

# NetHide: Secure and Practical Network Topology Obfuscation

Roland Meier<sup>(1)</sup>, Petar Tsankov<sup>(1)</sup>, Vincent Lenders<sup>(2)</sup>,  
Laurent Vanbever<sup>(1)</sup>, Martin Vechev<sup>(1)</sup>

[nethide.ethz.ch](http://nethide.ethz.ch)

USENIX Security 2018



(1)

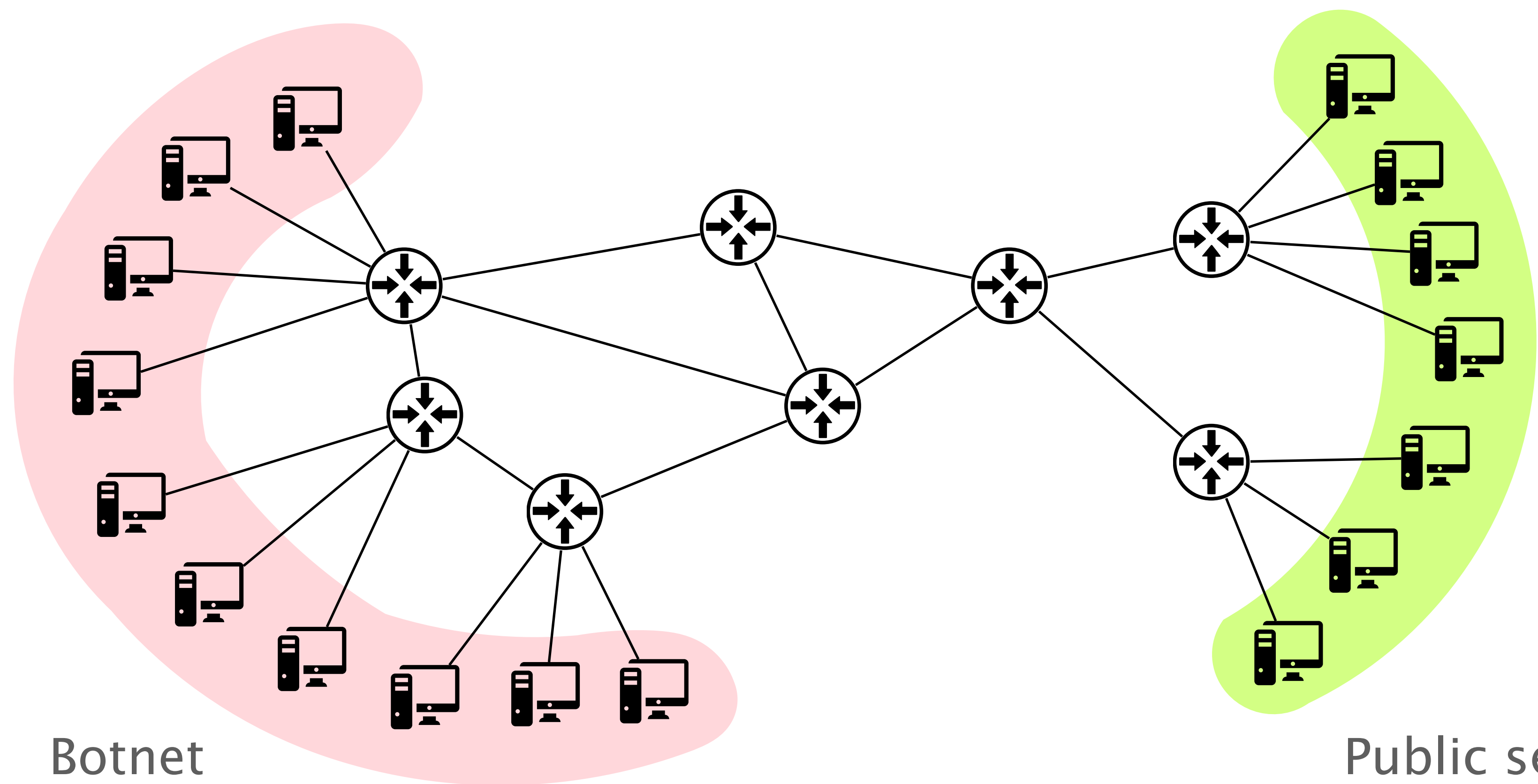
**ETH** zürich

(2)



Schweizerische Eidgenossenschaft  
Confédération suisse  
Confederazione Svizzera  
Confederaziun svizra

arnasuisse



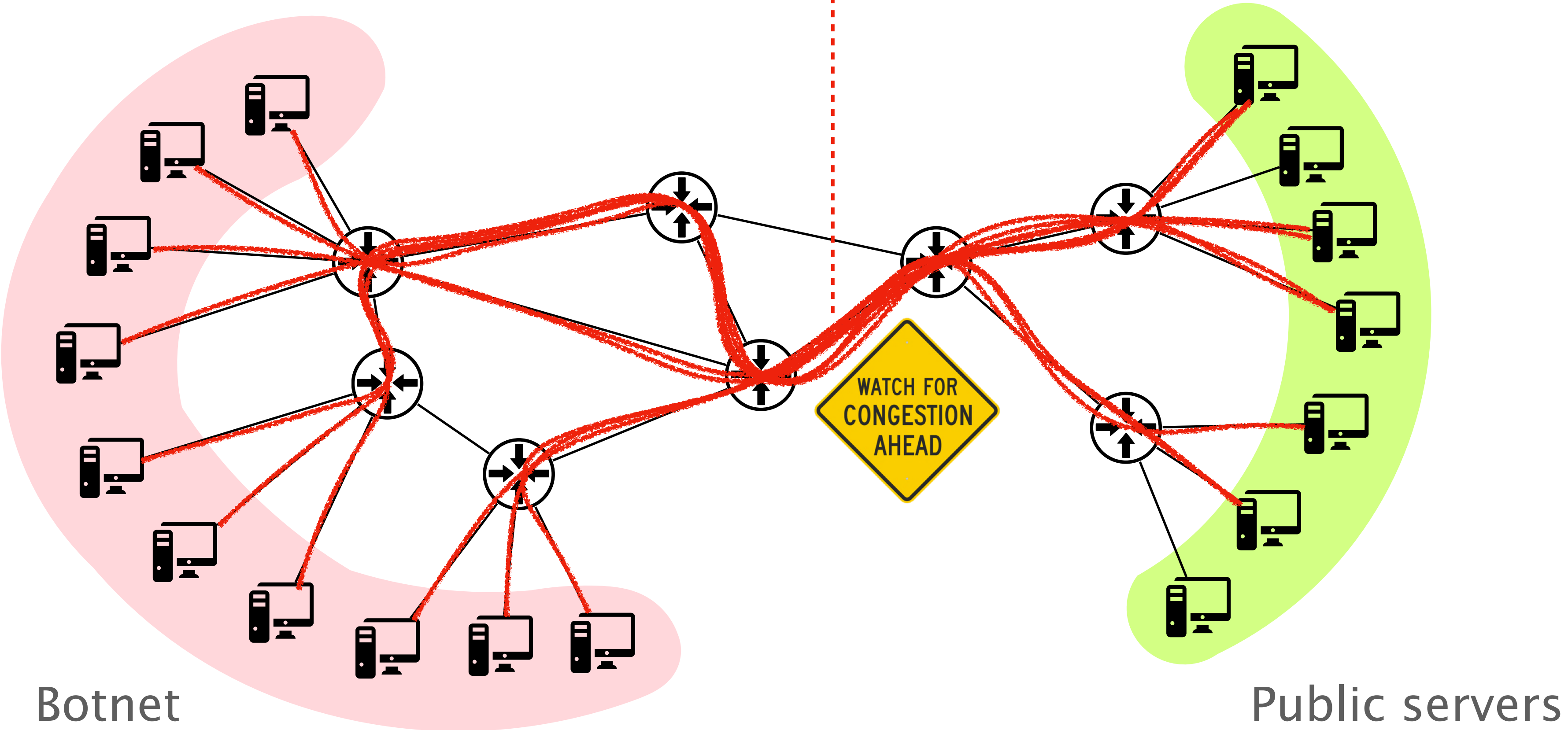
Botnet

Public servers

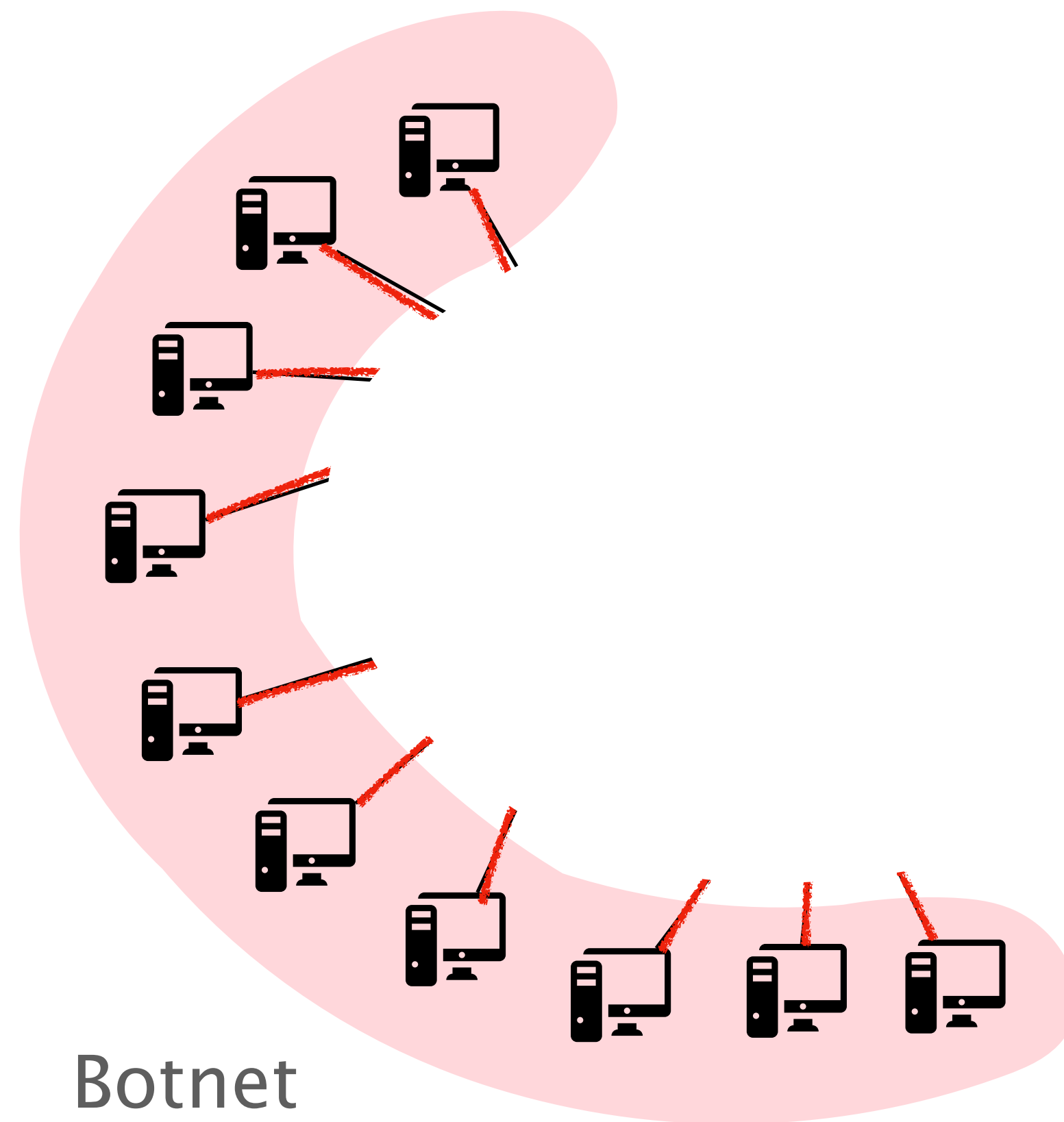


# Link-flooding attacks (LFAs) target the infrastructure

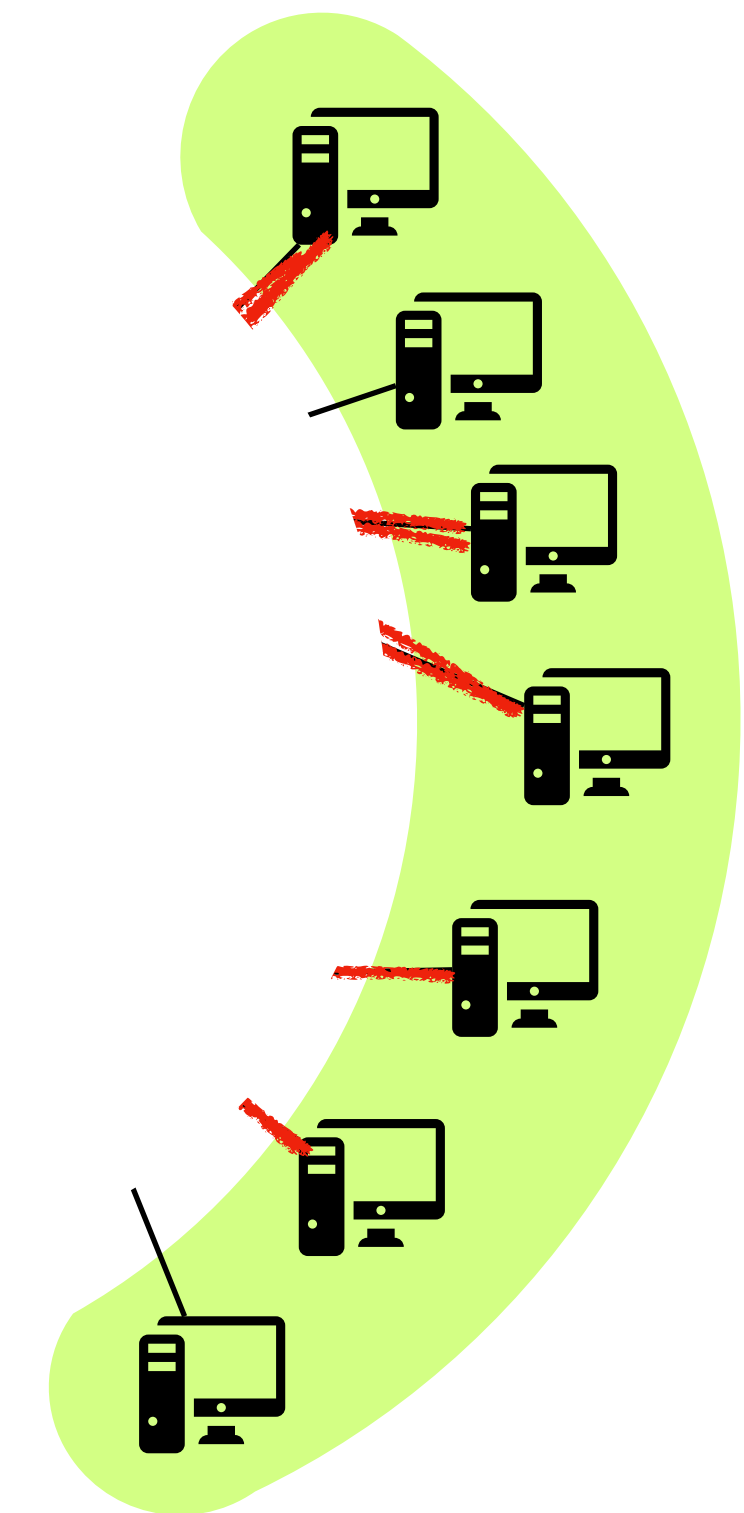
Low-rate, legitimate flows spread over many endpoints



