TOR:

The Onion Router

COS 561
11/09/2017

Yixin Sun

Internet communications are *not* anonymous

Five-tuple: (srcip, srcport, dstip, dstport, protocol)

Looking at an Internet communication, one can

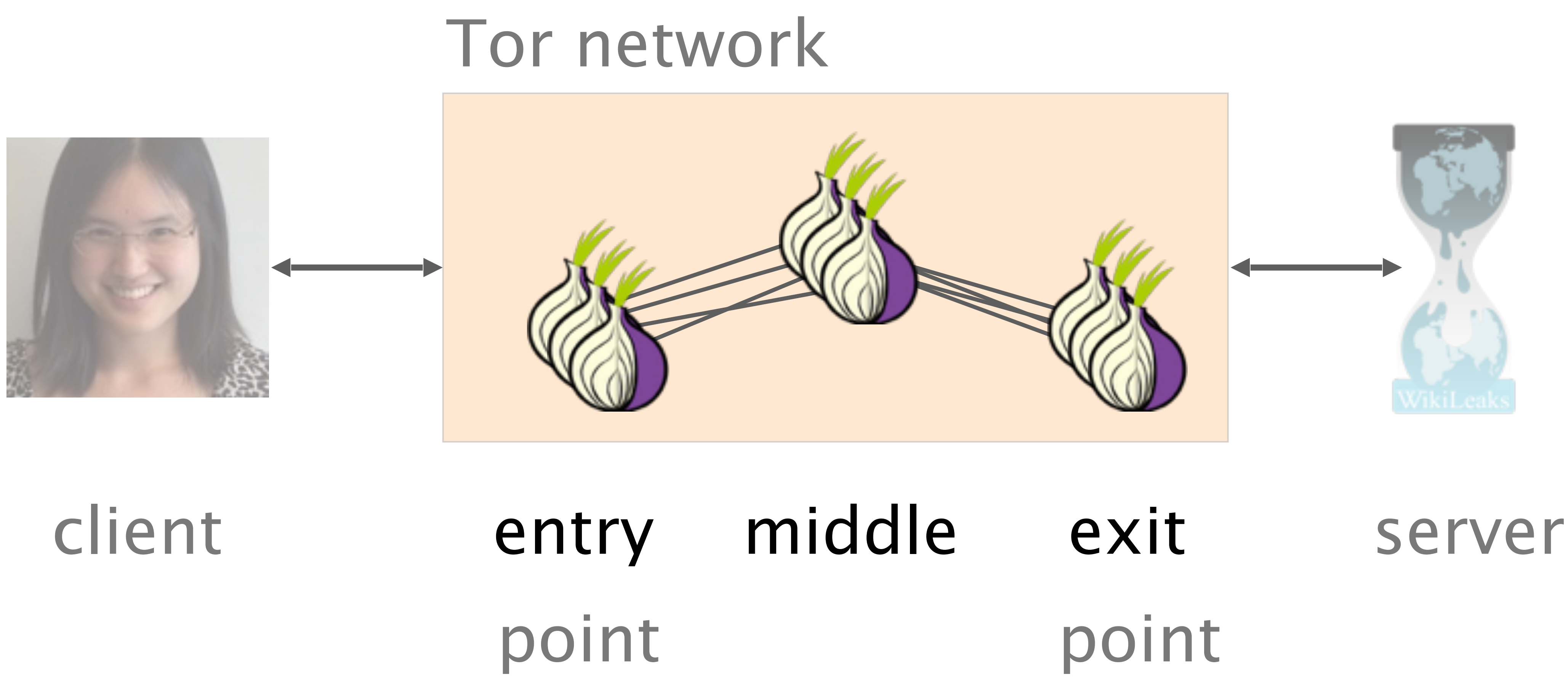
- infer who is talking to whom
- infer physical locations
- use that to track behavior and interests

even if the communication is encrypted

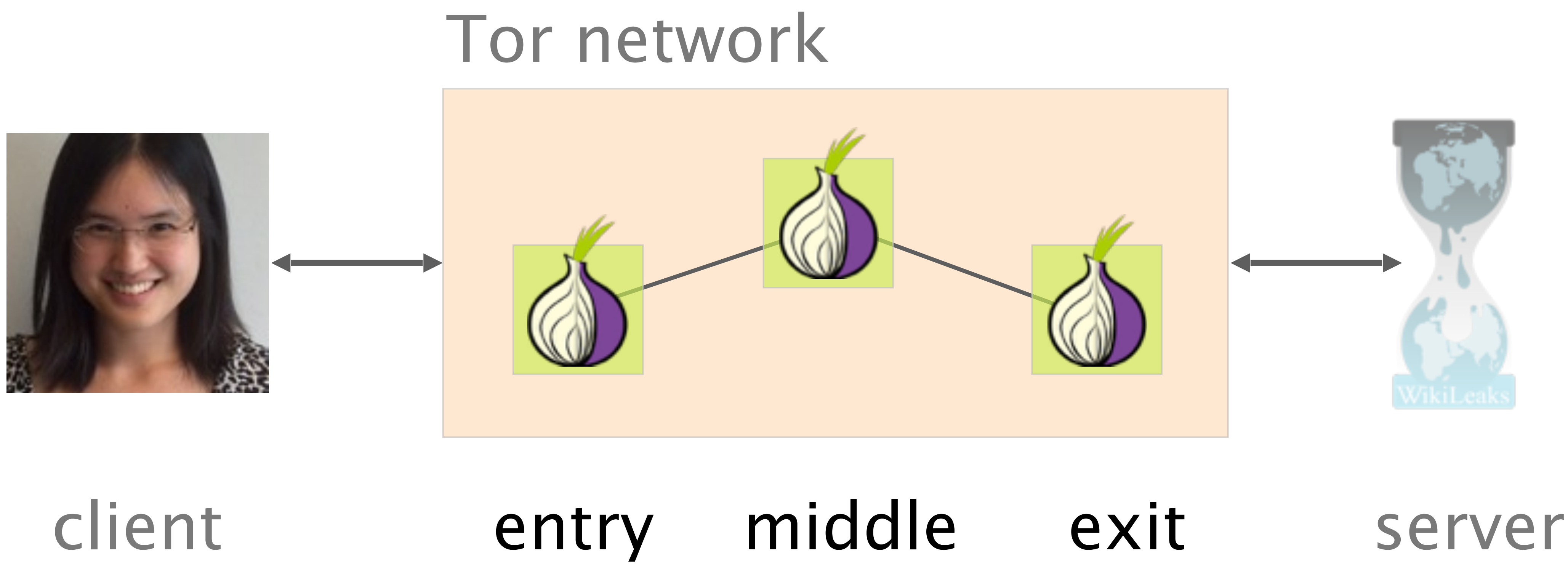
Tor aims at preventing adversaries to follow packets between a sender and a receiver

To do that,

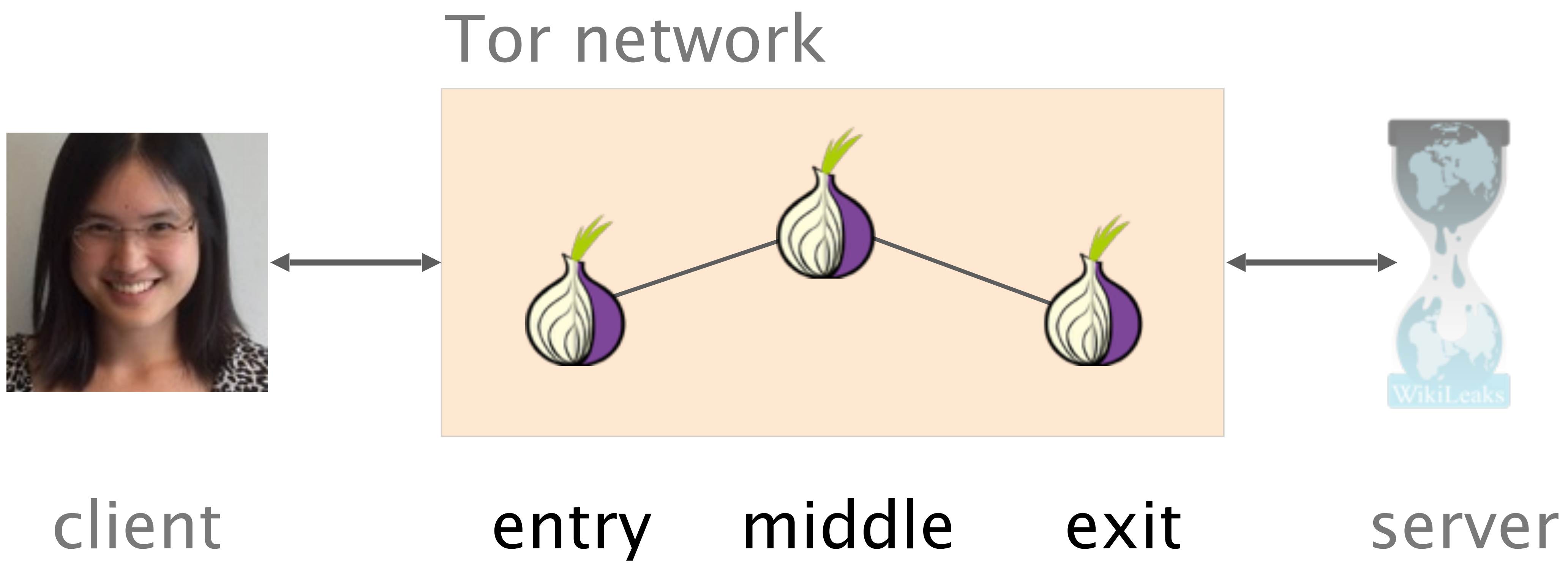
Tor bounces traffic around a network of relays

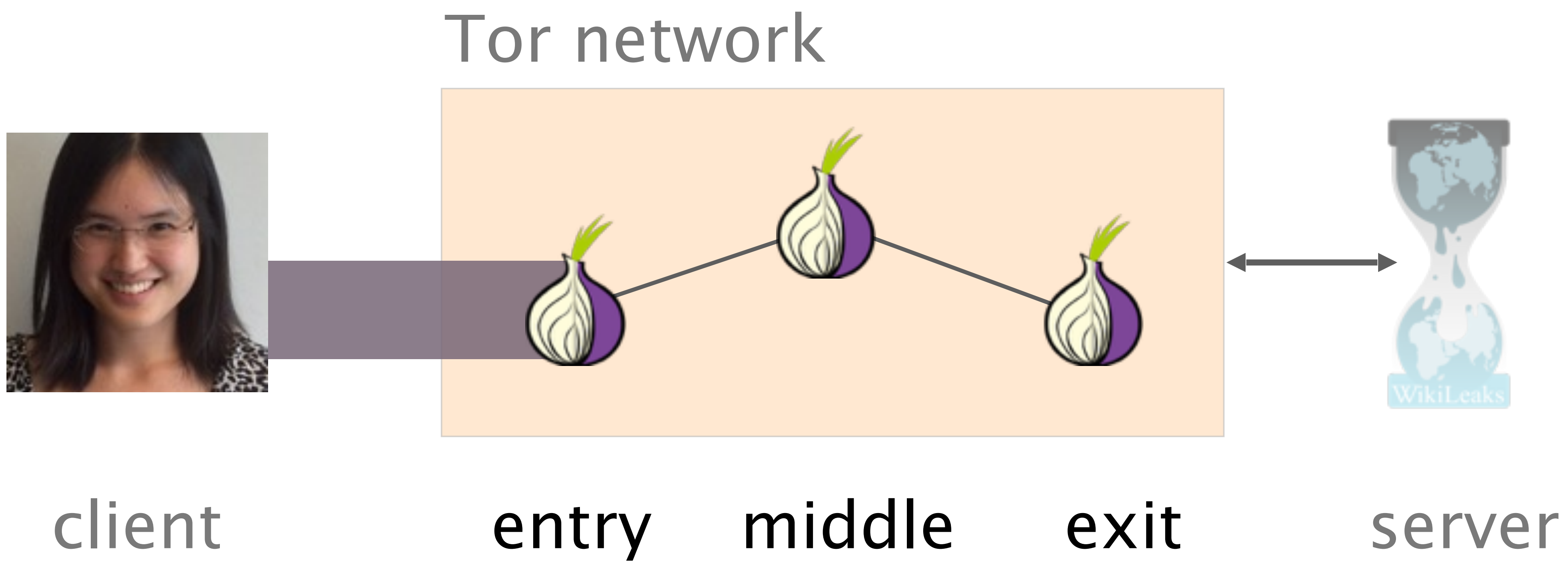


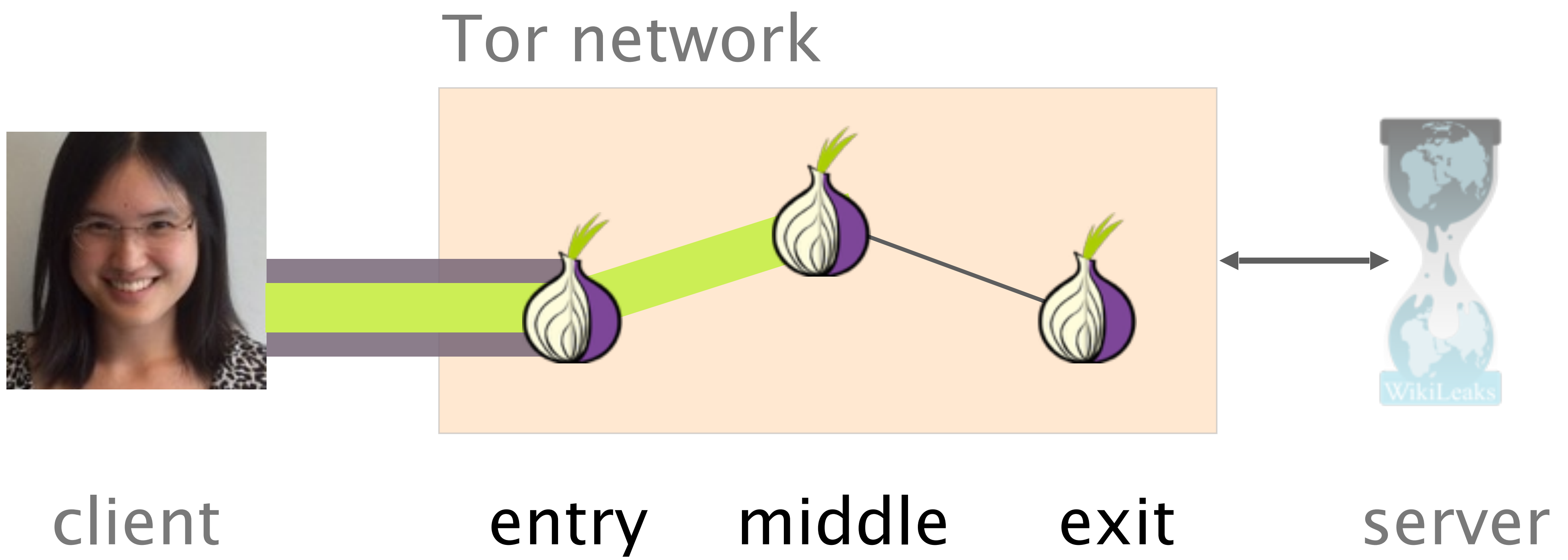
Tor clients start by selecting 3 relays, one of each type

