U.S. NAVAL RESEARCH LABORATORY

Shadow: Real Applications, Simulated Networks

Dr. Rob Jansen
U.S. Naval Research Laboratory
Center for High Assurance Computer Systems

Cyber Modeling and Simulation Technical Working Group
Mark Center, Alexandria, VA
October 25th, 2017

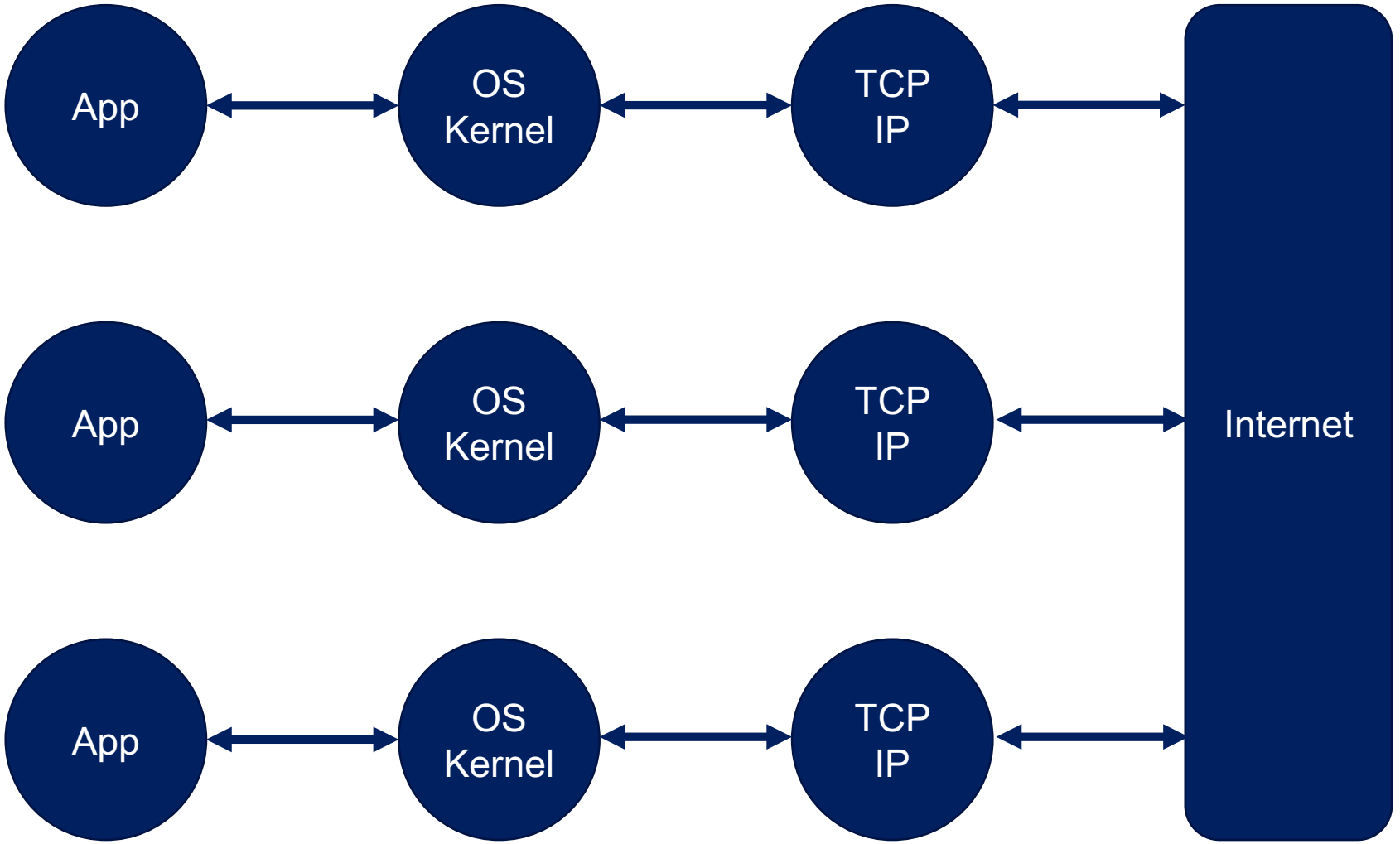
U.S. NAVAL RESEARCH LABORATORY

What is Shadow?

- Open-source network simulator / emulator hybrid
- Directly executes real applications like Tor and Bitcoin over a simulated network topology
- Efficient, scalable, deterministic, accurate, and more!

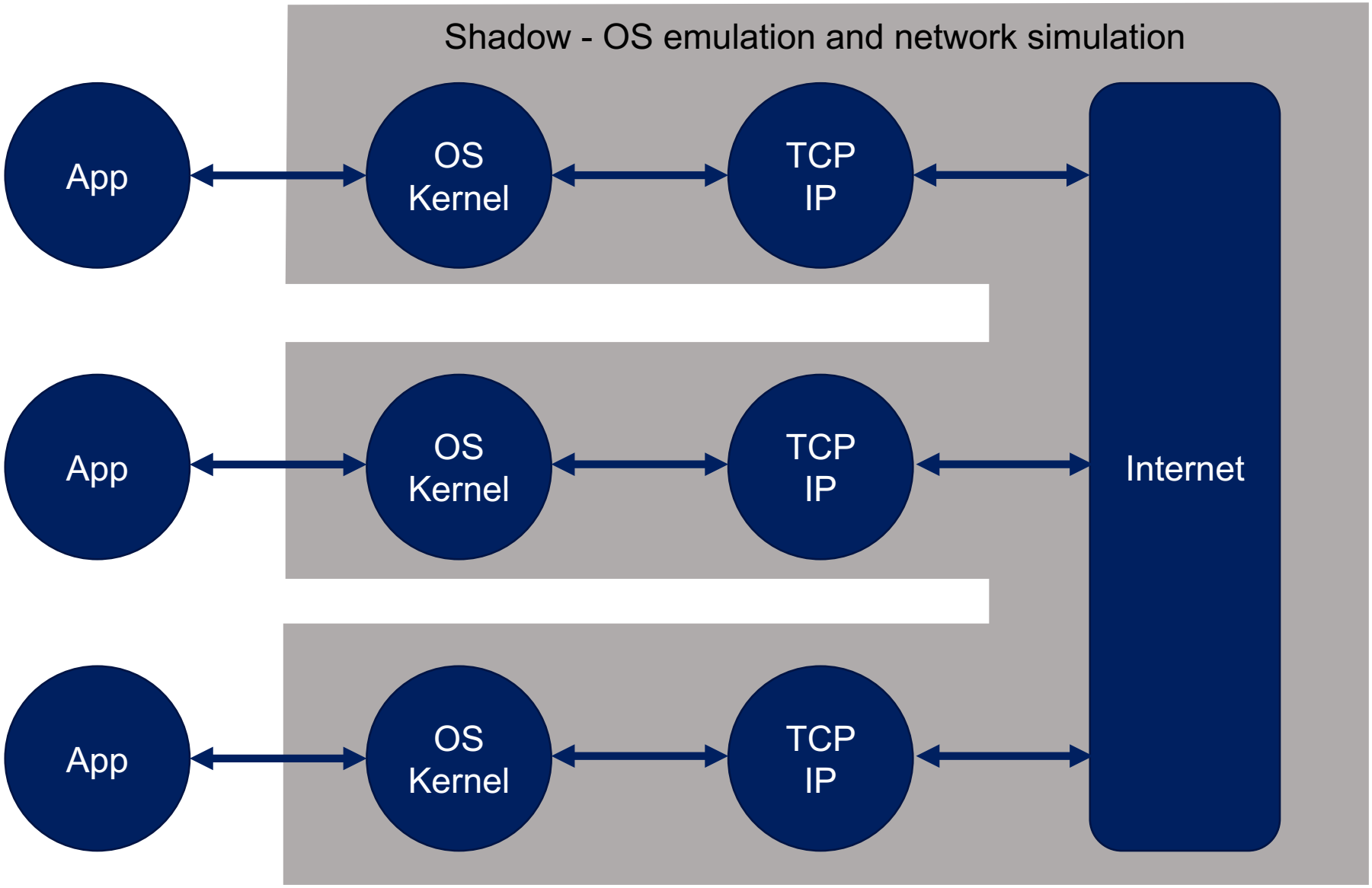
U.S. NAVAL RESEARCH LABORATORY

How does Shadow Work?



U.S. NAVAL RESEARCH LABORATORY

How does Shadow Work?



U.S. NAVAL RESEARCH LABORATORY

Why should you care?

- Expedite research and development
- Evaluate software mods or attacks without harming real users
- Understand holistic effects before deployment
- Shadow supports simulation for general-purpose applications

U.S. NAVAL RESEARCH LABORATORY

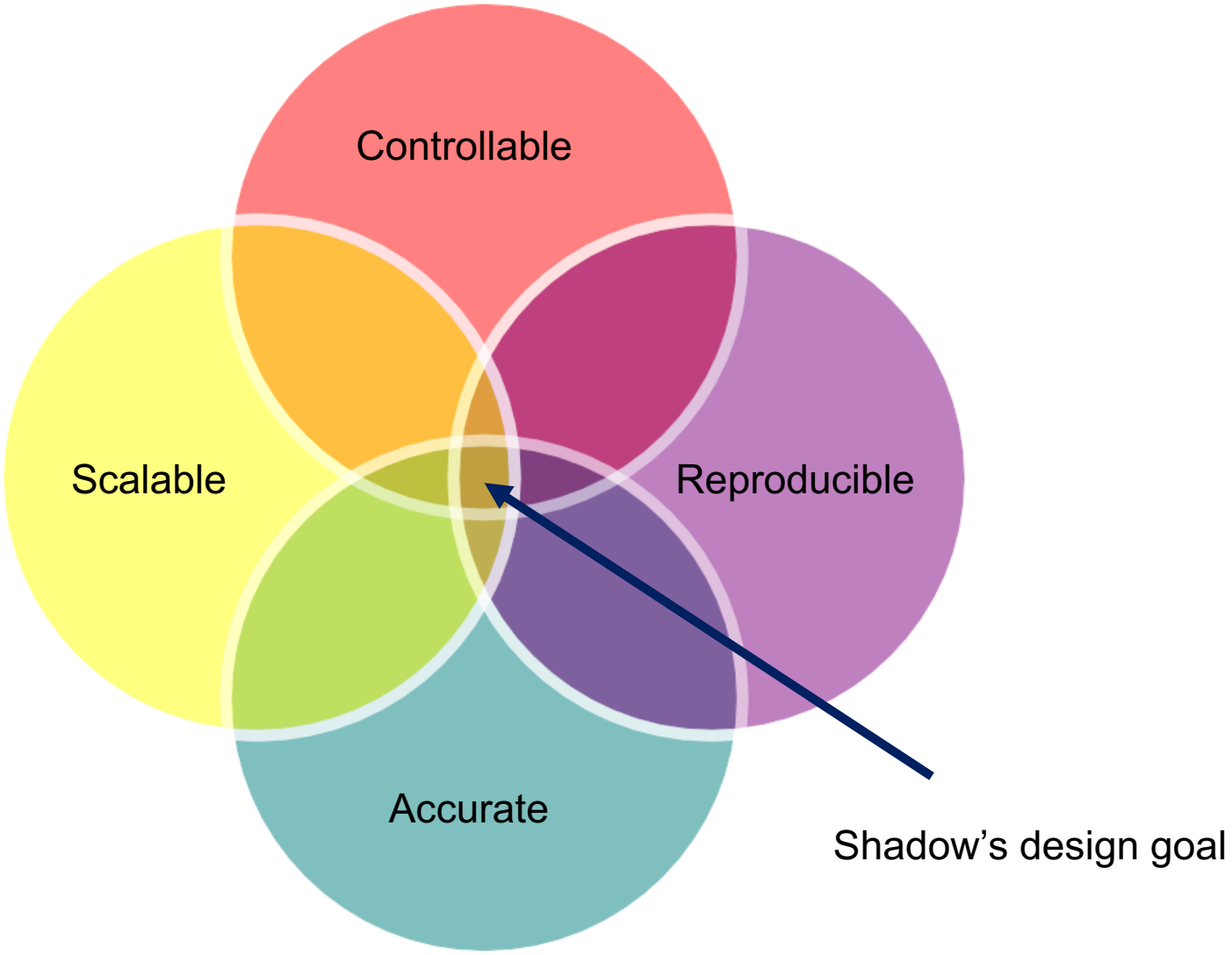
Talk Outline

- Experimentation options and tradeoffs
- Shadow design
- Simulation use cases – The Tor Anonymity Network
  - Gentle Tor introduction
  - Congestion problems
  - Scheduling algorithms
  - Performance enhancing algorithms
  - Denial of service attacks
- Conclusion