# Link-flooding attacks (LFAs) require knowing the topology

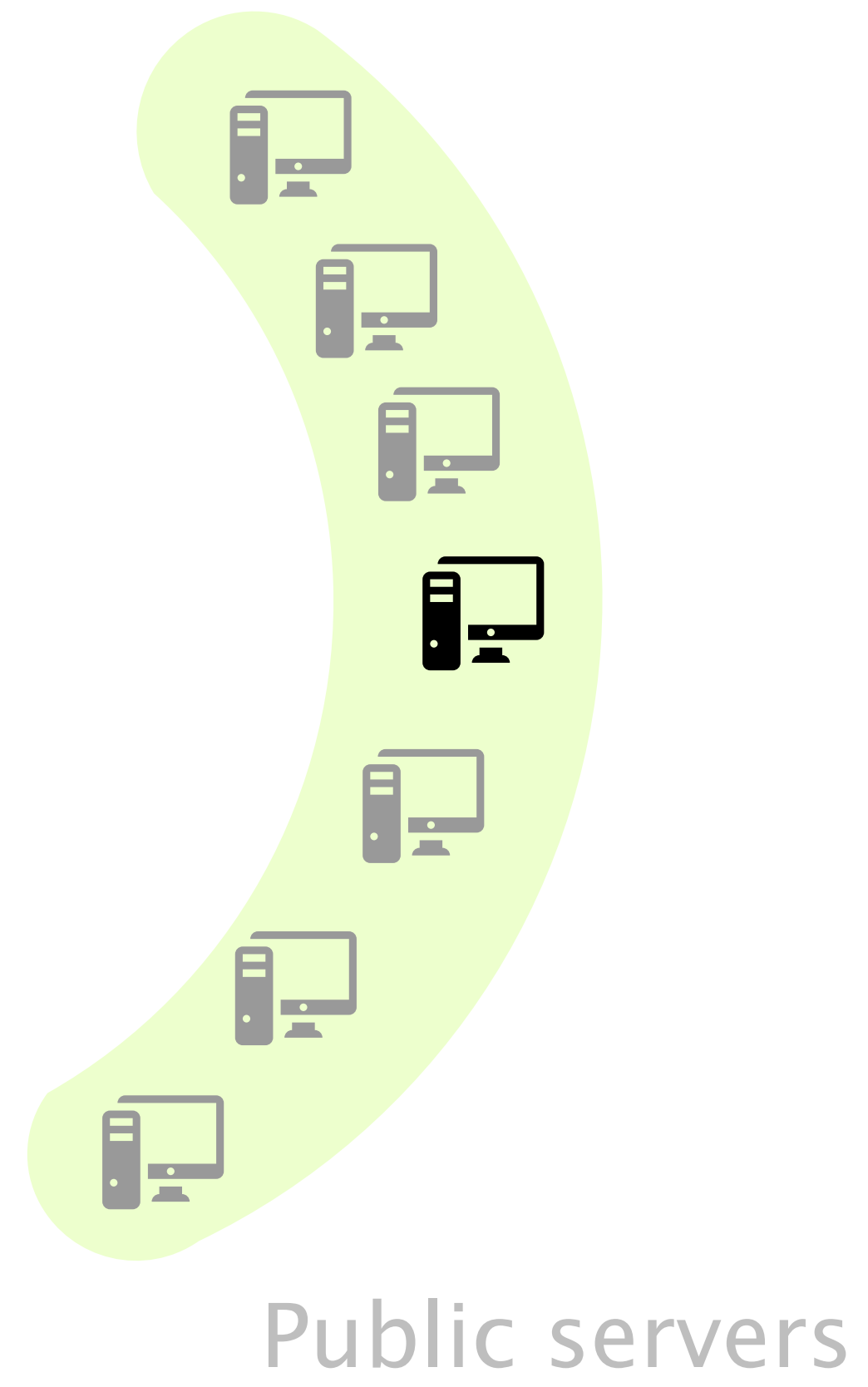
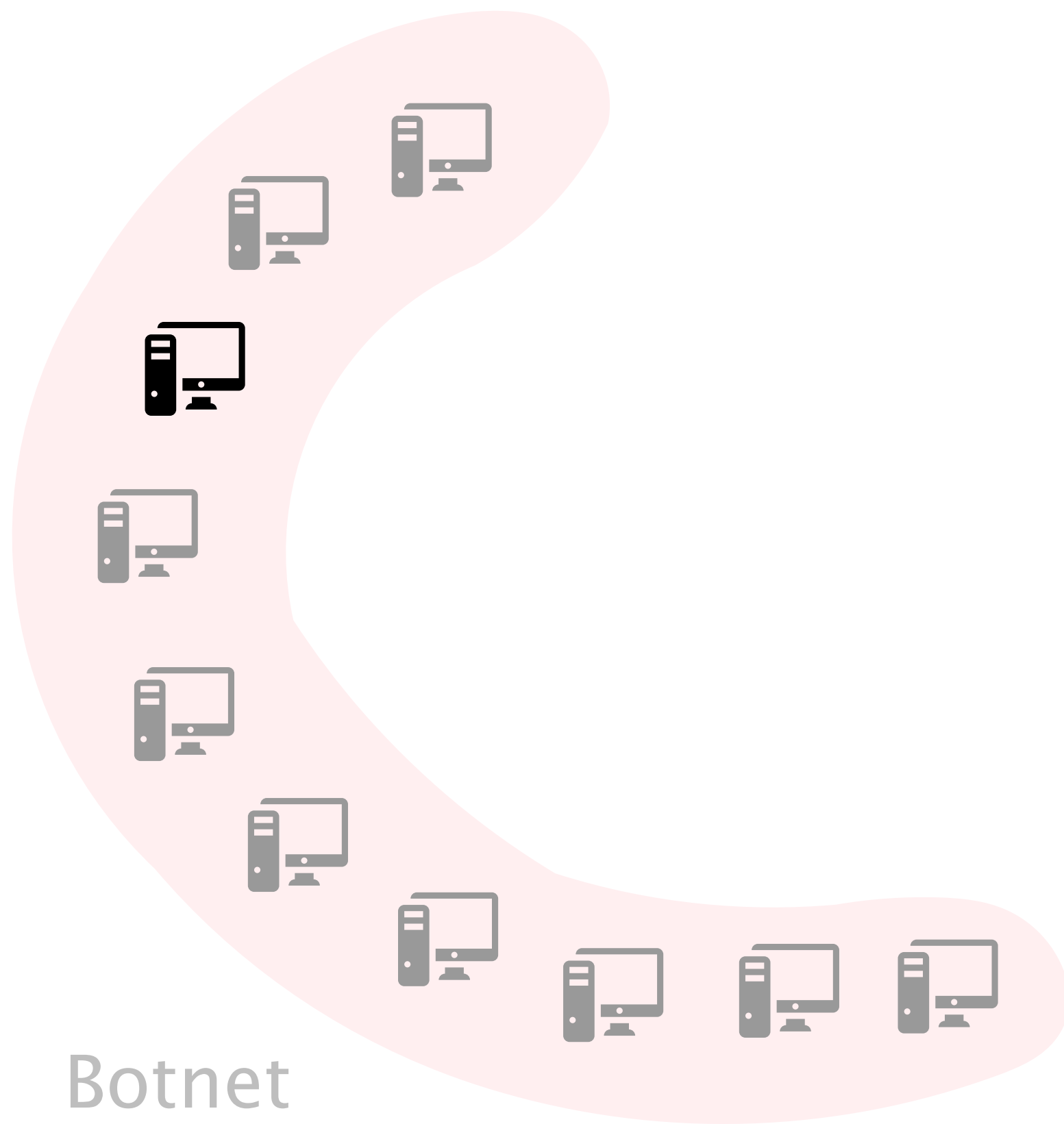


?

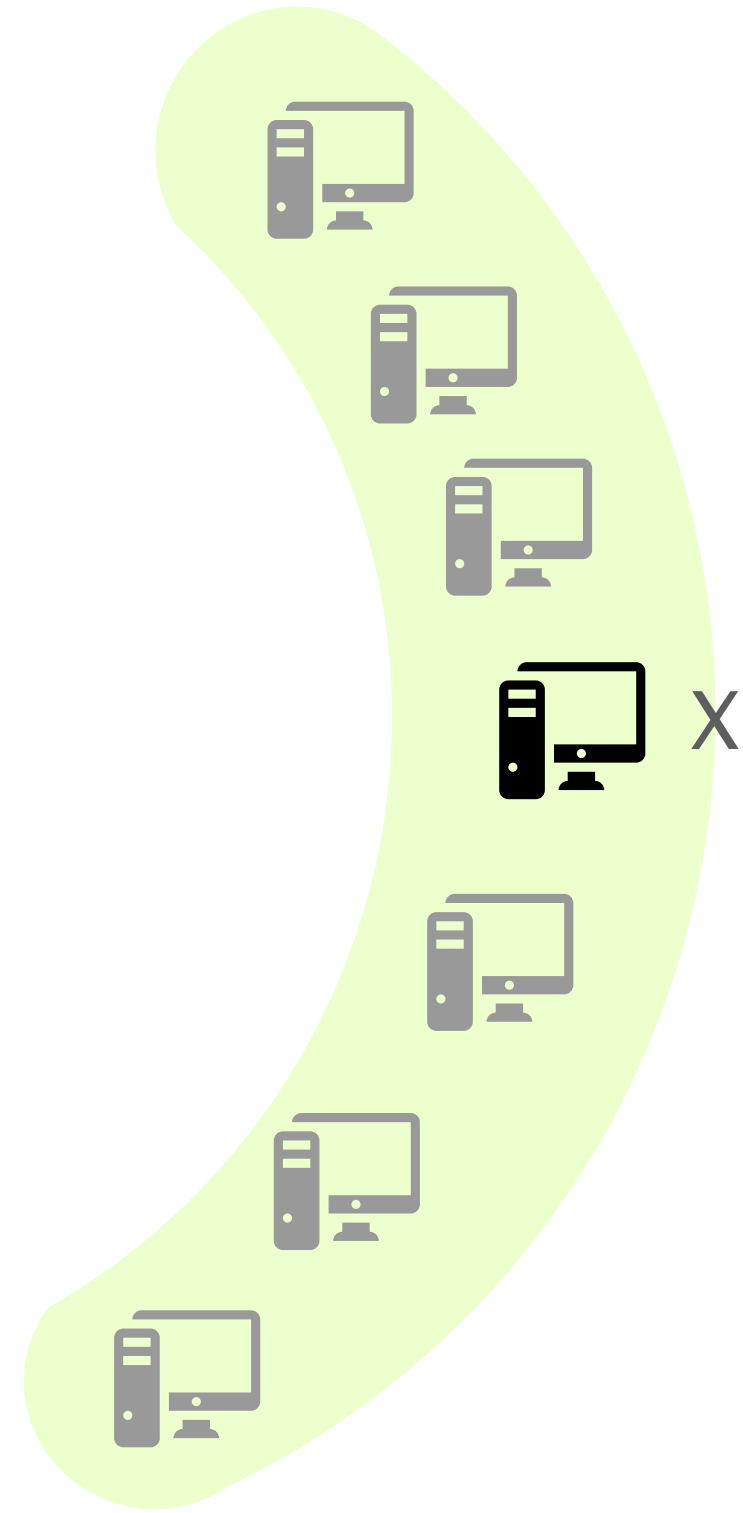
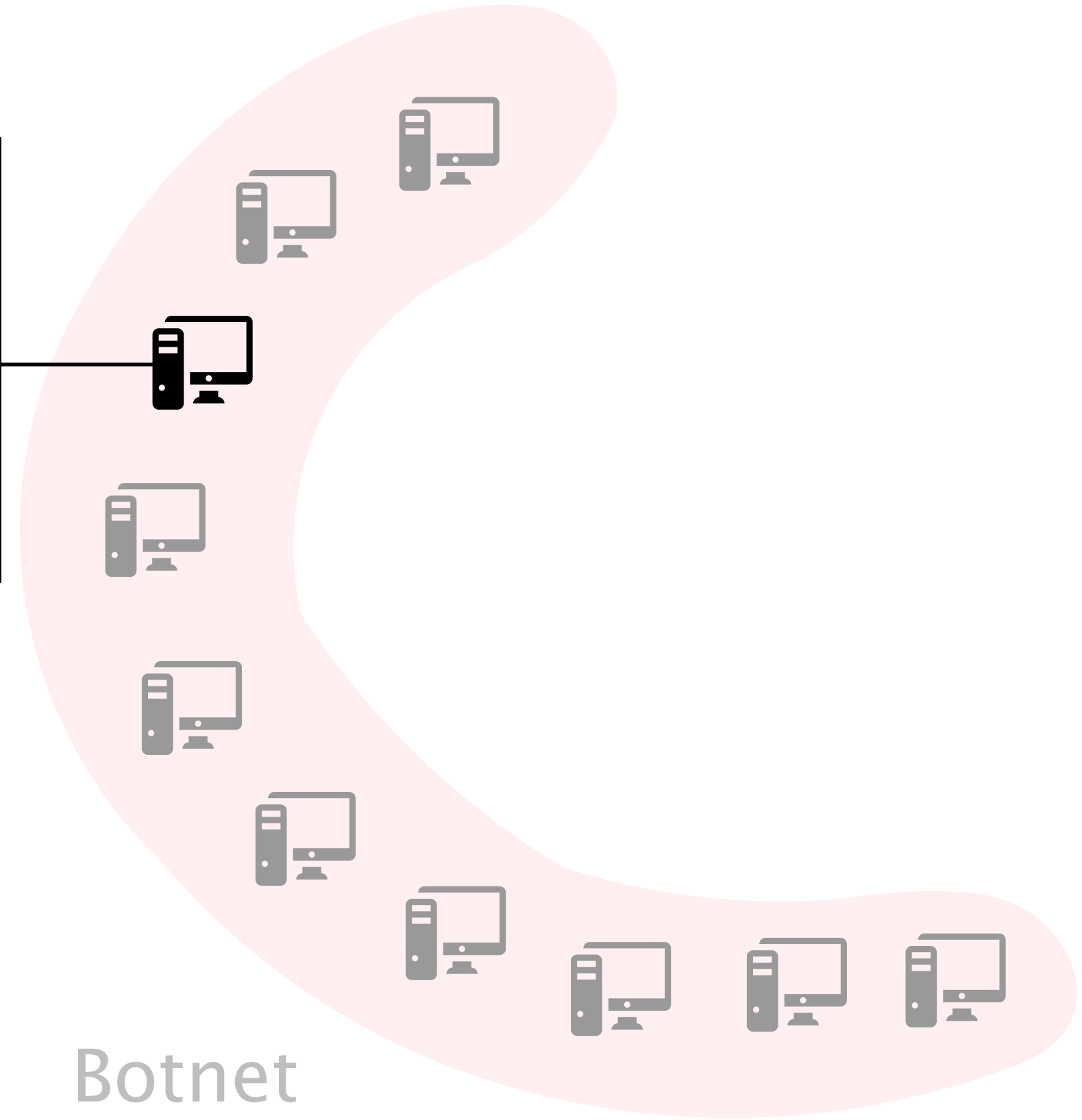


