# On Traffic Analysis in Tor

*Guest Lecture, ELE 574*

*Communications Security and Privacy*

*Princeton University*

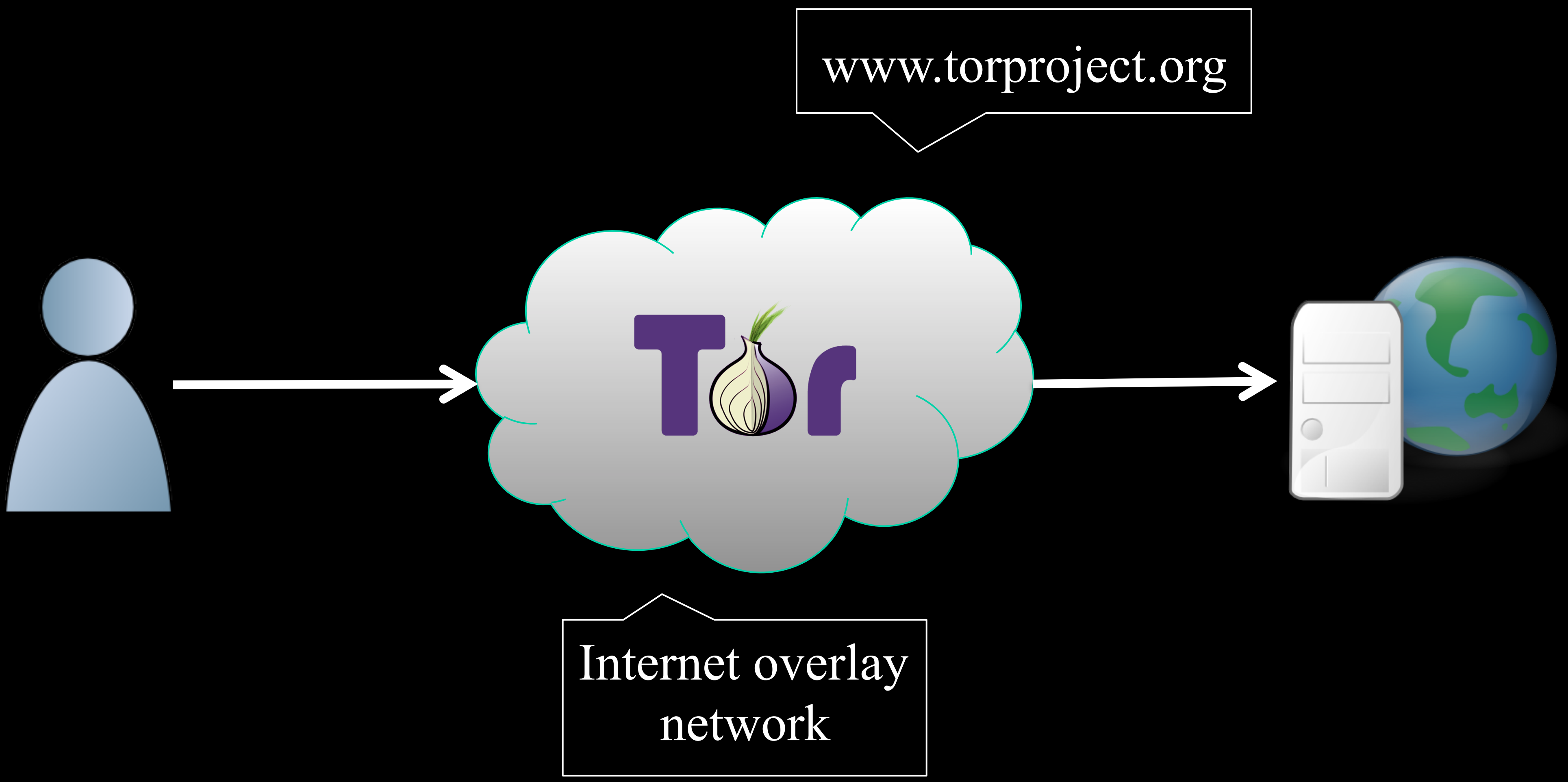
*April 3rd, 2014*


Dr. Rob Jansen

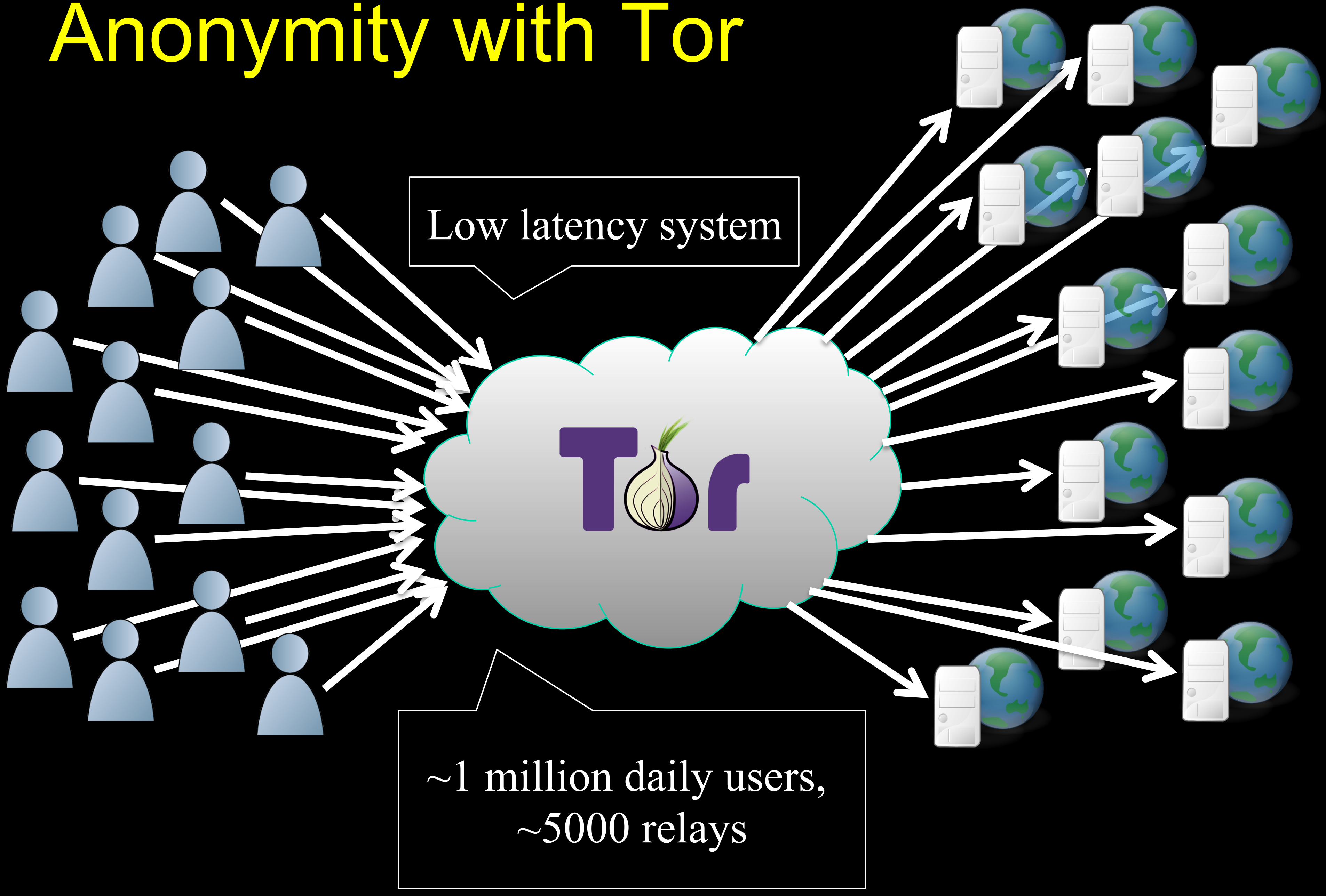
U.S. Naval Research Laboratory

rob.g.jansen@nrl.navy.mil

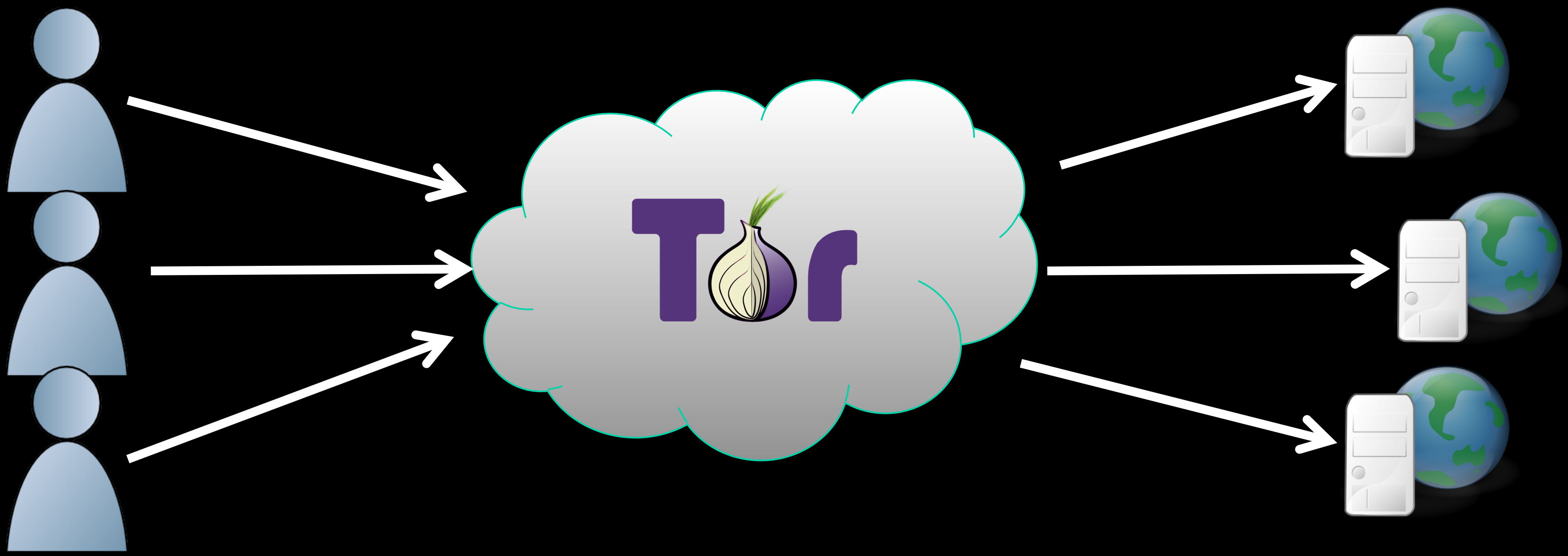
# Anonymity with Tor



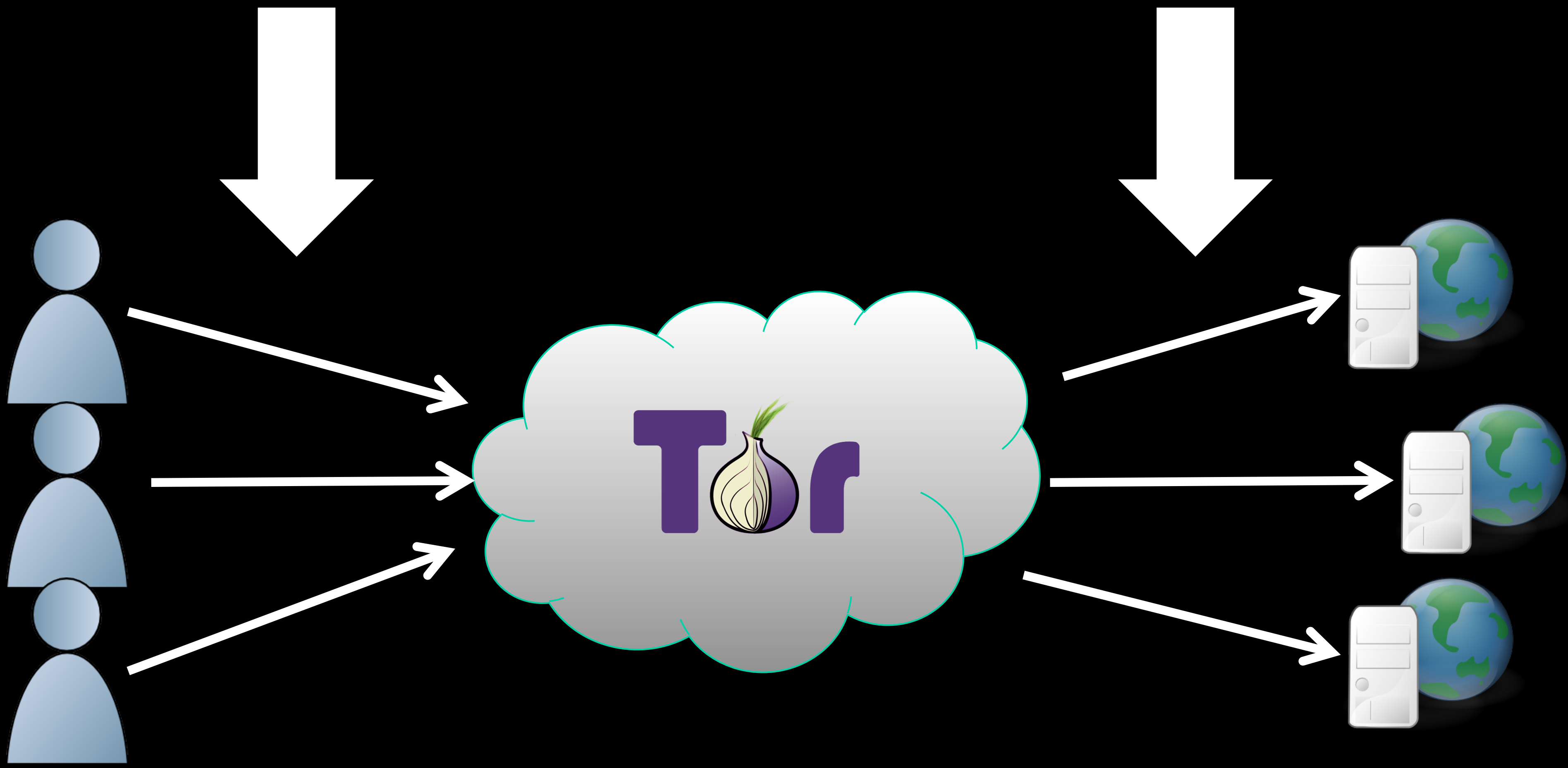
# Anonymity with Tor



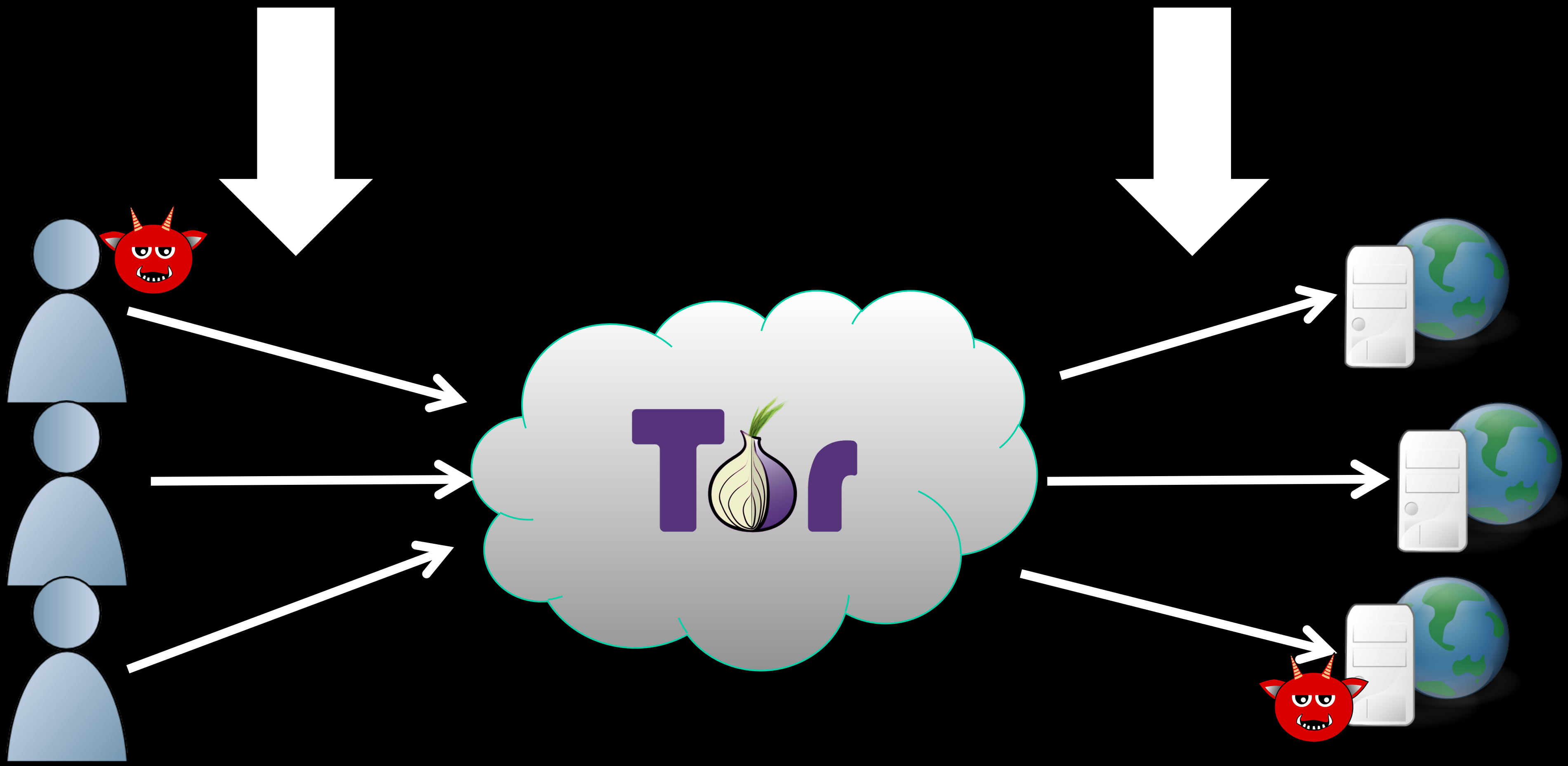
# Traffic Correlation



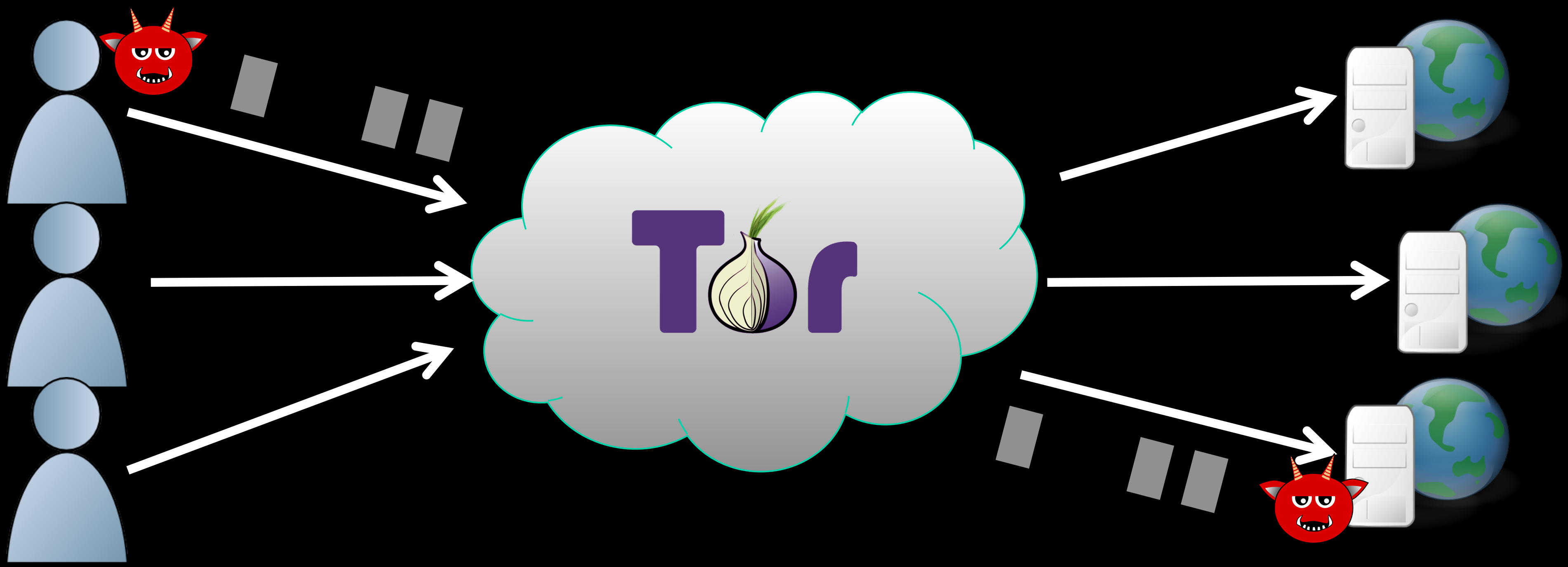
# Traffic Correlation



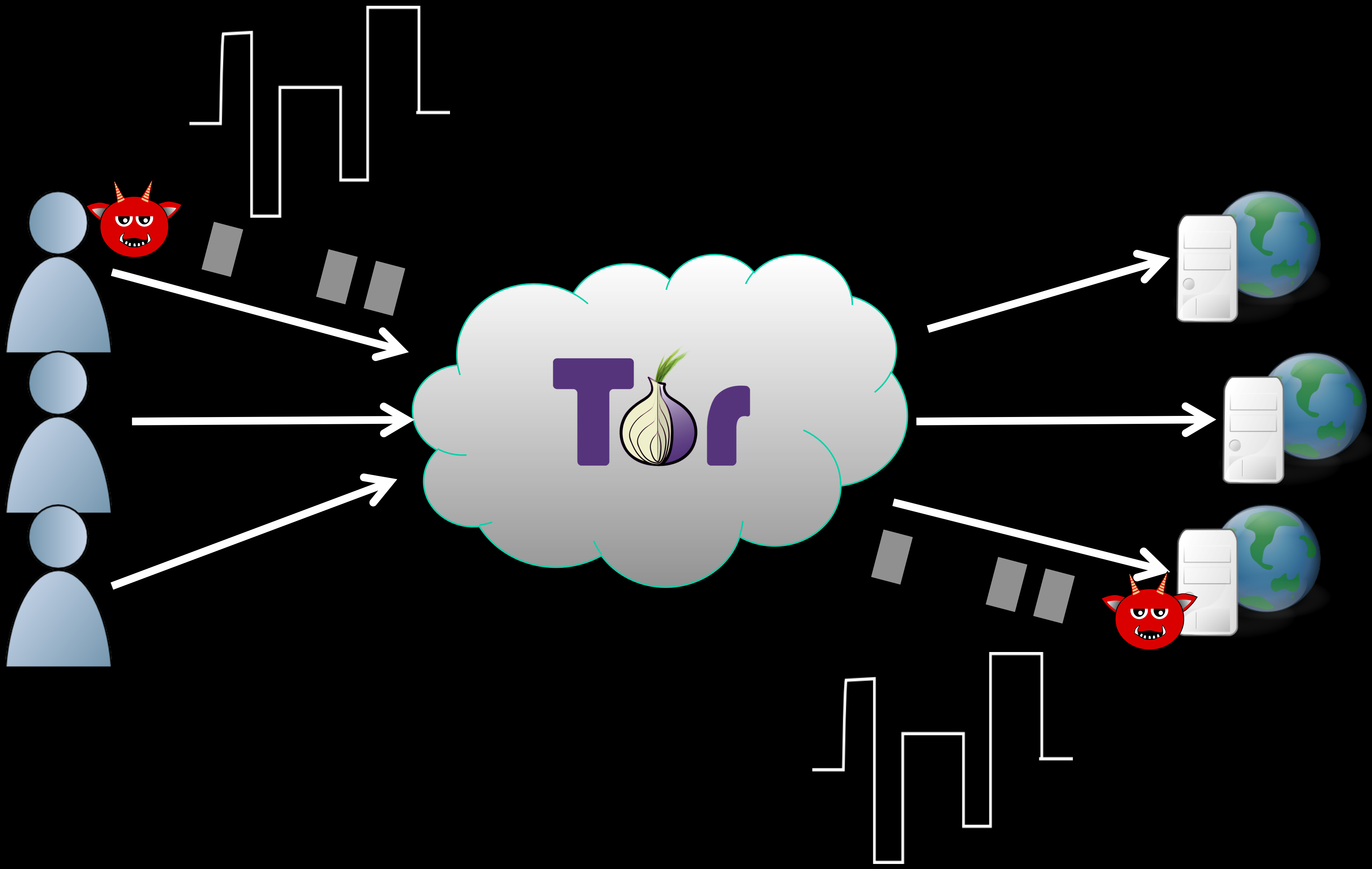
# Traffic Correlation



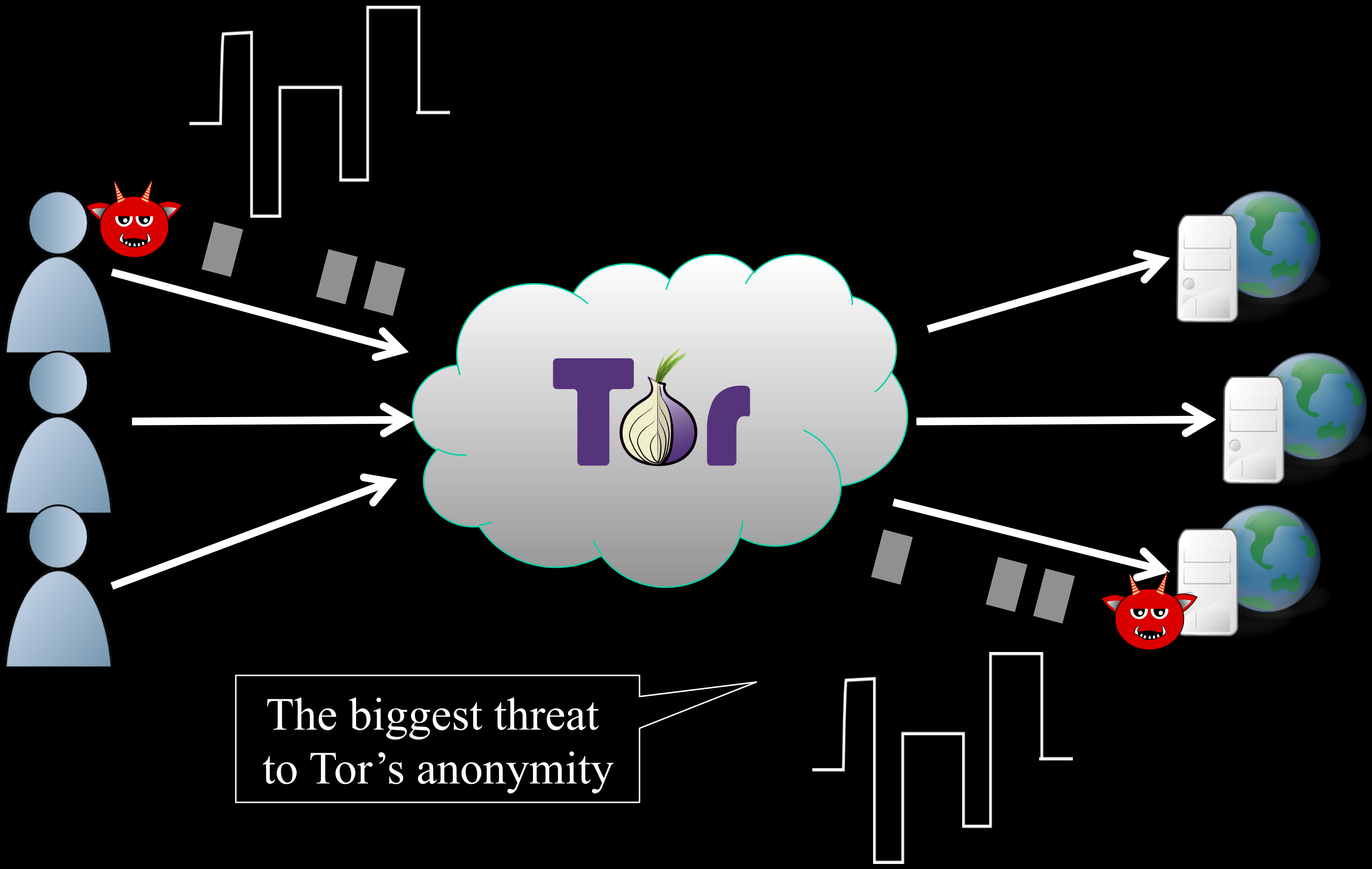
# Traffic Correlation



# Traffic Correlation

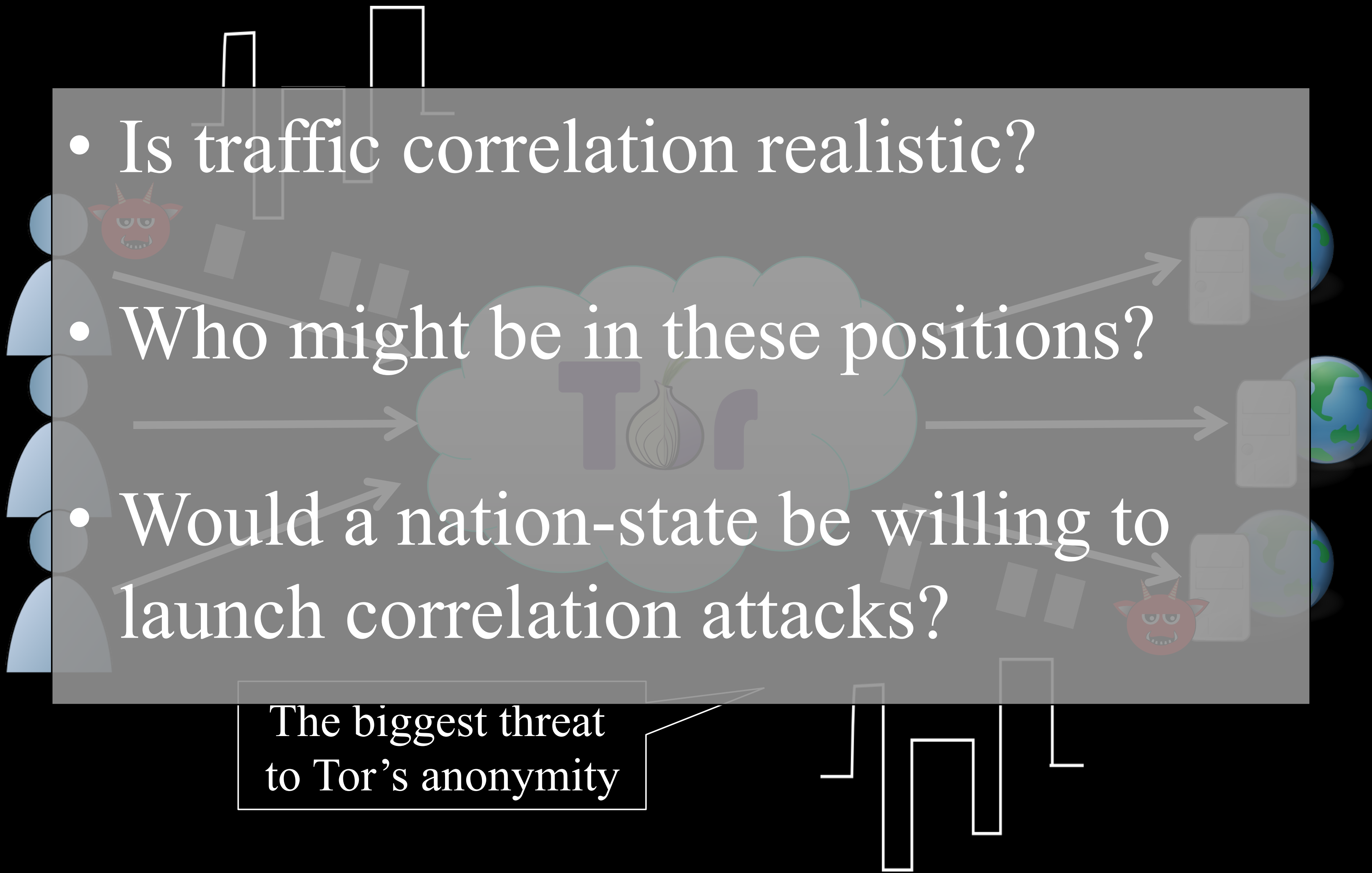


# Traffic Correlation

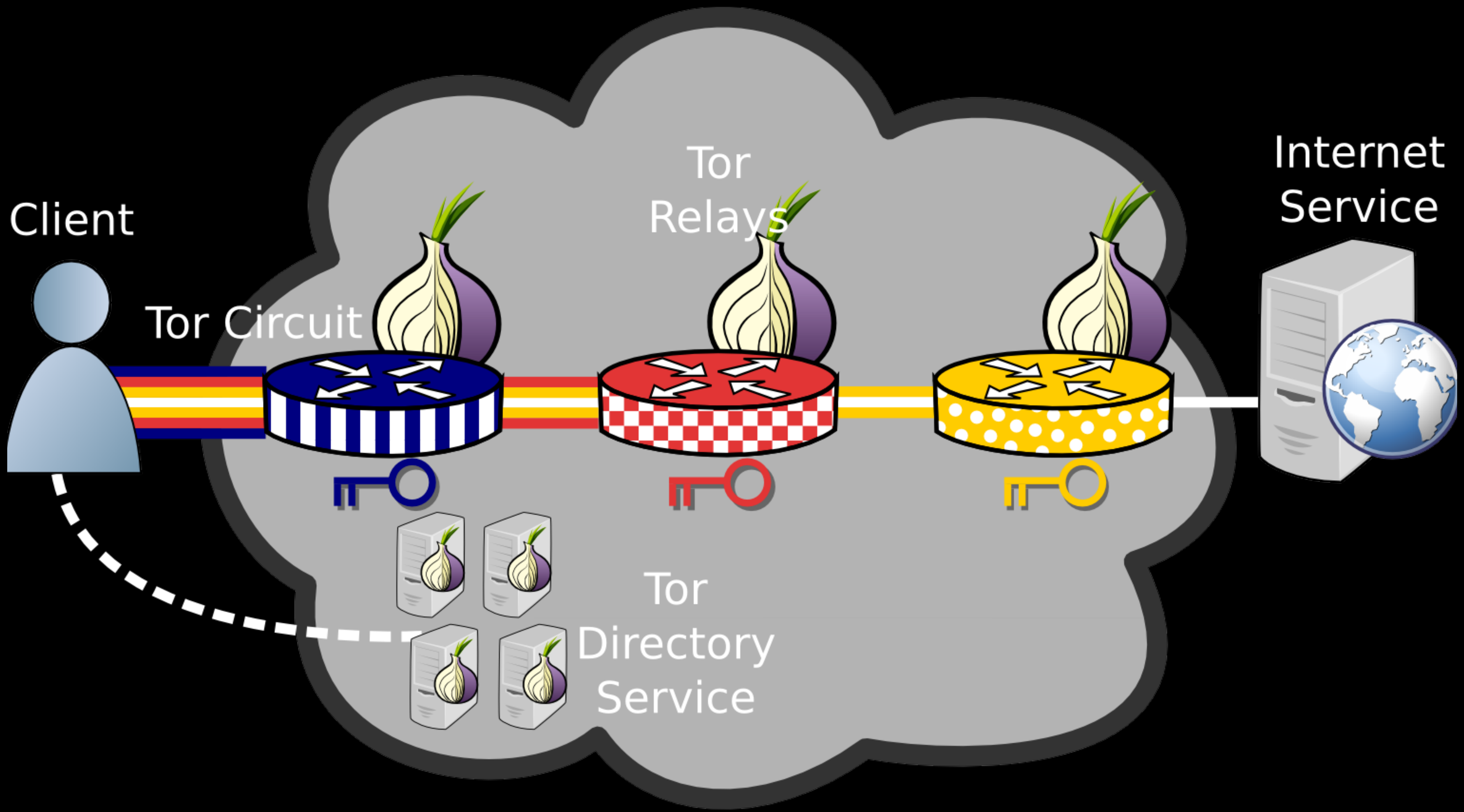


# Traffic Correlation

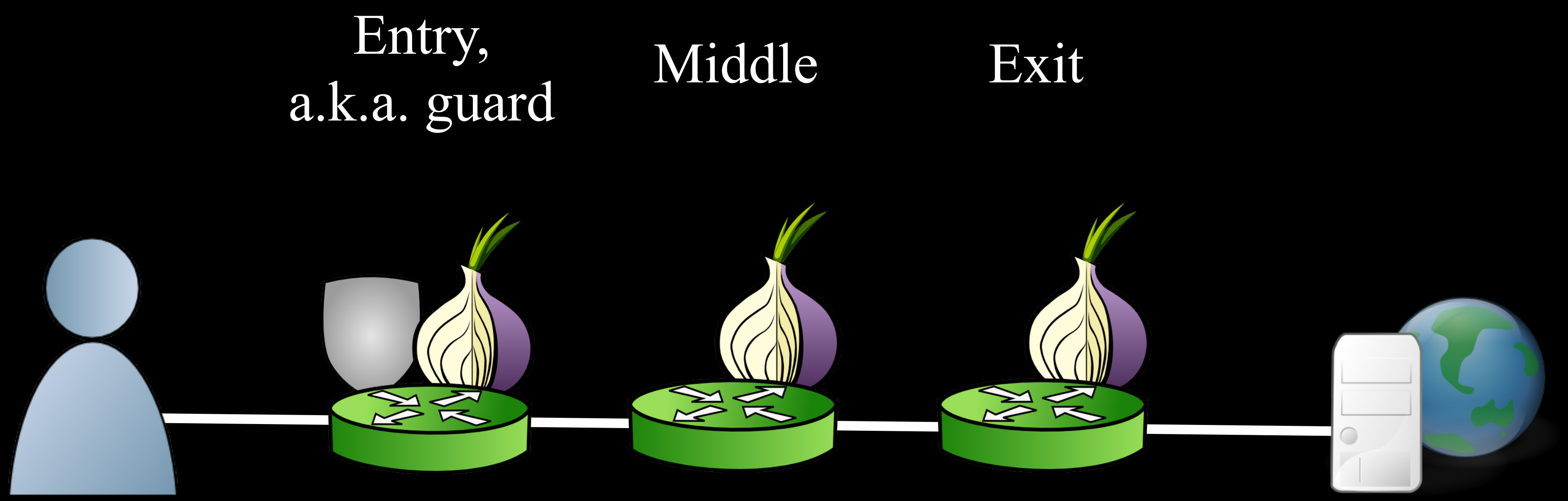
- Is traffic correlation realistic?
- Who might be in these positions?
- Would a nation-state be willing to launch correlation attacks?