Experimentation Options

U.S. NAVAL RESEARCH LABORATORY

Desirable Experiment Properties

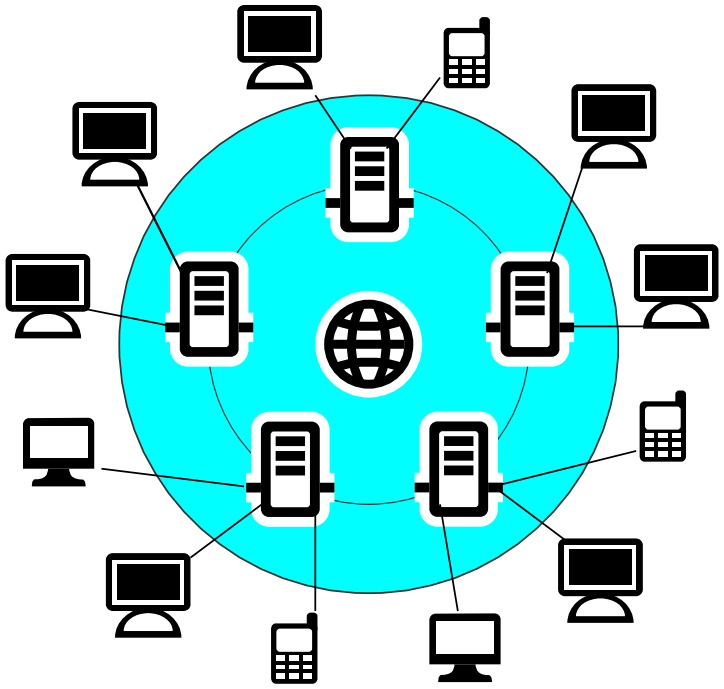


U.S. NAVAL RESEARCH LABORATORY

Live Network Experiments

Experimenting in a deployed distributed system

- Pros
  - Most realistic
  - The target environment



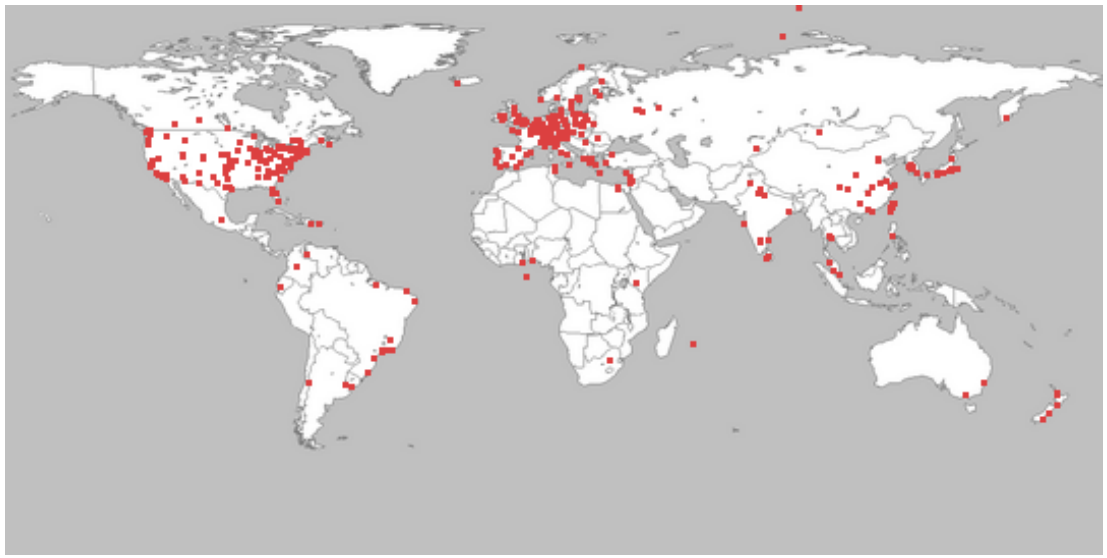
- Cons
  - Hard to manage/debug
  - Lengthy deployment
  - Security risks

U.S. NAVAL RESEARCH LABORATORY

Testbed Network Experiments

Experimenting in a distributed testbed (e.g. PlanetLab)

- Pros
  - Close to target environment
  - Runs on the Internet or uses Internet protocols
- Cons
  - Hard to manage and debug
  - Doesn't scale well in low-resource environs
  - Can be hard to model network properties

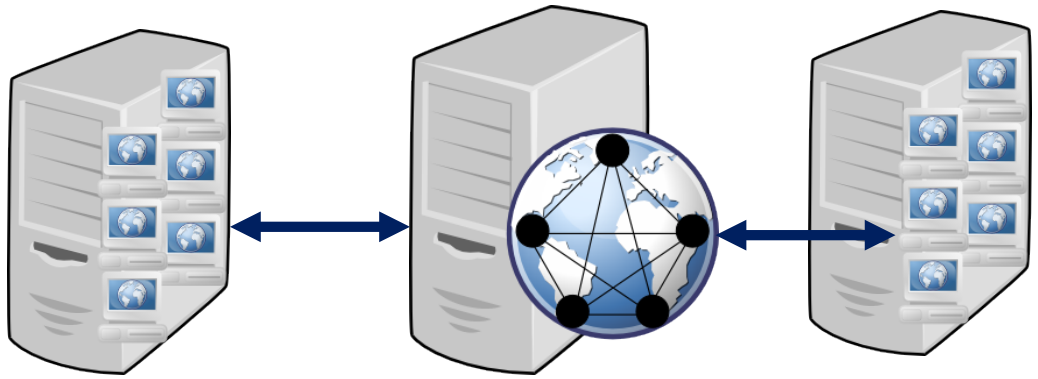


U.S. NAVAL RESEARCH LABORATORY

Network Emulation

Experimenting with network emulators (e.g. Modelnet)

- Pros
  - Runs on the target OS and uses Internet protocols
  - Can model various network properties
- Cons
  - Requires multiple machines and custom installed OS
  - Must run in real time
  - Per-process overhead may cause kernel issues



U.S. NAVAL RESEARCH LABORATORY

Network Simulation

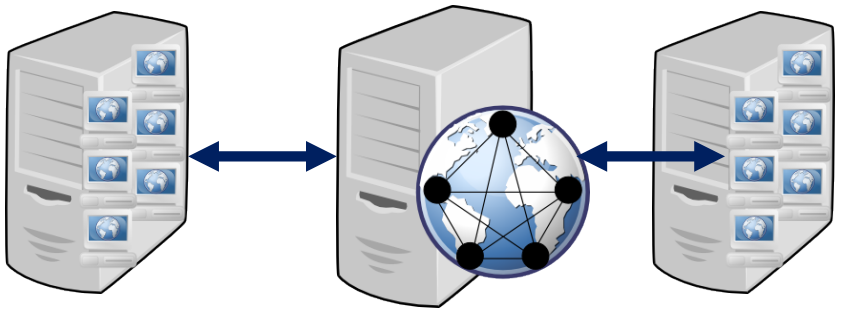
Experimenting with network simulators (e.g. NS3)

- Pros
  - Deterministic (reproducible)
  - Can model various network properties
  - Can scale very well
- Cons
  - Application model can be too abstract
  - Abstractions can lead to inaccurate results

U.S. NAVAL RESEARCH LABORATORY

Simulation vs. Emulation: Realism

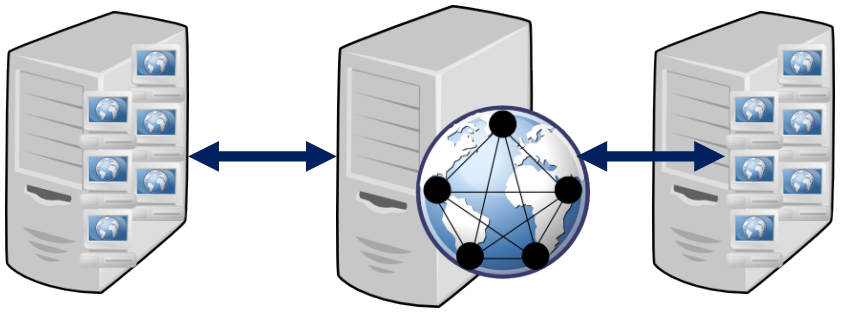
Simulation	Emulation
Abstracts away most system components	Runs the real OS, kernel, protocols, applications
Simulator is generally only internally consistent	Software is interoperable with external components
Less resource intensive	More resource intensive



U.S. NAVAL RESEARCH LABORATORY

Simulation vs. Emulation: Time

Simulation	Emulation
“As-fast-as-possible”	Real time
Control over clock, can pause time without issue	Time must advance in synchrony with wall-clock
Weak hardware extends total experiment runtime	Weak hardware causes glitches that are difficult to detect and diagnose

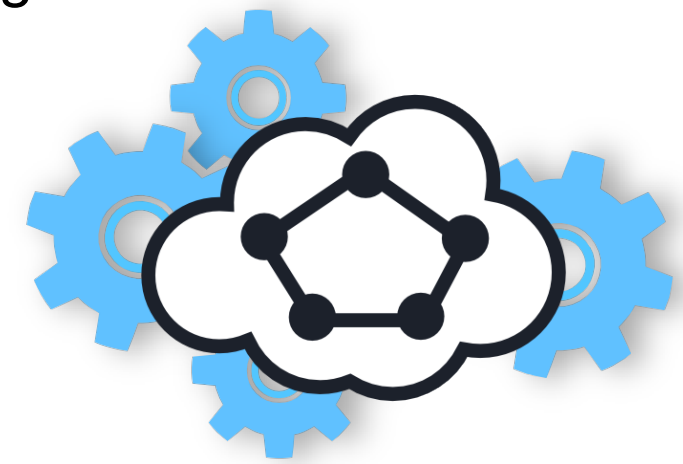


Shadow Design

U.S. NAVAL RESEARCH LABORATORY

Shadow Overview

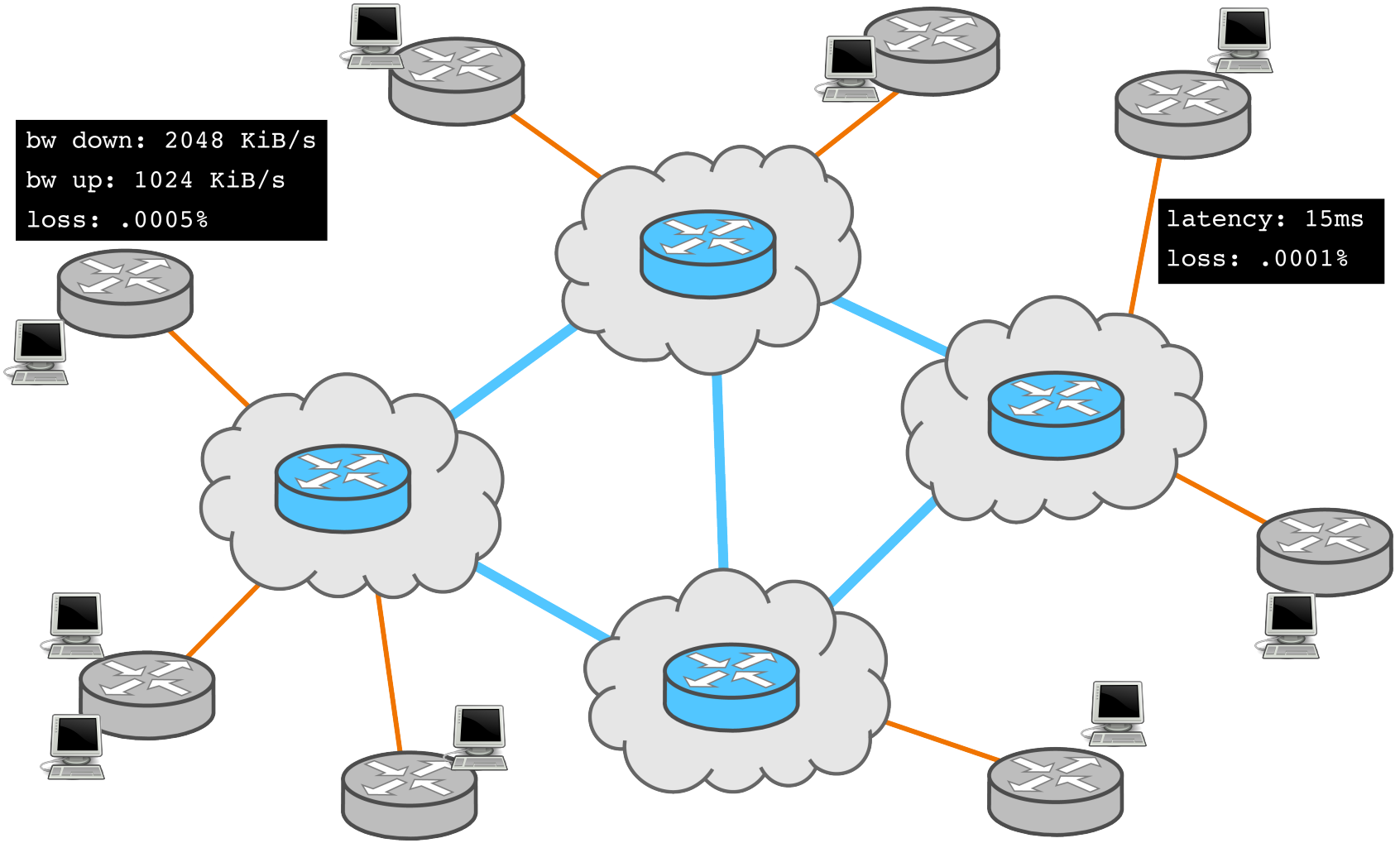
- Parallel discrete-event network simulator
- Models routing, latency, bandwidth
- Simulates time, CPU, OS
  - TCP/UDP, sockets, queuing, threading
- Emulates POSIX C API on Linux
- Directly executes apps as plug-ins