Public servers

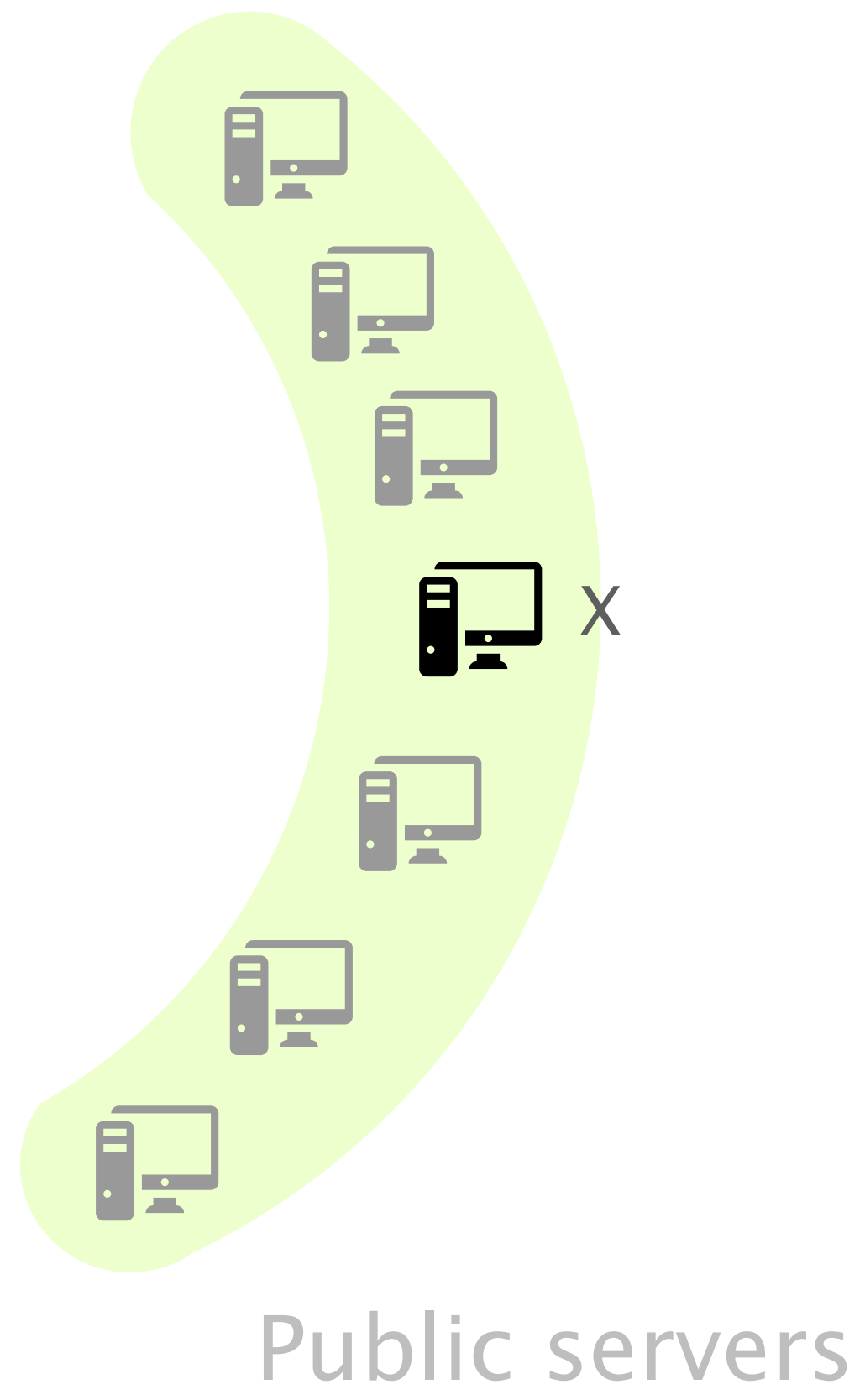
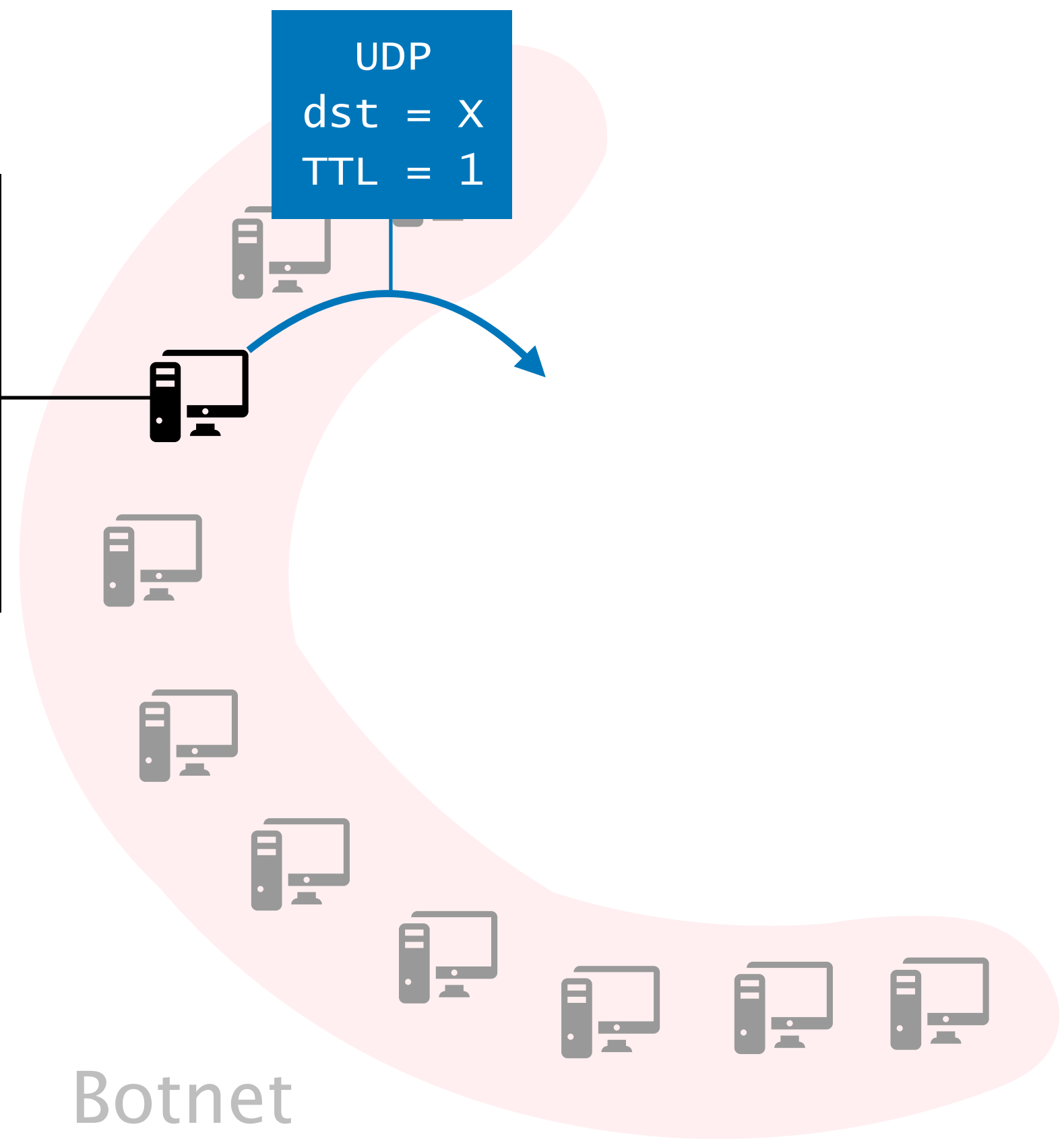




```
$ traceroute x
1
```

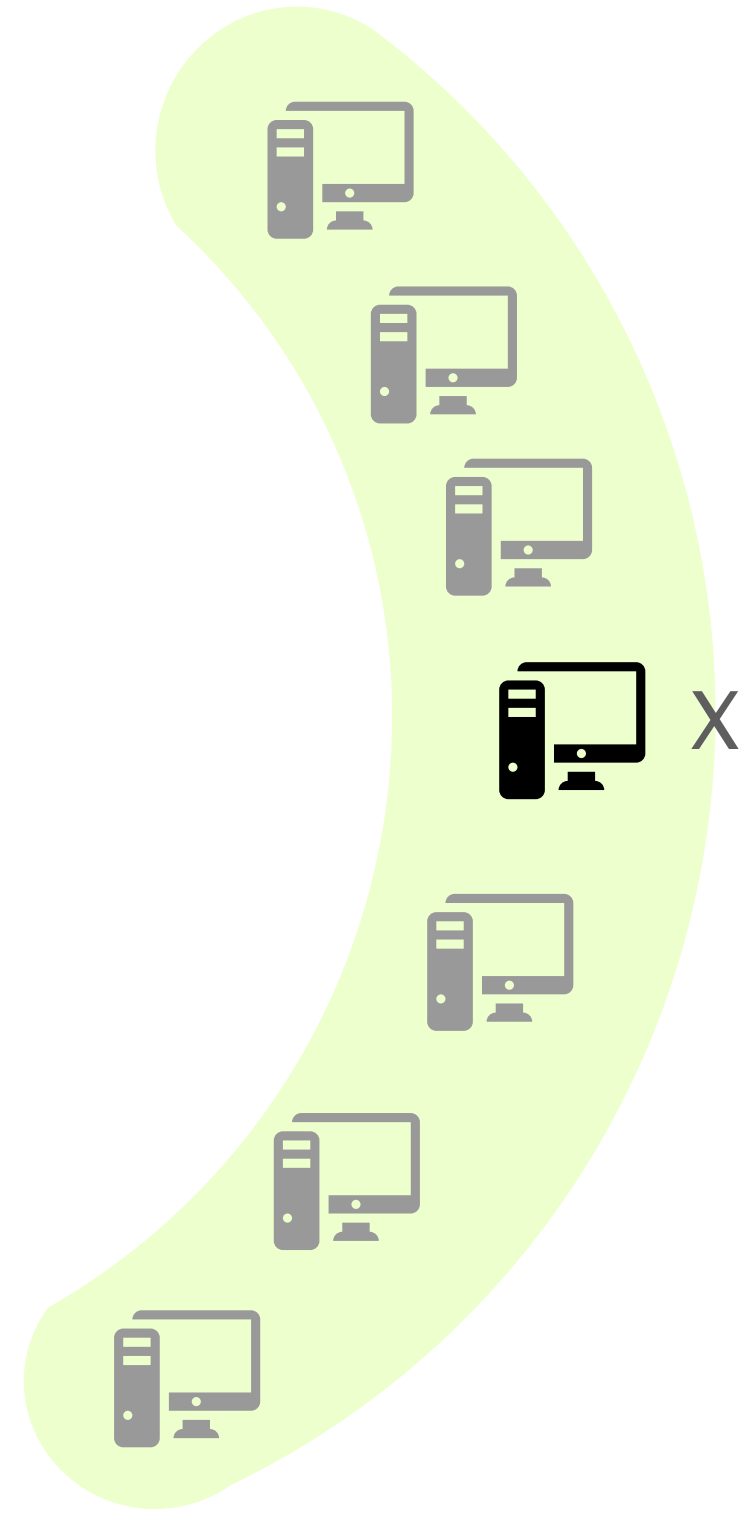
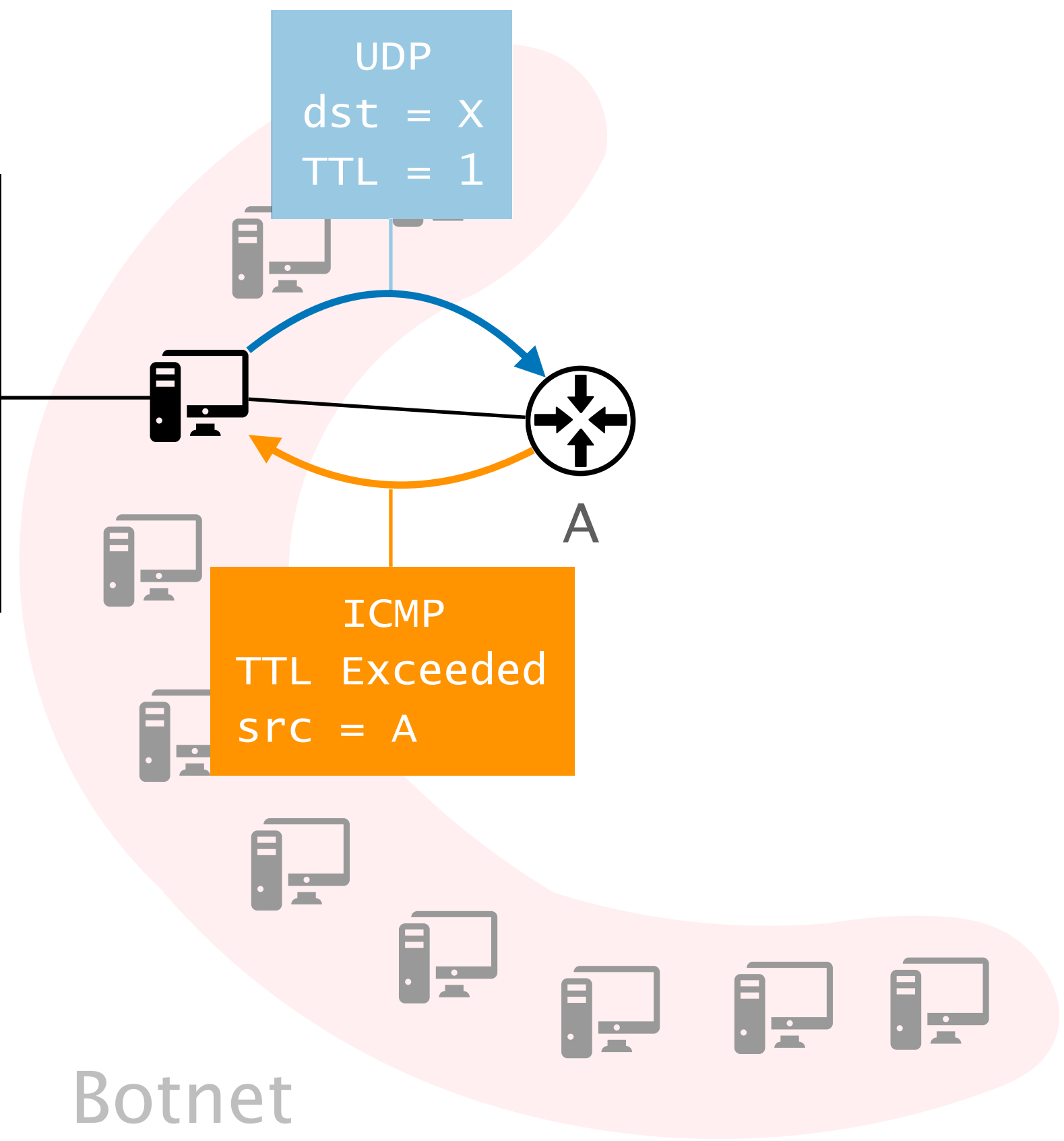


```
$ traceroute x  
1
```