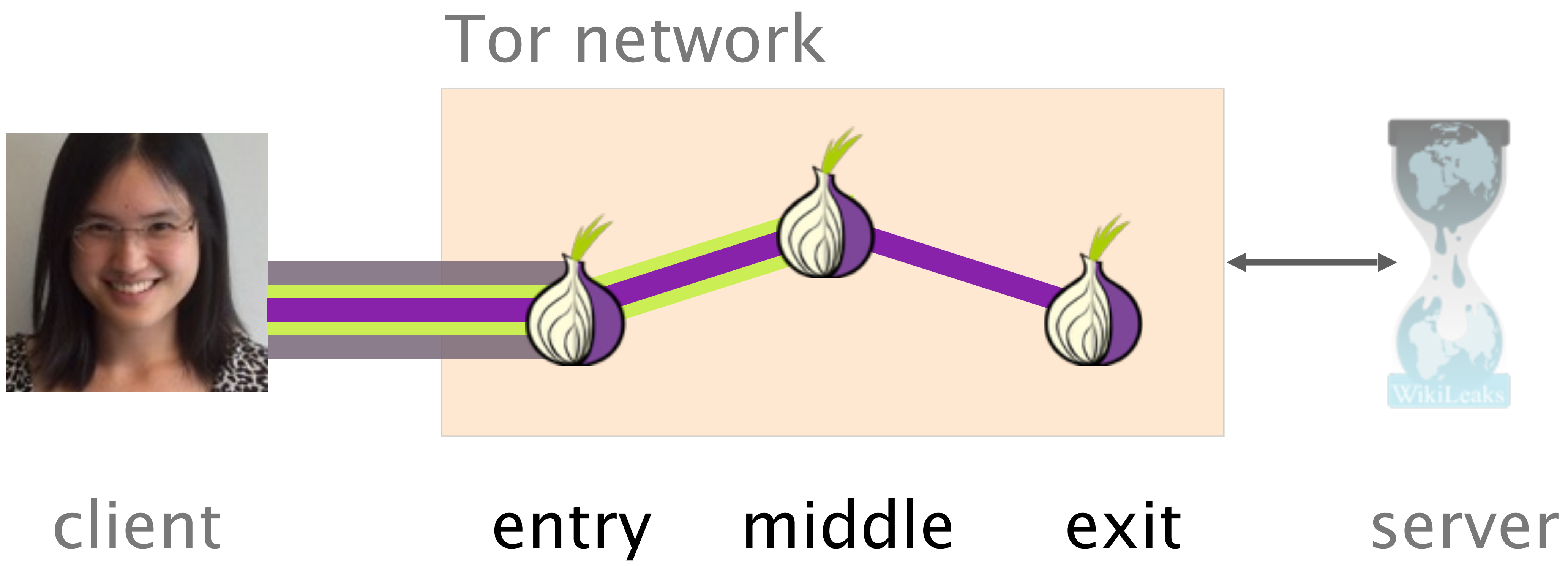


Tor clients then incrementally build encrypted circuits through them

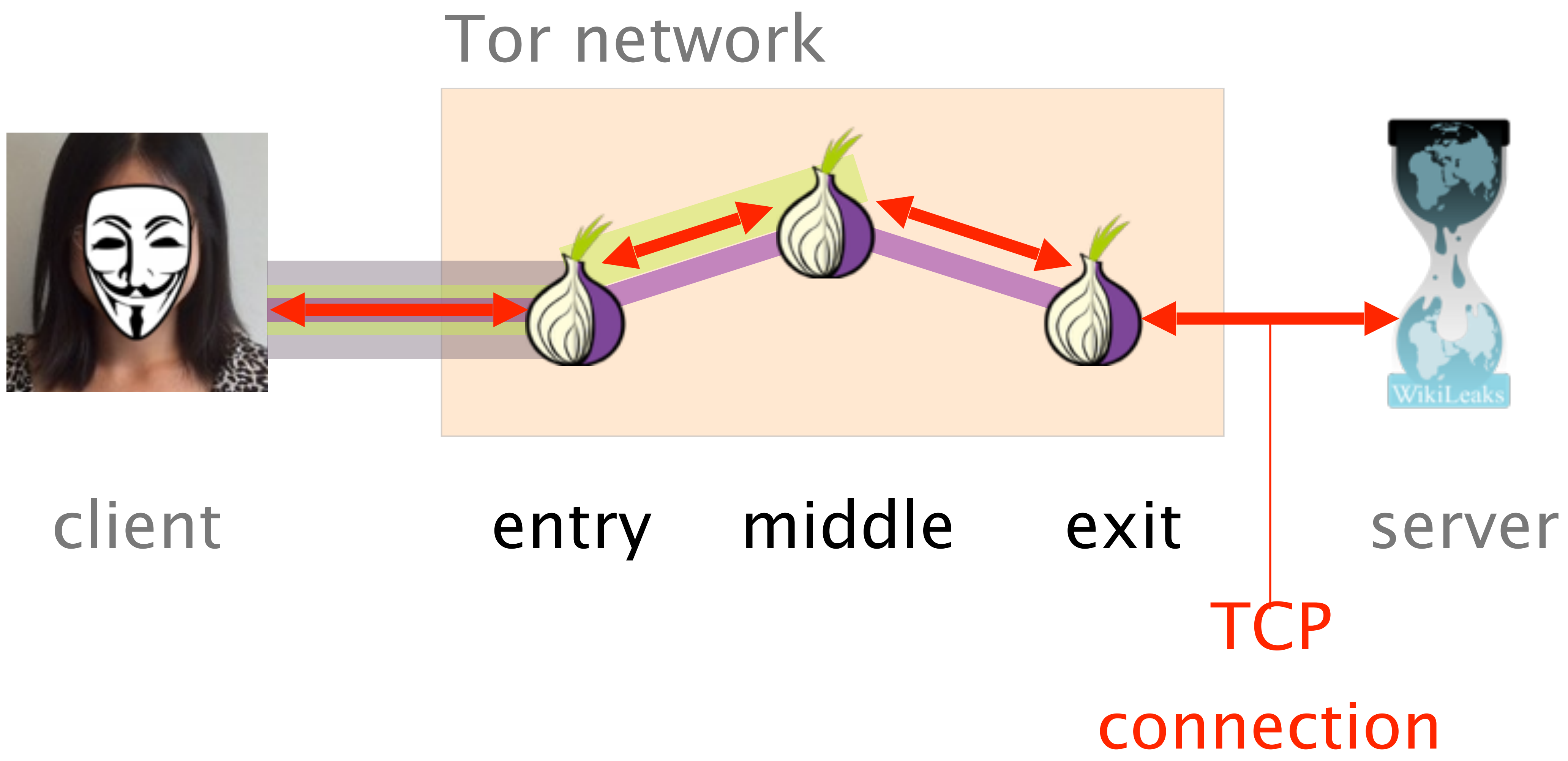



Tor network
WikiLeaks
client
entry
middle
exit
server

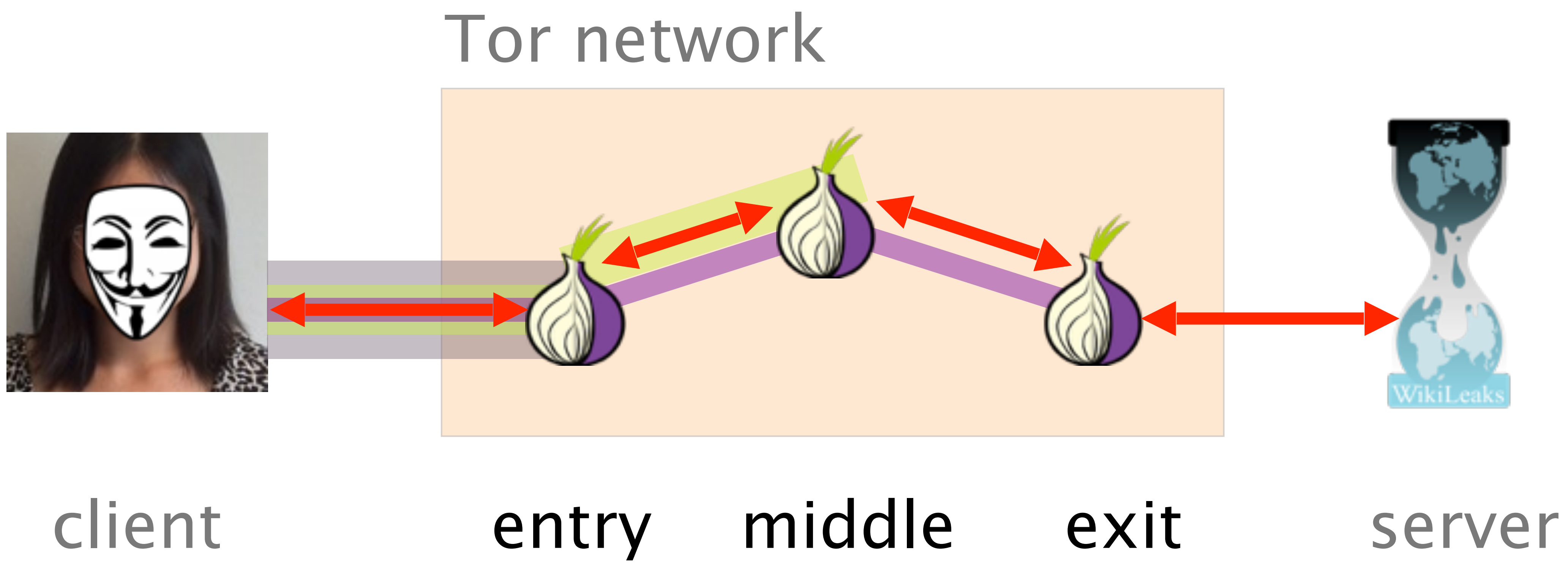

Tor network
WikiLeaks
client
entry
middle
exit
server

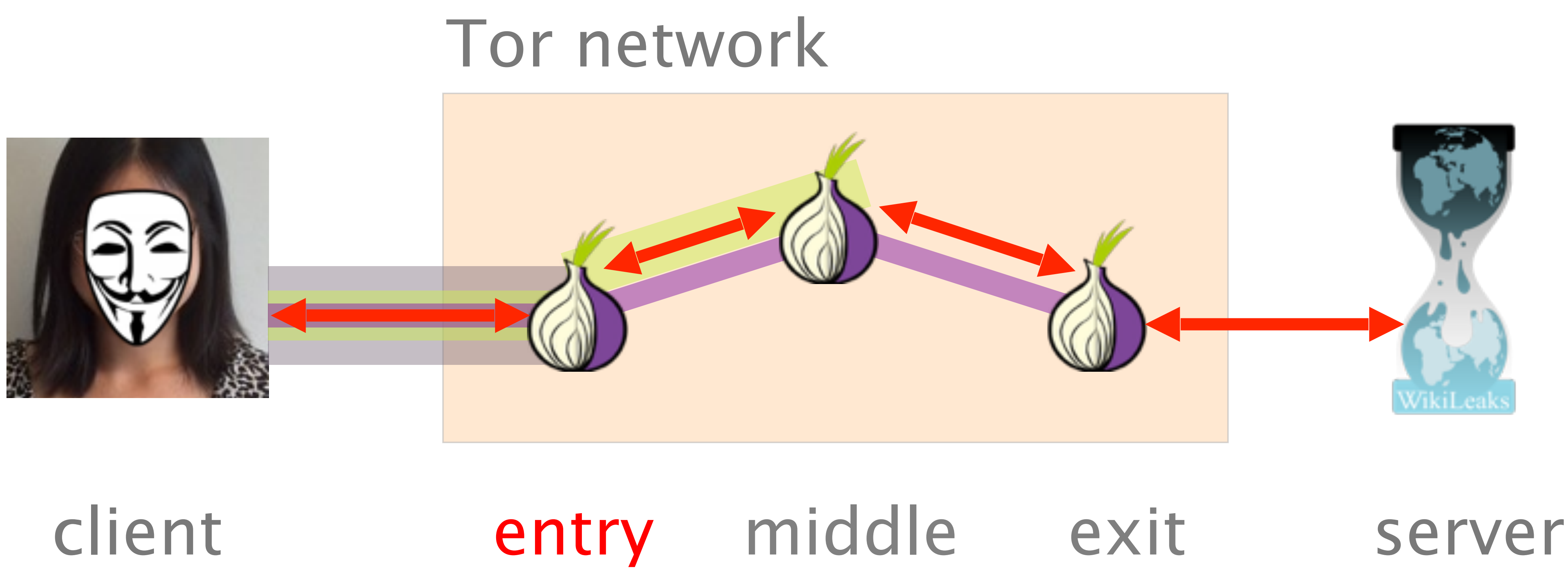

Tor network
WikiLeaks
client
entry
middle
exit
server

Anonymous communication takes place by forwarding across consecutive tunnels



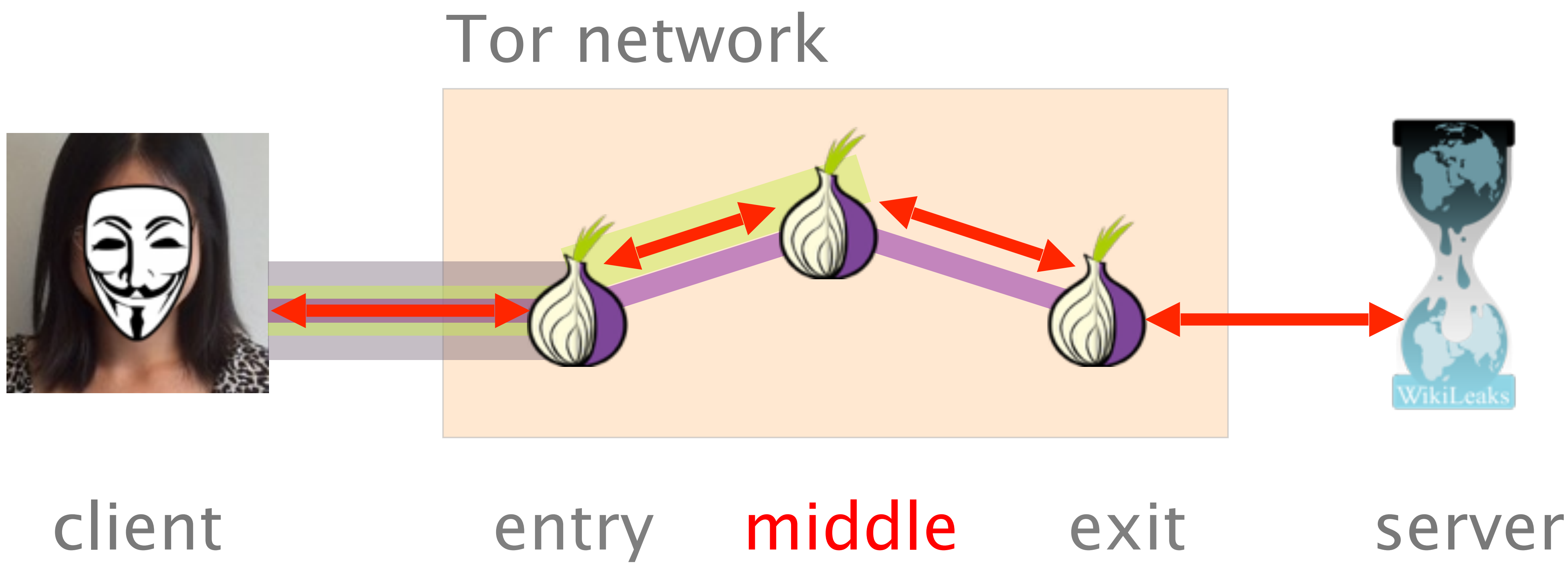
Not a single Tor entity knows the association (client, server)





