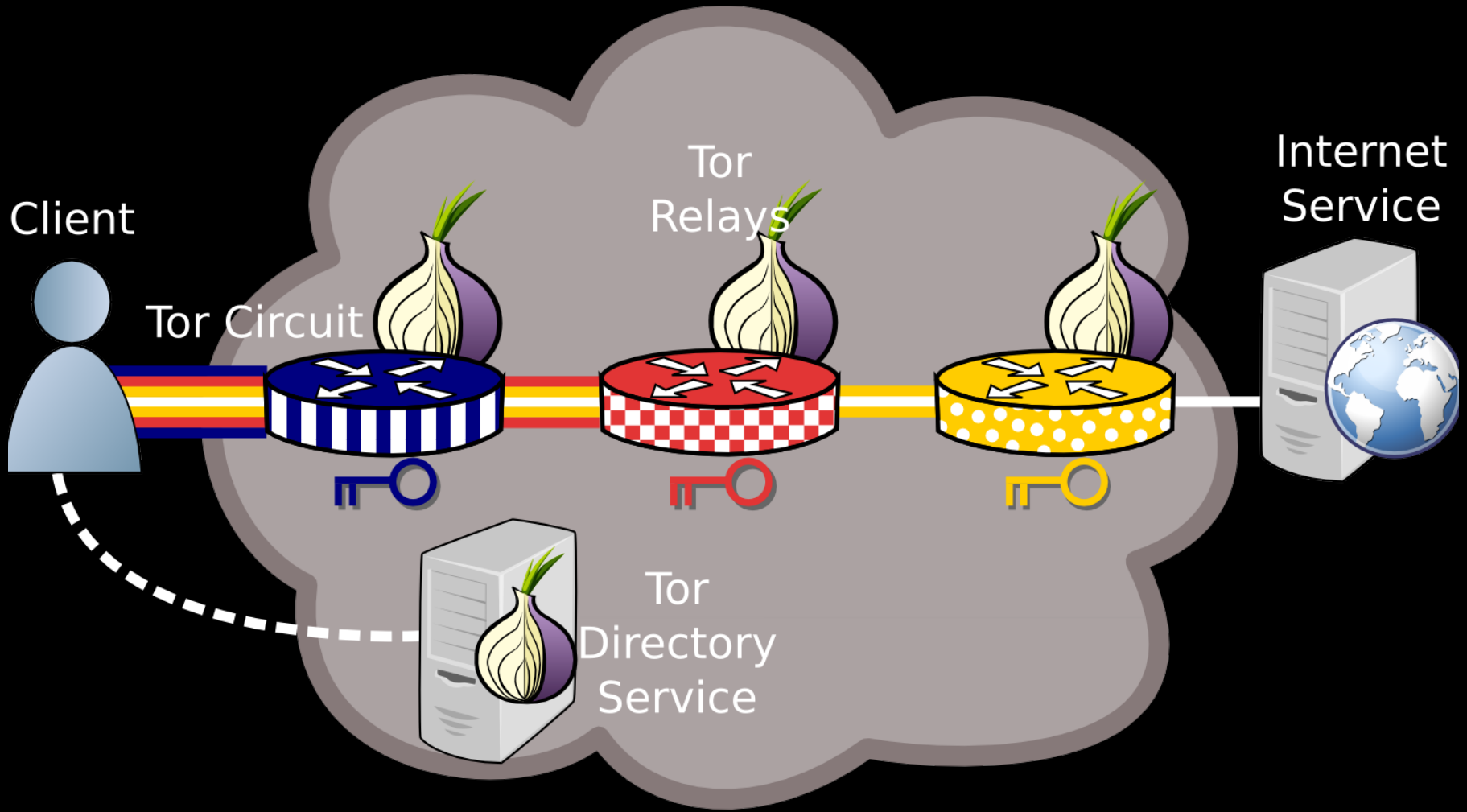
Never Been KIST: Tor's Congestion Management Blossoms with Kernel-Informed Socket Transport

23rd USENIX Security Symposium

August 20th 2014

Rob Jansen	US Naval Research Laboratory
John Geddes	University of Minnesota
Chris Wacek	Georgetown University
Micah Sherr	Georgetown University
Paul Syverson	US Naval Research Laboratory

Anonymous Communication: Tor



Tor is Slow!!! Research*

- PCTCP: Per-Circuit TCP-over-IPsec Transport for Anonymous Communication Overlay Networks (CCS ‘13)
- Reducing Latency in Tor Circuits with Unordered Delivery (FOCI ‘13)
- How Low Can You Go: Balancing Performance with Anonymity in Tor (PETS ‘13)
- The Path Less Travelled: Overcoming Tor's Bottlenecks with Traffic Splitting (PETS ’13)
- An Empirical Evaluation of Relay Selection in Tor (NDSS ‘13)
- LIRA: Lightweight Incentivized Routing for Anonymity (NDSS ‘13)
- Improving Performance and Anonymity in the Tor Network (IPCCC ‘12)
- Enhancing Tor's Performance using Real-time Traffic Classification (CCS ’12)
- Torchestra: Reducing interactive traffic delays over Tor (WPES ‘12)
- Throttling Tor Bandwidth Parasites (USENIX Sec ‘12)
- LASTor: A Low-Latency AS-Aware Tor Client (Oakland ‘12)
- Congestion-aware Path Selection for Tor (FC ‘12)

*Not a comprehensive list

Tor is Slow!!! Research*

- PCTCP: Per-Circuit TCP-over-IPsec Transport for Anonymous Communication Overlay Networks (CCS '13)
- Reducing Latency in Tor Circuits with Unordered Delivery (FOCI '13)
- How [illegible] (PETS '13)
- The [illegible]litting (PETS '13)
- An Er[illegible]
- LIRA: [illegible]
- Impro[illegible])
- Enha[illegible]CS '12)
- Torchestra: Reducing interactive traffic delays over Tor (WPES '12)
- Throttling Tor Bandwidth Parasites (USENIX Sec '12)
- LASTor: A Low-Latency AS-Aware Tor Client (Oakland '12)
- Congestion-aware Path Selection for Tor (FC '12)

Where?

*Not a comprehensive list

This Talk

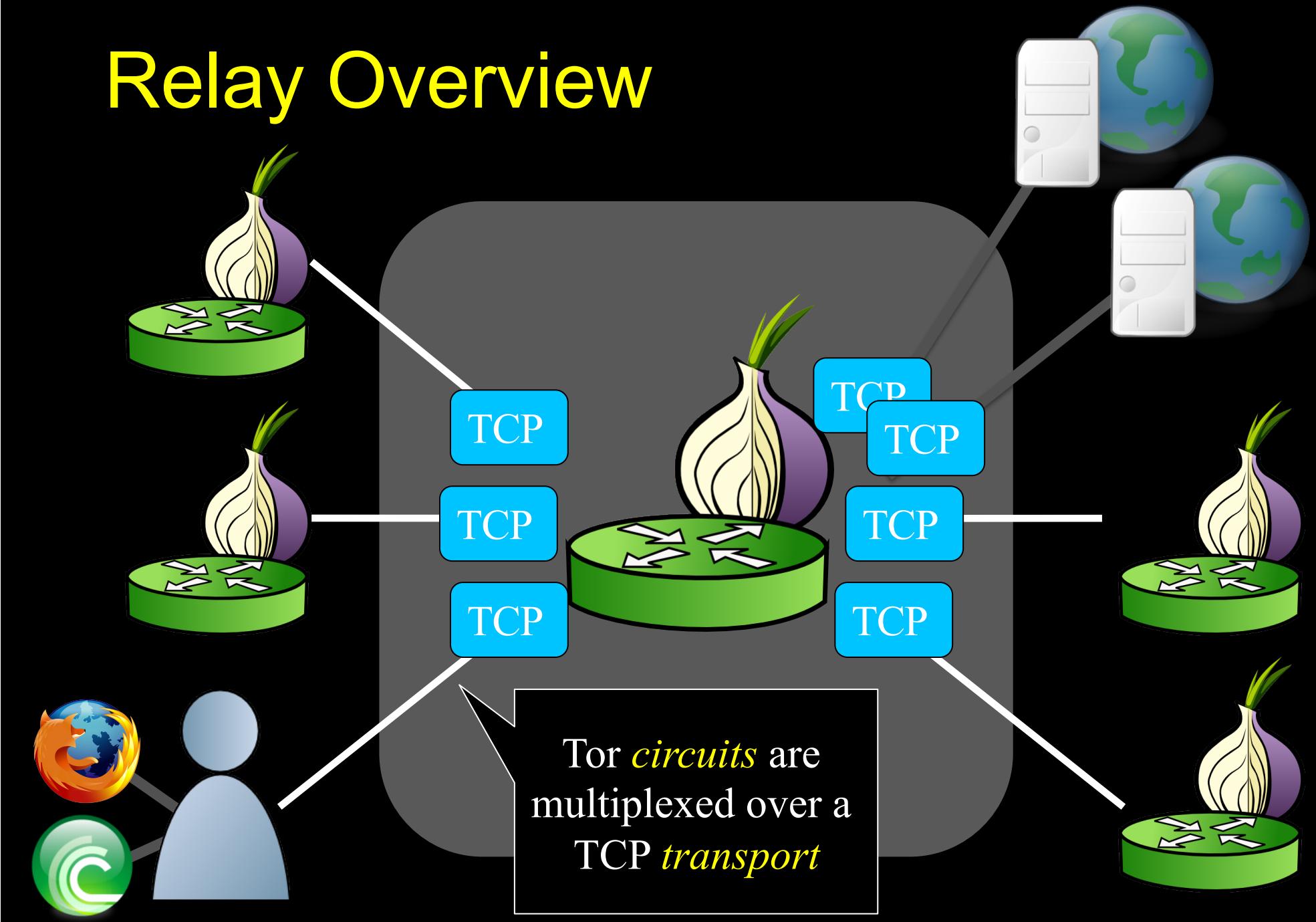
- Where is Tor slow?
  - Measure public Tor and private Shadow-Tor networks
  - Identify circuit scheduling and socket flushing problems
- Design KIST: Kernel-Informed Socket Transport
  - Use TCP snd_cwnd to limit socket writes
- Evaluate KIST Performance and Security
  - Reduces kernel and end-to-end circuit congestion
  - Throughput attacks unaffected, speeds up latency attacks

Outline

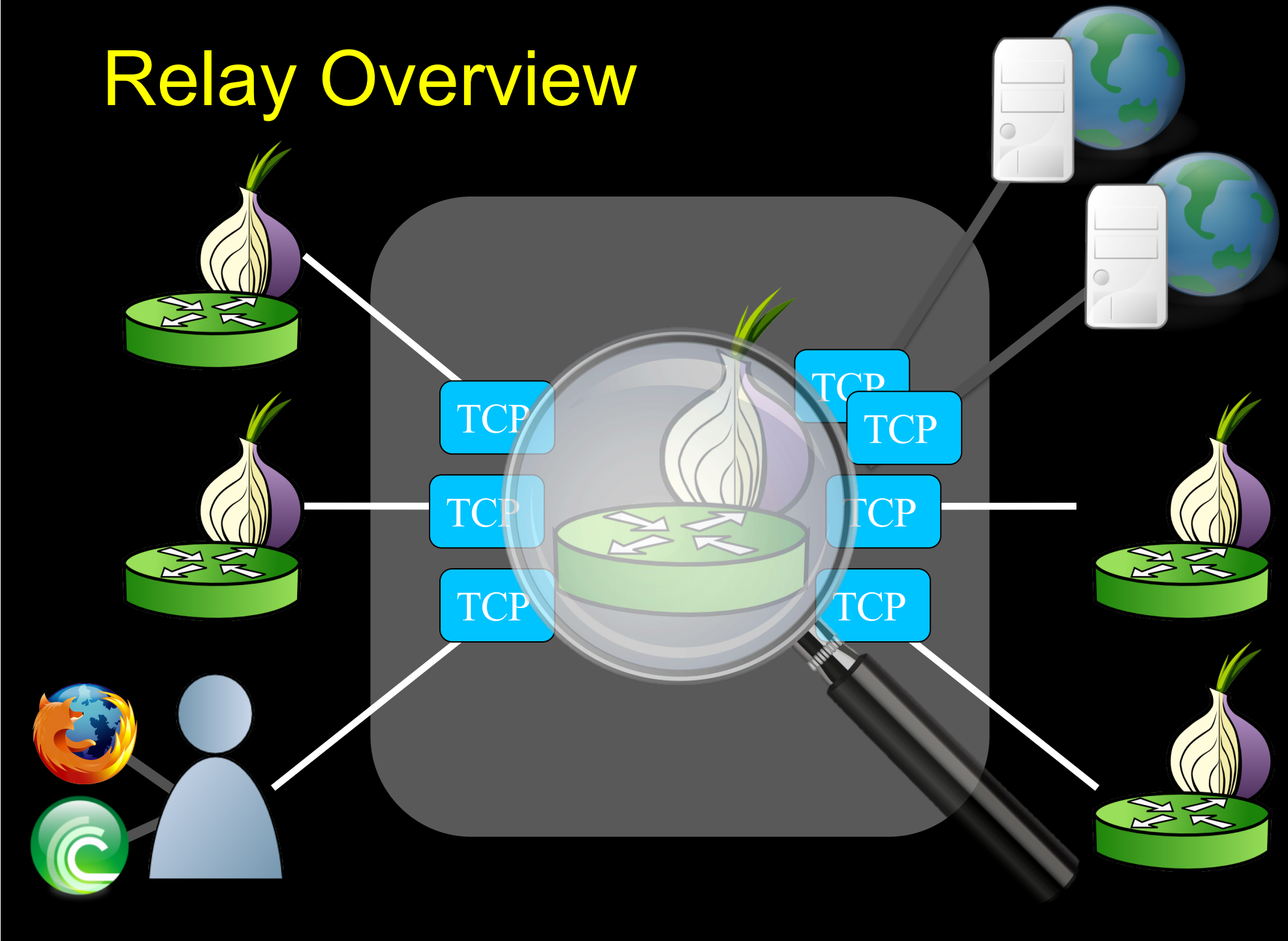
- Background
- Instrument Tor, measure congestion
- Analyze causes of congestion
- Design and evaluate KIST
  - Performance
  - Security