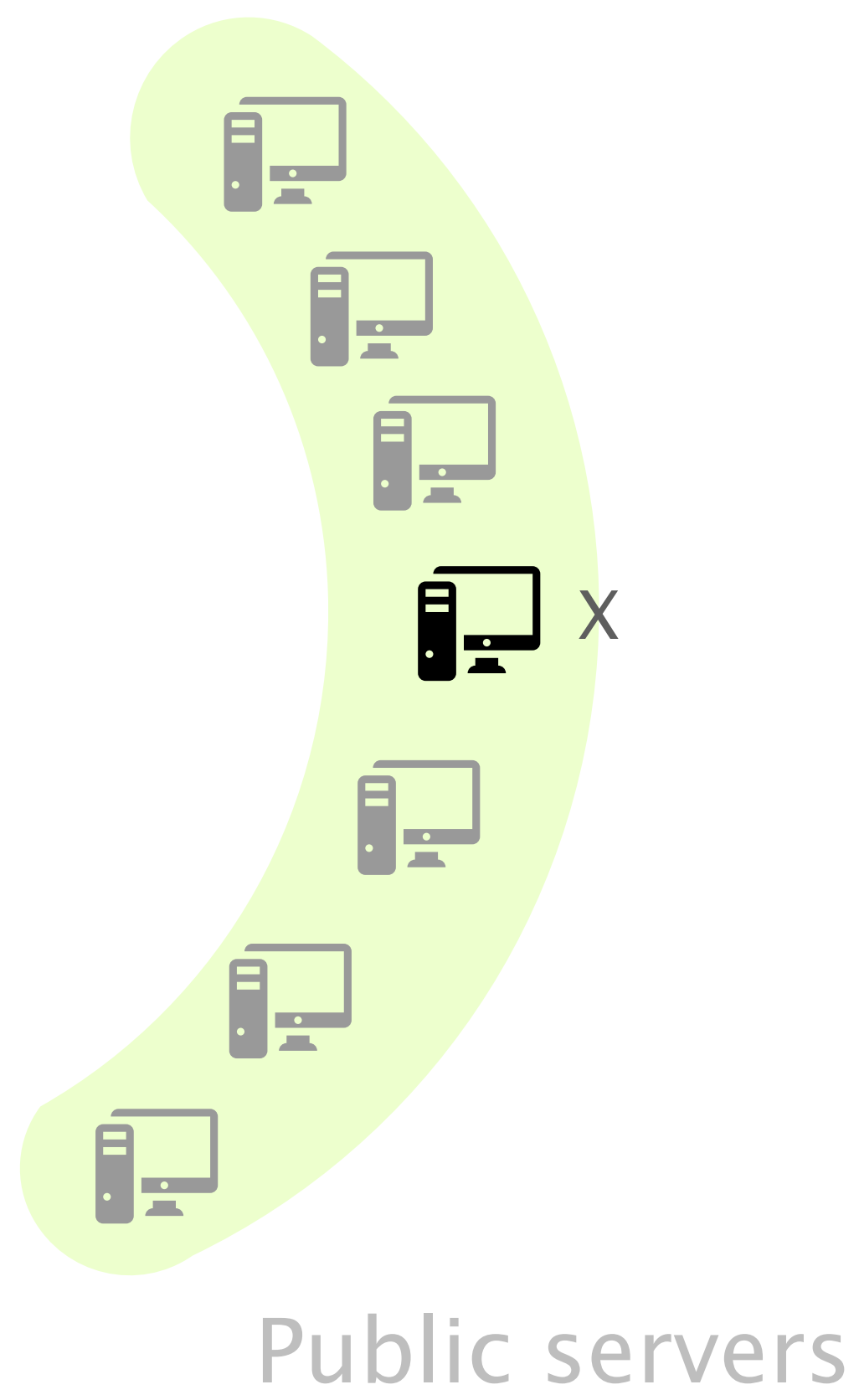
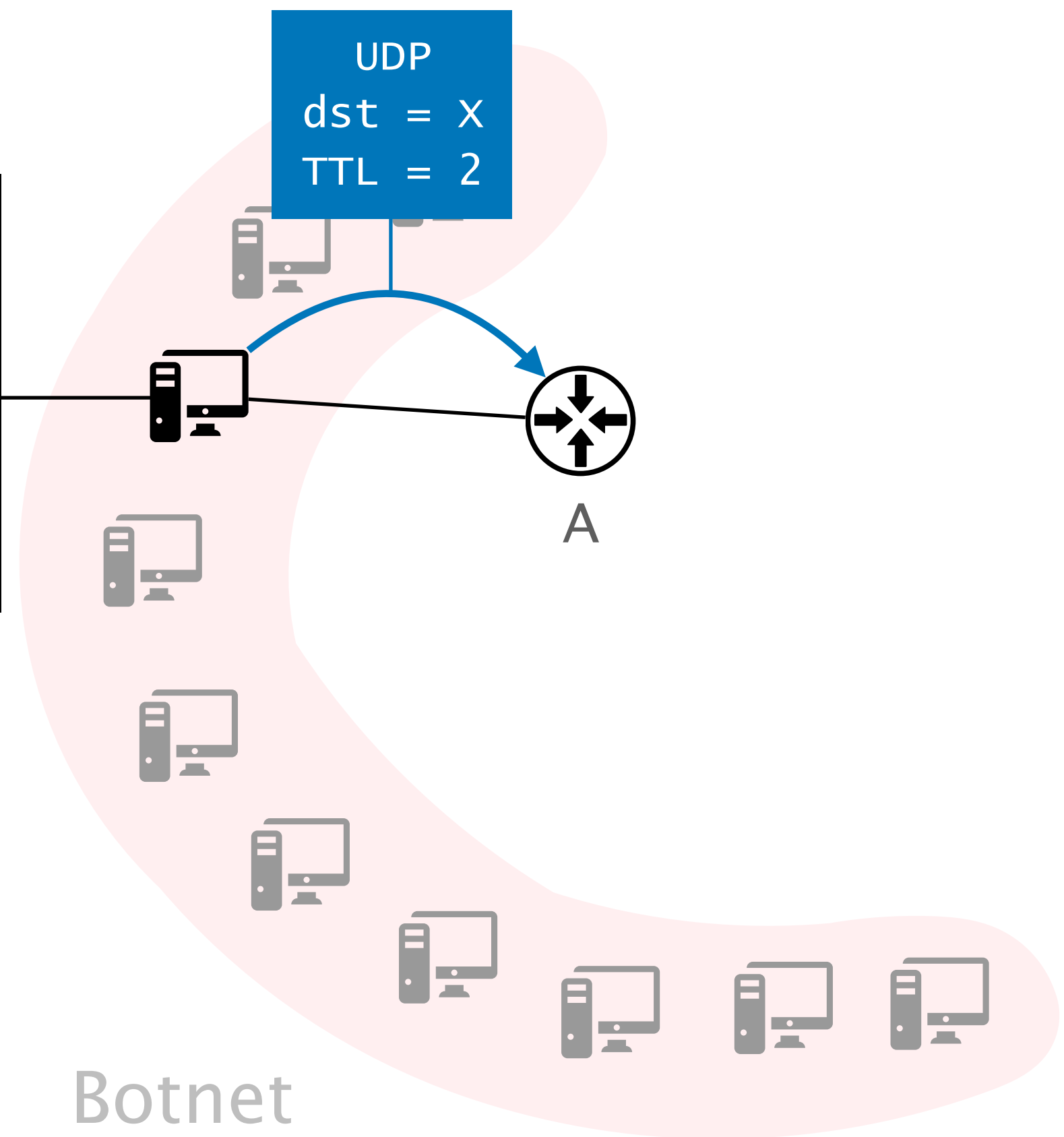




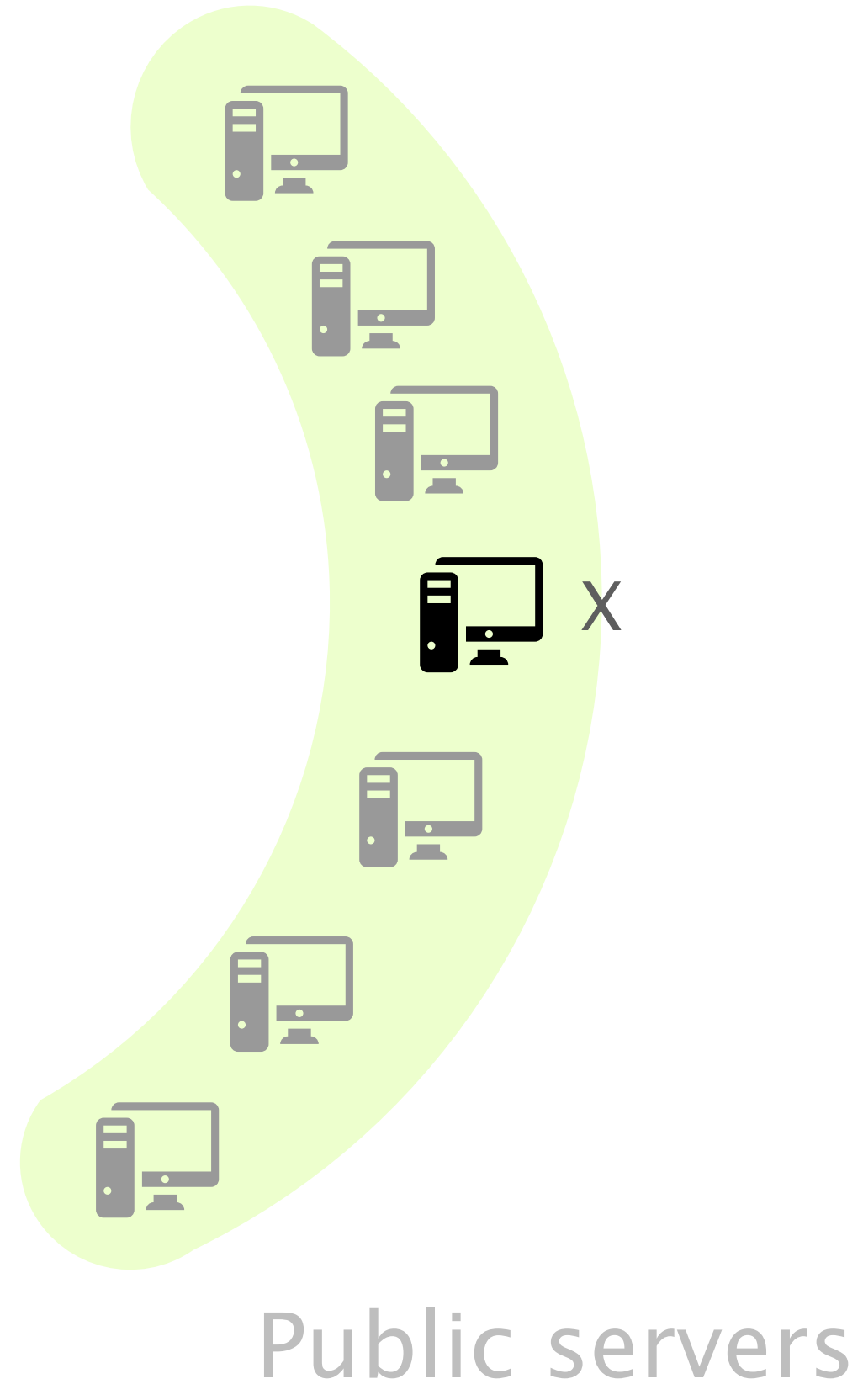
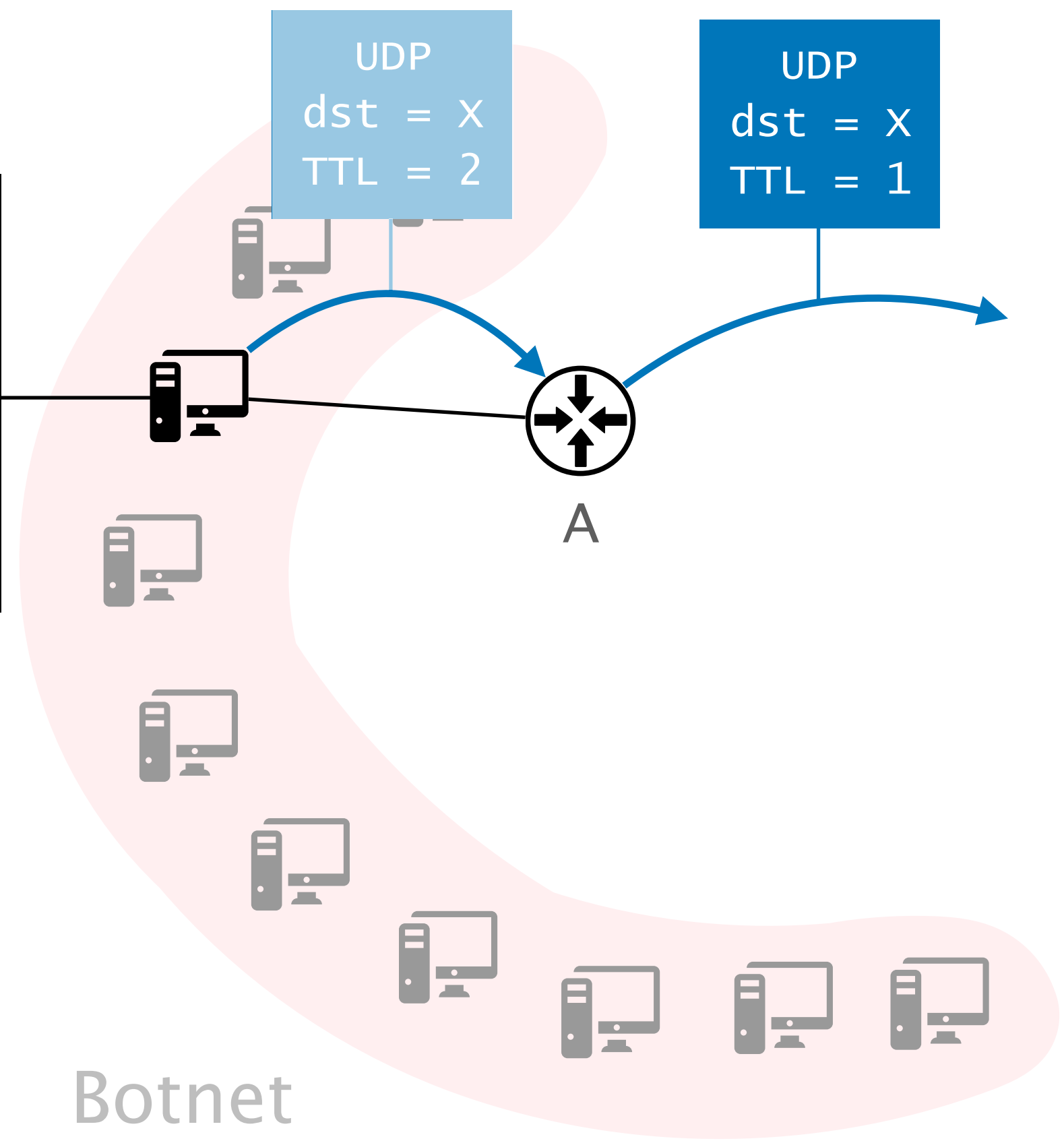
```
$ traceroute x
1 -A- 1.755 ms
```



```
$ traceroute X
1  -A-  1.755 ms
2
```

