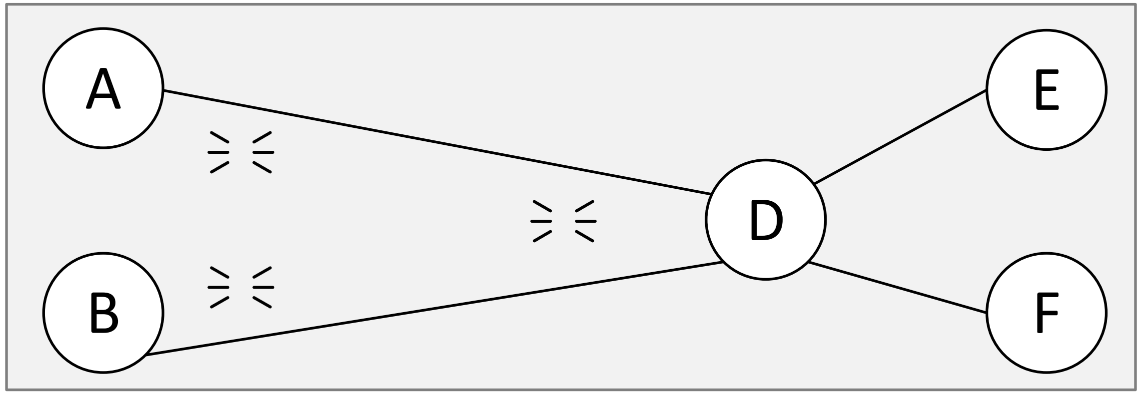


Network Topology

Virtual Topology

Attacker

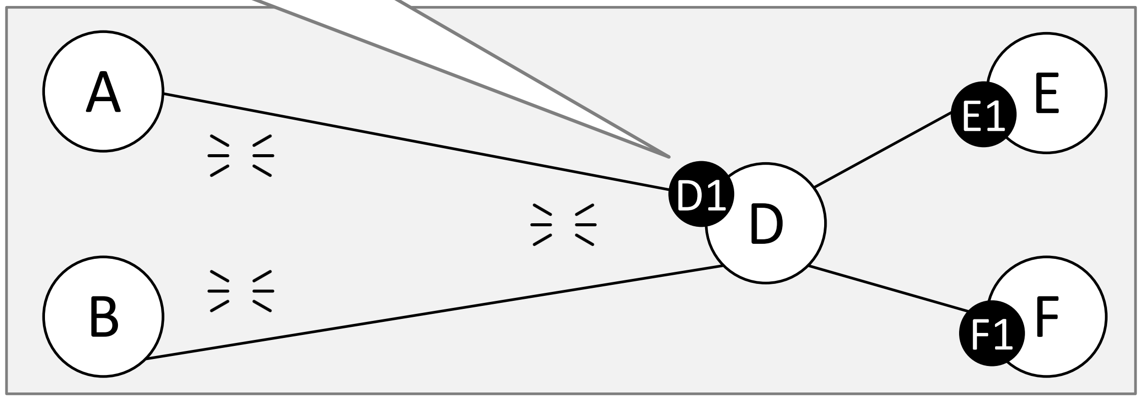
Limitations of Prior Solutions



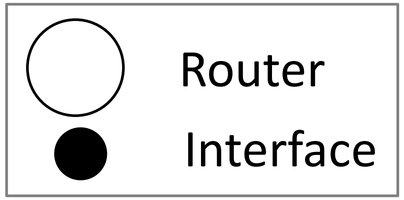
Virtual Topology

Limitations of Prior Solutions

#1 Not hide popular interfaces
→ ***Expose other targets***



Virtual Topology

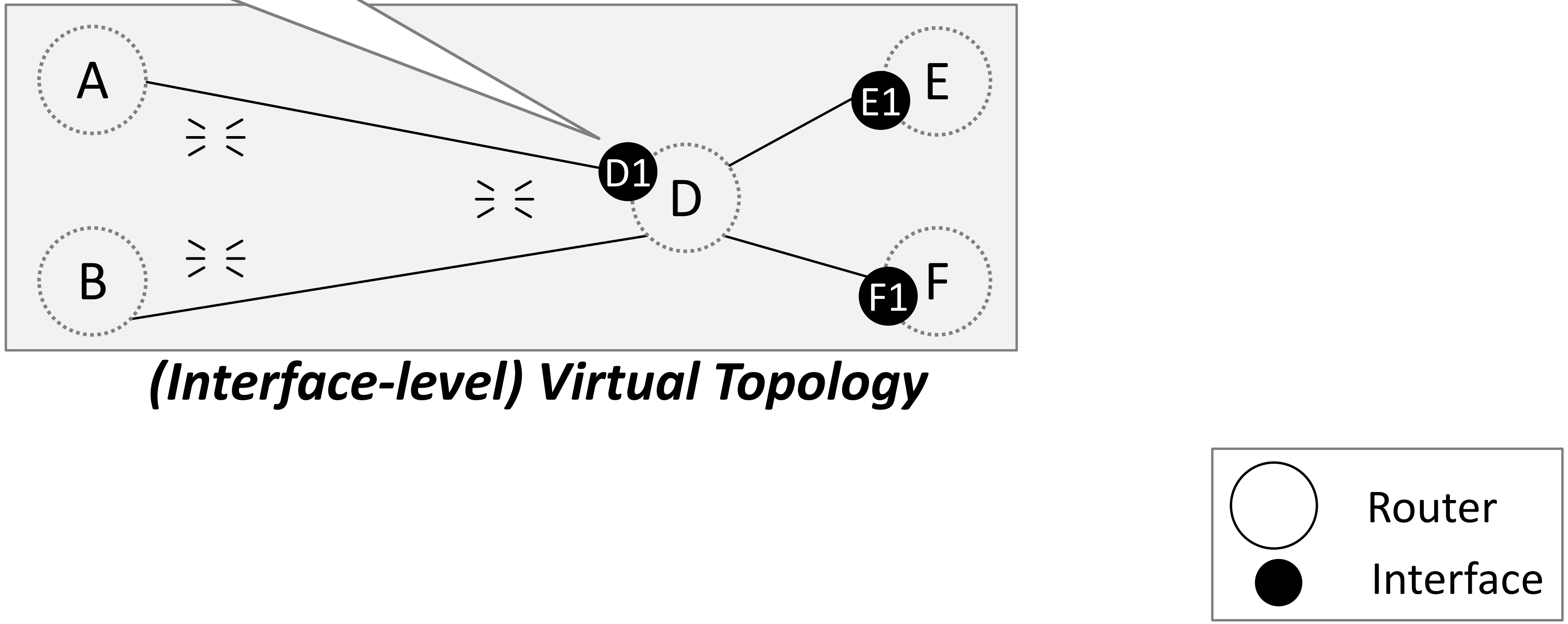


Limitations of Prior Solutions

#1 Not hide popular interfaces
→ ***Expose other targets***

$ traceroute X → Y
1 10.0.0.1 2.367ms
2 10.0.10.1 1.977ms
3 10.0.20.1 2.042ms
4 ...