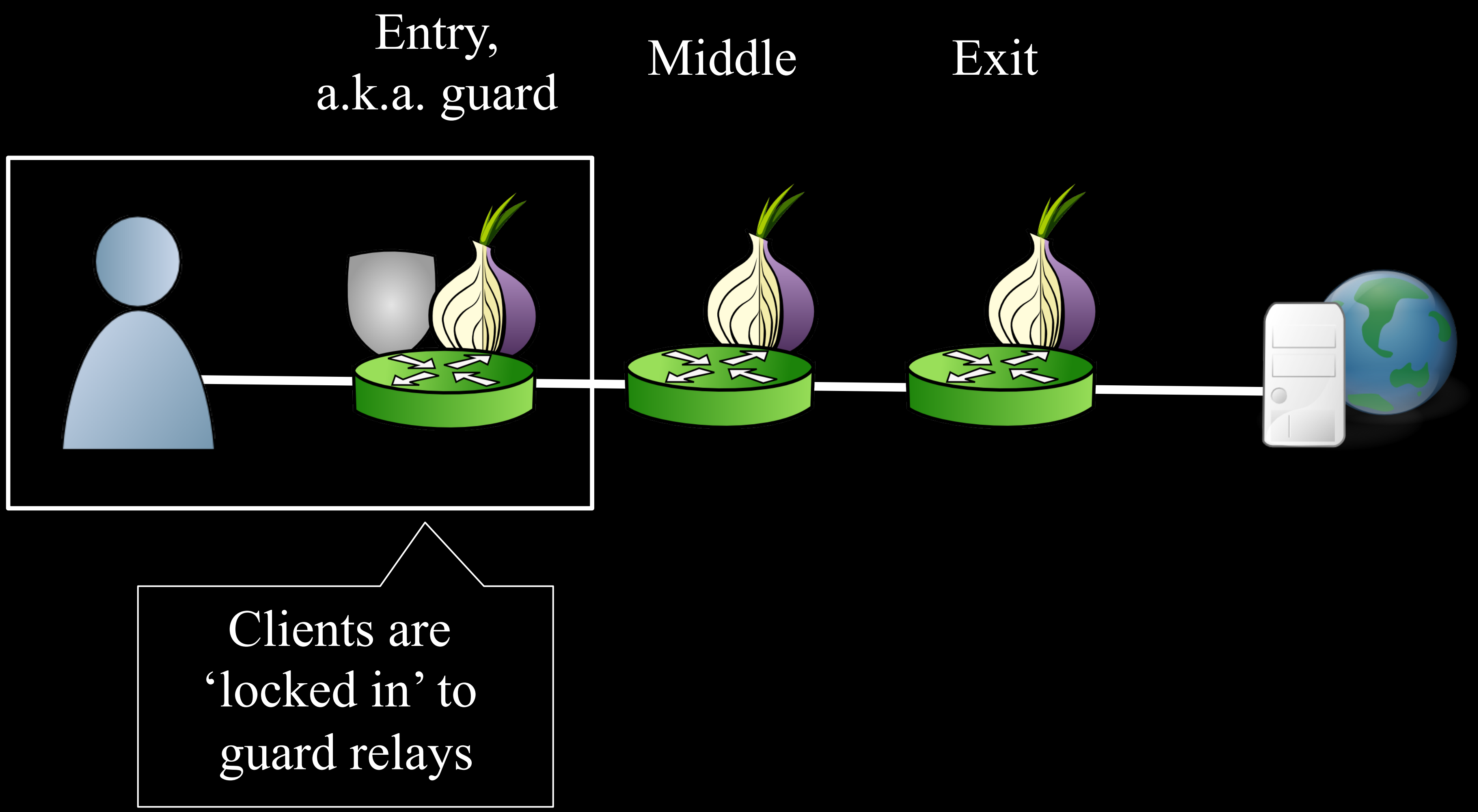
# Anonymity with Onion Routing



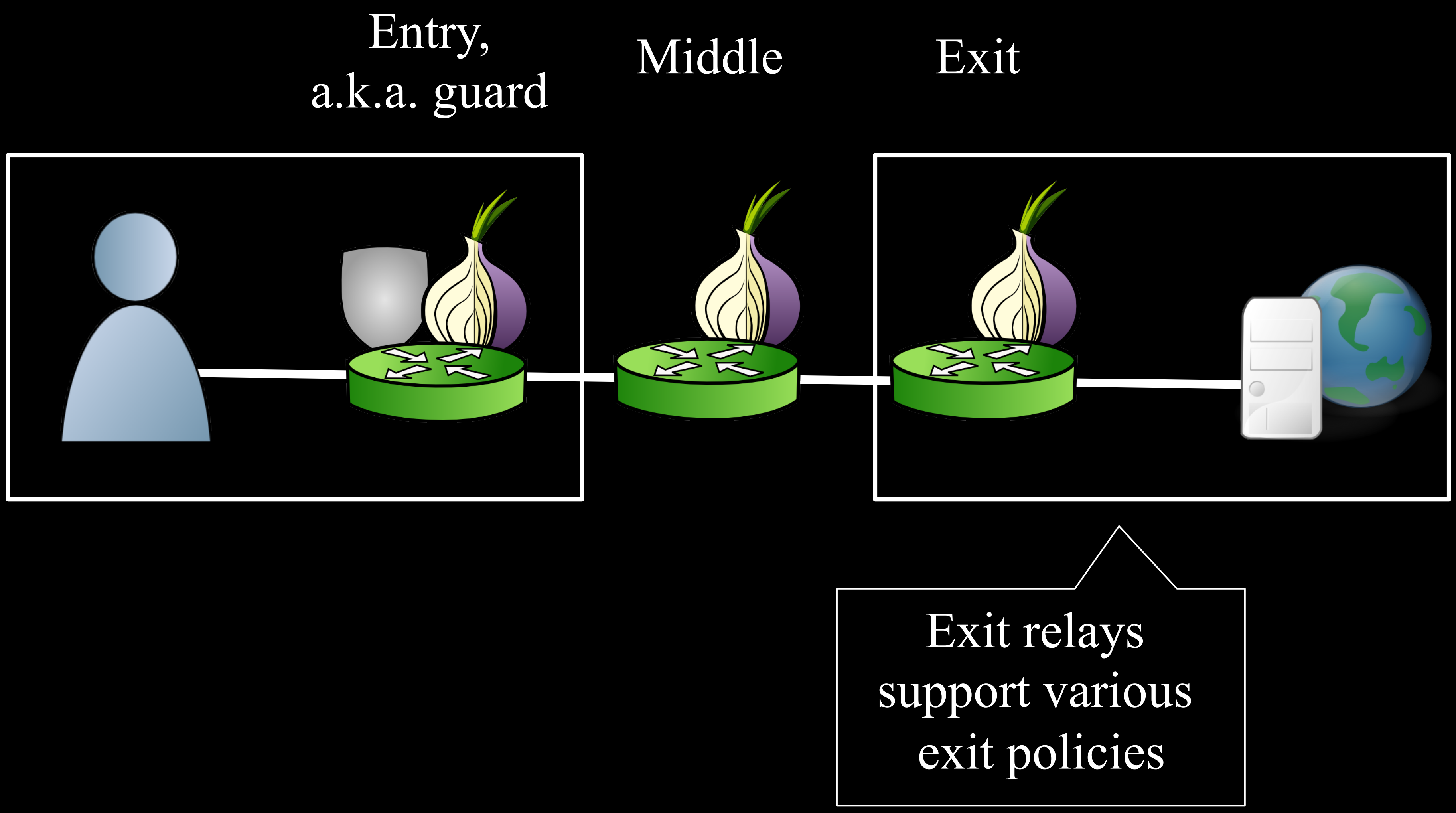
# Traffic Correlation



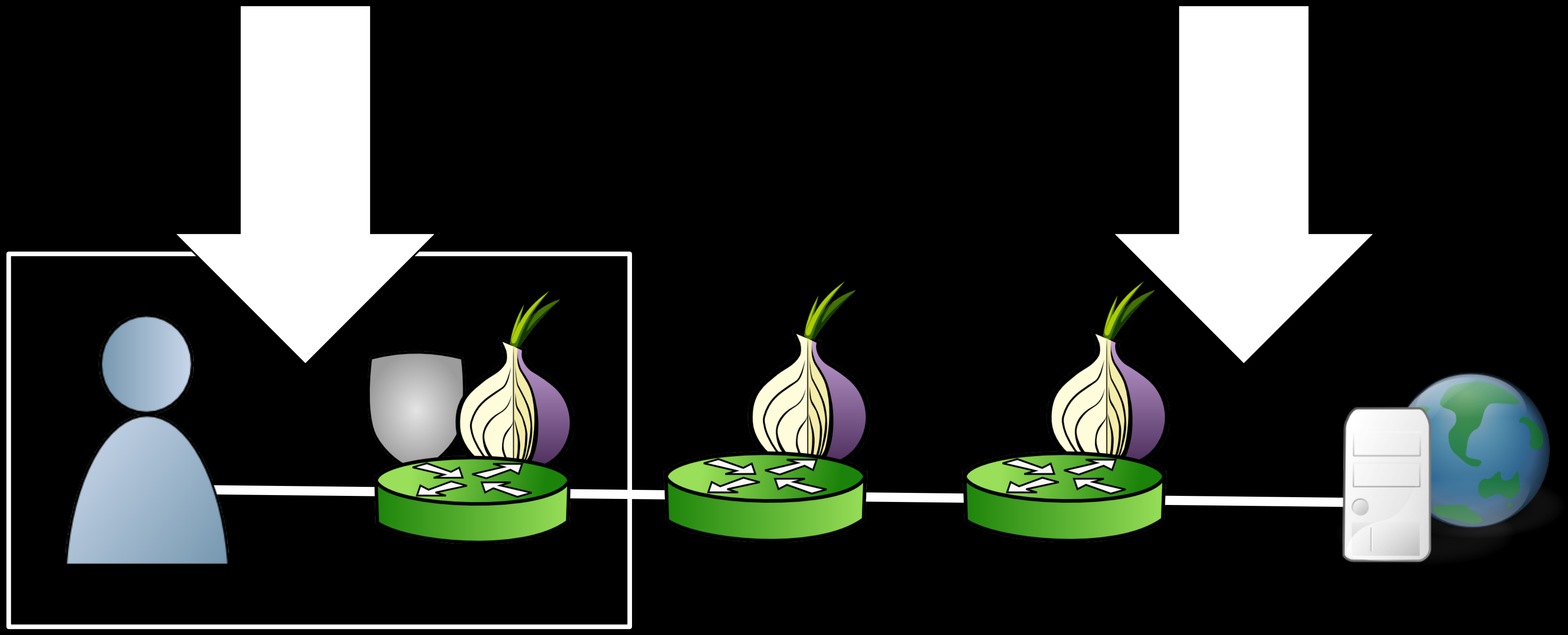
# Traffic Correlation



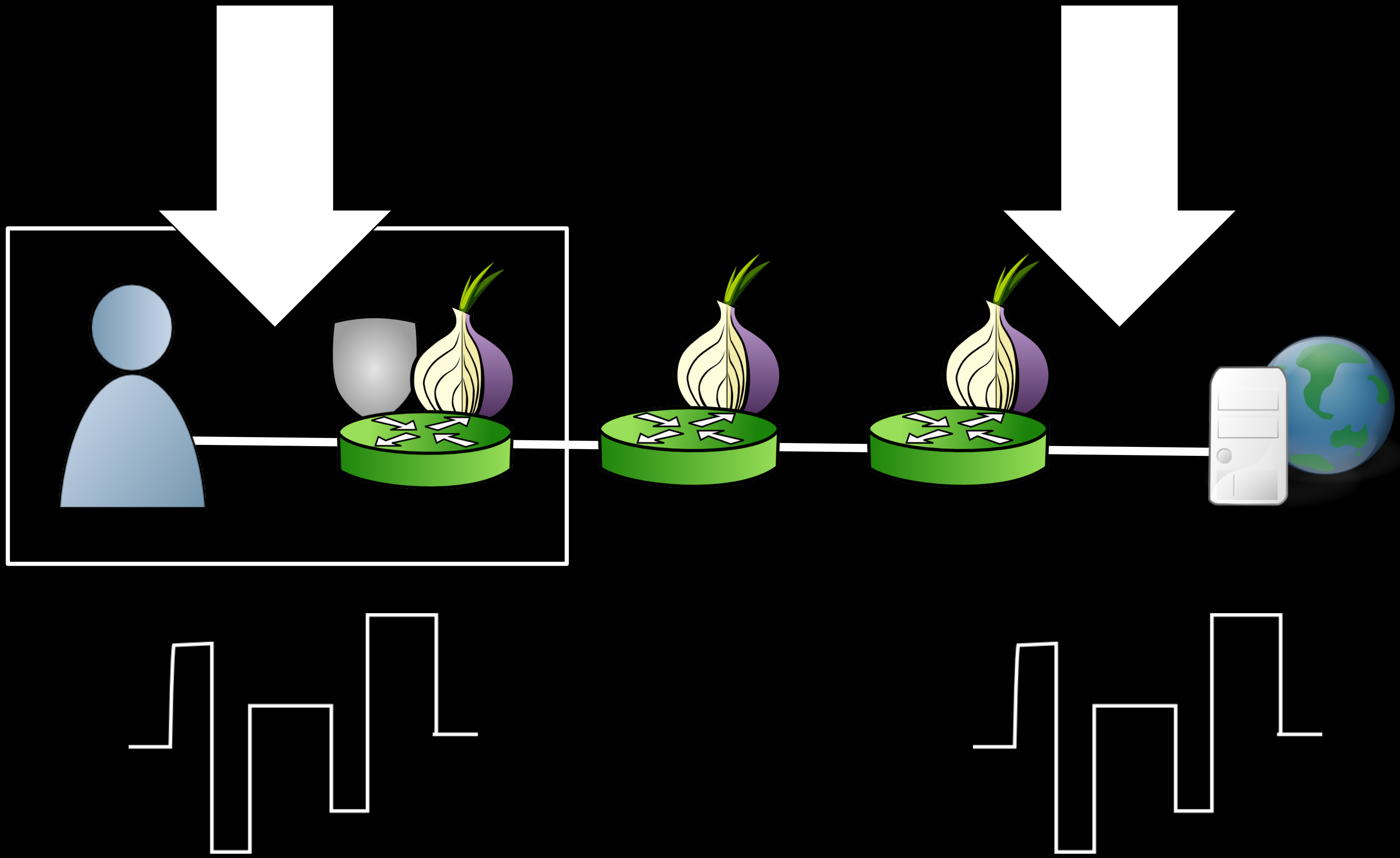
# Traffic Correlation



# Traffic Correlation

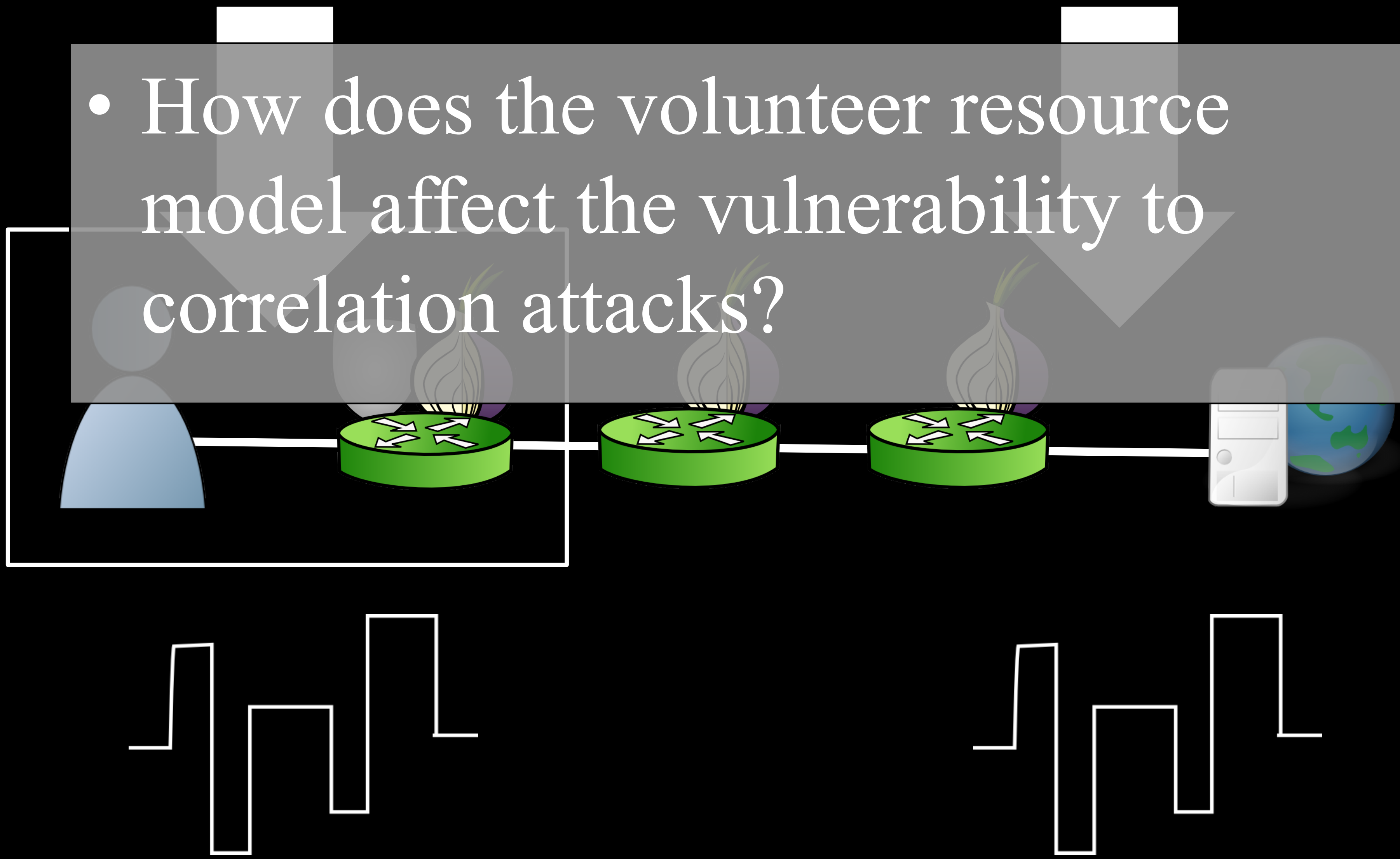


# Traffic Correlation



# Traffic Correlation

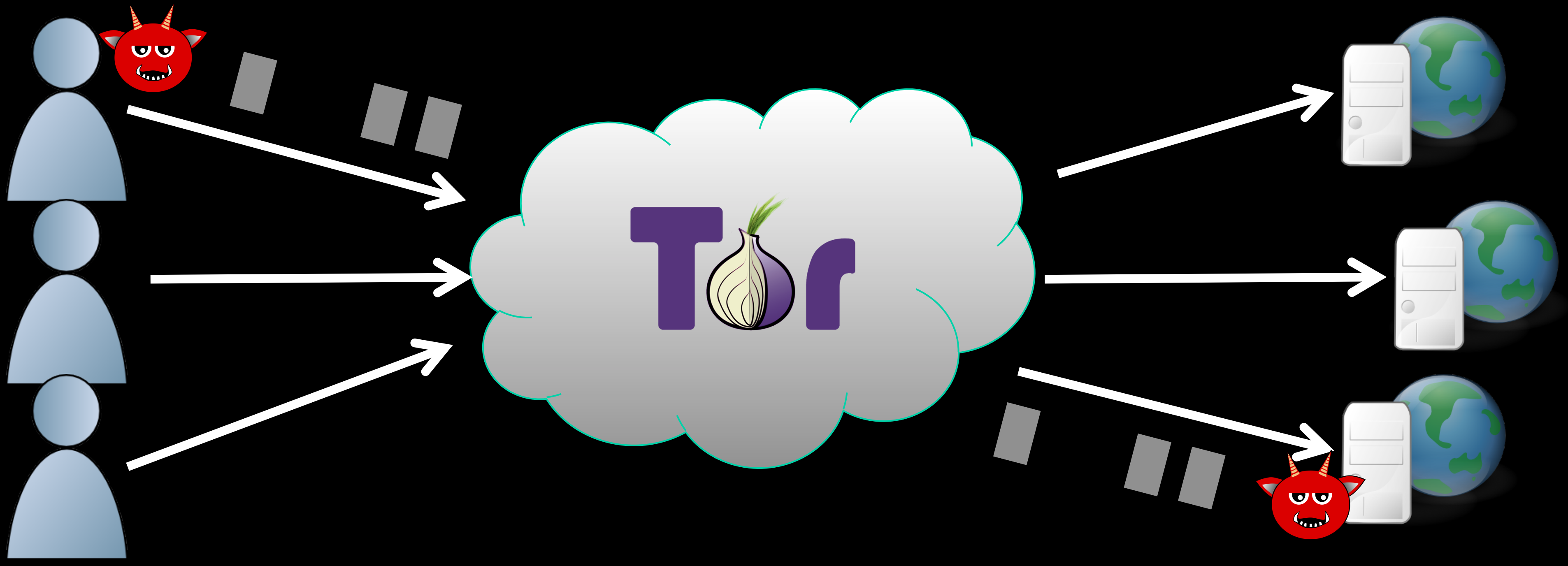
- How does the volunteer resource model affect the vulnerability to correlation attacks?



# Outline

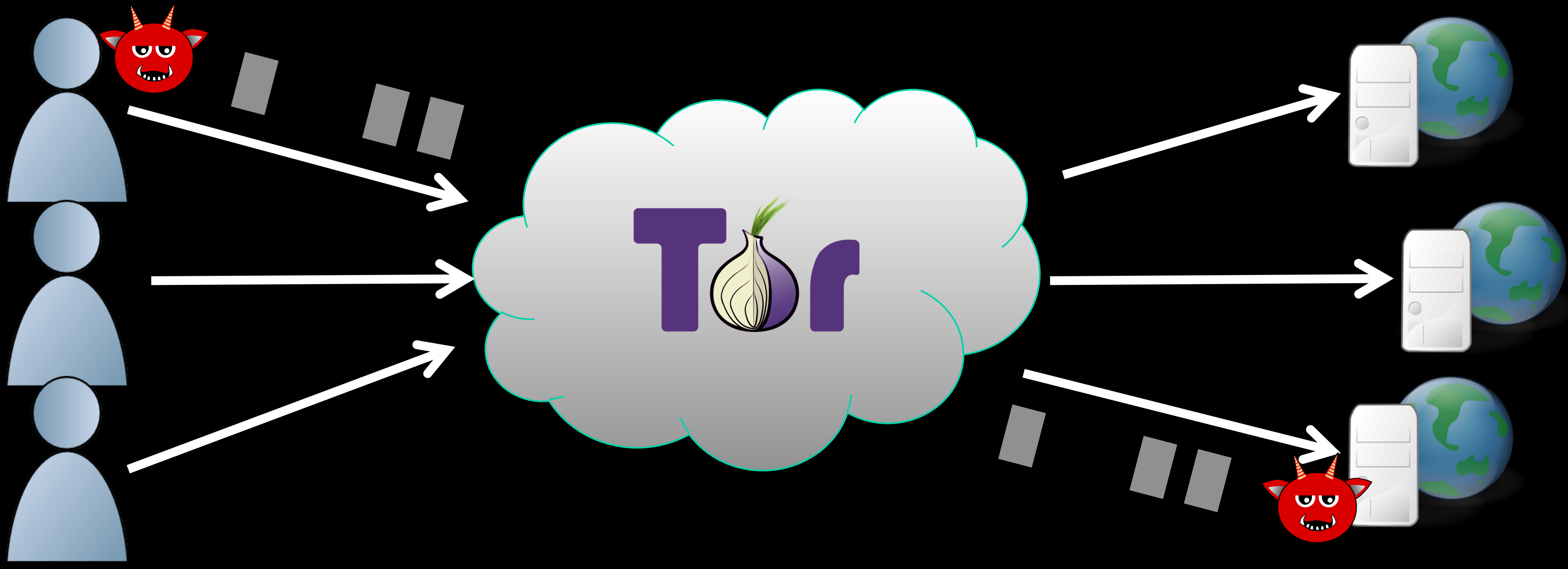
- ~~Background~~
- Security against correlation (end-to-end)
  - Metrics and methodology
  - Node adversaries
  - Link adversaries
- Correlation attacks (partial)
  - Stealthy throughput
  - Induced throttling
    - Traffic admission control
    - Congestion control

# Traffic Correlation




- How can one measure how vulnerable real clients on the real network are to traffic correlation?

# Traffic Correlation



- Is there a difference between targeted correlation and general surveillance?

# Security Metrics

## Principles

- Probability distribution
- Measured on human timescales
- Based on real network and adversaries

# Security Metrics

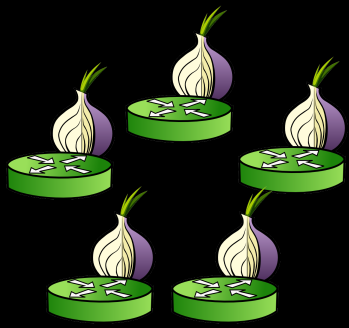
**Principles**

- Probability distribution
- Measured on human timescales
- Based on real network and adversaries

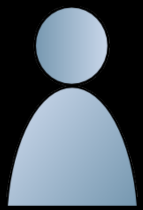
**Metrics (Probability distributions)**

- Time until first path compromise
- Number of path compromises for a given user over given time period

# Approach: Overview

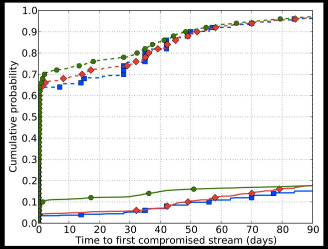


Tor Network Data



User Profiles

Path Simulator




Attack Analysis

# Approach: User Profiles

Consider how users actually use Tor

**Typical**

- Gmail/GChat
- GCal/GDocs
- Facebook
- Web search

**Chat**

- IRC

**File Sharing**

- BitTorrent

Build a 20-minute trace of each activity. Capture destinations/ports visited

# Approach: User Profiles

"Replay" traces to generate streams based on user behavior

**Typical**

- 2632 traces per week
- 205 destinations
- 2 ports

**Chat**

- 135 traces per week
- 1 destinations
- 1 port

**File Sharing**

- 6768 traces per week
- 171 destinations
- 118 ports

# Approach: User Profiles

“Replay” traces to generate streams based on user behavior

| Typical | Chat | File Sharing |
| --- | --- | --- |