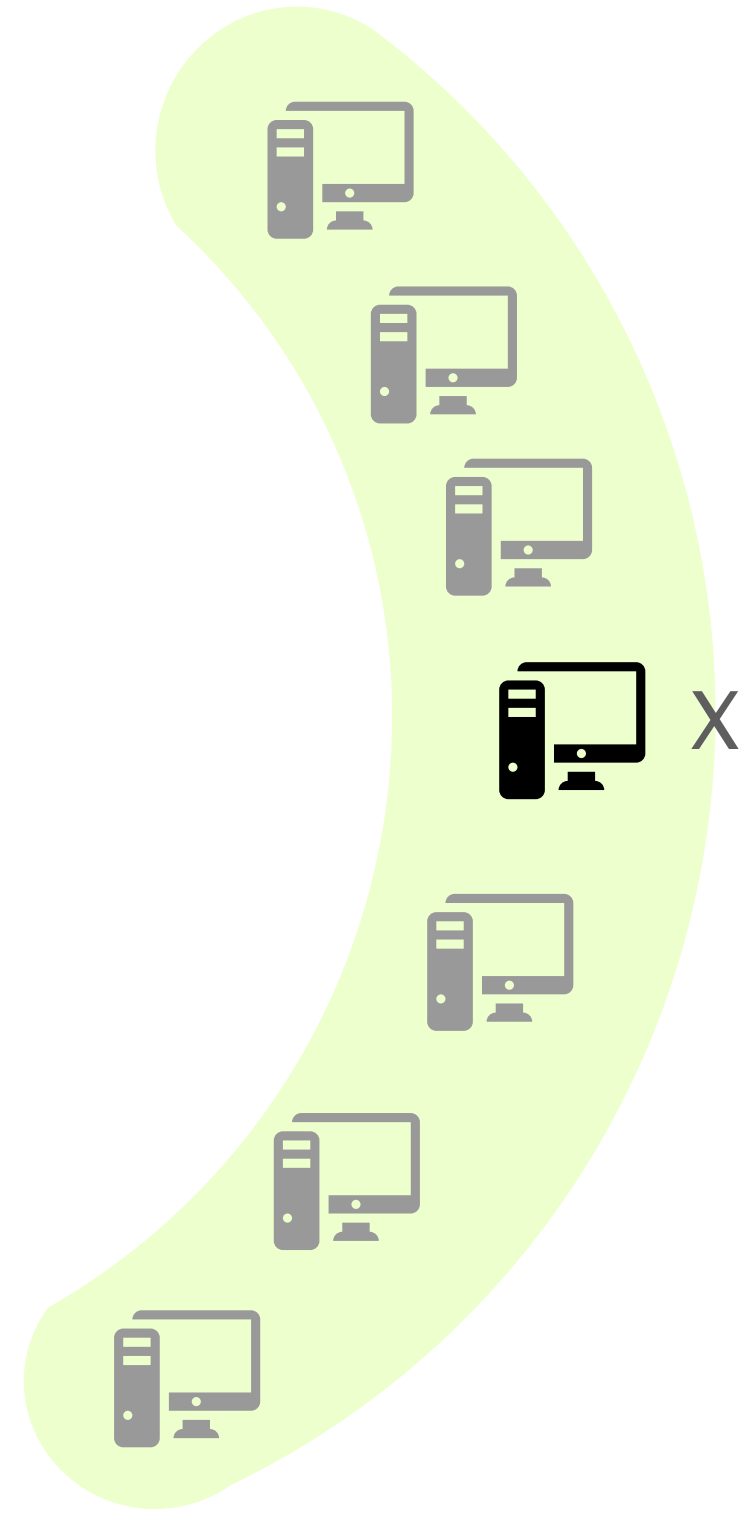
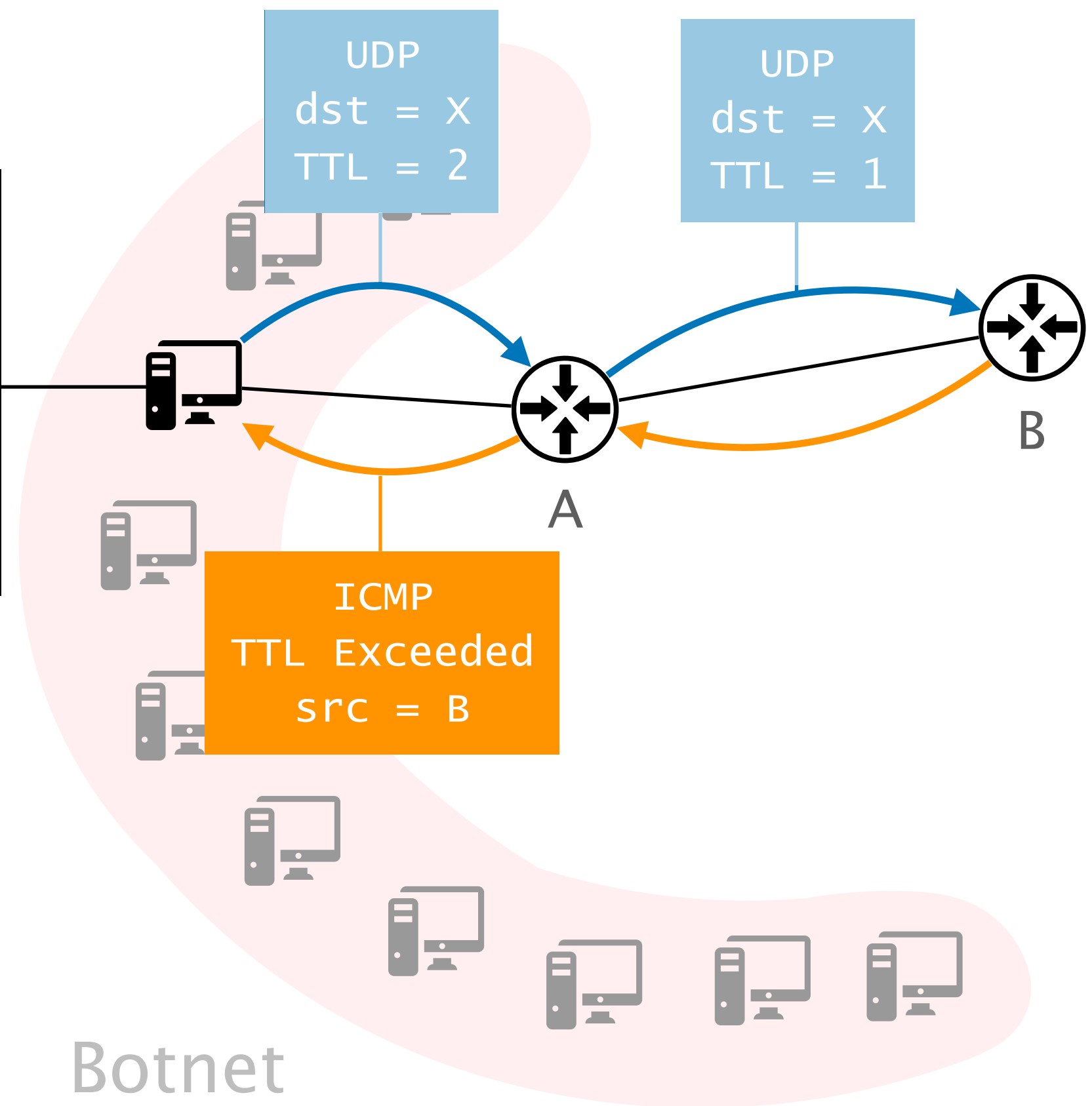


```
$ traceroute x
1  -A-  1.755 ms
2
```





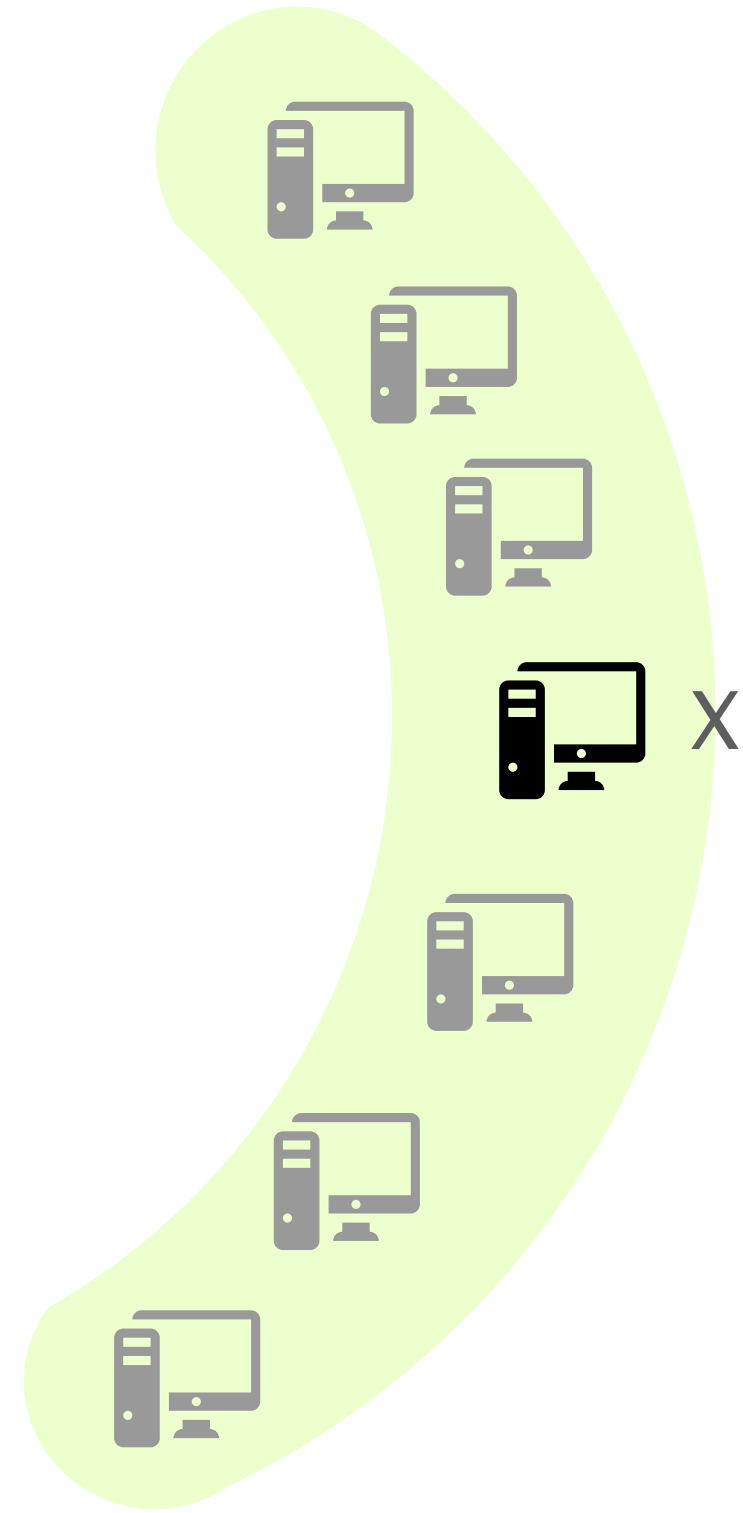
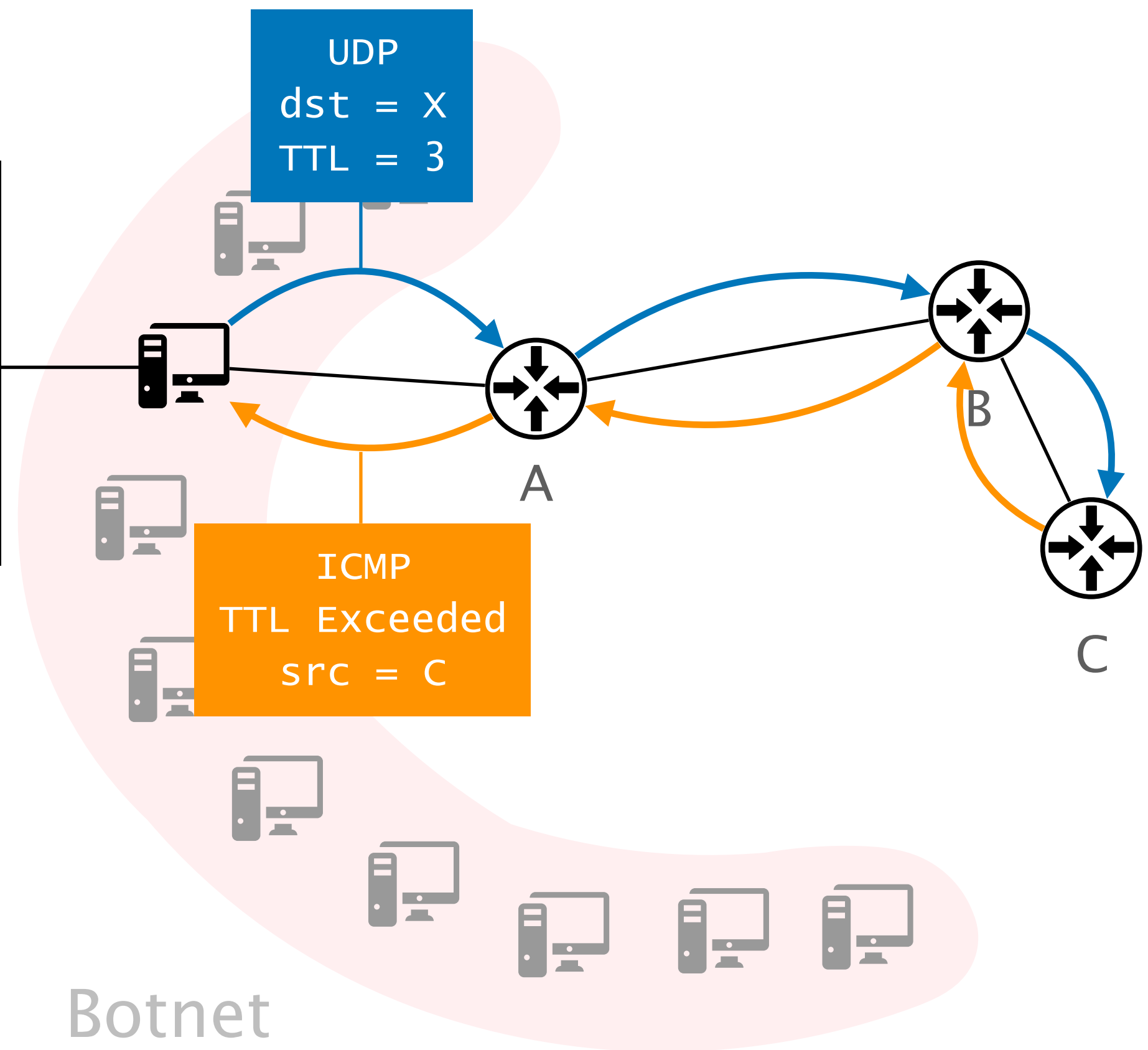
```
$ traceroute X
1  -A-  1.755 ms
2  -B-  1.062 ms
```