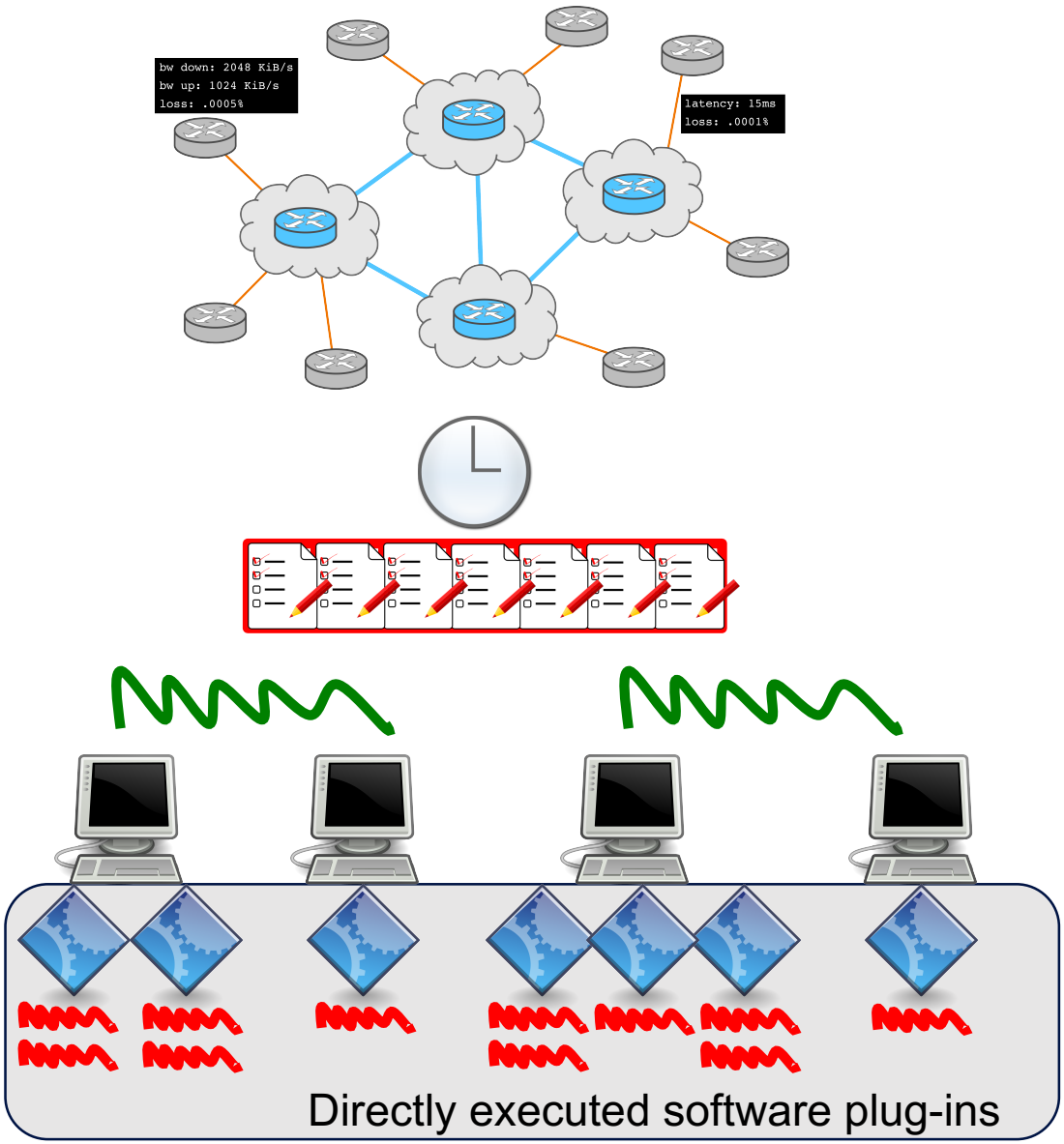
U.S. NAVAL RESEARCH LABORATORY

Initializing the Simulation

Load network model, create virtual hosts



Shadow Simulation Layout

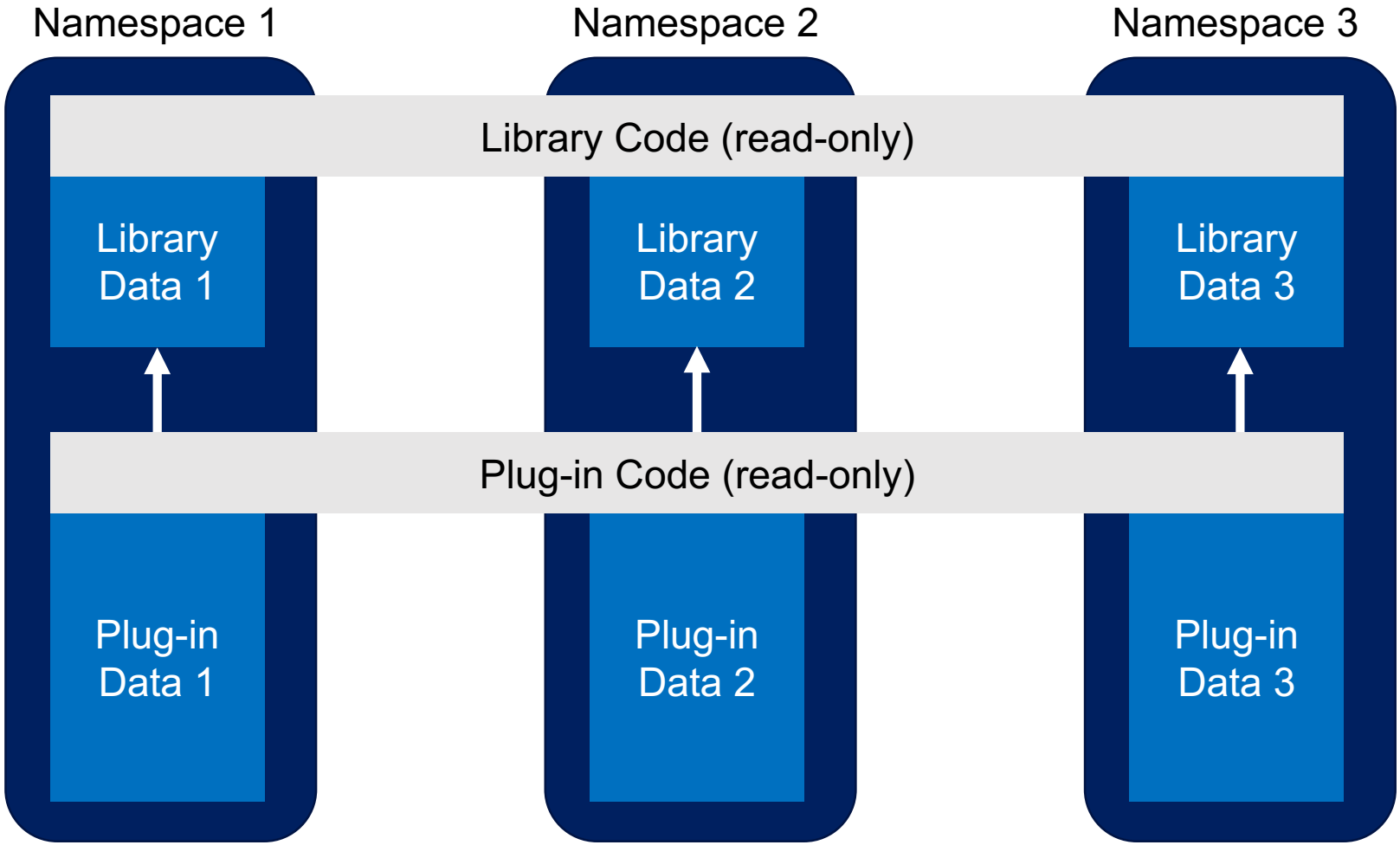


- Network model
- Global simulation clock
- Discrete event queue
- Shadow worker threads
- Virtual hosts
- Virtual processes (i.e. namespaces)
- Virtual threads

U.S. NAVAL RESEARCH LABORATORY

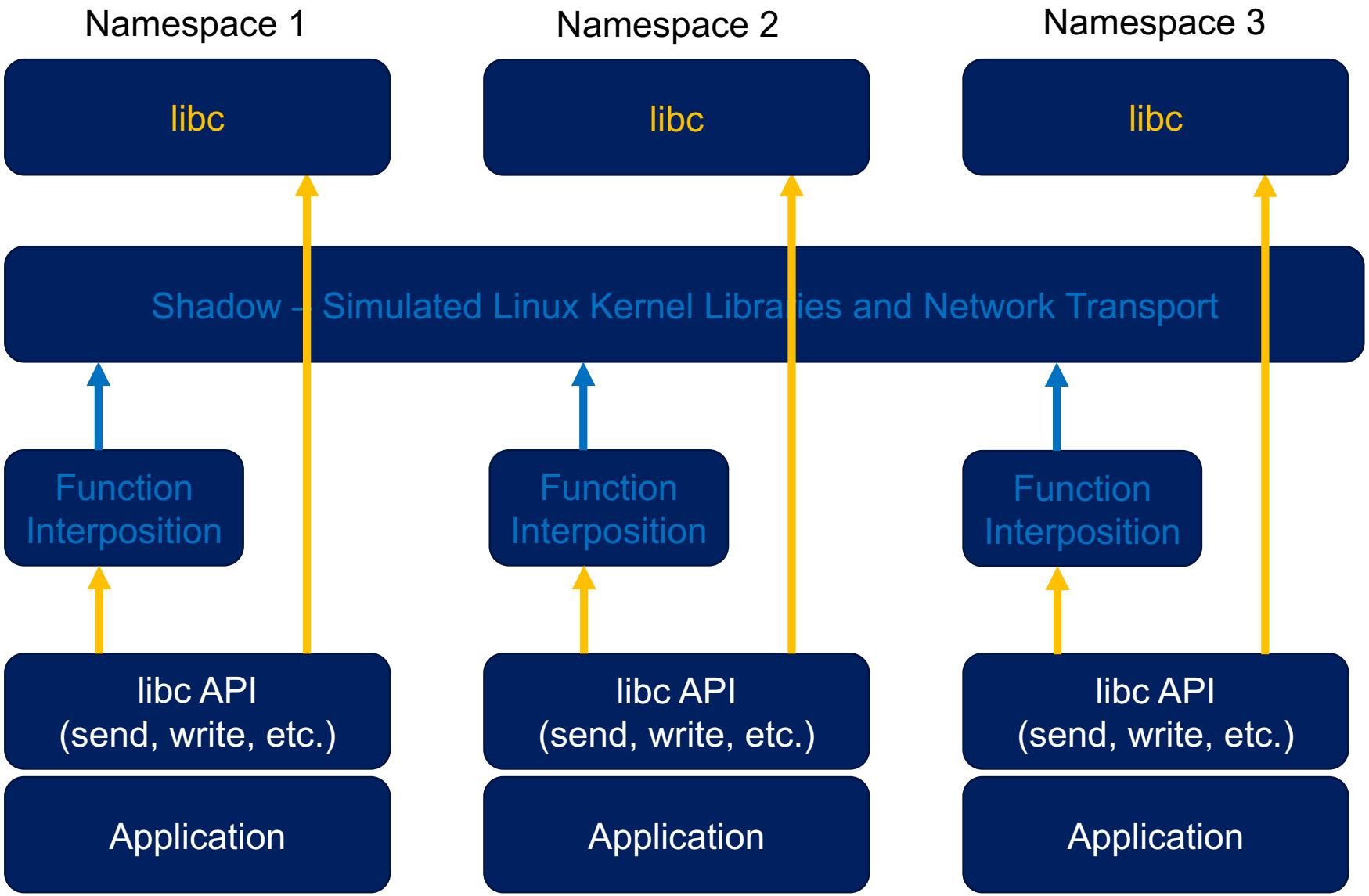
App Memory Management

Apps loaded in independent namespaces, copy-on-write



U.S. NAVAL RESEARCH LABORATORY

Direct Execution in a Simulator



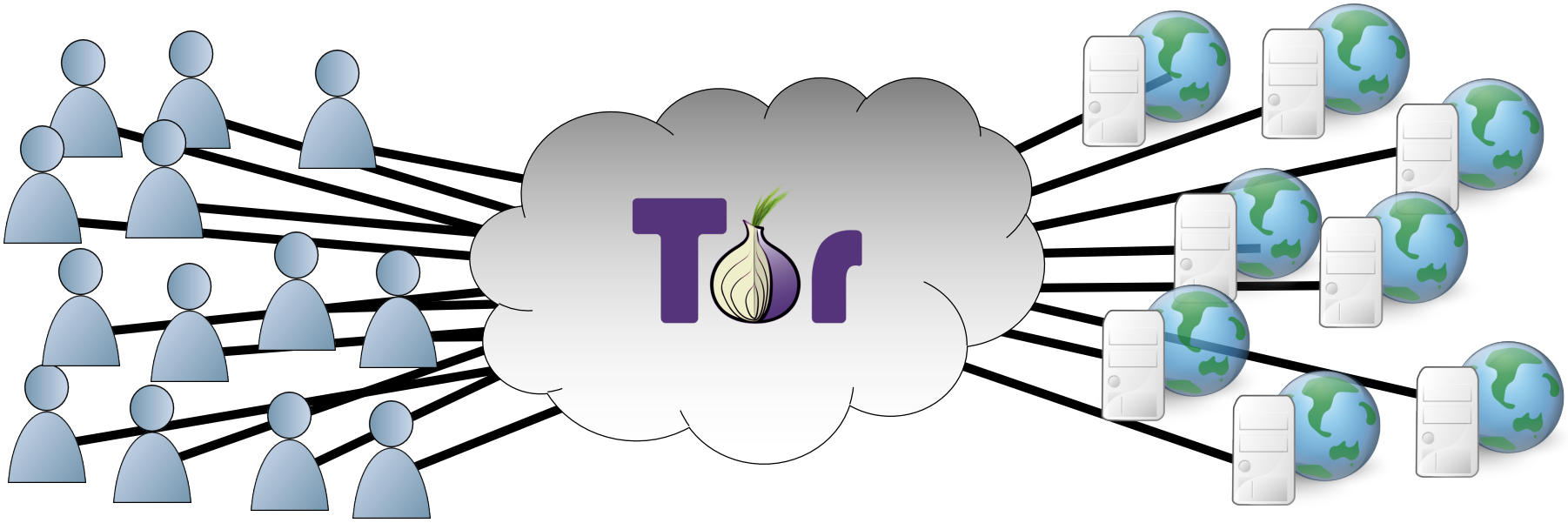
Simulation Use Cases – The Tor Anonymity Network