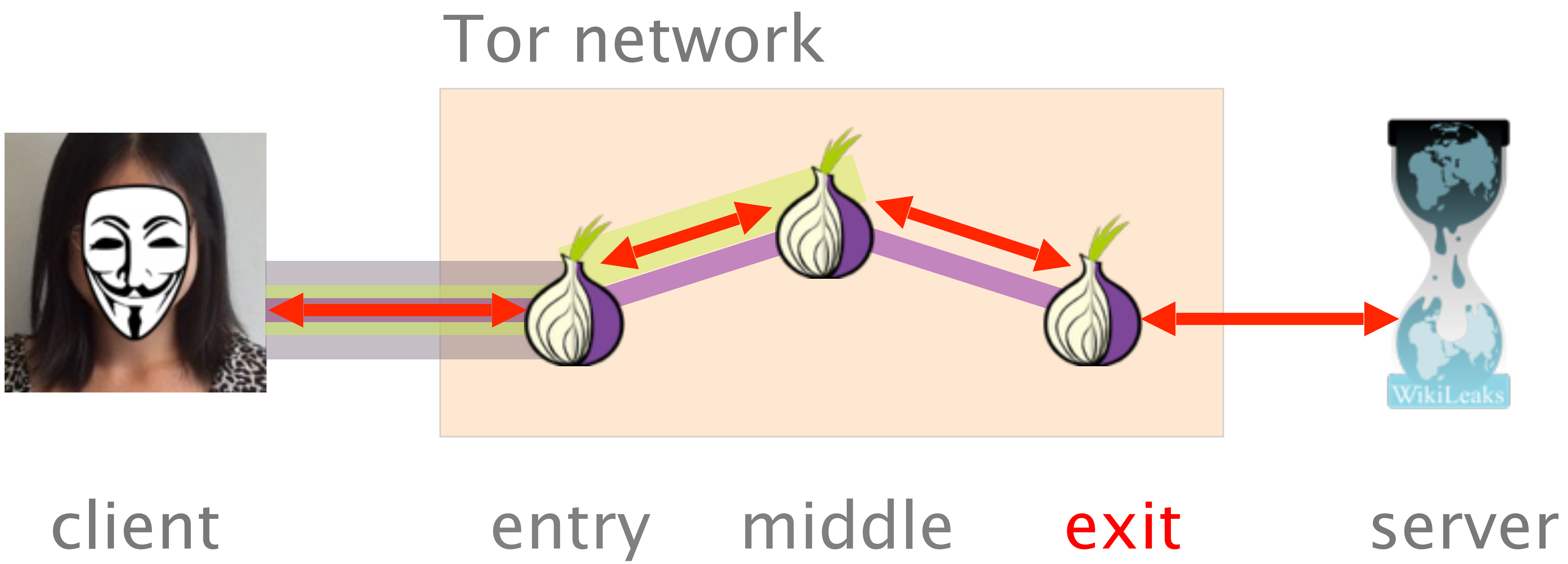
knows the source,

not the destination



knows neither the source,

nor the destination

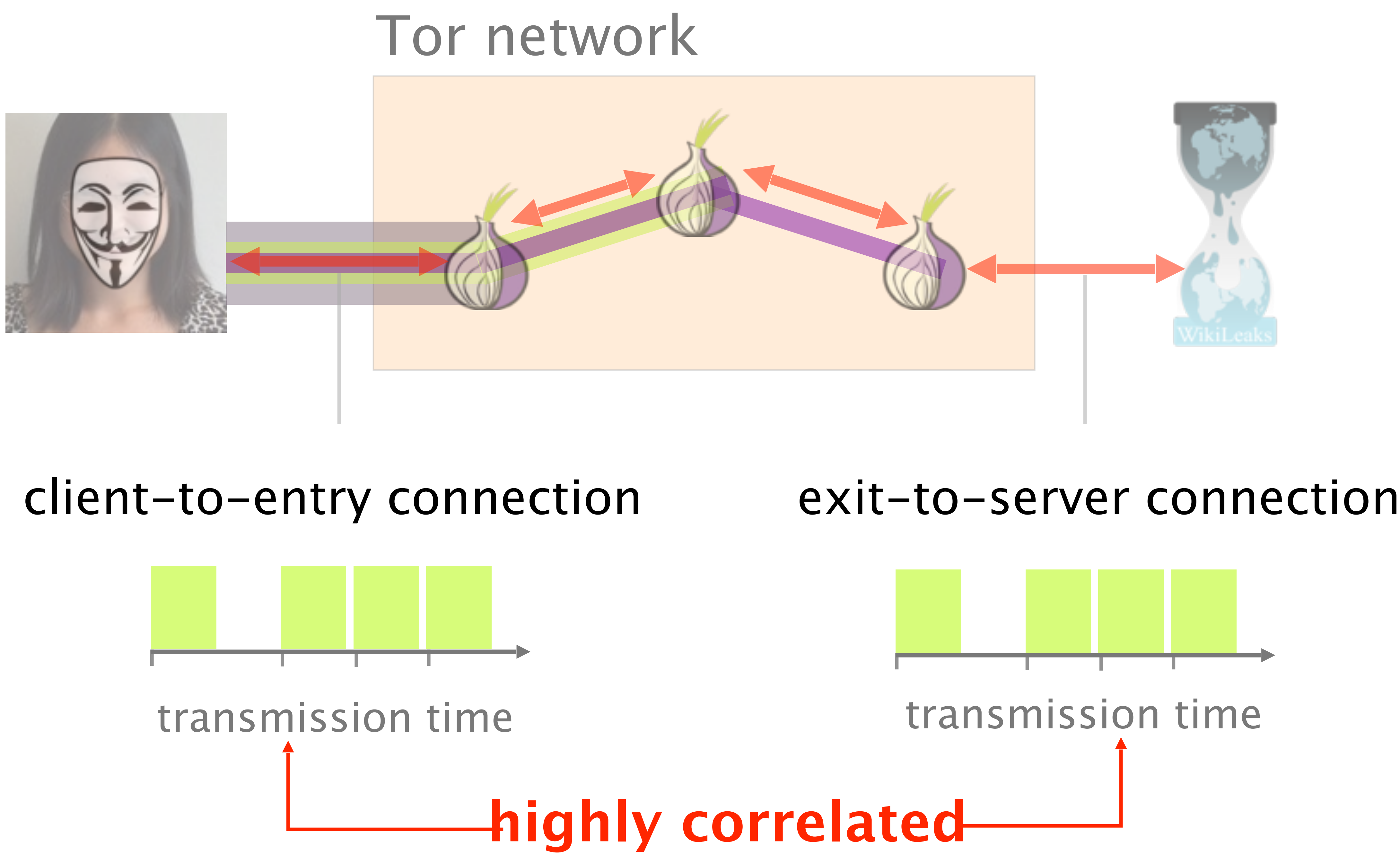


knows the destination,

not the source

However, Tor is known to be vulnerable to traffic correlation analysis

Traffic entering and leaving Tor is highly correlated



Traffic correlation attacks require to see client-to-entry and exit-to-server traffic

Traffic correlation attacks require to see client-to-entry and exit-to-server traffic

How?