Botnet

Public servers

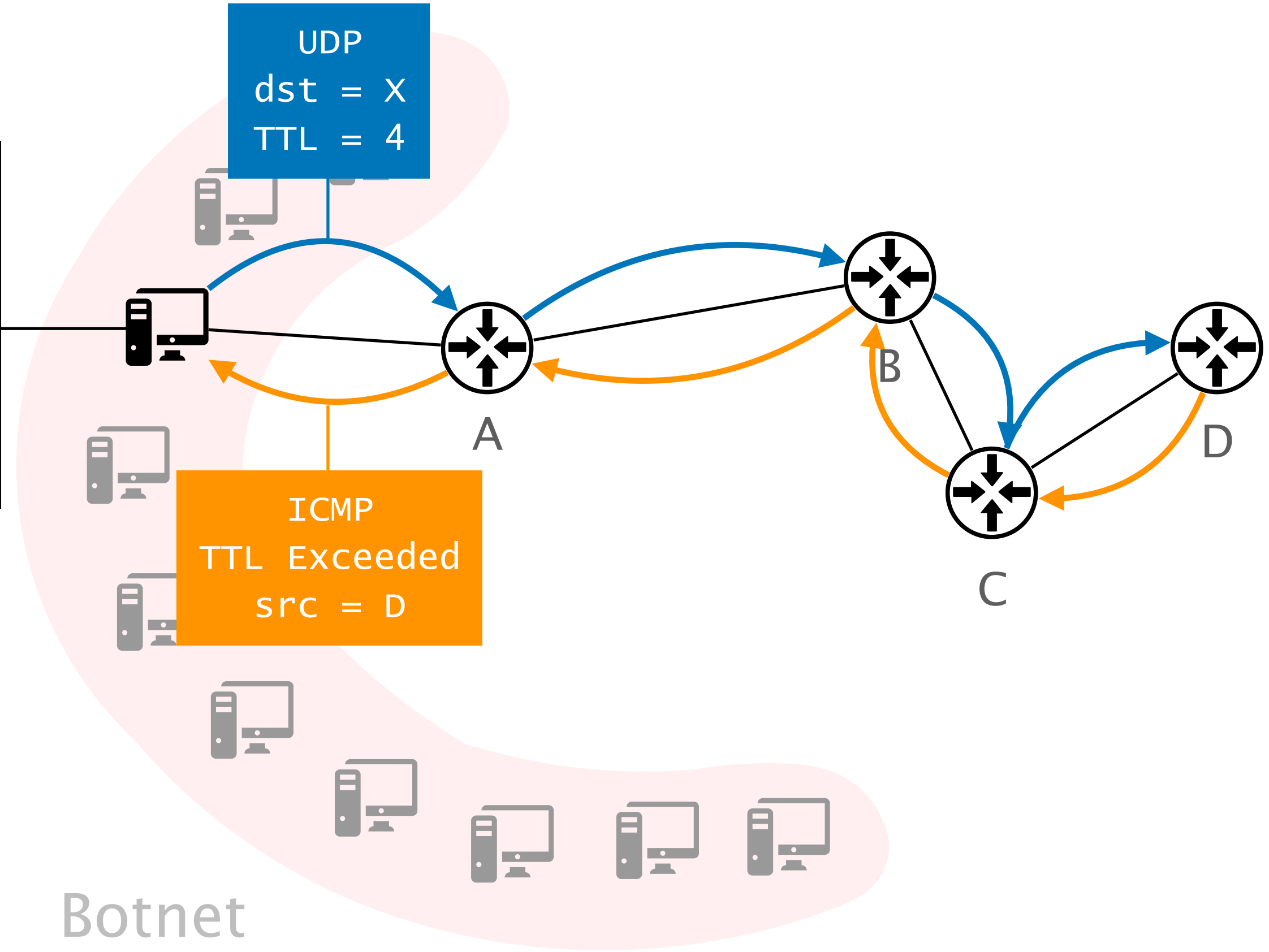
```
$ traceroute X
1  -A-  1.755 ms
2  -B-  1.062 ms
3  -C-  0.880 ms
```



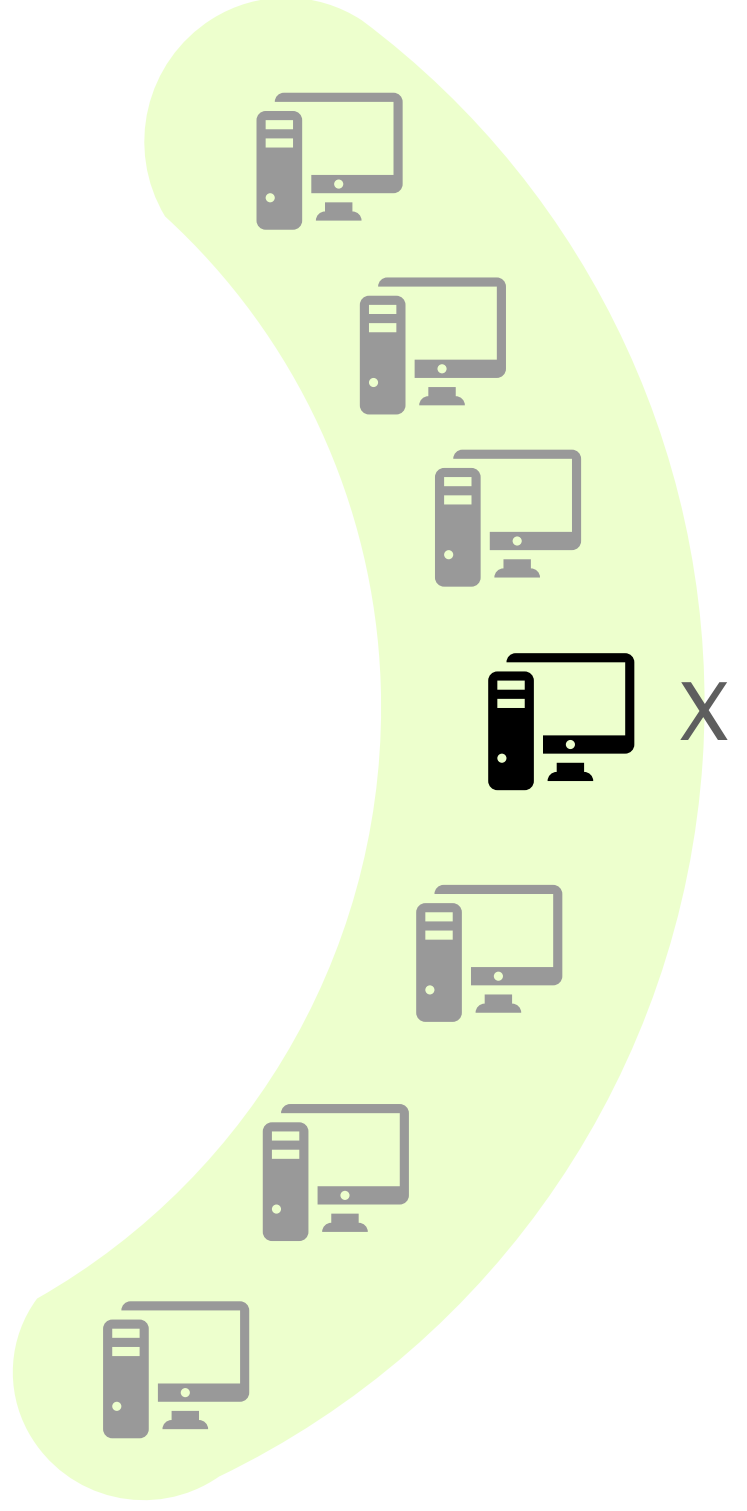
```
$ traceroute X
1  -A-  1.755 ms
2  -B-  1.062 ms
3  -C-  0.880 ms
4  -D-  0.929 ms
```

UDP  
dst = X  
TTL = 4

ICMP  
TTL Exceeded  
src = D



Botnet



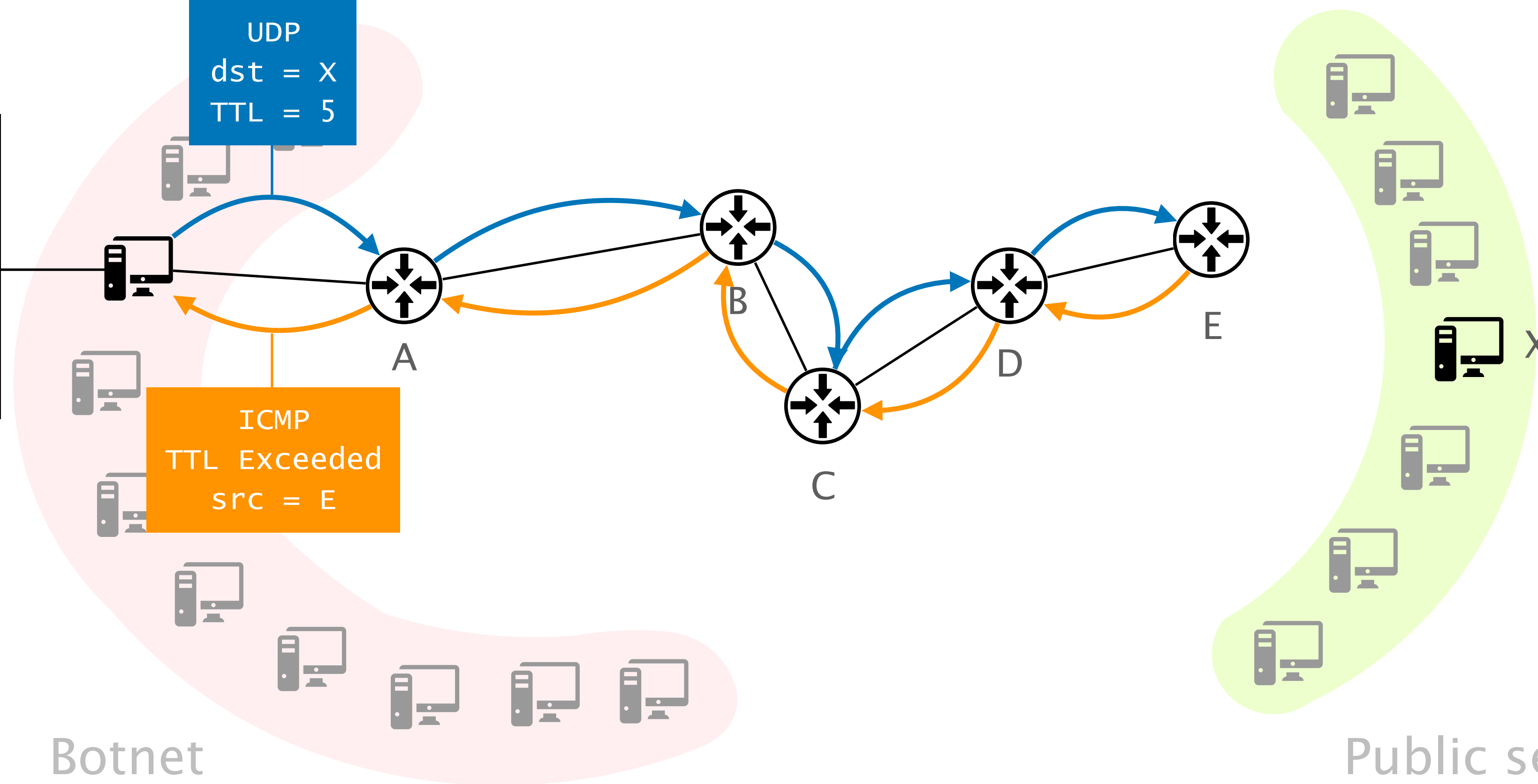
Public servers



```
$ traceroute X
1  -A-  1.755 ms
2  -B-  1.062 ms
3  -C-  0.880 ms
4  -D-  0.929 ms
5  -E-  0.827 ms
```

UDP  
dst = X  
TTL = 5

ICMP  
TTL Exceeded  
src = E



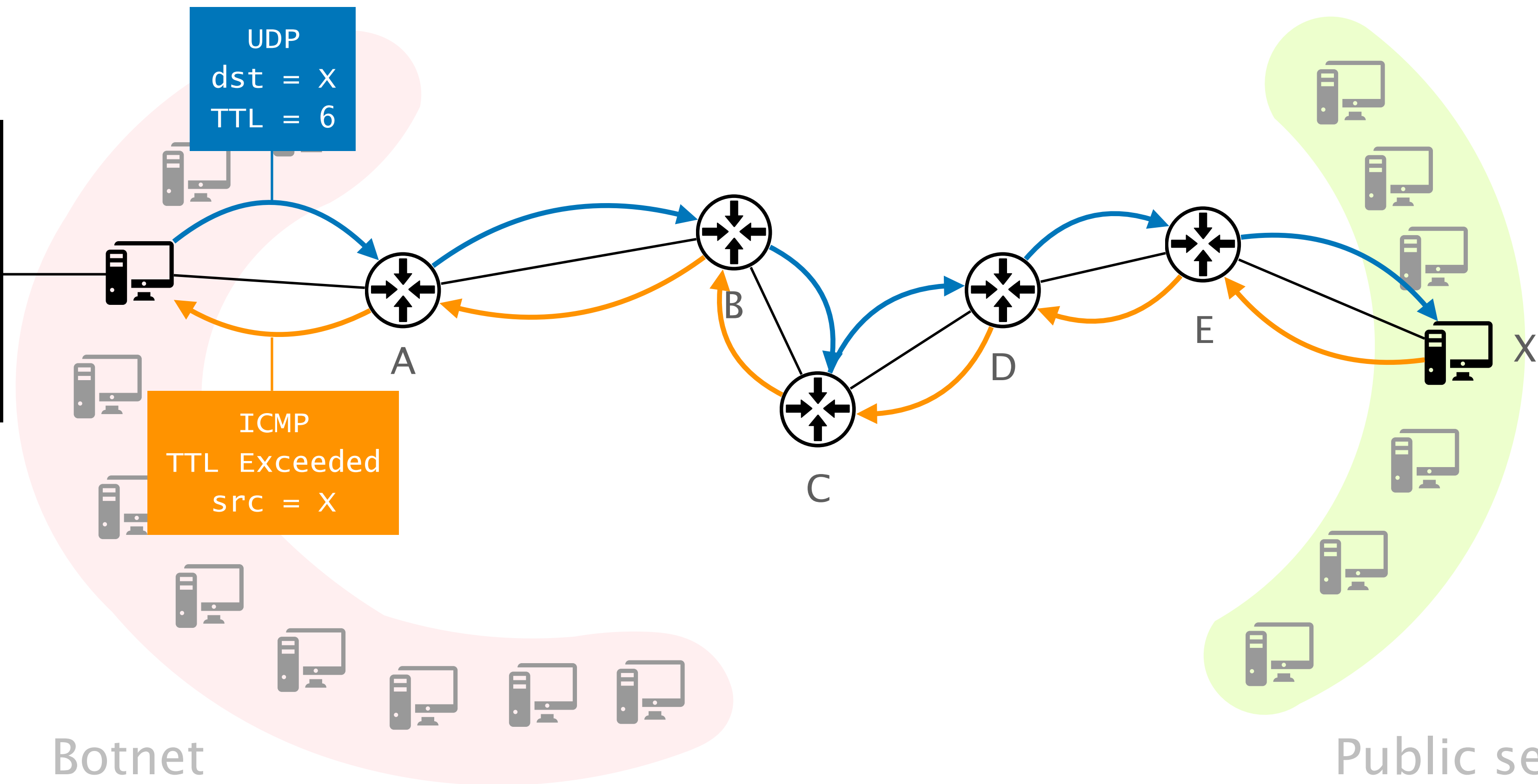
Botnet

Public servers

```

$ traceroute X
1  -A-  1.755 ms
2  -B-  1.062 ms
3  -C-  0.880 ms
4  -D-  0.929 ms
5  -E-  0.827 ms
6  -X-  0.819 ms

```



So the solution is to hide the topology?

yes, but...