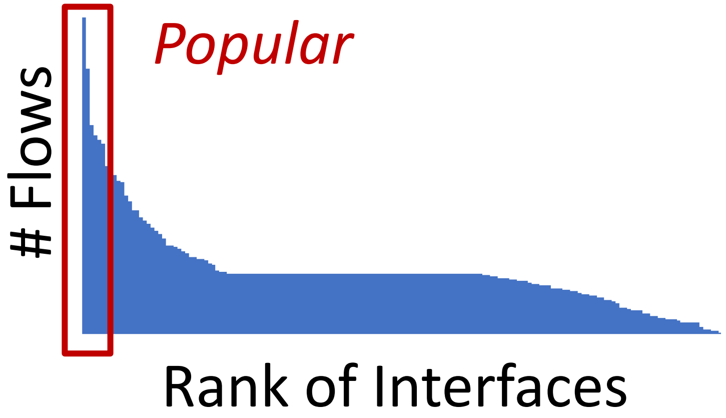
Attacker



(Interface-level) Virtual Topology

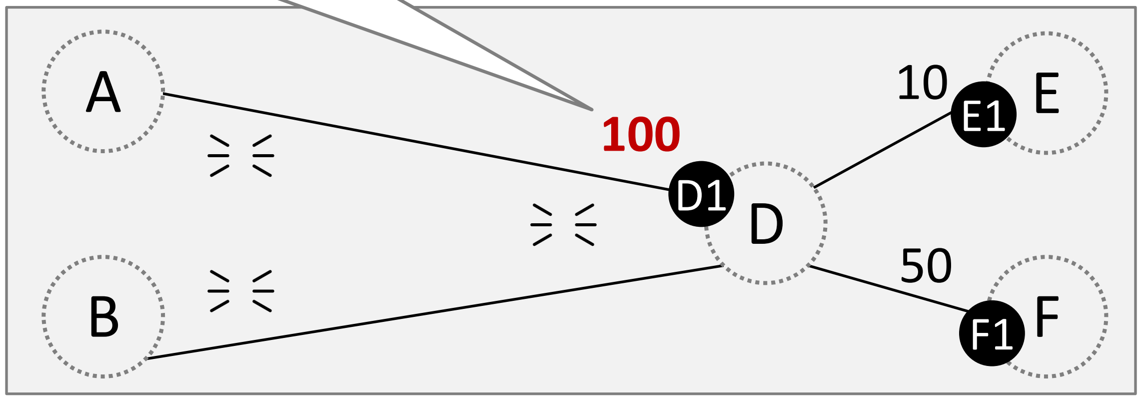
Limitations of Prior Solutions

#1 Not hide popular interfaces
→ ***Expose other targets***

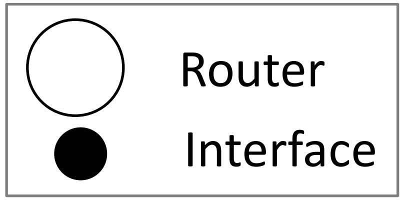


Target
the interface D1
alternatively

Attacker



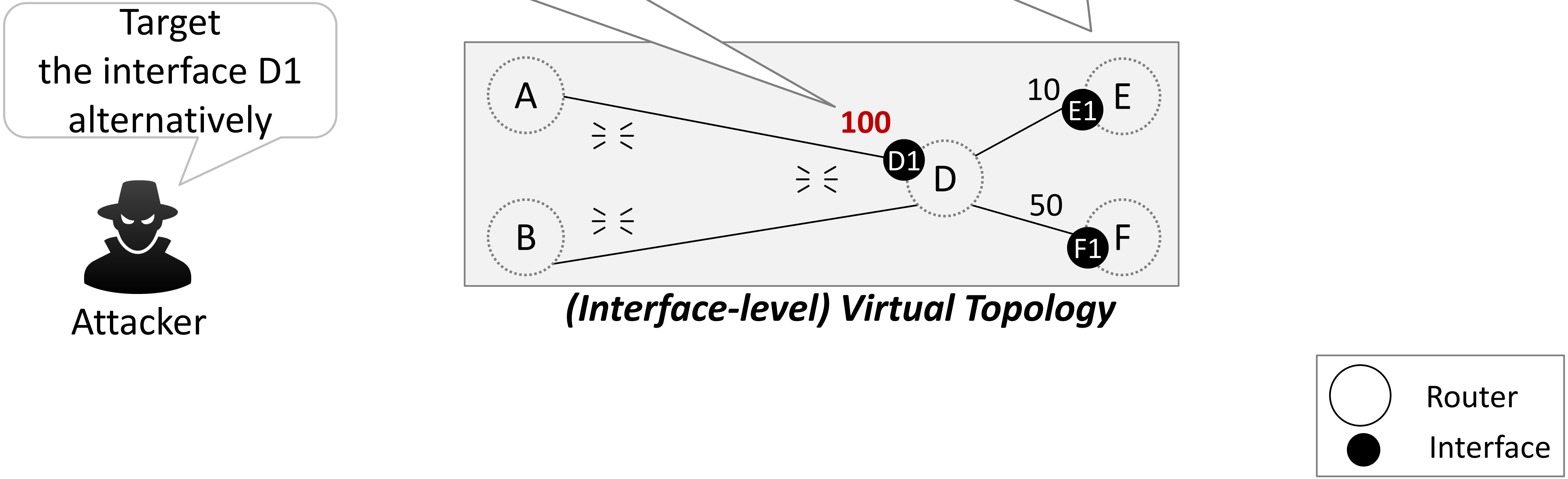
(Interface-level) Virtual Topology



Limitations of Prior Solutions

#1 Not hide popular interfaces
→ ***Expose other targets***

#2 Keep a single virtual topology only
→ ***Not secure for long-term***



(Interface-level) Virtual Topology

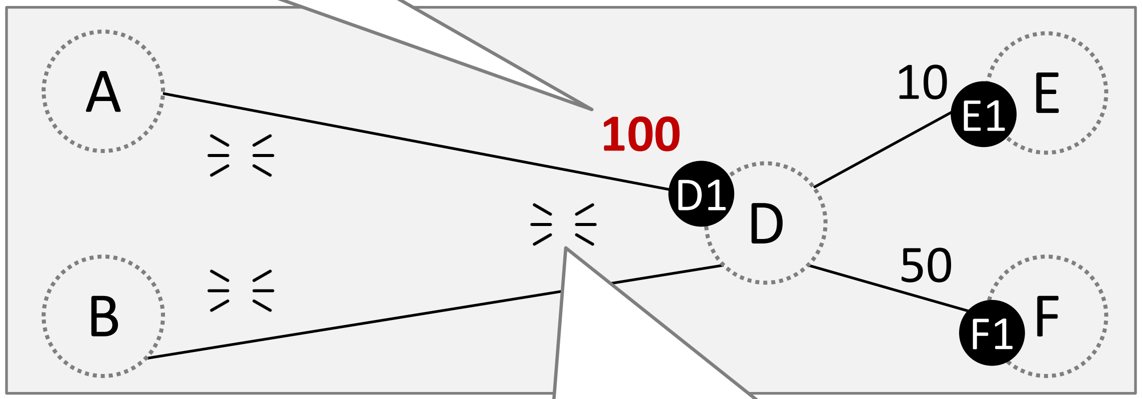
Limitations of Prior Solutions

#1 Not hide popular interfaces
→ ***Expose other targets***

#2 Keep a single virtual topology only
→ ***Not secure for long-term***

Target
the interface D1
alternatively

Attacker



Virtual Topology

#3 Hide network information partially
→ ***Reduce topology visibility***

Router
Interface

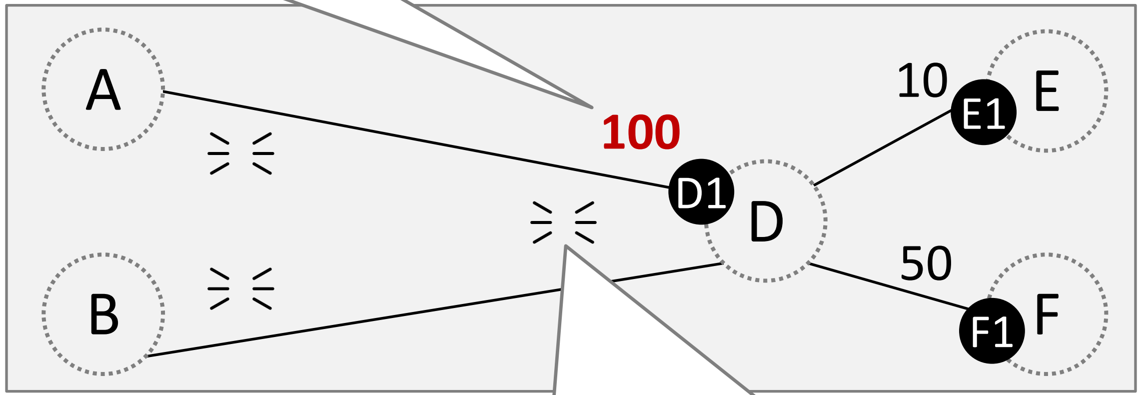
Limitations of Prior Solutions

#1 Not hide popular interfaces
→ ***Expose other targets***

#2 Keep a single virtual topology only
→ ***Not secure for long-term***

Target
the interface D1
alternatively

Attacker



Virtual Topology

Unable to find
the link C-D

Operator

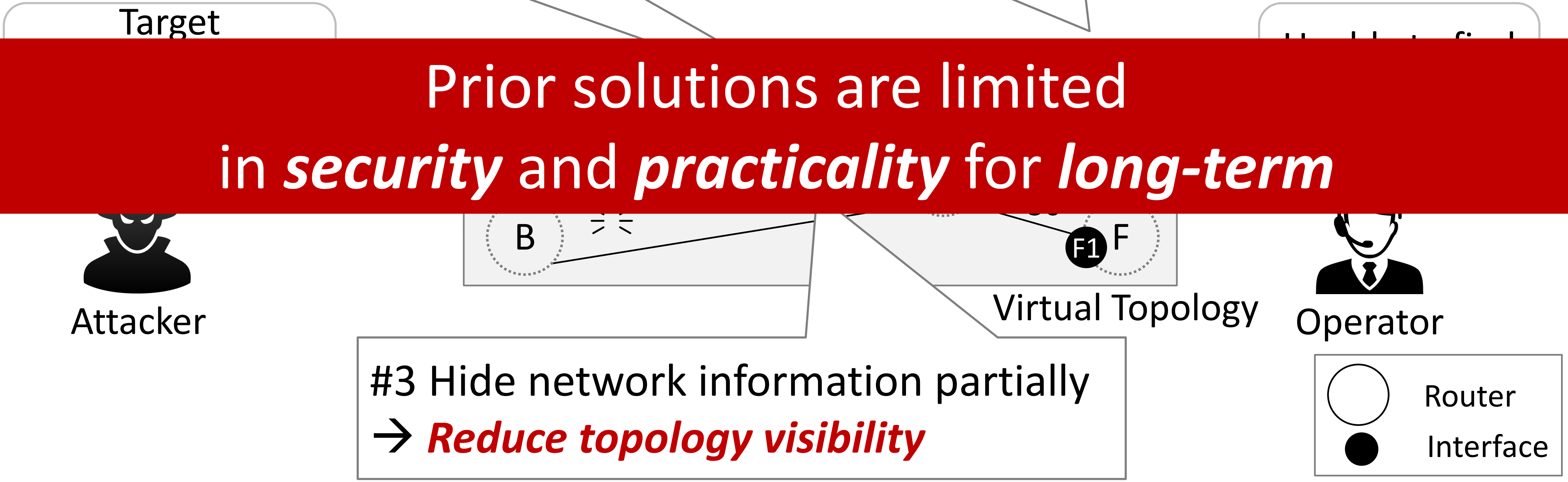
#3 Hide network information partially
→ ***Reduce topology visibility***

Router
Interface

Limitations of Prior Solutions

#1 Not hide popular interfaces
→ ***Expose other targets***

#2 Keep a single virtual topology only
→ ***Not secure for long-term***



Prior solutions are limited
in ***security*** and ***practicality*** for ***long-term***

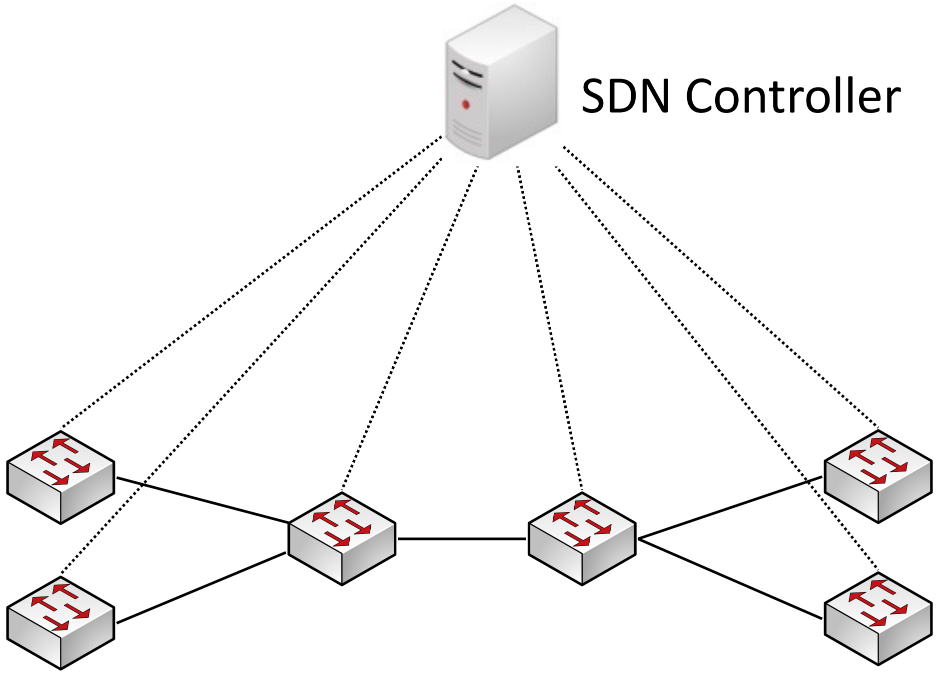
#3 Hide network information partially
→ ***Reduce topology visibility***

EqualNet:

A **Secure** and **Practical** Defense for **Long-term** Network Topology Obfuscation

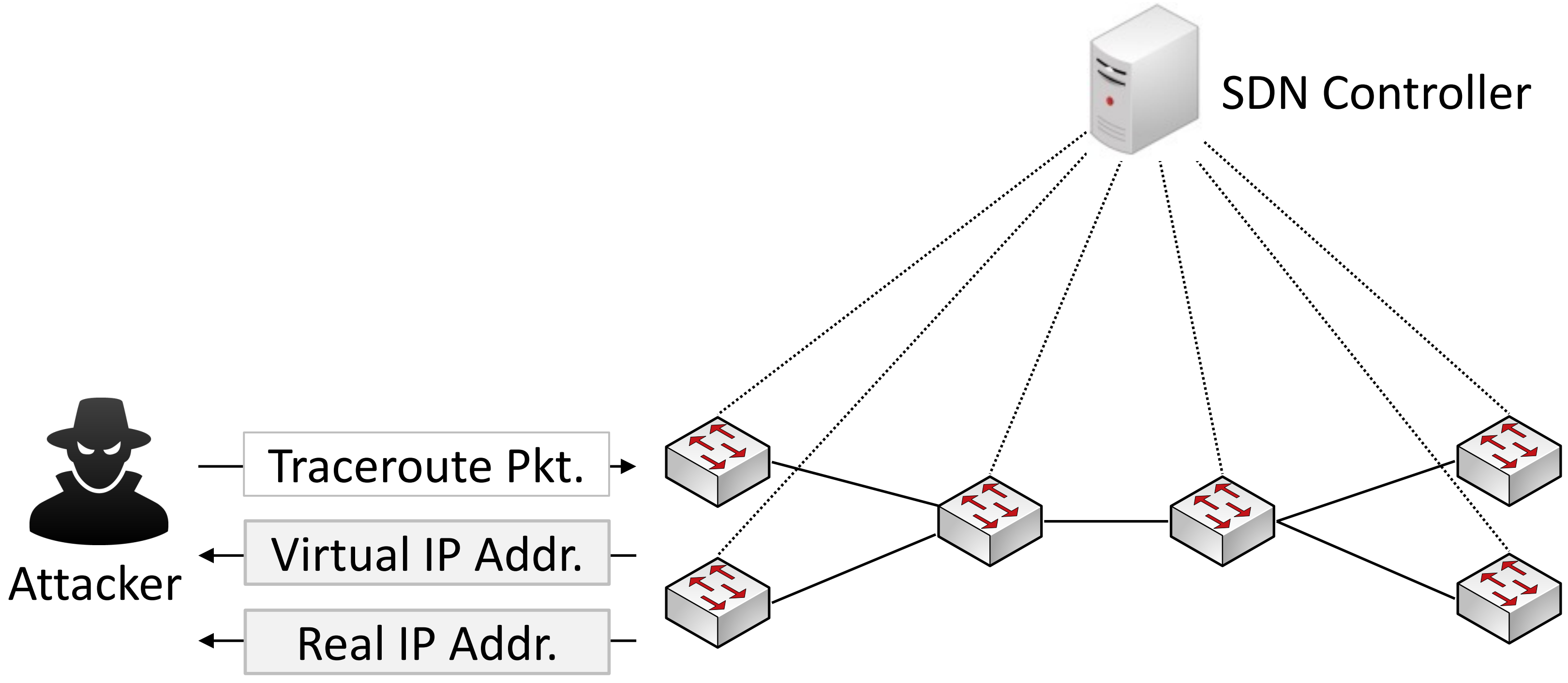
Key Idea

- Generate fake responses having virtual IP addresses
  - By utilizing SDN's centralized management