Gentle Tor Introduction

U.S. NAVAL RESEARCH LABORATORY

Tor Overview

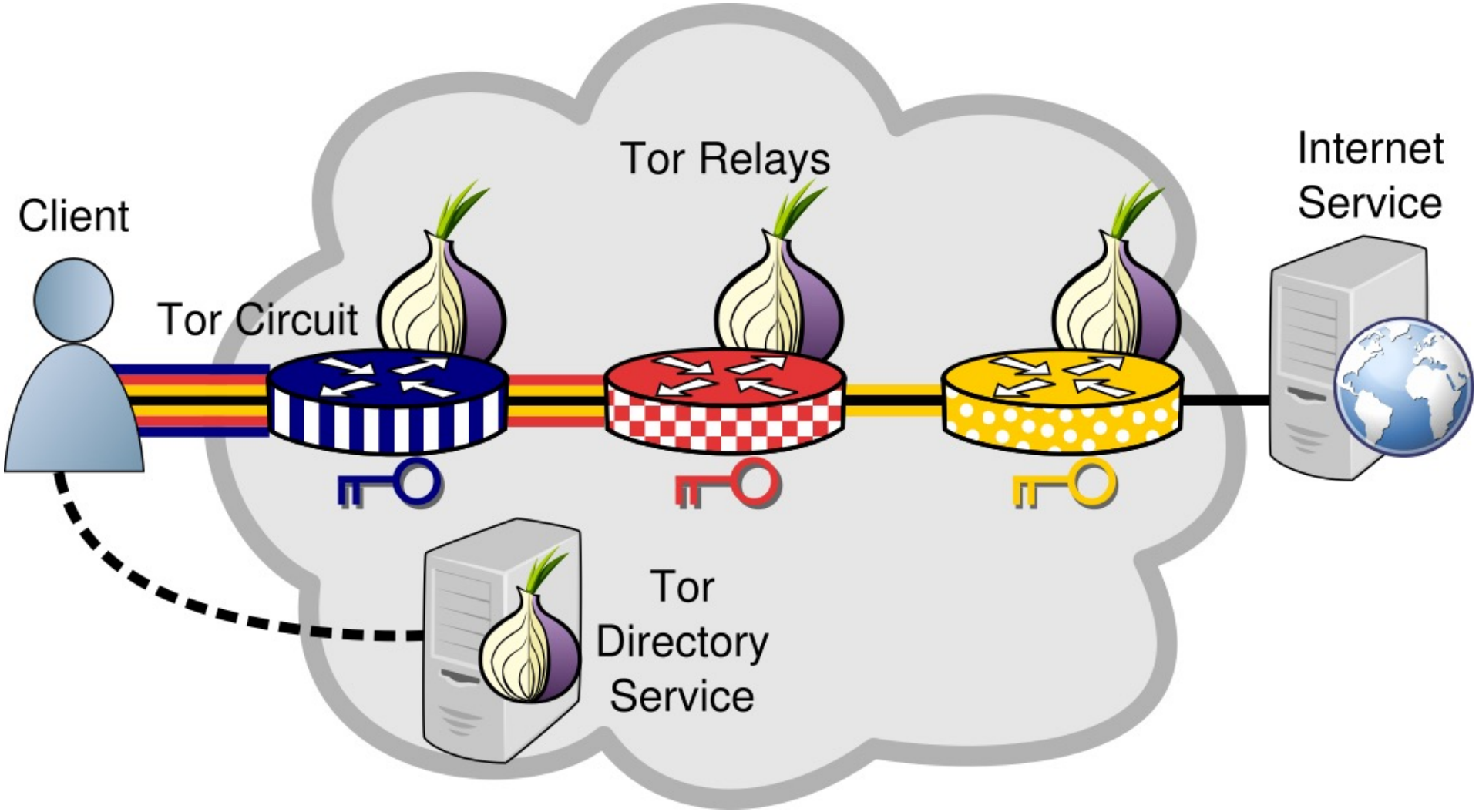
Tor: a censorship resistant, privacy-enhancing anonymous communication system



Estimated ~1.75 M. Users/Day
(metrics.torproject.org)