| • 2632 traces per week<br>• 205 destinations<br>• 2 ports | • 135 traces per week<br>• 1 destinations<br>• 1 port | • 6768 traces per week<br>• 171 destinations<br>• 118 ports |

- Is the user model accurate?
- What are the challenges?

# User Behavior Affects Relay Selection

Some applications are not well-supported by Tor due to exit policies


**Port 6523**
*Gobby Collaborative Editor*

Permitted by 20% of exits measured by bandwidth

**Port 443**
*HTTPS*

Permitted by 93% of exits measured by bandwidth

# Approach: Tor Network Data

Consider the Tor network as it changes over a long period of time:

- Relays join and leave
- Bandwidth changes
- Exit/Guard designations change

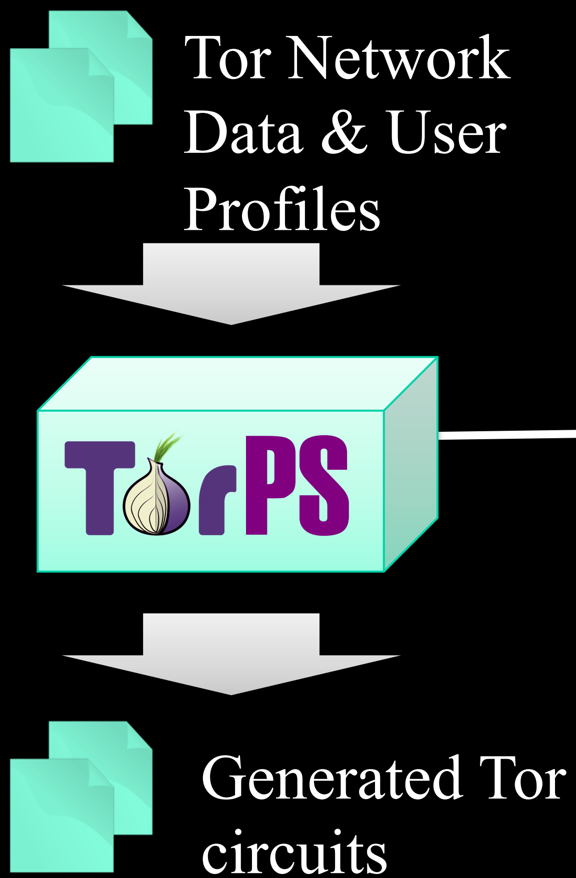
Use Tor Project archives to obtain state of network over 3 to 6 months

Hourly consensuses

Monthly server descriptors

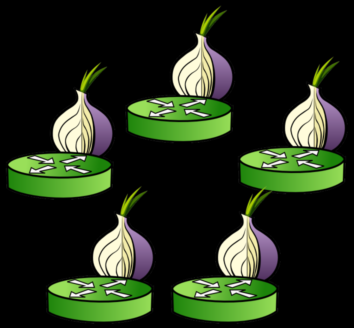
# Approach: Simulate Tor with TorPS

Combine User and Tor Network models using TorPS to produce the circuits Tor would use

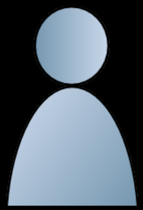



- Re-implements path selection
- Based on Tor stable version (0.2.3.25)
- Considers:
  - Bandwidth weighting
  - Exit policies
  - Guards and guard rotation
  - Hibernation
  - /16 and family conflicts
- Omits effects of network performance

# Approach: Overview



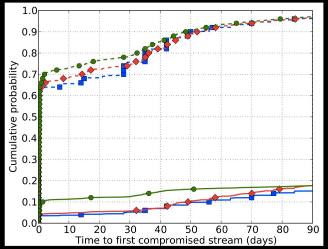
Tor Network Data



User Profiles

TorPS

Path Simulator

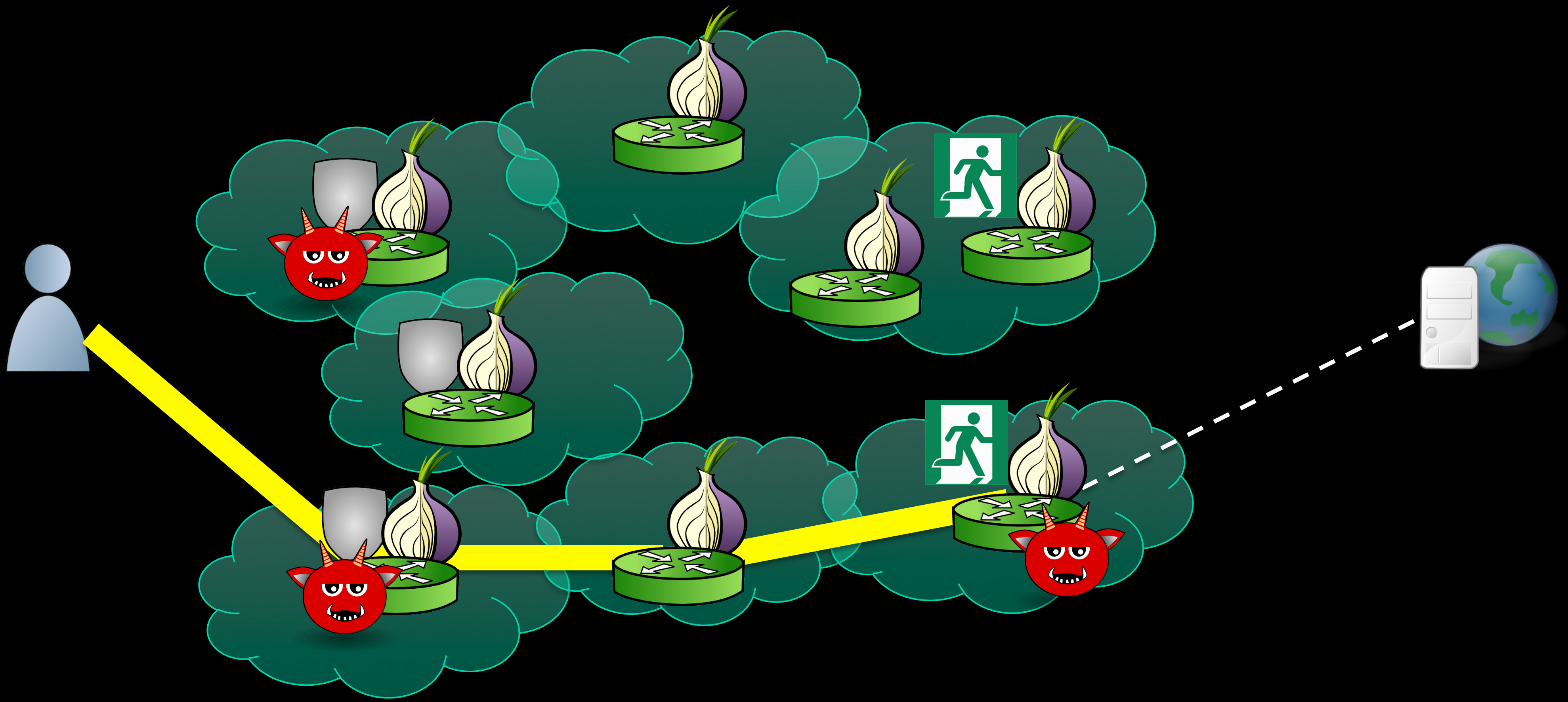



Attack Analysis

# Outline

- ~~Background~~
- Security against correlation (end-to-end)
  - ~~Metrics and methodology~~
  - Node adversaries
  - Link adversaries
- Correlation attacks (partial)
  - Stealthy throughput
  - Induced throttling
    - Traffic admission control
    - Congestion control

# Node Adversary



# Node Adversary

Controls a fixed allotment of relays based on bandwidth budget

- We assume adversary has 100 MiB/s – comparable to large family of relays

- Adversaries apply 5/6th of bandwidth to guard relays and the rest to exit relays. (We found this to be the most effective allocation we tested.)

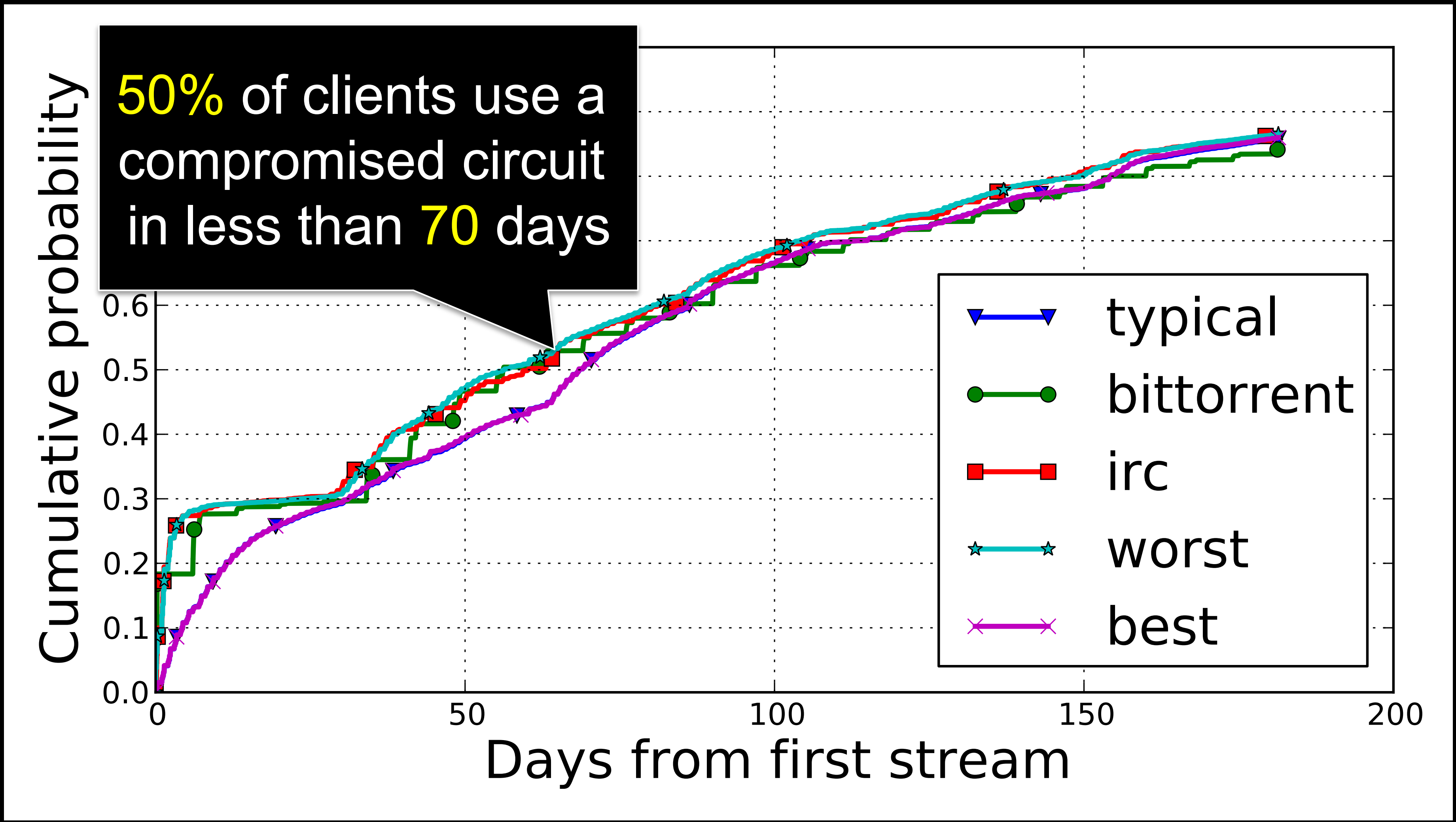
# Node Adversary

Controls a fixed allotment of relays based on bandwidth budget

- We assume adversary has 100 MiB/s – comparable to large family of relays
- Adversaries apply 5/6th of bandwidth to guard relays and the rest to exit relays. (We found this to be the most effective allocation we tested.)

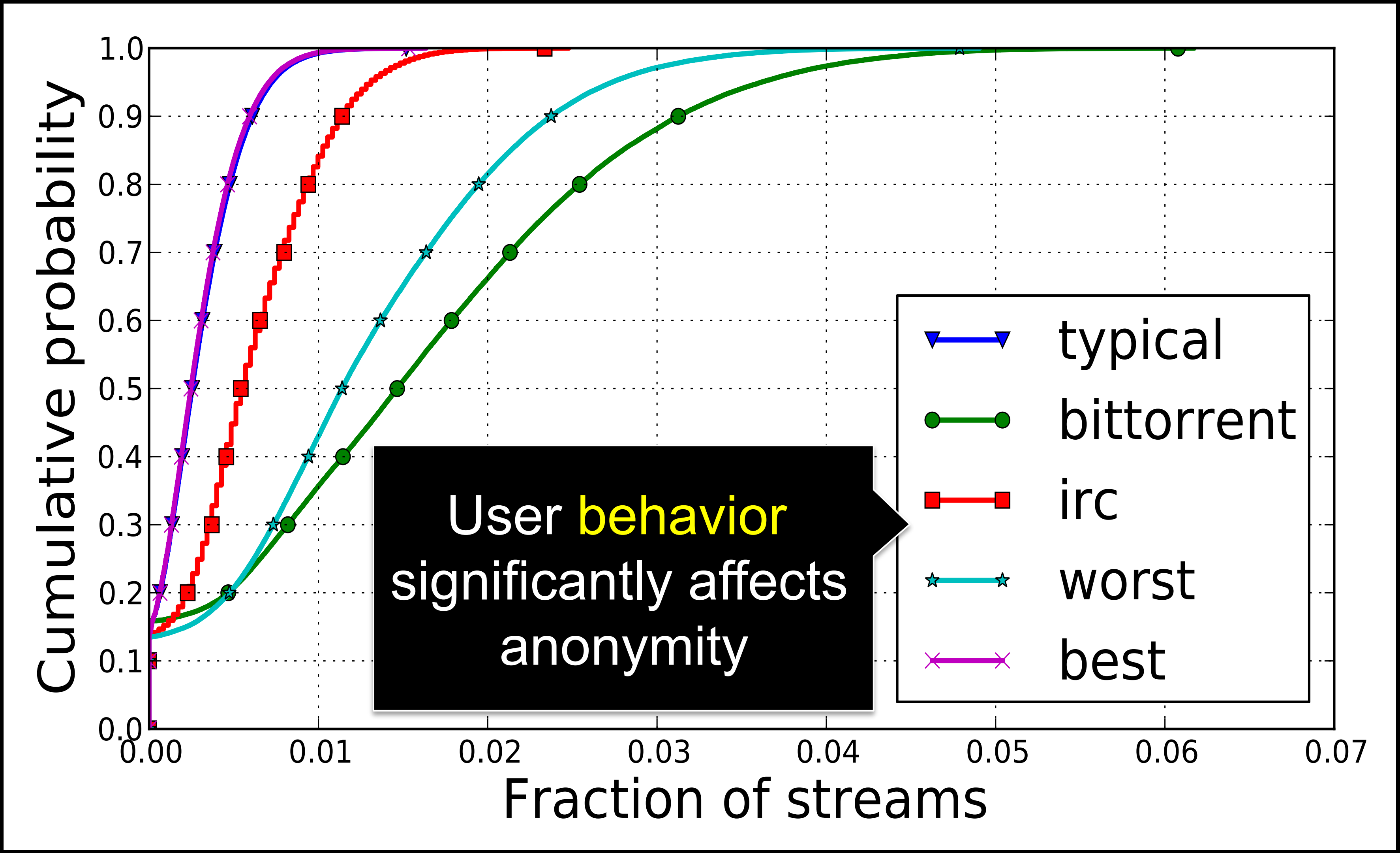
- Is 100 MiB/s realistic for an adversary?