## traceroute from XO network?

## Traceroute from within Colombia?

### Cloudflare 1.1.1.1 public DNS broken w/ AT&T CPE

Paul Rolland (=?UTF-8?B?44Od44O844Or44O744Ot44Op44Oz?=[rol at witbe.net](#))

Tue Apr 3 06:22:04 UTC 2018

- Previous message (by thread): [Can anyone check this routing against Charter in WI?](#)
- Next message (by thread): [Can anyone check this routing against Charter in WI?](#)
- Messages sorted by: [\[ date \]](#) [\[ thread \]](#) [\[ subject \]](#) [\[ author \]](#)

Hello,

On Mon, 2 Apr 2018 16:26:13  
Marty Strong via NANOG <[nango](#)>

- > So far we know about a few
- >
- > - Pace 5268
- > - Calix GigaCenter
- > - Various Cisco Wifi access
- >
- > If you know of others please

It seems that in France, Orange

```
215 [6:20] rol at riri:~> traceroute to 1.1.1.1 (1.1.1.1)
1 * * *
2 * * *
```

### Can anyone check this routing against Charter in WI?

Shawn L [shawnl at up.net](#)

Sat Jun 14 17:04:58 UTC 2014

- Previous message: [Can anyone check this routing against Charter in WI?](#)
- Next message: [Can anyone check this routing against Charter in WI?](#)
- Messages sorted by: [\[ date \]](#) [\[ thread \]](#) [\[ subject \]](#) [\[ author \]](#)

It seems ok from here

```
traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
 4 dtr02rhn1wi-bue-1.rhn1.wi.charter.com (96.34.16.250) 20.843 ms
 5 crr02euclwi-bue-7.eucl.wi.charter.com (96.34.17.32) 27.662 ms
 6 bbr02euclwi-bue-4.eucl.wi.charter.com (96.34.2.6) 29.236 ms
```

### Has Level3 done away with traceroute??

Mel Beckman [mel at beckman.org](#)

Thu Sep 21 17:57:51 CST 2017

- Previous message (by thread): [Has Level3 done away with traceroute??](#)
- Next message (by thread): [Bell Canada \(AS 577\) and NTT \(AS 2914\) routing](#)
- Messages sorted by: [\[ date \]](#) [\[ thread \]](#) [\[ subject \]](#) [\[ author \]](#)

I am able to traceroute in and out of Level3, but it seems like some L3 internal hops are missing, as I appear to go straight from the L3 edge router to my L3 customer agg router:

```
traceroute to 206.83.0.42 (206.83.0.42), 64 hops max, 52 byte packets
 1 47.155.227.1 (47.155.227.1) 4.318 ms 4.661 ms 4.715 ms
 2 172.102.107.166 (172.102.107.166) 10.202 ms 7.521 ms
 3 ae8---0.scr02.lsan.ca.frontiernet.net (74.40.3.49) 9.609 ms 7.501 ms
 4 ae0---0.cbr01.lsan.ca.frontiernet.net (74.40.3.198) 7.169 ms 7.015 ms 7.277 ms
 5 * * *
 6 ae-4-90.edgel.losangeles9.level3.net (4.69.144.202) 8.384 ms 7.012 ms
 7 ae-2-70.edgel.losangeles9.level3.net (4.69.144.74) 9.865 ms
 8 4.68.111.22 (4.68.111.22) 9.010 ms 7.445 ms 7.314 ms
 9 sbal-ar1-xe-11-0-0.us.twtelecom.net (35.248.2.6) 9.788 ms 9.536 ms 10.266 ms
10 206-190-77-10.static.twtelecom.net (206.190.77.10) 14.739 ms 12.525 ms 9.979 ms
11 iris1.jet.net (206.83.0.42) 10.285 ms 9.880 ms 10.063 ms

traceroute to 47.155.227.1 (47.155.227.1), 64 hops max, 40 byte packets
```

traceroute is an essential debugging tool

parts of

So the solution is to hide the topology?



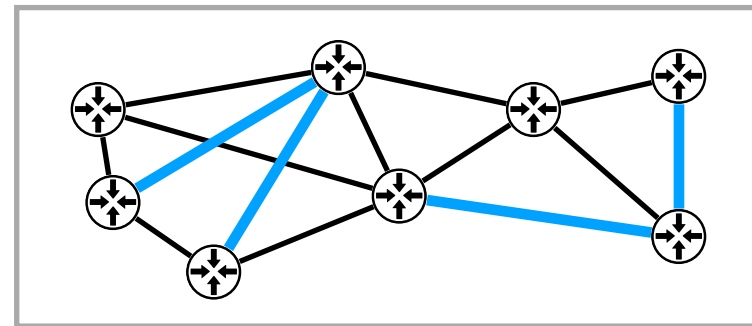
which parts?

parts of

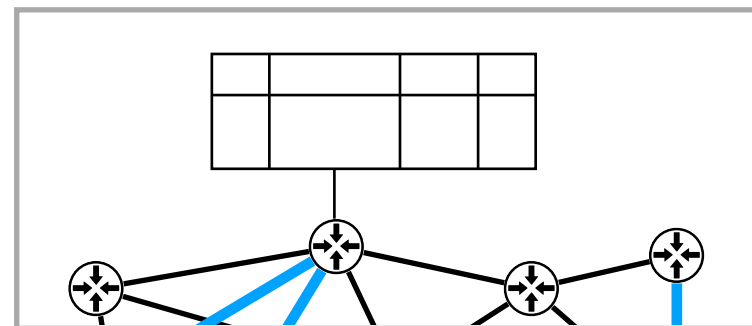
So the solution is to hide the topology?

how?

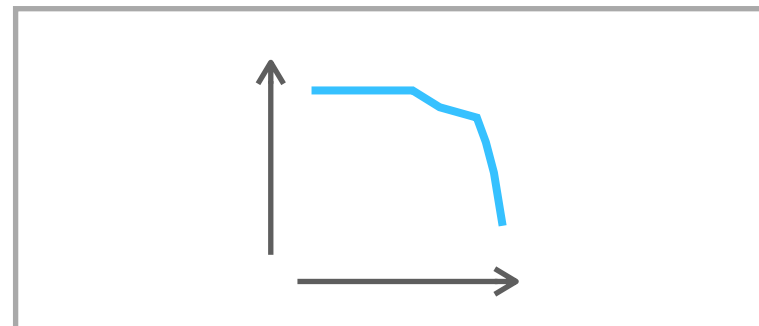
# NetHide: Secure and Practical Network Topology Obfuscation



NetHide computes a secure virtual topology that is similar to the physical topology



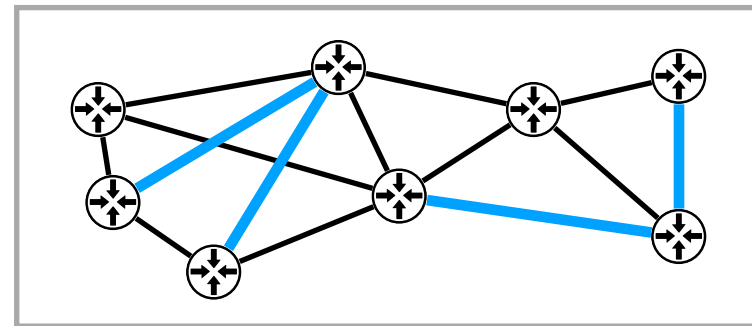
NetHide deploys the virtual topology using programmable networks



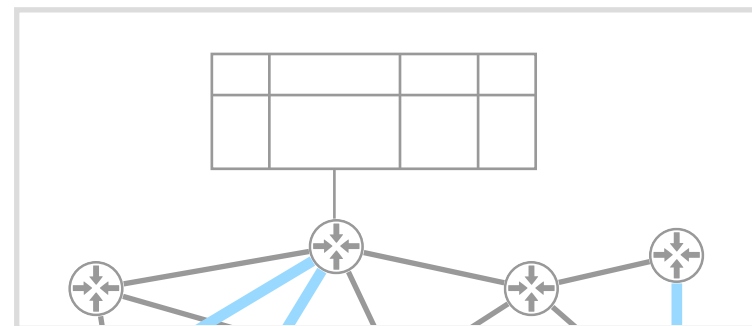
NetHide works for realistic topologies and maintains the utility of debugging tools



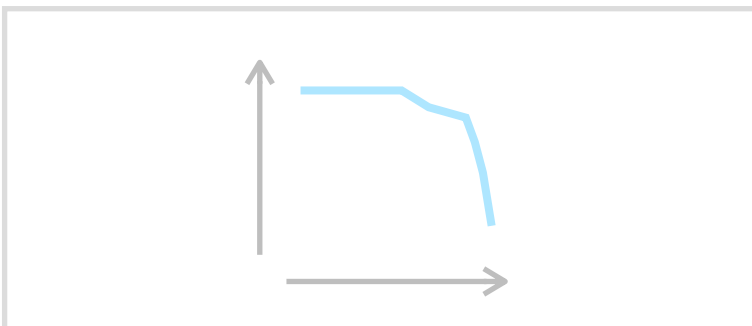
# NetHide: Secure and Practical Network Topology Obfuscation



NetHide computes a secure virtual topology that is similar to the physical topology



NetHide deploys the virtual topology using programmable networks



NetHide works for realistic topologies and maintains the utility of debugging tools

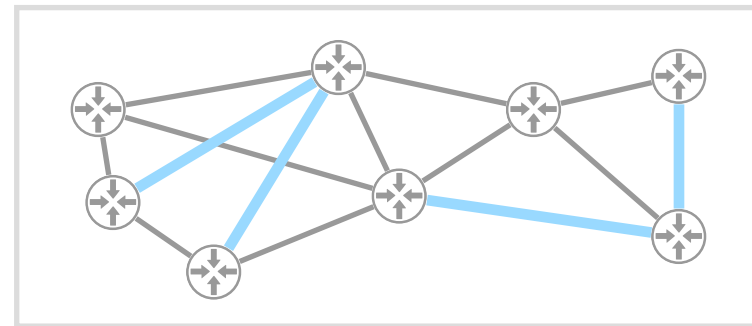
# Topology obfuscation as an optimization problem

Given the **physical topology P**,

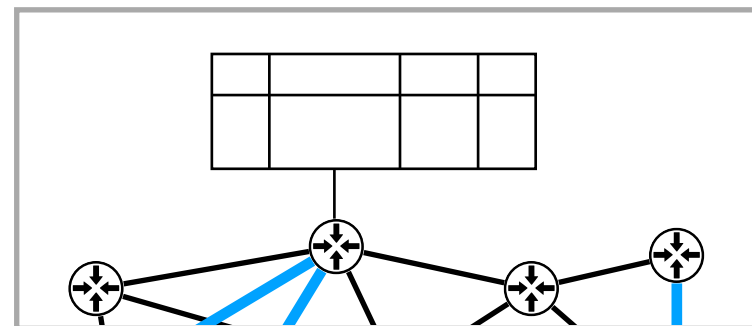
compute a **virtual topology V**, such that

- **V** is robust against link-flooding attacks
- **V** has maximal practicality

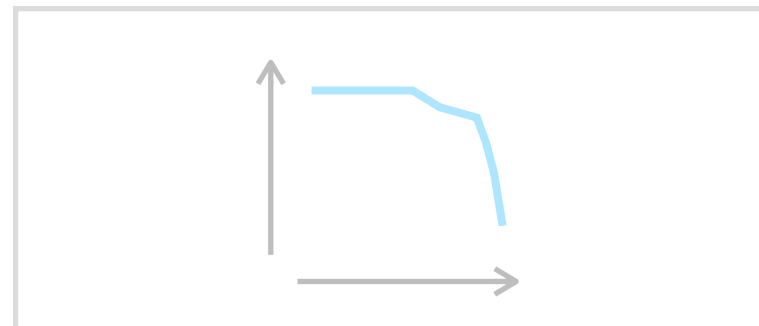
# NetHide: Secure and Practical Network Topology Obfuscation



NetHide computes a secure virtual topology that is similar to the physical topology



NetHide deploys the virtual topology using programmable networks



NetHide works for realistic topologies and maintains the utility of debugging tools

# Utility-preserving topology deployment

Deploy the **virtual topology  $V$** , such that

- debugging tools still work
- network performance is not impacted
- it scales to large networks



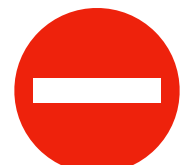
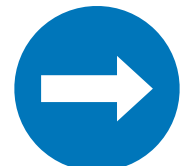


# Utility-preserving topology deployment

Deploy the **virtual topology  $V$** , such that

- debugging tools still work
- network performance is not impacted
- it scales to large networks

# Maintaining the utility of debugging tools requires sending packets through the actual network

-  Answer from a central controller
-  Answer at the edge
-  Answer in a virtual clone of the network
-  Answer from the correct device  
that appears on the path

# Utility-preserving topology deployment

Deploy the **virtual topology  $V$** , such that

- debugging tools still work
- network performance is not impacted
- it scales to large networks

# Programmable network devices allow modifying tracing packets at line rate



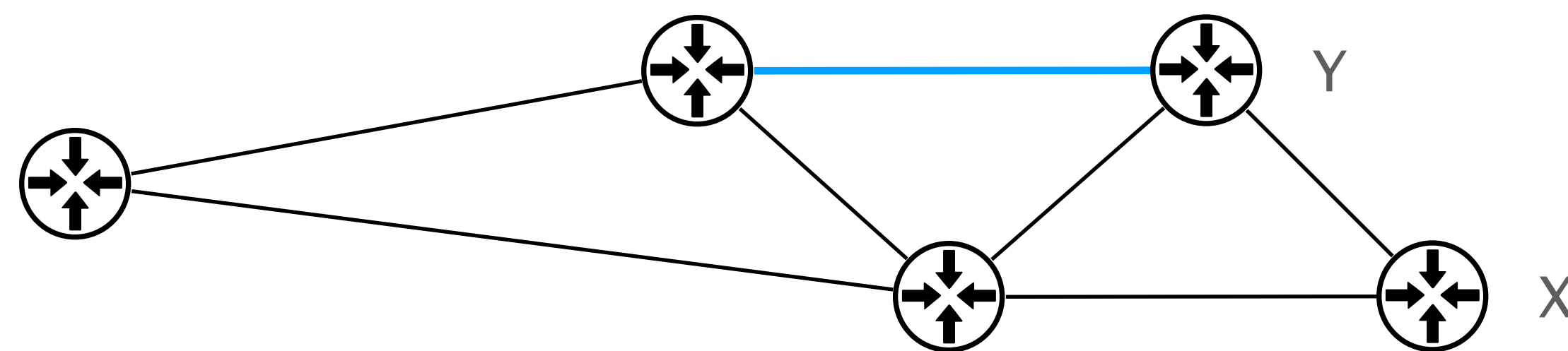
- Read & modify packet headers  
e.g. the TTL value
- Basic operations  
e.g. hash functions and checksums
- Add or remove custom headers  
to store information