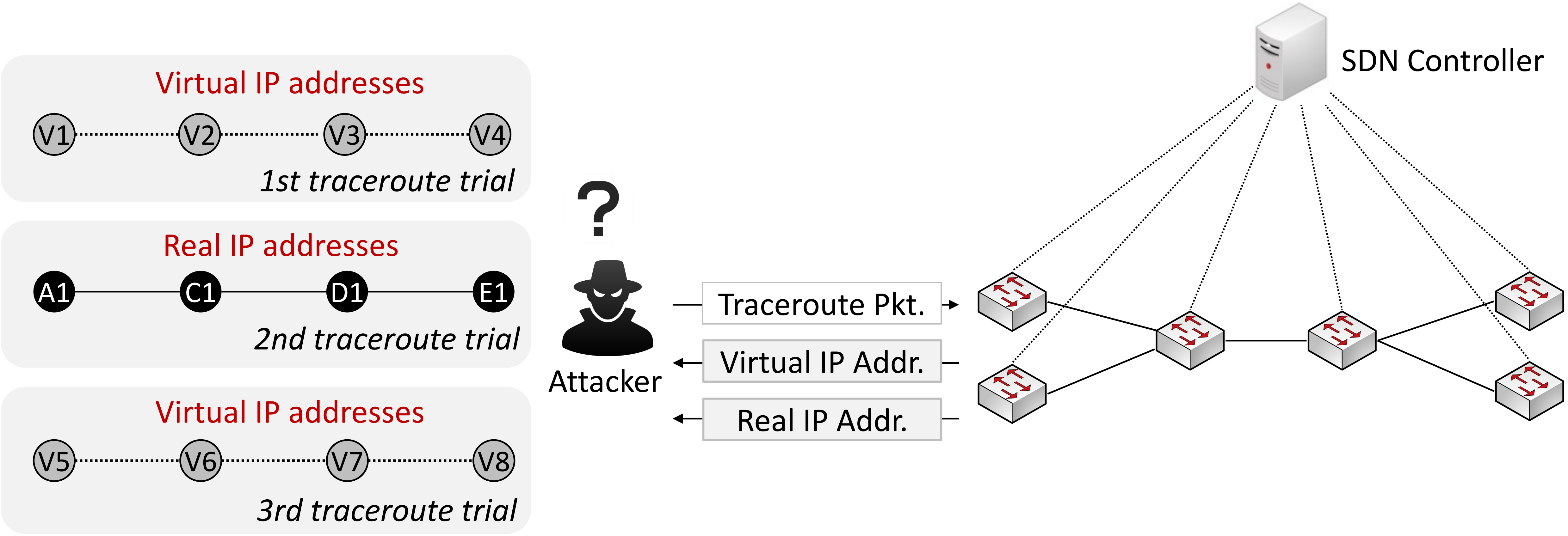
Key Idea

- Generate fake responses having virtual IP addresses
  - By utilizing SDN's centralized management

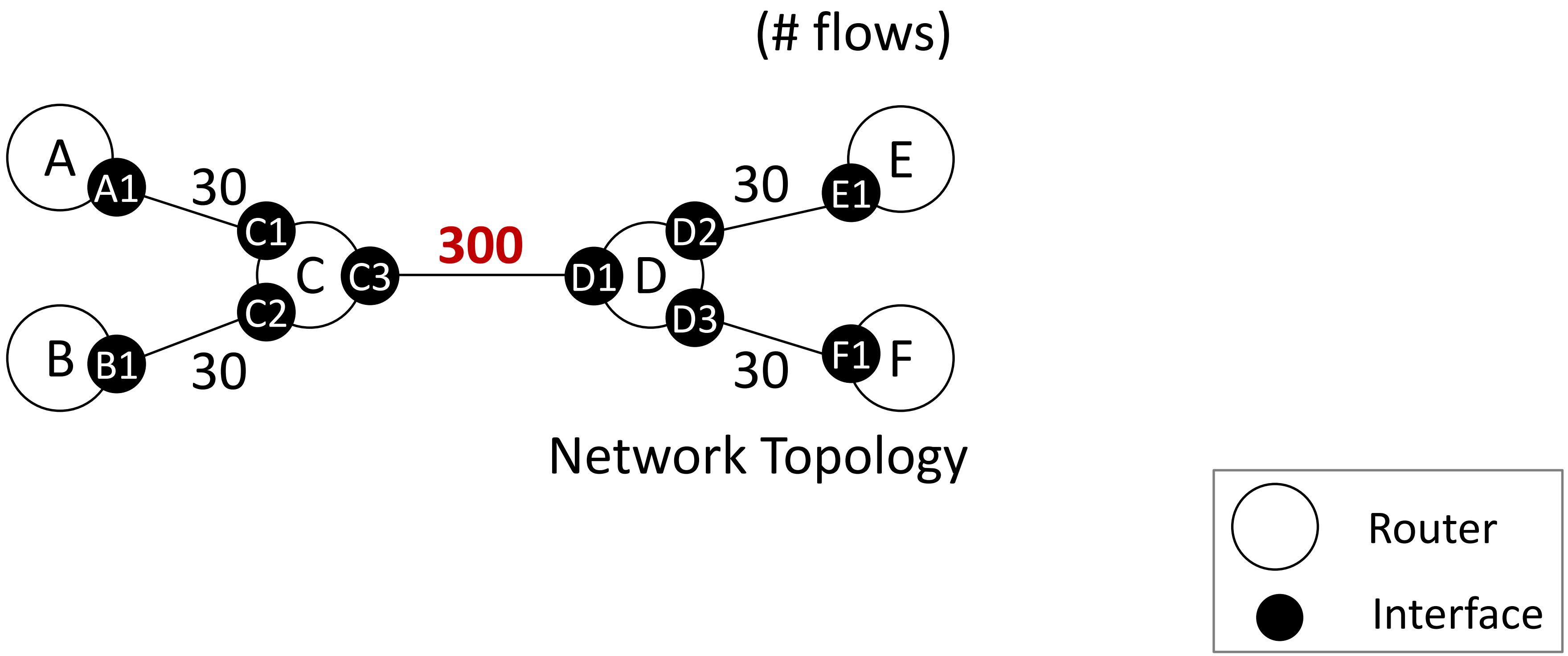


Key Idea

- Generate fake responses having virtual IP addresses
  - By utilizing SDN's centralized management



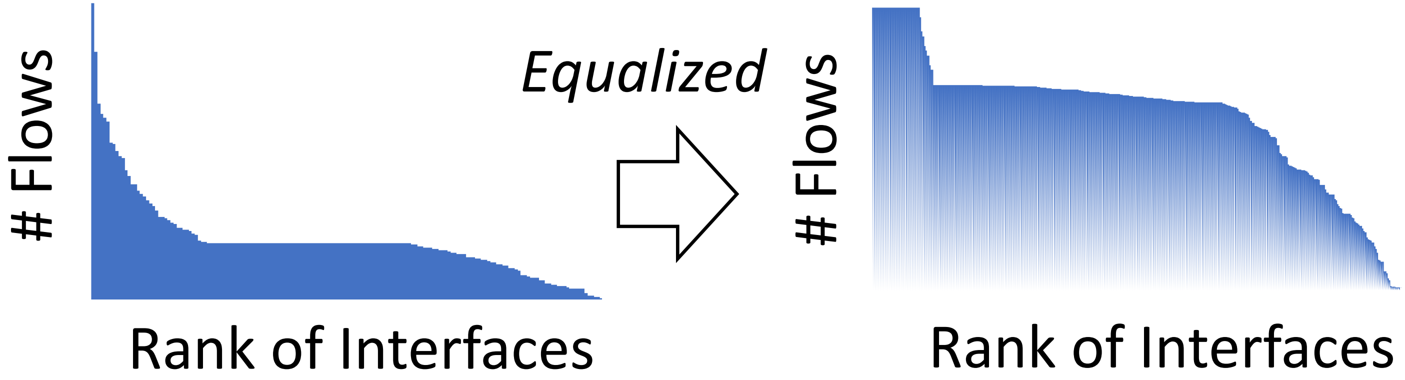
Our Approach

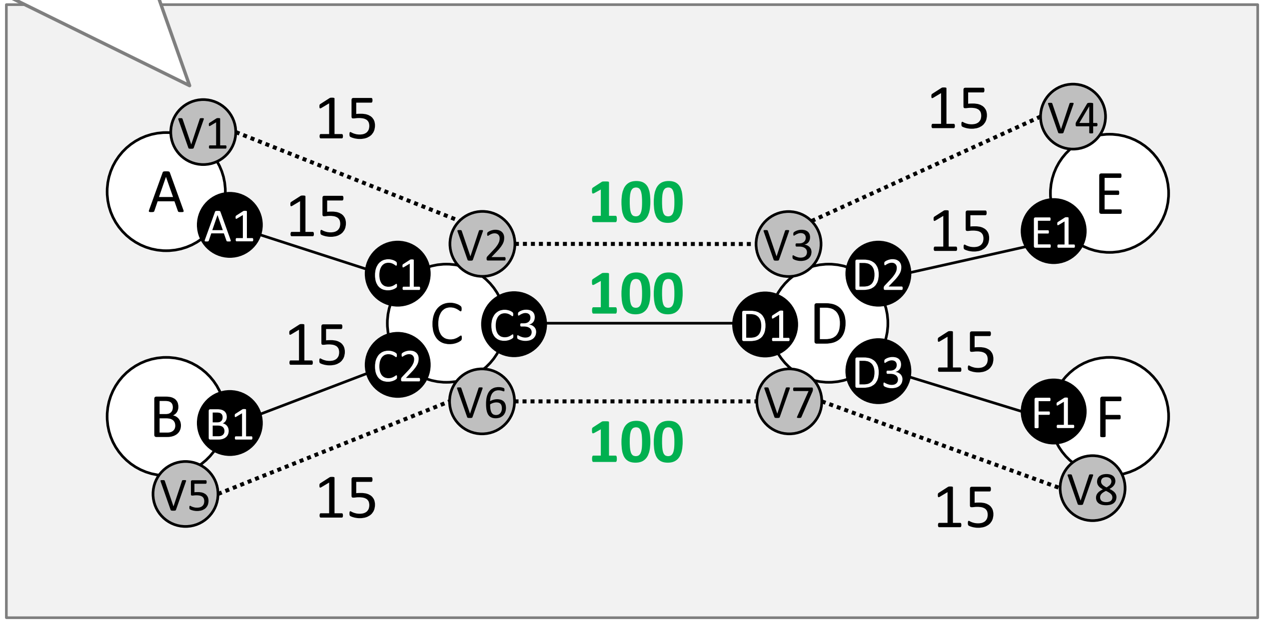


Our Approach

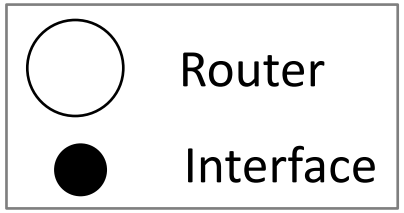
#1 Not hide popular interfaces

→ *Expose interfaces equally likely*



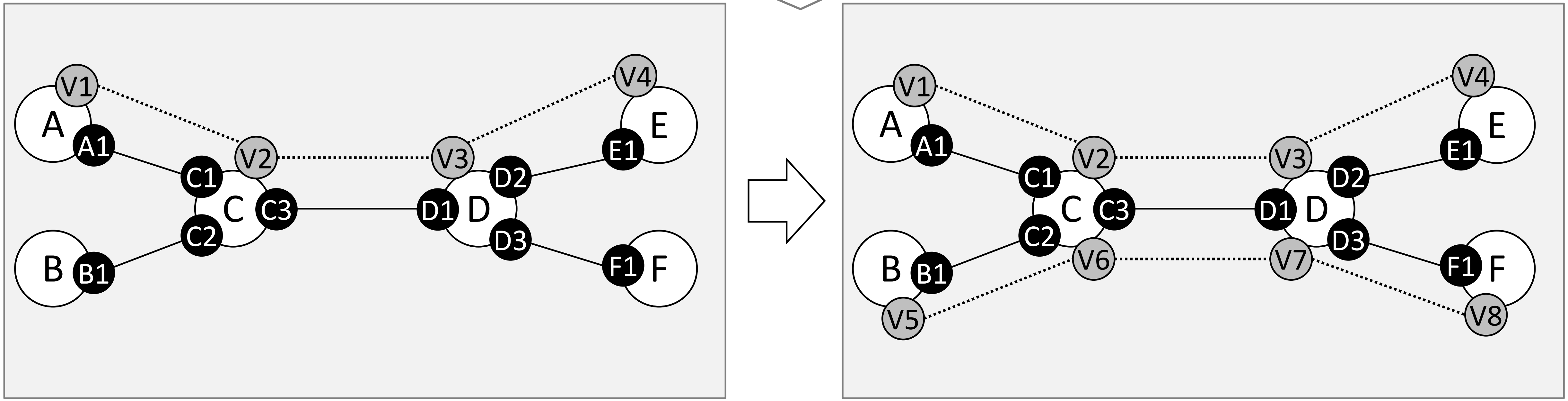


Virtual Topology



Our Approach

#2 Keep a single virtual topology only
→ ***Adjust the virtual topology on-the-fly***