# Time to First Compromised Circuit




October 2012 – March 2013

# Fraction of Compromised Streams

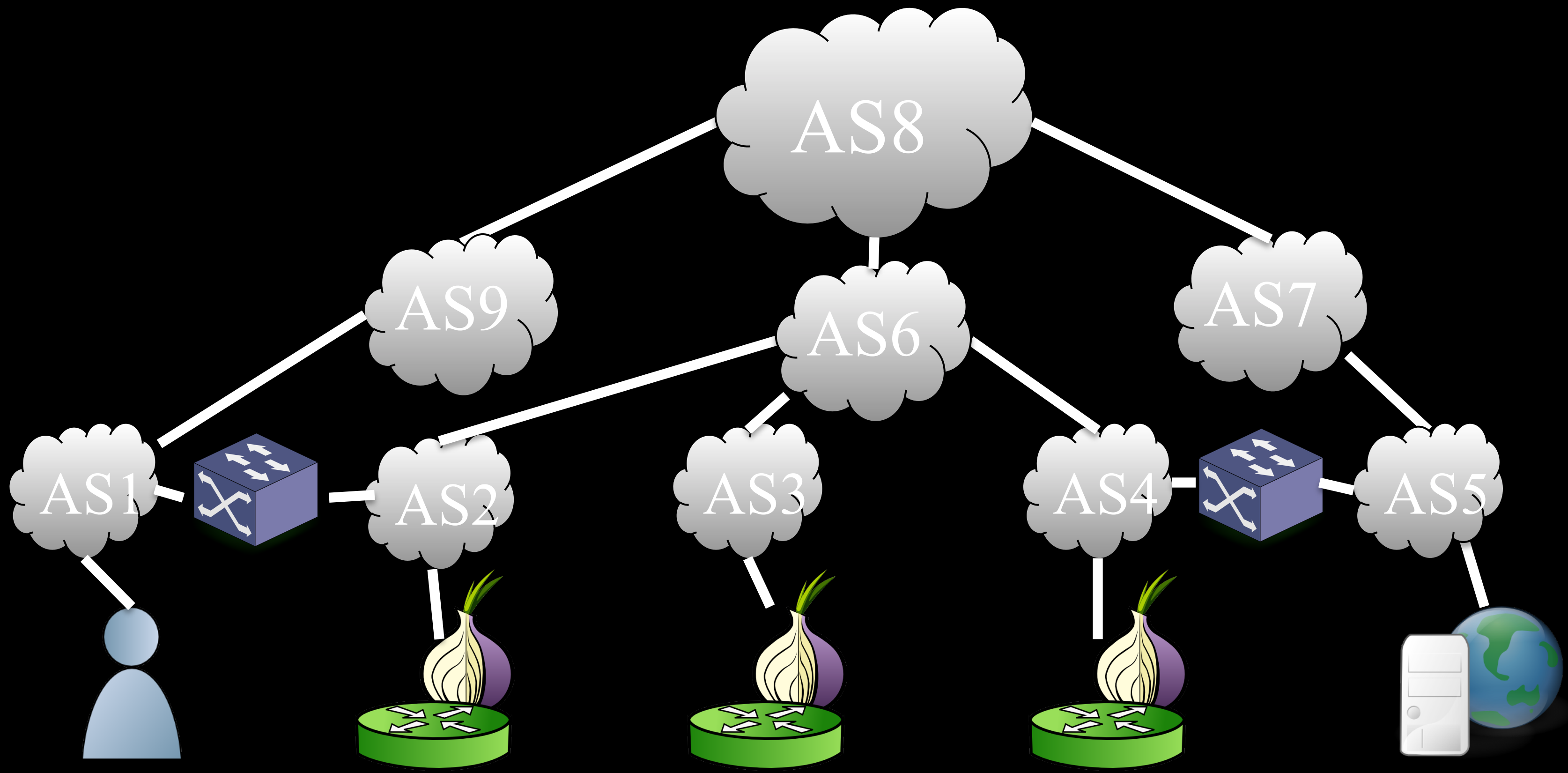



October 2012 – March 2013

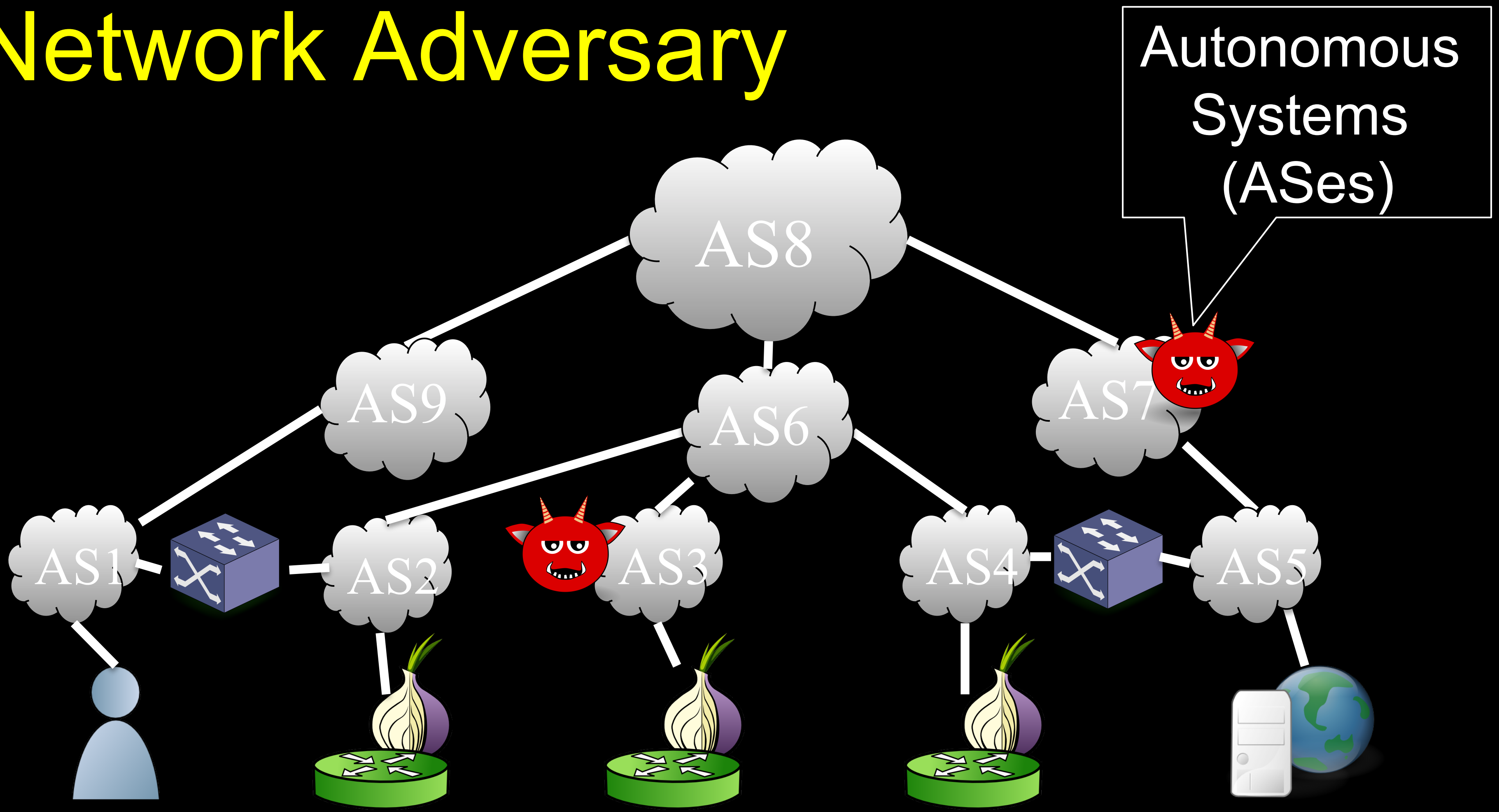
# Outline

- ~~Background~~
- Security against correlation (end-to-end)
  - ~~Metrics and methodology~~
  - ~~Node adversaries~~
  - Link adversaries
- Correlation attacks (partial)
  - Stealthy throughput
  - Induced throttling
    - Traffic admission control
    - Congestion control

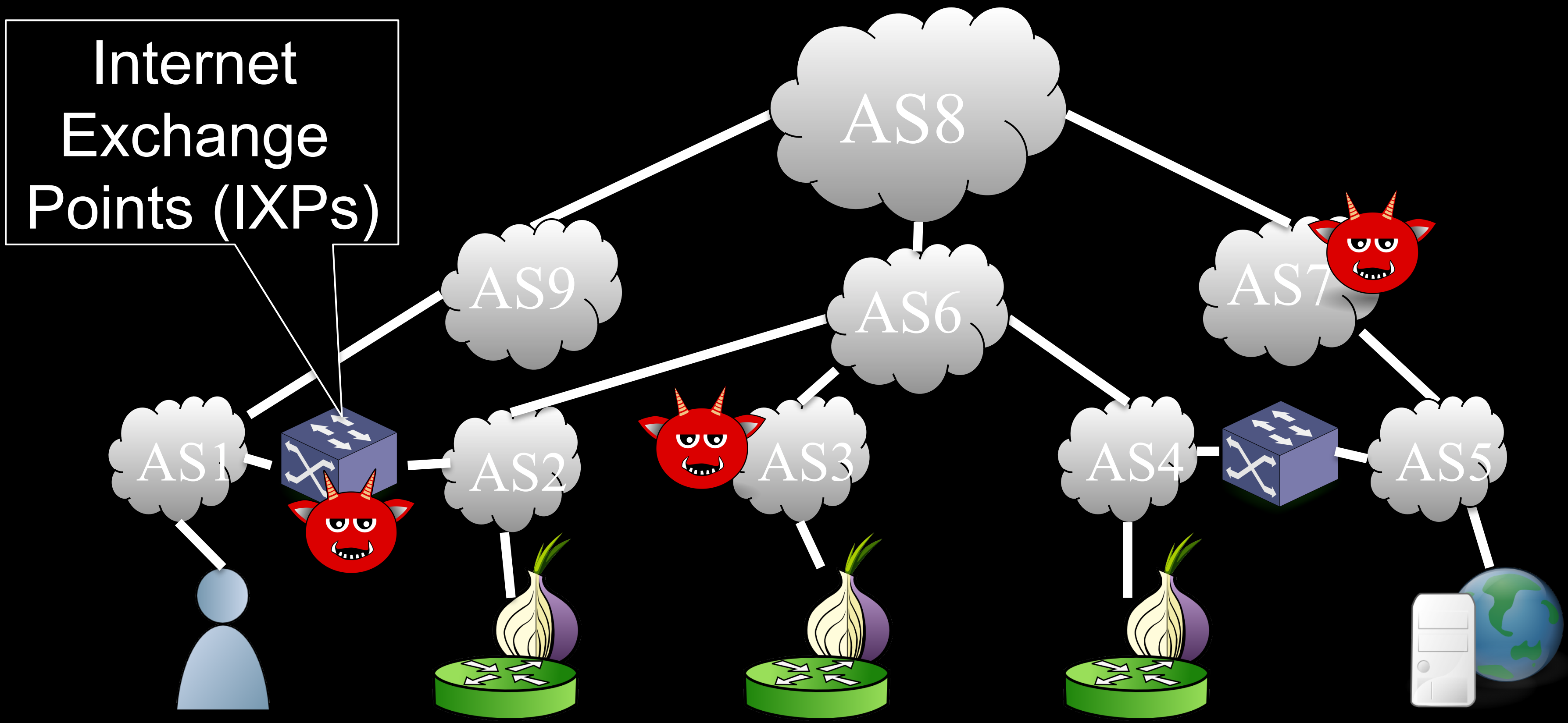
# Network Adversary



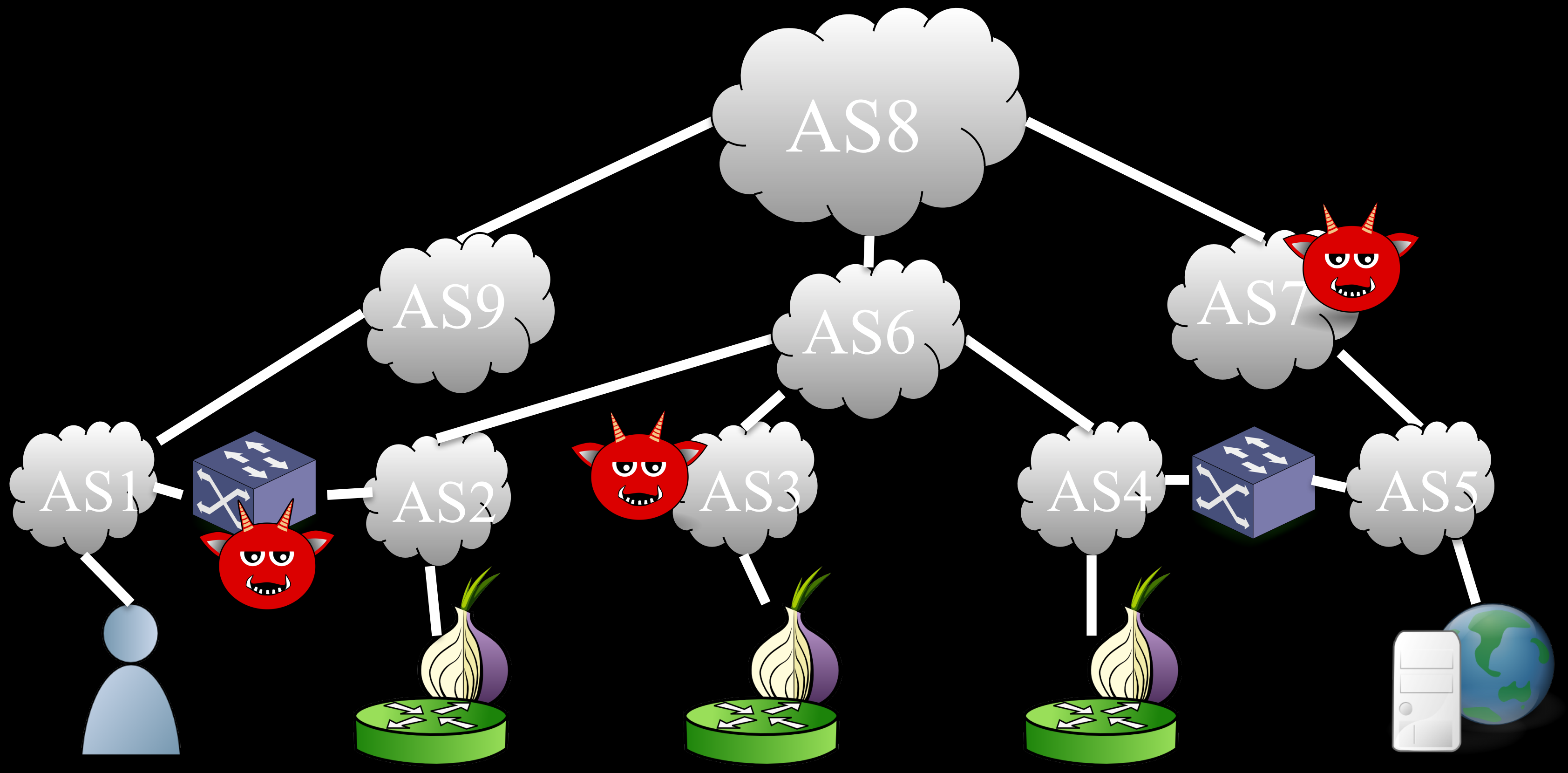
# Network Adversary



# Network Adversary

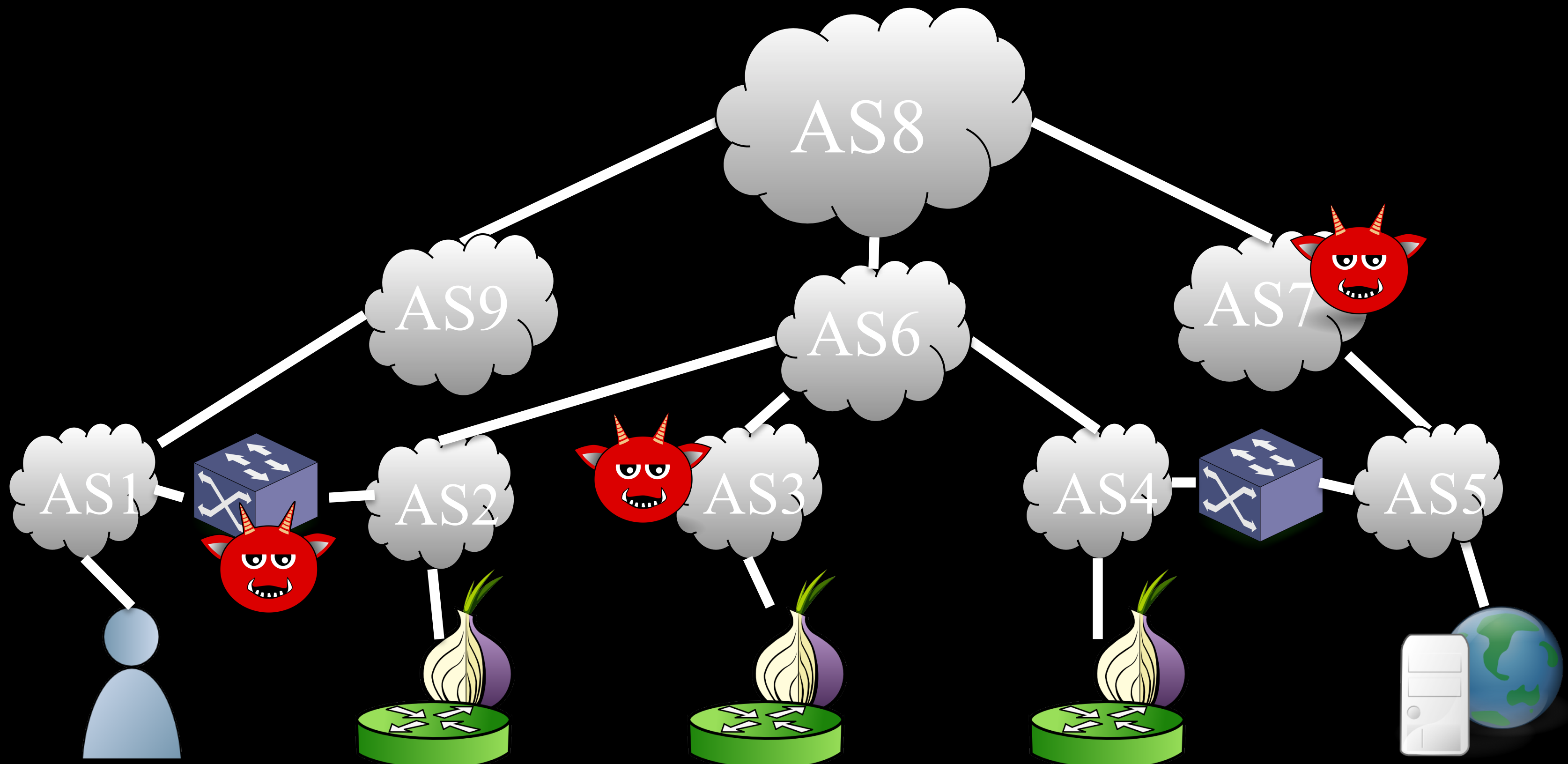


# Network Adversary




- Adversary has fixed location
- Adversary may control multiple entitites

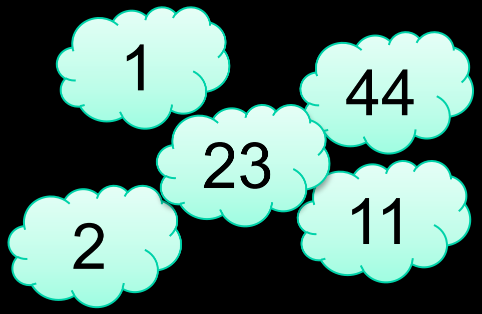
# Network Adversary

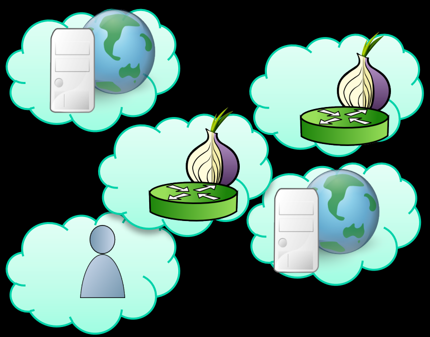


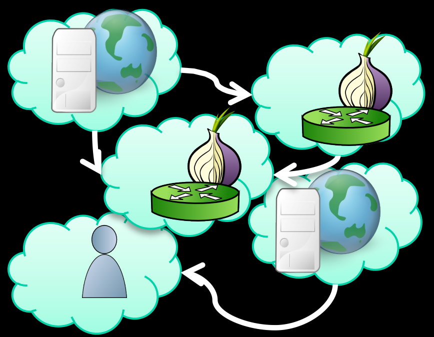

- Should most users be concerned with a network adversary?

- Adversary has fixed location
- Adversary may control multiple entitites

# Simulating a Network Adversary

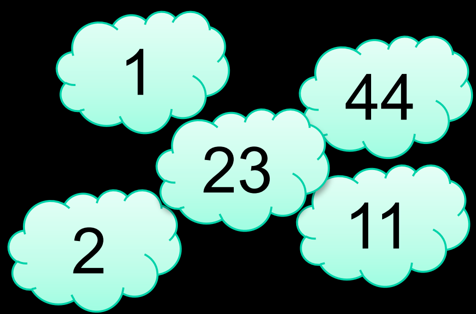


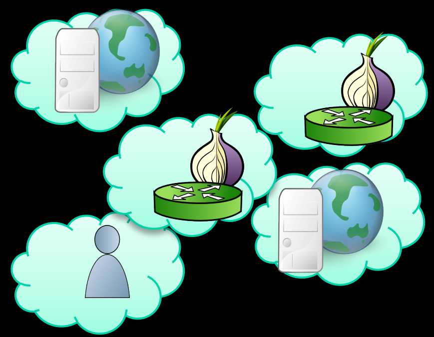



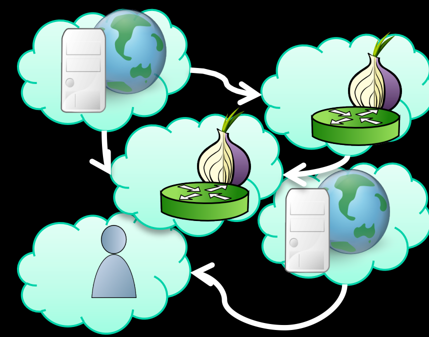


Build AS-level Graph (CAIDA)

# Simulating a Network Adversary



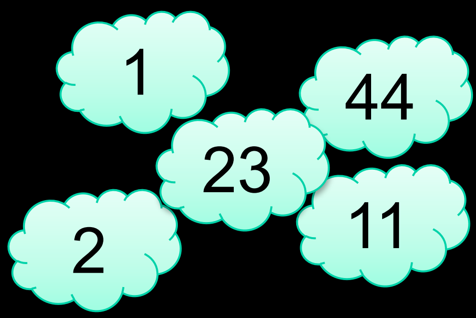



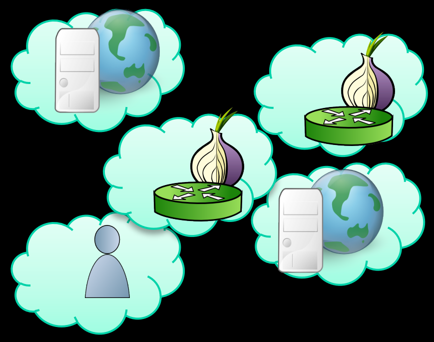


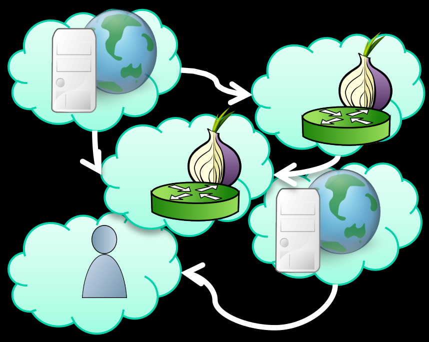
Build AS-level Graph (CAIDA)

Place points of interest (Maxmind, traces)

# Simulating a Network Adversary





Build AS-level Graph (CAIDA)

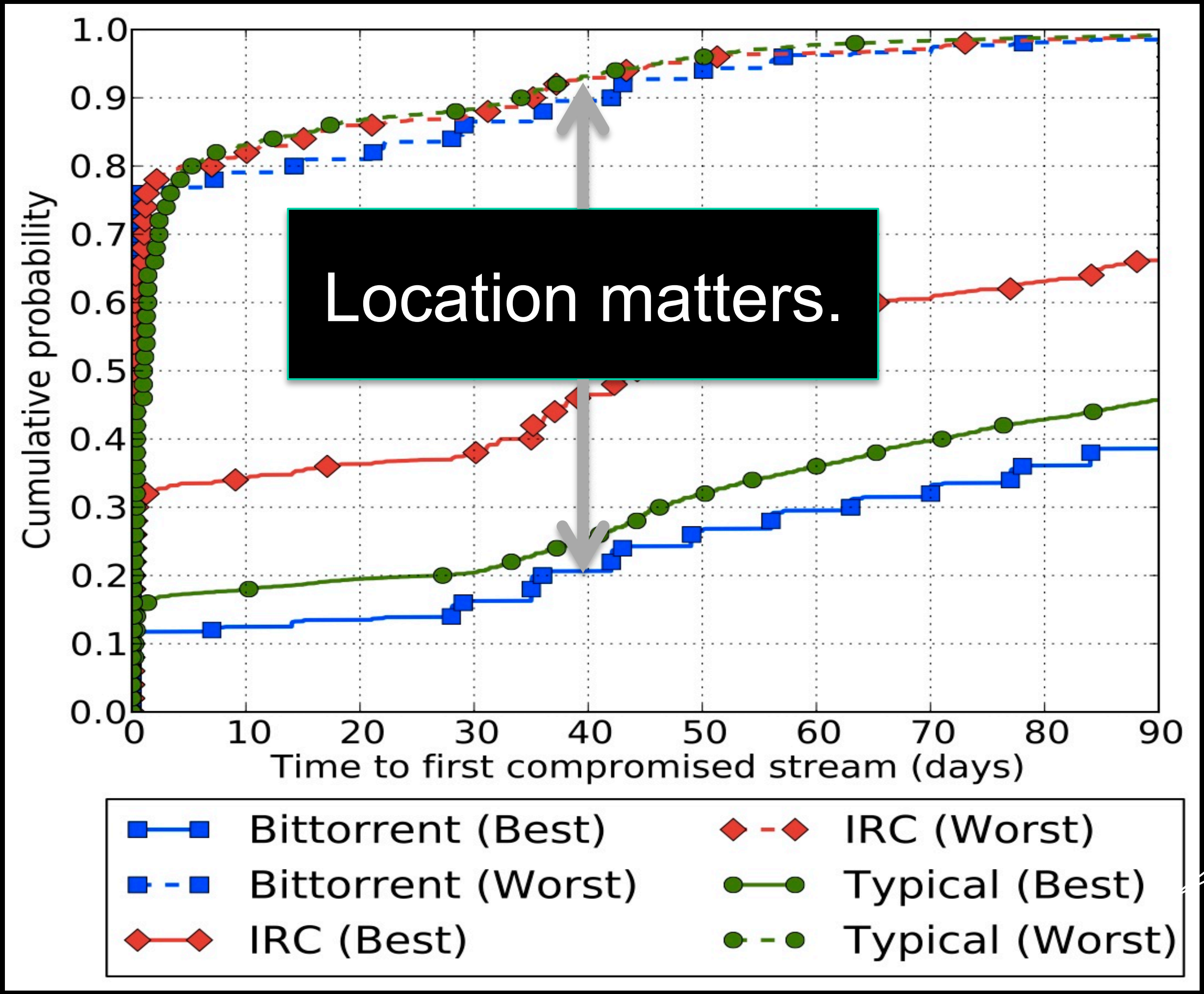
Place points of interest (Maxmind, traces)

Find AS-level routes (Gao'02, CAIDA)

# Selecting Network Adversaries

1. Rank each AS/IXP for each client location by frequency on entry or exit paths;
2. Exclude src/dst ASes (compromises nearly all paths); and
3. Assign adversary to top *k* ASes or IXPs