Relay Overview

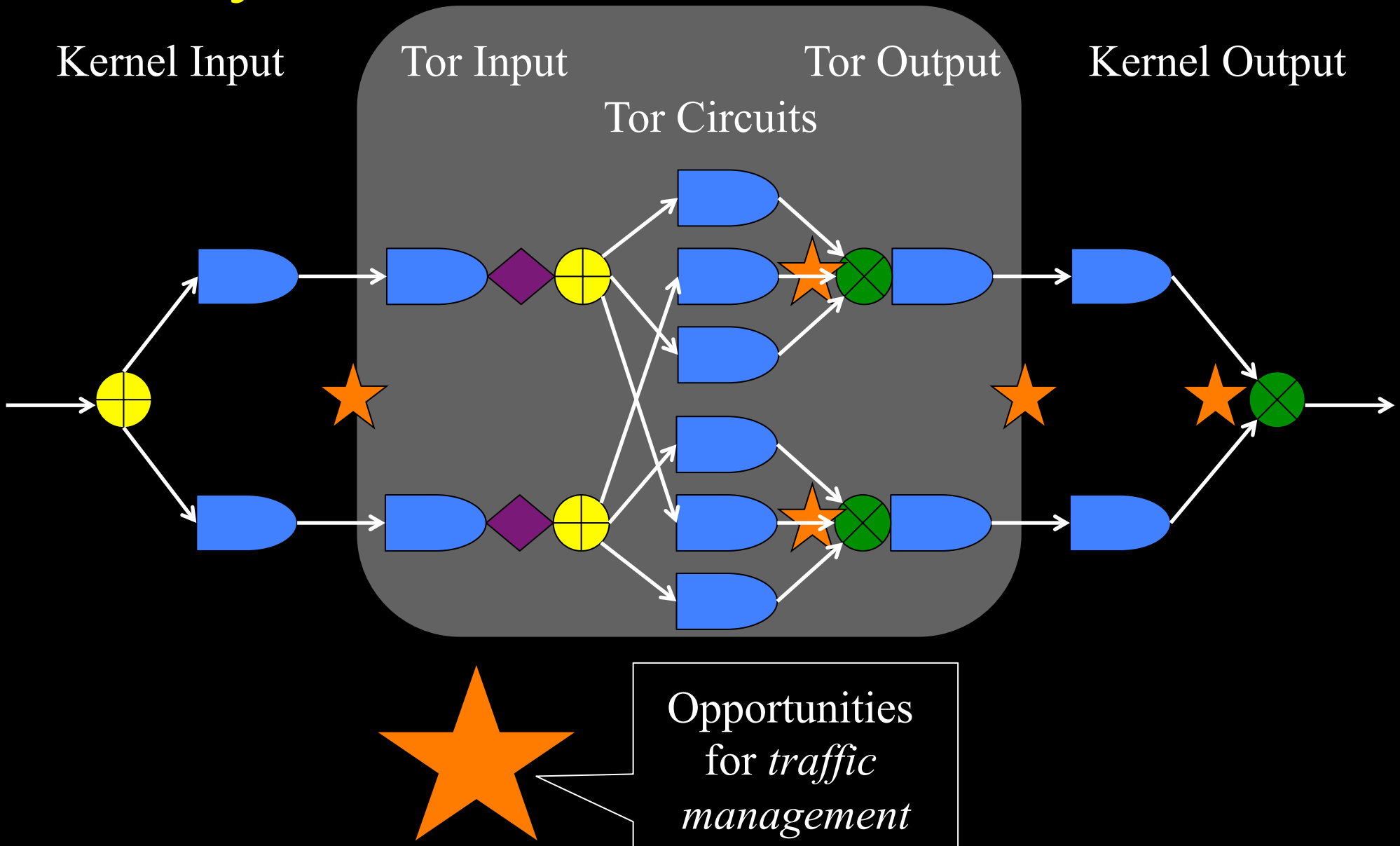
Relay Overview



Relay Overview



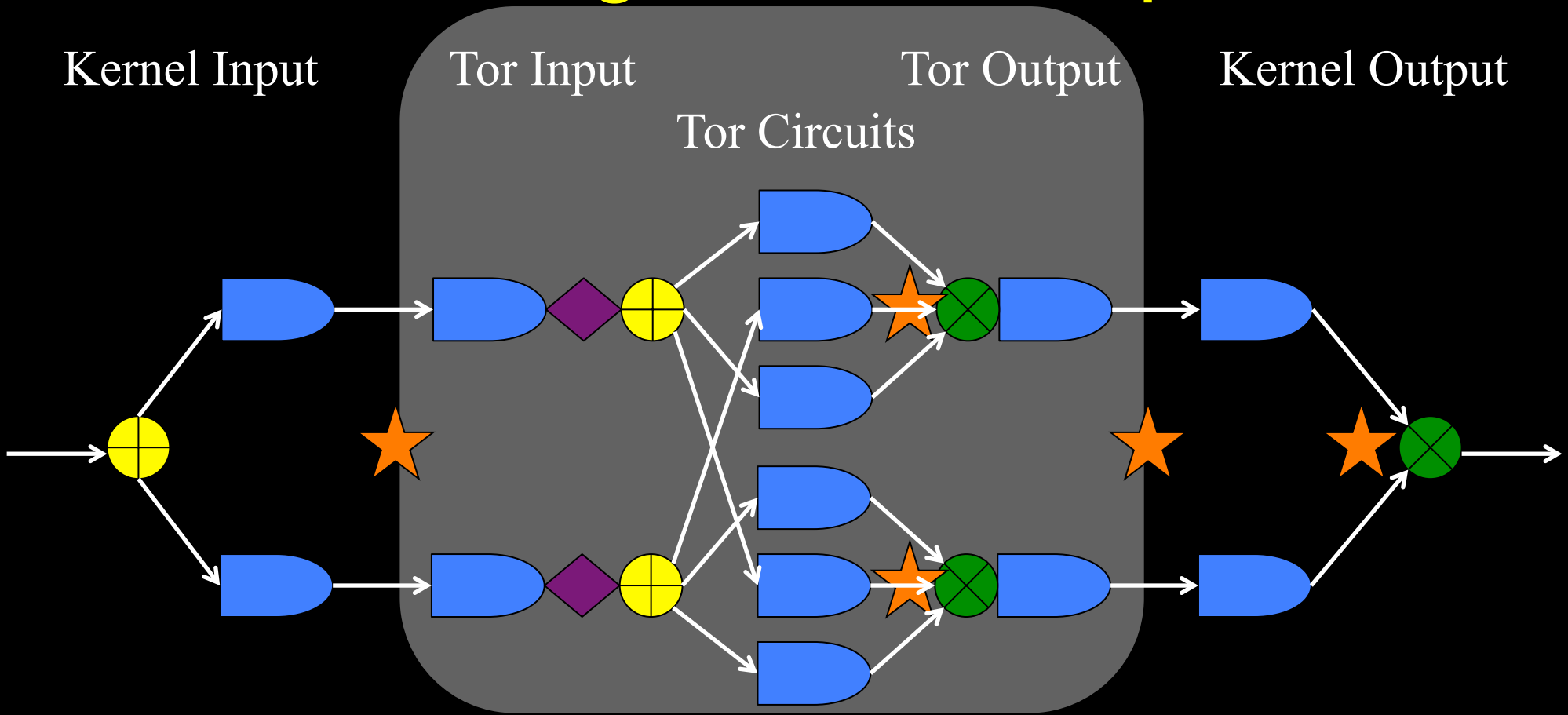
Relay Internals



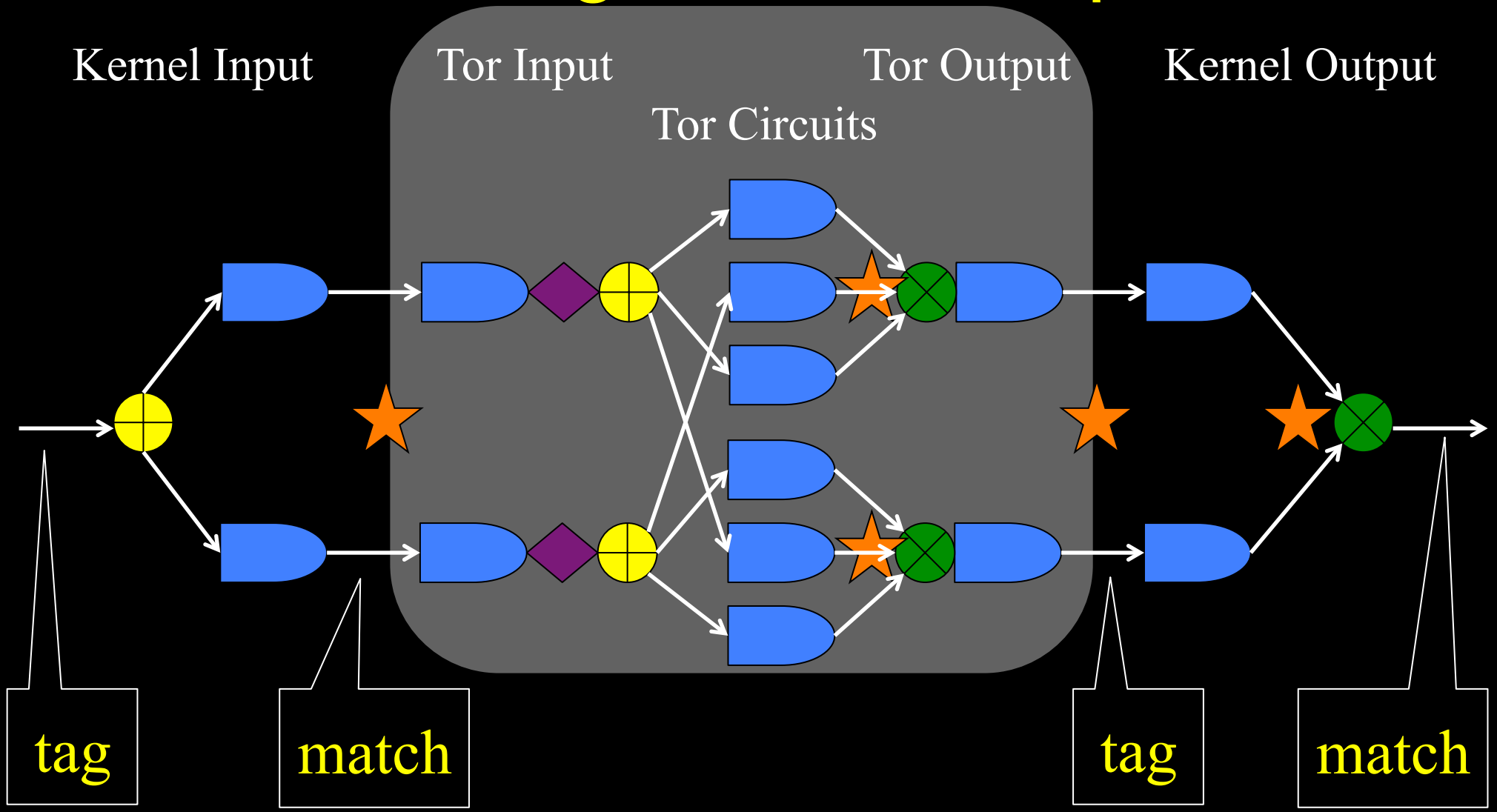
Outline

- ~~Background~~
- Instrument Tor, measure congestion
- Analyze causes of congestion
- Design and evaluate KIST
  - Performance
  - Security

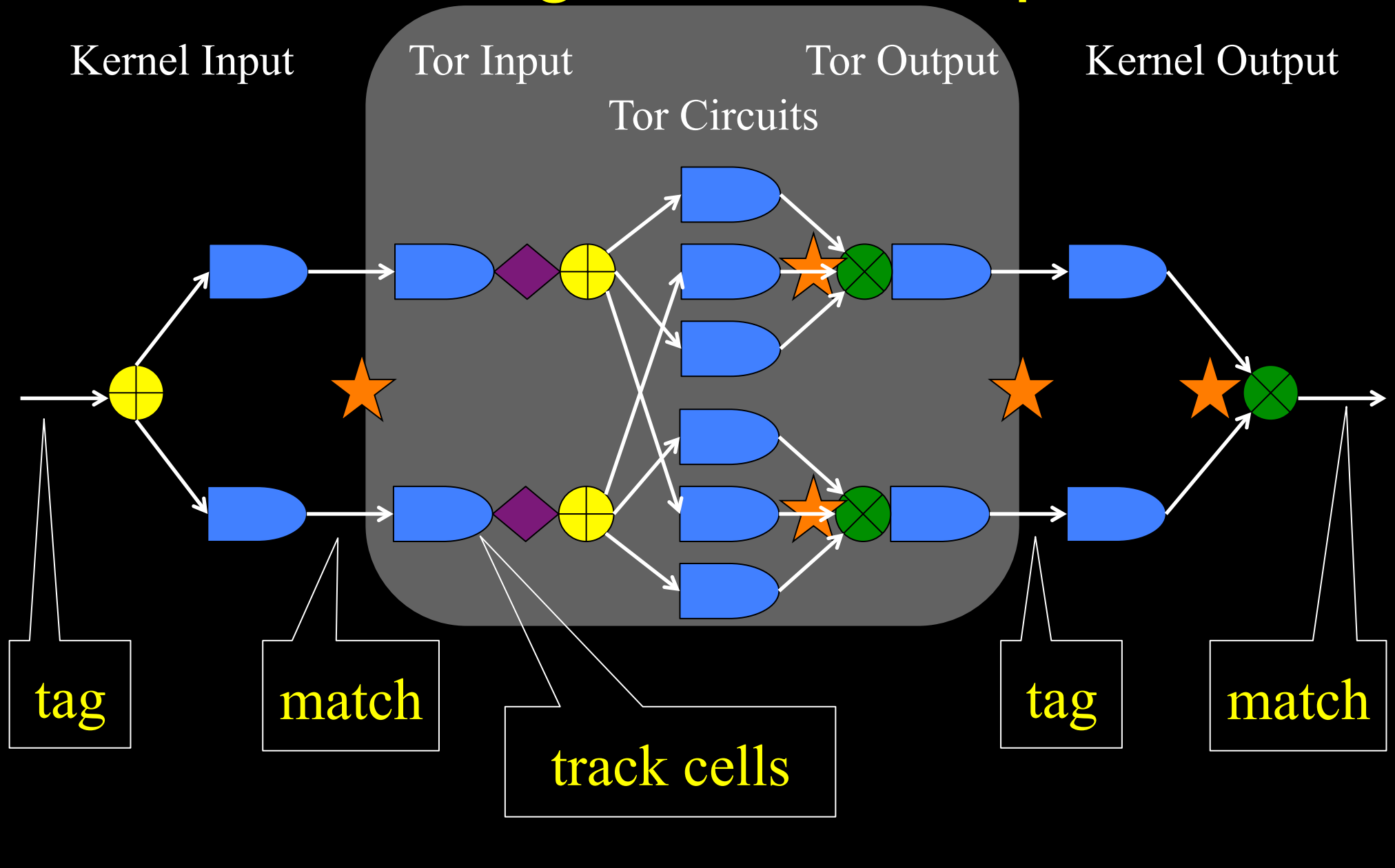
Live Tor Congestion - libkqtime



Live Tor Congestion - libkqtime



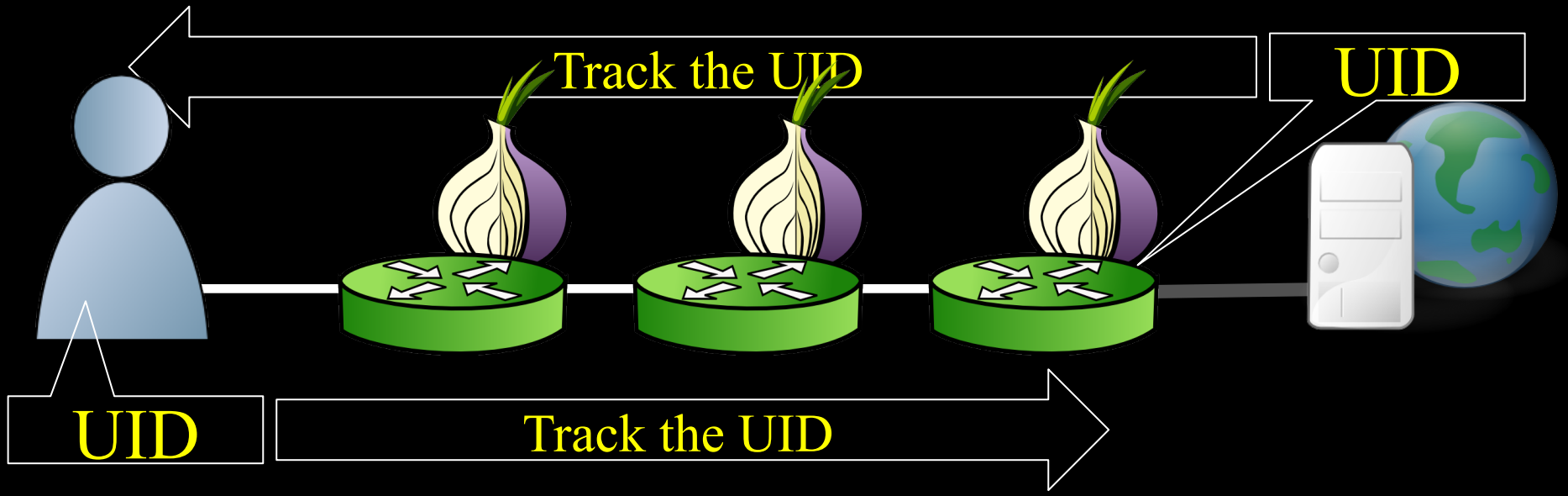
Live Tor Congestion - libkqtime



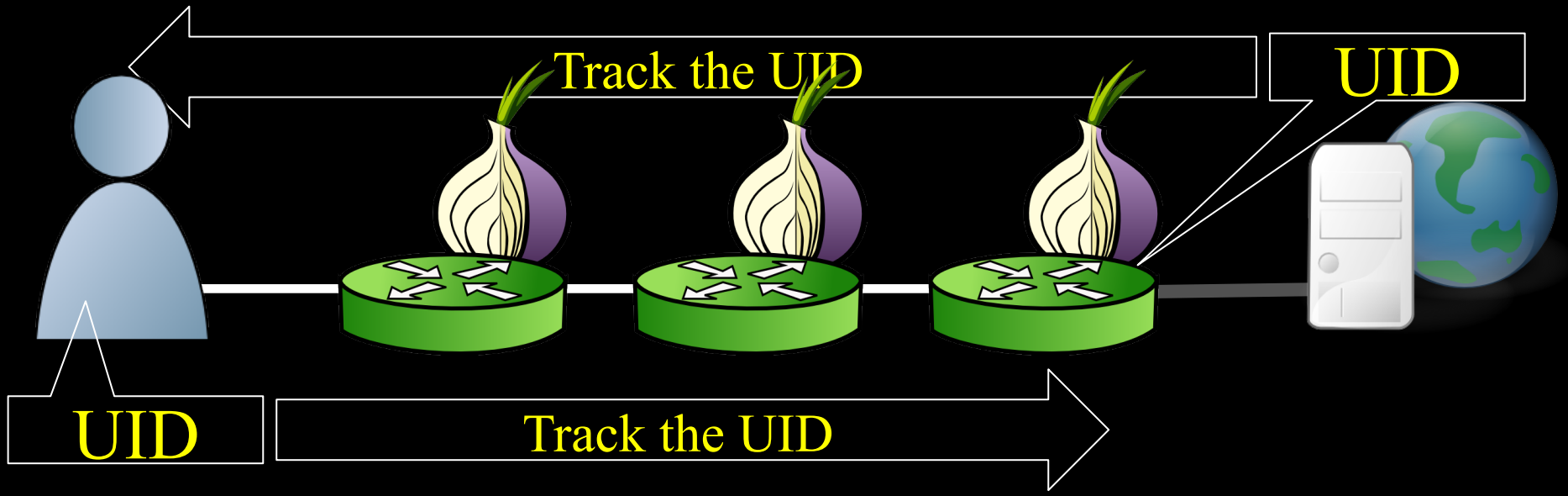
Shadow Network Simulation

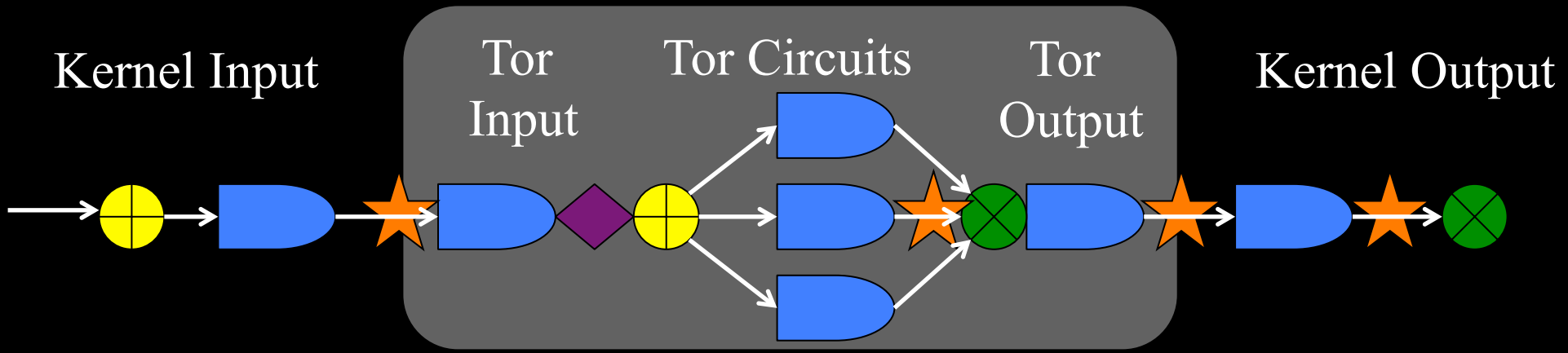
- Enhanced Shadow with several missing TCP algorithms
  - CUBIC congestion control
  - Retransmission timers
  - Selective acknowledgements (SACK)
  - Forward acknowledgements (FACK)
  - Fast retransmit/recovery
- Designed largest known private Tor network
  - 3600 relays and 12000 simultaneously active clients
  - Internet topology graph: ~700k nodes and 1.3m links

Shadow-Tor Congestion – UIDs



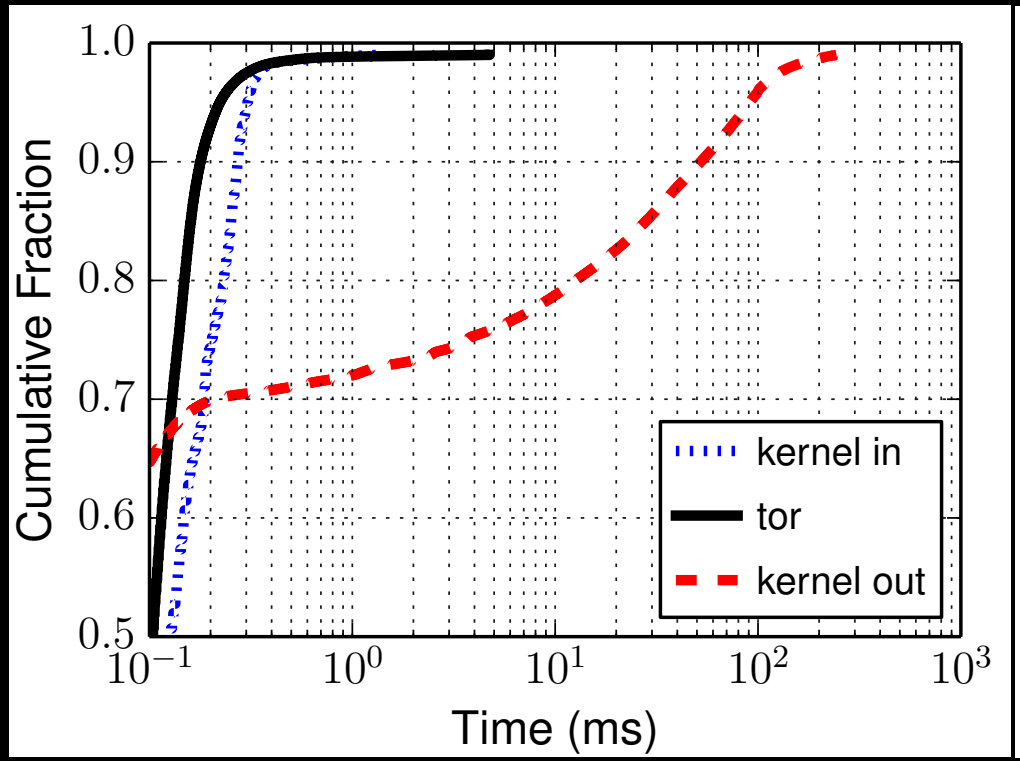
Shadow-Tor Congestion – UIDs



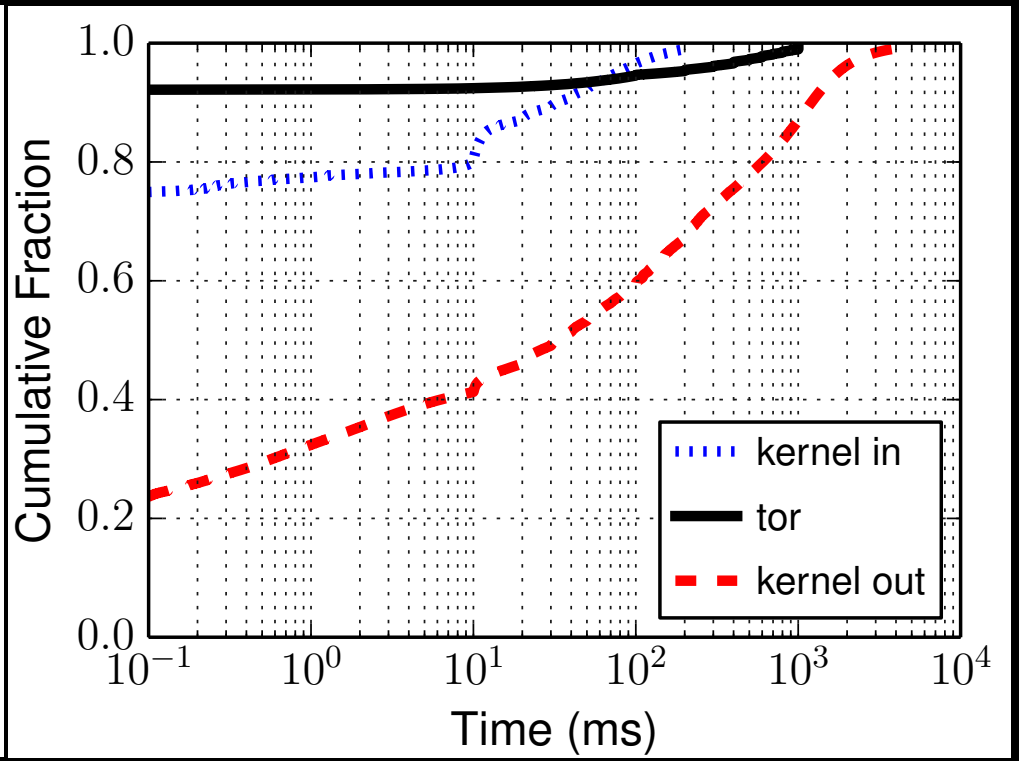


Tor and Shadow-Tor Congestion

Live-Tor



Shadow-Tor

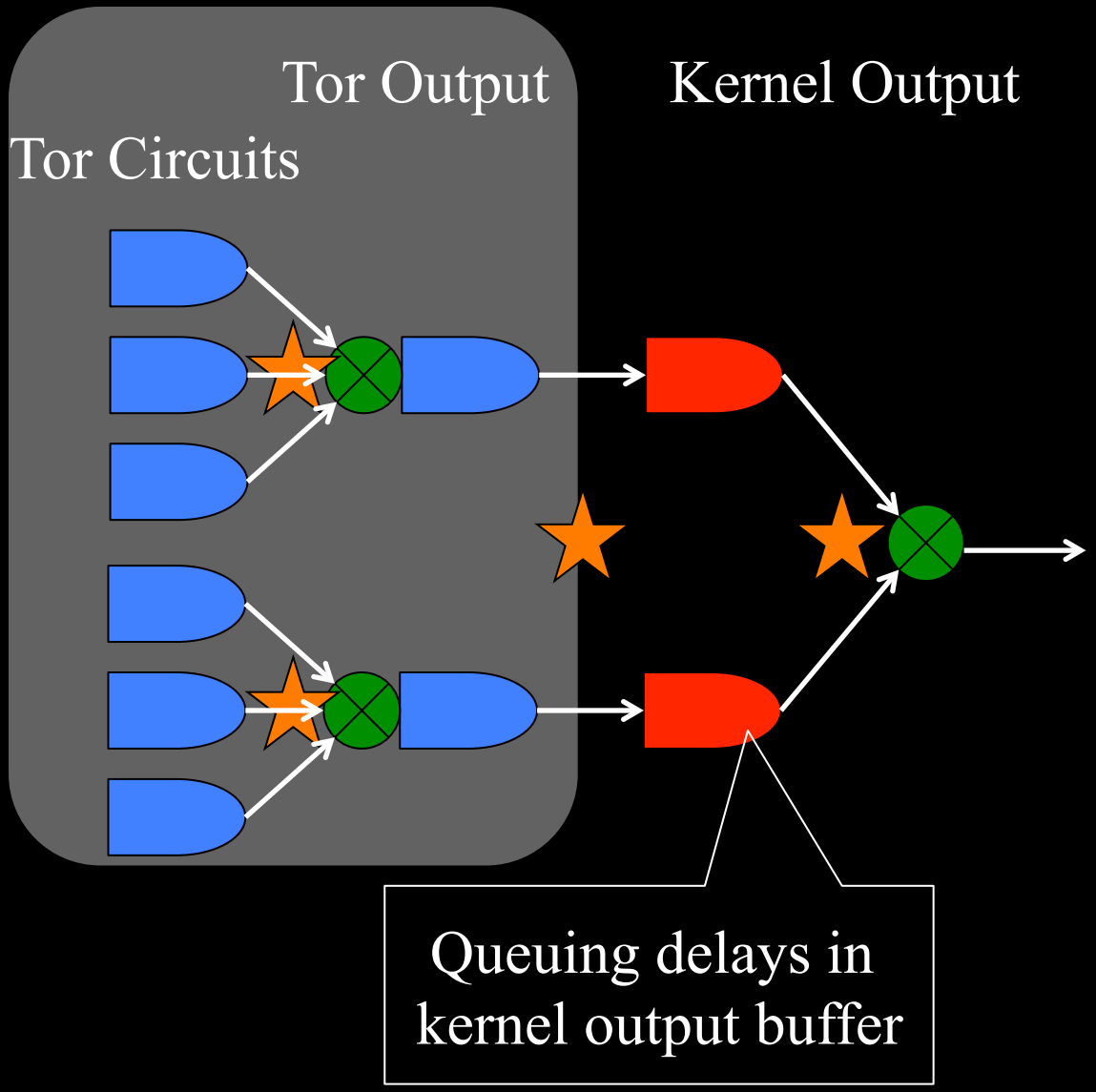


Congestion occurs almost exclusively in outbound kernel buffers

Outline

- ~~Background~~
- ~~Instrument Tor, measure congestion~~
- Analyze causes of congestion
- Design and evaluate KIST
  - Performance
  - Security