U.S. NAVAL RESEARCH LABORATORY

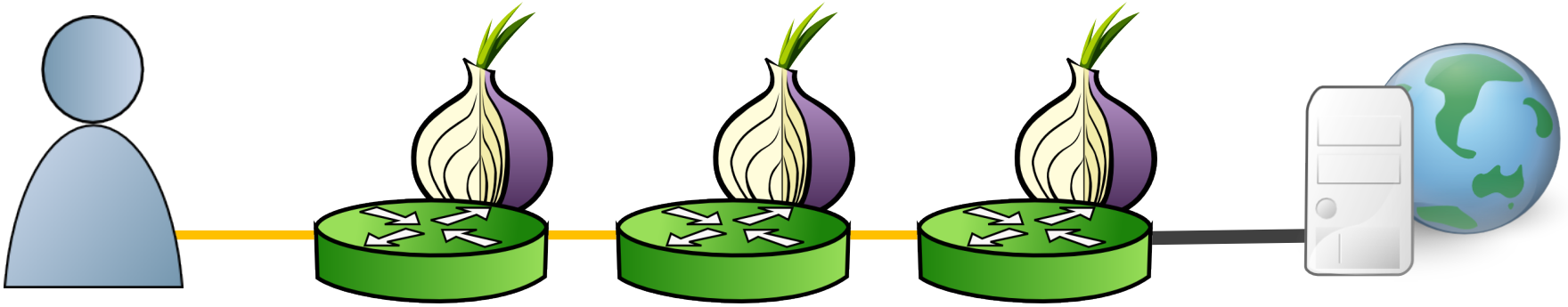
Anonymous Communication



Congestion Analysis

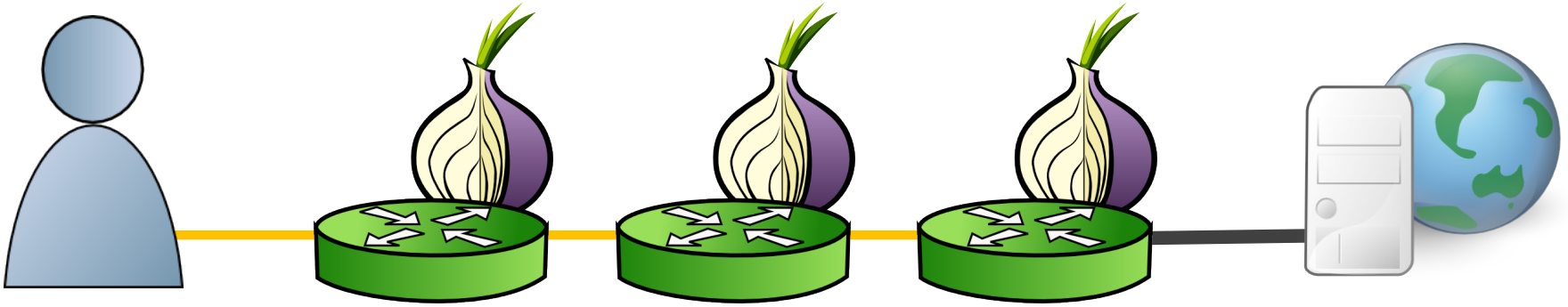
U.S. NAVAL RESEARCH LABORATORY

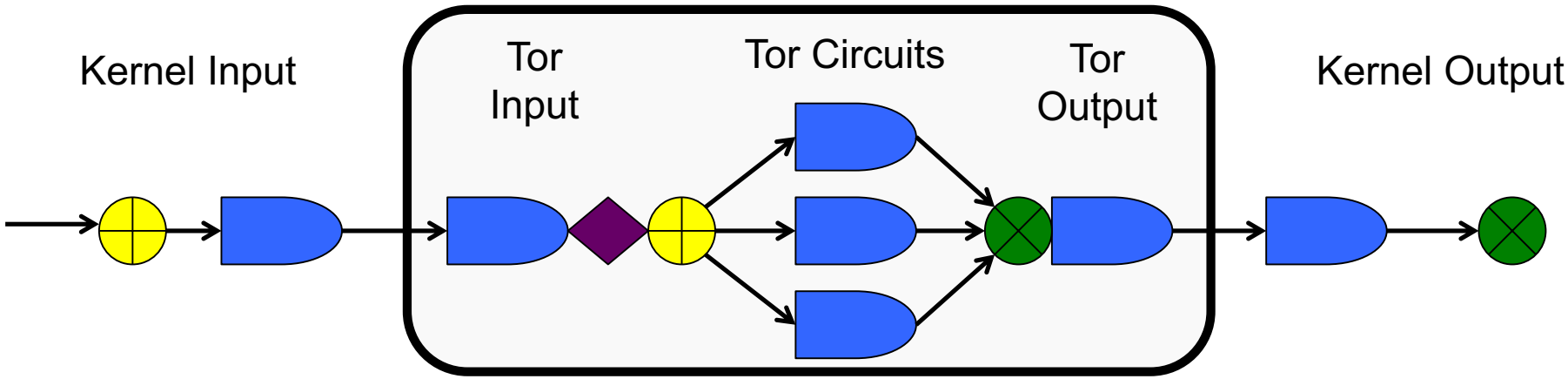
Multiple Hops – Tor is Slower



U.S. NAVAL RESEARCH LABORATORY

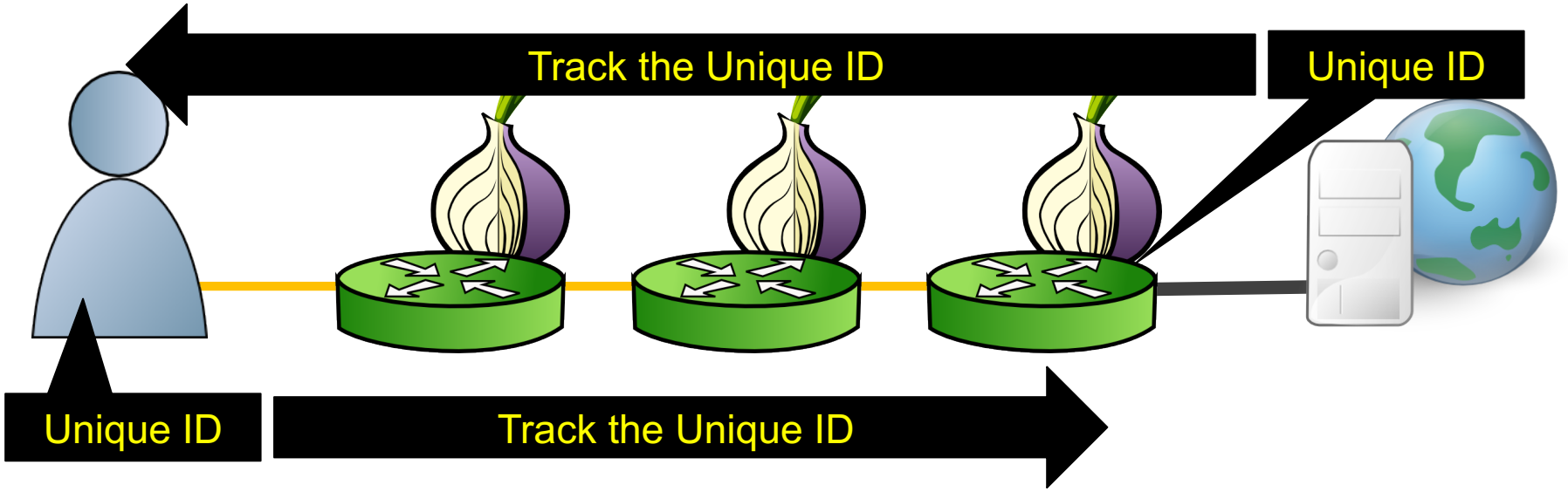
Buffers in a Tor Relay

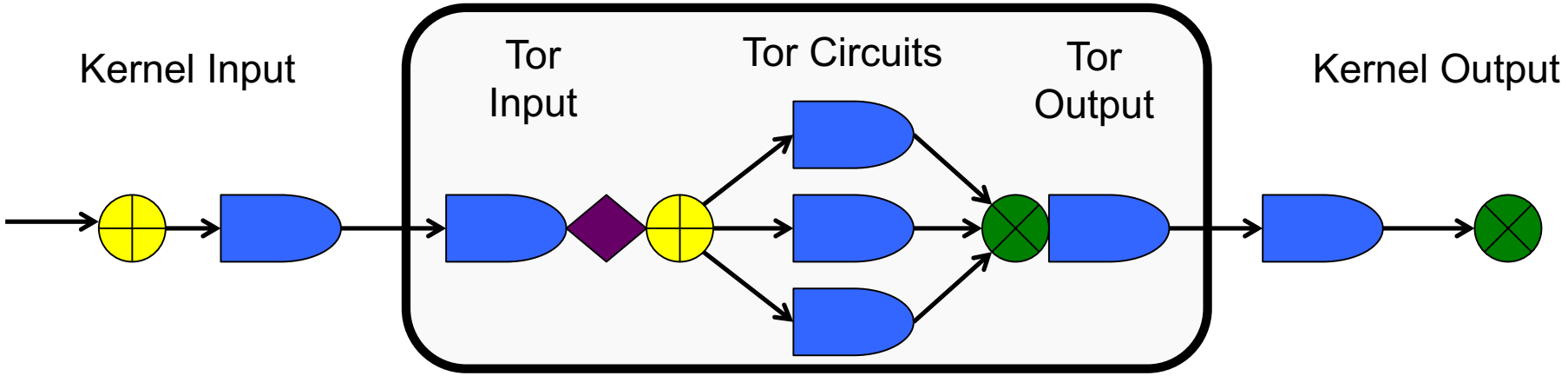




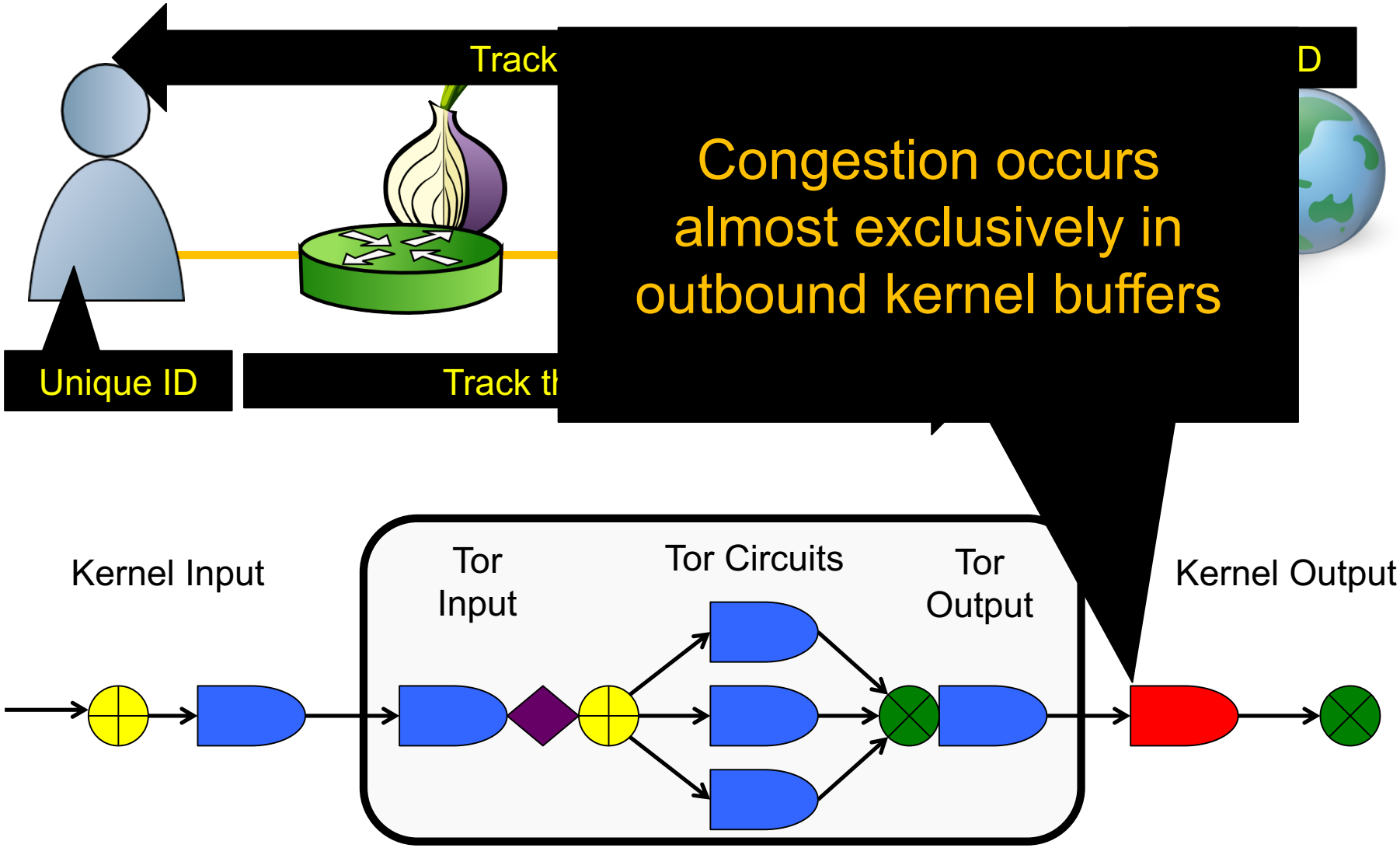
U.S. NAVAL RESEARCH LABORATORY

Tracking Congestion in Tor





Tracking Congestion in Tor



Scheduling Analysis

U.S. NAVAL RESEARCH LABORATORY

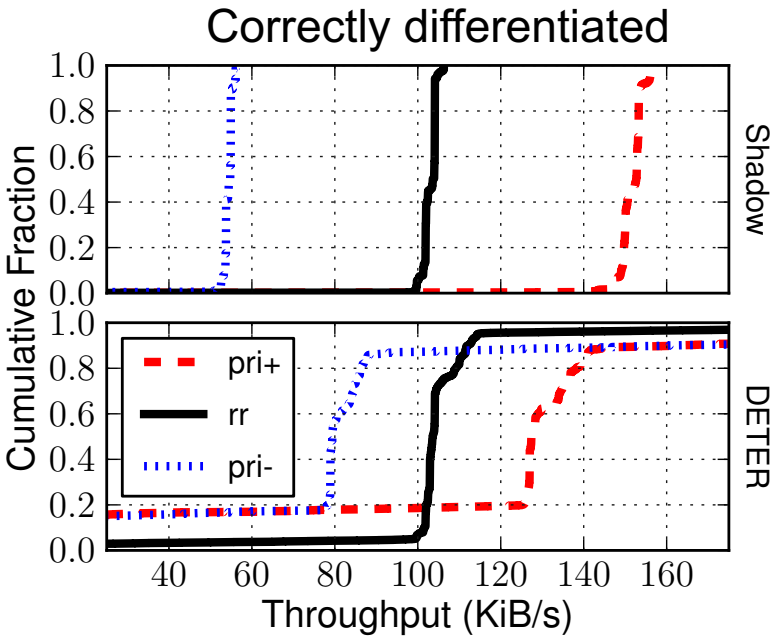
Tor Circuit Priority

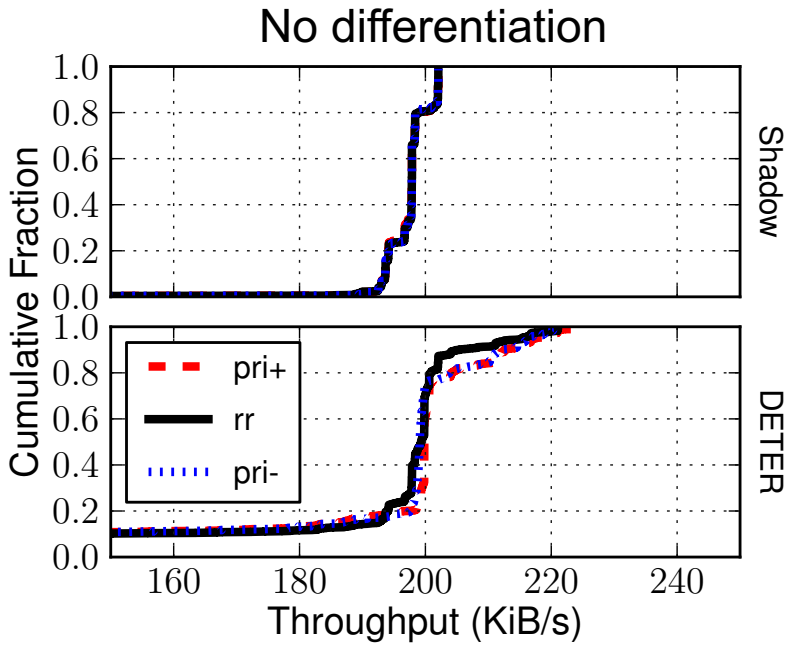
Scenarios to understand outbound queue congestion

U.S. NAVAL RESEARCH LABORATORY

Tor Circuit Priority

Scenarios to understand outbound queue congestion

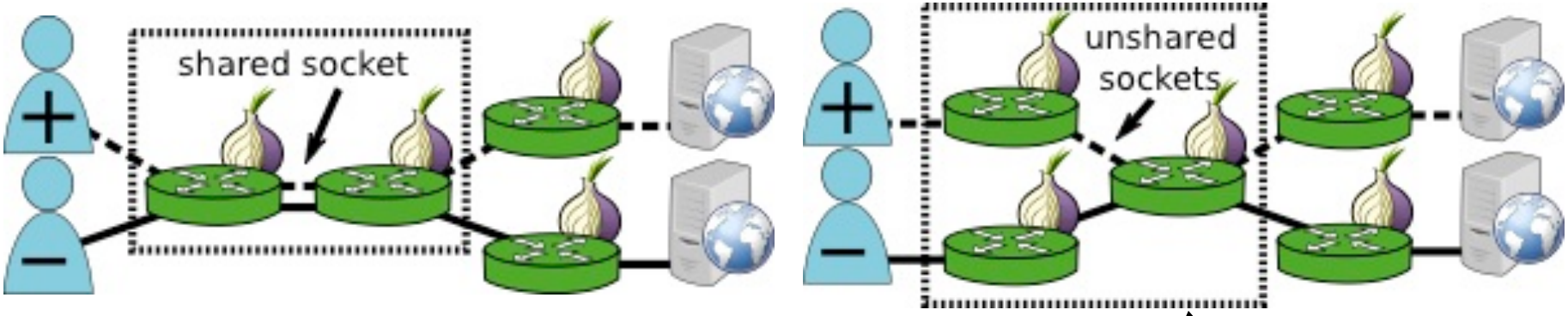


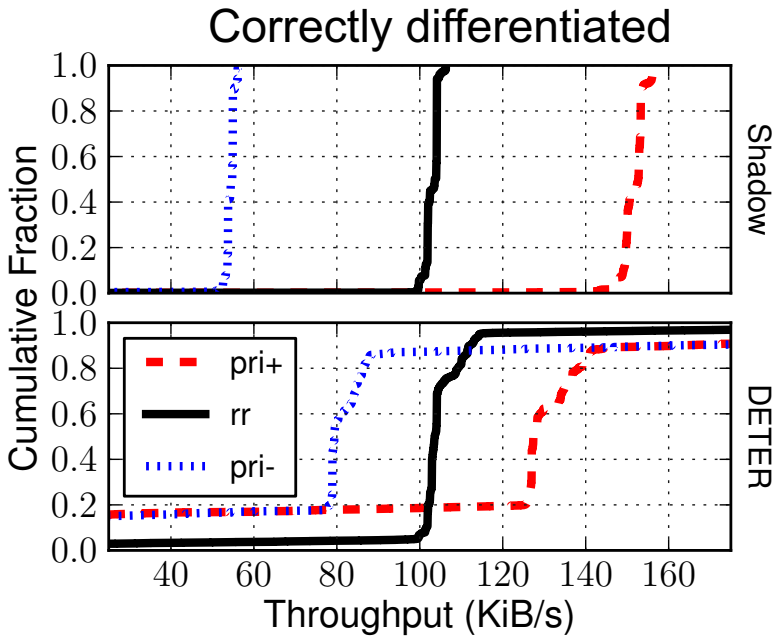


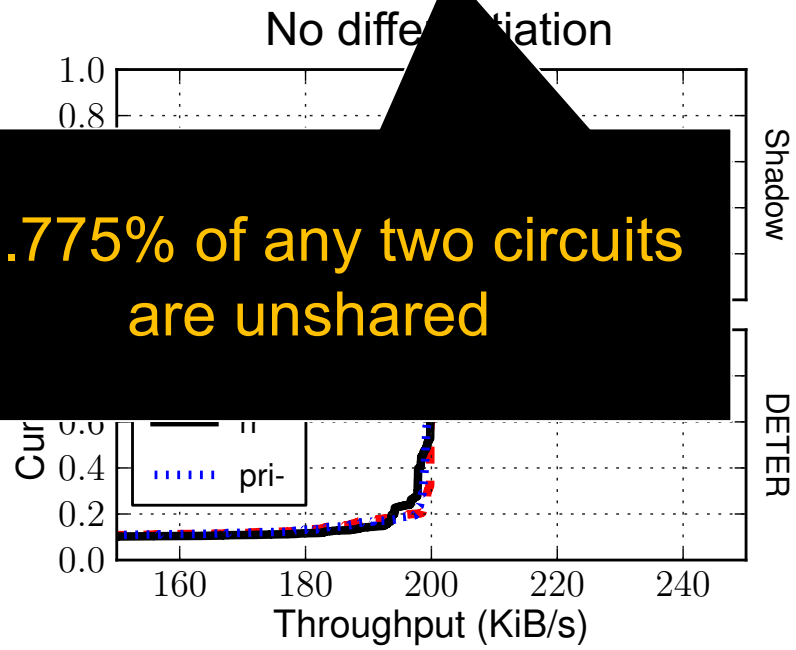
U.S. NAVAL RESEARCH LABORATORY

Tor Circuit Priority

Scenarios to understand outbound queue congestion







99.775% of any two circuits are unshared

Performance Analysis

U.S. NAVAL RESEARCH LABORATORY

Kernel-Informed Socket Transport

Queuing delays in kernel output buffer

Problem 1: Circuit scheduling

Solution:

- Don’t handle sockets individually
- Process flush requests from all writable sockets

Problem 2: Flushing to sockets – buffer bloat

Solution:

- Don’t write if kernel can’t send
- Use TCP info to bound kernel writes

U.S. NAVAL RESEARCH LABORATORY

Network Simulation with Shadow

Enhanced Shadow with several missing TCP algorithms

- CUBIC congestion control
- Retransmission timers
- Selective acknowledgements (SACK)
- Forward acknowledgements (FACK)
- Fast retransmit/recovery

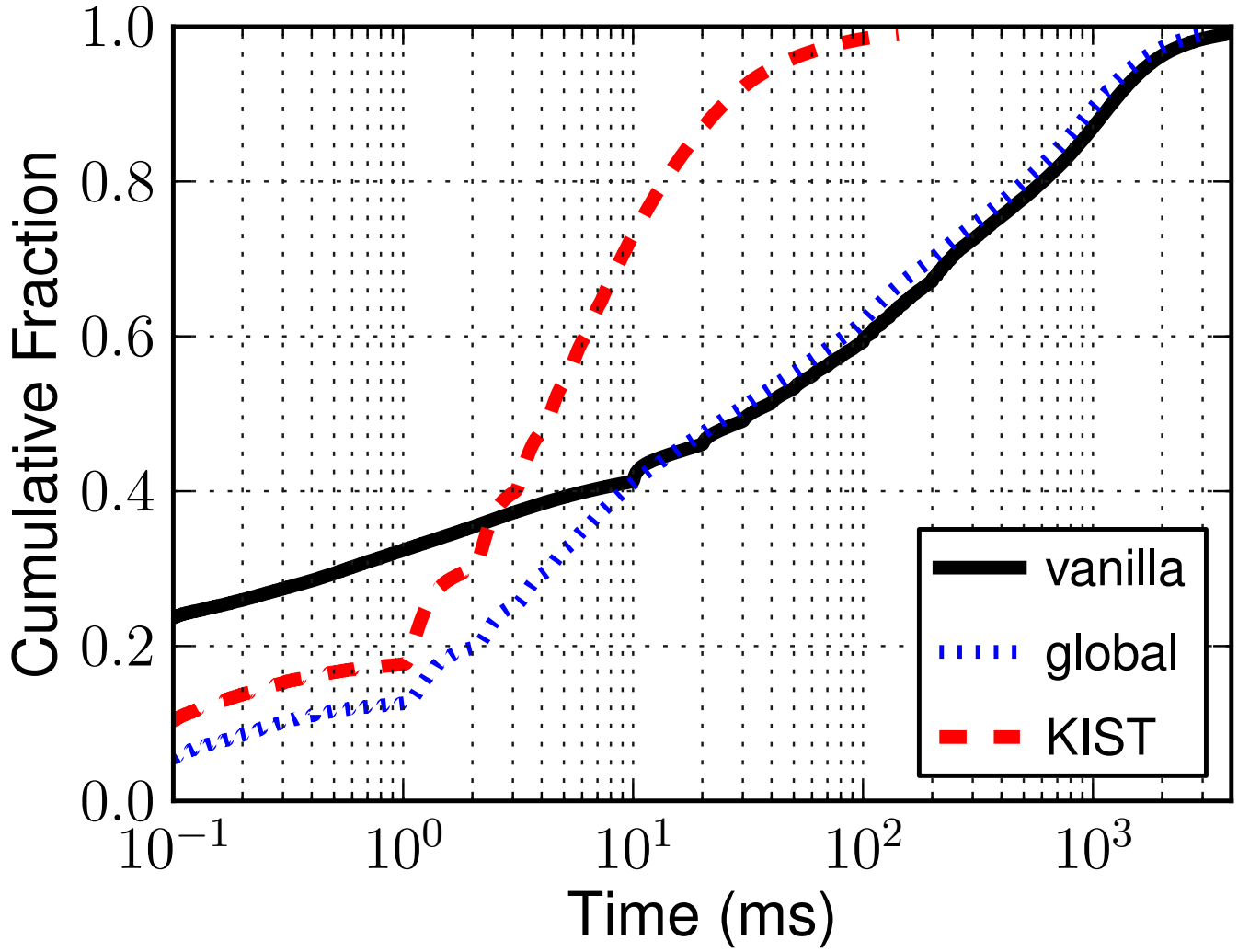
Designed largest known private Tor network