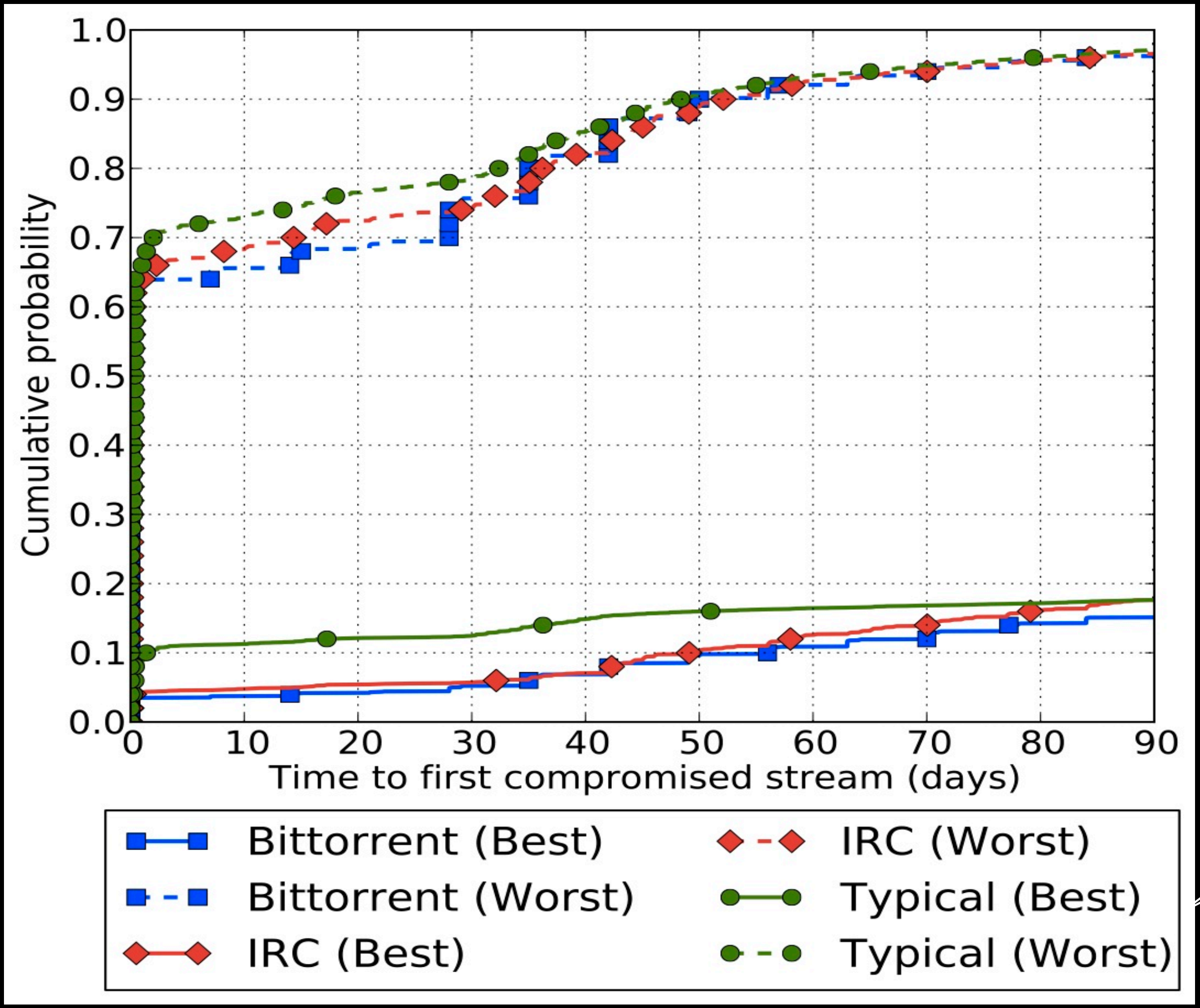
# Adversary Controls One AS




"best"/"worst" denote most/least secure client

January 2013 – March 2013

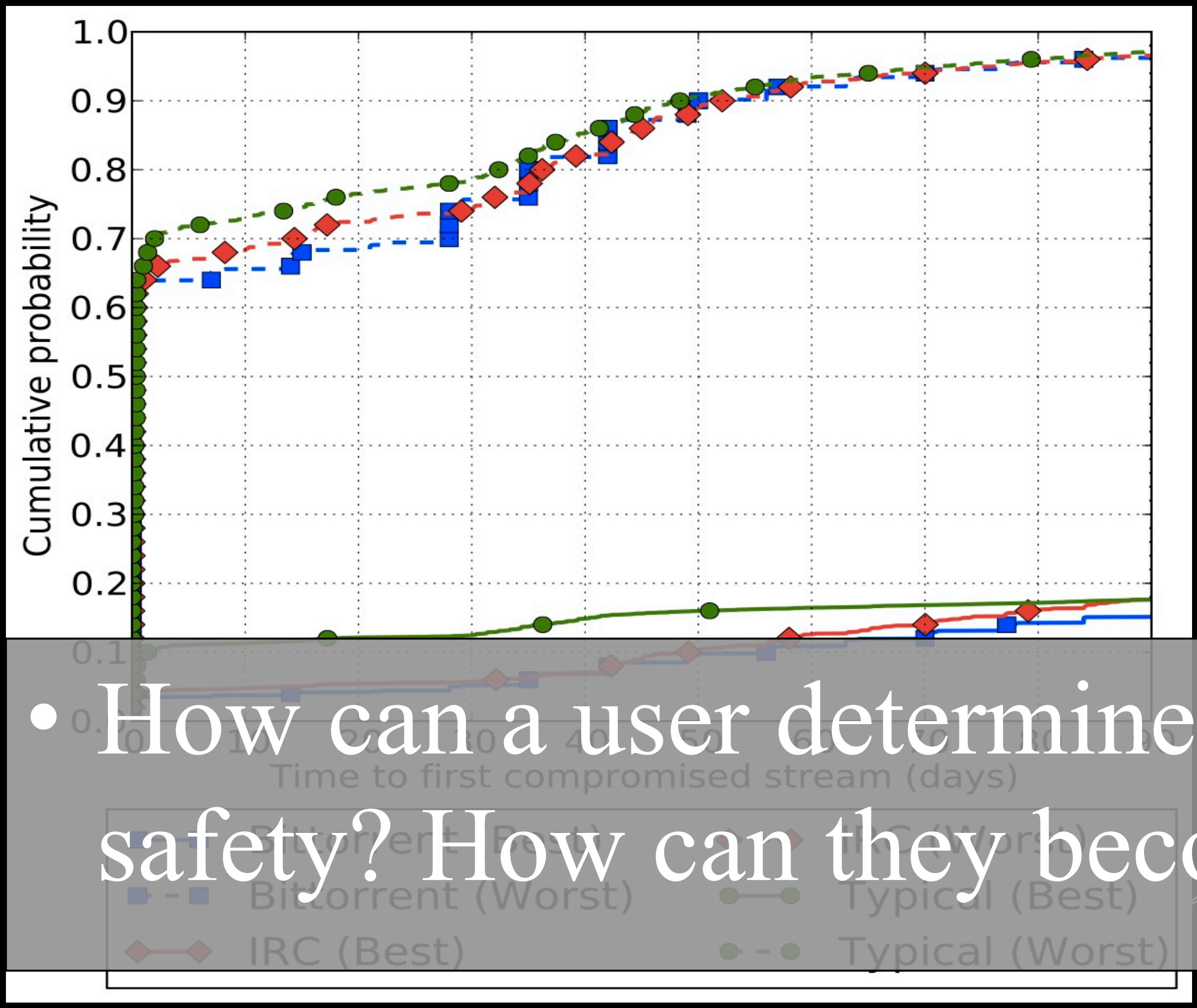
# Adversary Controls One IXP Organization




“best”/“worst” denote most/least secure client

January 2013 – March 2013

# Adversary Controls One IXP Organization

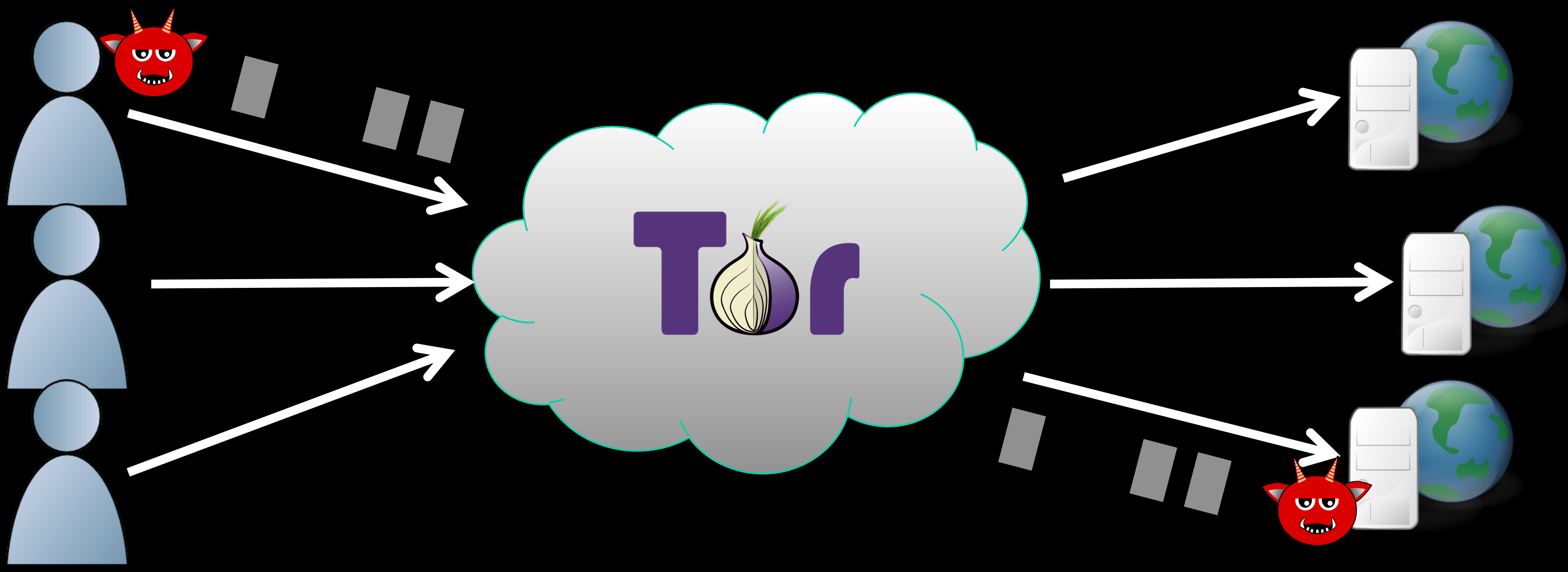



"best"/"worst" denote most/least secure client

- How can a user determine their safety? How can they become safer?

January 2013 – March 2013

# Traffic Correlation

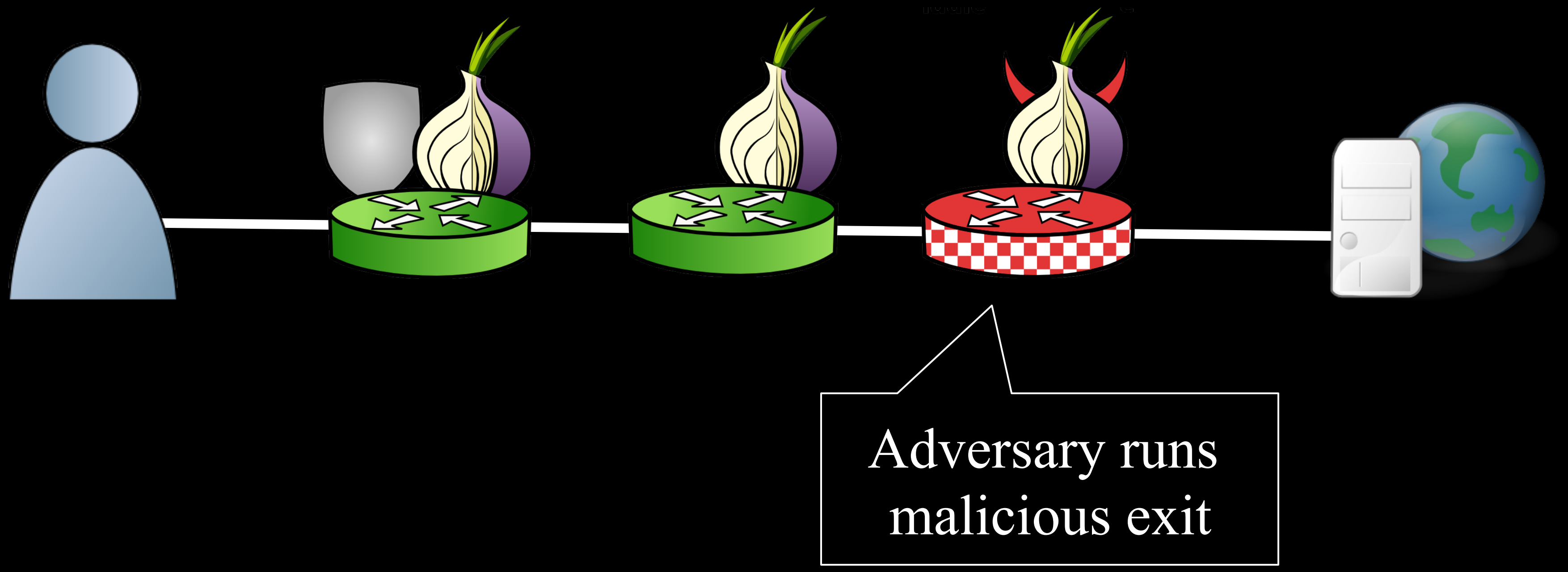


- What if the adversary only controls one of the ends?

# Outline

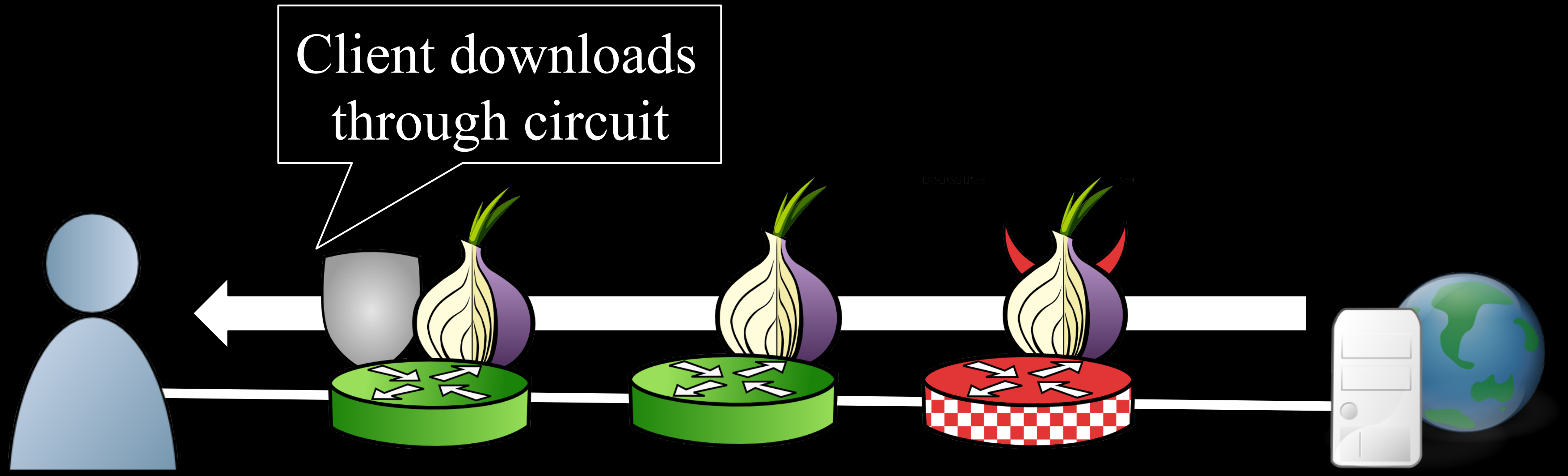
- ~~Background~~
- ~~Security against correlation (end-to-end)~~
  - ~~Metrics and methodology~~
  - ~~Node adversaries~~
  - ~~Link adversaries~~
- Correlation attacks (partial)
  - Stealthy throughput
  - Induced throttling
    - Traffic admission control
    - Congestion control

# Traffic Correlation: Throughput



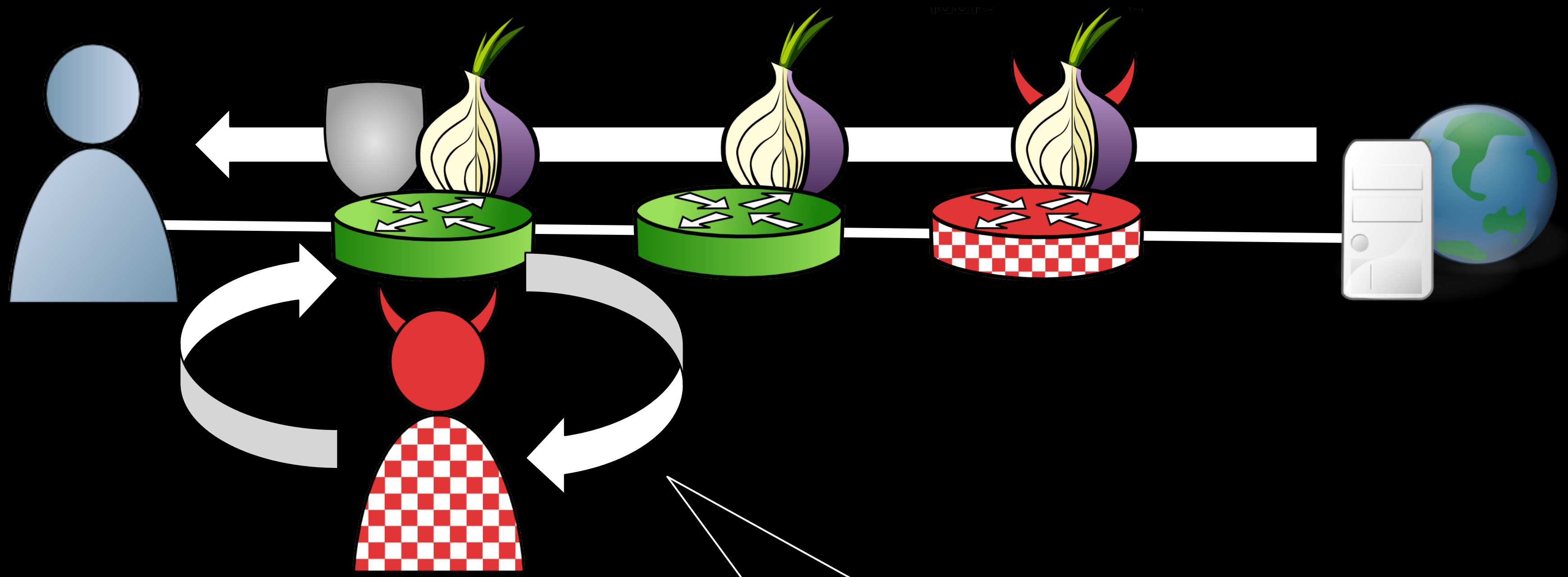

Mittal et.al. CCS'11

# Traffic Correlation: Throughput




Mittal et.al. CCS'11

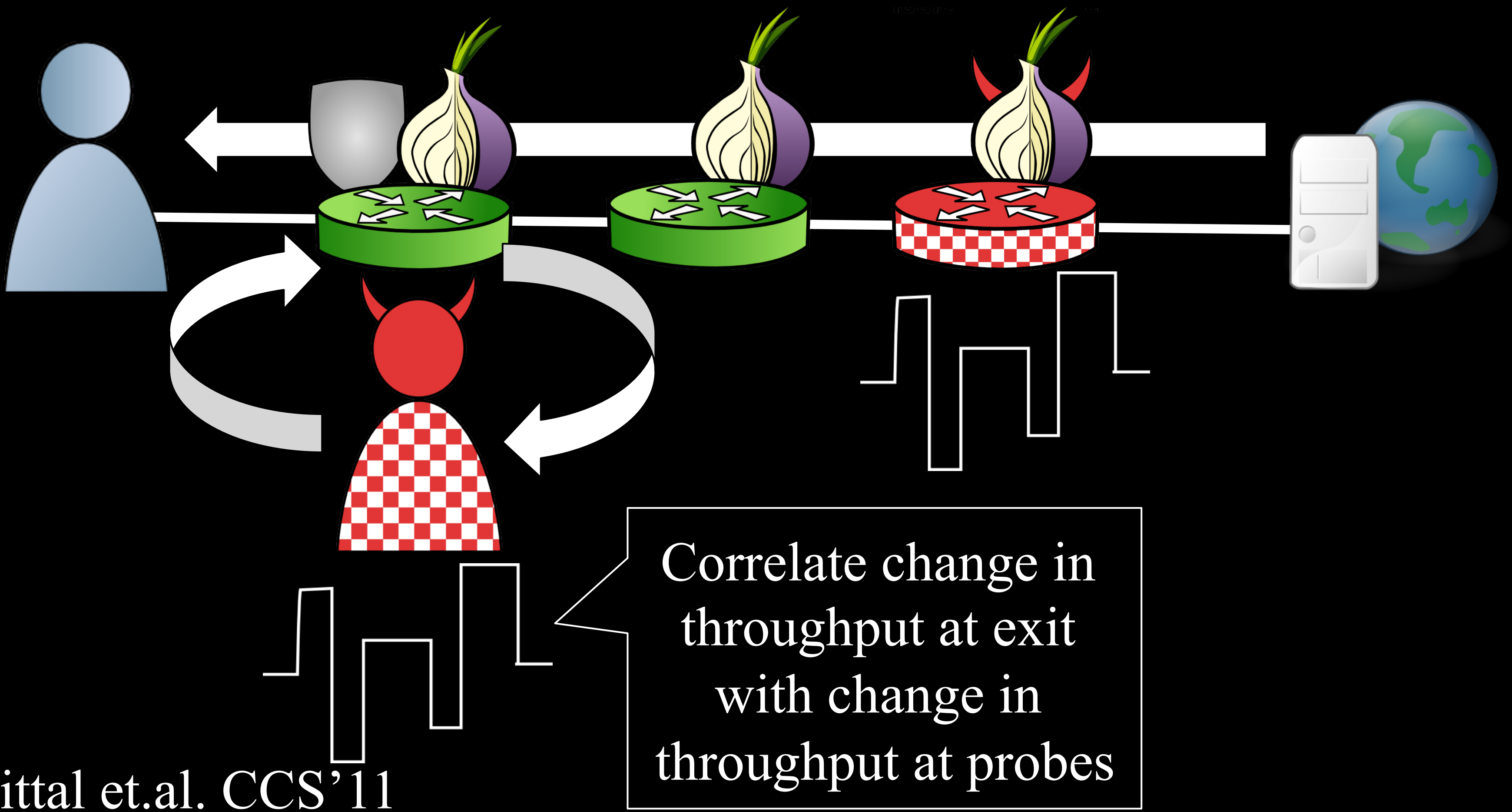
# Traffic Correlation: Throughput



Probes download through all guards

Mittal et.al. CCS'11

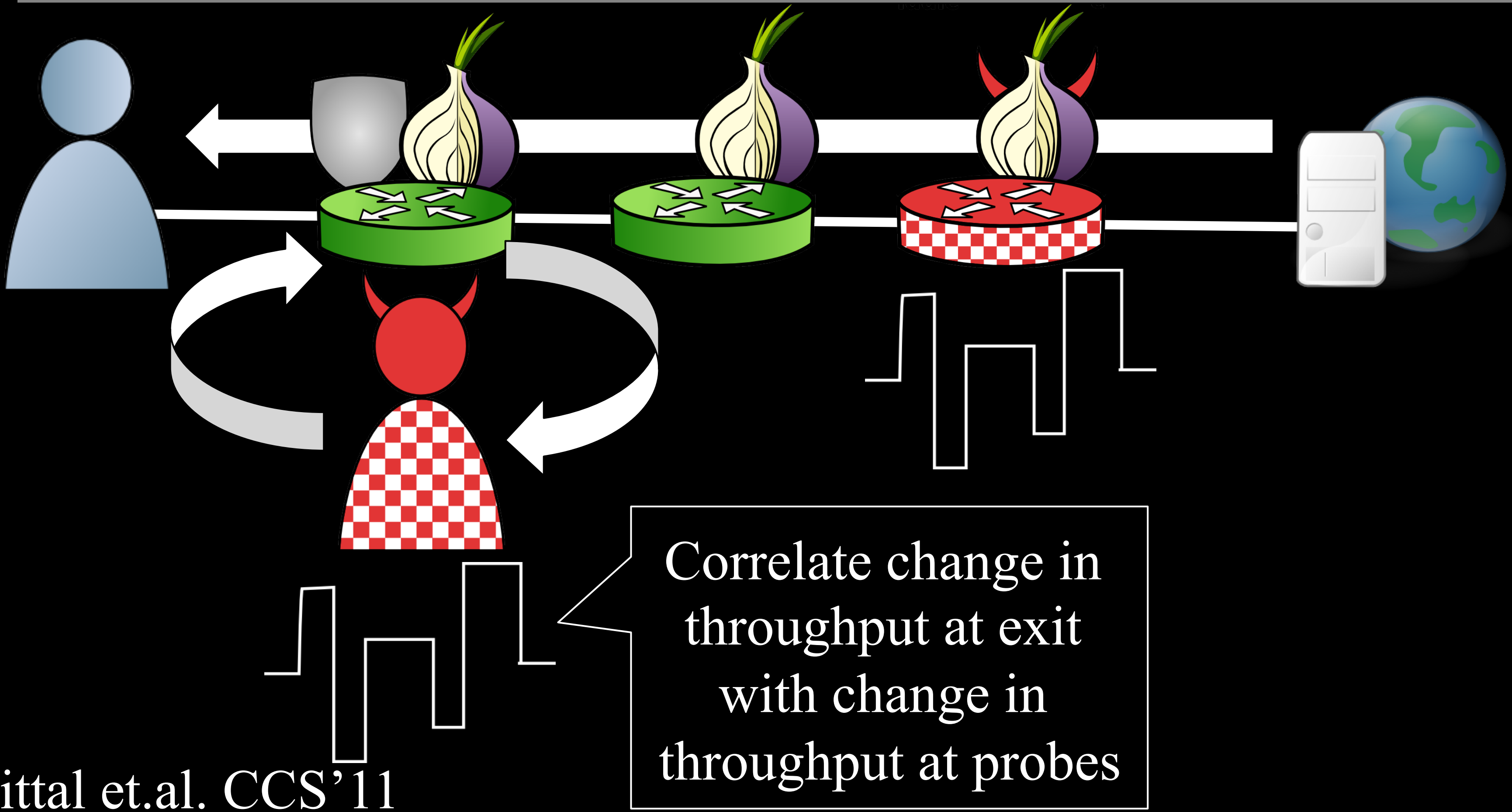
# Traffic Correlation: Throughput




Mittal et.al. CCS'11

# Traffic Correlation: Throughput

- How is this attack "stealthy"?