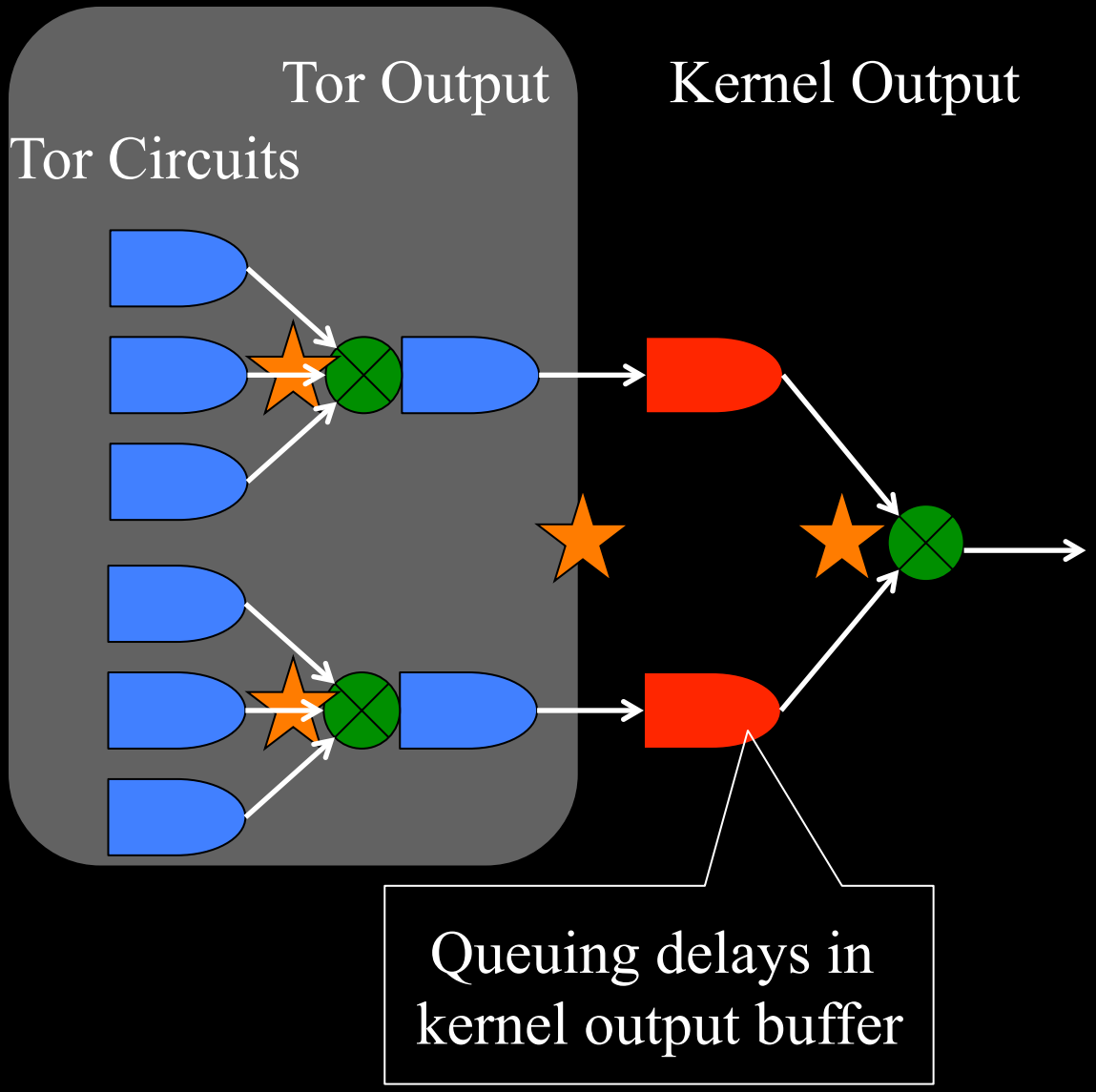
Analyzing Causes of Congestion



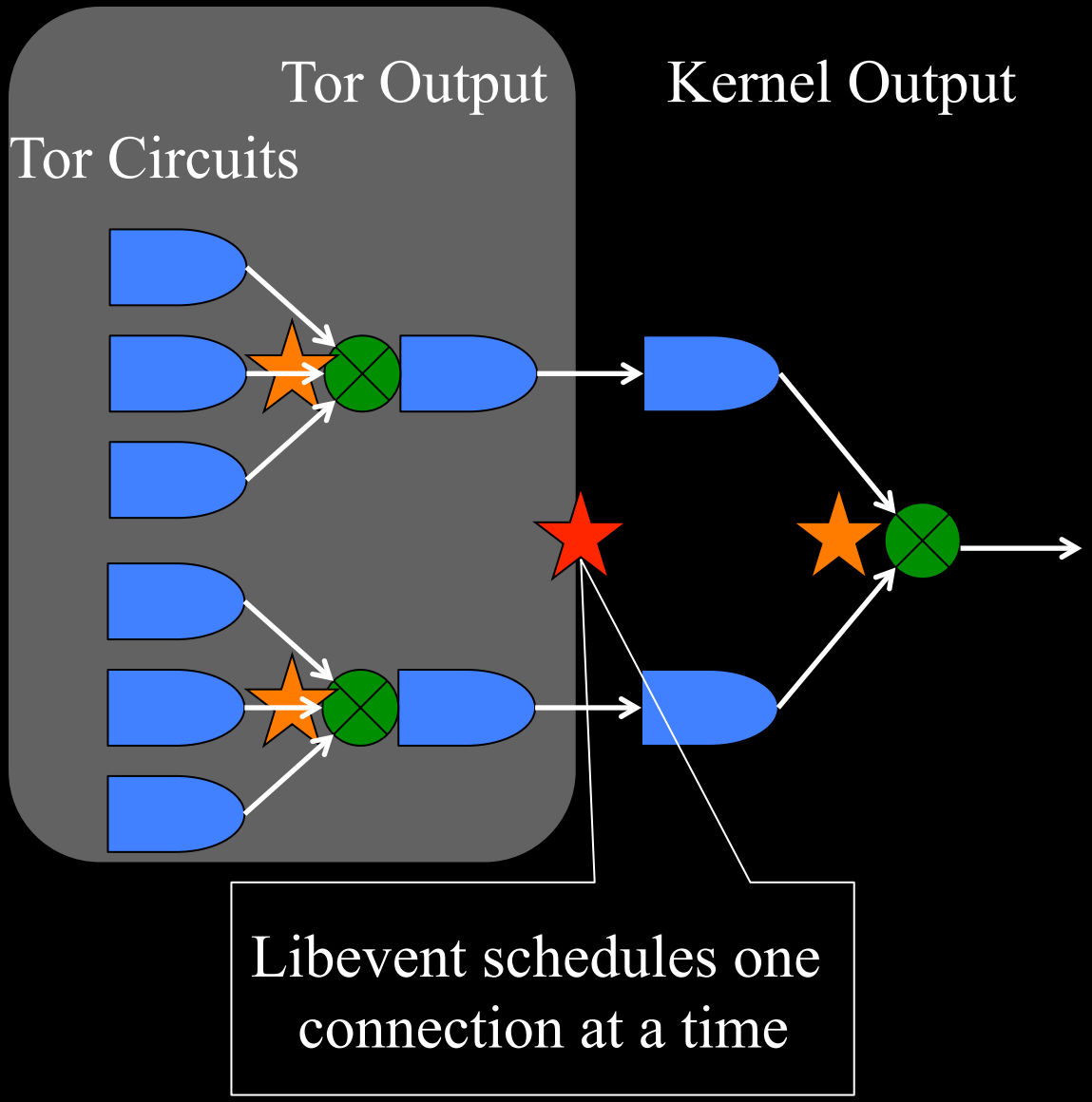
Analyzing Causes of Congestion

Problem 1:
Circuit scheduling

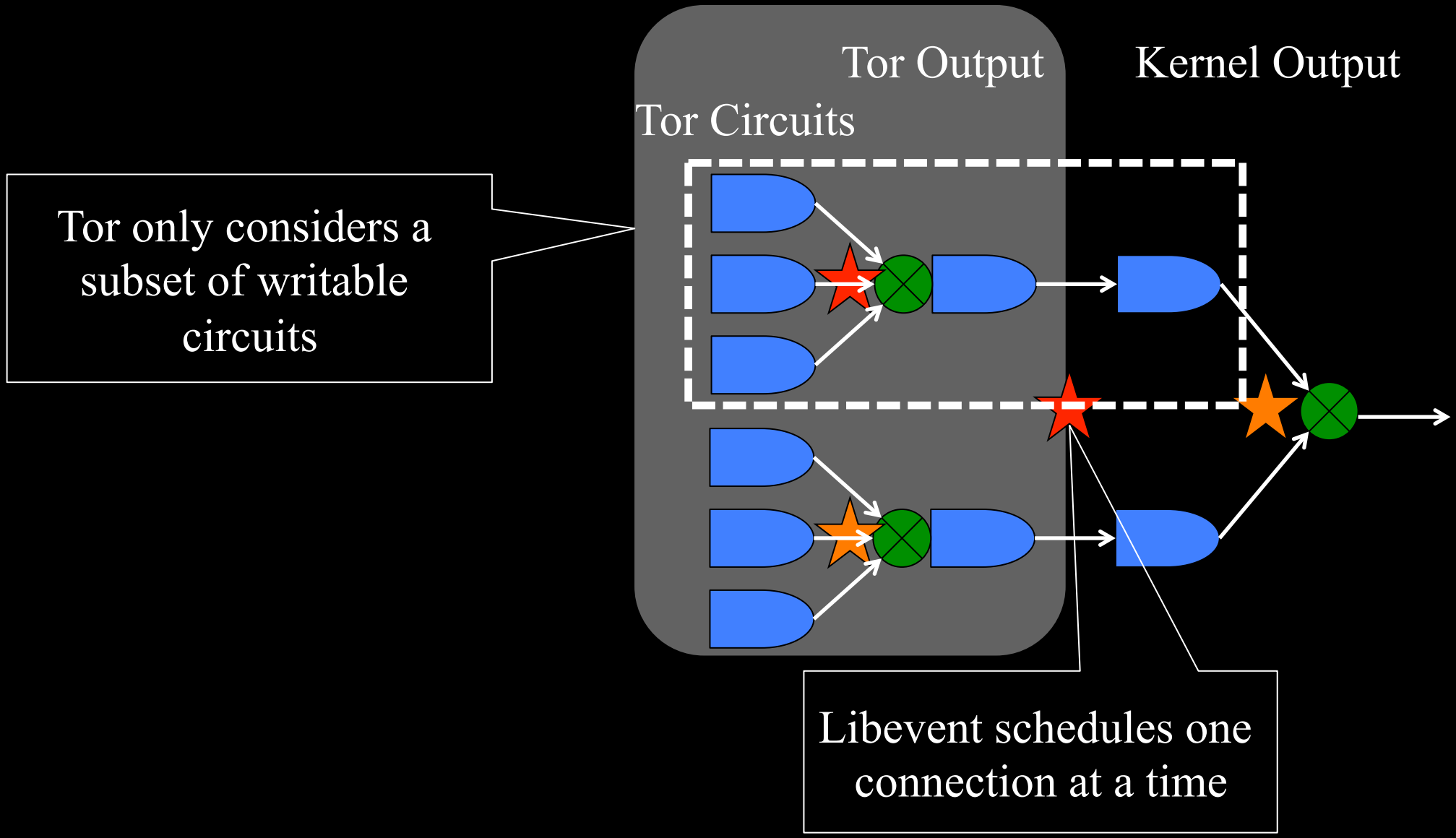
Problem 2:
Flushing to Sockets



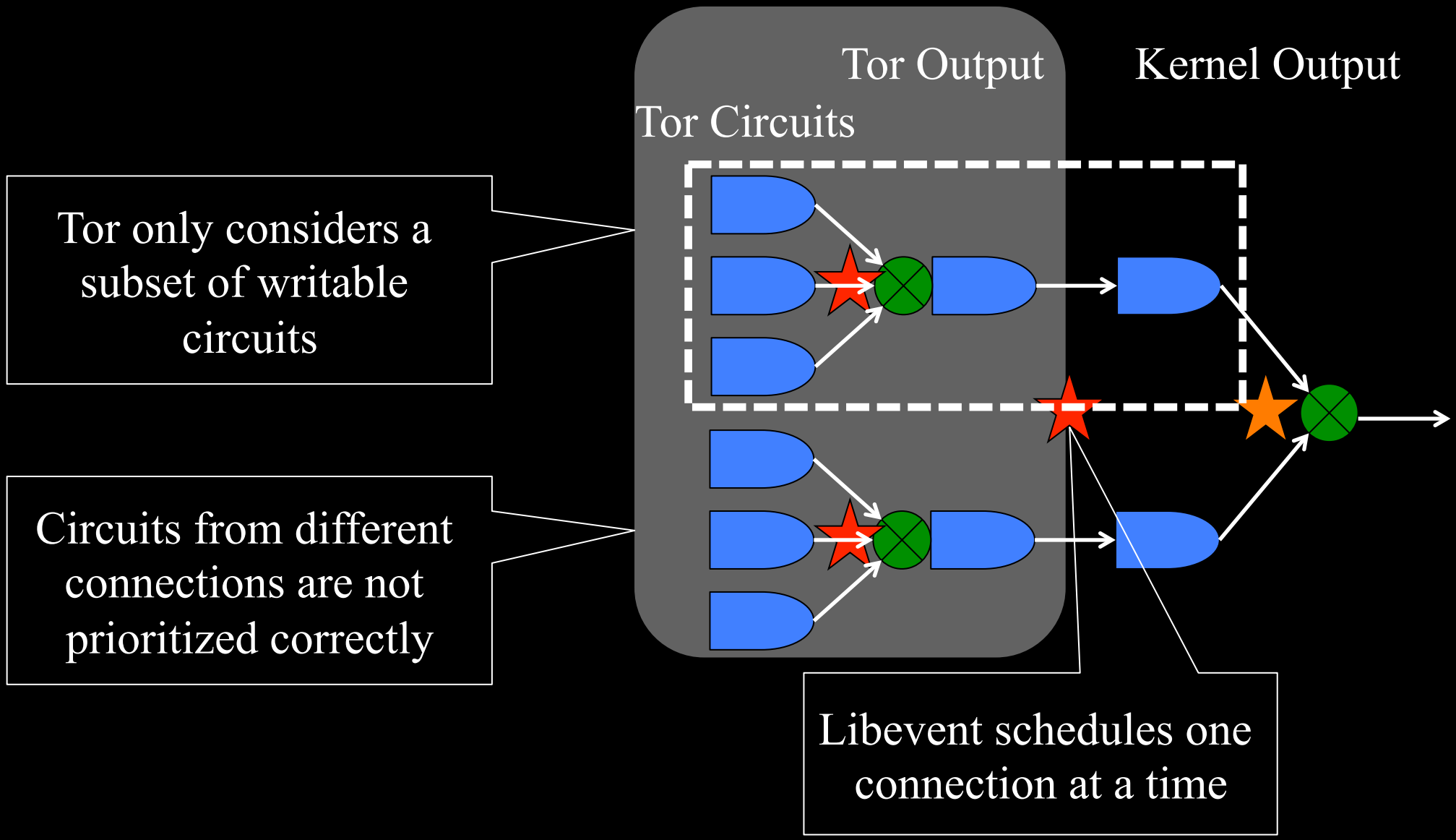
Problem 1: Circuit Scheduling



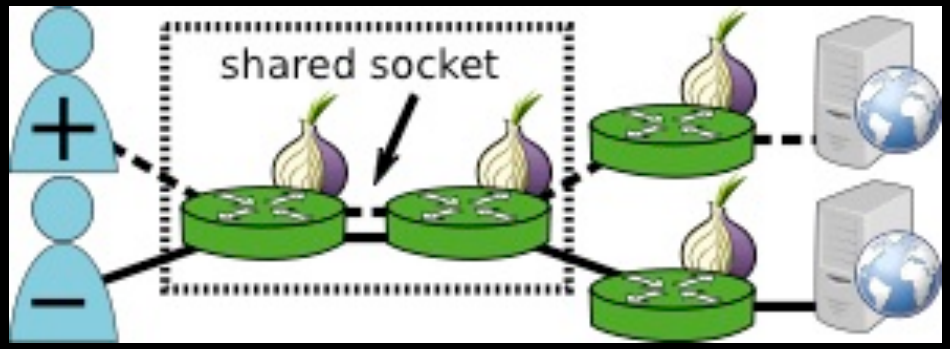
Problem 1: Circuit Scheduling

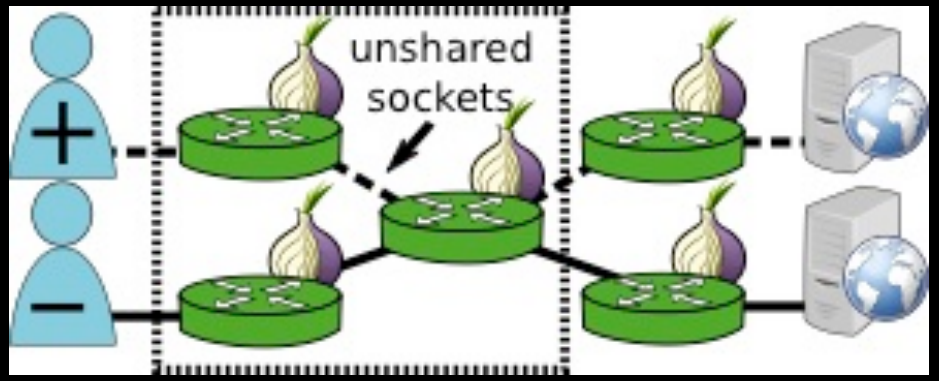


Problem 1: Circuit Scheduling

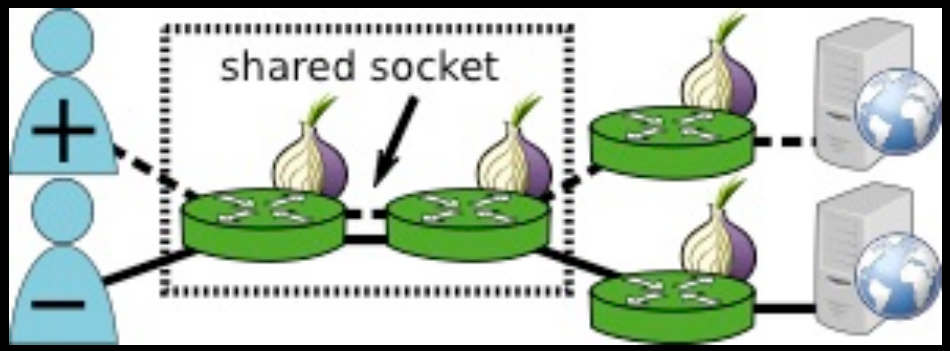


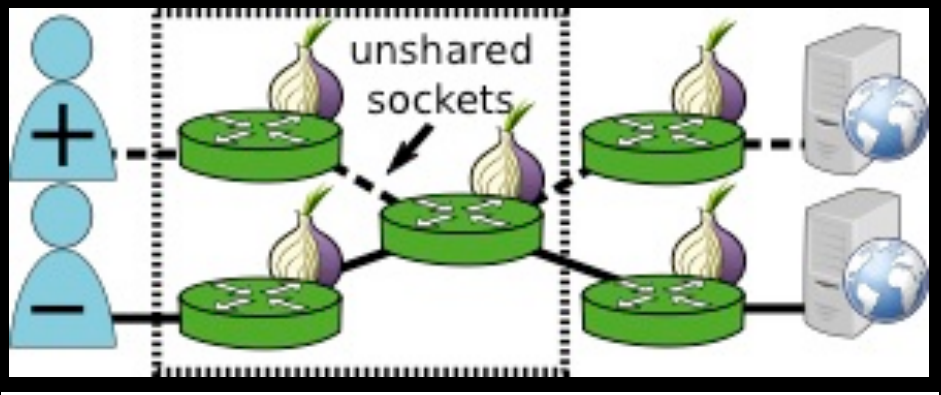
Problem 1: Circuit Scheduling

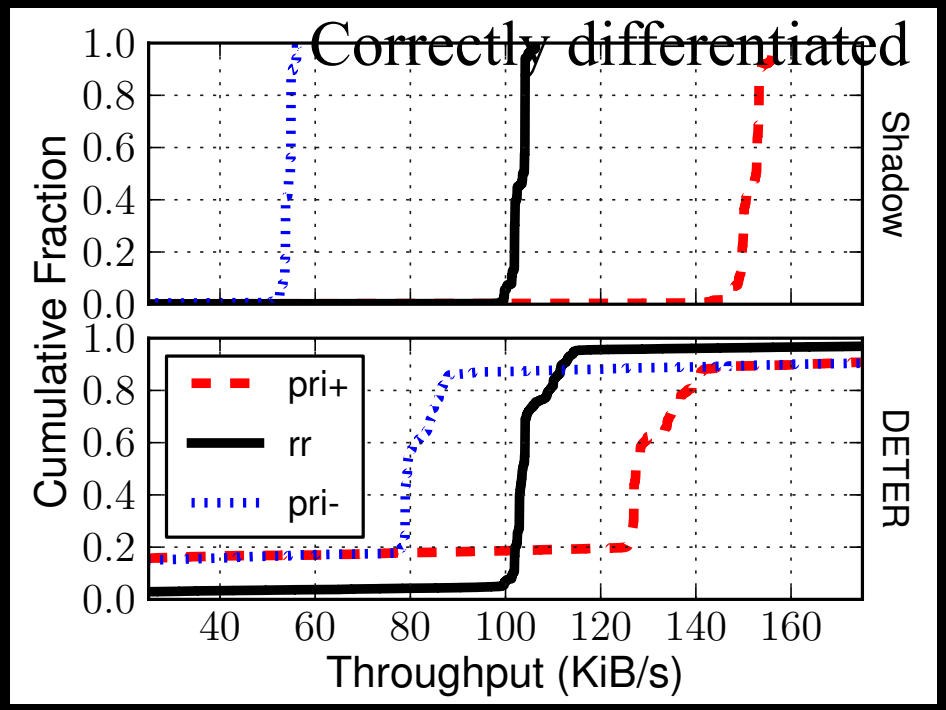


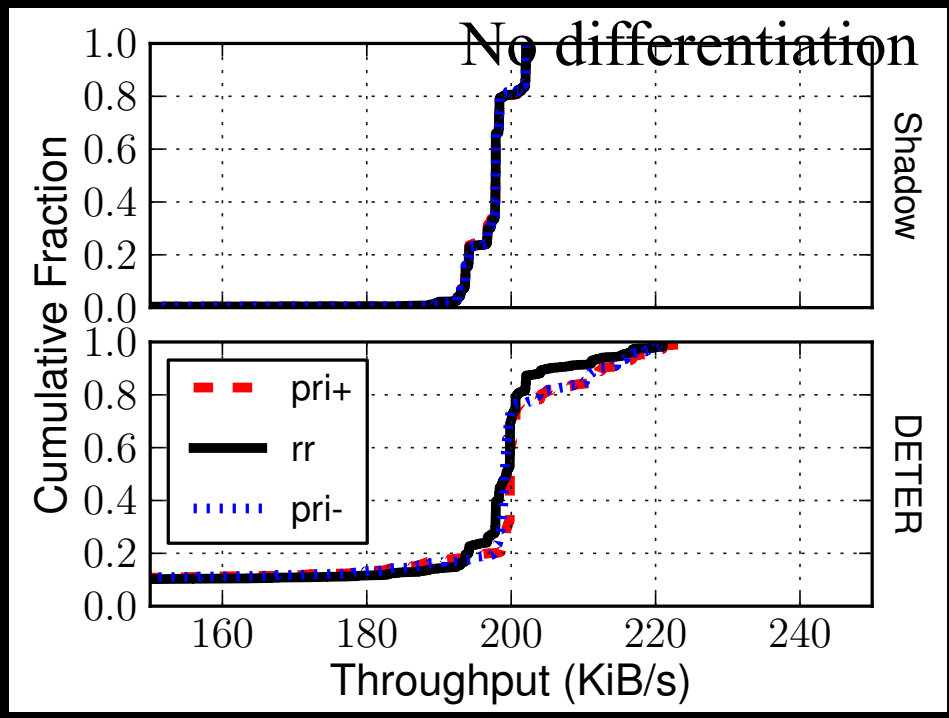


Problem 1: Circuit Scheduling

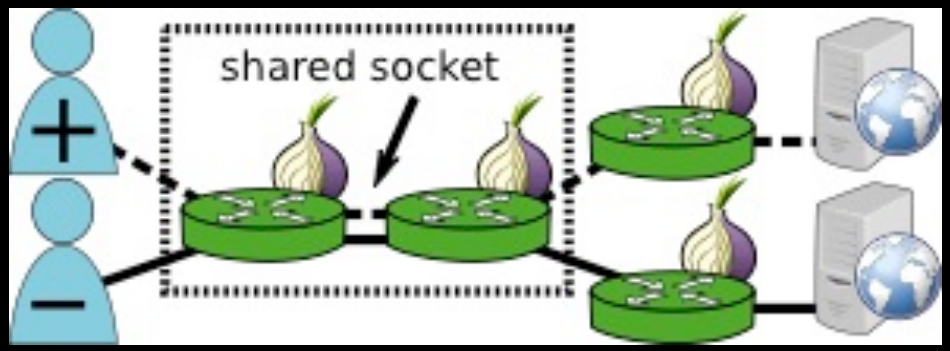


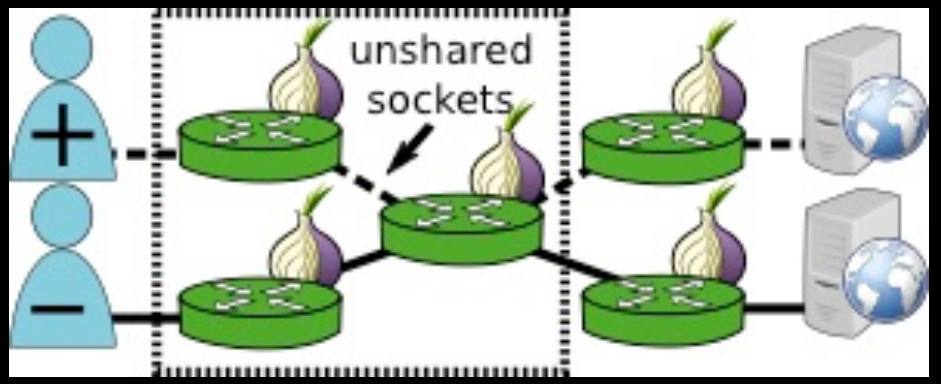


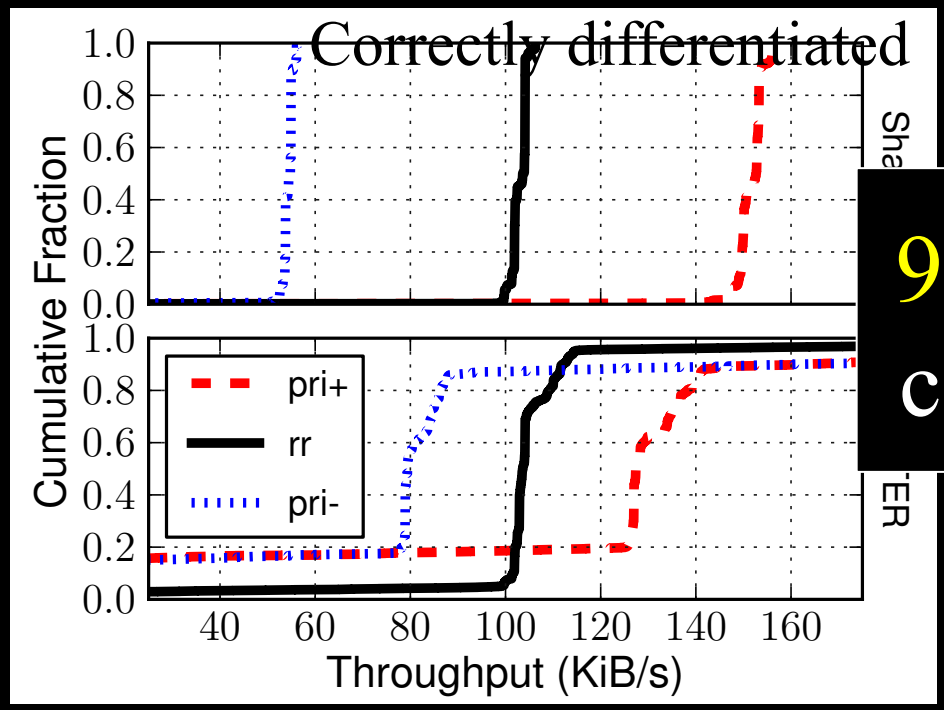