Two ways

Manipulate Tor
malicious relays

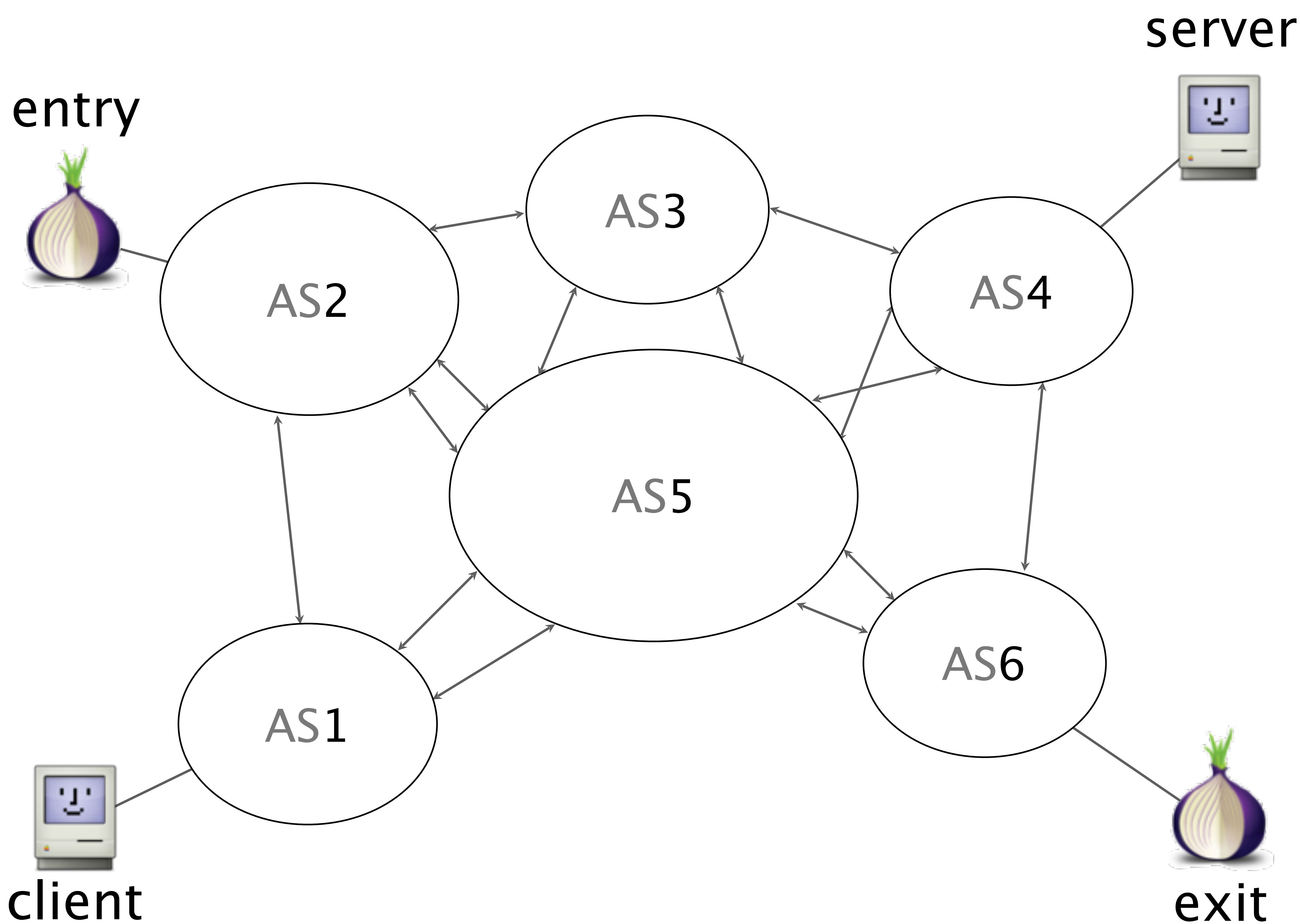
Manipulate routing
malicious networks

Two ways

Manipulate Tor
malicious relays

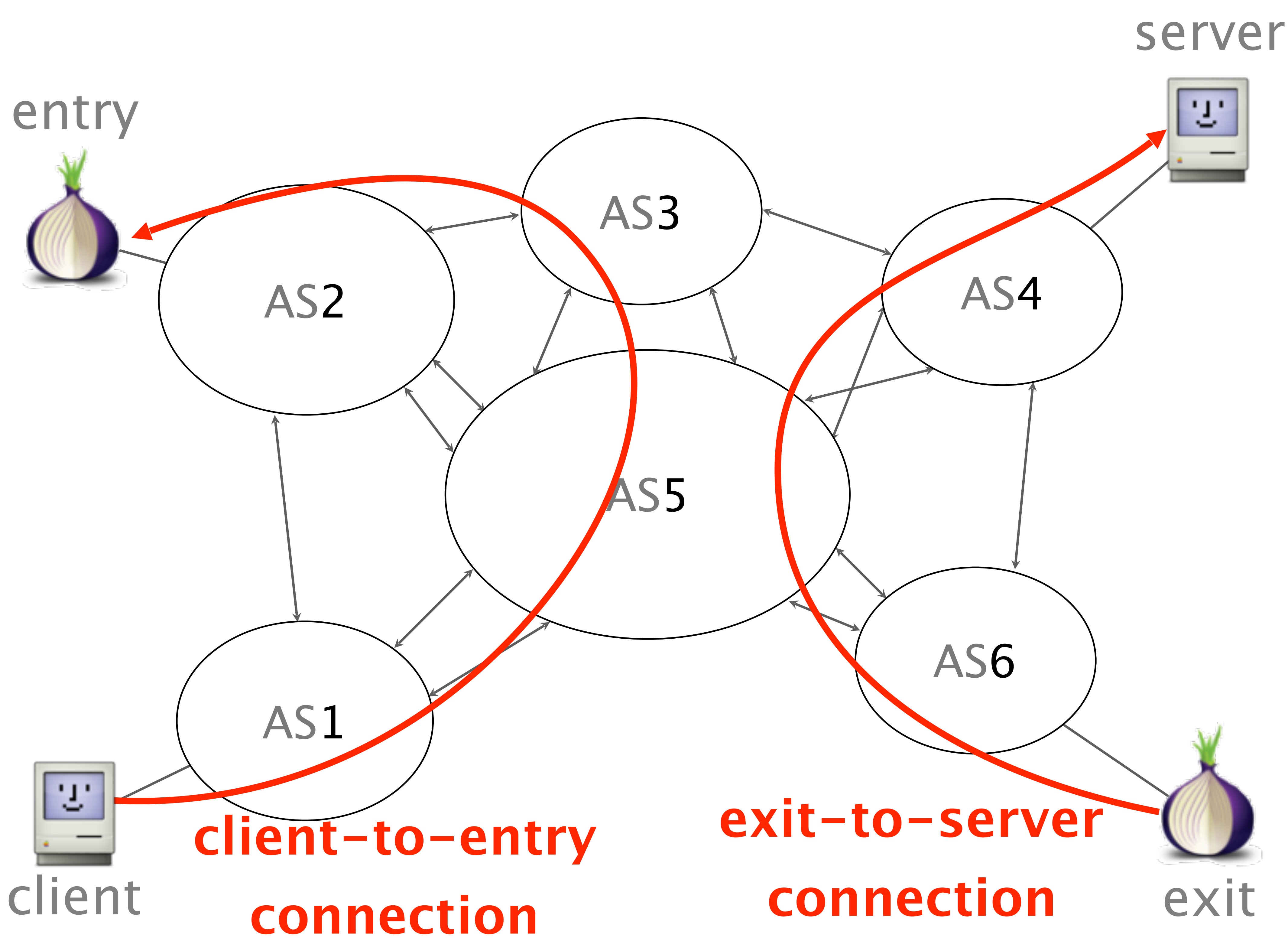
Manipulate routing
malicious networks

We'll talk about this

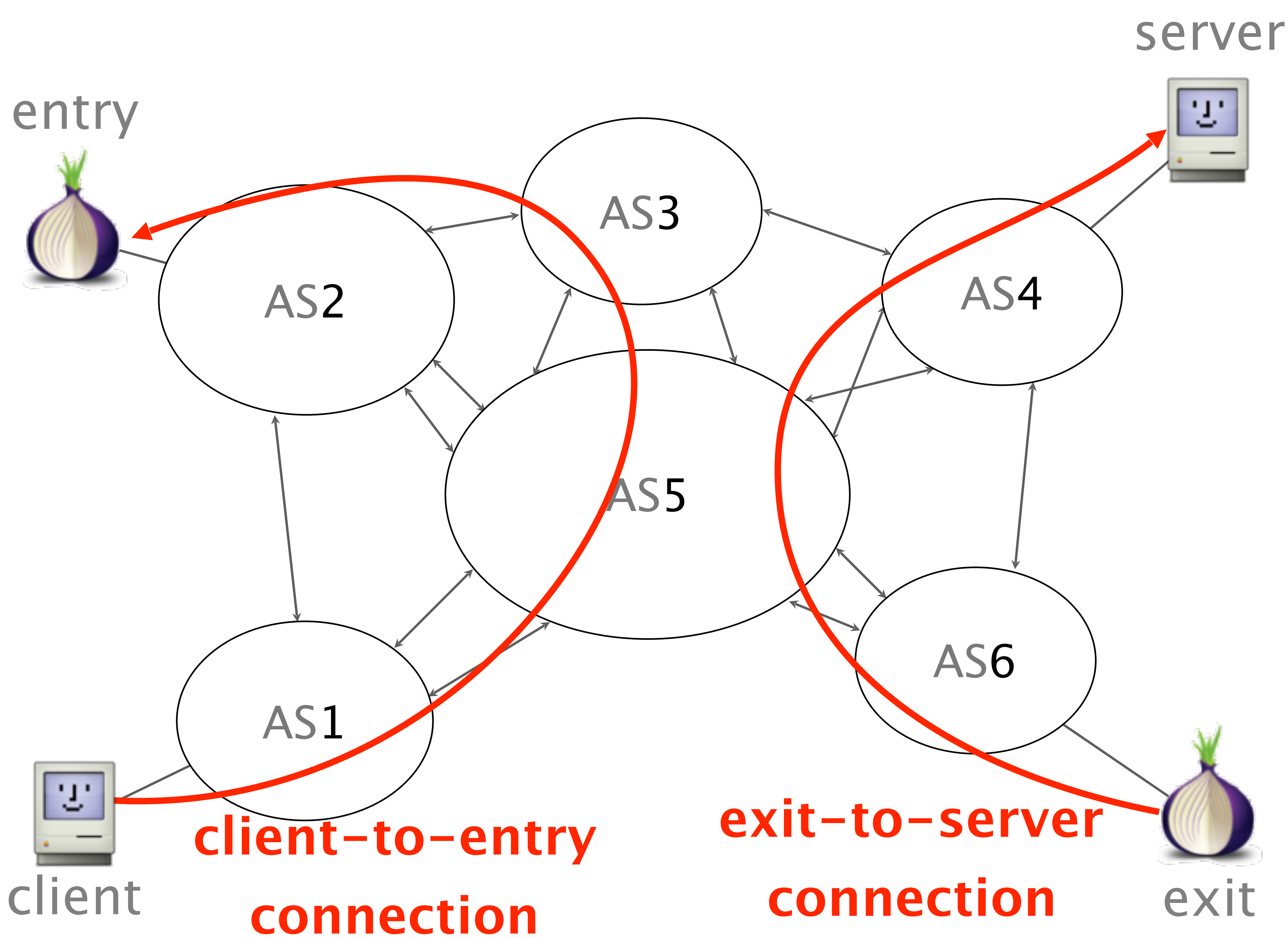


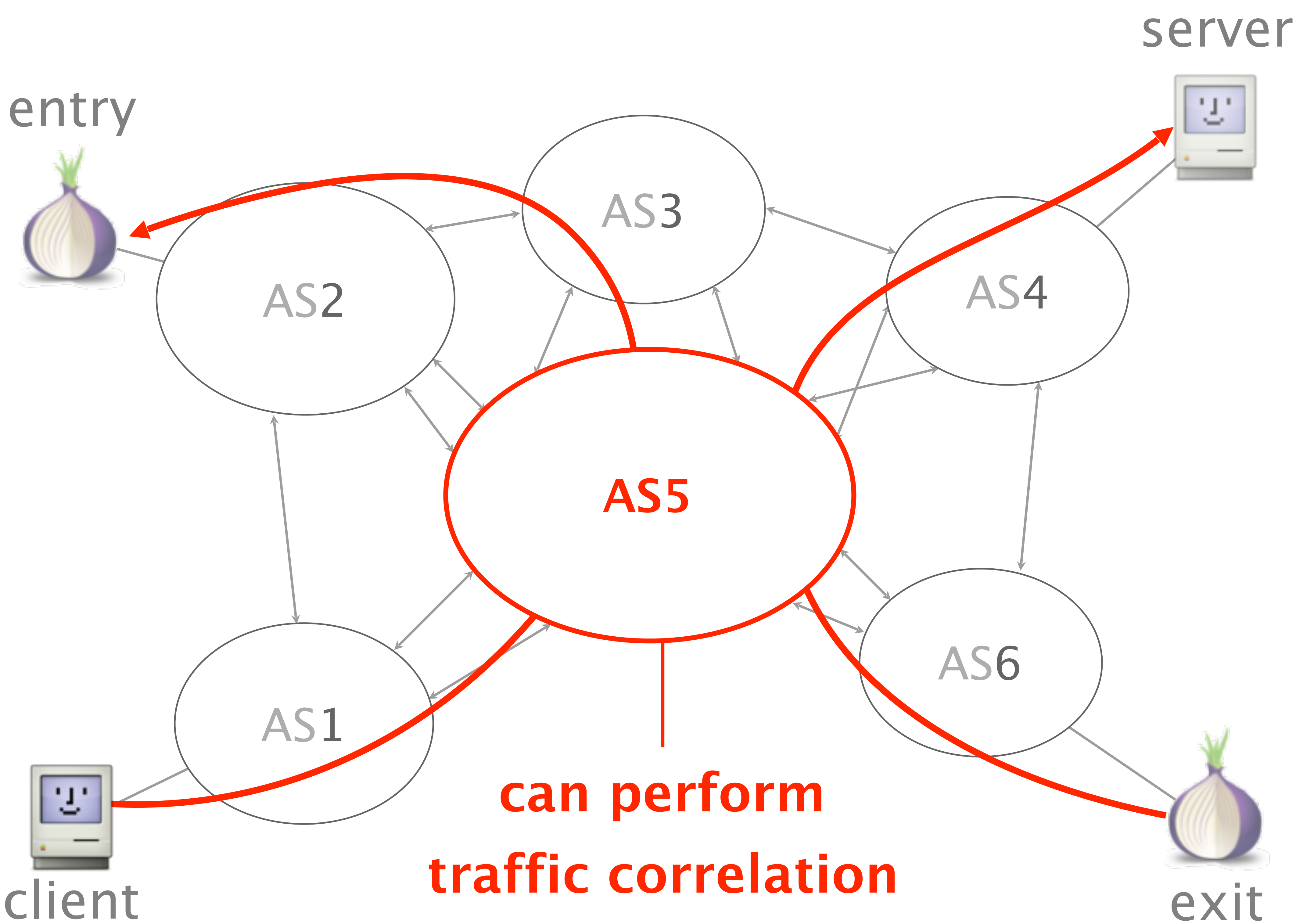
server
entry
AS3
AS2
AS4
AS5
AS1
AS6
client
exit

Tor connections get routed according to BGP



Traffic correlation attacks require to see client-to-entry *and* exit-to-server traffic




server
entry
AS3
AS2
AS4
AS5
AS6
AS1
can perform
traffic correlation
client
exit

User anonymity decreases over time due to BGP dynamics

Asymmetric routing

path from A to B != from B to A

Natural BGP convergence

policy changes, failures, etc.

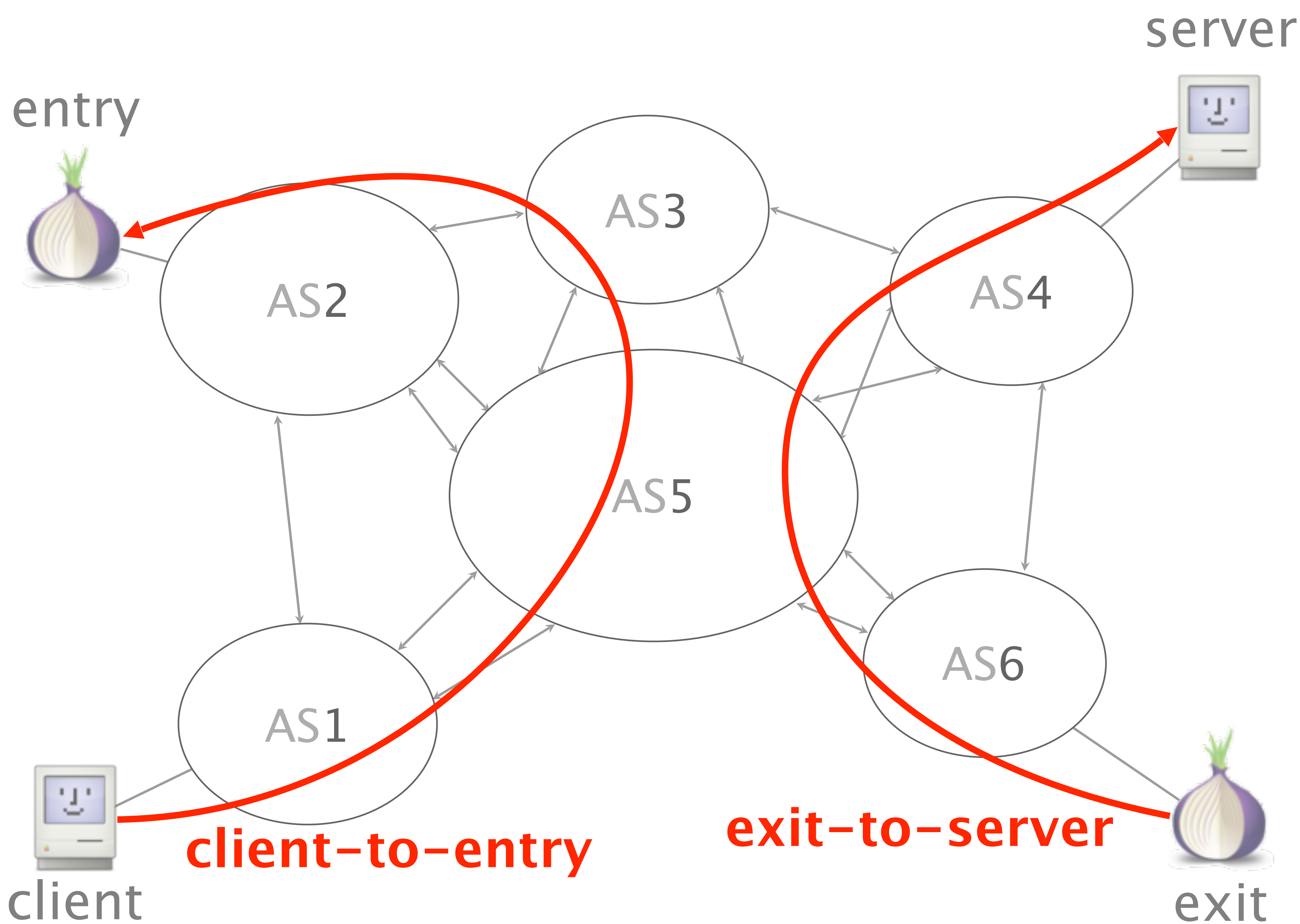
Active BGP manipulation

IP prefix hijack, interception (MITM)…

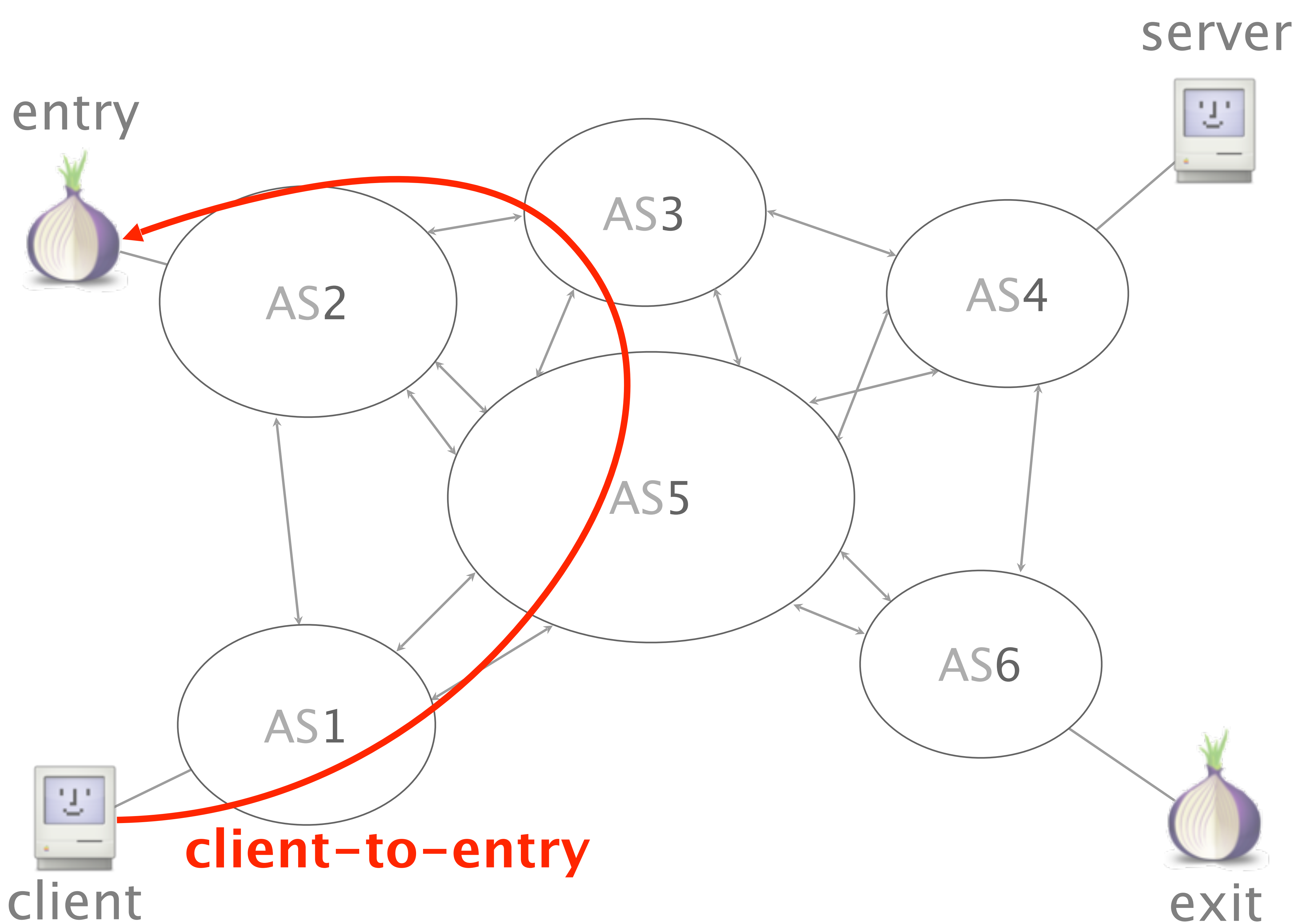
#1.

Asymmetric routing increases the numbers of AS-level adversaries

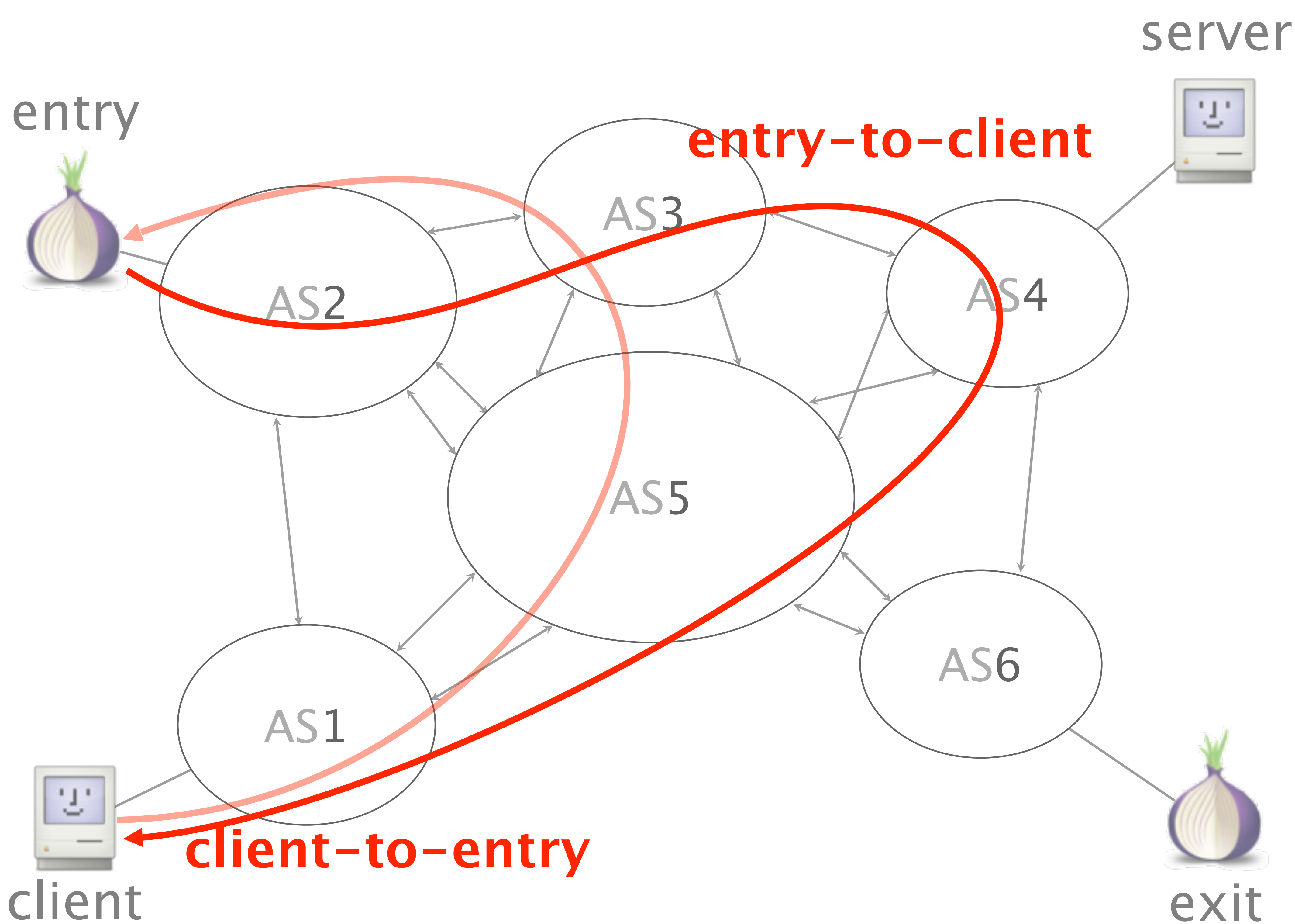
So far, we have considered one side of Tor traffic: client-to-entry and exit-to-server