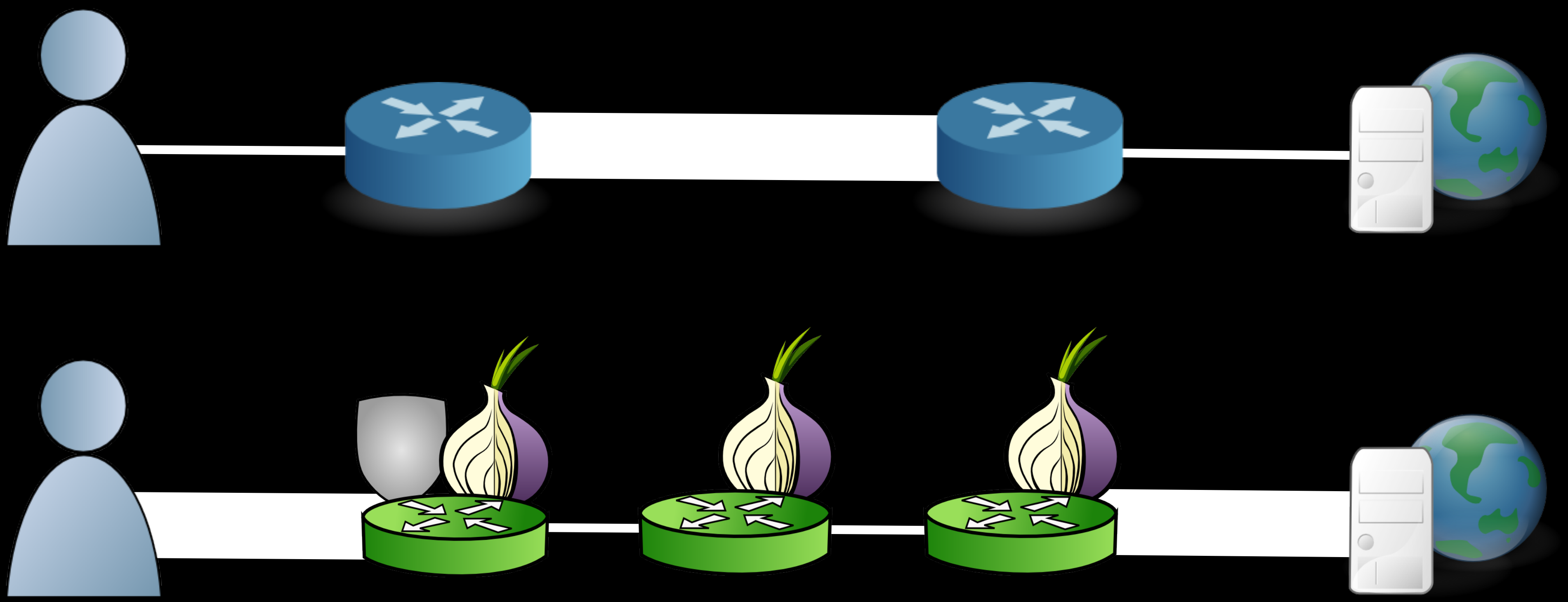



Mittal et.al. CCS'11

# Outline

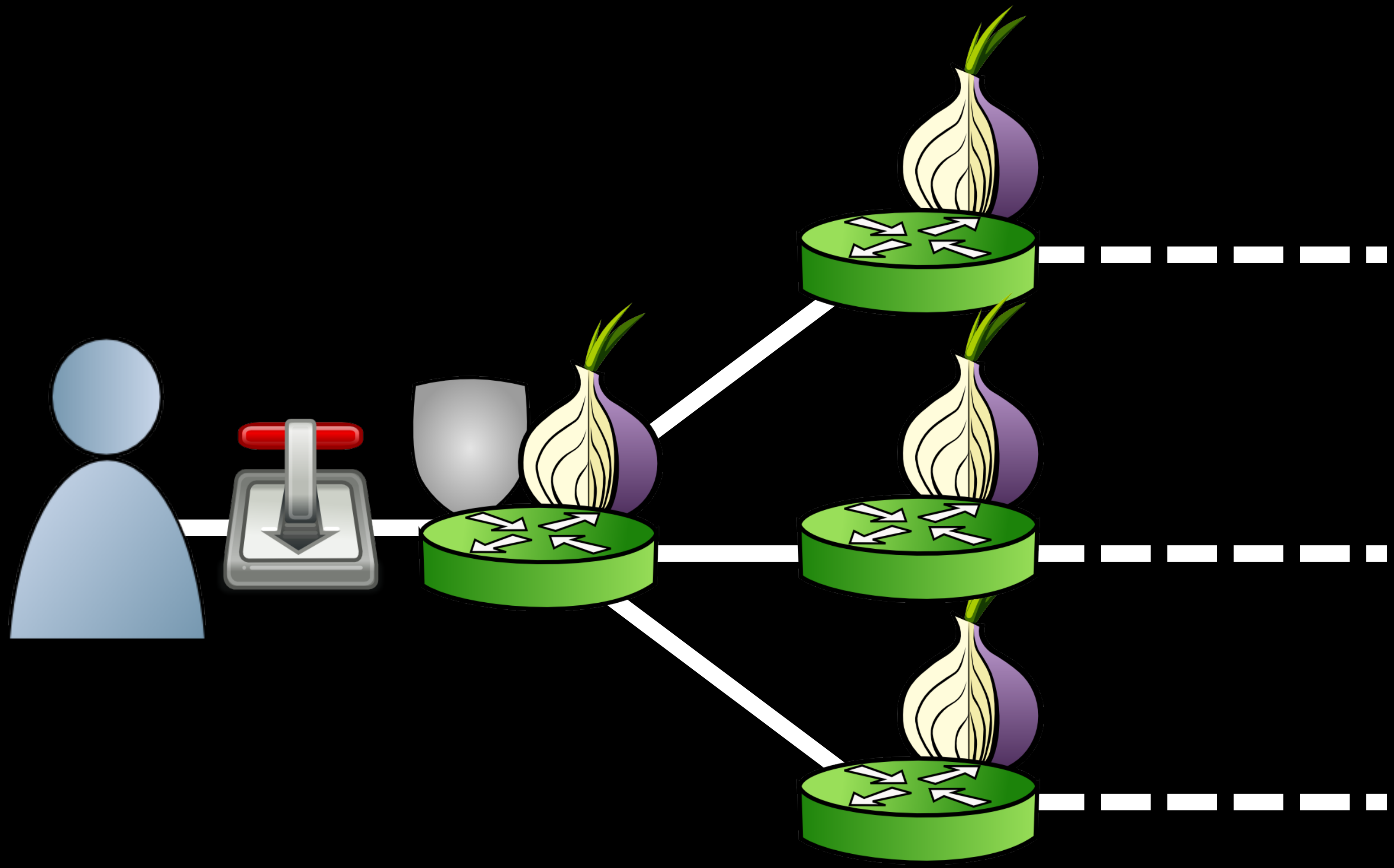
- ~~Background~~
- ~~Security against correlation (end-to-end)~~
  - ~~Metrics and methodology~~
  - ~~Node adversaries~~
  - ~~Link adversaries~~
- Correlation attacks (partial)
  - ~~Stealthy throughput~~
  - Induced throttling
    - Traffic admission control
    - Congestion control

# Tor != Internet

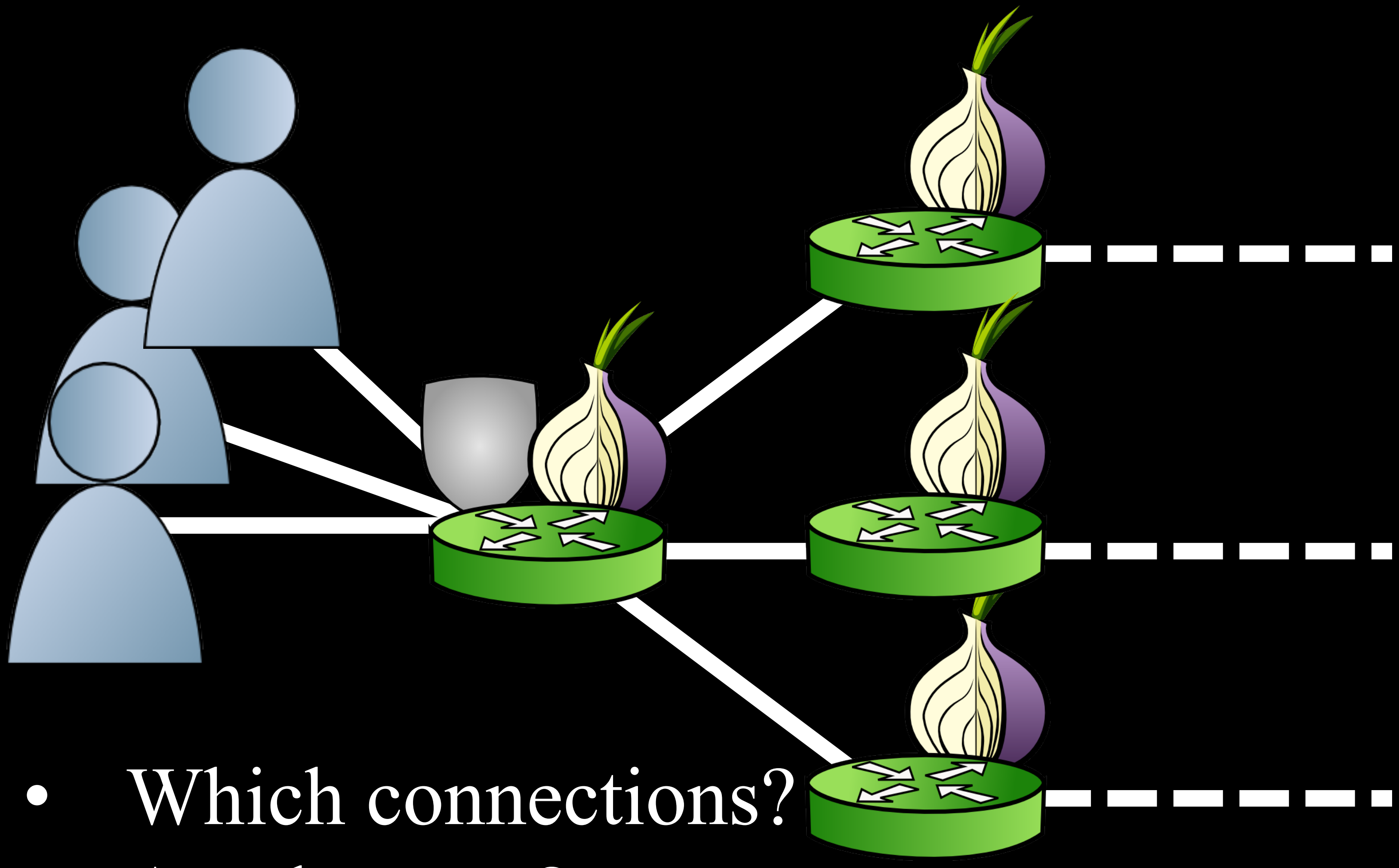


- Specialized Tor performance enhancements
  - Reducing load: traffic admission control
  - Reducing load, improving utilization: congestion control

# Traffic Admission Control

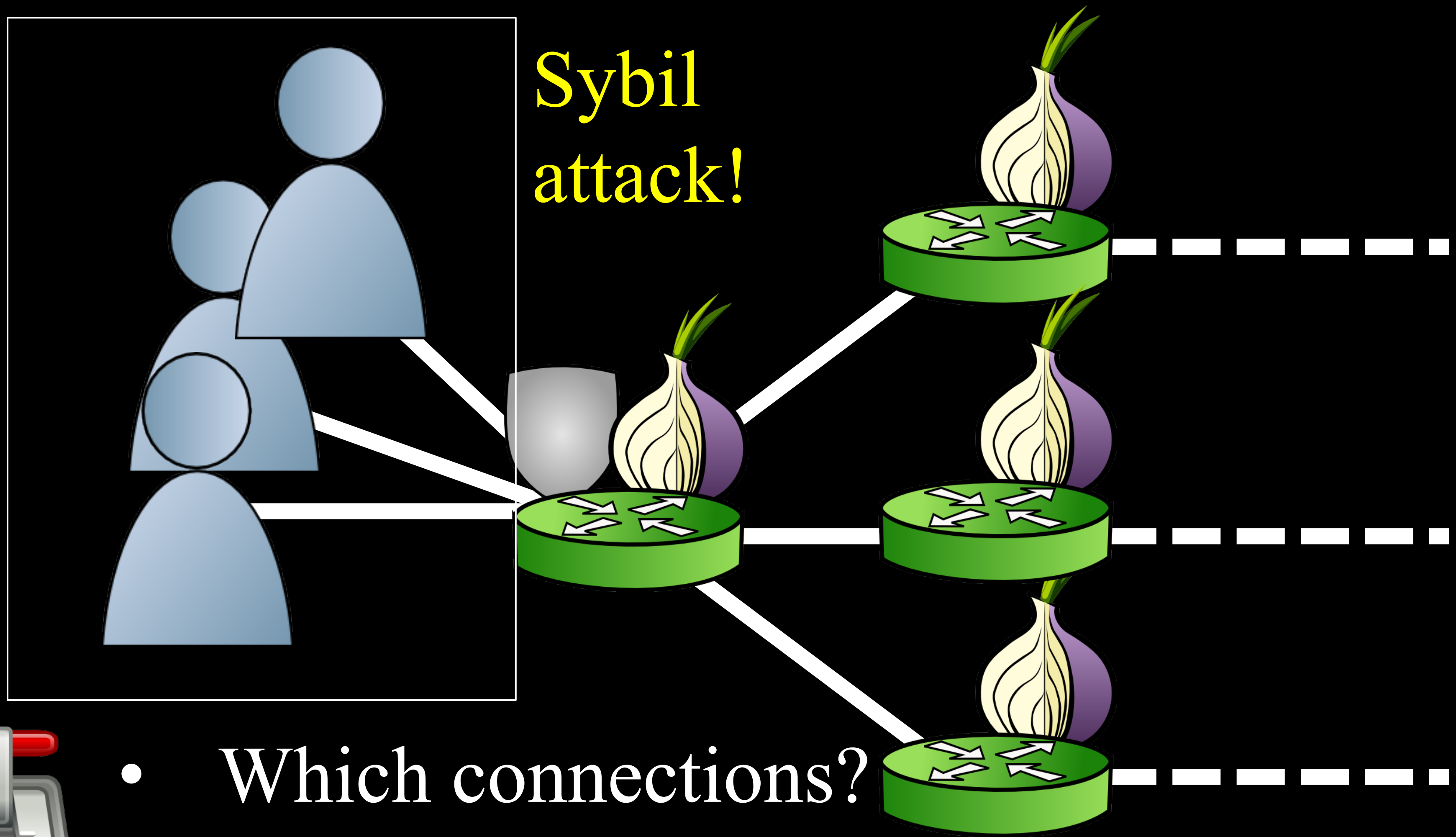


# Traffic Admission Control



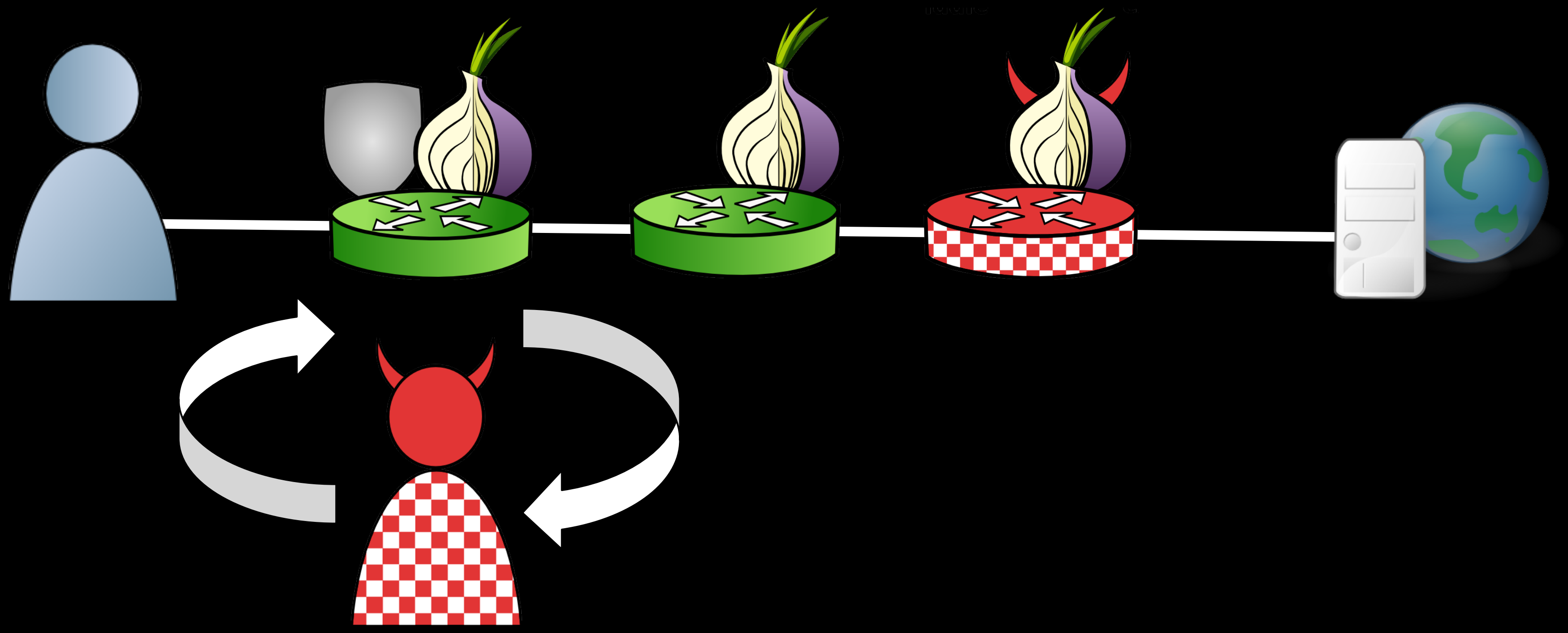
- Which connections?
- At what rate?

# Traffic Admission Control



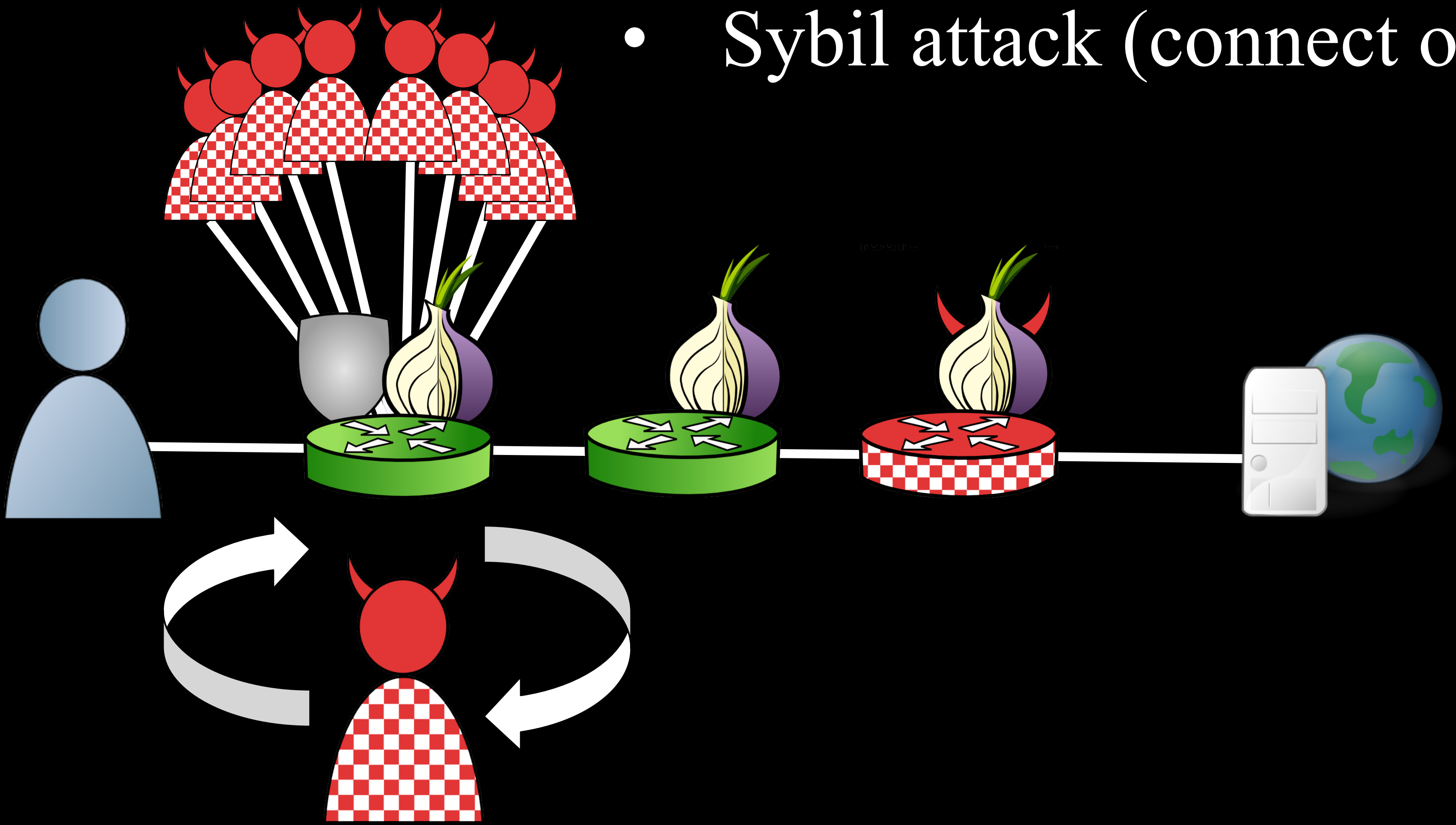

- Which connections?
- At what rate?

# Traffic Admission Control



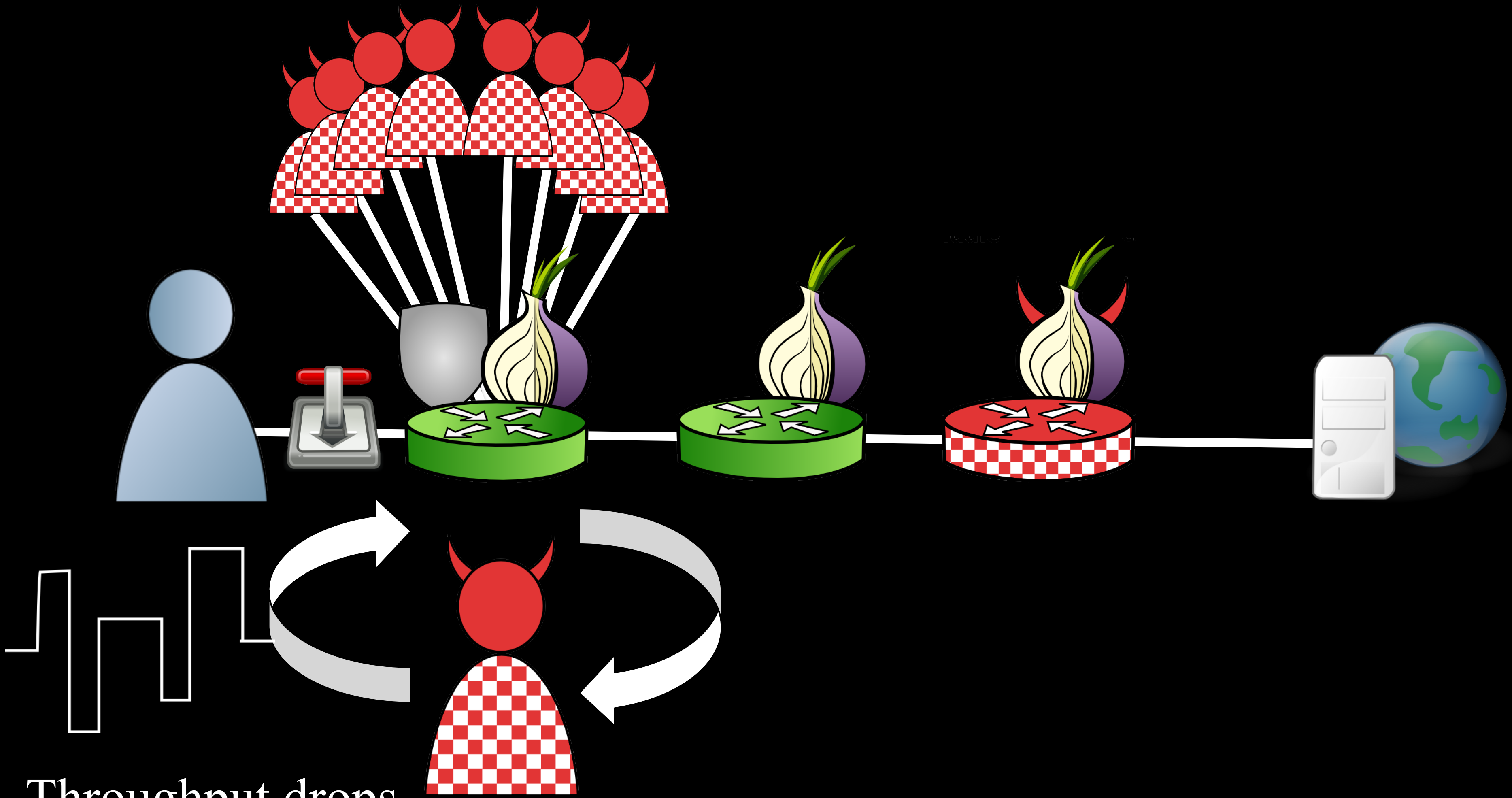
# Traffic Admission Control

- Sybil attack (connect only)