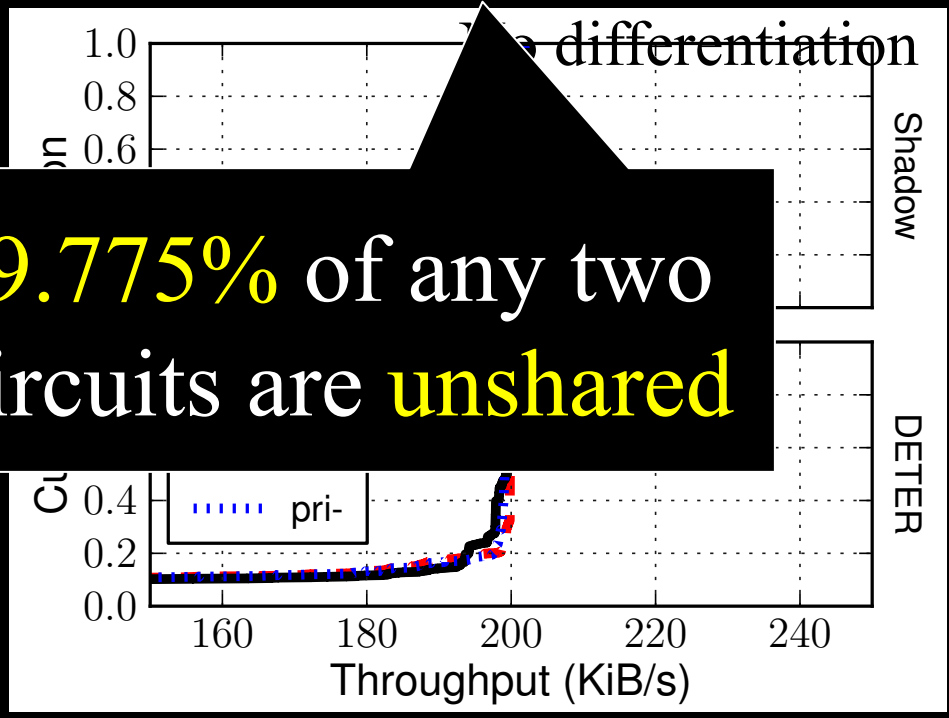


Problem 1: Circuit Scheduling



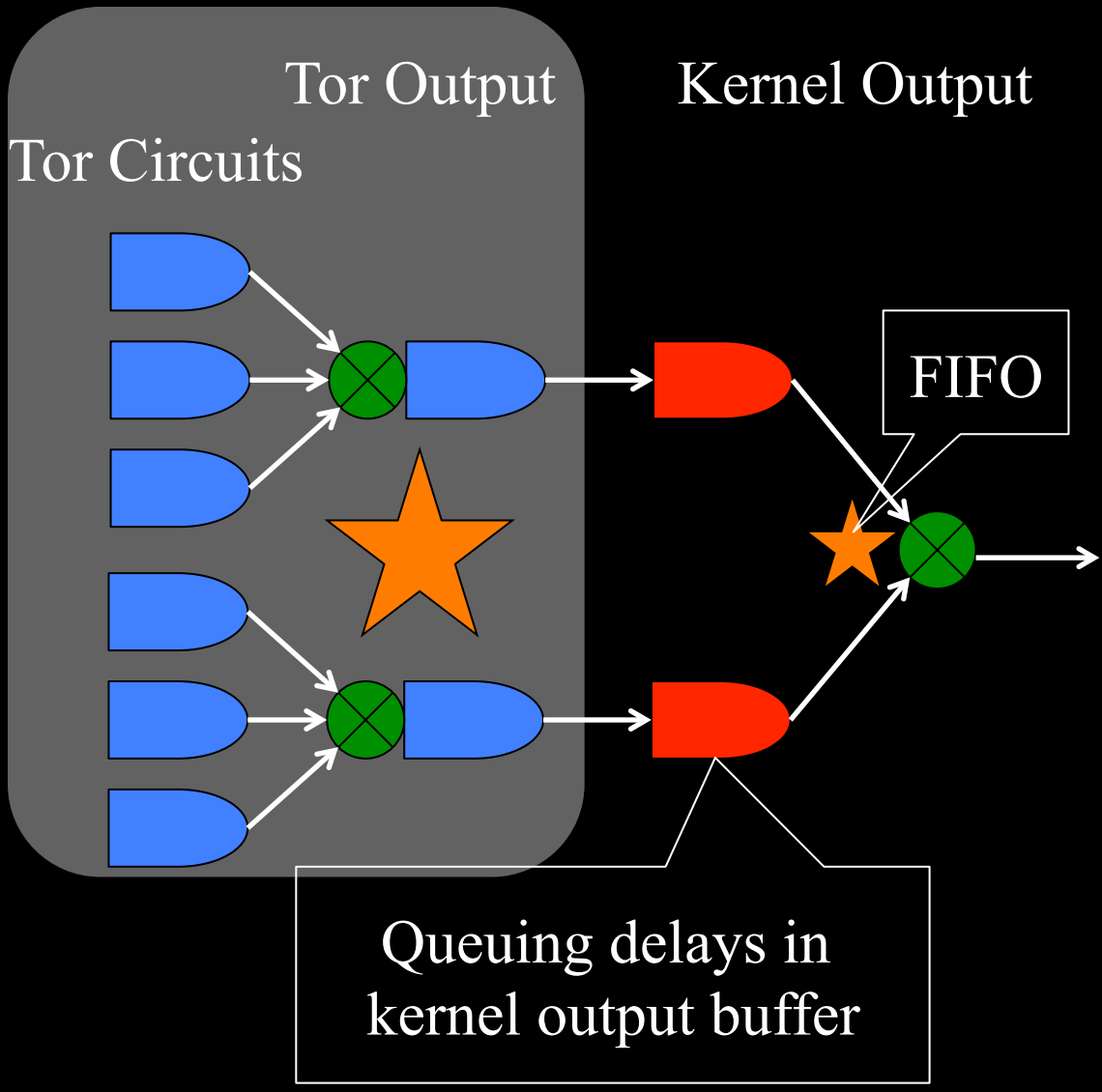




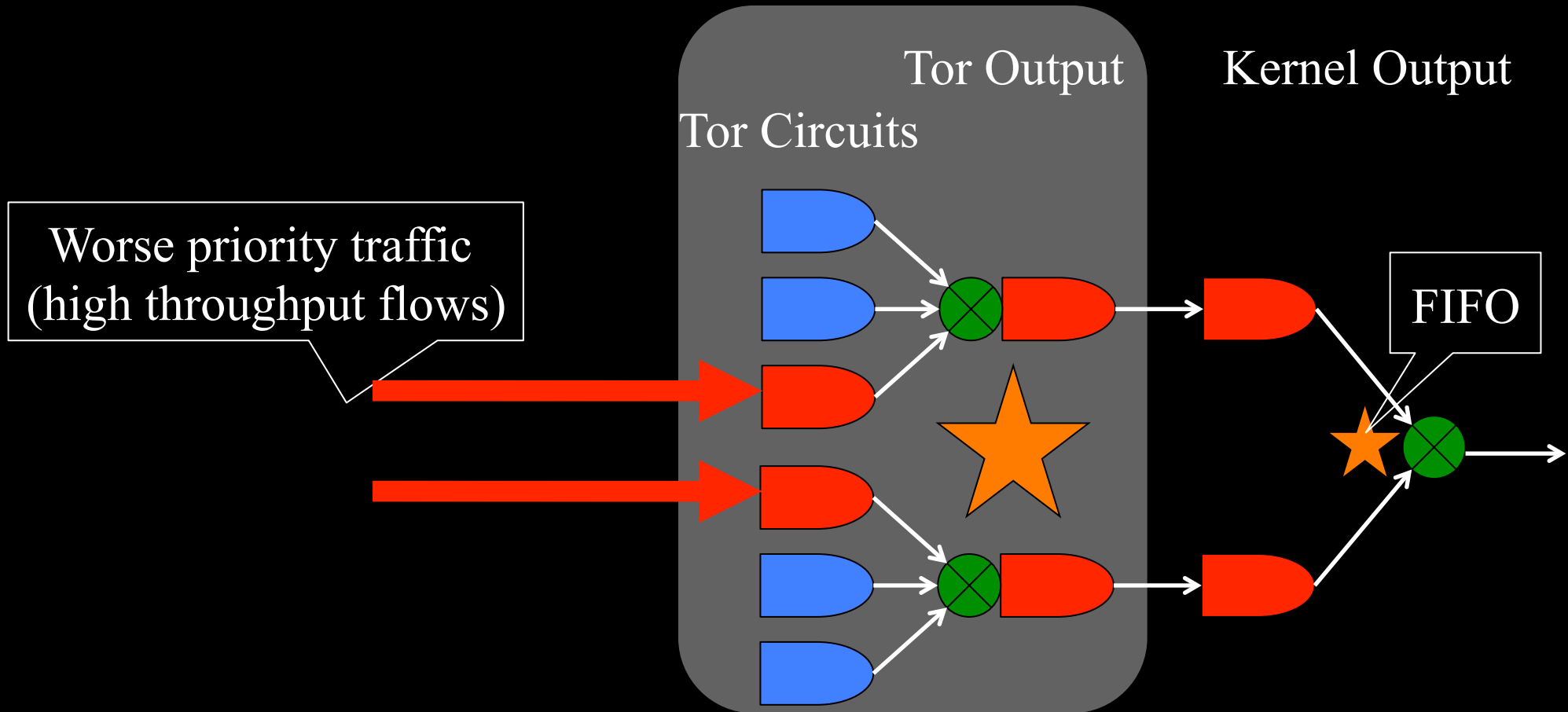


99.775% of any two circuits are unshared

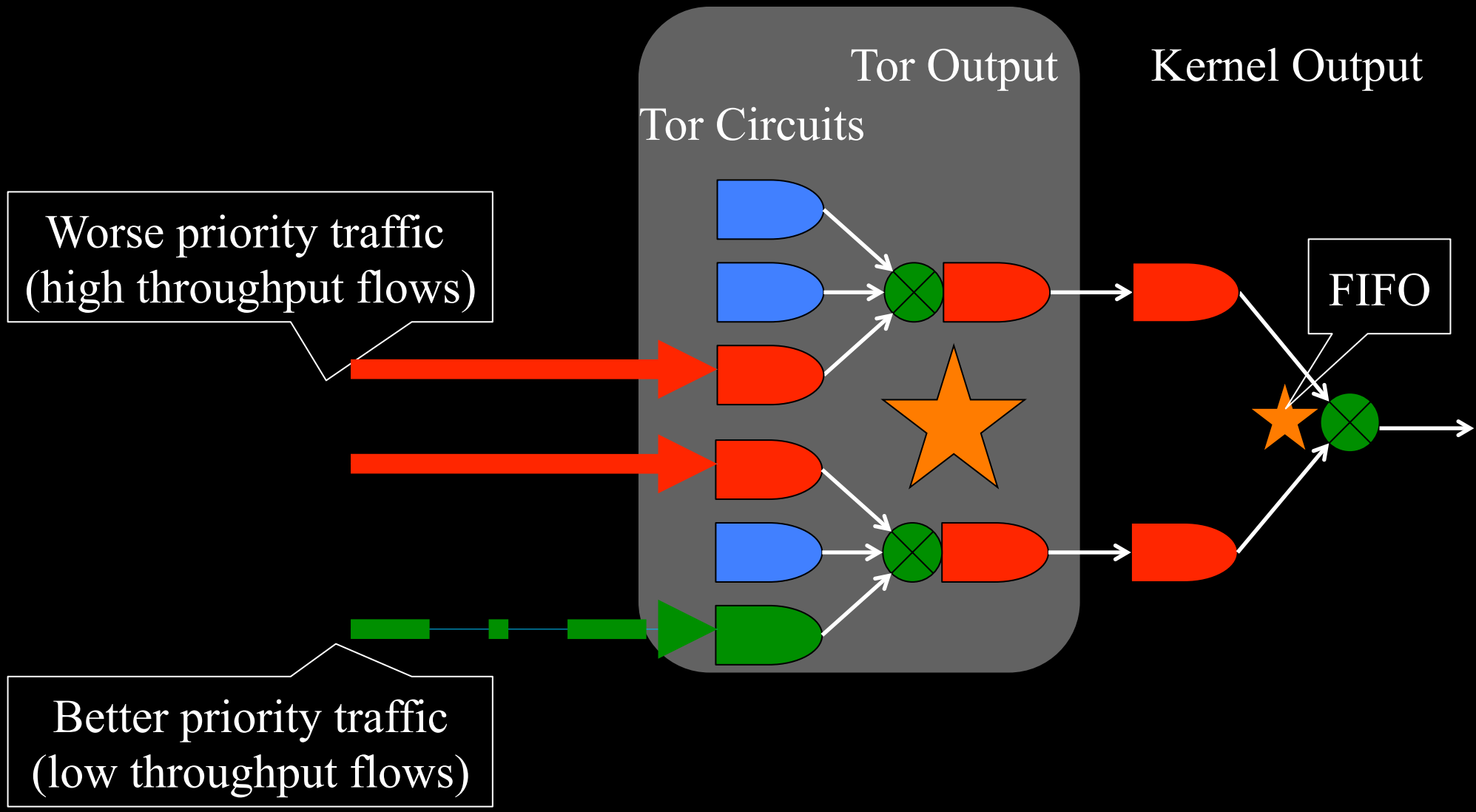
Problem 2: Flushing to Sockets



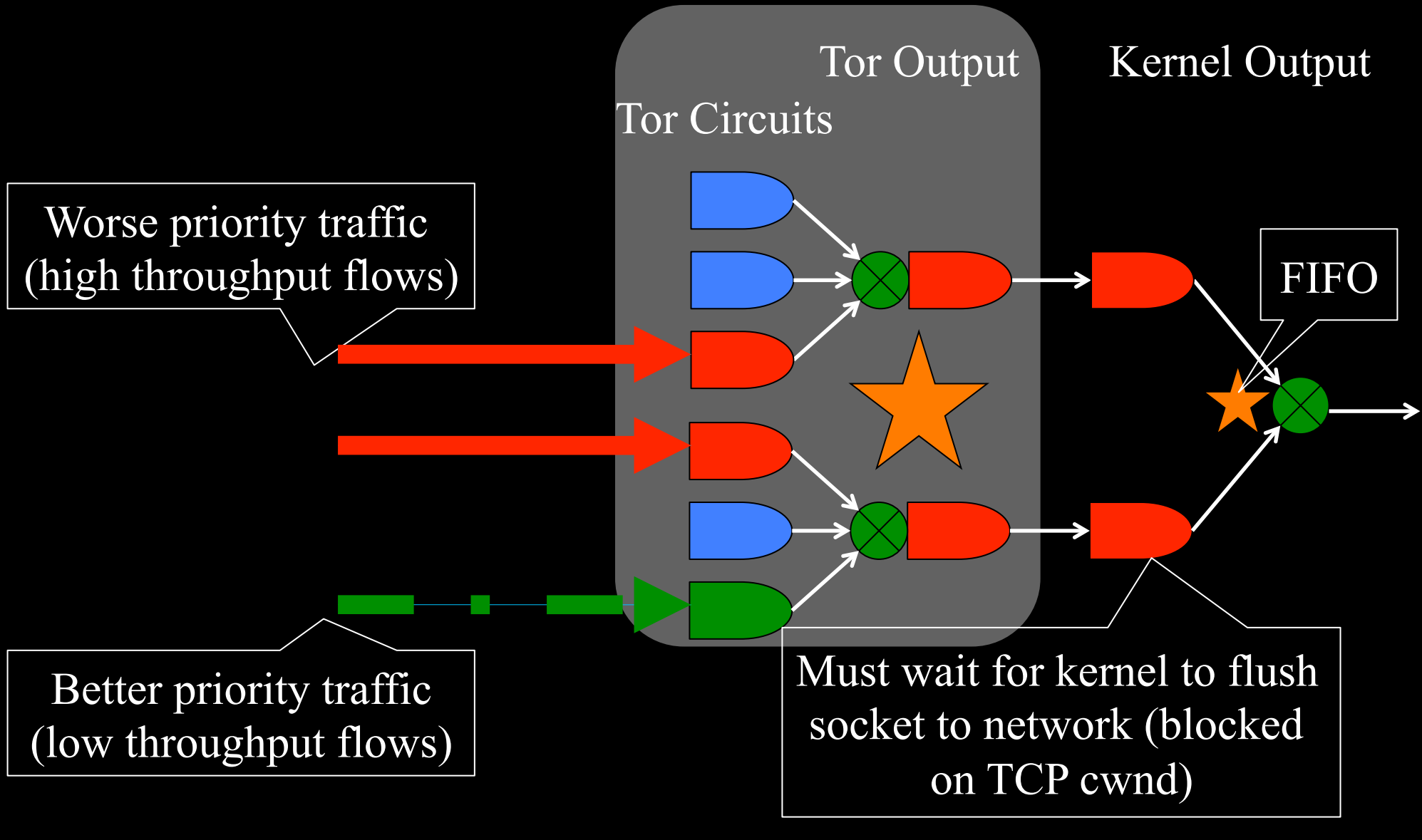
Problem 2: Flushing to Sockets



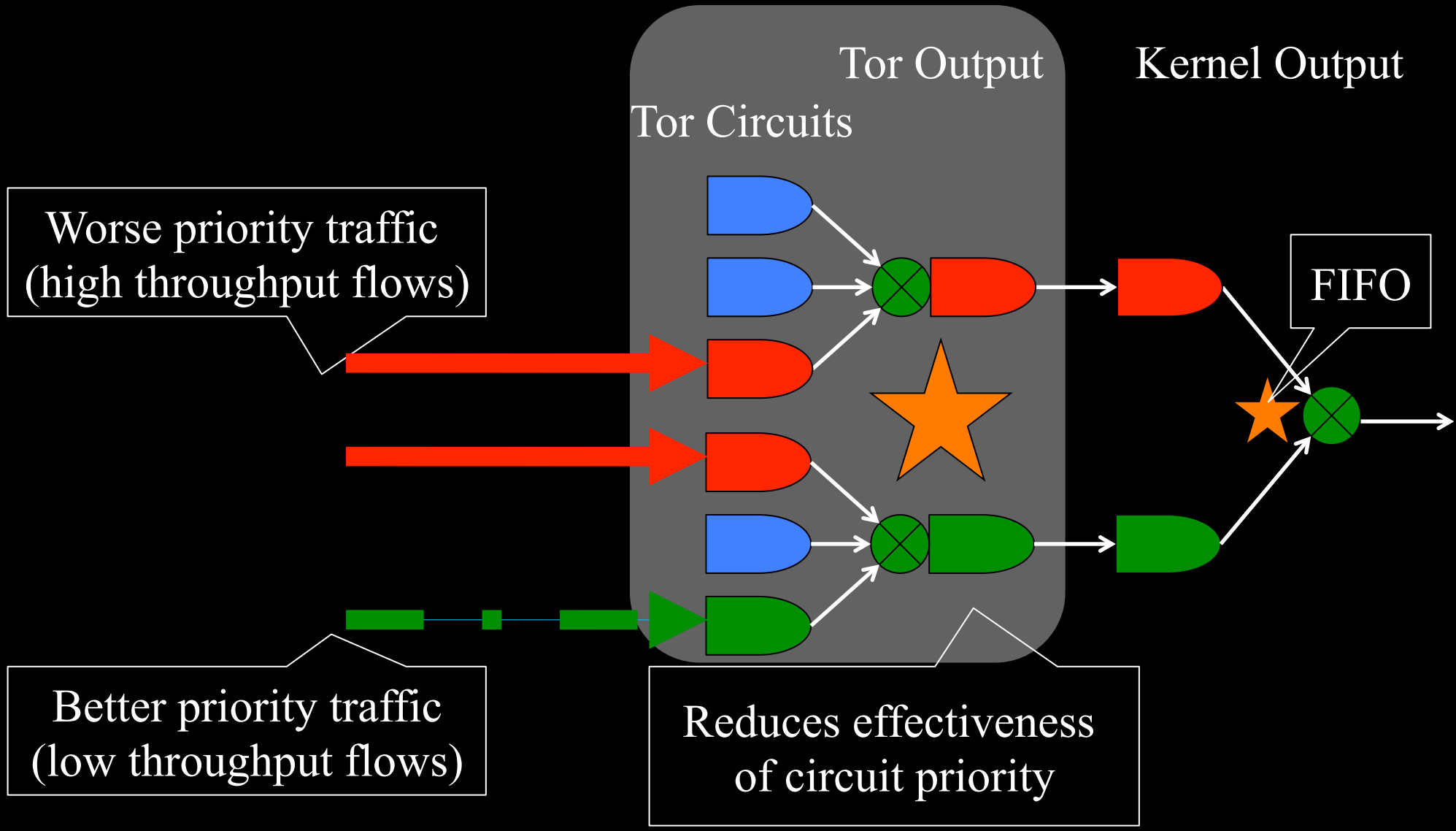
Problem 2: Flushing to Sockets



Problem 2: Flushing to Sockets



Problem 2: Flushing to Sockets



Outline

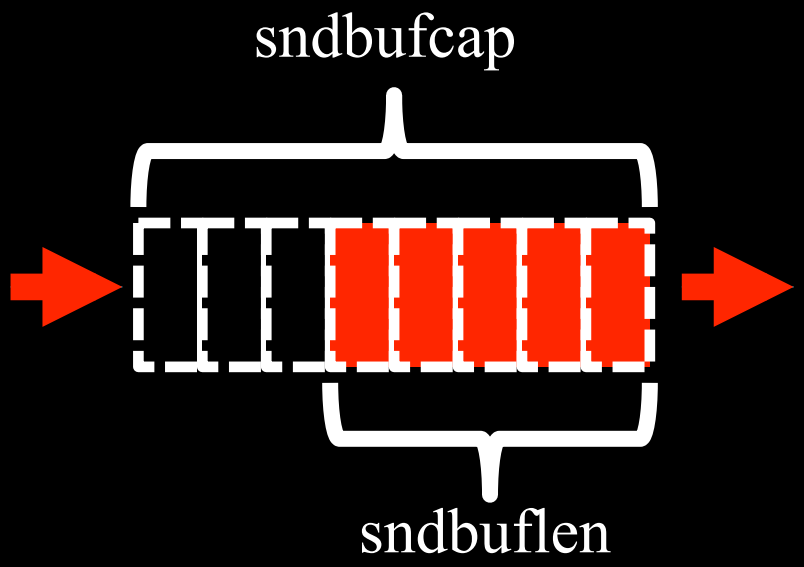
- ~~Background~~
- ~~Instrument Tor, measure congestion~~
- ~~Analyze causes of congestion~~
- Design and evaluate KIST
  - Performance
  - Security

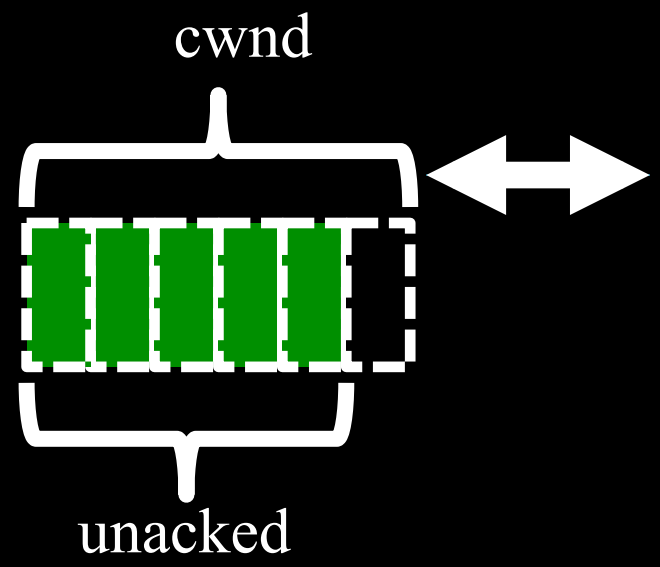
Ask the kernel, stupid!

- Utilize getsockopt and ioctl syscalls

socket_space =