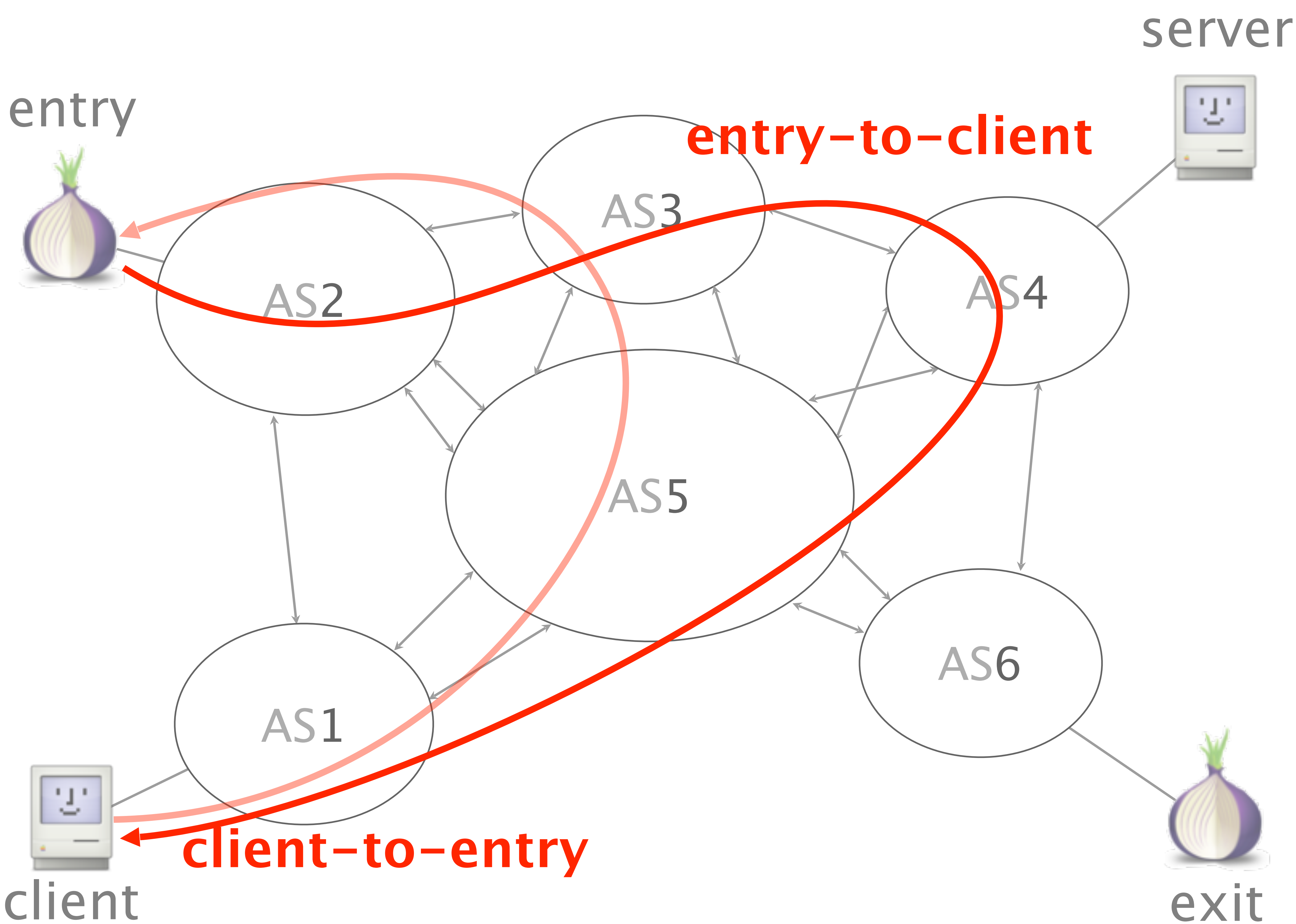
However, because of policies, routing is often *asymmetric*



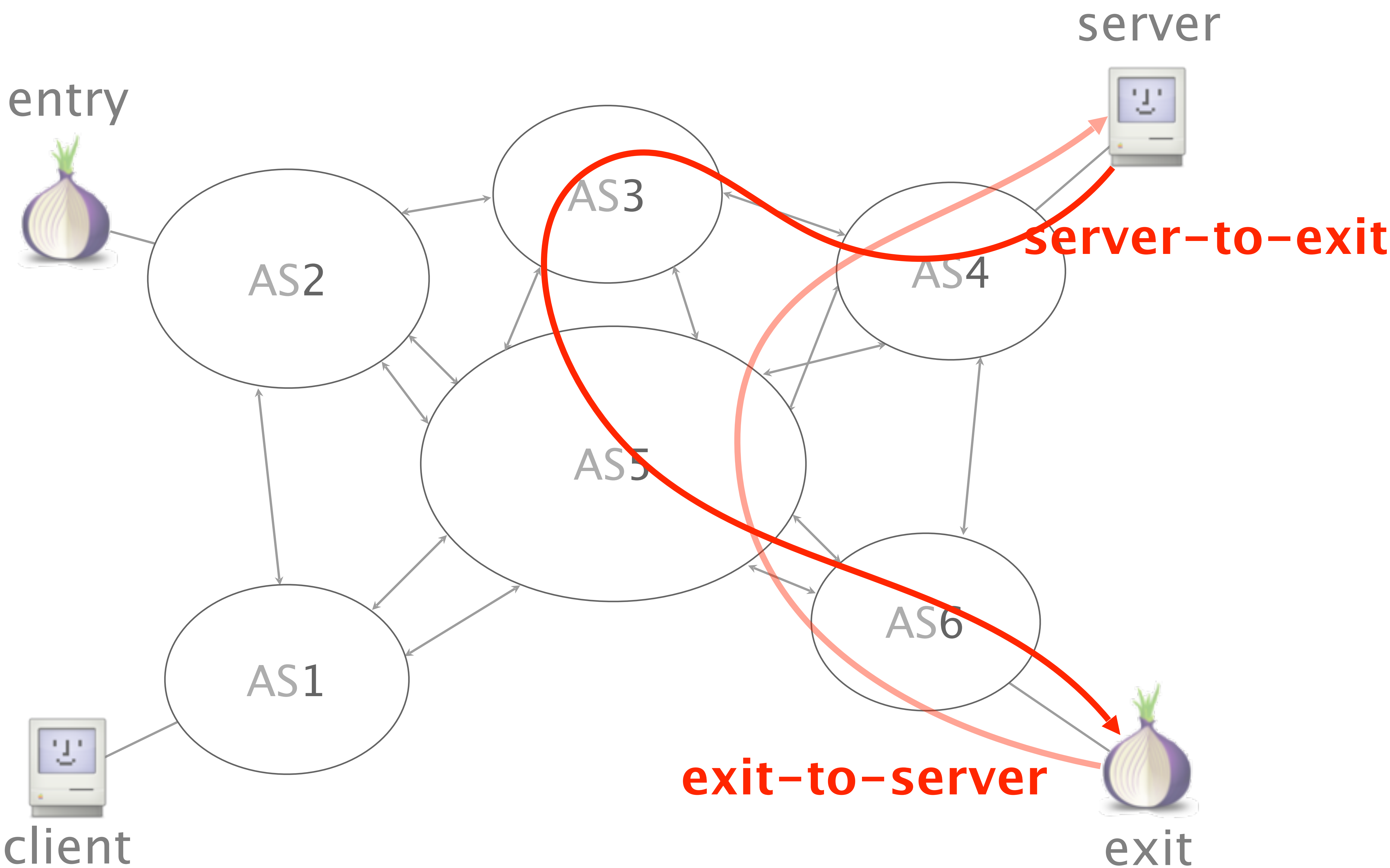
However, because of policies, routing is often *asymmetric*



While AS4 does not see client-to-entry traffic, it sees entry-to-client traffic



The same applies to server-to-exit traffic



In terms of timing properties,
both sides of a TCP connection are
highly correlated

In terms of timing properties, both sides of a TCP connection are highly correlated

When collecting TCP timing information,

seeing one direction is almost equivalent to seeing two directions

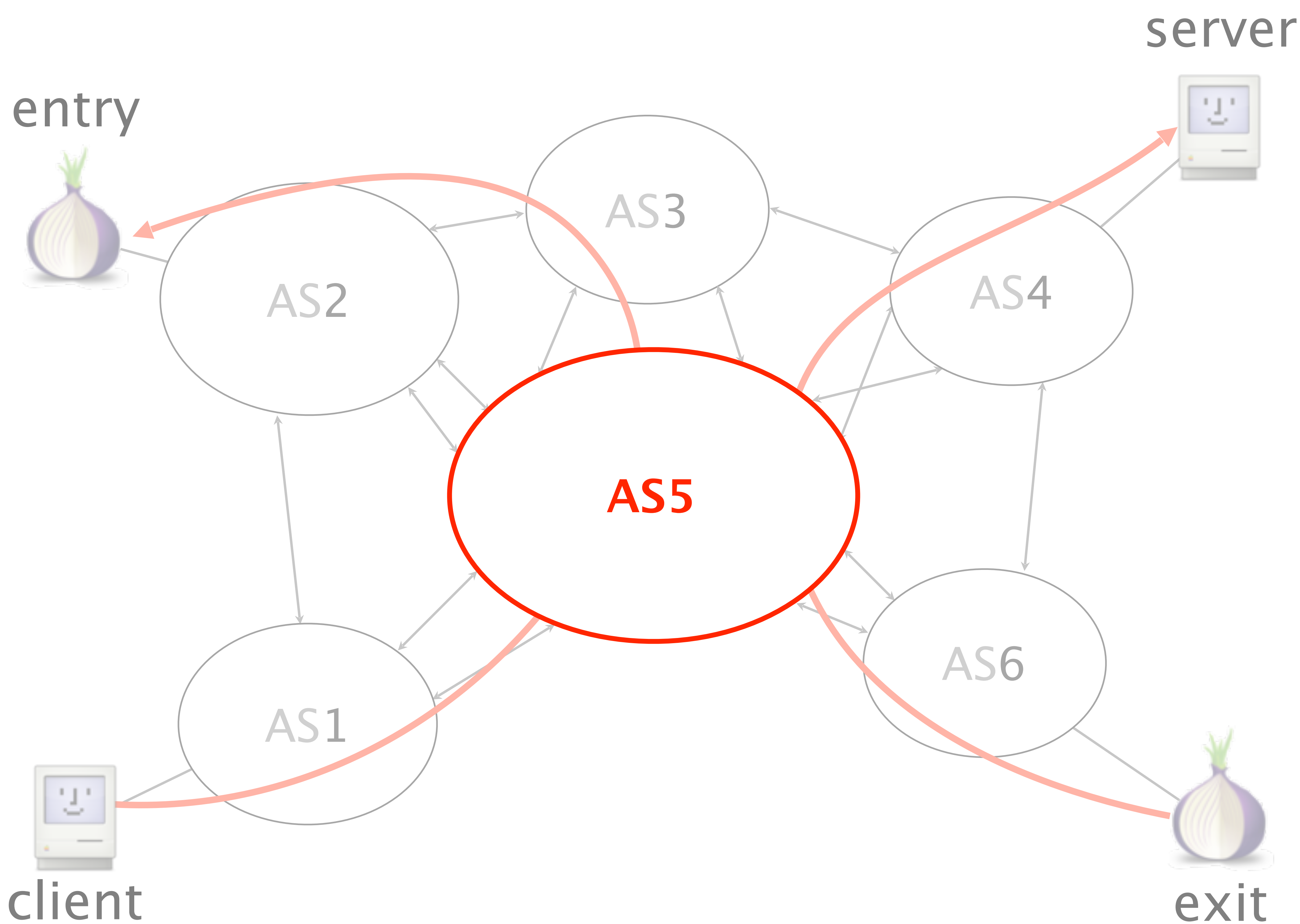
(*e.g.*, data packets)

Seq: 8282, ACK: 392

(ACKs & data packets)

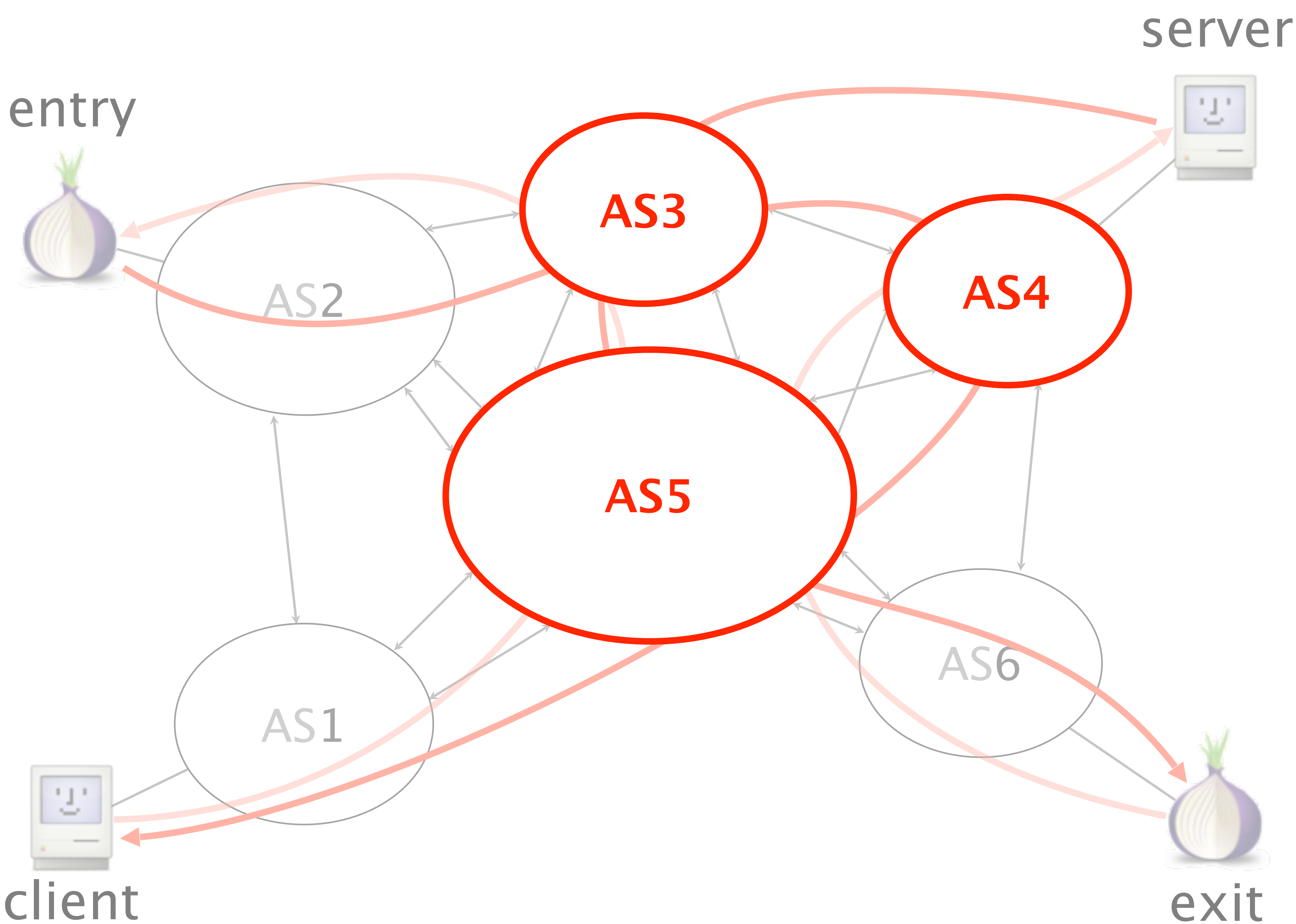
Seq: 392, ACK: 8282

Considering only one direction, only AS5 is potentially compromising



Considering both directions,

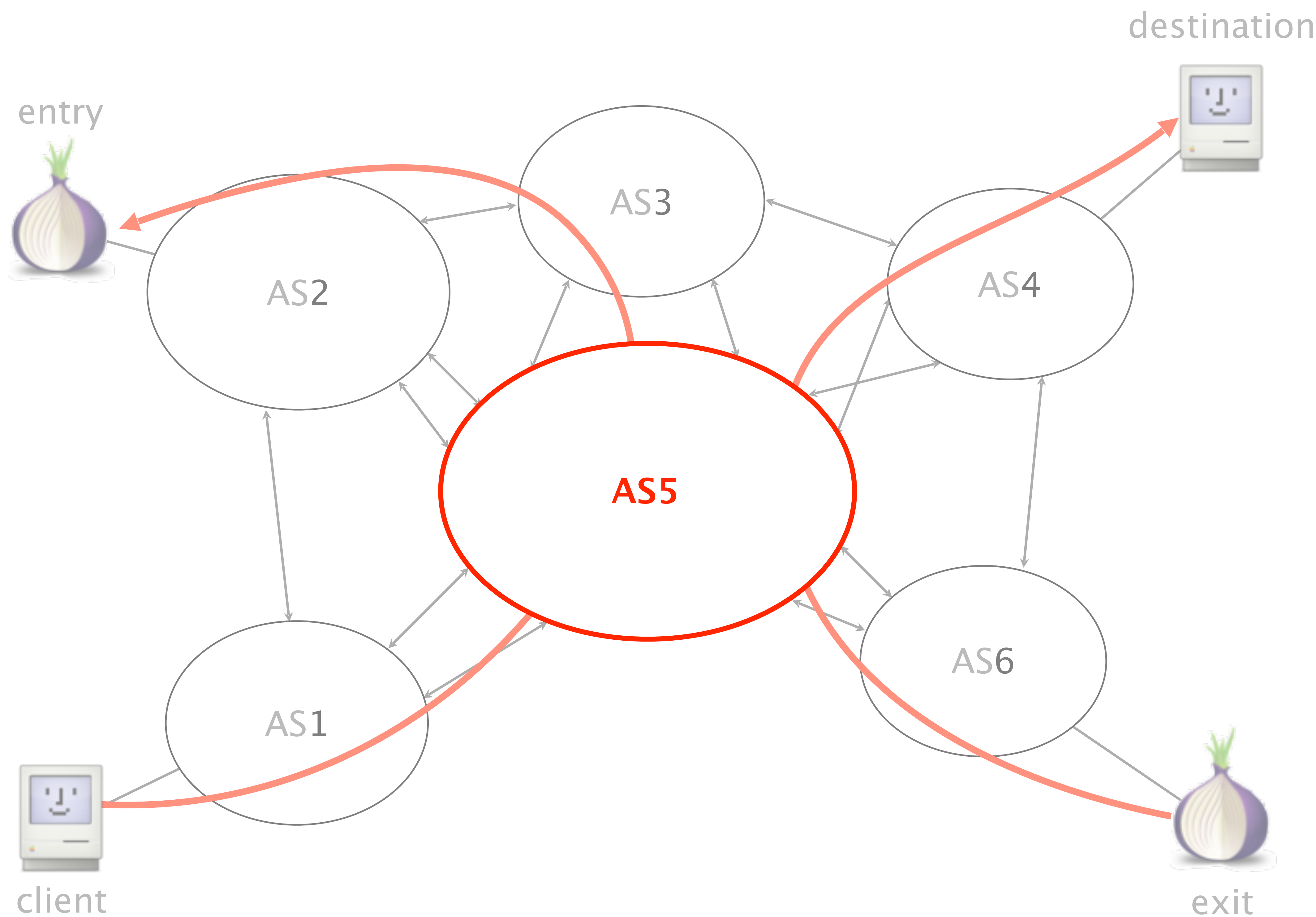
AS3, AS4 and AS5 are potentially compromising