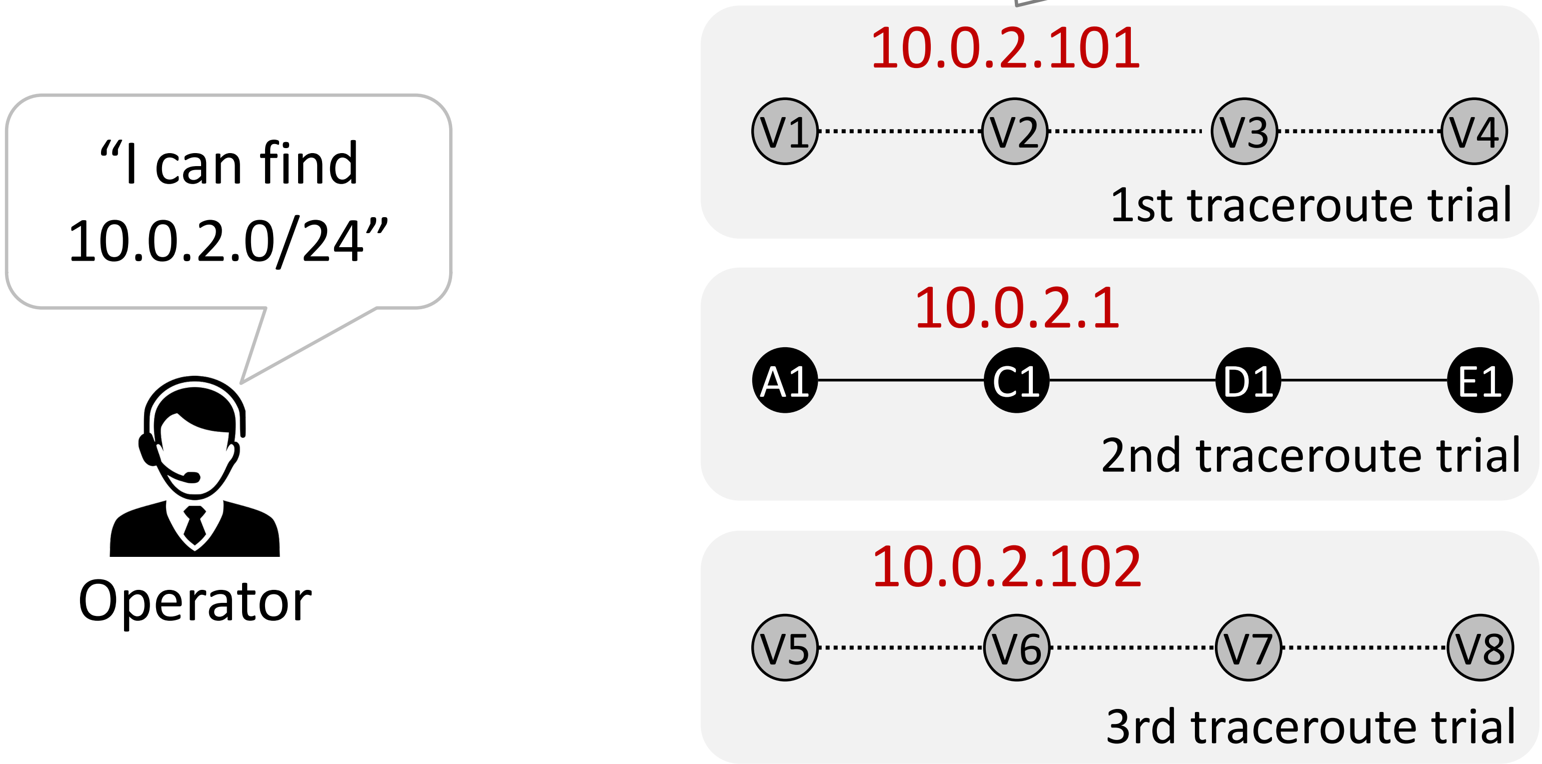


Virtual Topology A

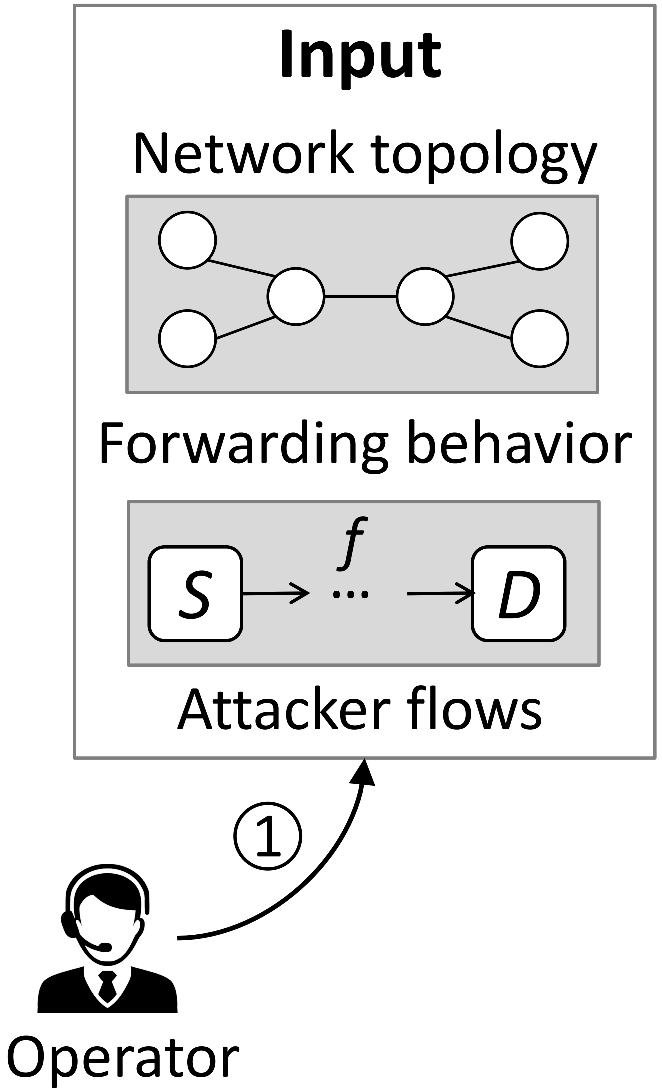
Virtual Topology B

Our Approach

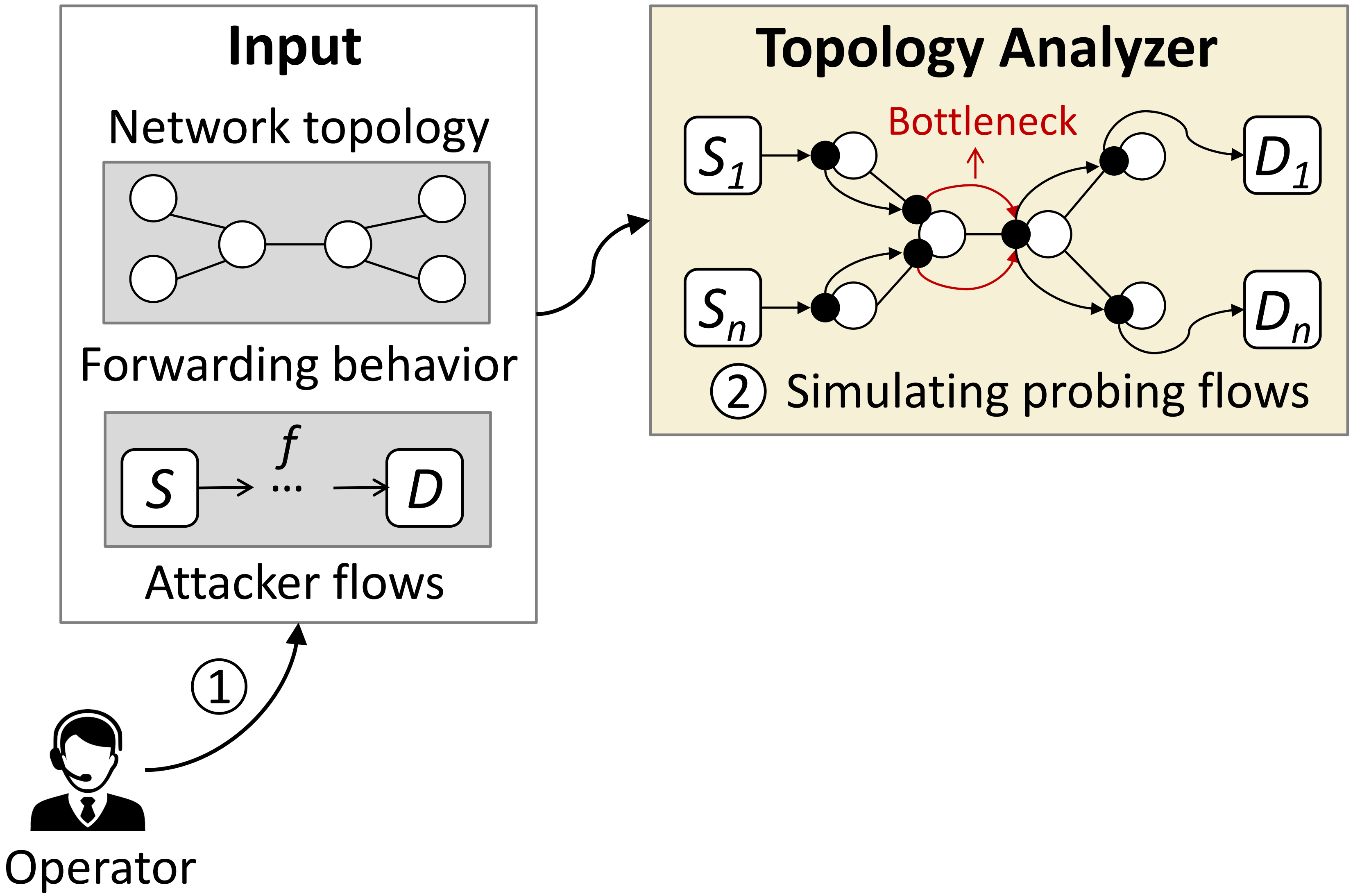
#3 Hide network information partially
→ ***Choose IP addresses in the same subnet***