sndbufcap – sndbuflen

tcp_space =
(cwnd – unacked) * mss





Kernel-Informed Socket Transport

- Don’t write it if the kernel can’t send it; bound kernel writes by:
  - Socket: min(socket_space, tcp_space)
  - Global: upstream bandwidth capacity

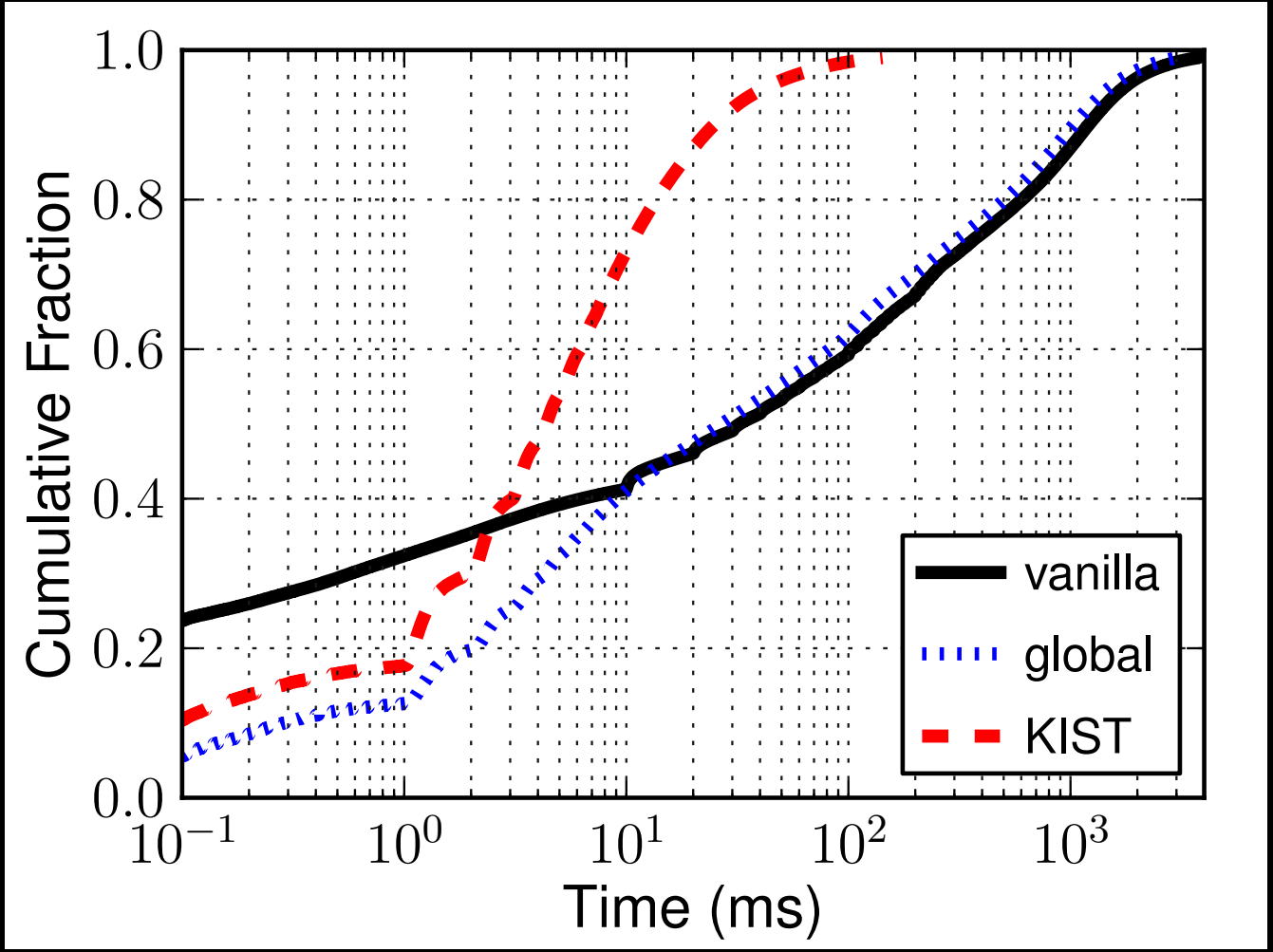
Kernel-Informed Socket Transport

- Don’t write it if the kernel can’t send it; bound kernel writes by:
  - Socket: min(socket_space, tcp_space)
  - Global: upstream bandwidth capacity
- Choose globally from all writable circuits

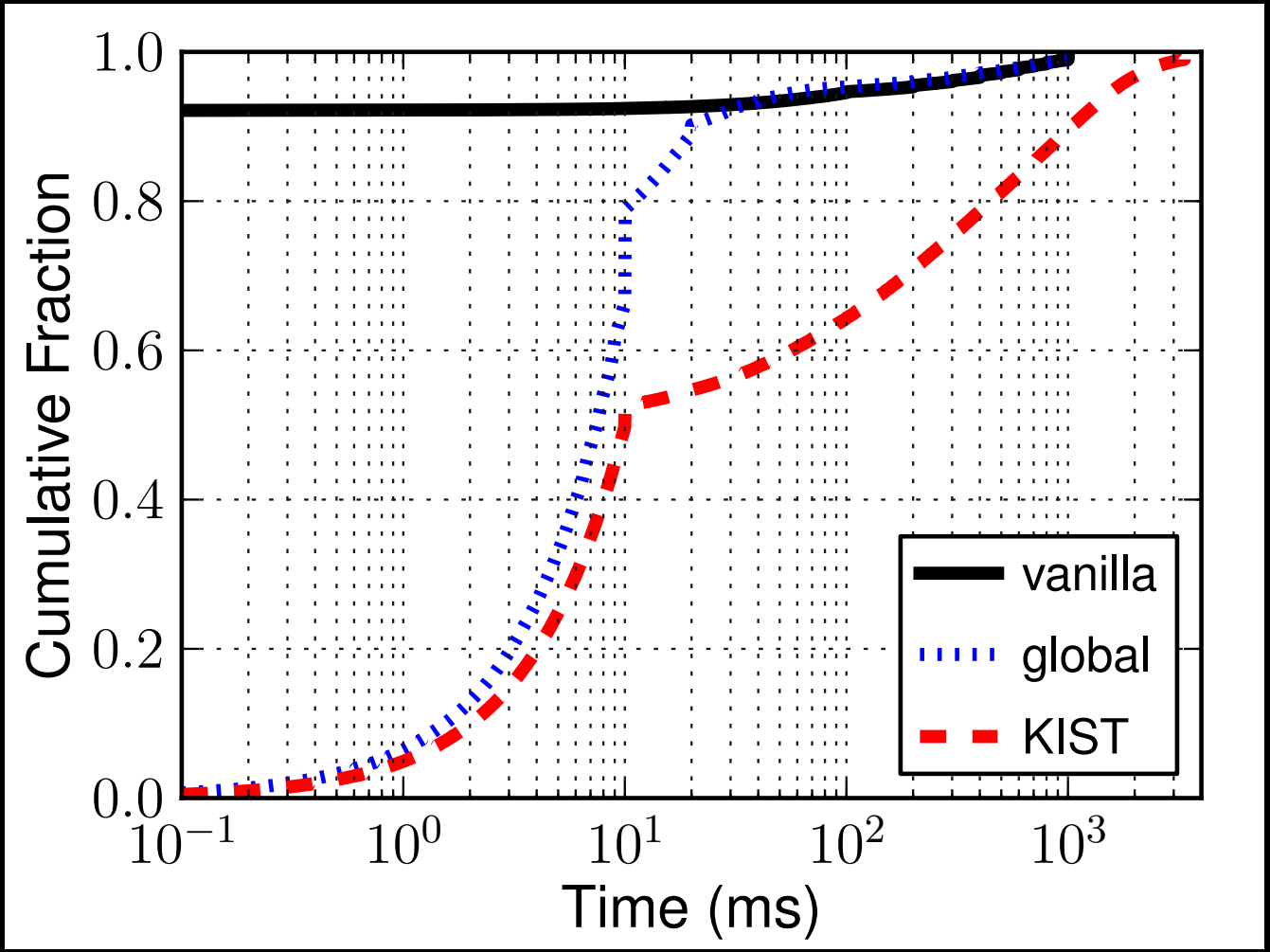
Kernel-Informed Socket Transport

- Don’t write it if the kernel can’t send it; bound kernel writes by:
  - Socket: min(socket_space, tcp_space)
  - Global: upstream bandwidth capacity
- Choose globally from all writable circuits
- Try to write again before kernel starvation