Geddes et.al.
PETS'13

# Traffic Admission Control

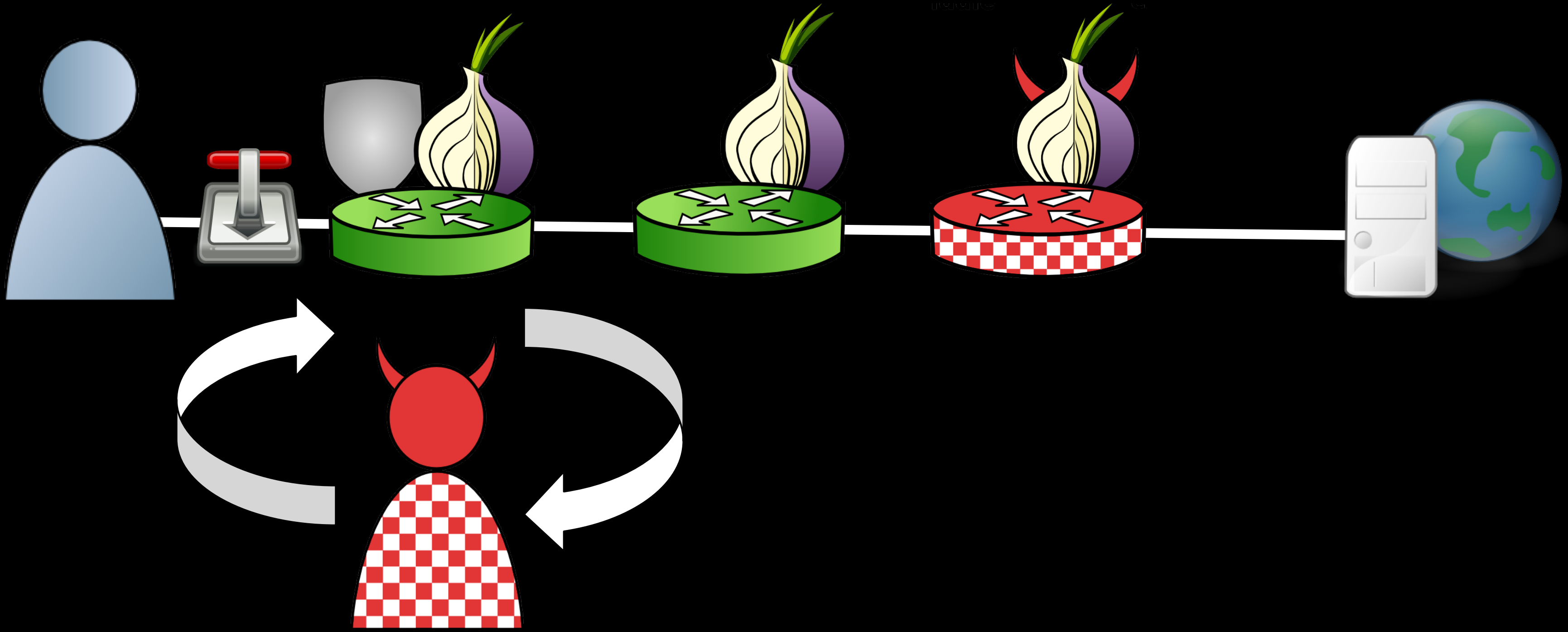


Throughput drops to throttle rate

Geddes et.al. PETS’13

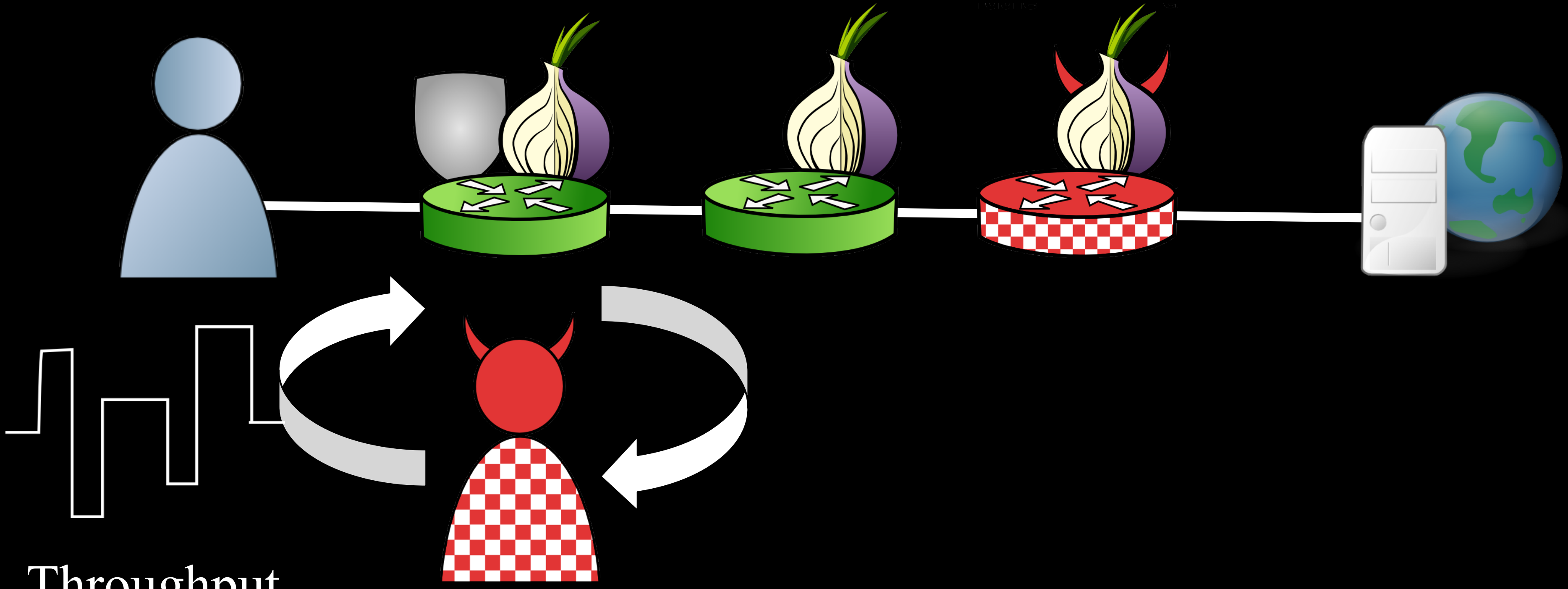
# Traffic Admission Control

- Disconnect sybils



Geddes et.al.
PETS'13

# Traffic Admission Control

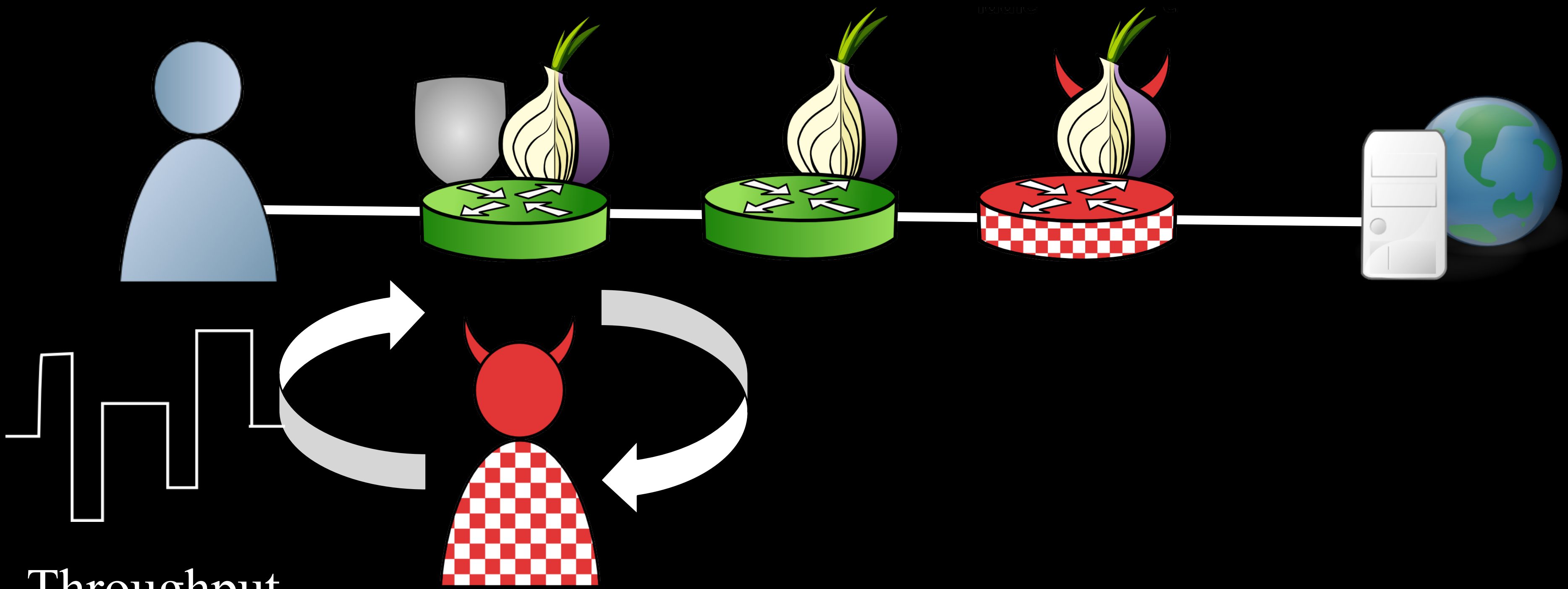


Throughput increases

Geddes et.al. PETS'13

# Traffic Admission Control

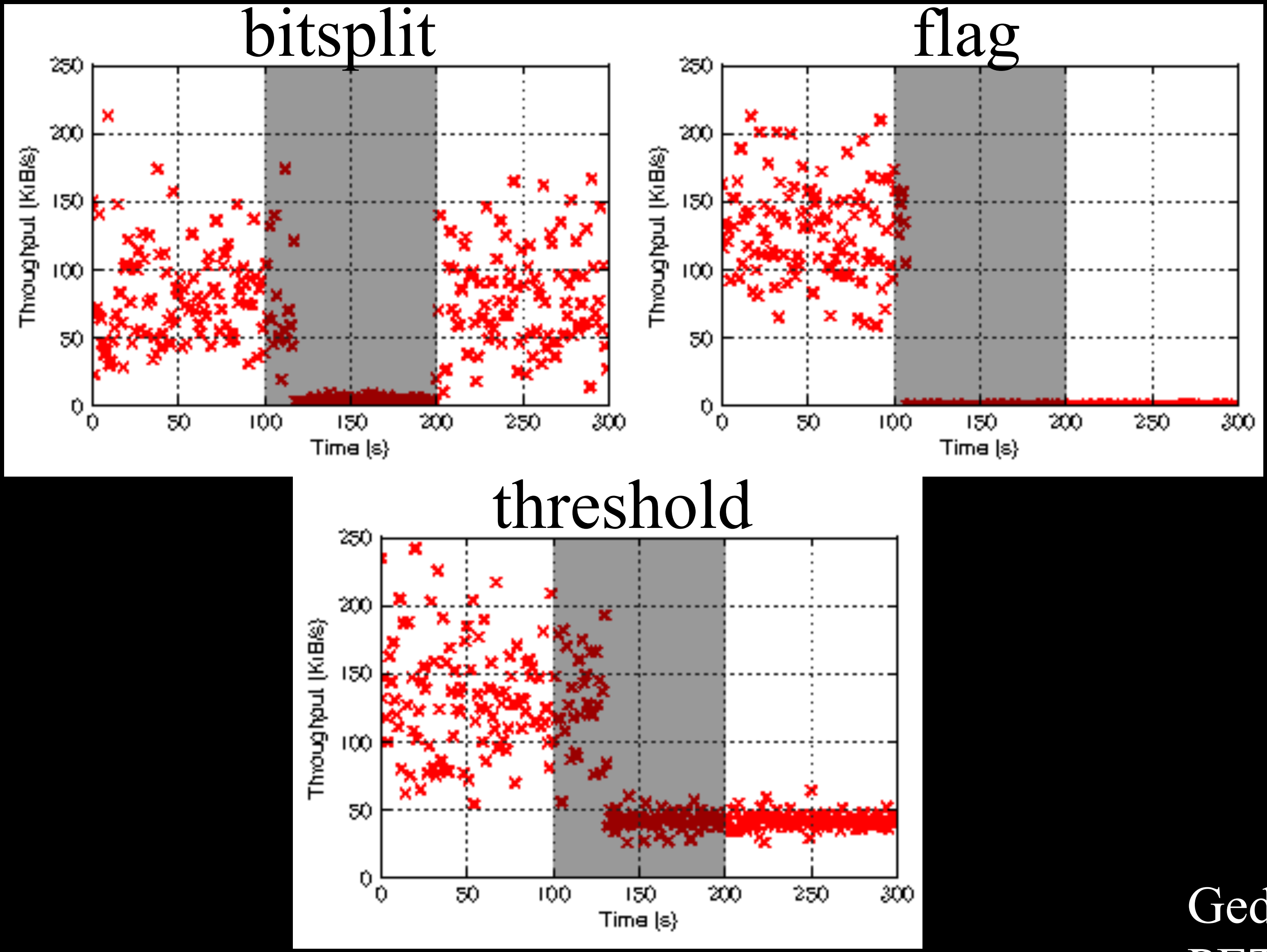
- Is this attack “stealthy”?



Throughput increases

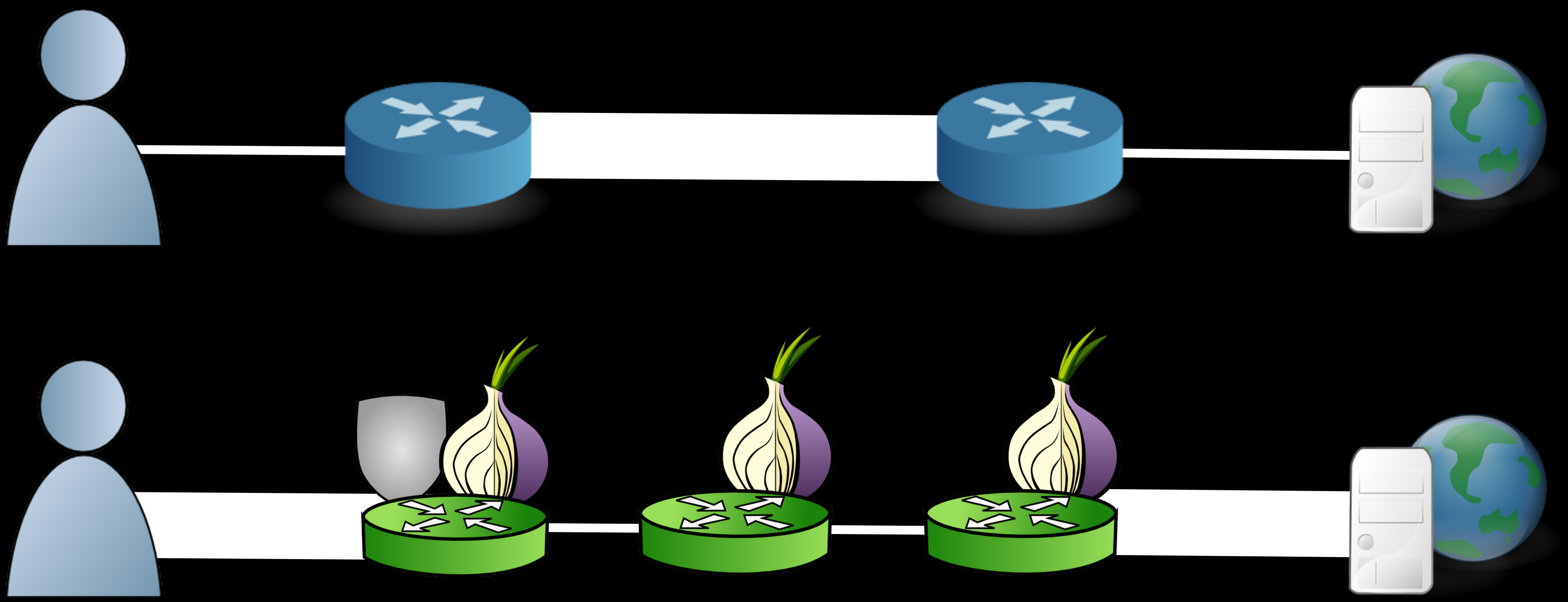
Geddes et.al. PETS’13

# Induced Throttling Prototype



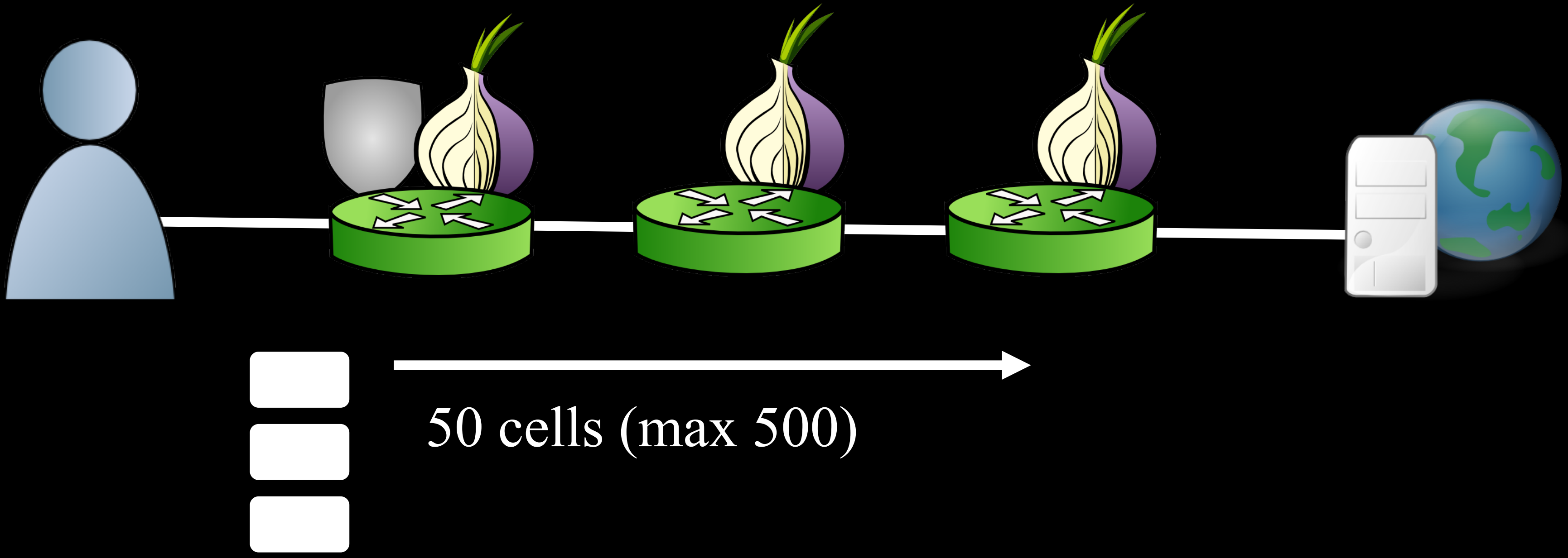

Geddes et.al.
PETS'13

# Tor != Internet

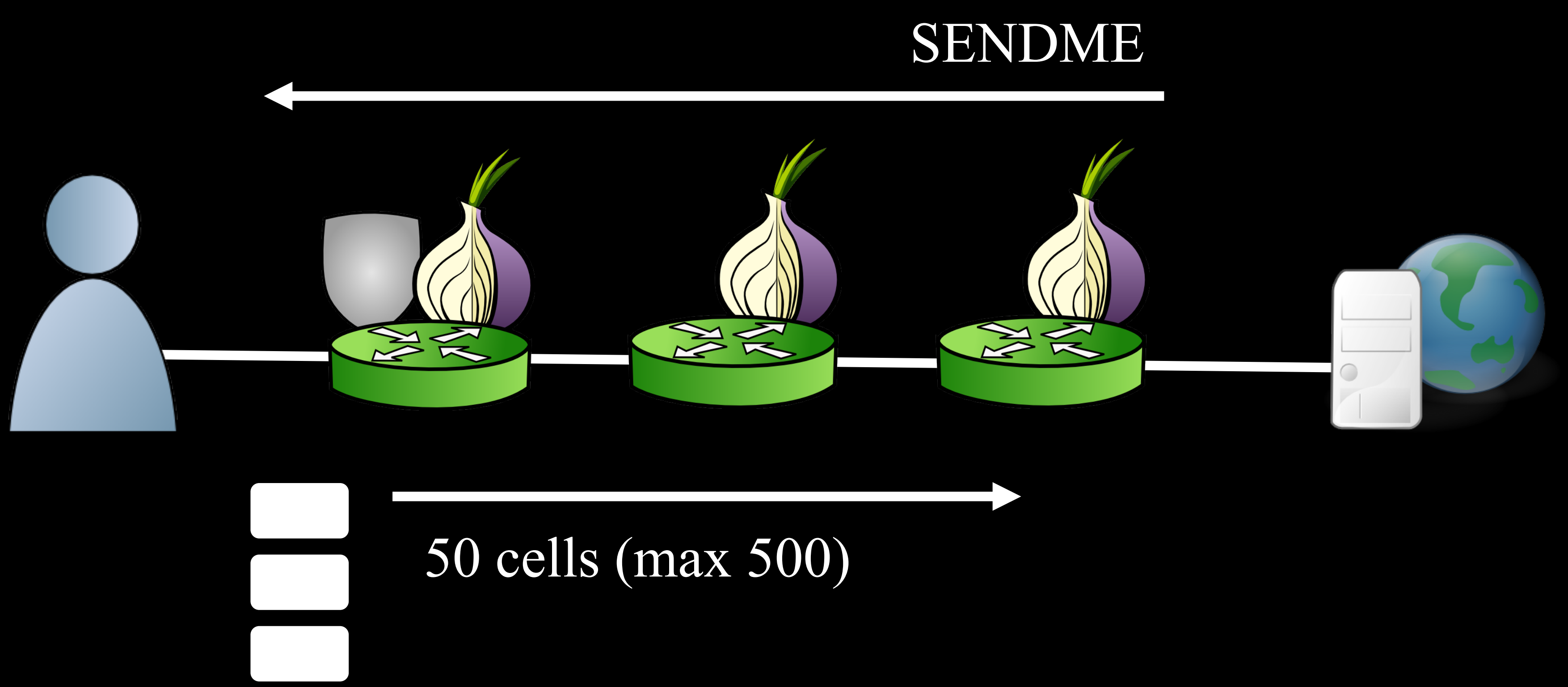


- Specialized Tor performance enhancements
  - ~~Reducing load: traffic admission control~~
  - Reducing load, improving utilization: congestion control