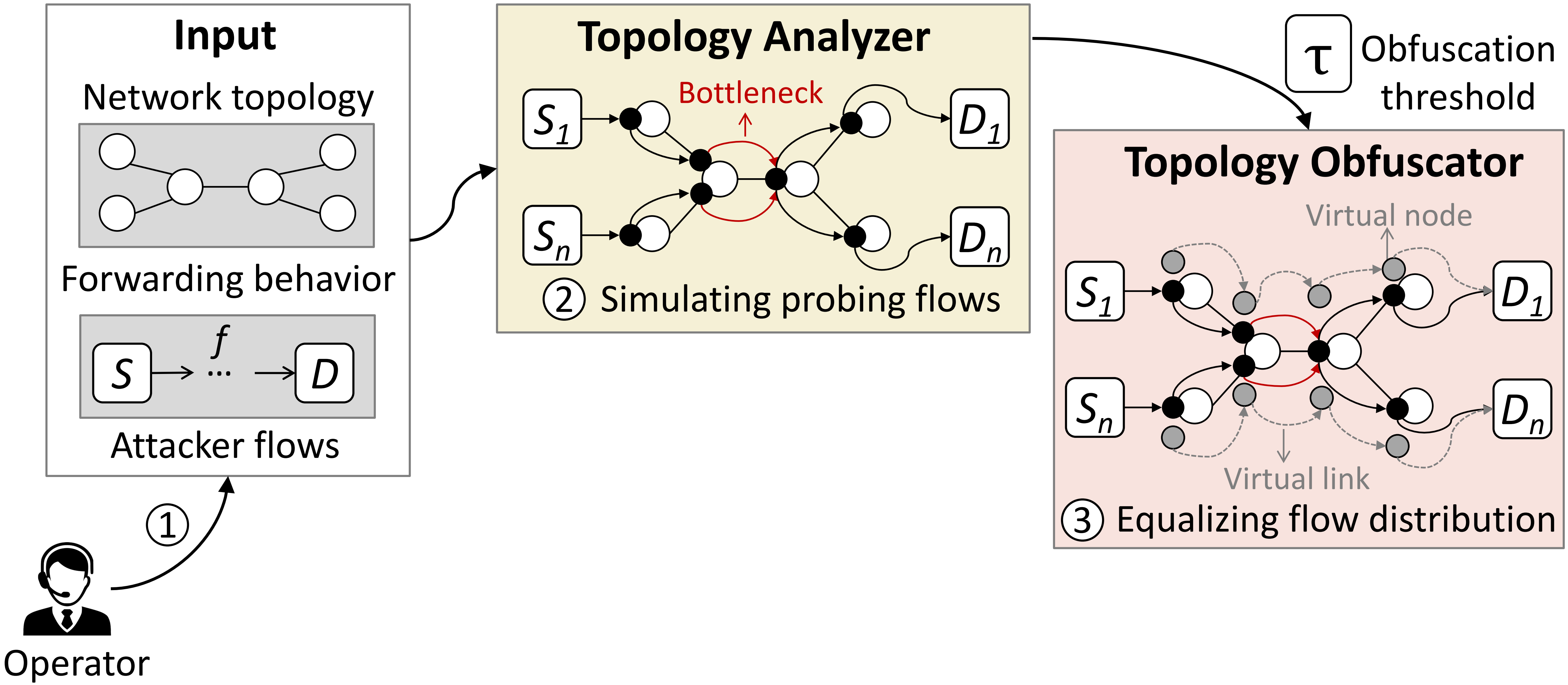
EqualNet Overview



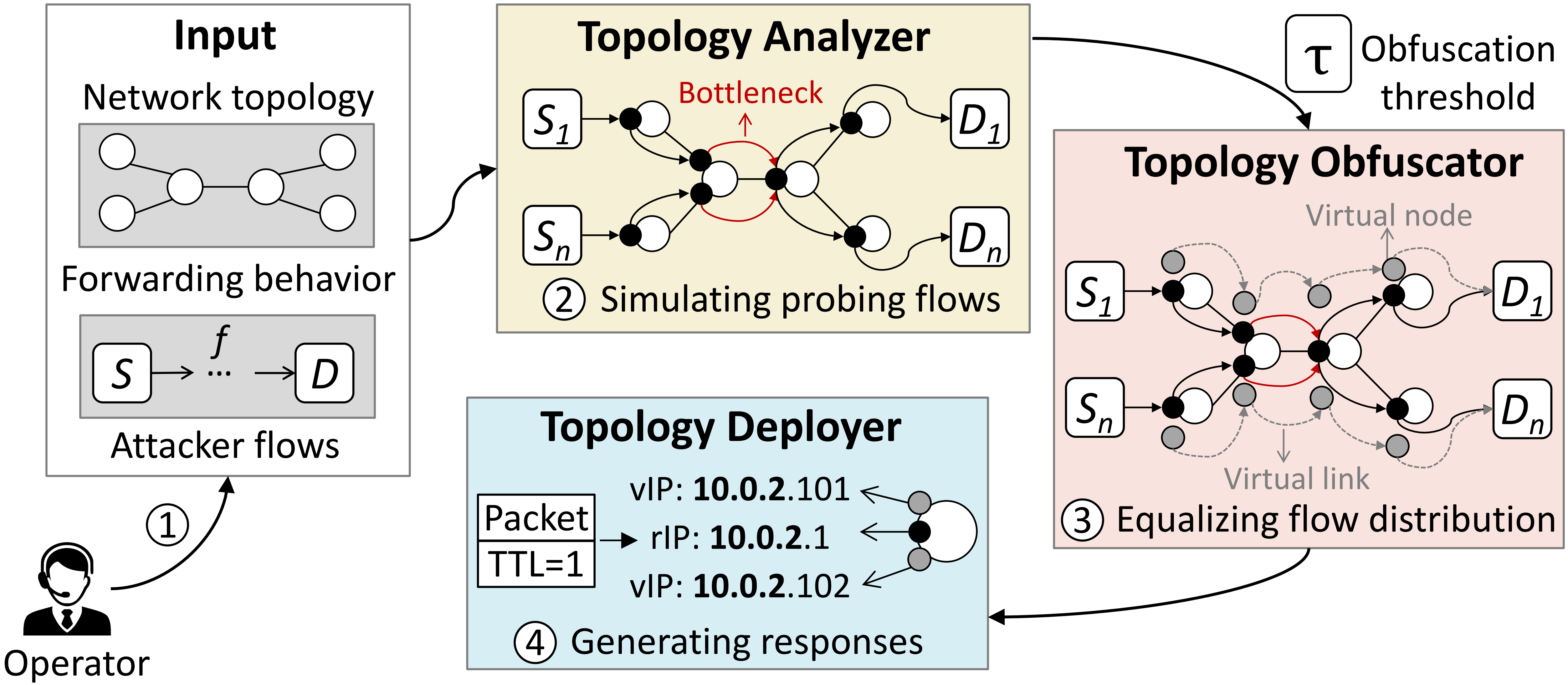
EqualNet Overview



EqualNet Overview

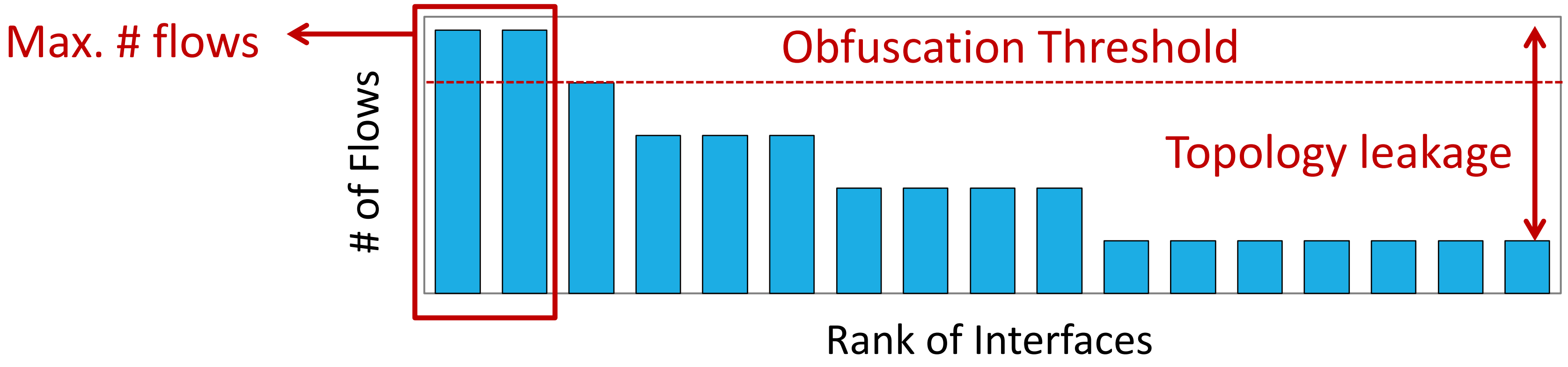


EqualNet Overview



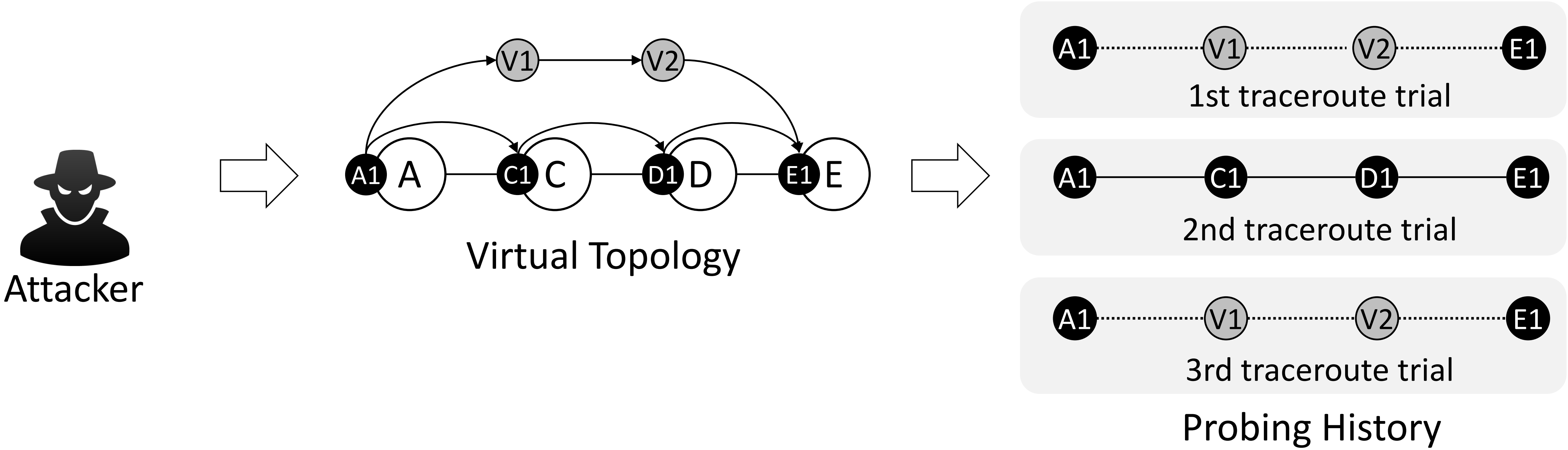
Analyzing Network Topology

- ***Topology leakage***
  - Diff. between the max. and min. # flows per interface
    - The lower, the more indistinguishable
- ***Obfuscation threshold***
  - Operator’s desired topology leakage
    - E.g., 80% of the topology leakage, 500 flows per interface



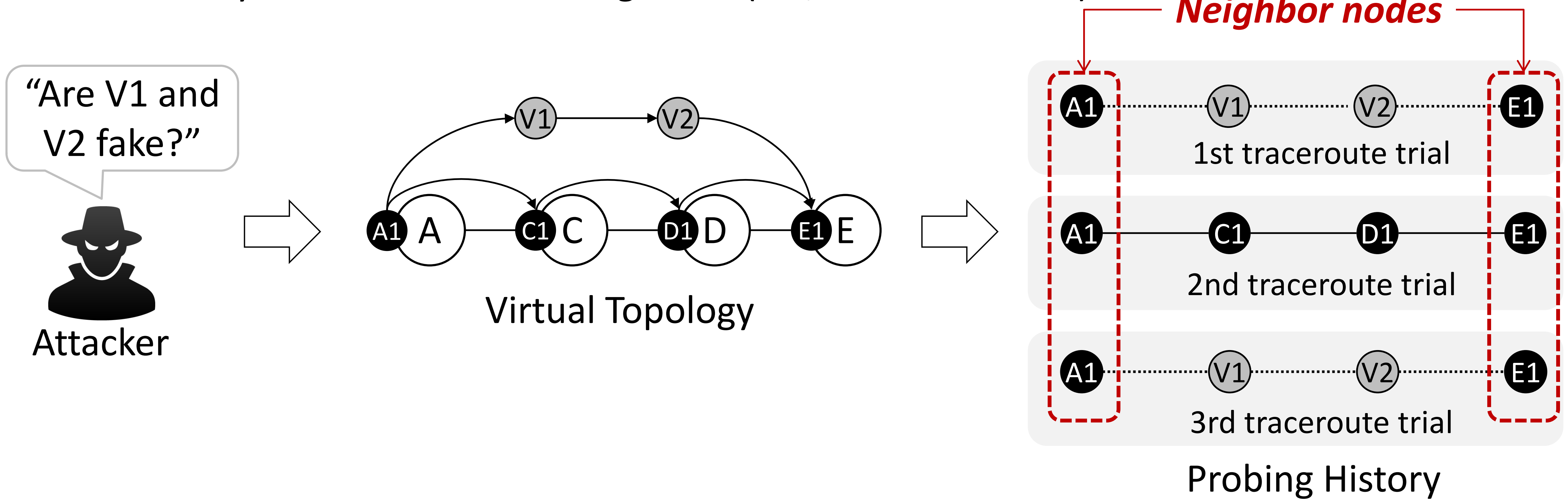
Obfuscating Network Topology

- Challenge
  – Attackers can compare differences of probing history



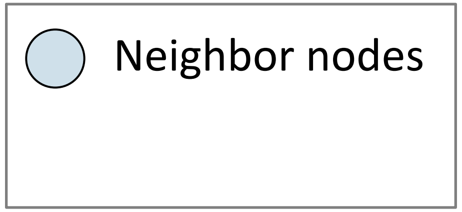
Obfuscating Network Topology

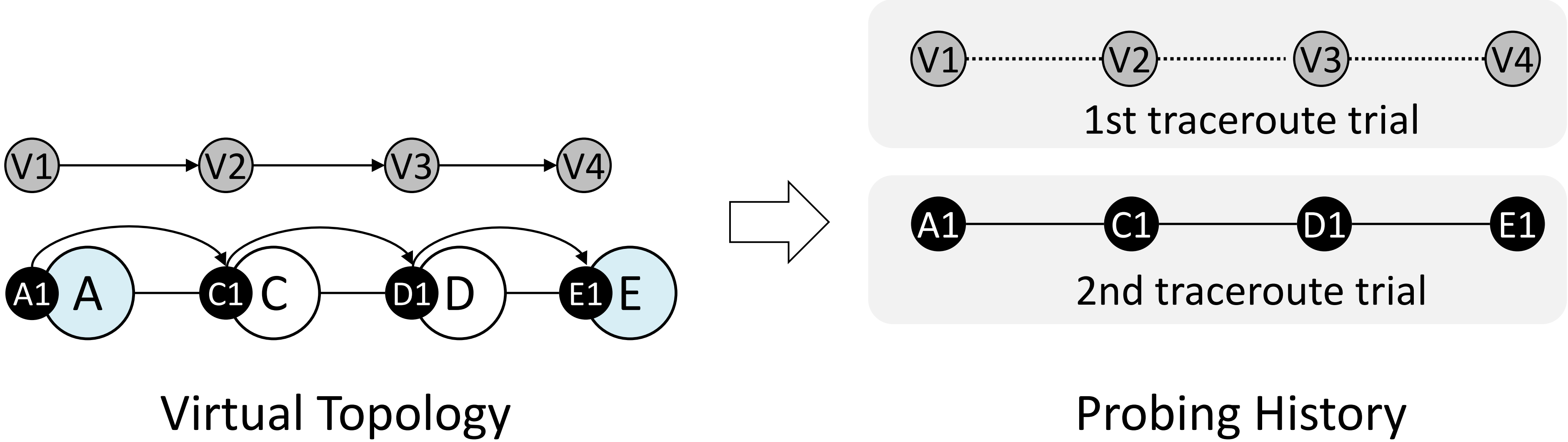
- Challenge
  - Attackers can compare differences of probing history
  - If they observe the same ***neighbors*** (i.e., alias resolution)