Programmable network devices allow modifying **tracing packets** at line rate

- Packets with a small TTL value expire in the network
- Packets with different path lengths in **P** and **V** need to increase or decrease TTL

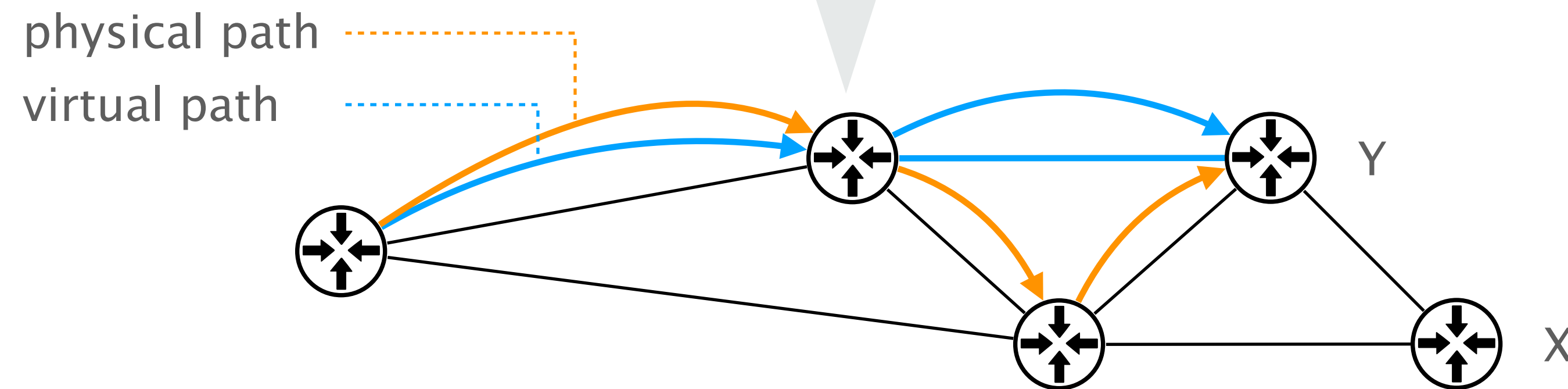
# Programmable network devices are configured through match+action tables

*If I receive a packet to X with TTL = i,  
I should send it to Y with TTL = j*



# Programmable network devices are configured through match+action tables

original		new	
dst	TTL	dst	TTL
X	1	Y	2



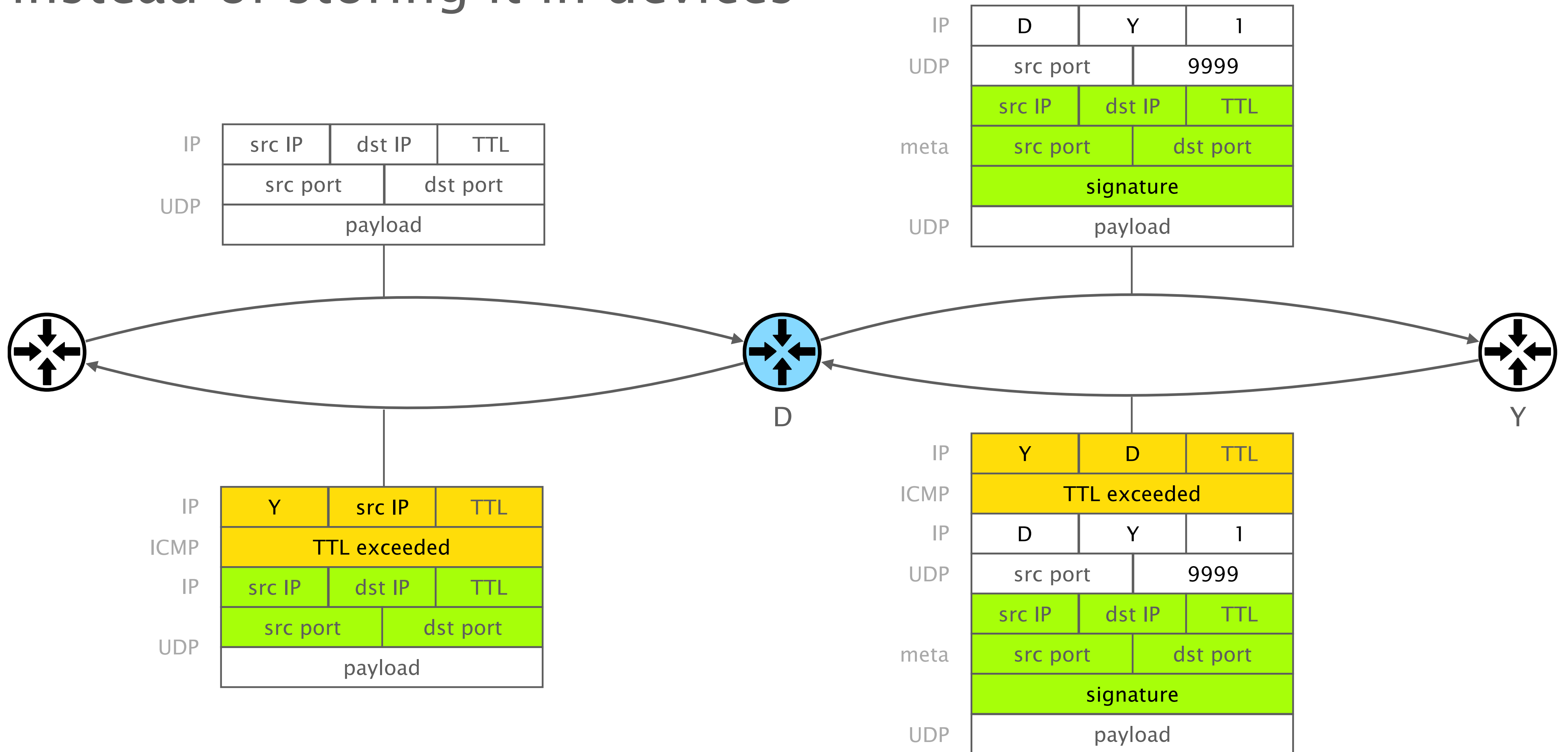
# Utility-preserving topology deployment

Deploy the **virtual topology  $V$** , such that

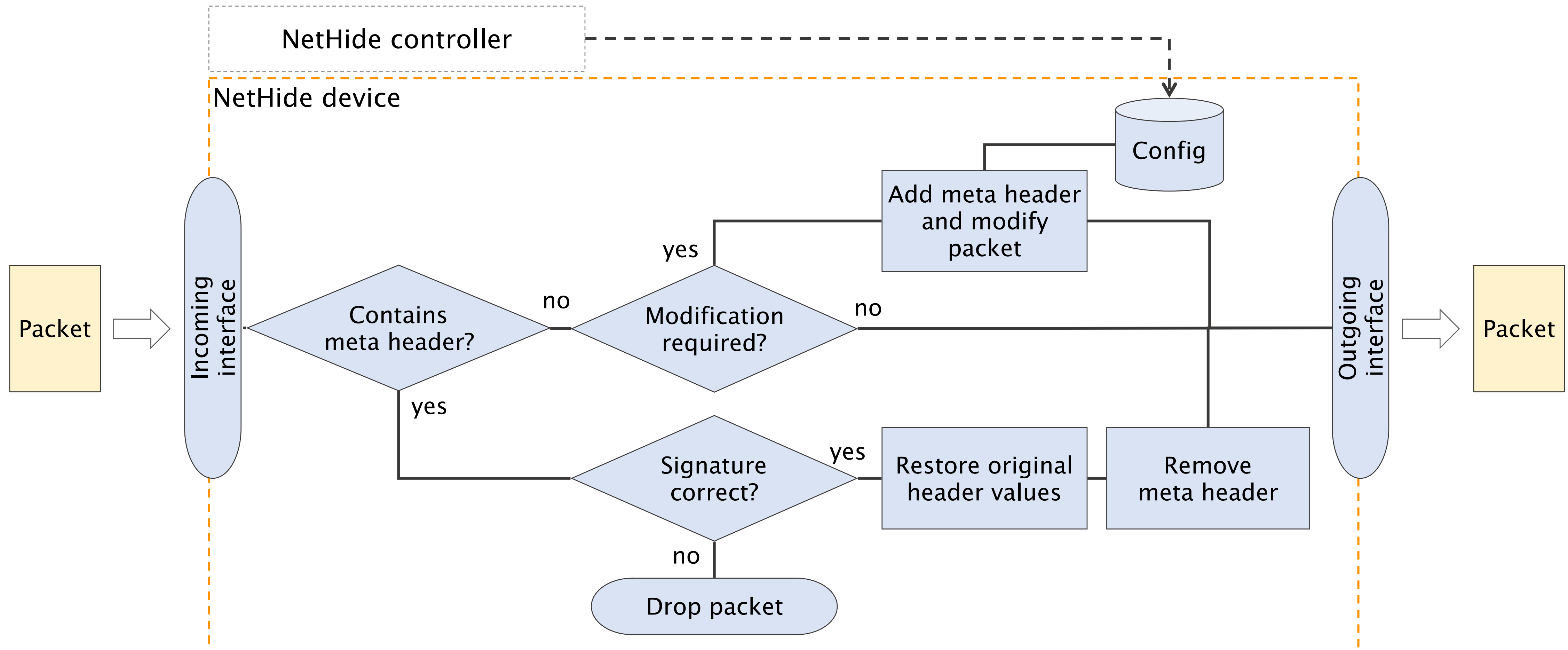
- debugging tools still work
- network performance is not impacted
- it scales to large networks



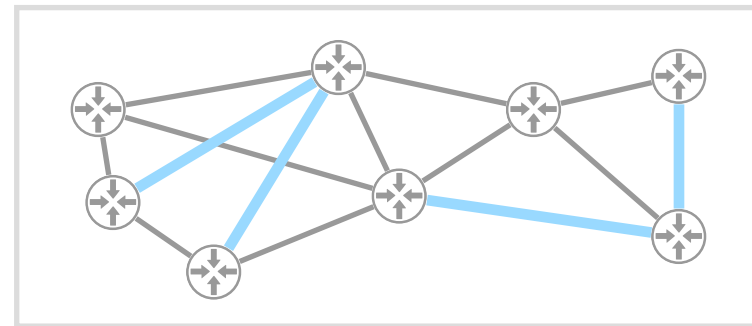
# Encoding state in packets instead of storing it in devices



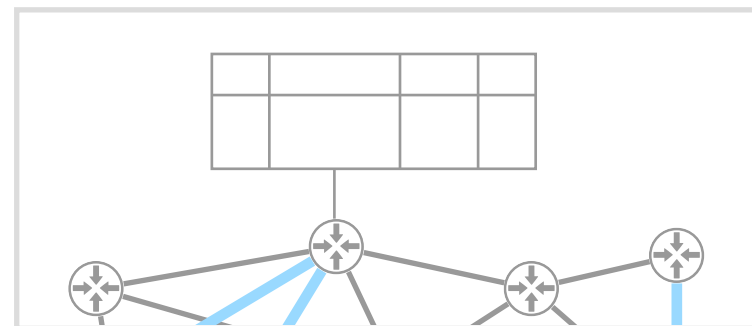
# P4 program architecture



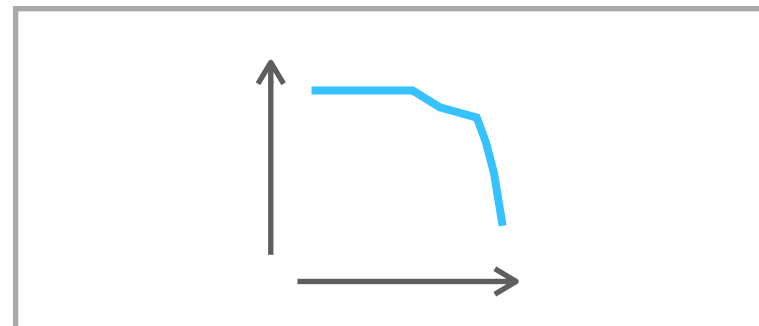
# NetHide: Secure and Practical Network Topology Obfuscation



NetHide computes a secure virtual topology that is similar to the physical topology



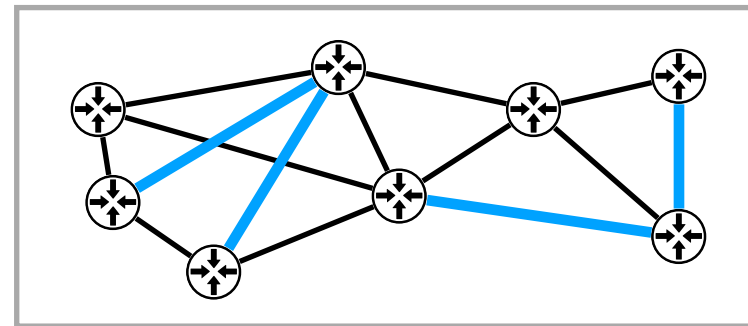
NetHide deploys the virtual topology using programmable networks



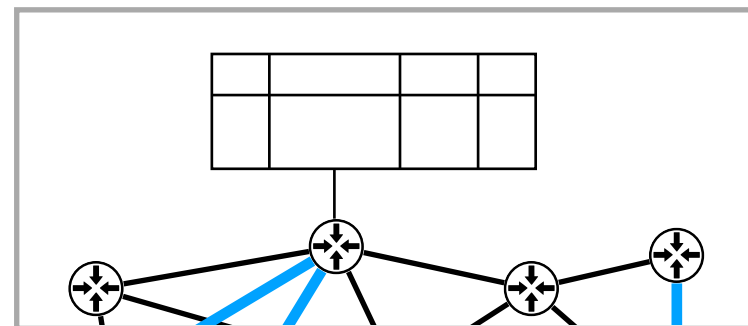
NetHide works for realistic topologies and maintains the utility of debugging tools

# NetHide: Secure and Practical Network Topology Obfuscation

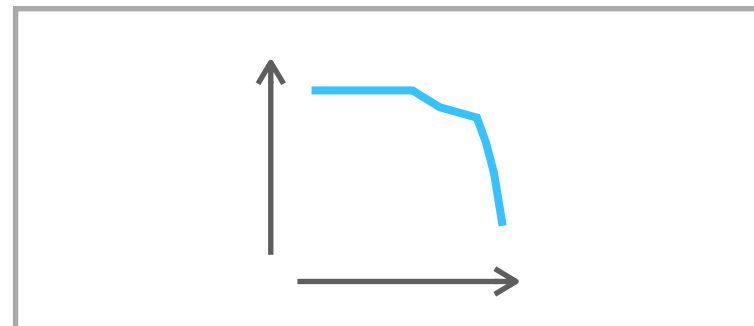
[nethide.ethz.ch](http://nethide.ethz.ch)



NetHide computes a secure virtual topology that is similar to the physical topology



NetHide deploys the virtual topology using programmable networks



NetHide works for realistic topologies and maintains the utility of debugging tools