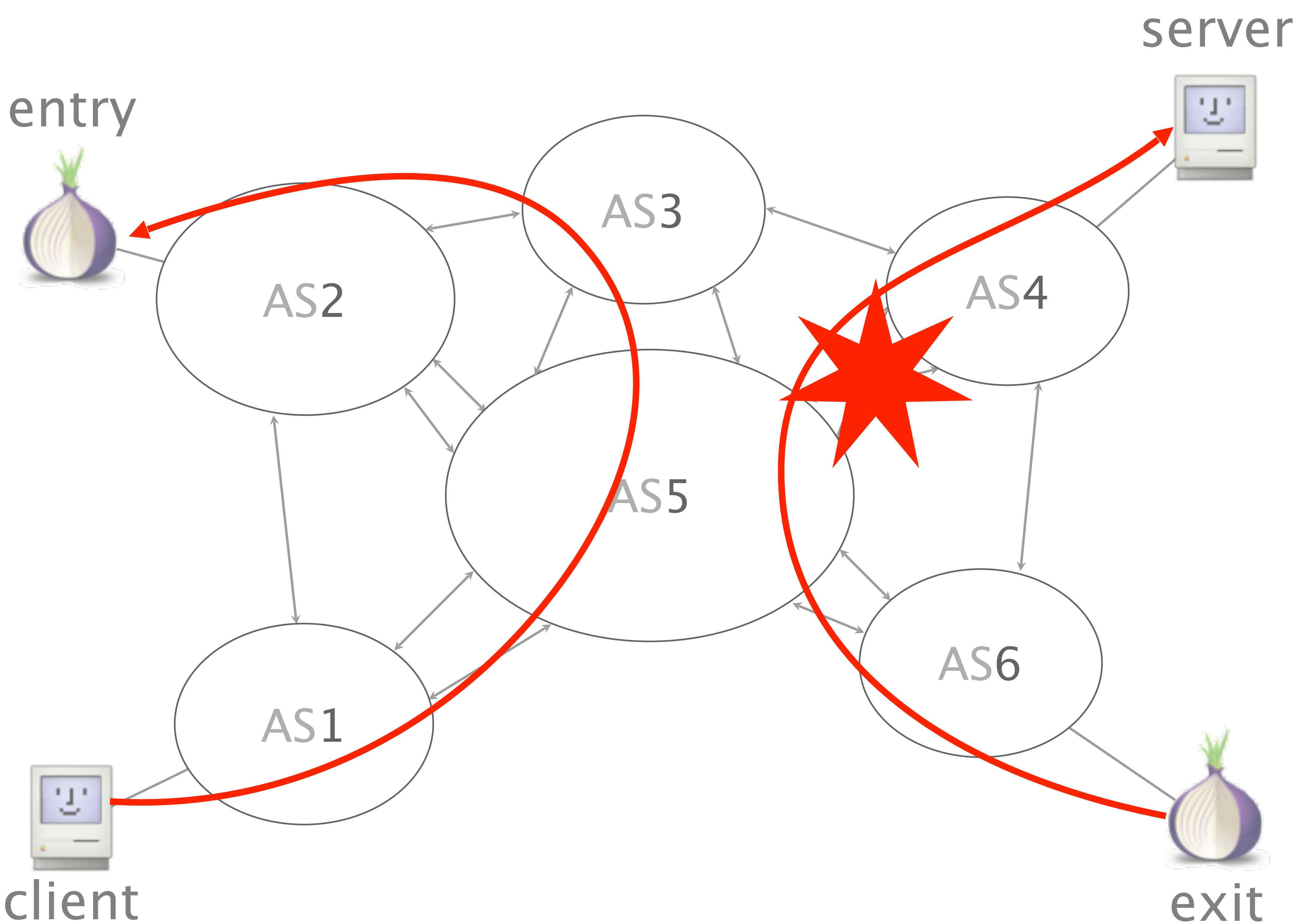
#2.

Natural BGP dynamics increases
the number of AS-level adversaries

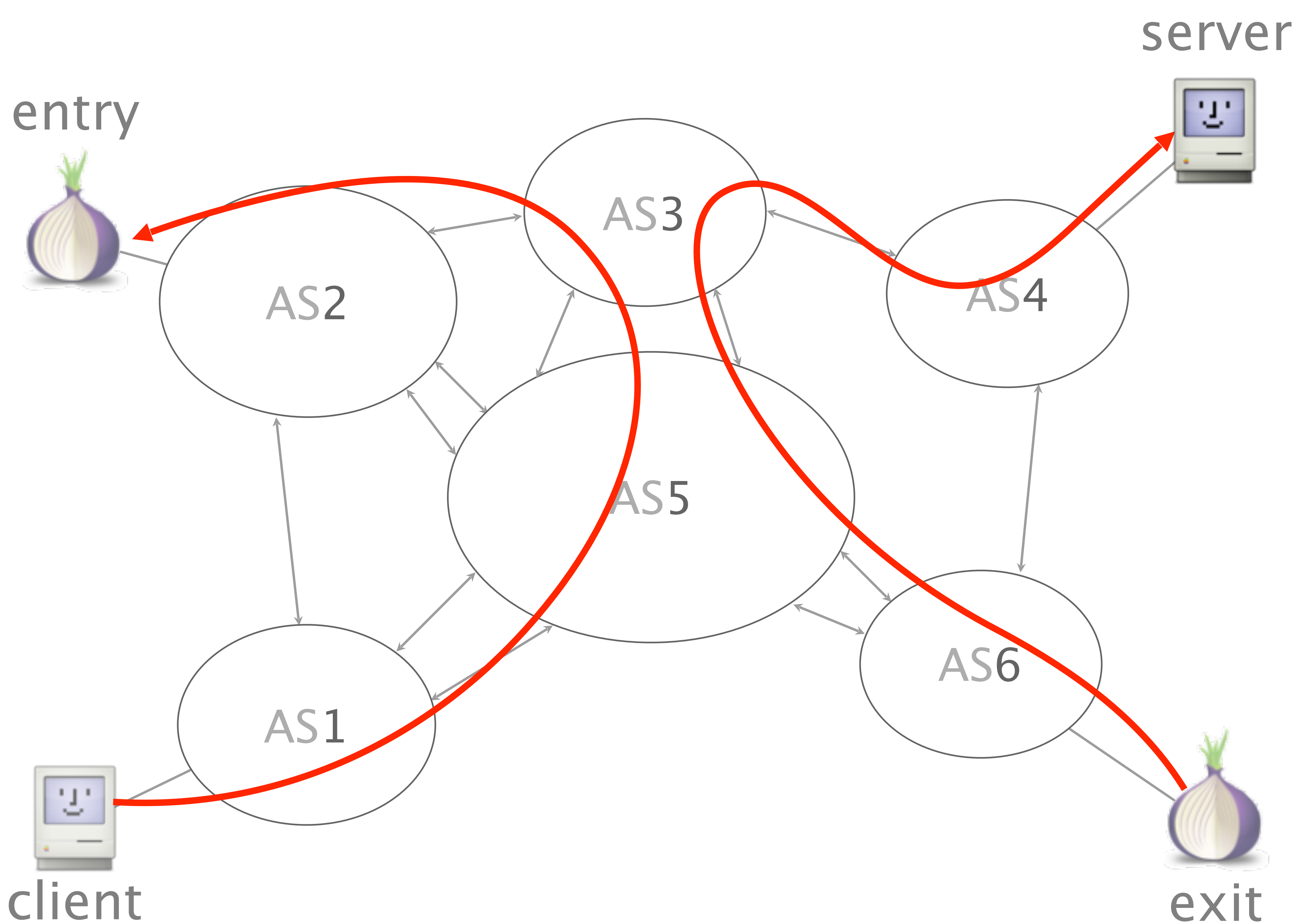
Initially, only AS5 is compromising



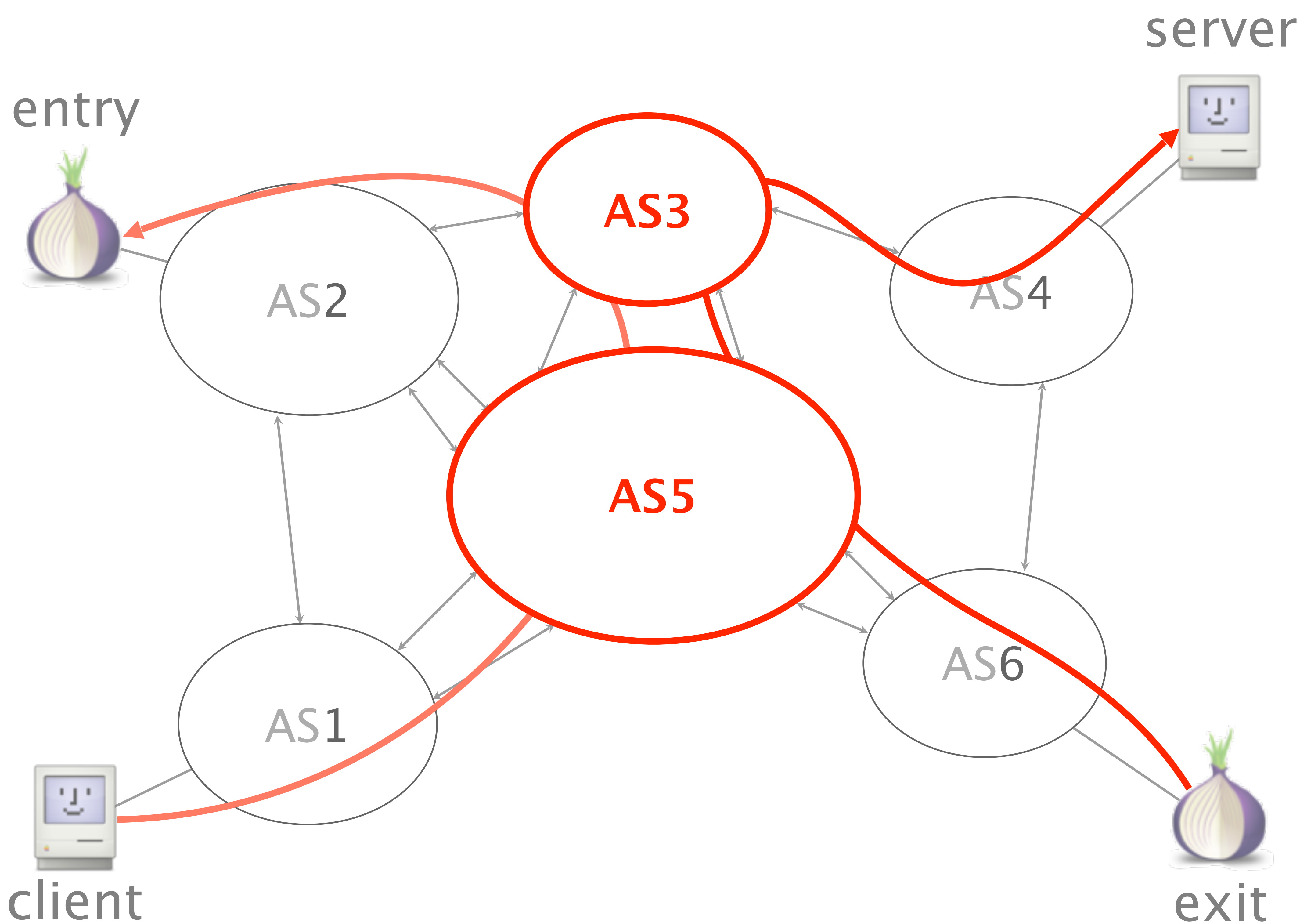
Assume that the link between AS4 and AS5 fails



Traffic gets rerouted via AS3



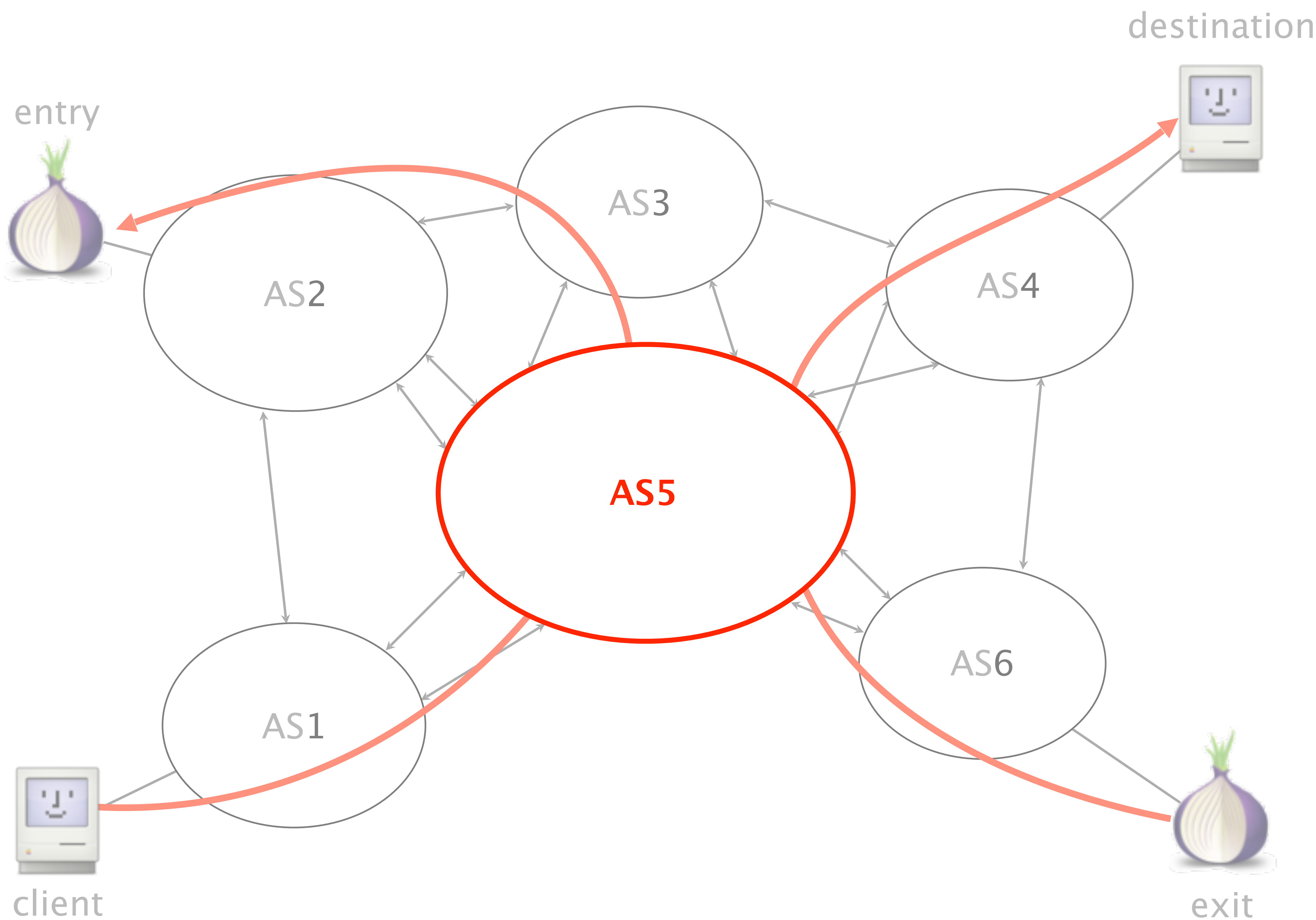
Now, both AS3 and AS5 are seeing client-to-entry and exit-to-server traffic