Obfuscating Network Topology

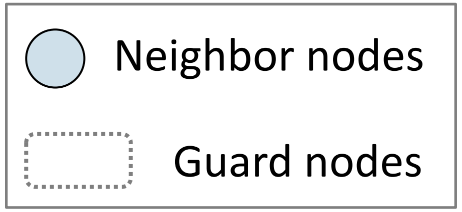
- Solution
  - Add virtual nodes to neighbors to form separate probing paths

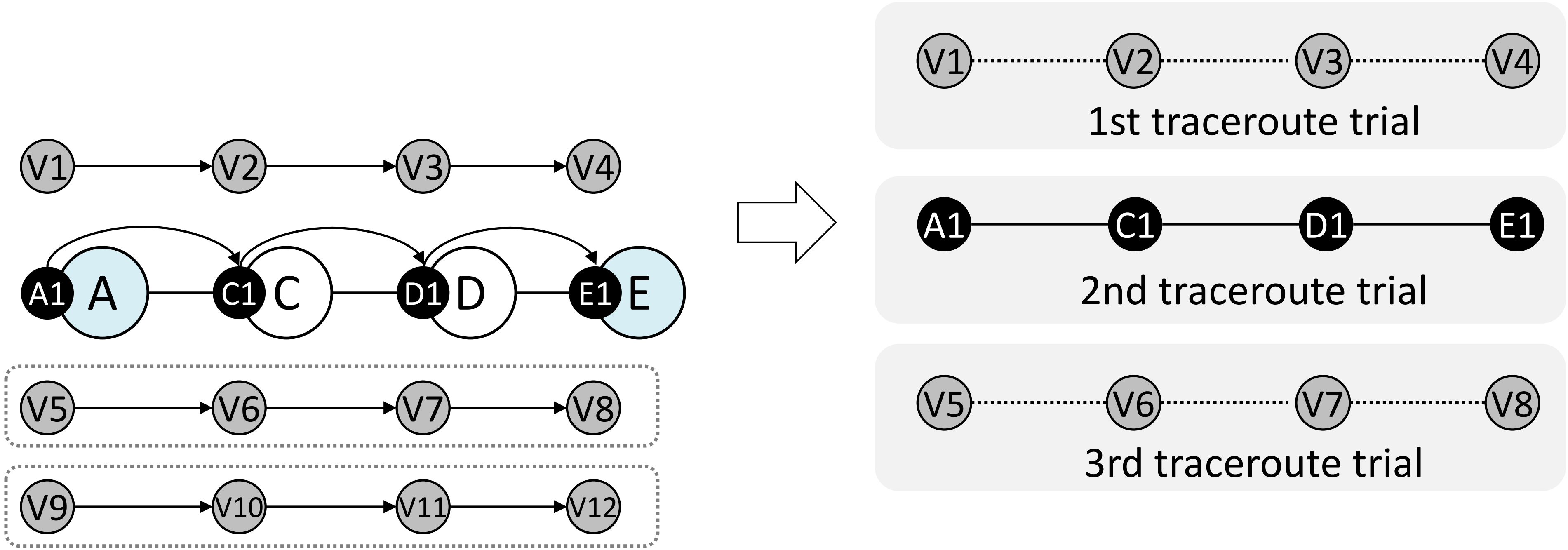




Obfuscating Network Topology

- Solution
  - Add virtual nodes to neighbors to form separate probing paths
  - Keep the minimum number of virtual nodes (i.e., guard nodes)



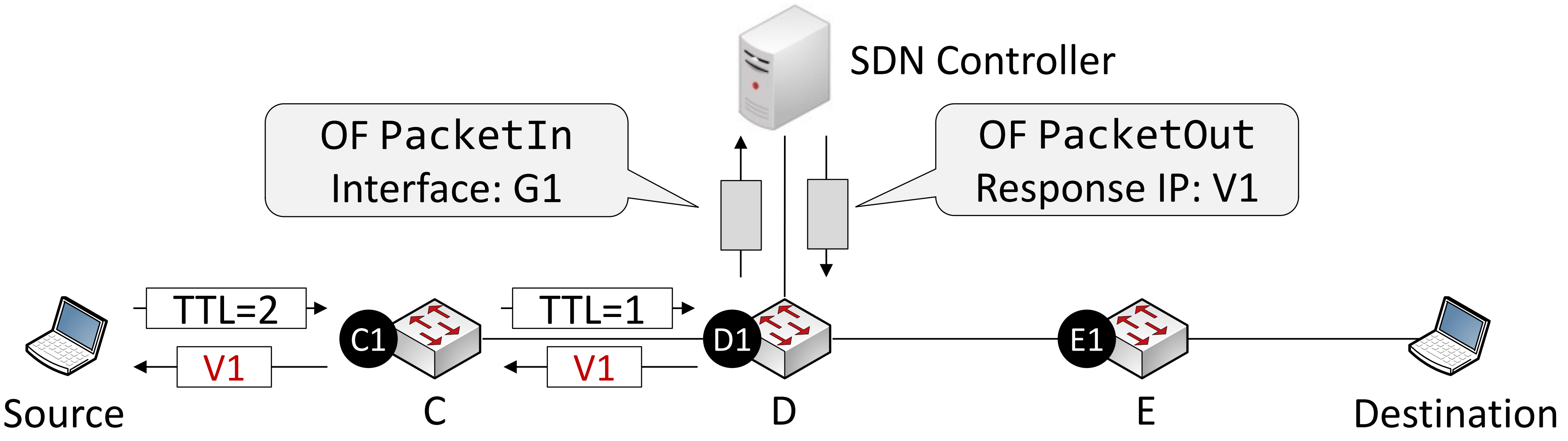


Virtual Topology

Probing History

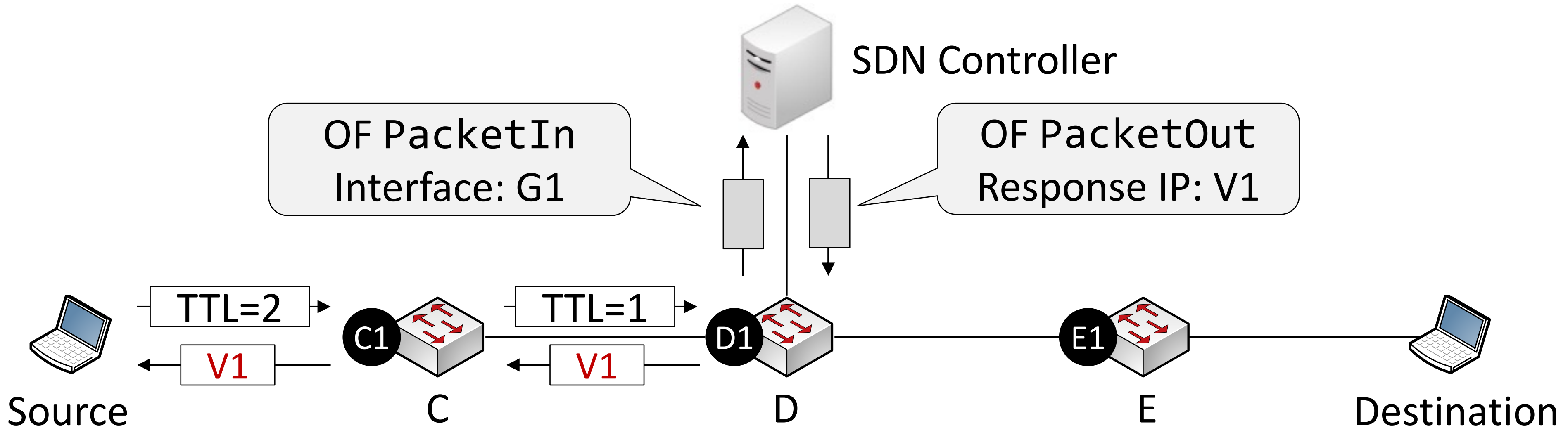
Deploying Virtual Topology

- Utilizing OpenFlow
  - To detect probing packets and generate fake responses



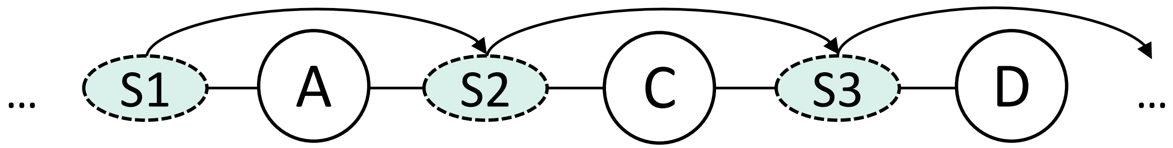
Deploying Virtual Topology

- Utilizing OpenFlow
  - To detect probing packets and generate fake responses
- Producing indistinguishable responses
  - Choose IP addresses randomly within the same subnet
  - Randomize packet headers (e.g., IP ID) to prevent inference attacks



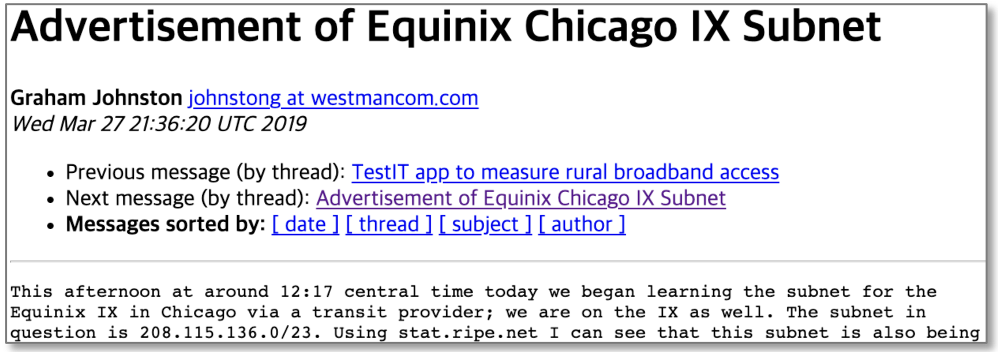
Why subnets?

- Routing policies are designed based on subnets



<Operator’s topology view>

- Operators mention subnets to inform failure positions
  - *E.g.,* Is *S3* (e.g., 10.0.1/24) reachable from *S2* (e.g., 10.0.2/24)?



Advertisement of Equinix Chicago IX Subnet

Graham Johnston johnstong at westmancom.com
Wed Mar 27 21:36:20 UTC 2019

- Previous message (by thread): TestIT app to measure rural broadband access
- Next message (by thread): Advertisement of Equinix Chicago IX Subnet
- **Messages sorted by:** [date] [thread] [subject] [author]

This afternoon at around 12:17 central time today we began learning the subnet for the Equinix IX in Chicago via a transit provider; we are on the IX as well. The subnet in question is 208.115.136.0/23. Using stat.ripe.net I can see that this subnet is also being

Geolocation: IPv4 Subnet blocked by HULU, and others

Michael Crapse michael at wi-fiber.io
Wed Dec 6 21:38:20 UTC 2017

- Previous message (by thread): Sys admin has gotten of topic...
- Next message (by thread): Geolocation: IPv4 Subnet blocked by HULU, and others
- **Messages sorted by:** [date] [thread] [subject] [author]

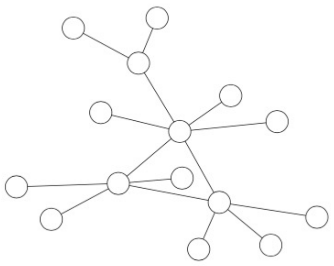
I am a local WISP. And my customers have trouble reaching Hulu, Disney now,
and previously netflix and amazon prime(both resolved).
I have emailed, mailed, and called both HULU and Disney now to get my

<Mail threads from NANOG>

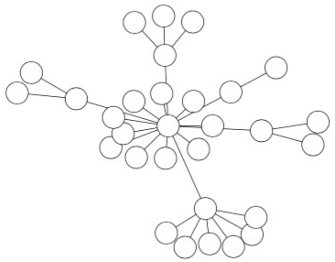
Evaluation

- Experiments
  1. Leakage reduction vs. virtual nodes
  2. Resistance to topology inference
  3. Protection against alias resolution
  4. Topology similarity and utility
  5. Fingerprinting via RTT measurement
  6. Partial deployment

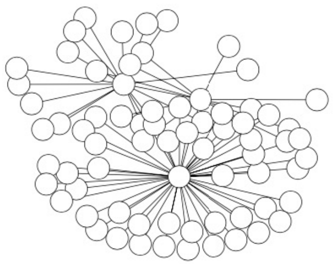
- Dataset
  - Three router-level topology data
    - From CAIDA ITDK (Internet Topology Data Kit)



AS 13576 (small)



AS 35132 (medium)