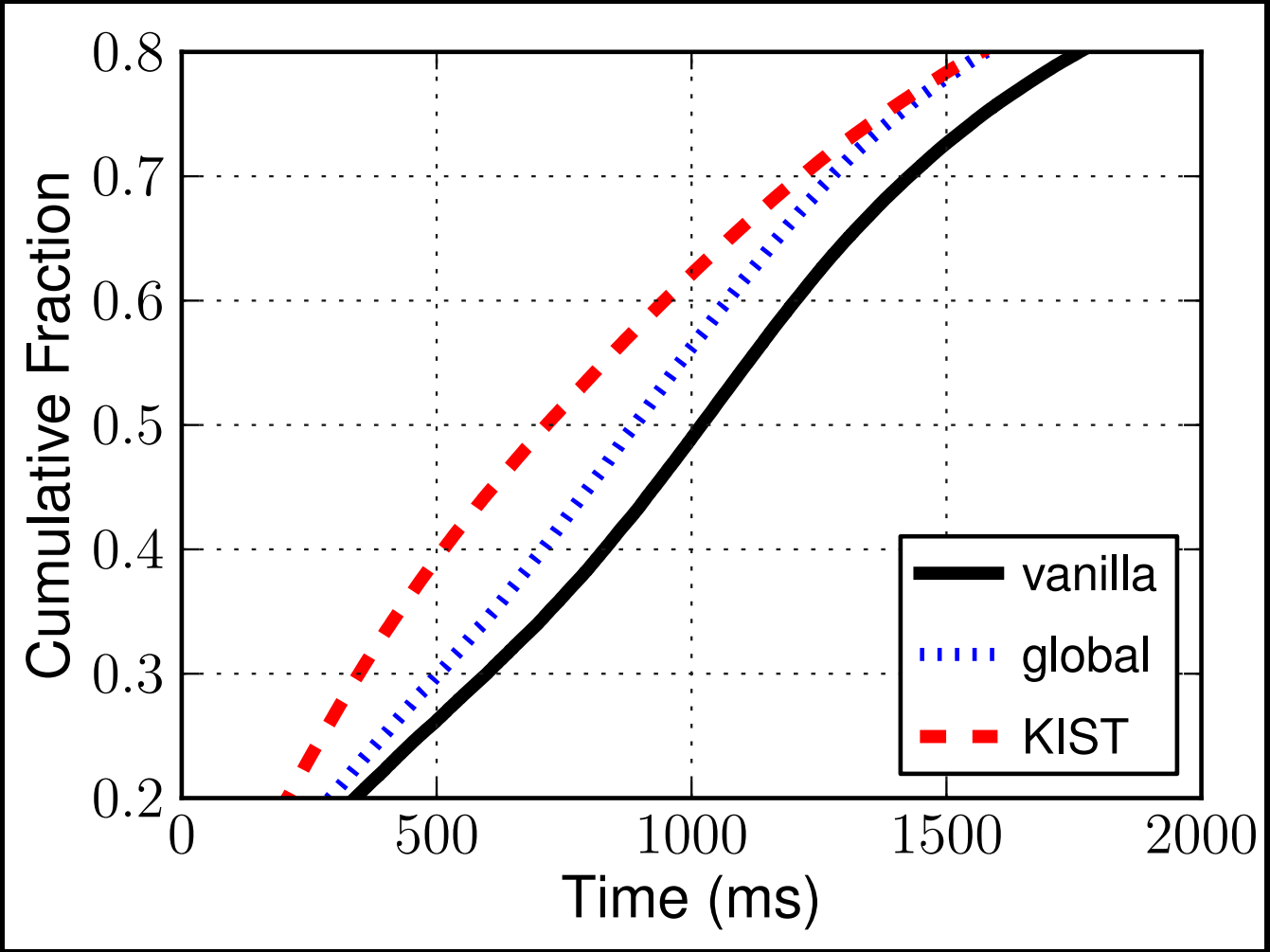
KIST Reduces Kernel Congestion



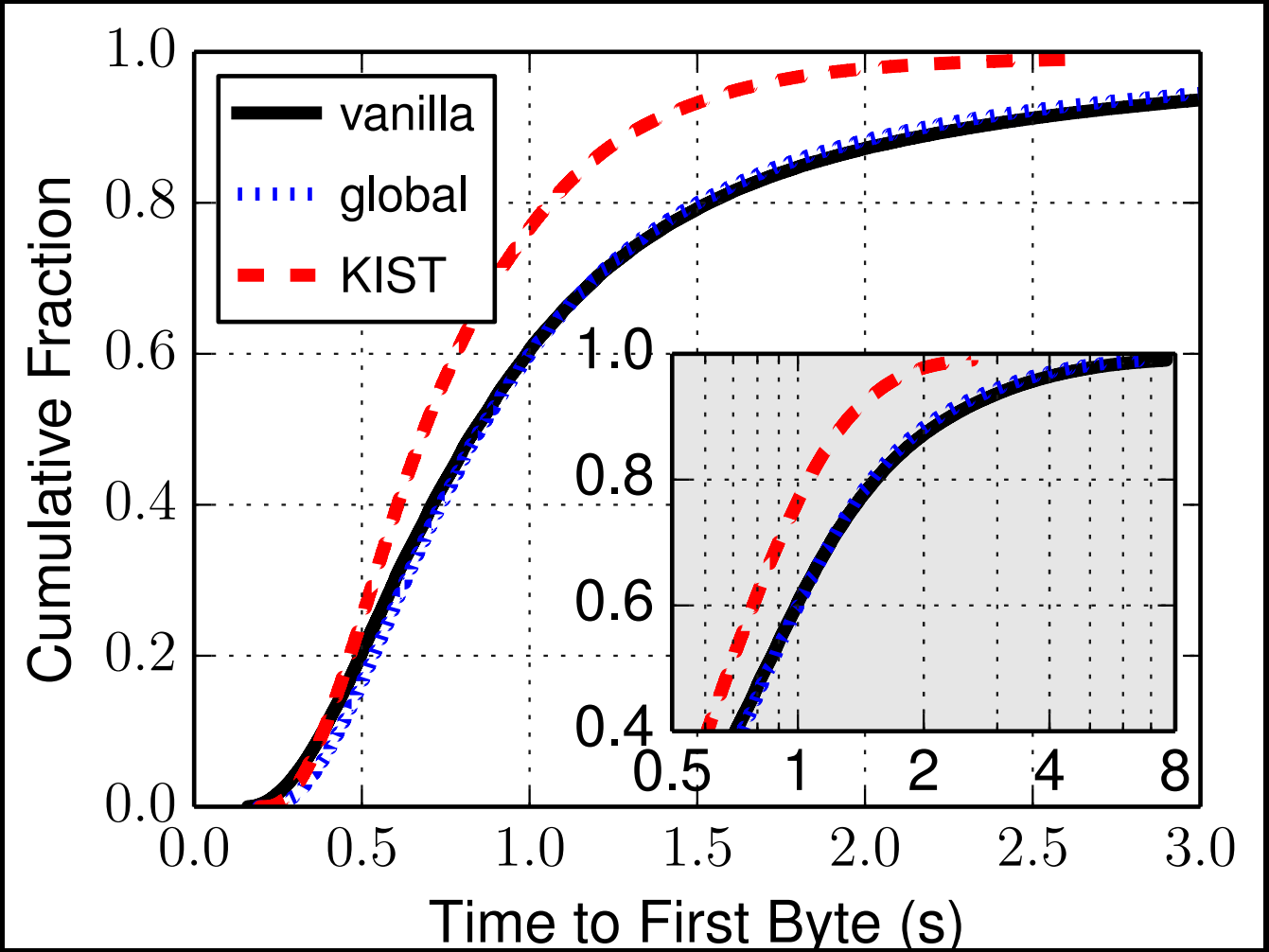
KIST Increases Tor Congestion



KIST Reduces Circuit Congestion



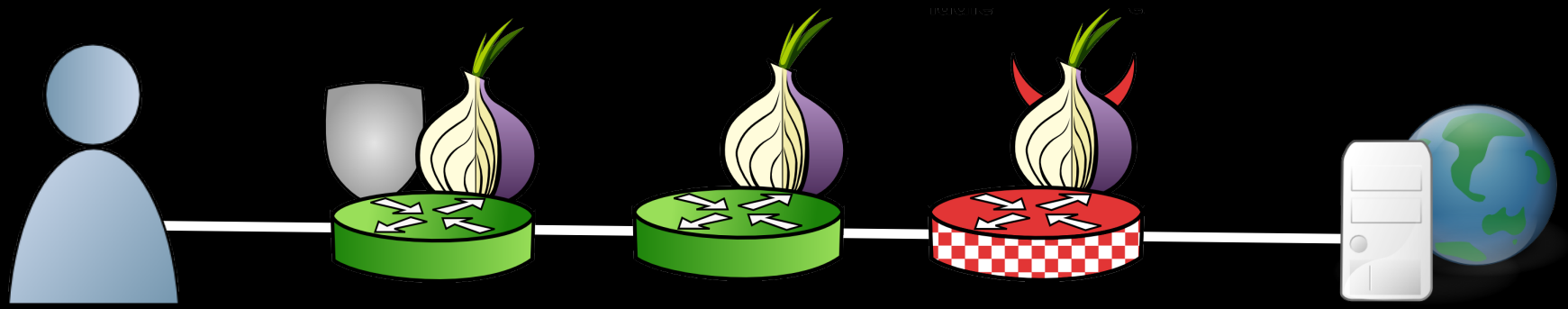
KIST Improves Network Latency



Outline

- ~~Background~~
- ~~Instrument Tor, measure congestion~~
- ~~Analyze causes of congestion~~
- Design and evaluate KIST
  - ~~Performance~~
  - Security