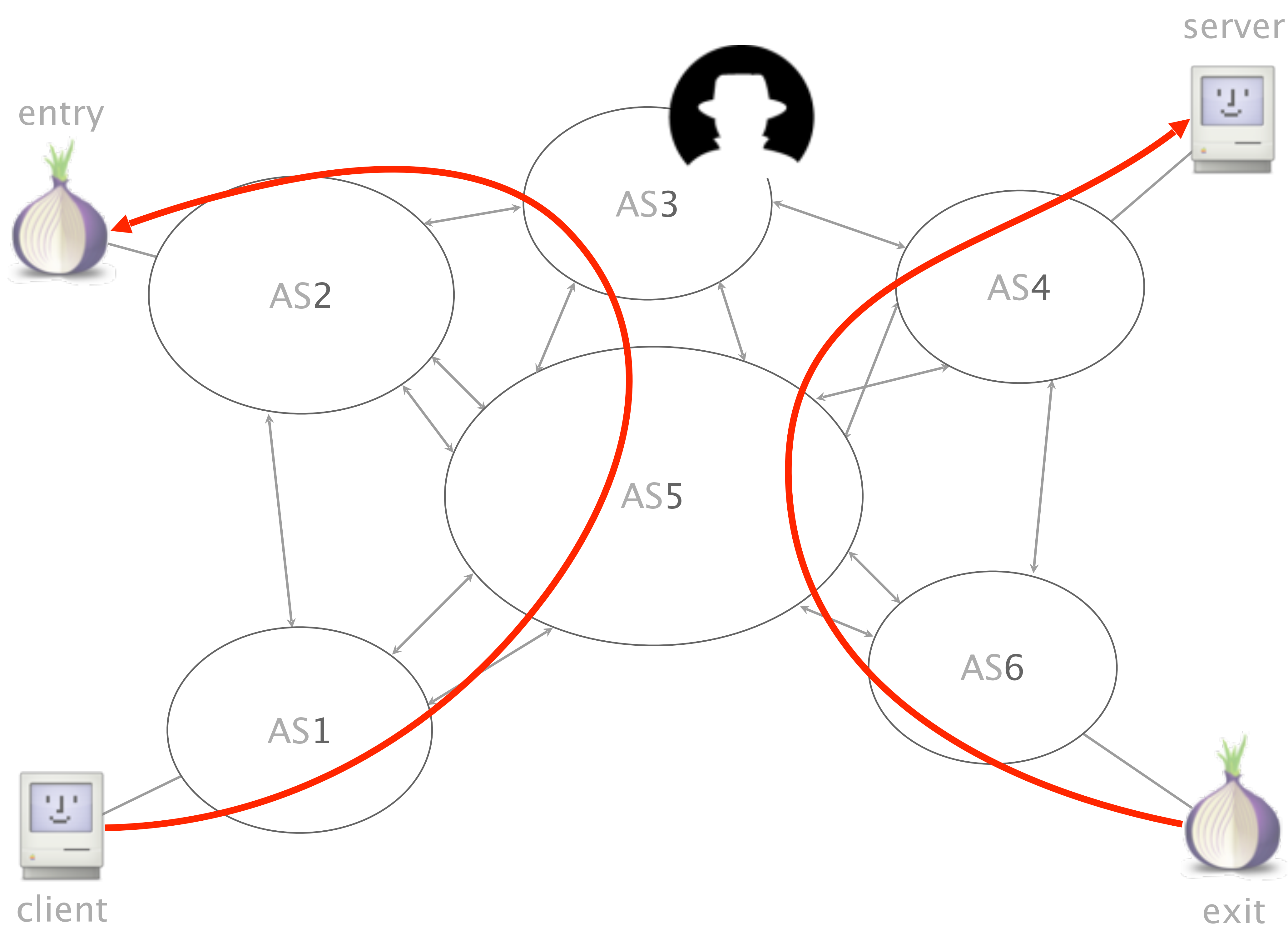
#3.

BGP hijacking attacks enable on-demand, fine-grained Tor attacks

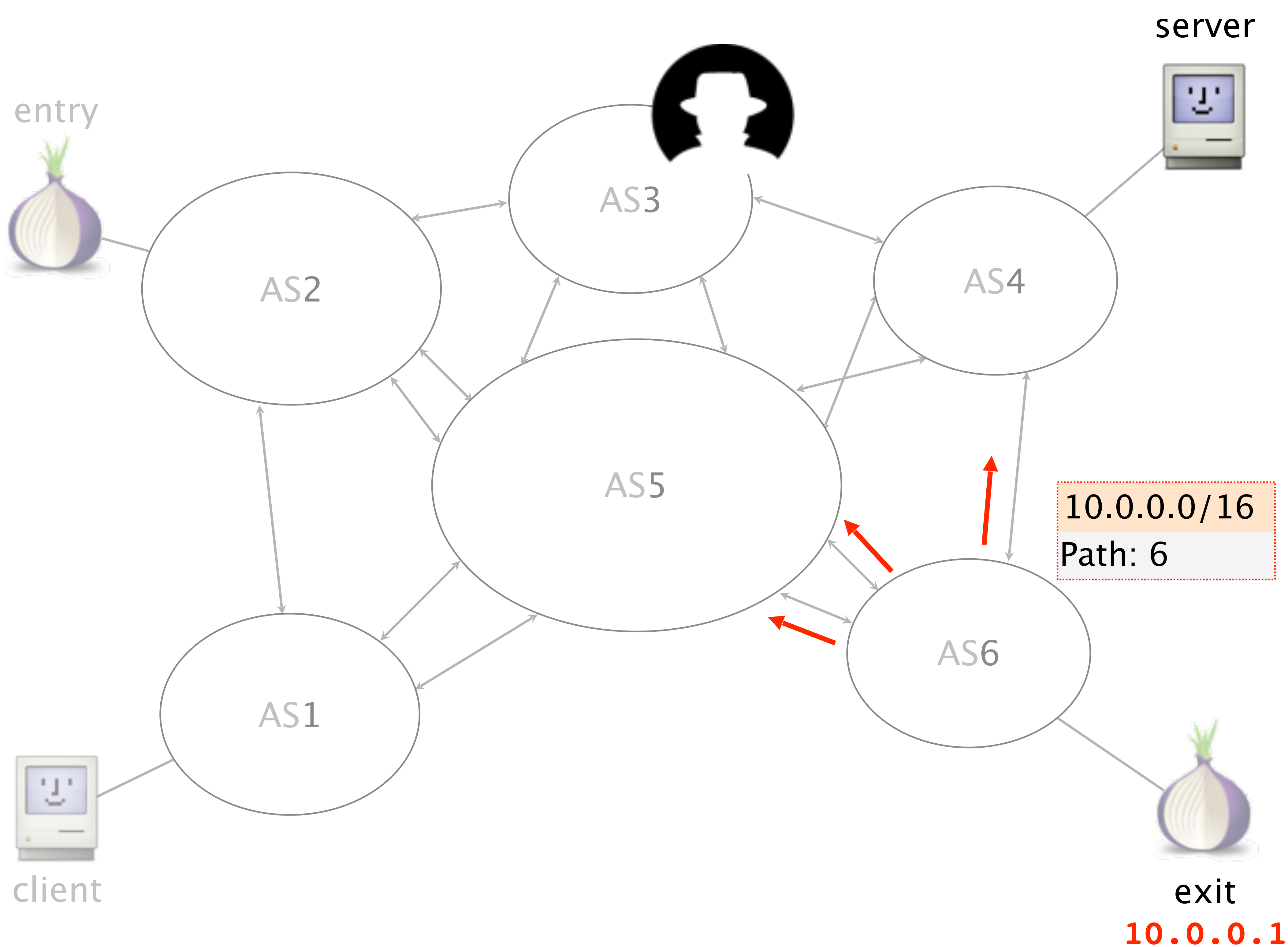
Initially, only AS5 is compromising



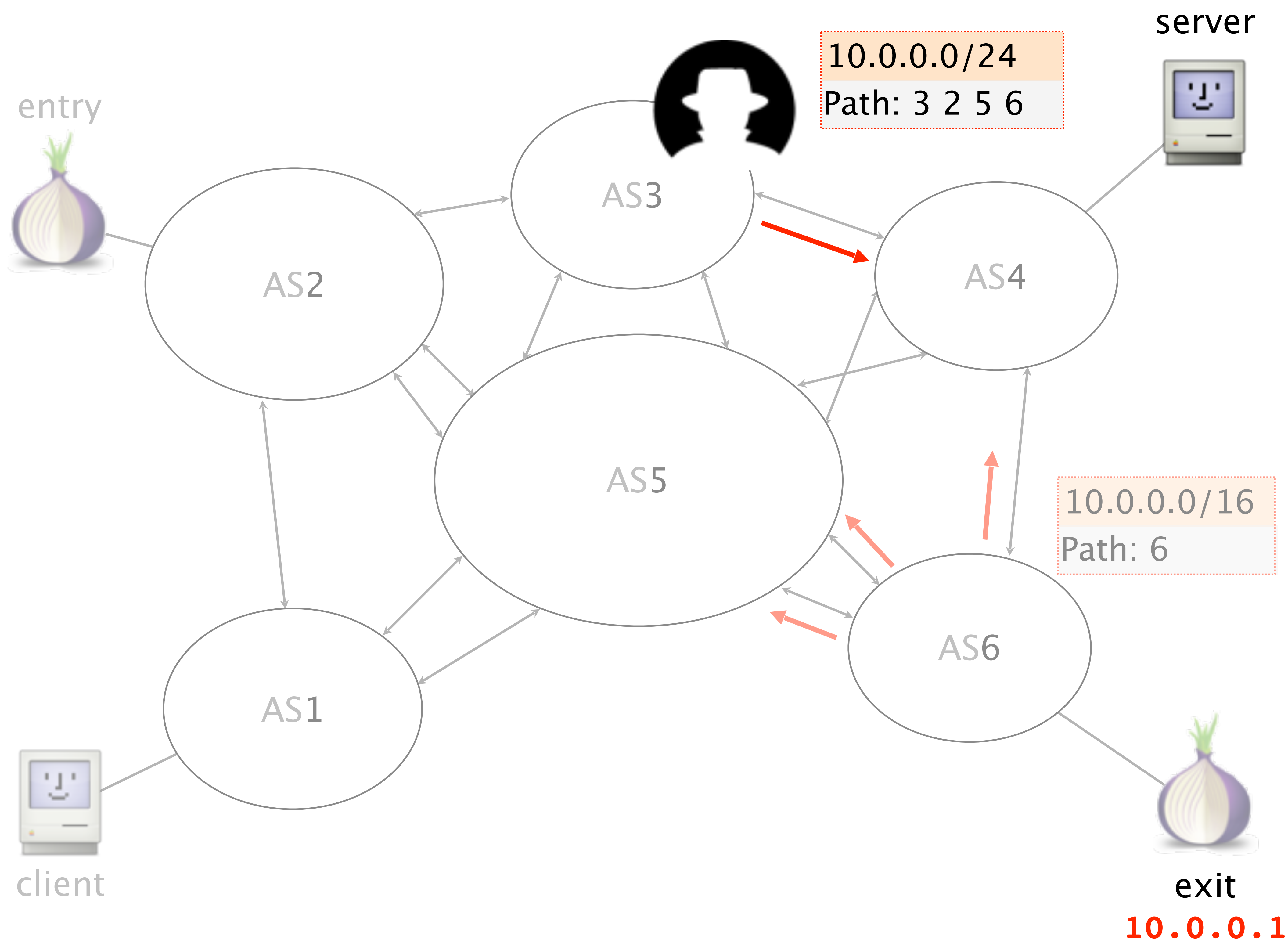
Assume that AS3 is a malicious AS, and wants to observe Tor traffic

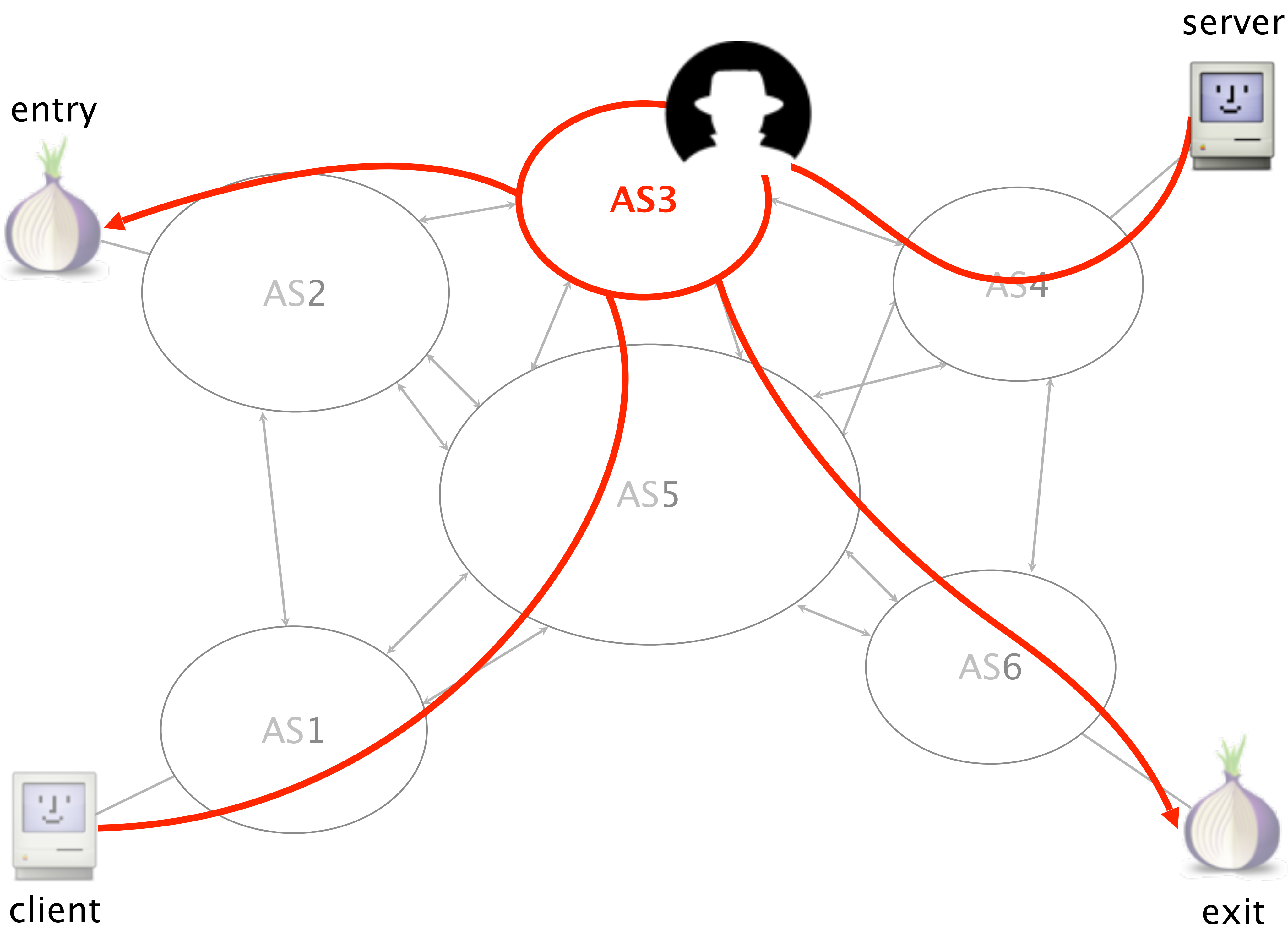


AS3 can put itself on server-to-exit paths by hijacking Tor prefixes



AS3 can put itself on server-to-exit paths by hijacking Tor prefixes




server
entry
AS3
AS2
AS4
AS5
AS6
AS1
client
exit

In April 2014,

Indosat leaked >320k BGP routes over 2 hours

Indonesia Hijacks the World

photo by null0 on Flickr | CC

Yesterday, Indosat, one of Indonesia's largest telecommunications providers, leaked large portions of the global routing table multiple times over a two-hour period. This means that, in effect, Indosat claimed that it "owned" many of the world's networks. Once someone makes such an assertion, typically via an honest mistake in their routing policy, the only question remaining is how much of the world ends up believing them and hence, what will be the

Indosat

One of Indonesia's largest telecommunications providers

Affected 44 Tor Relays

Include 38 guard and 17 exit
11 were both guard and exit

Defenses

- Against Passive Attacker: asymmetric traffic analysis
  - IPSec, traffic obfuscation, etc.
  - Avoid having the same ASes on both ends
- Against Active Attacker: BGP attacks
  - Reactive: monitoring control plane and data plane
  - Proactive: select more “resilient” relays