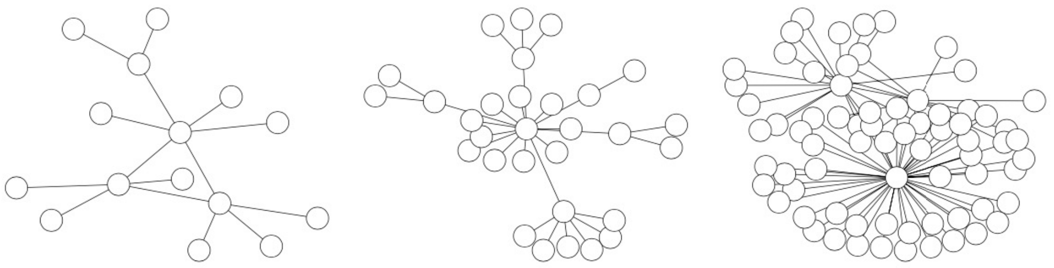
AS 35575 (large)

Evaluation

- Experiments
  1. Leakage reduction vs. virtual nodes
  2. Resistance to topology inference
  3. Protection against alias resolution
  4. Topology similarity and utility
  5. Fingerprinting via RTT measurement
  6. Partial deployment

→ *Please refer to our paper* (items 4–6)

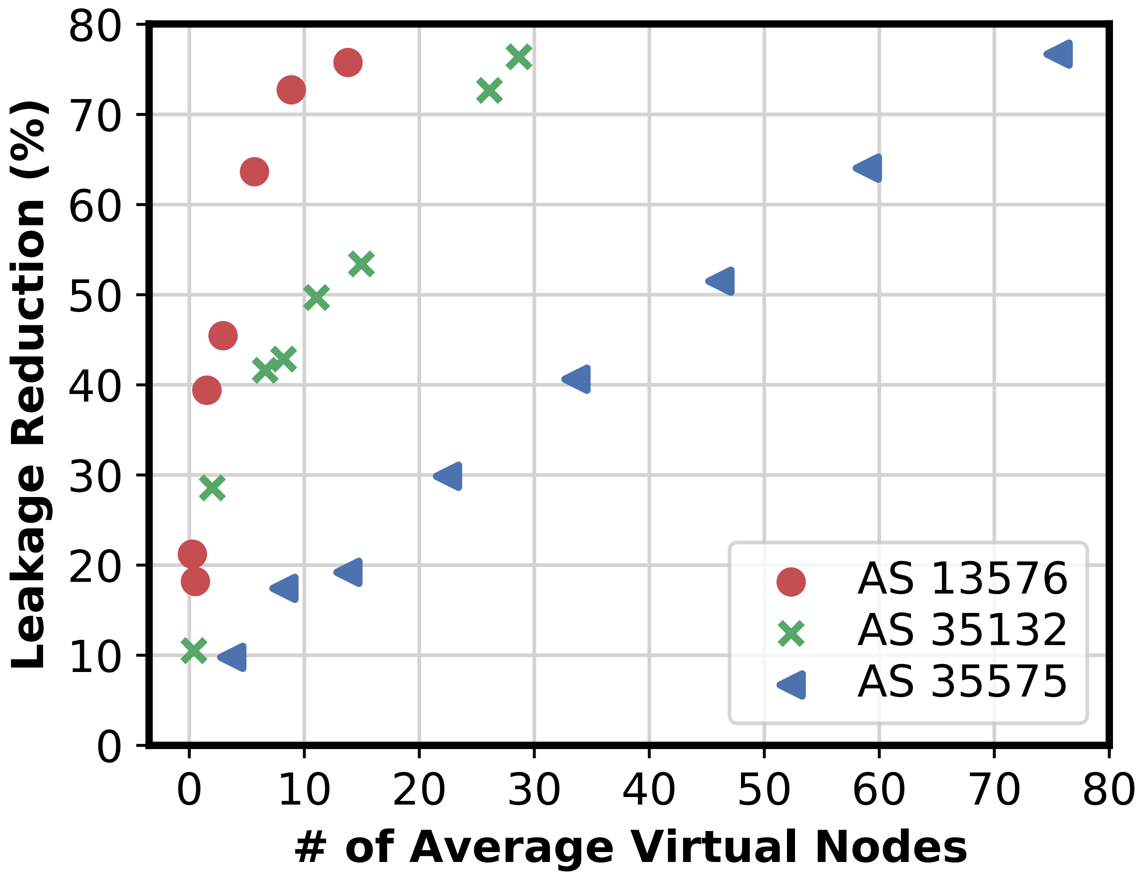
- Dataset
  - Three router-level topology data
    - From CAIDA ITDK (Internet Topology Data Kit)



AS 13576 (small)   AS 35132 (medium)   AS 35575 (large)

Leakage Reduction vs. Virtual Nodes

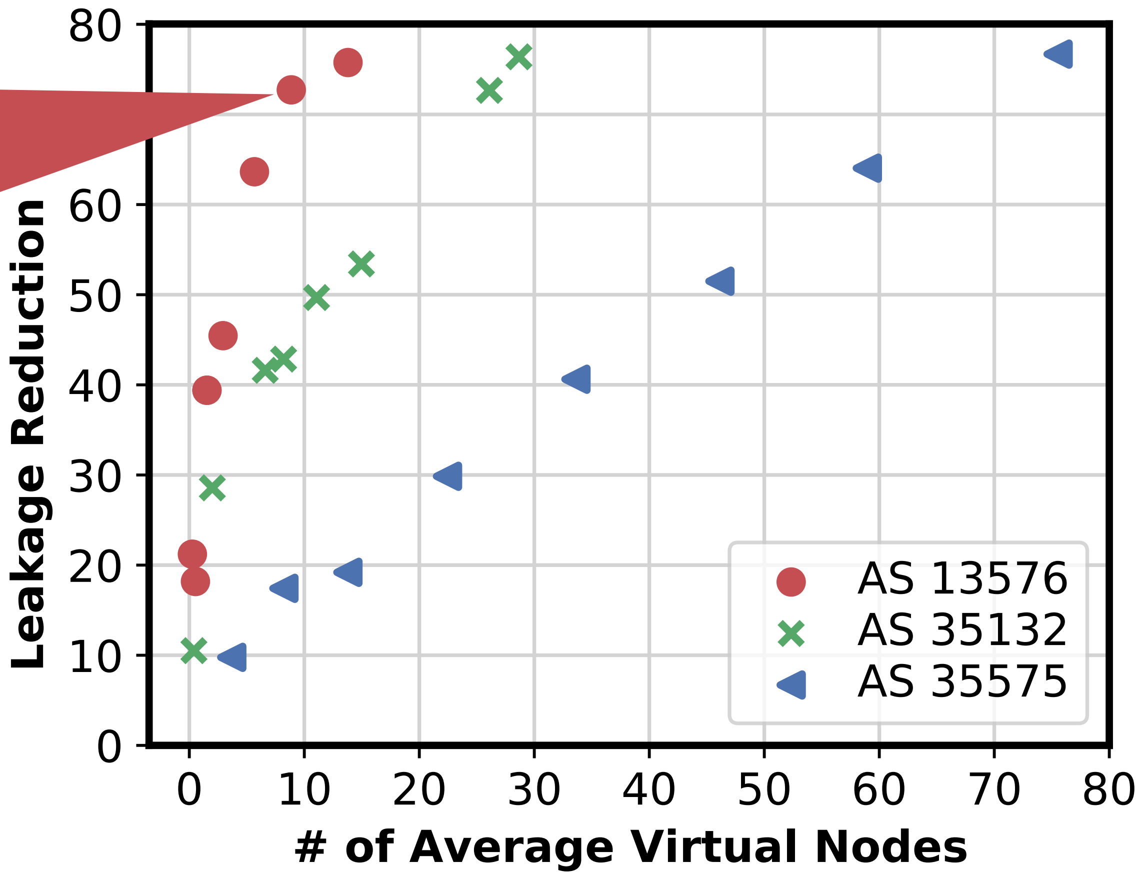
- Measured leakage reduction to evaluate equalization effectiveness



Leakage Reduction vs. Virtual Nodes

- Measured leakage reduction to evaluate equalization effectiveness

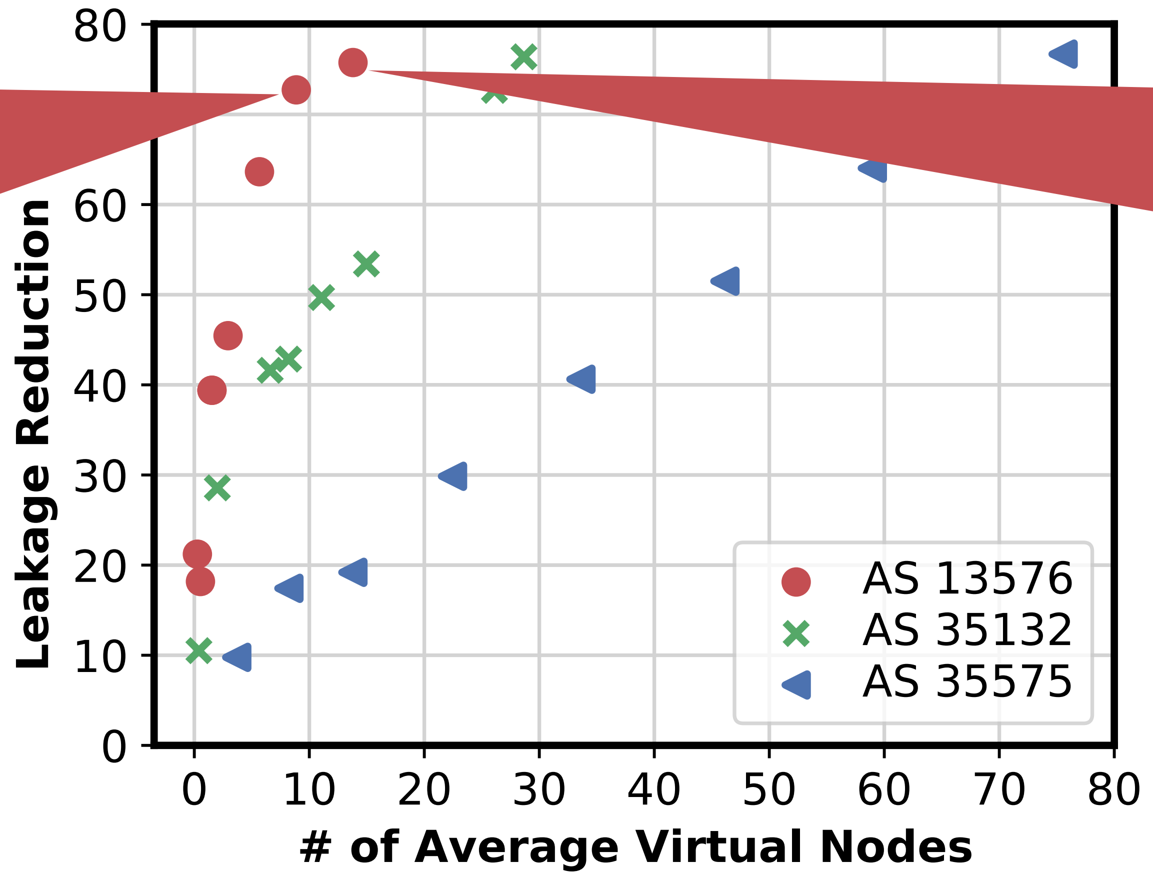
71% leakage reduction with ***9*** virtual nodes per router