- 3600 relays and 12000 simultaneously active clients
- Internet topology graph: ~700k vertices and 1.3m edges

Analyze network-wide effects

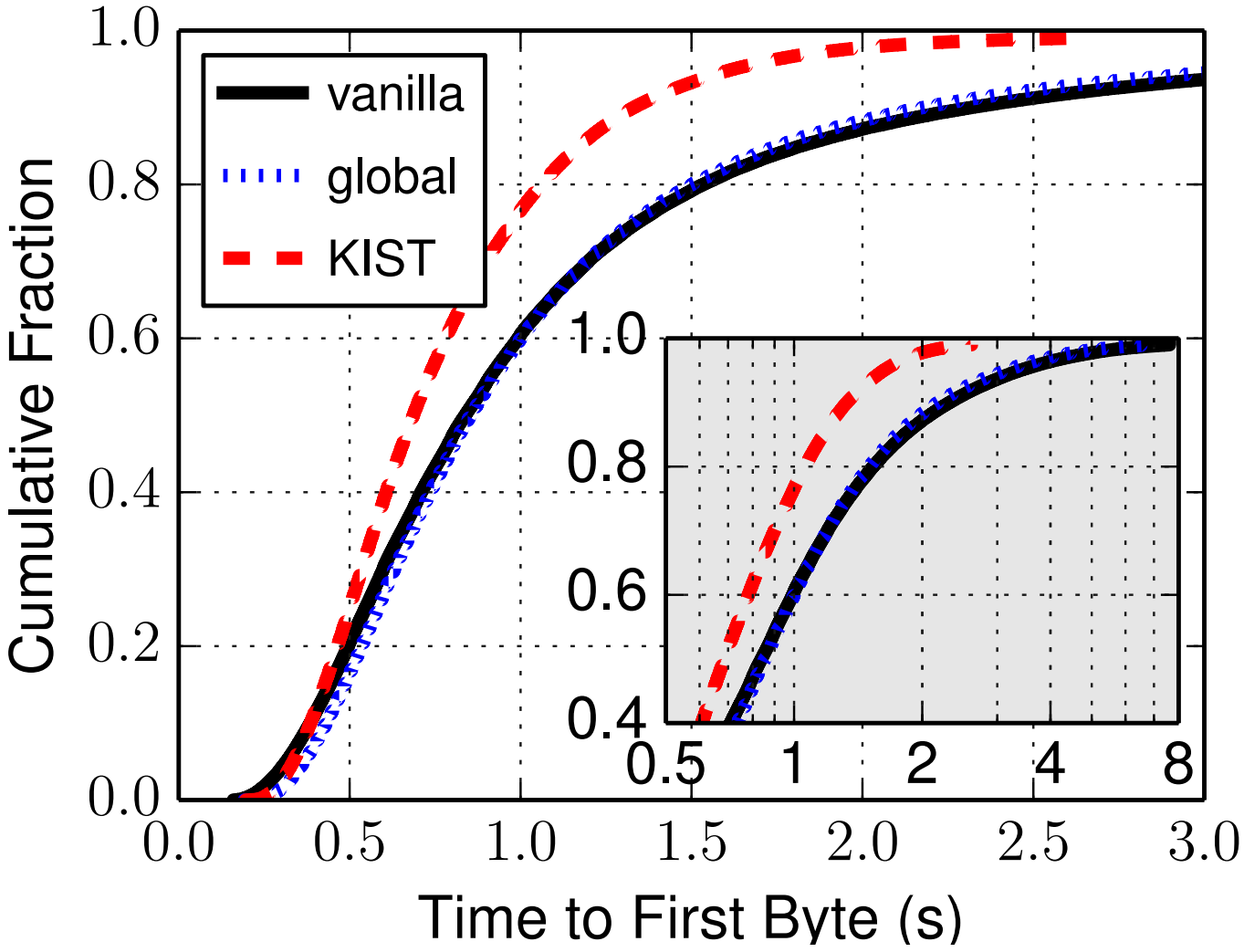
U.S. NAVAL RESEARCH LABORATORY

KIST Reduces Kernel Congestion



U.S. NAVAL RESEARCH LABORATORY

KIST Improves Network Latency



Denial of Service Attacks

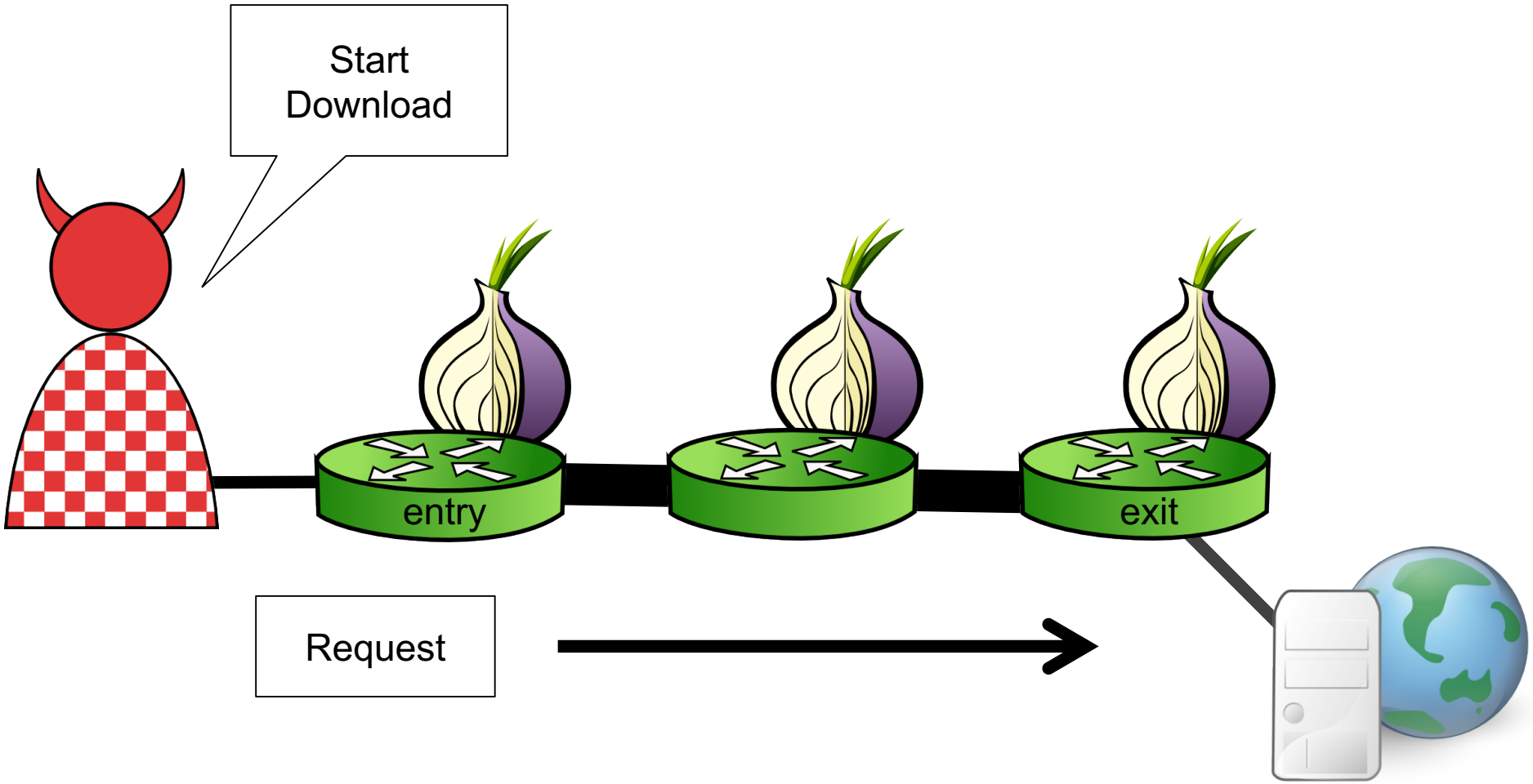
U.S. NAVAL RESEARCH LABORATORY

The Sniper Attack

- Memory-based denial of service (DoS) attack
- Exploits vulnerabilities in Tor's flow control protocol
- Can be used to disable arbitrary Tor relays