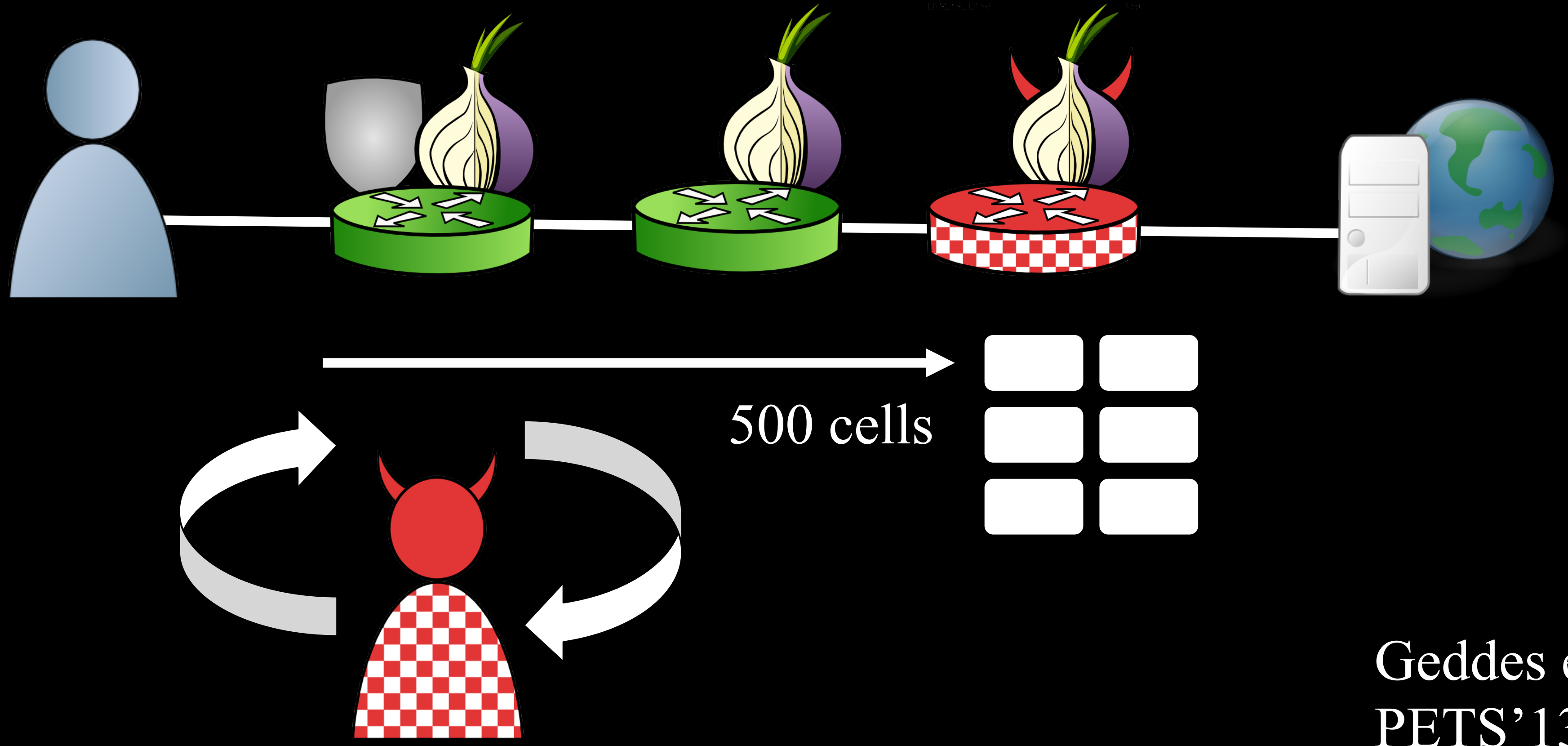
# Congestion Control



# Congestion Control

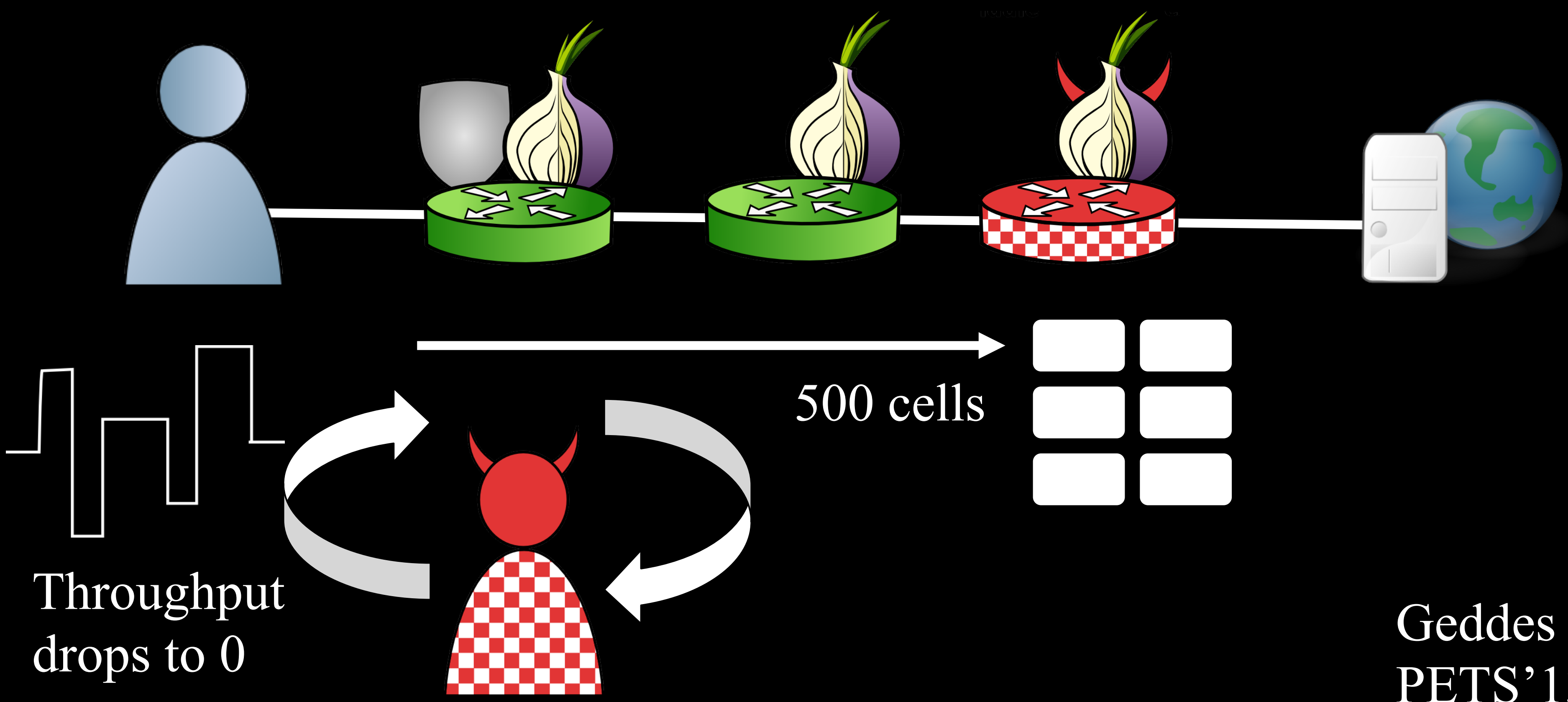


# Congestion Control



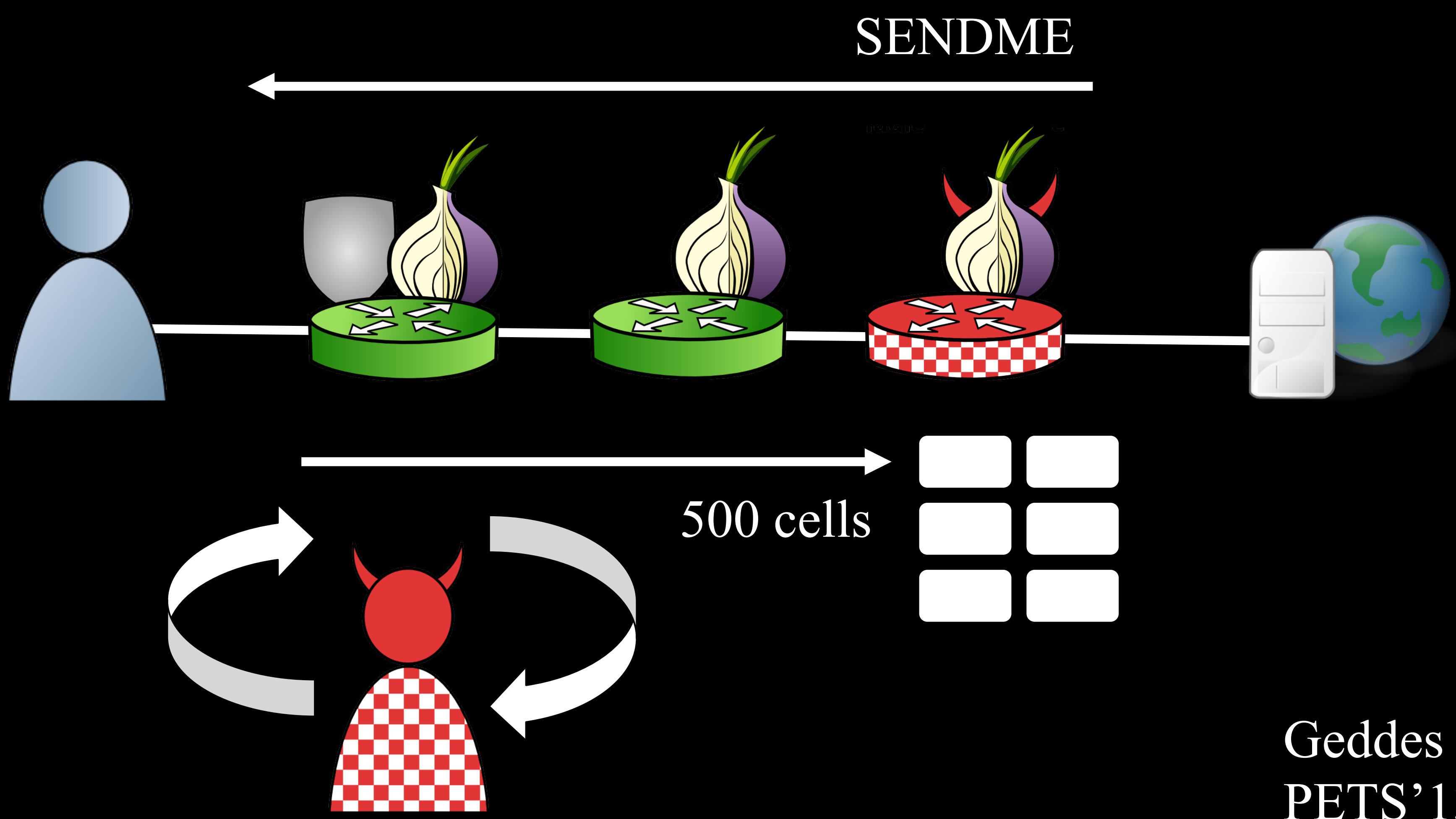

Geddes et.al.
PETS'13

# Congestion Control



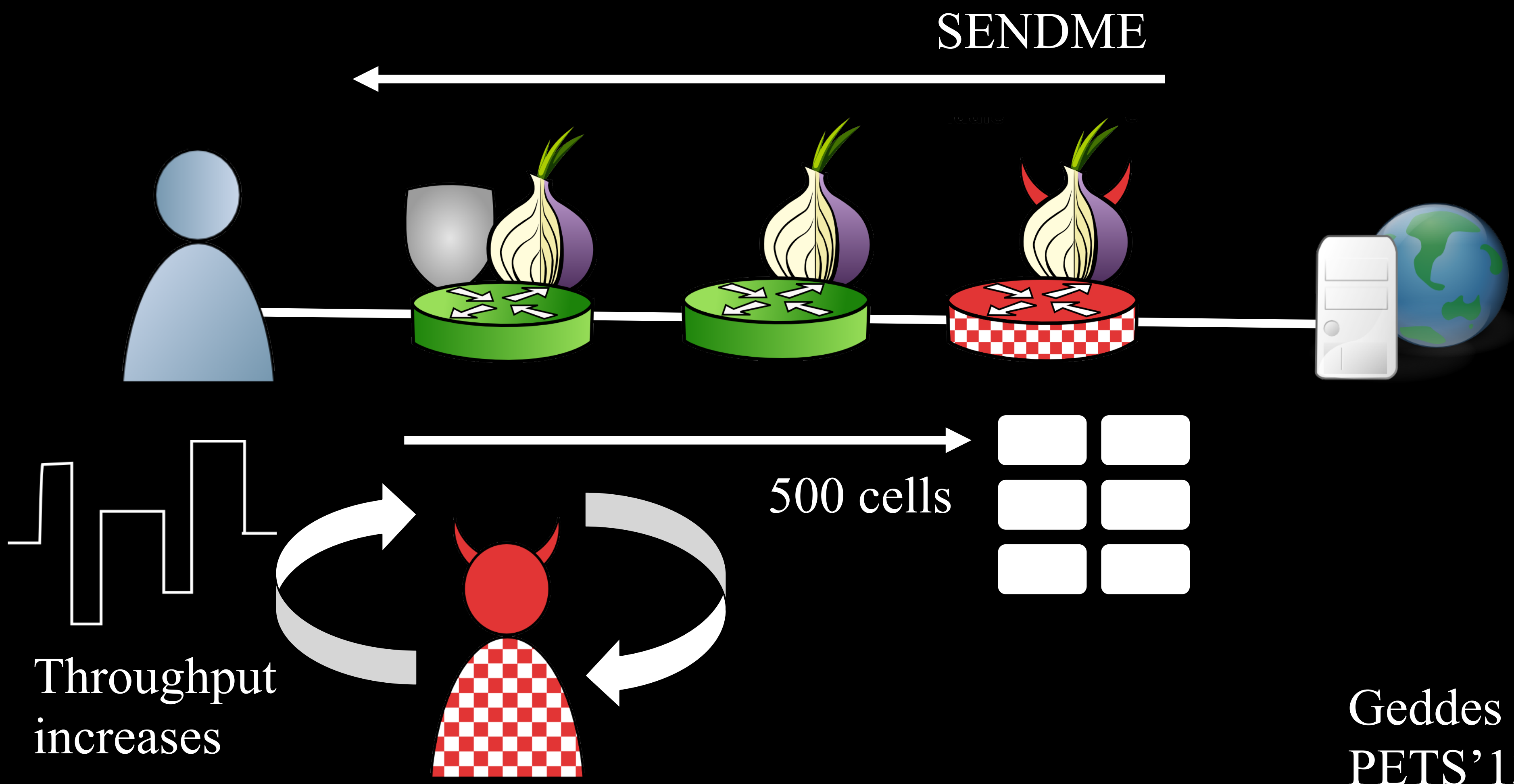

Geddes et.al.
PETS'13

# Congestion Control




Geddes et.al.
PETS'13

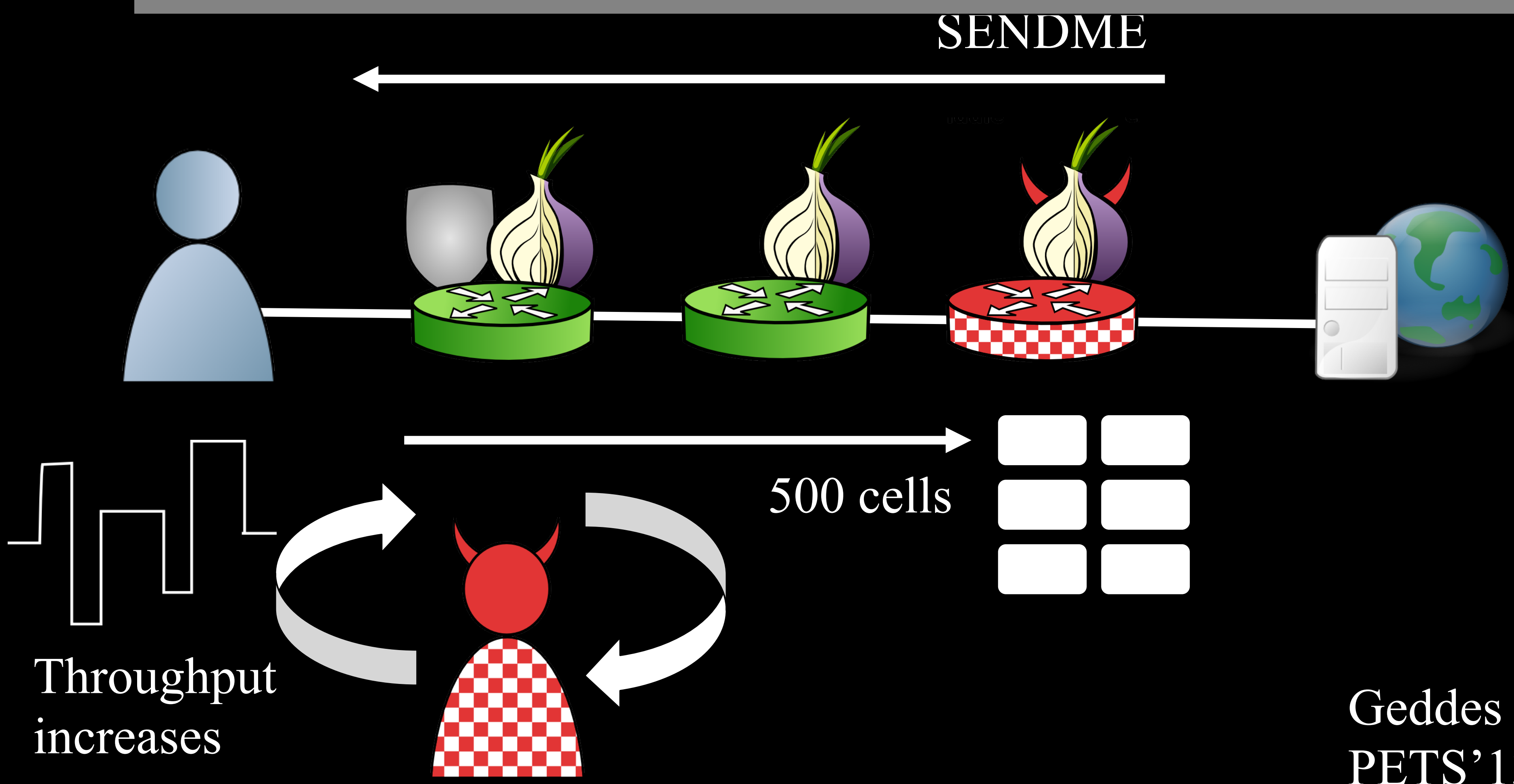
# Congestion Control




Geddes et.al. PETS’13

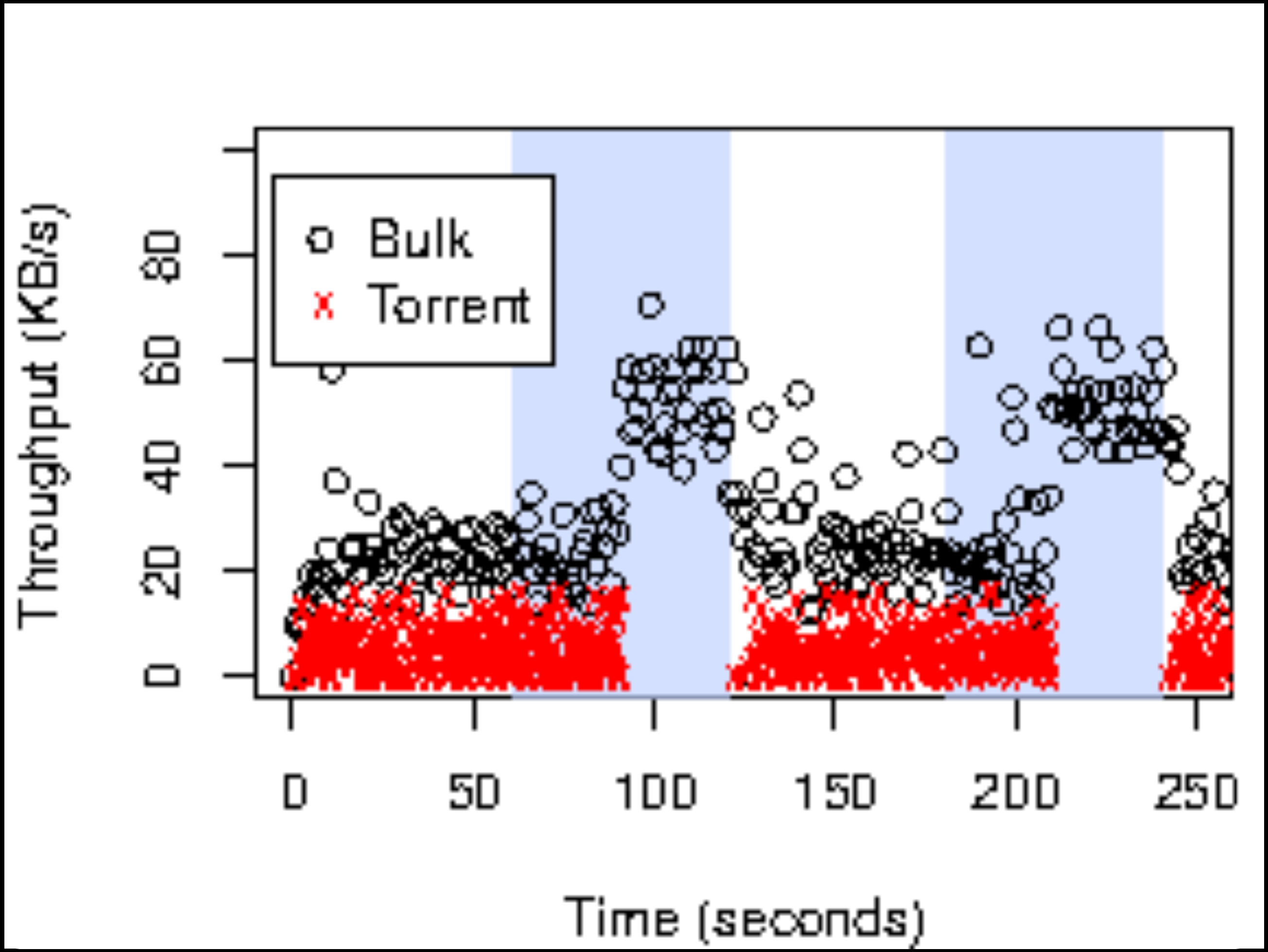
# Congestion Control

- Is this attack “stealthy”?




Geddes et.al.
PETS’13

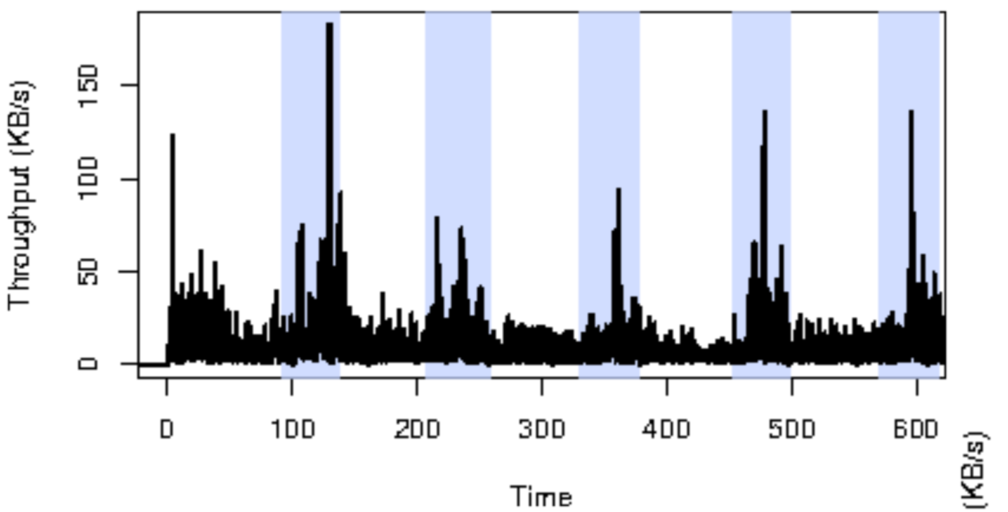
# Induced Throttling Prototype



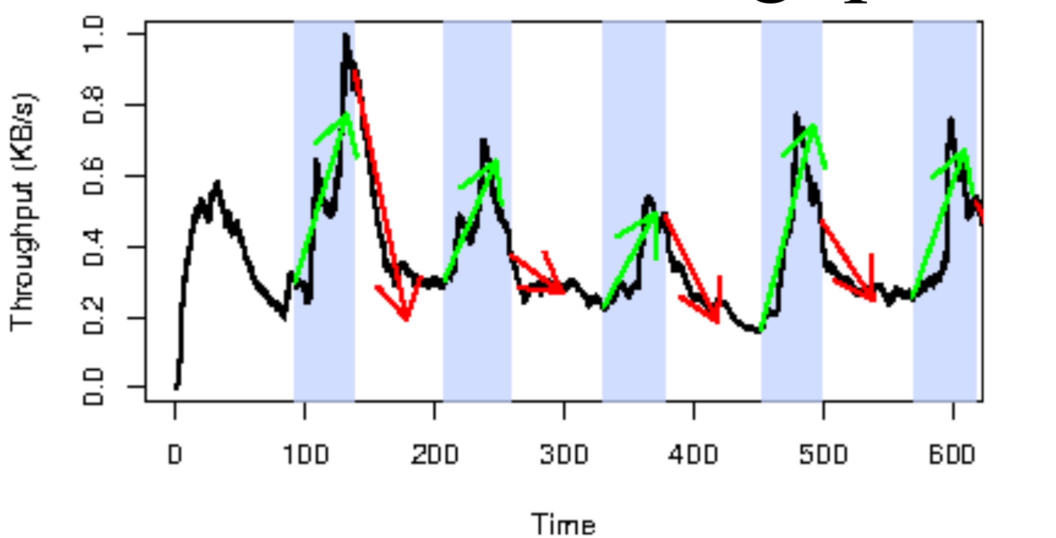

Geddes et.al.
PETS'13

# Induced Throttling Results

Raw throughput




Smoothed throughput

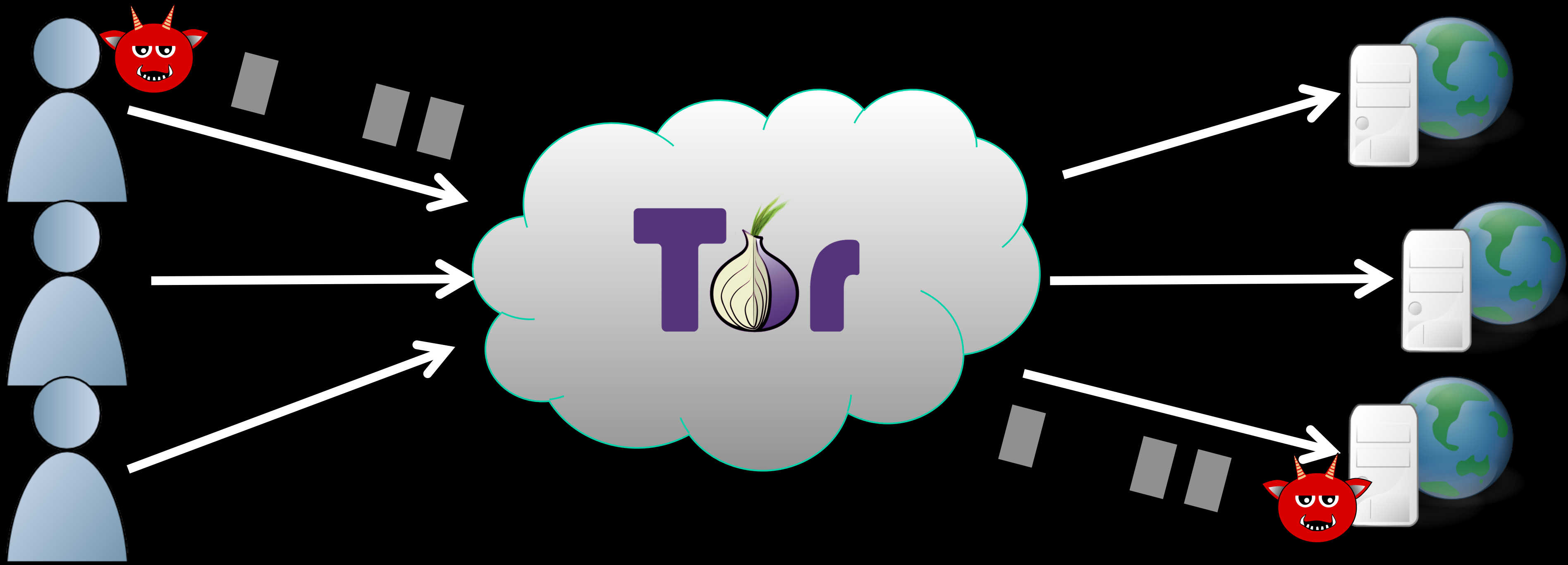



Geddes et.al.
PETS'13

# Outline

- ~~Background~~
- ~~Security against correlation (end-to-end)~~
  - ~~Metrics and methodology~~
  - ~~Node adversaries~~
  - ~~Link adversaries~~
- ~~Correlation attacks (partial)~~
  - ~~Stealthy throughput~~
  - ~~Induced throttling~~
    - ~~Traffic admission control~~
    - ~~Congestion control~~

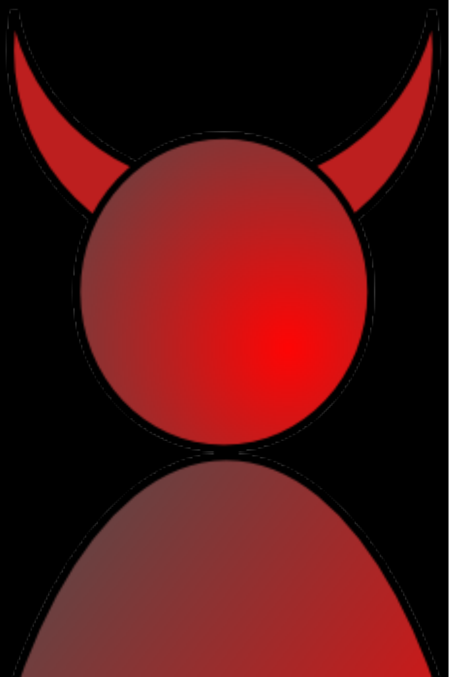
# Traffic Correlation



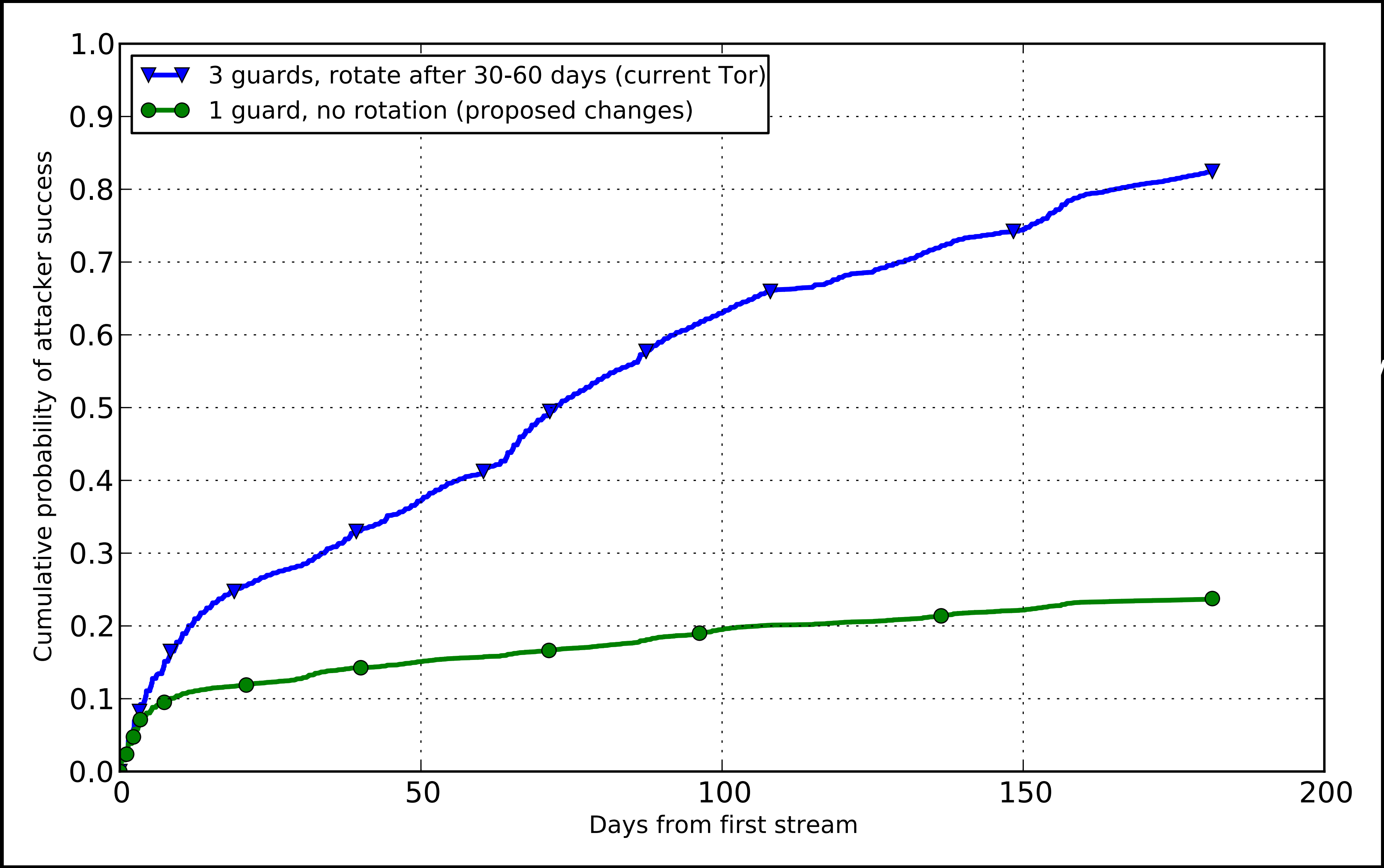
- How might we defend against ALL traffic correlation attacks?

# Questions?

rob.g.jansen@nrl.navy.mil



# Conclusion



3 guards, rotate after 30-60 days (current Tor)
1 guard, no rotation (proposed changes)
Cumulative probability of attacker success
Days from first stream
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
50
100
150
200

# Tor is Efficient: ~65% Utilization