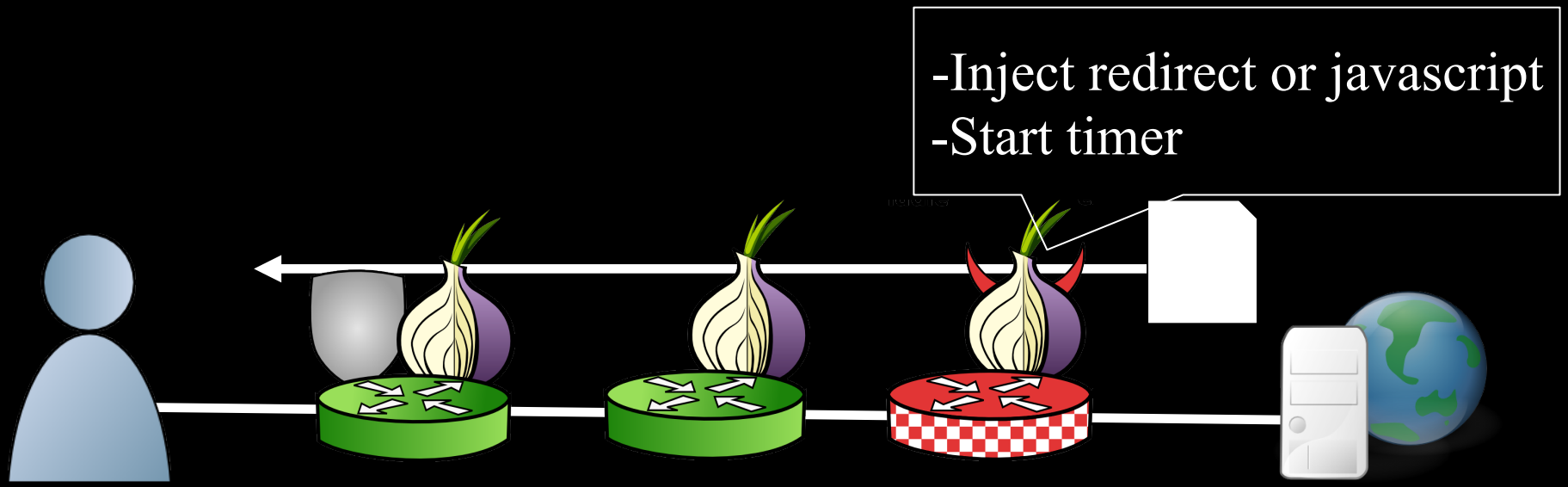
Traffic Correlation: Latency



Goal: narrow down potential locations of the client on a target circuit

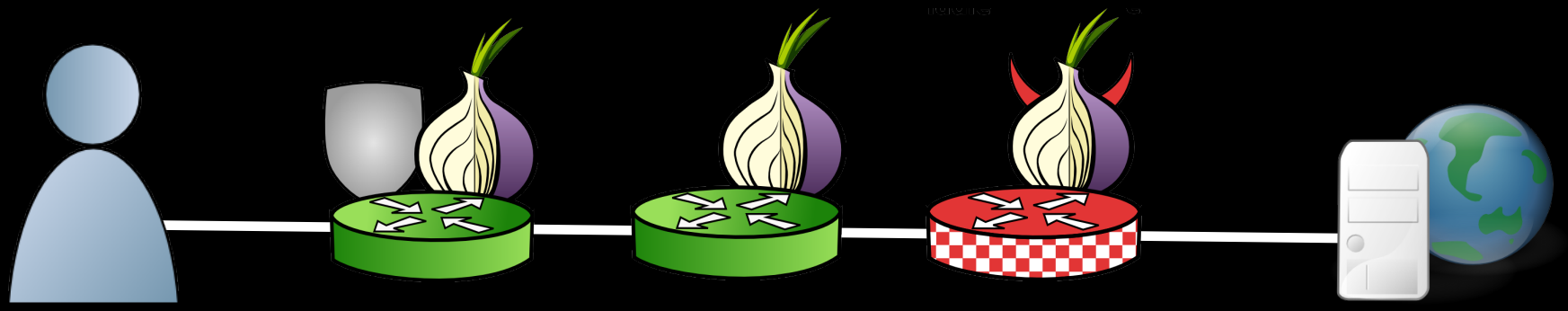
Hopper et.al. CCS'07

Traffic Correlation: Latency



Hopper et.al. CCS'07

Traffic Correlation: Latency

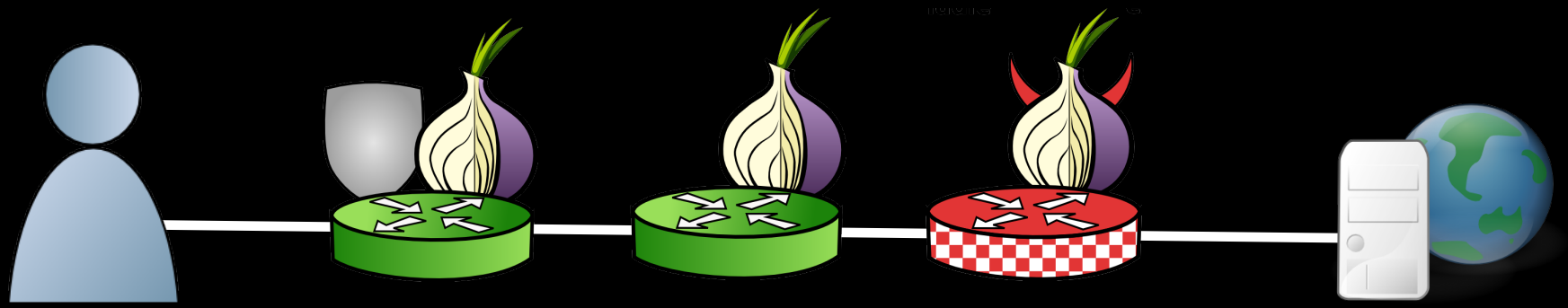


GET

Request redirected page or embedded object

Hopper et.al. CCS'07

Traffic Correlation: Latency

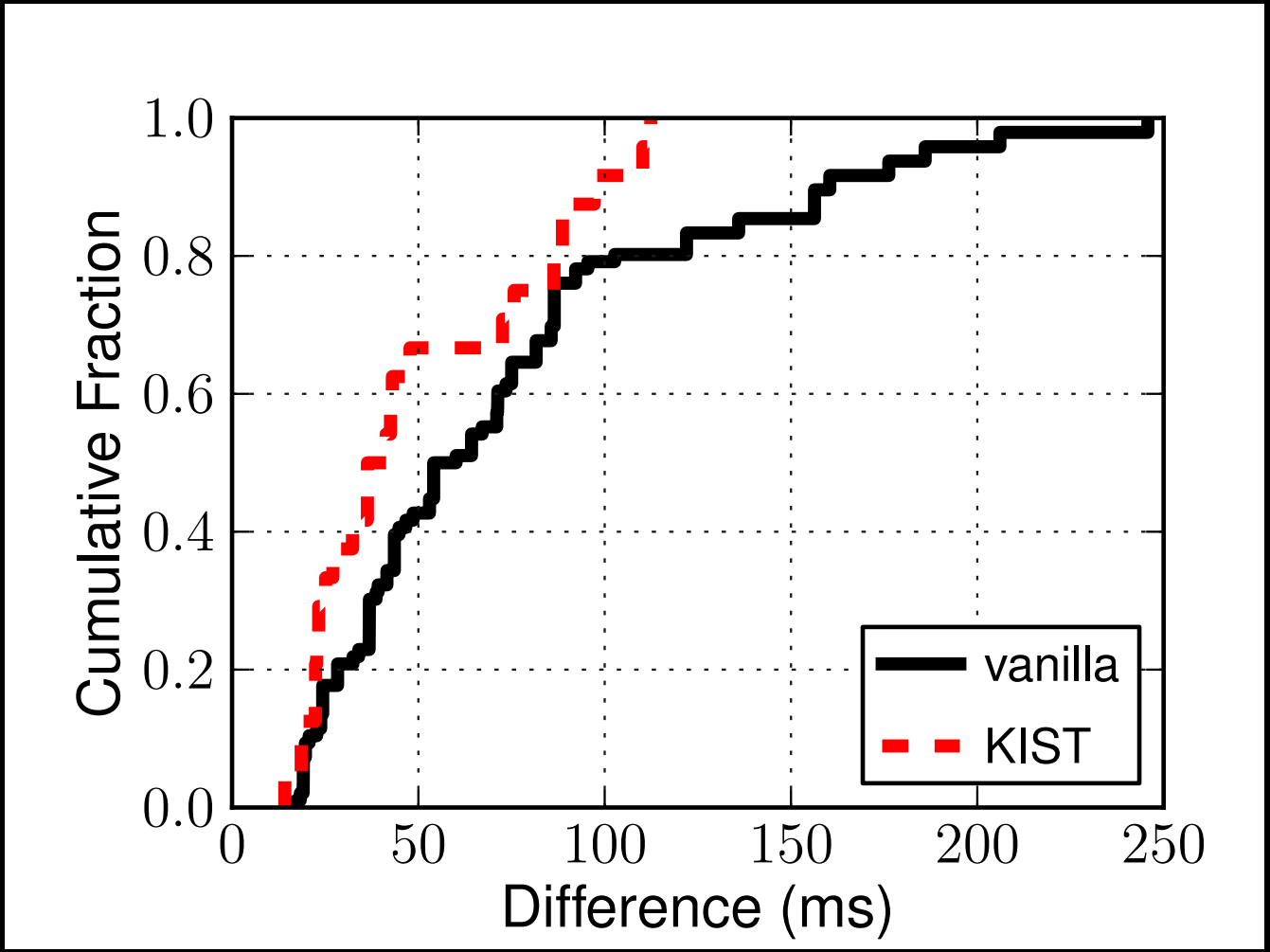


GET →

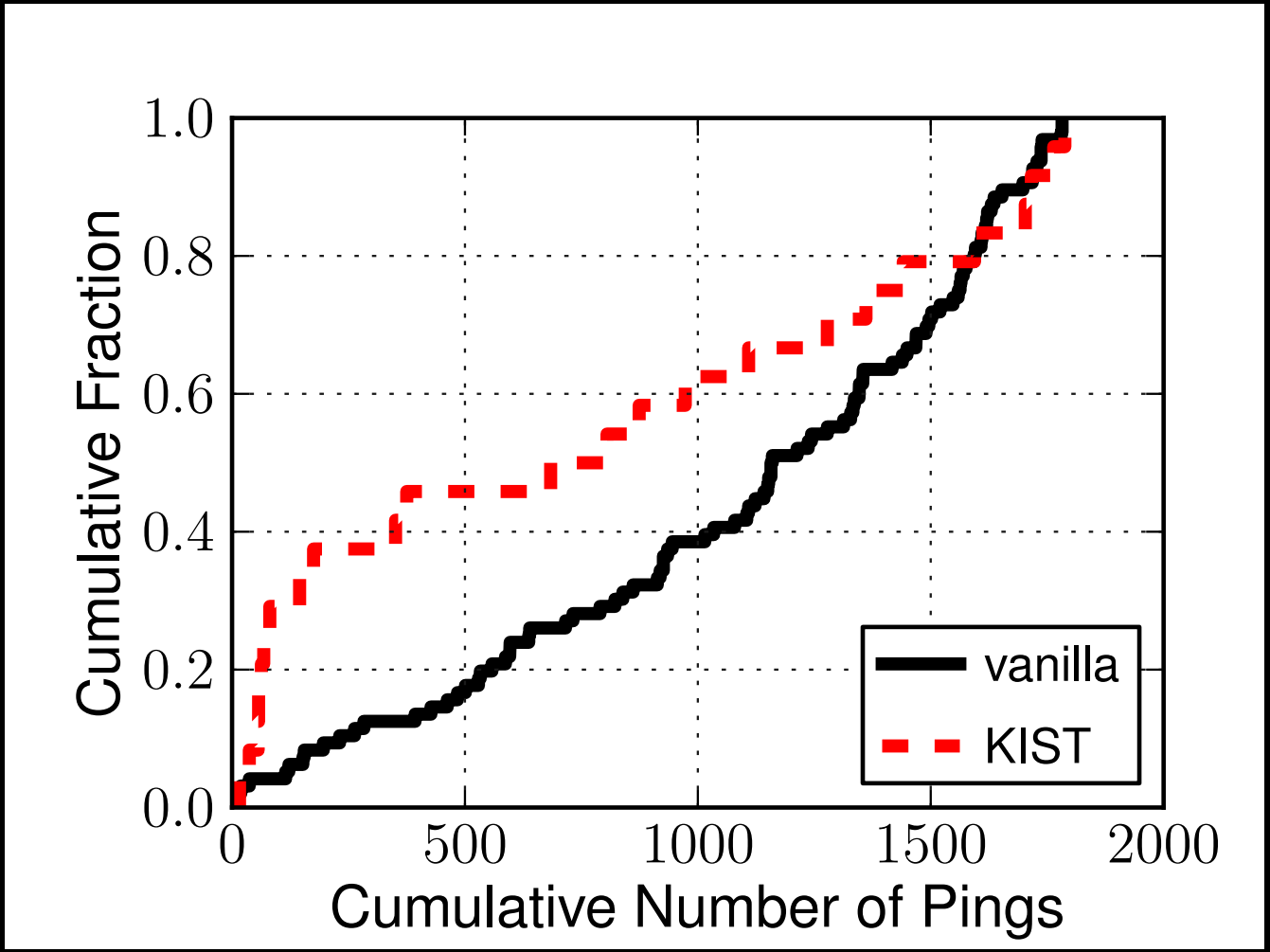
-Stop timer
-Estimate latency

Hopper et.al. CCS'07

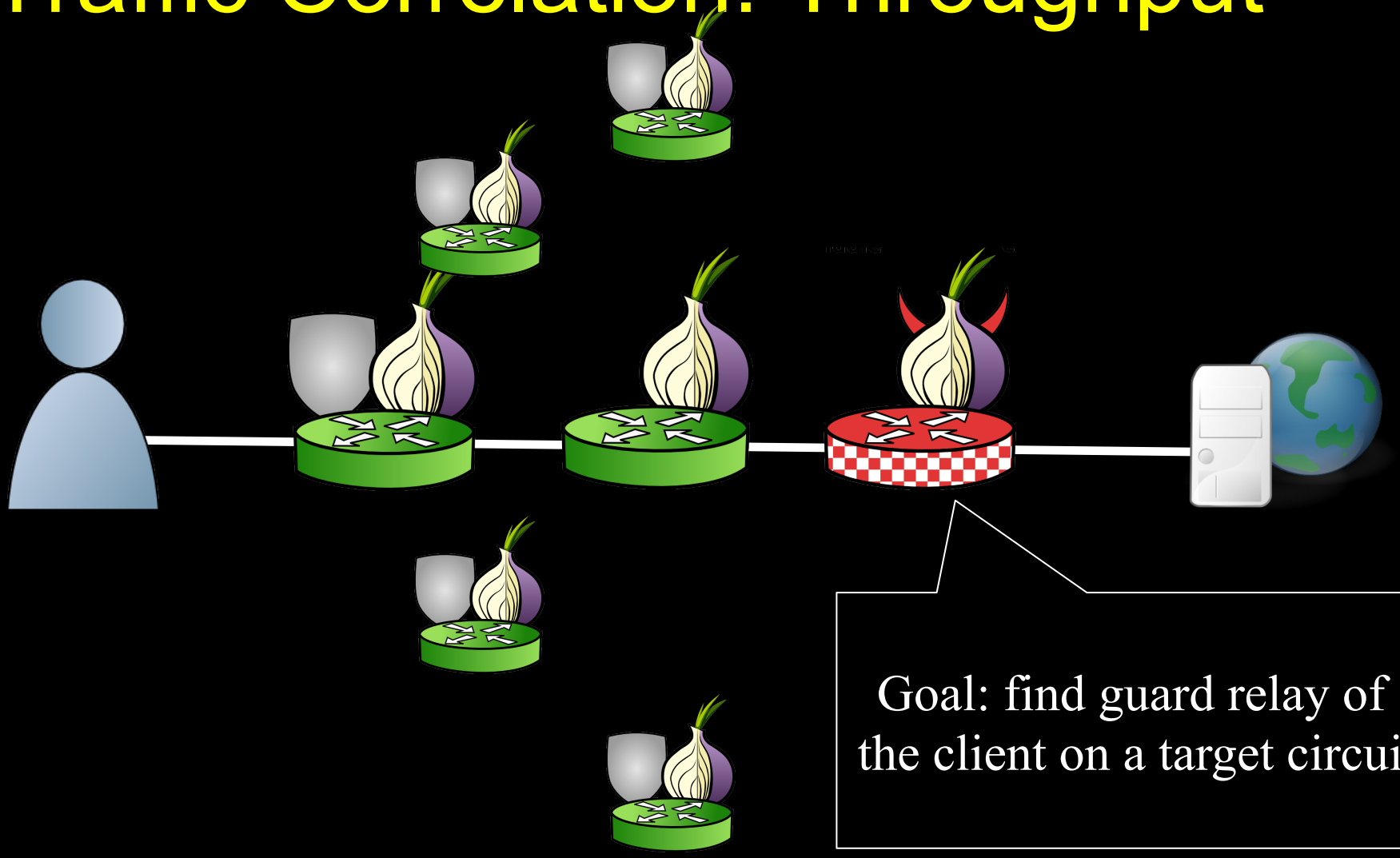
Latency Attack
| estimate – actual |



Latency Attack
num pings until best estimate

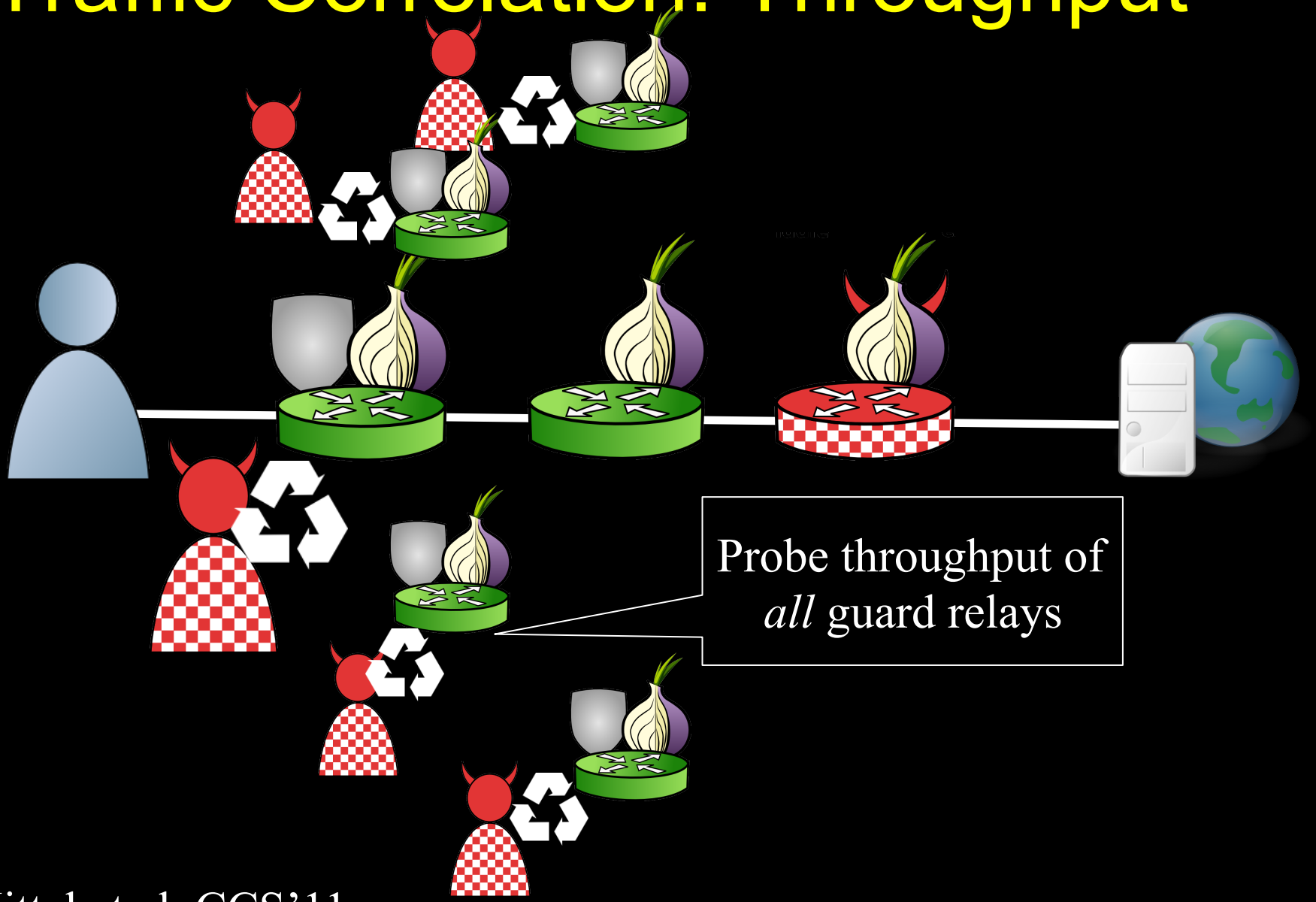


Traffic Correlation: Throughput



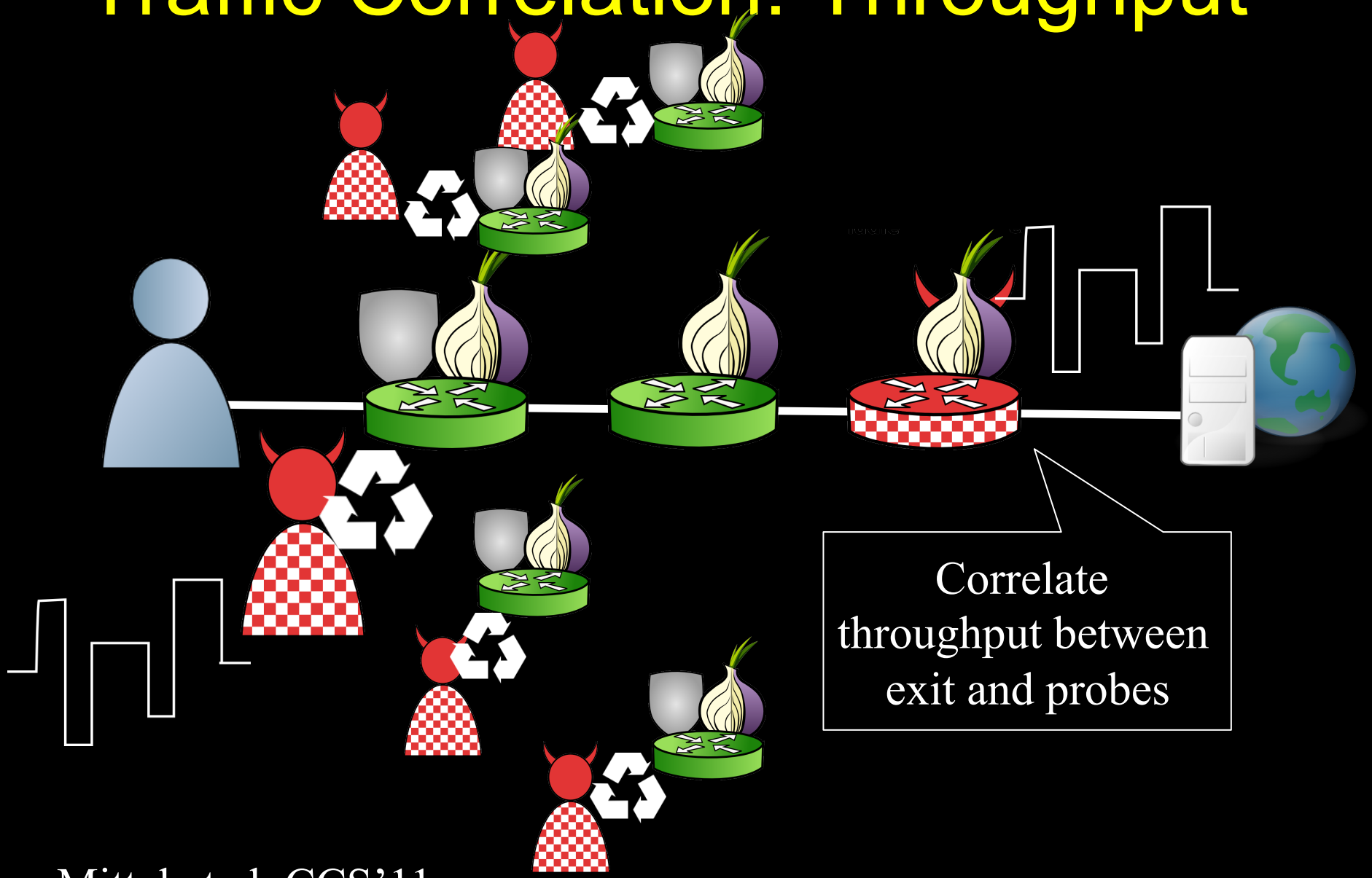
Mittal et.al. CCS'11

Traffic Correlation: Throughput



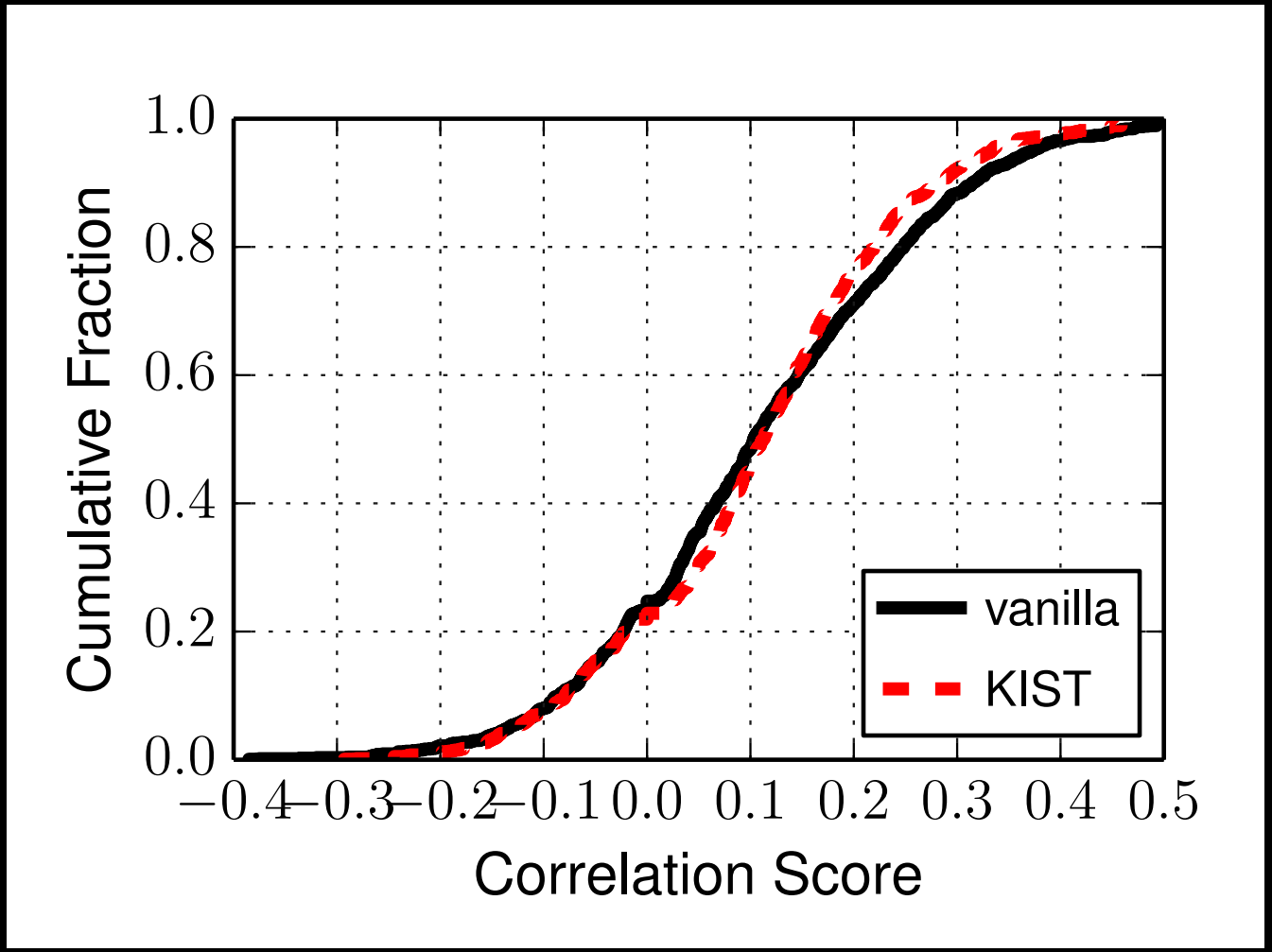
Mittal et.al. CCS'11

Traffic Correlation: Throughput



Mittal et.al. CCS'11

Throughput Attack Results



Conclusion

- Where is Tor slow?
- KIST complements other performance enhancements, e.g. circuit priority
- Next steps
  - Currently exploring various algorithmic optimizations
  - Test KIST in the wild and deploy in Tor

Questions?