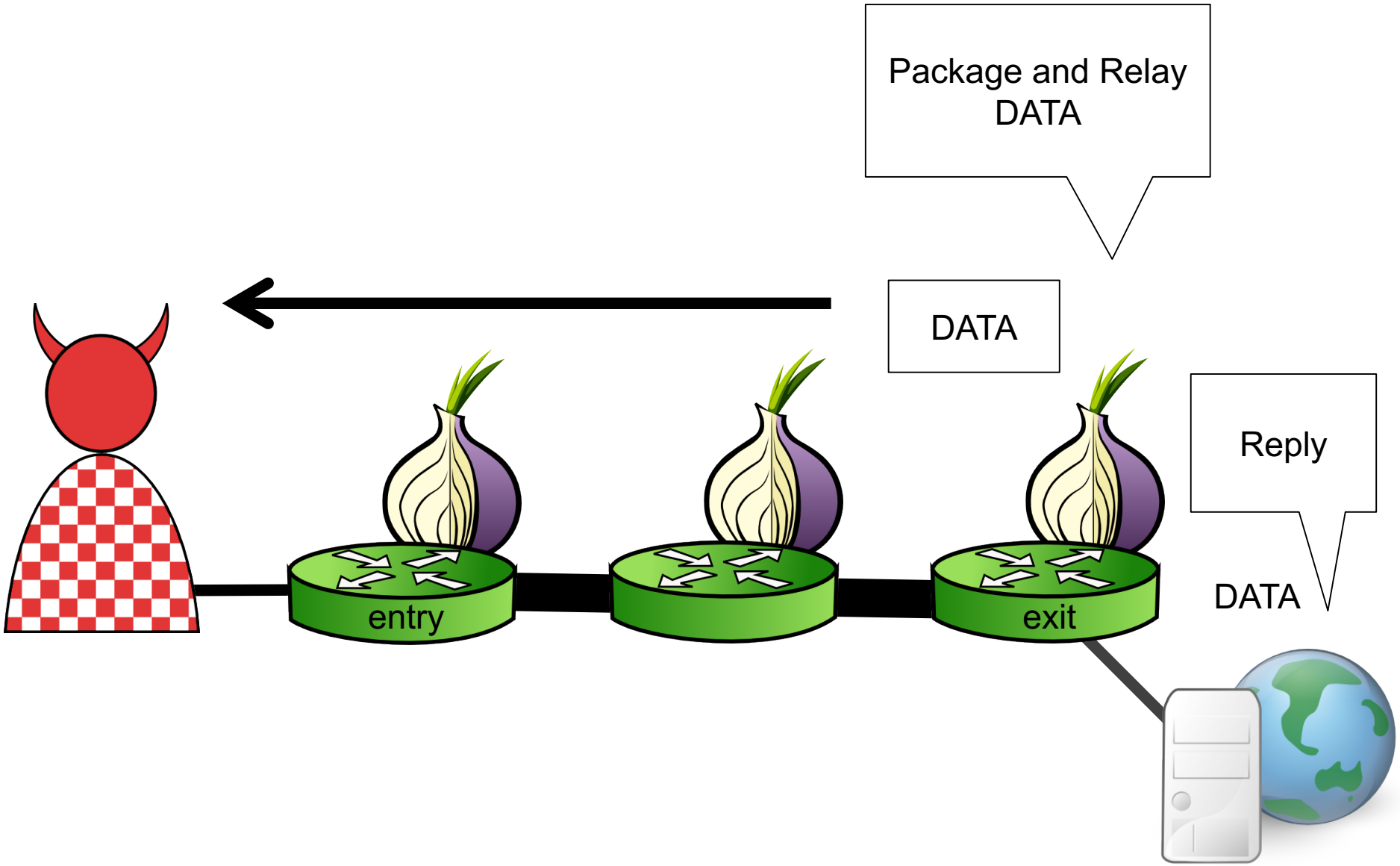
U.S. NAVAL RESEARCH LABORATORY

The Sniper Attack



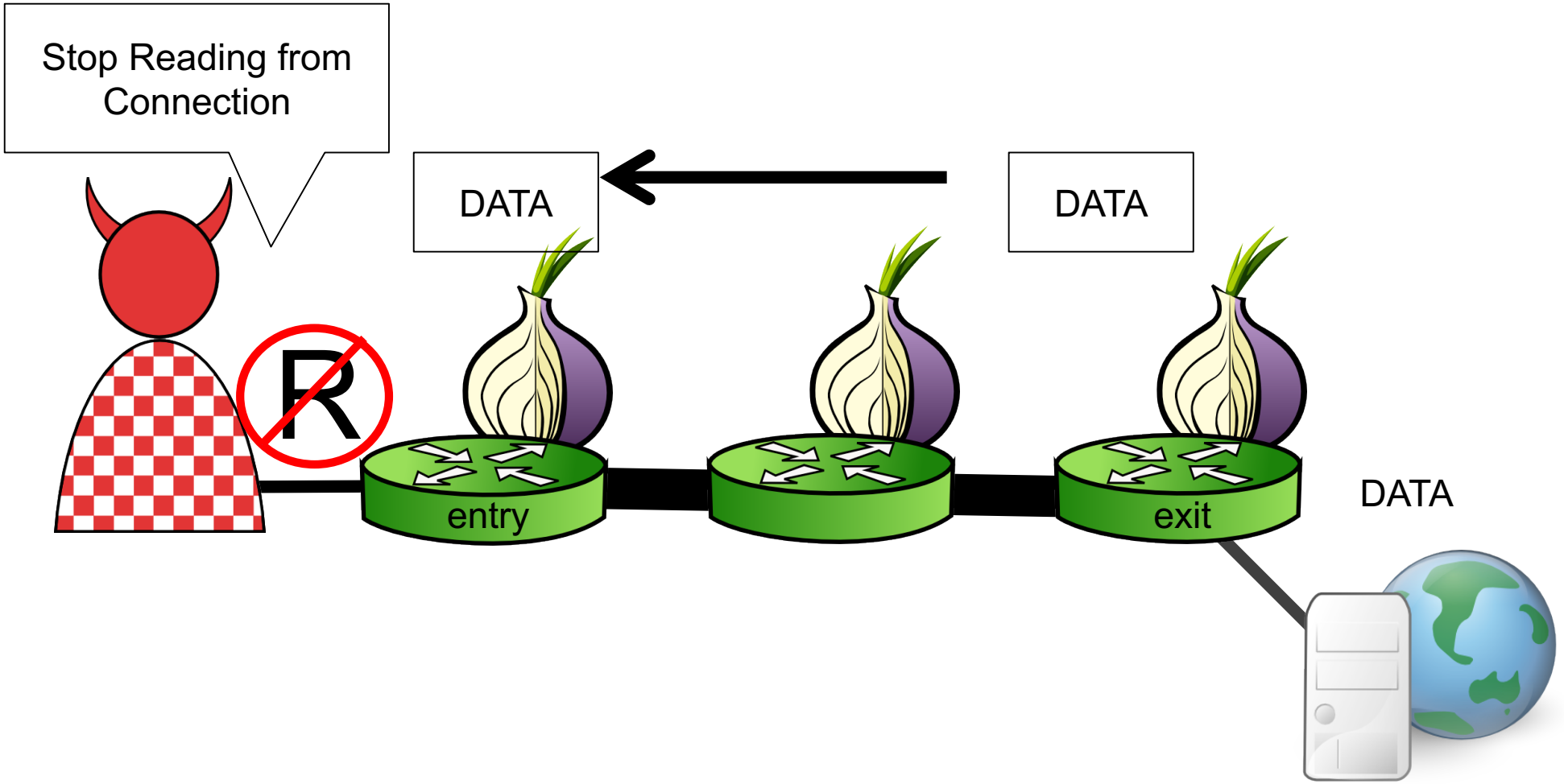
U.S. NAVAL RESEARCH LABORATORY

The Sniper Attack



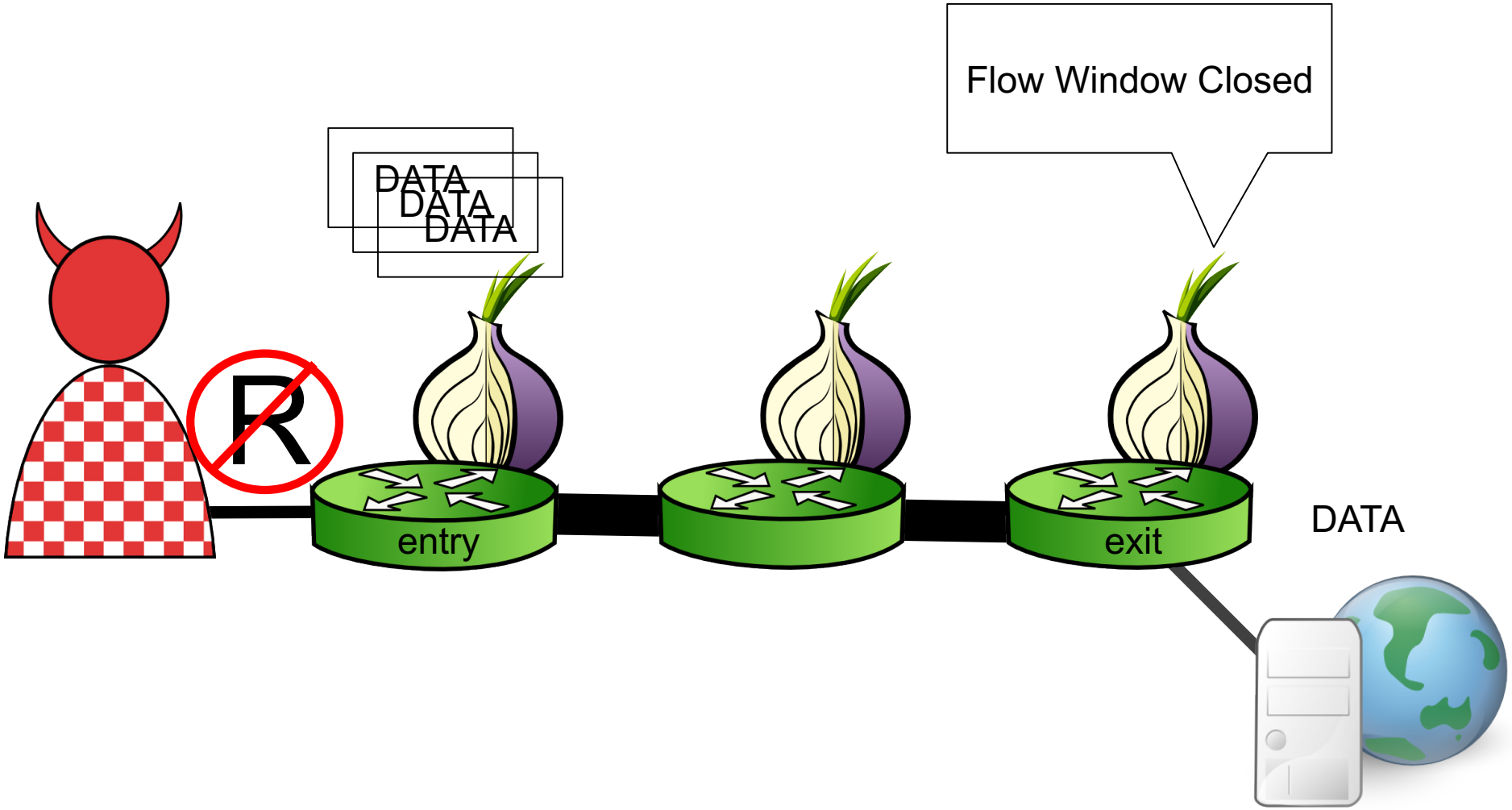
U.S. NAVAL RESEARCH LABORATORY

The Sniper Attack



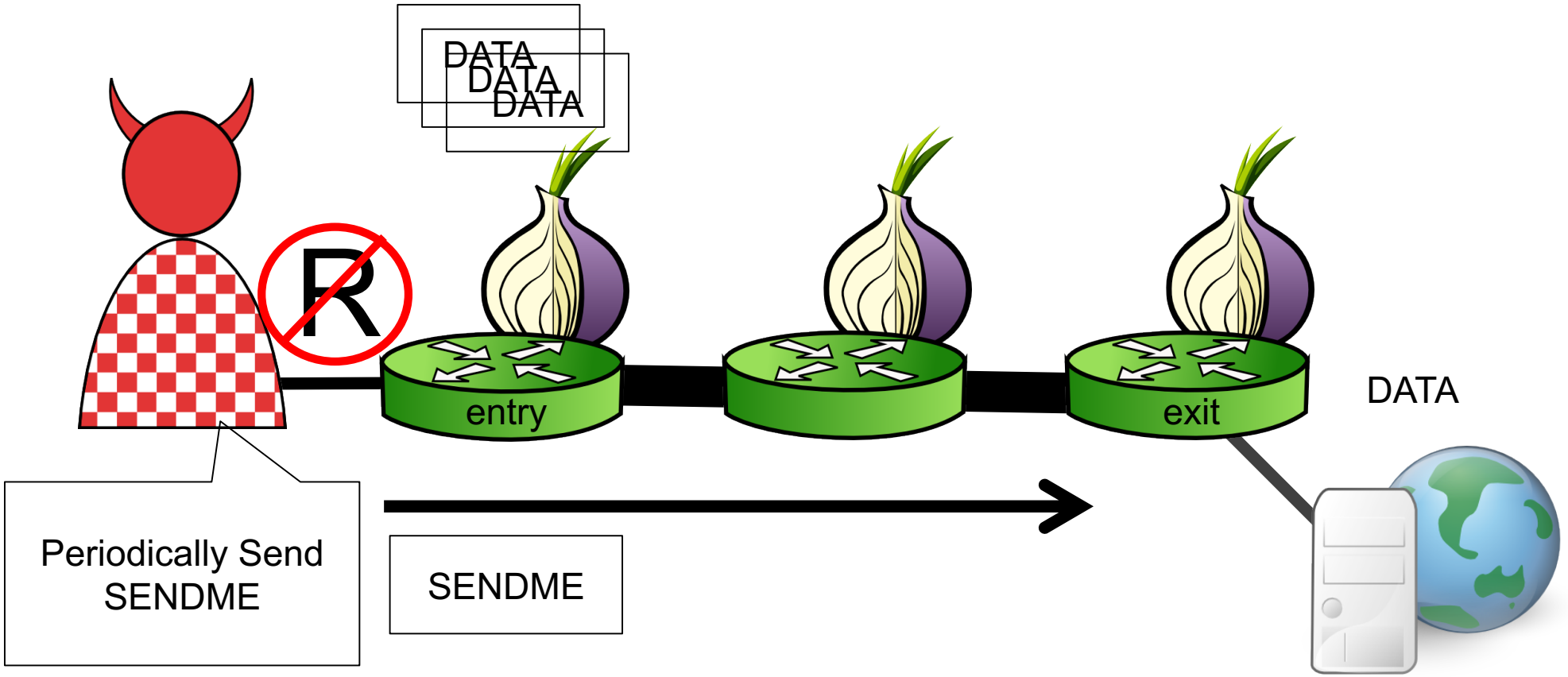
U.S. NAVAL RESEARCH LABORATORY

The Sniper Attack

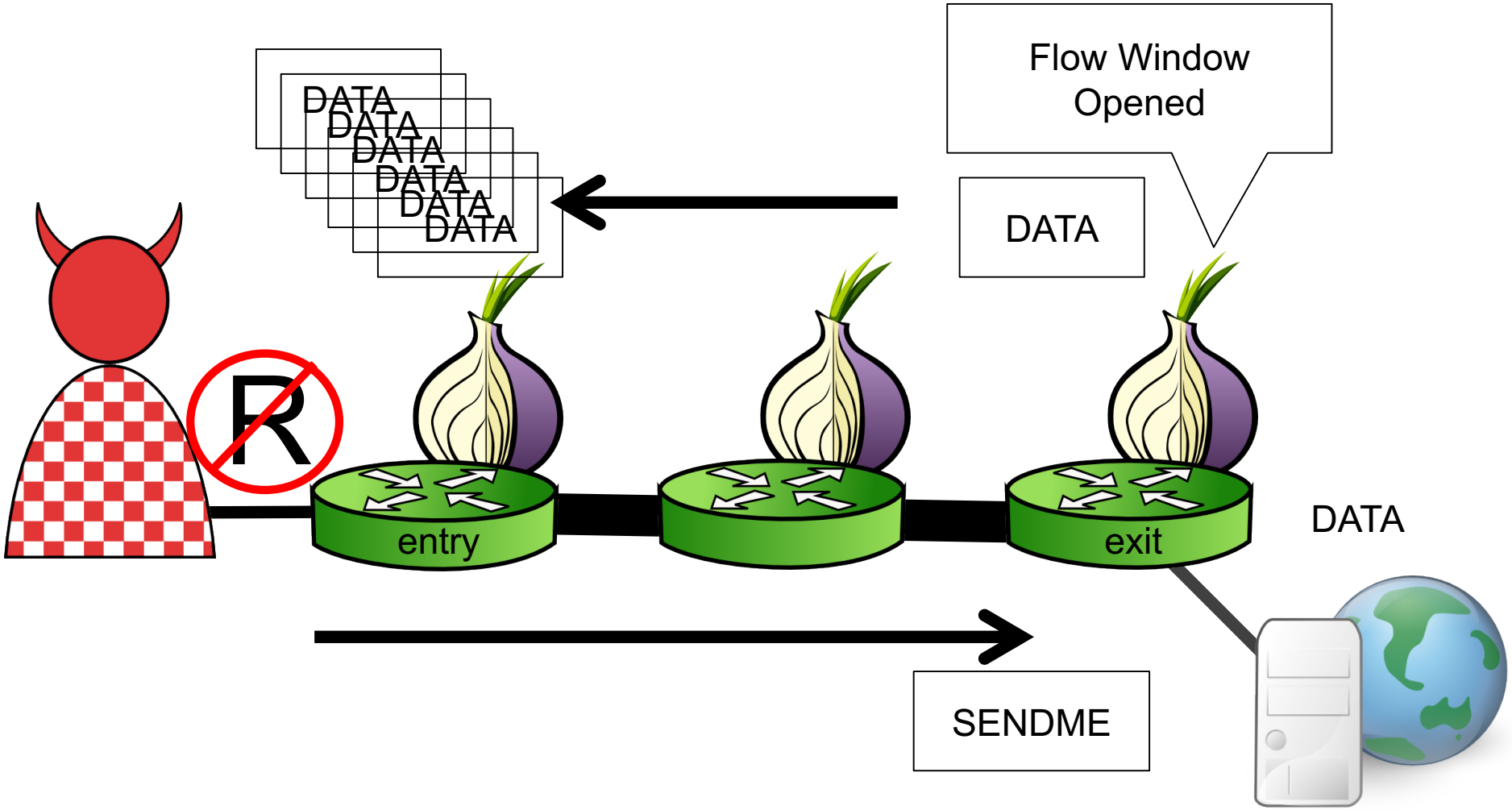


U.S. NAVAL RESEARCH LABORATORY

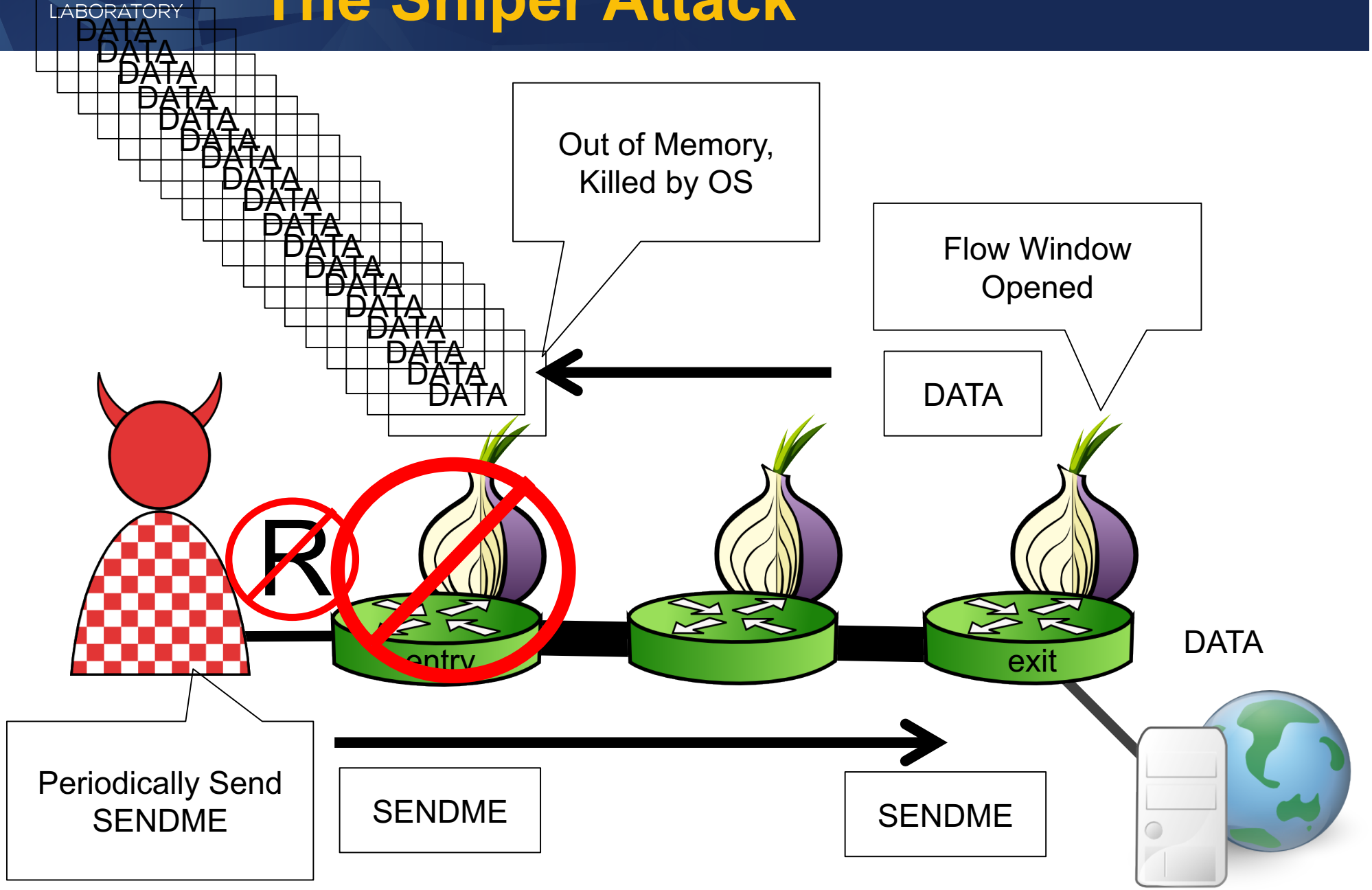
The Sniper Attack



The Sniper Attack



The Sniper Attack



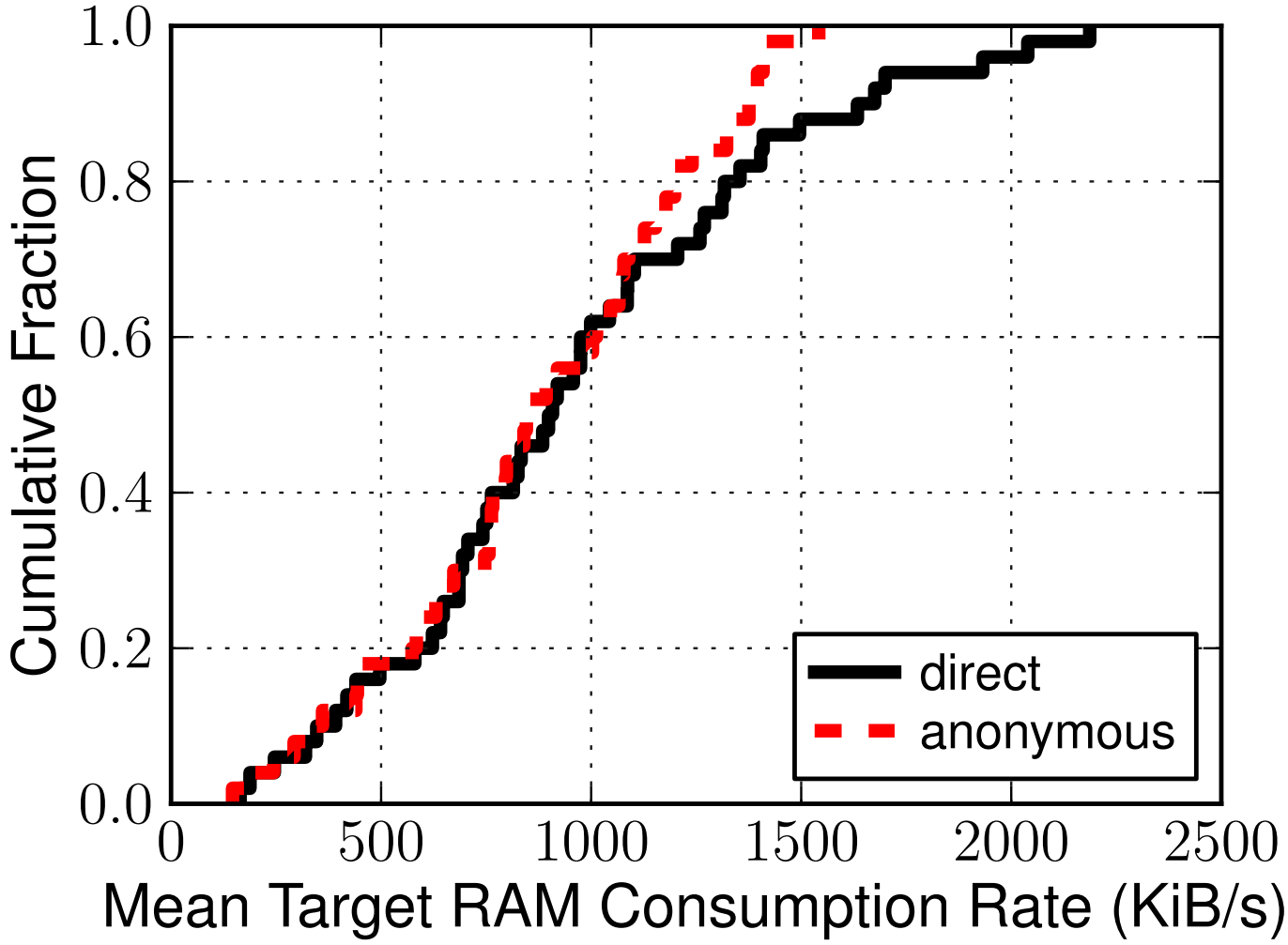
U.S. NAVAL RESEARCH LABORATORY

The Sniper Attack: Results

- Implemented Sniper Attack Prototype
  - Control Sybils via the Tor control protocol
- Tested in Shadow for safety
- Measured:
  - Victim memory consumption rate
  - Adversary bandwidth usage
- Developed defense, tested in Shadow, merged in Tor

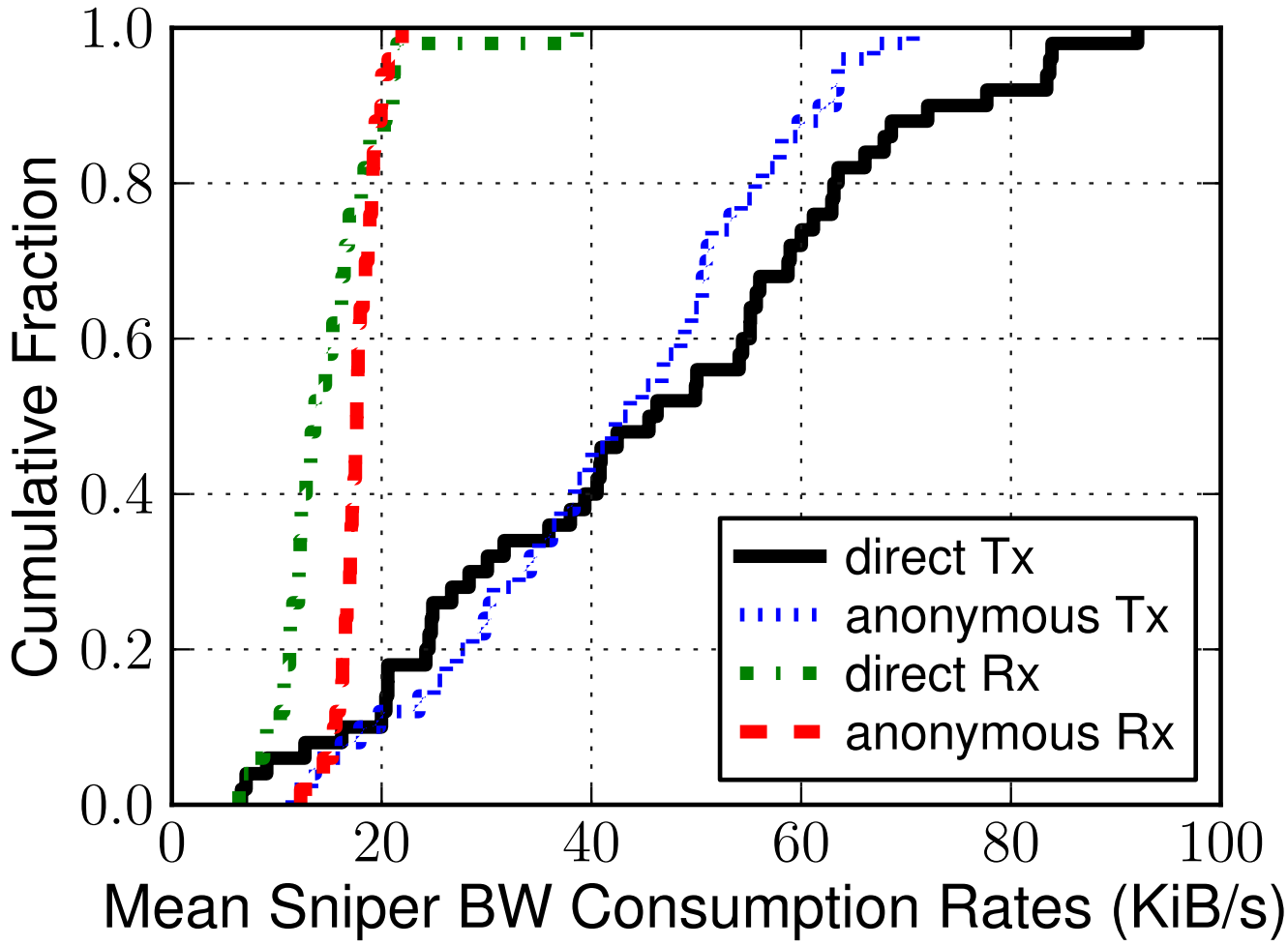
U.S. NAVAL RESEARCH LABORATORY

RAM Consumed at Victim



U.S. NAVAL RESEARCH LABORATORY

Bandwidth Consumed at Adversary



U.S. NAVAL RESEARCH LABORATORY

Speed of the Sniper Attack

		Direct		Anonymous	
Relay Groups	**Select %**	**1 GiB**	**8 GiB**	**1 GiB**	**8 GiB**
Top Entry	1.7	0:01	0:18	0:02	0:14
Top 5 Entries	6.5	0:08	1:03	0:12	1:37
Top 20 Entries	19	0:45	5:58	1:07	8:56