Leakage Reduction vs. Virtual Nodes

- Measured leakage reduction to evaluate equalization effectiveness

71% leakage reduction with ***9*** virtual nodes per router

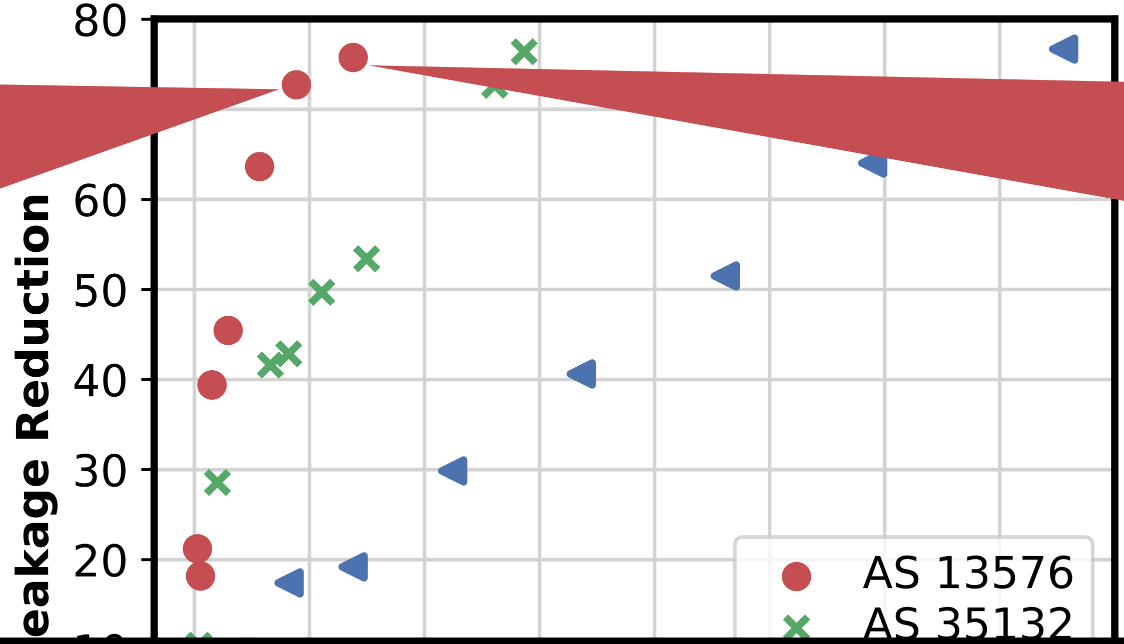


76% leakage reduction with ***14*** virtual nodes per router

Leakage Reduction vs. Virtual Nodes

- Measured leakage reduction to evaluate equalization effectiveness

71% leakage reduction with ***9*** virtual nodes per router

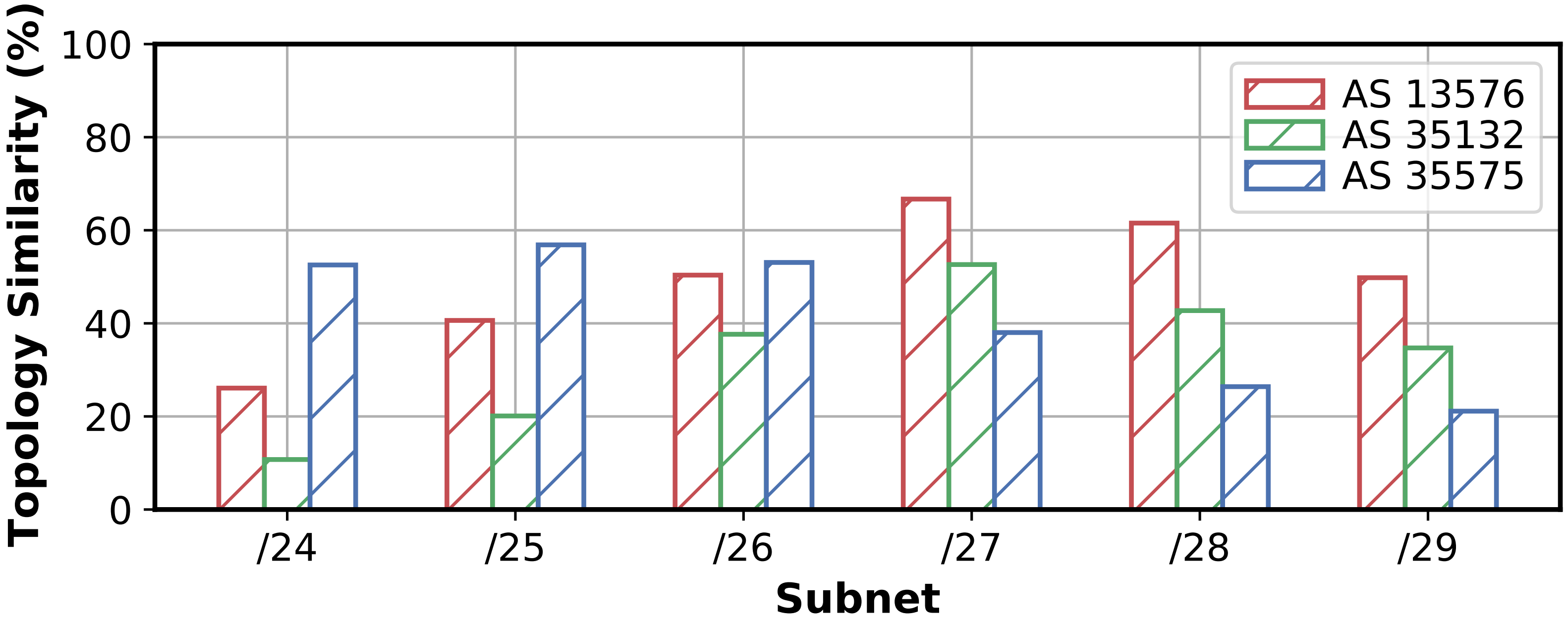


76% leakage reduction with ***14*** virtual nodes per router

There is a need to consider a trade-off between the topology leakage and virtual nodes

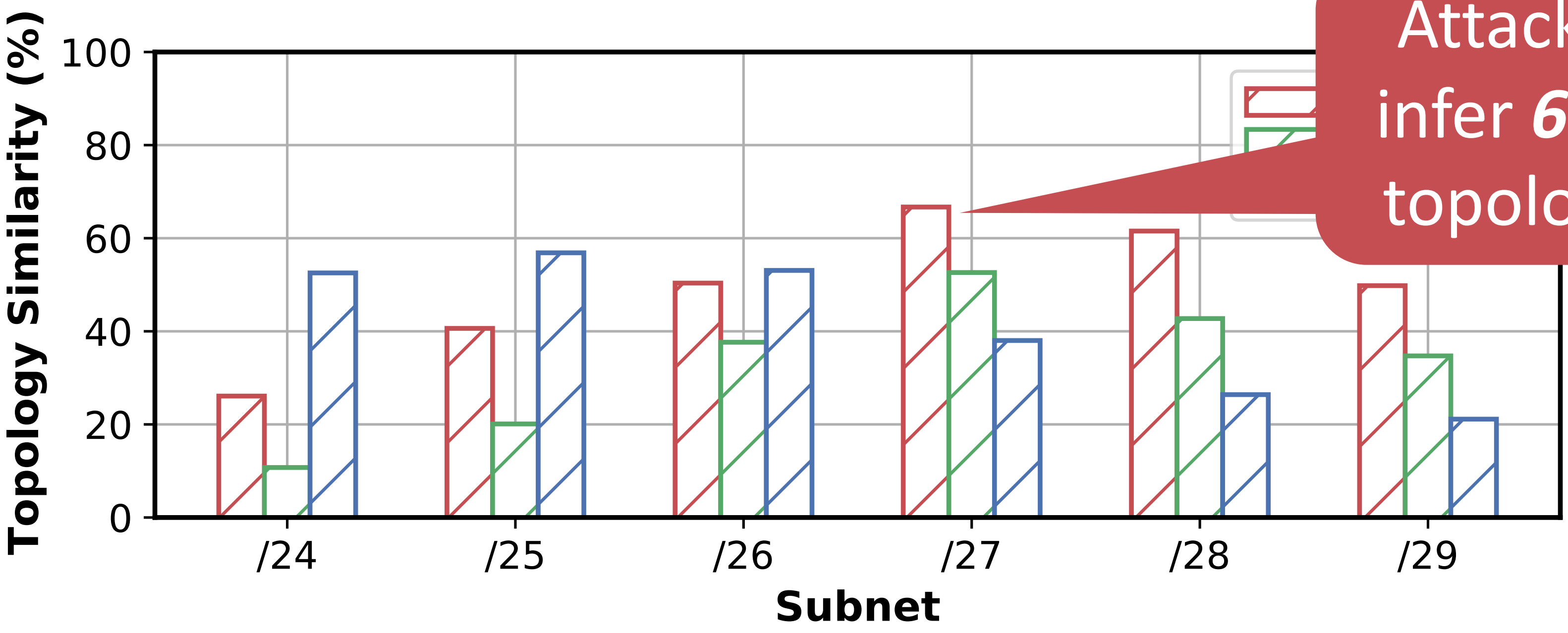
Resistance to Topology Inference

- Can attackers infer a real network topology?
  - Assuming that all links use the same mask (best for attackers)
  - By trying all possible masks (e.g., from /24 to /29)
- Compared how the inferred topology is similar with the real



Resistance to Topology Inference

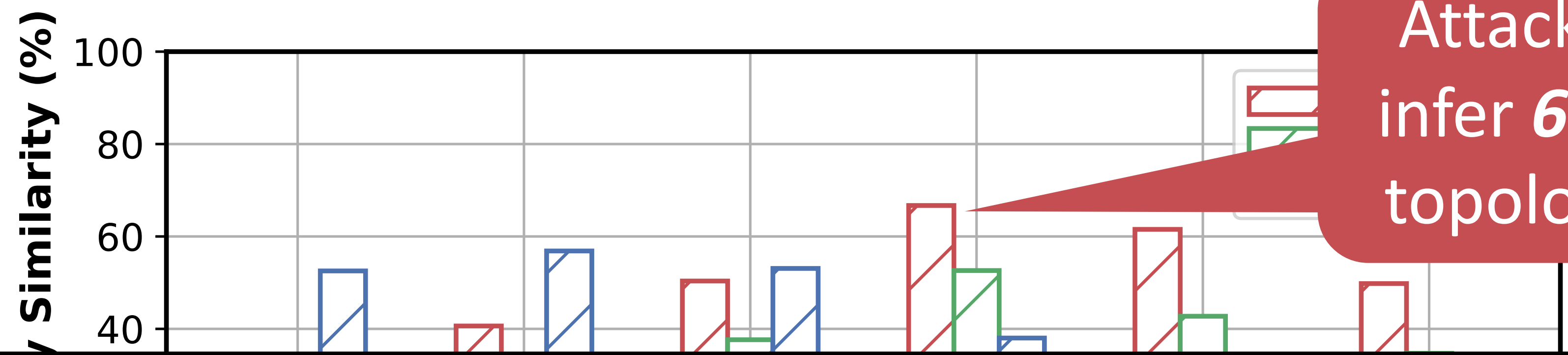
- Can attackers infer a real network topology?
  - Assuming that all links use the same mask (best for attackers)
  - By trying all possible masks (e.g., from /24 to /29)
- Compared how the inferred topology is similar with the real



Attackers could infer ***60%*** similar topology at best

Resistance to Topology Inference

- Can attackers infer a real network topology?
  - Assuming that all links use the same mask (best for attackers)
  - By trying all possible masks (e.g., from /24 to /29)
- Compared how the inferred topology is similar with the real



Attackers could infer ***60%*** similar topology at best

In practice, inferring a real network topology is more difficult because attackers cannot know exact mask (i.e., CIDR)

Protection Against Alias Resolution

- Can attackers distinguish fake responses using ***alias resolution***?
  - To identify the same router from different responses

- Tested with the popularly used tools
  - `Scamper`: Comparison of IP ID patterns
  - `Kapar`: Analysis of common neighbors
  - `iffinder`: Utilization of direct UDP probes to unused ports

Protection Against Alias Resolution

```
root@ambuhser:~# scamper -I "dealias -W 1000 -m ally -p '-P udp' 10.0.2.102 10.0.2.101"
10.0.2.102 10.0.2.101 not aliases
```

Cannot detect aliases

<Scamper>

```
# command line: ./kapar -ial -py -r31 -sir -c0.5 -nv -adms -d1 -mn -lb -1a -oal
-z 24
#    -B bongo.txt
#    -P pathlist1.txt
#    -P pathlist2.txt
#
# found 3 nodes, containing 4 interfaces (0 redundant (omitted), 0 anonymous, 4
named).
node N1:  10.0.2.102 10.0.3.101
node N2:  10.0.2.101
node N3:  10.0.0.3
```

Cannot detect aliases

<Kapar>

```
Using local port 48196.
# iffinder revision:     $Revision: 1.48 $
# addr          alias          o-TTL-r       RTT result  discovr feature record_route
Pass 0: probing 2 known addresses...
10.0.2.102      10.0.2.102     255  64  0.012433 S       -       -
10.0.2.101      10.0.2.101     255  64  0.010792 S       -       -
```

Cannot detect aliases

<Iffinder>

Protection Against Alias Resolution

```
root@ambuhser:~# scamper -I "dealias -W 1000 -m ally -p '-P udp' 10.0.2.102 10.0.2.101"
10.0.2.102 10.0.2.101 not aliases
```

→ ***Cannot detect aliases***

<Scamper>

```
# command line: ./kapar -ial -py -r31 -sir -c0.5 -nv -adms -d1 -mn -lb -1a -oal
-z 24
#    -B bongo.txt
#    -P pathlist1.txt
#    -P pathlist2.txt
#
# found 3 nodes, containing 4 interfaces (0 redundant (omitted), 0 anonymous, 4
named).
node N1:   10.0.2.102 10.0.3.101
node N2:   10.0.2.101
node N3:   10.0.0.3
```

→ ***Cannot detect aliases***

<Kapar>

```
Using local port 48196.
```

→ ***Cannot detect aliases***

Attackers cannot distinguish fake responses from real ones even if they use a sophisticated analysis technique

Summary

- Prior network topology obfuscation solutions
  - Proposed to mitigate link flooding attacks proactively
  - Limited in security and practicality for long-term

- EqualNet: A **secure** and **practical** defense for **long-term** network topology obfuscation
  - Generates fake responses having virtual IP addresses
  - Hides interfaces by adding virtual nodes
  - Keeps topology utility for subnet-level

Thank you for listening

jinwoo.kim@kaist.ac.kr