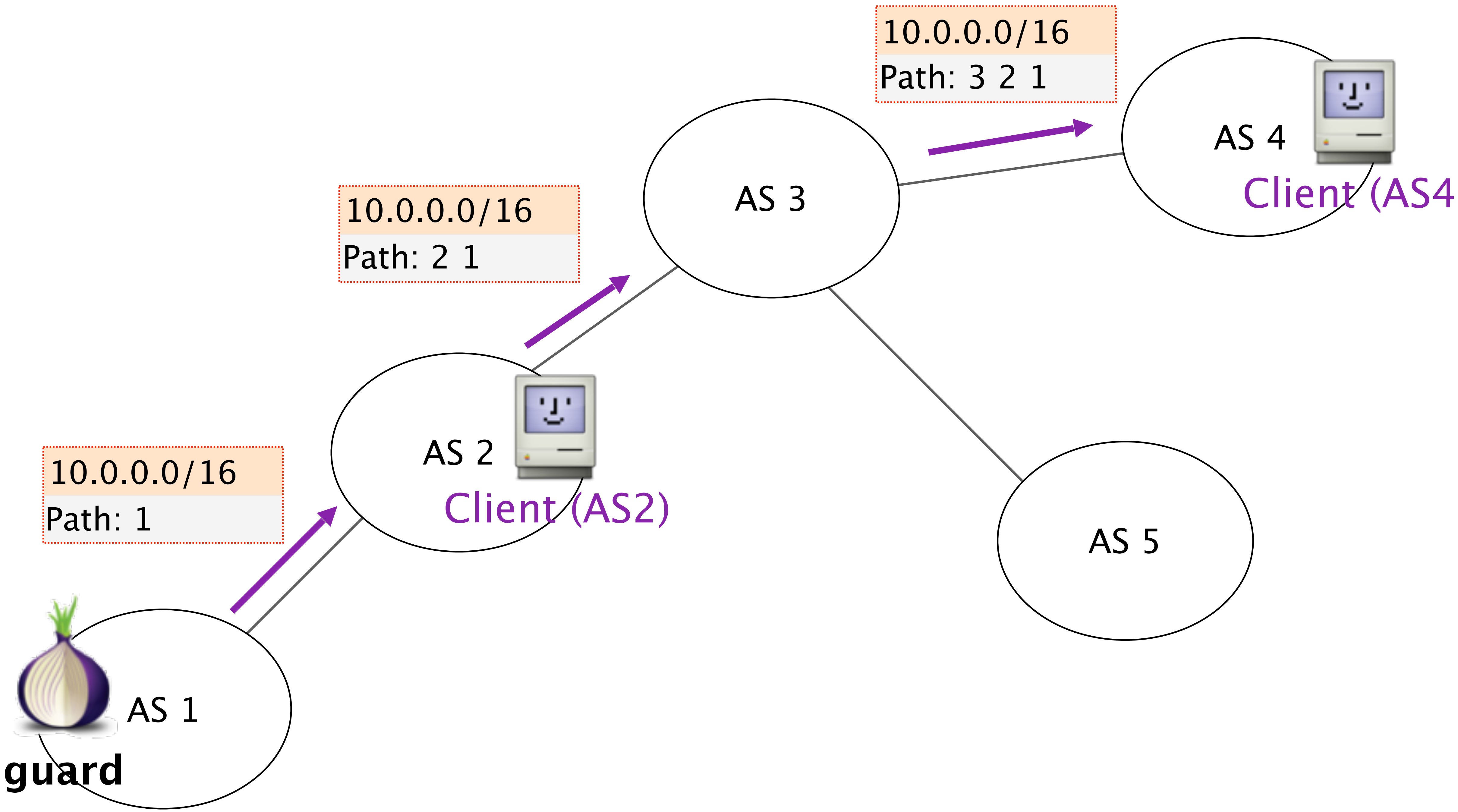
Defenses

- Against Passive Attacker: asymmetric traffic analysis
  - IPSec, traffic obfuscation, etc. — not so practical
  - Avoid having the same ASes on both ends — LasTor, Astoria, etc.
- Against Active Attacker: BGP attacks
  - Proactive: select more “resilient” relays
  - Reactive: monitoring system

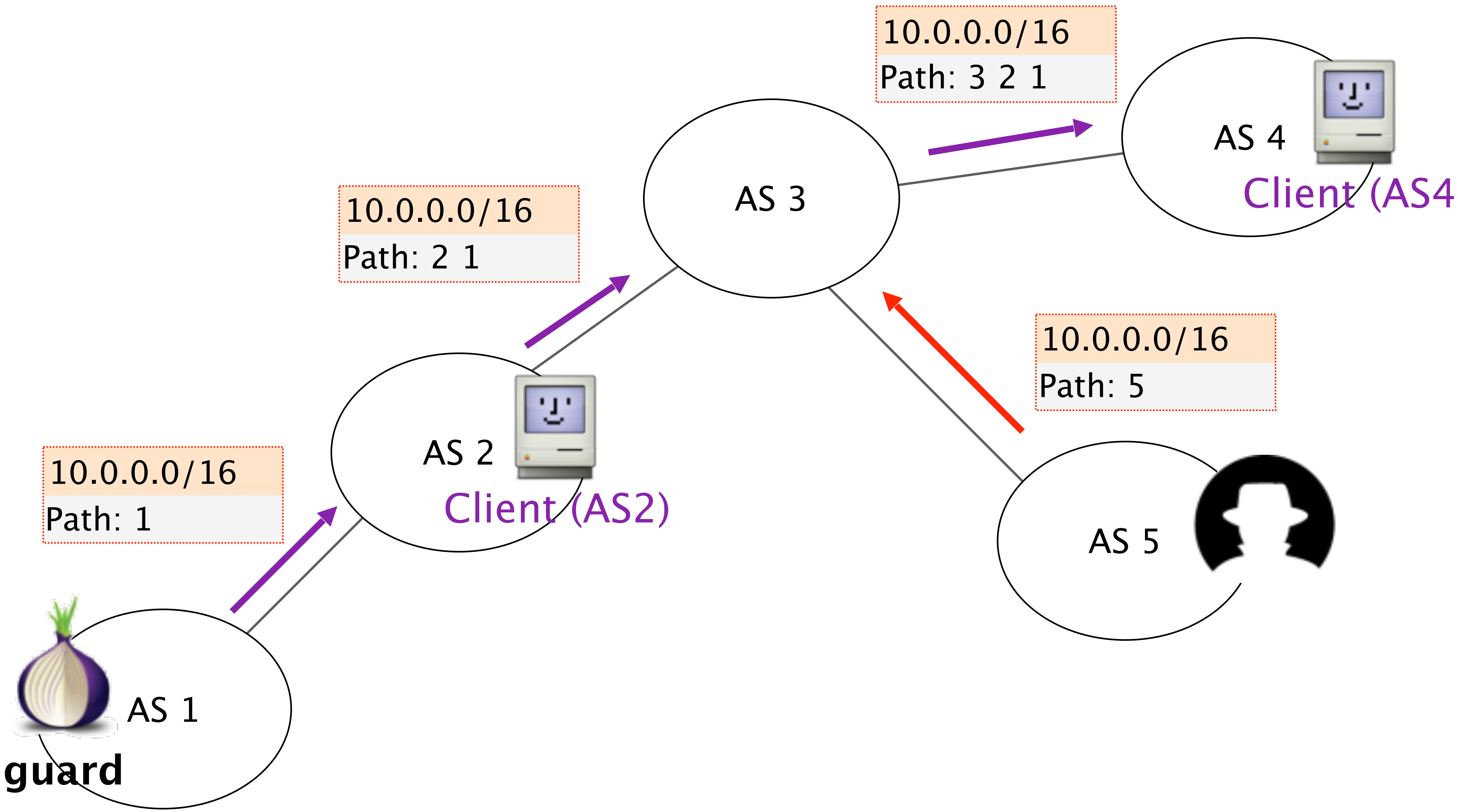
  Our work

Proactive Defense

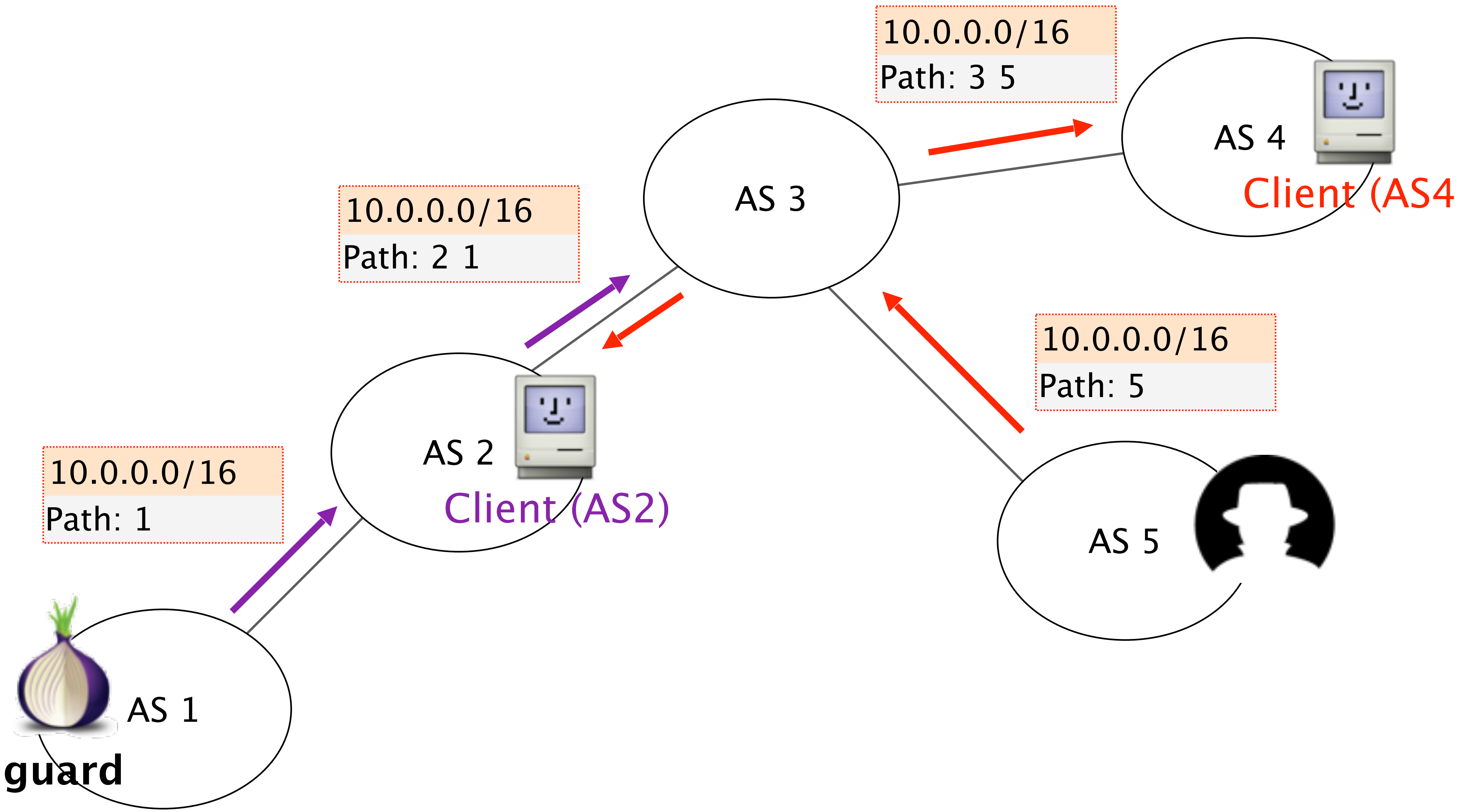
Two Tor clients are using the same Tor guard



AS 5 hijacks Tor prefix (equally-specific)

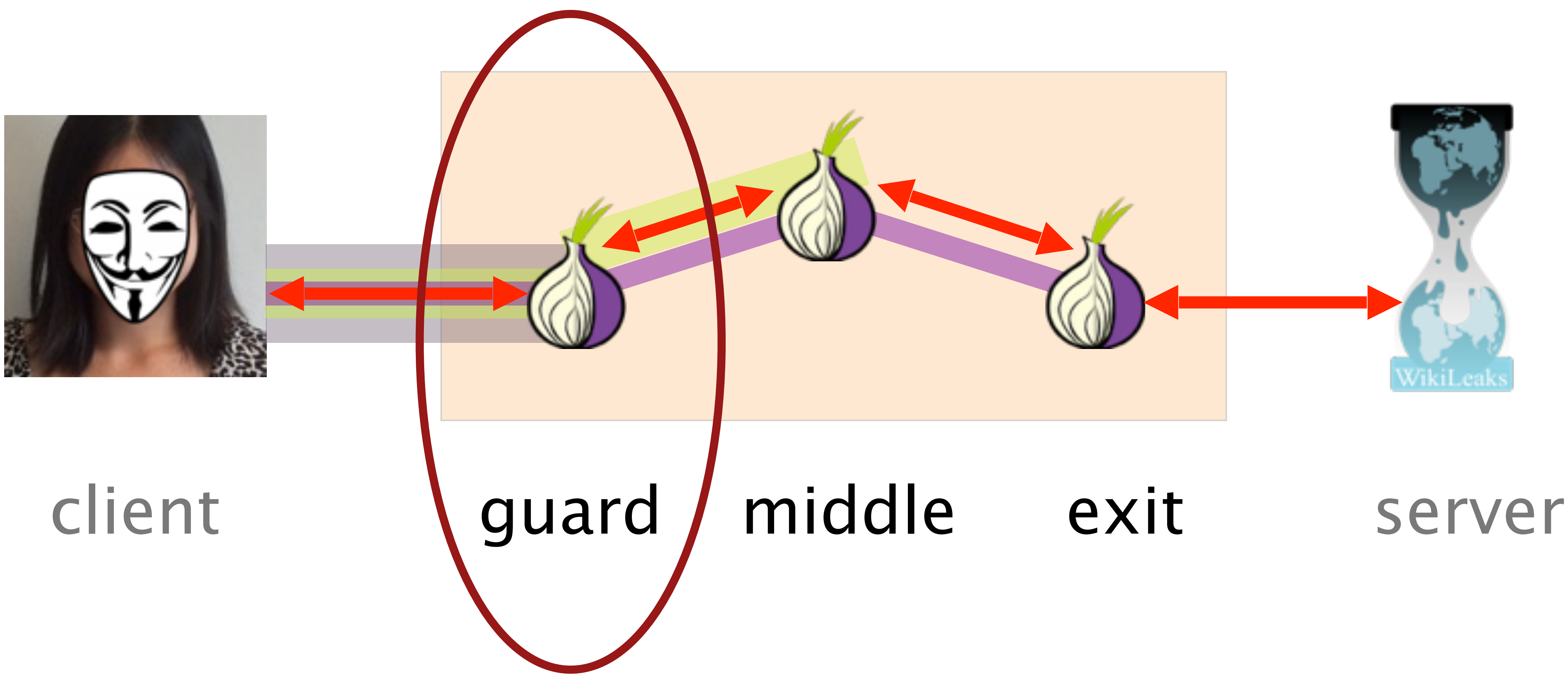


Tor client (AS2) is resilient to this attack, while Tor client (AS4) is not



Key Insight:

Choose a guard relay such that a Tor client AS is resilient to attacks on its guard relay



Reactive Defense

BGP Monitoring System

Live monitoring system

- live BGP updates for Tor relay IPs
- run detection analytics on the updates
- trigger/log warnings

Detection Analytics

Anomaly detection in real time

- Frequency Analytic
- Time Analytic

Key Insight:
Attacks are infrequent and short-lived

Detection Analytics

Anomaly detection in real time

- Frequency Analytic
- Time Analytic

Key Insight:
Attacks are infrequent and short-lived

Evaluation

Preliminary evaluation from March to May 2016

Frequency Analytic: False Positive 0.38%

Time Analytic: False Positive 0.19%

Most Tor prefixes are announced by a single AS in all updates

Data/script available on:

raptor.princeton.edu/tor_metrics/

Index of /tor_metrics

Name	Last modified	Size	Description
Parent Directory		-	
all-updates.tar	2017-11-07 19:13	83M	
all-updates/	2017-11-07 19:12	-	
counter-raptor.html	2017-11-07 14:40	3.7K	
detection.py	2017-11-07 14:40	8.8K	

Data/script available on:

raptor.princeton.edu/tor_metrics/

```
YS-MacBook-Pro:bgp-tor yixinsun$ python detection.py
usage: detection.py [-h] [--freq_thresh FREQ_THRESH]
                    [--time_thresh TIME_THRESH] --method {
                    --cur_month CUR_MONTH --prev_month PREV
detection.py: error: argument --method is required
YS-MacBook-Pro:bgp-tor yixinsun$ python detection.py --cur_
-06.txt --method=time
00:05:16.525397
00:05:26.529785
Finished previous month...
00:05:40.253500
Num of FP unique (prefix,AS) pair: 23
Num of unique (prefix,AS) pair: 1673
Num of FP updates: 2317
Num of total updates: 1532147
```

Future works on monitoring system

- Play with the data

- Tune parameters: threshold, time window, etc.

- Interpret warnings: pattern? duplicated warnings?

- BGP Collectors: which ones to pick?

Summary & Resources

- Raptor: network dynamics empower adversaries
- Counter-Raptor: proactive and reactive defenses

Project site: raptor.princeton.edu

Tor BGP data/script: raptor.princeton.edu/tor_metrics

Tor code (resilient relay):
github.com/inspire-group/Counter-Raptor-Tor-Client