Top Exit	3.2	0:01	0:08	0:01	0:12
Top 5 Exits	13	0:05	0:37	0:07	0:57
Top 20 Exits	35	0:29	3:50	0:44	5:52

< 1 GiB RAM
< 50 KiB/s Downstream BW
< 100 KiB/s Upstream BW

Time (hours:minutes) to Consume RAM

Conclusion

U.S. NAVAL RESEARCH LABORATORY

Other Shadow Uses

- Tor
  - Latency and throughput correlation attacks
  - Denial of Service attacks (sockets, RAM, bandwidth)
  - Changes to path selection algorithms
  - Traffic admission control algorithms
  - Traffic scheduling and prioritization algorithms
  - Network load balancing algorithms
  - Process RAM consumption and optimization
- Network and memory attacks in Bitcoin
- Distributed secure multiparty computation algorithms
- Software debugging

U.S. NAVAL RESEARCH LABORATORY

Future Shadow Enhancements

- Distribute across physical machines
- Support for multiple programming languages
- Host mobility
- Internet routing, network modeling
- User behavior modeling
- CPU performance modeling
- User interface
- Support additional applications (HTTP clients/server, bitcoin, etc.)
- Improve code stability, documentation, testing, etc.

Questions

Dr. Rob Jansen
U.S. Naval Research Laboratory
Center for High Assurance Computer Systems
rob.g.jansen@nrl.navy.mil
robgjansen.com, @robgjansen

The Shadow Simulator
shadow.github.io
github.com/shadow