rob.g.jansen@nrl.navy.mil

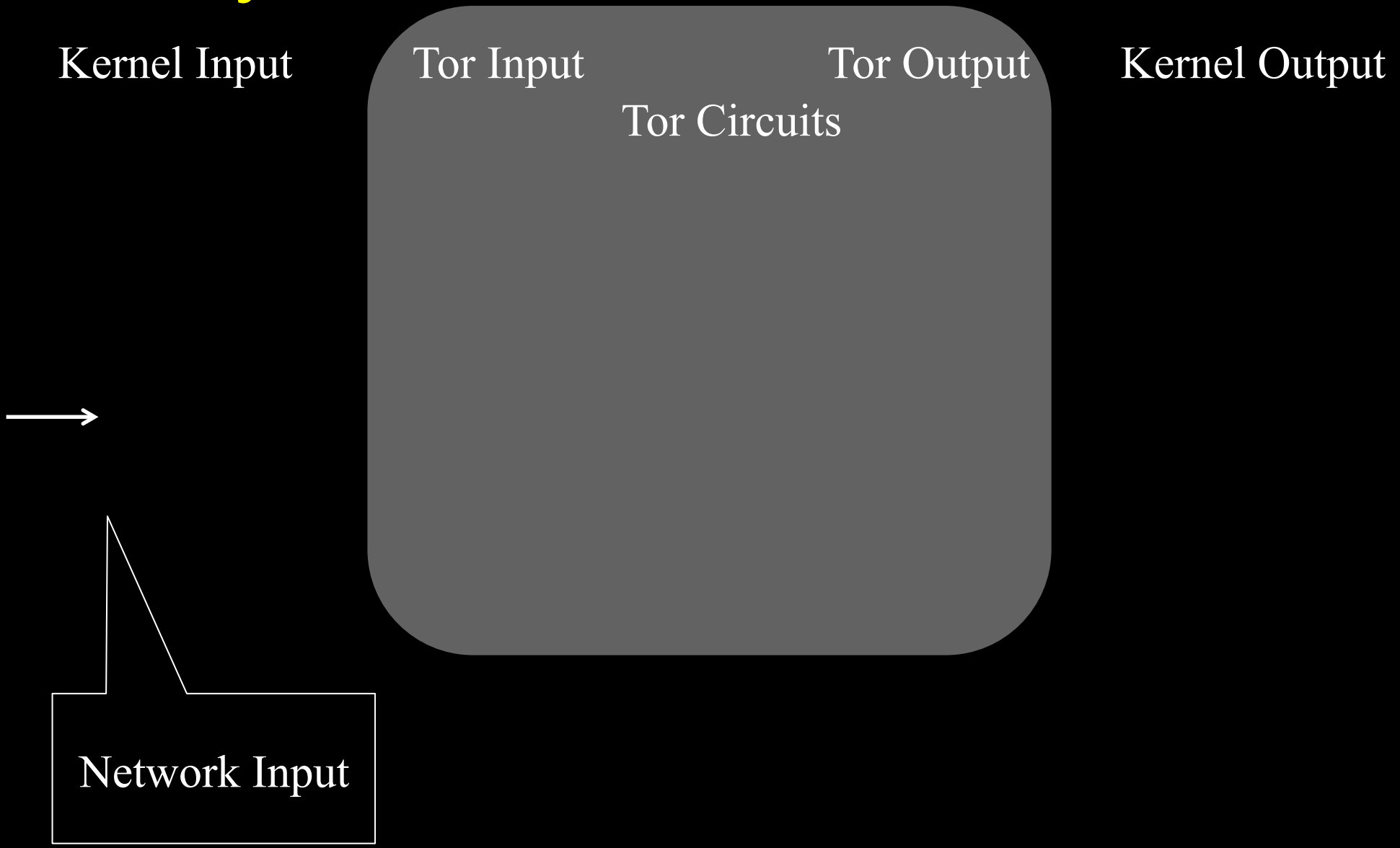
robgjansen.com

github.com/robgjansen/libkqtime

github.com/shadow

think like an adversary

Relay Internals



Relay Internals

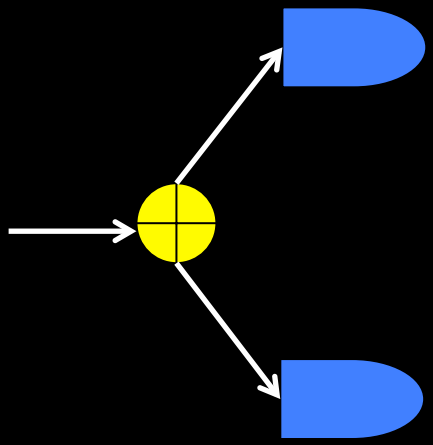
Kernel Input

Tor Input

Tor Output

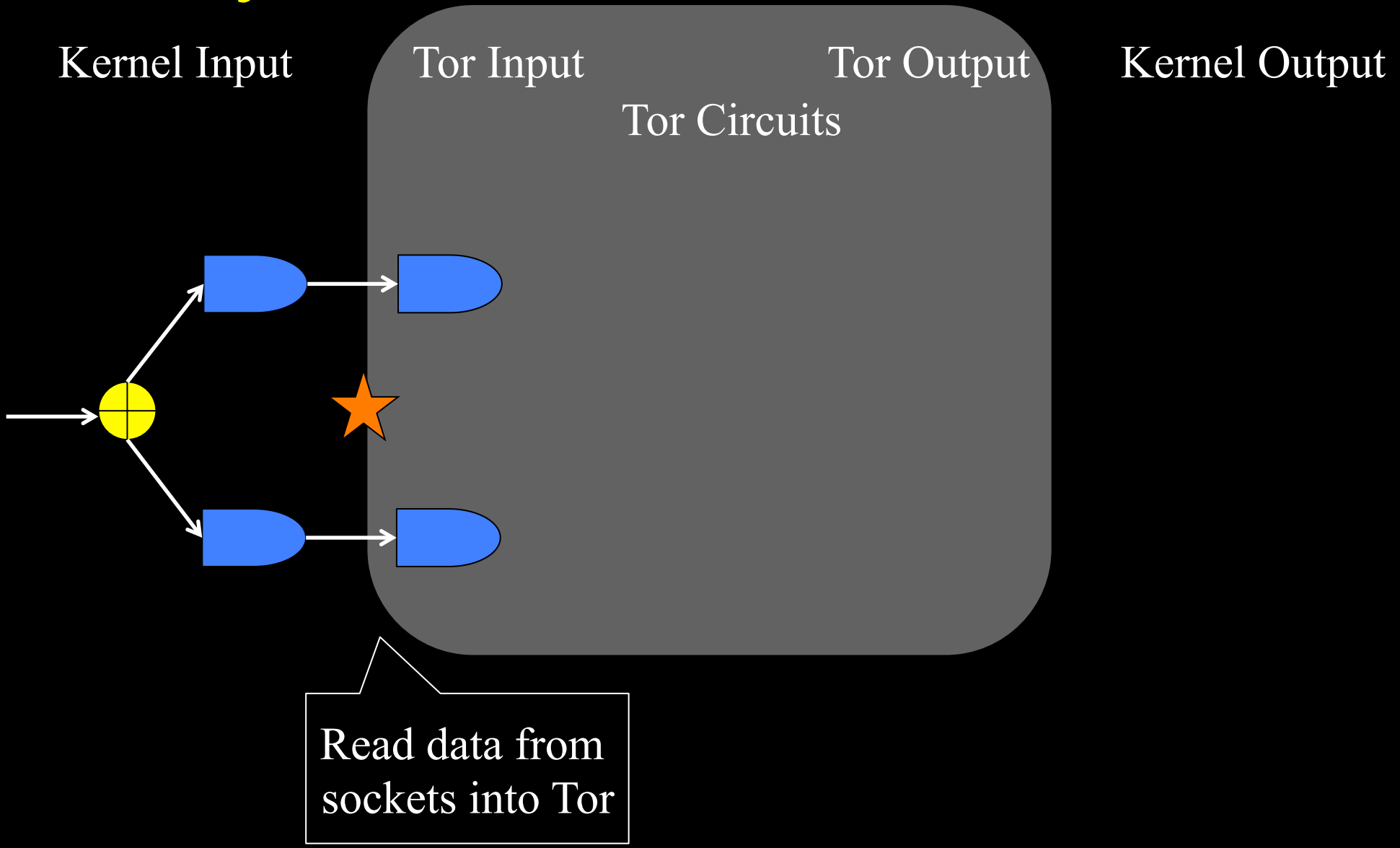
Kernel Output

Tor Circuits

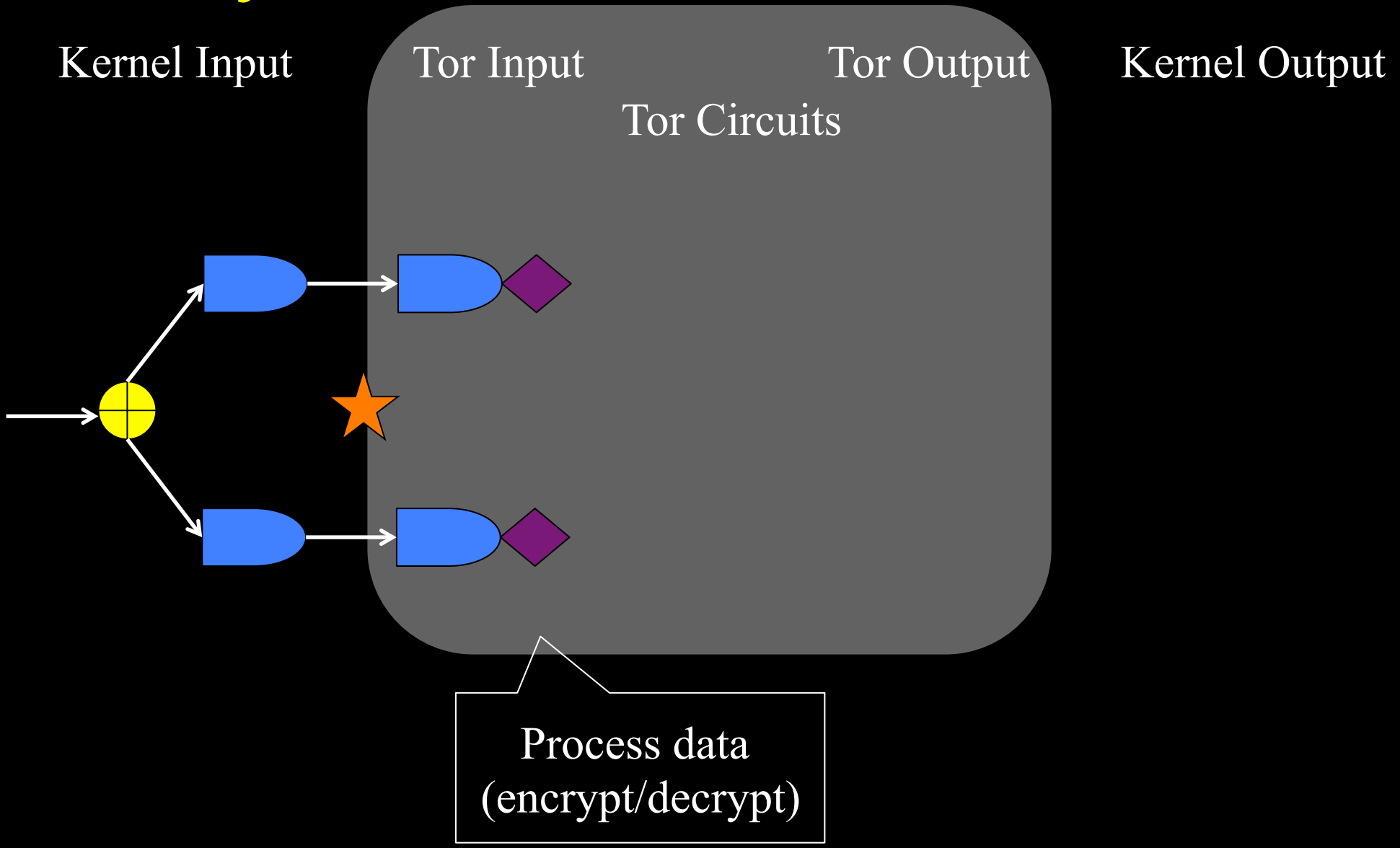


Split data into socket buffers

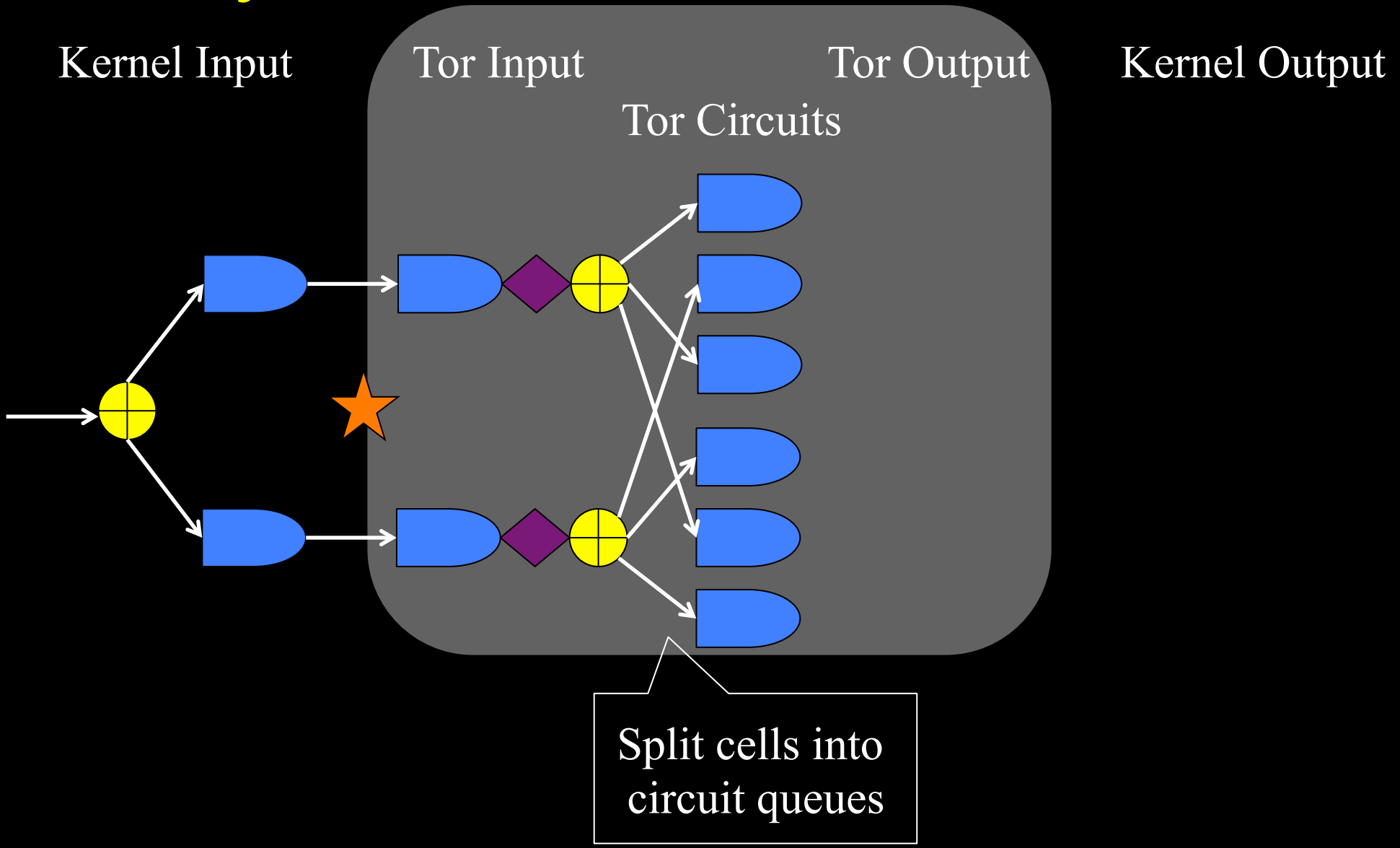
Relay Internals



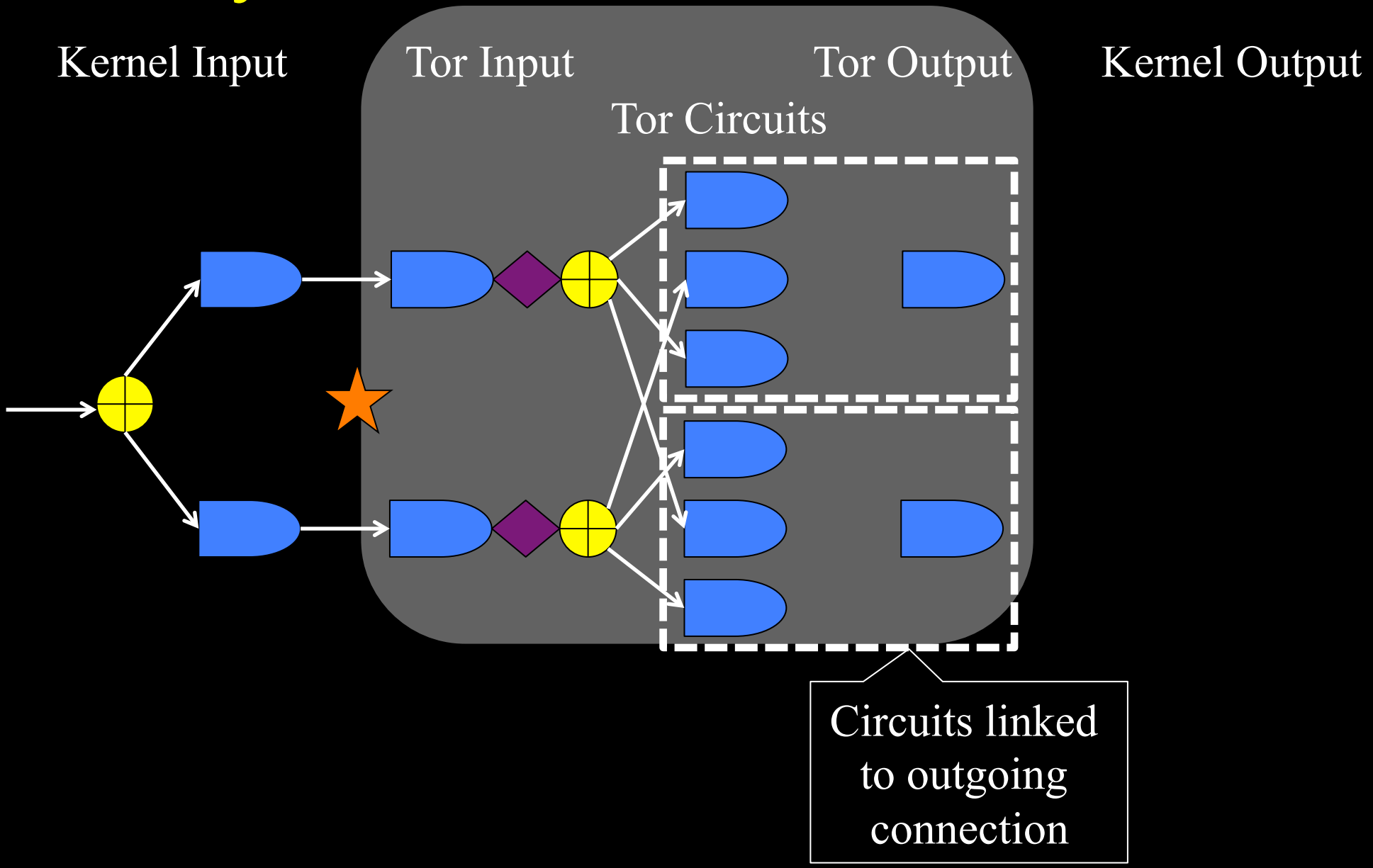
Relay Internals



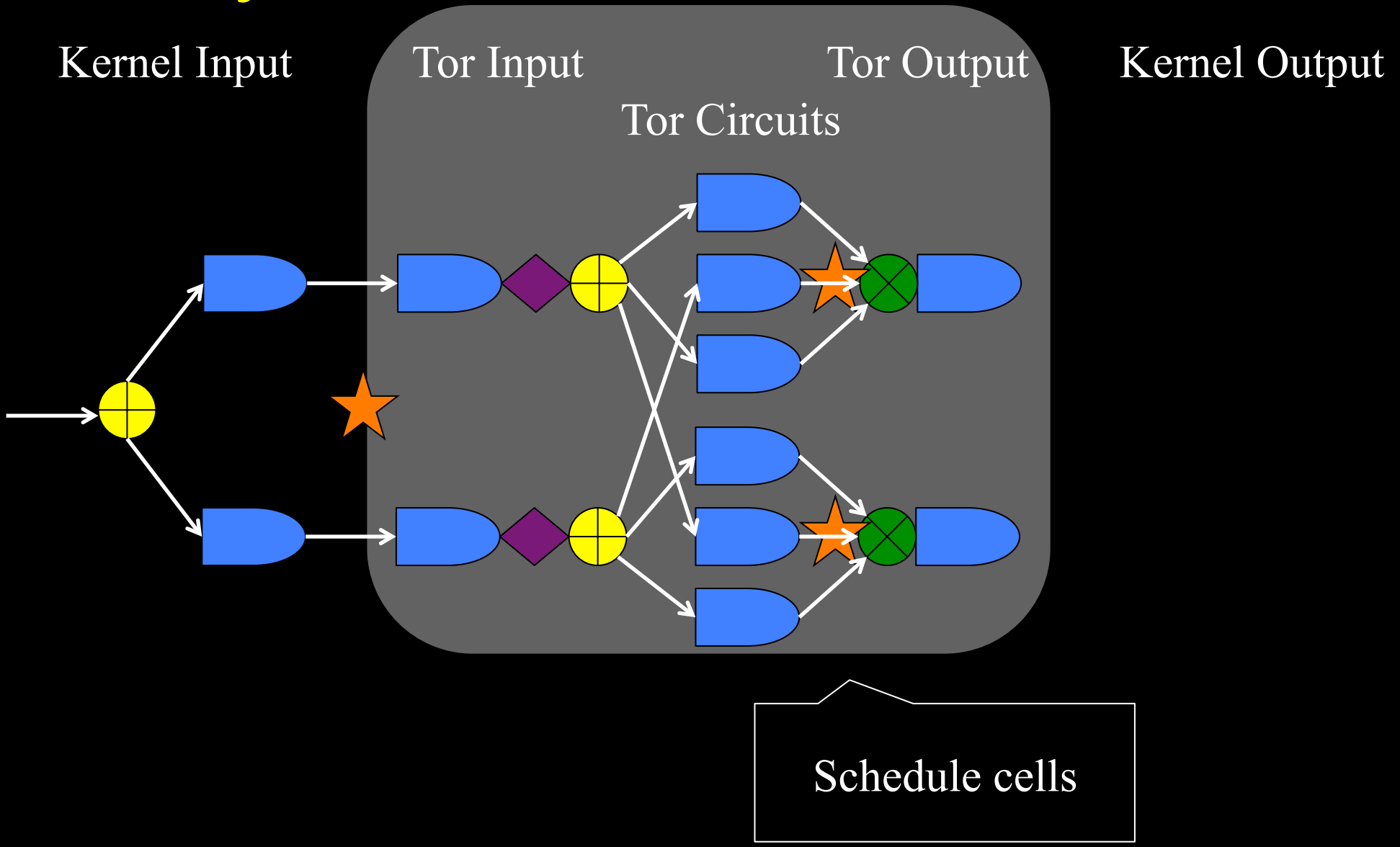
Relay Internals



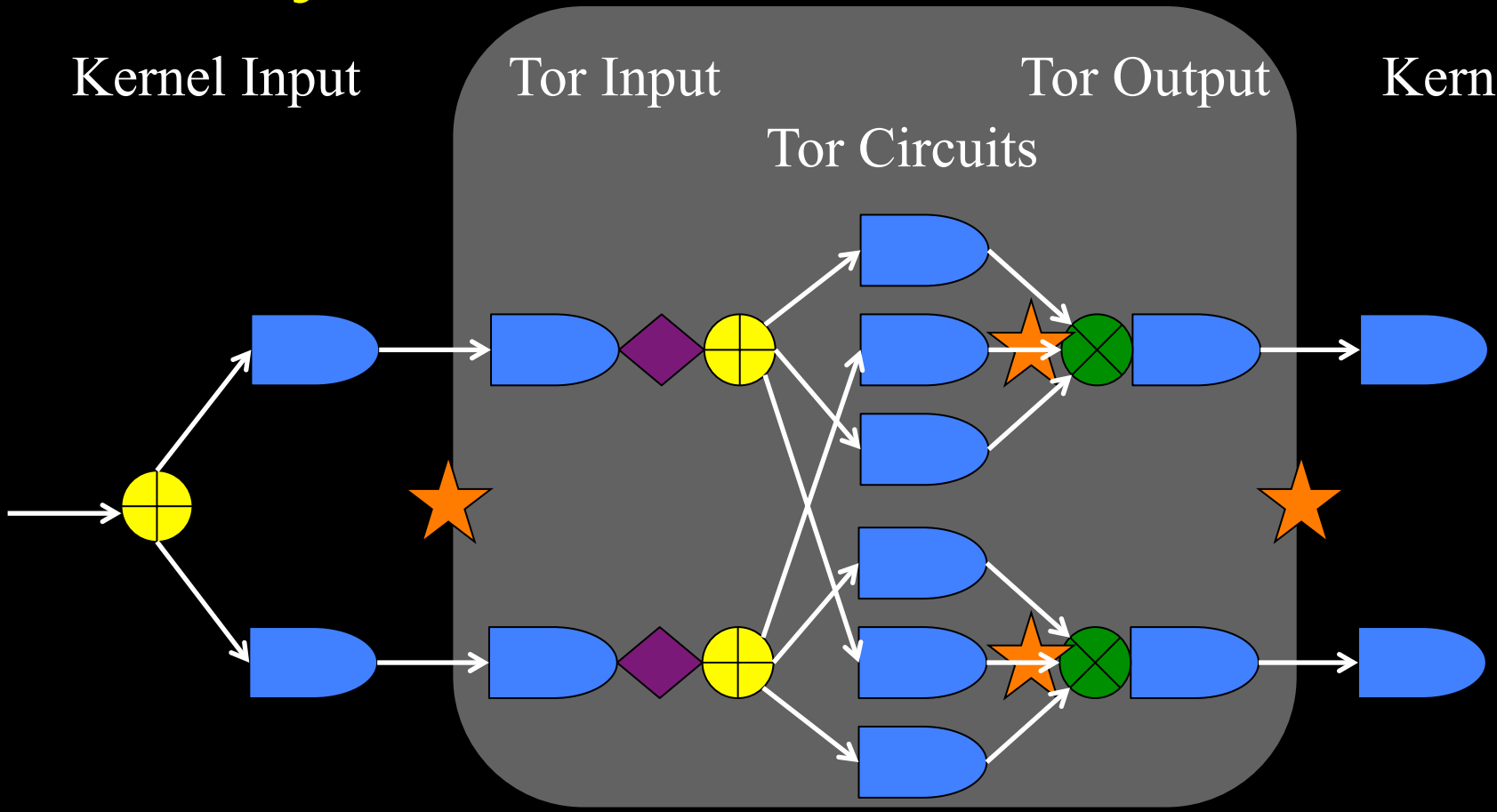
Relay Internals



Relay Internals

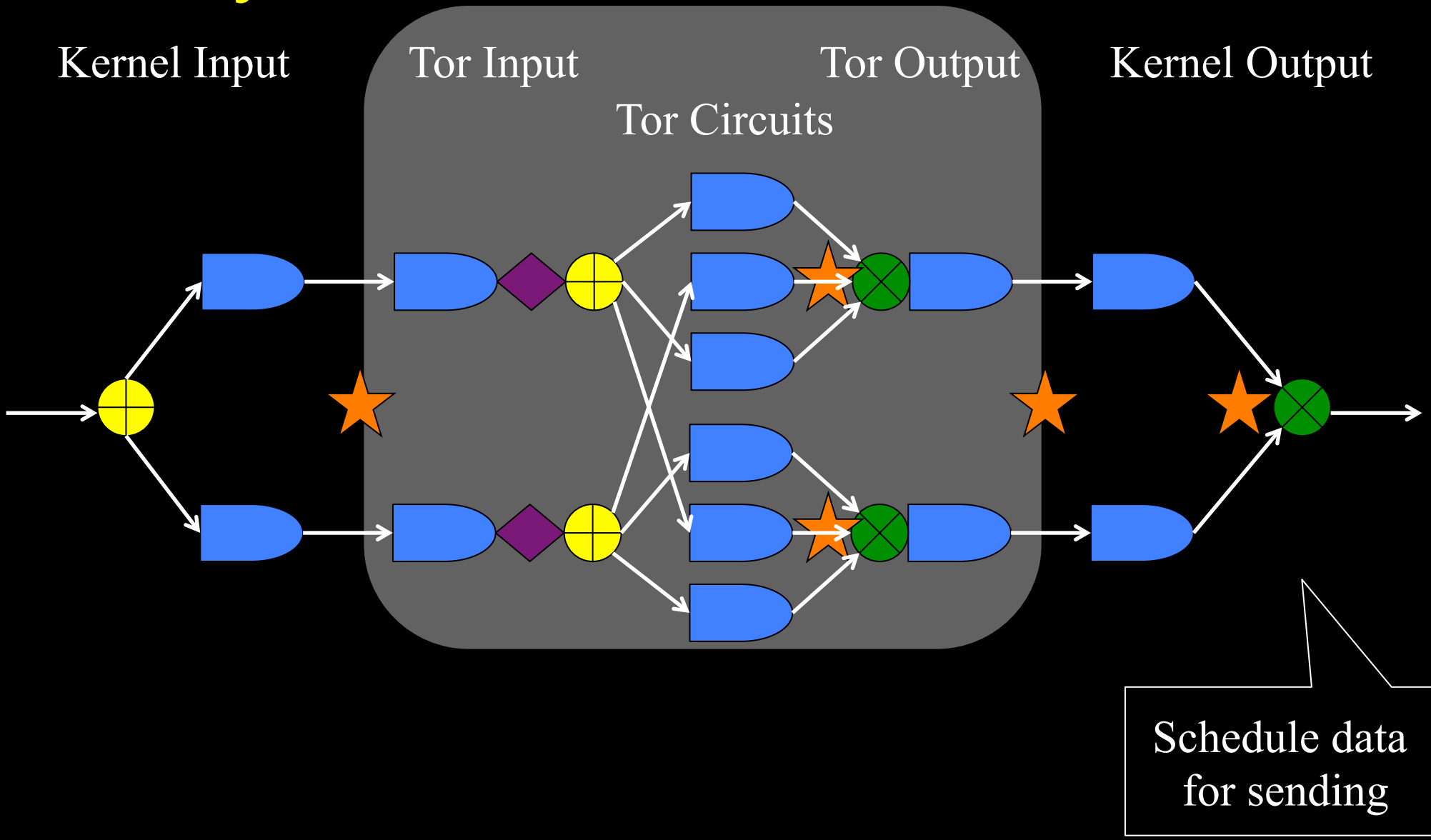


Relay Internals

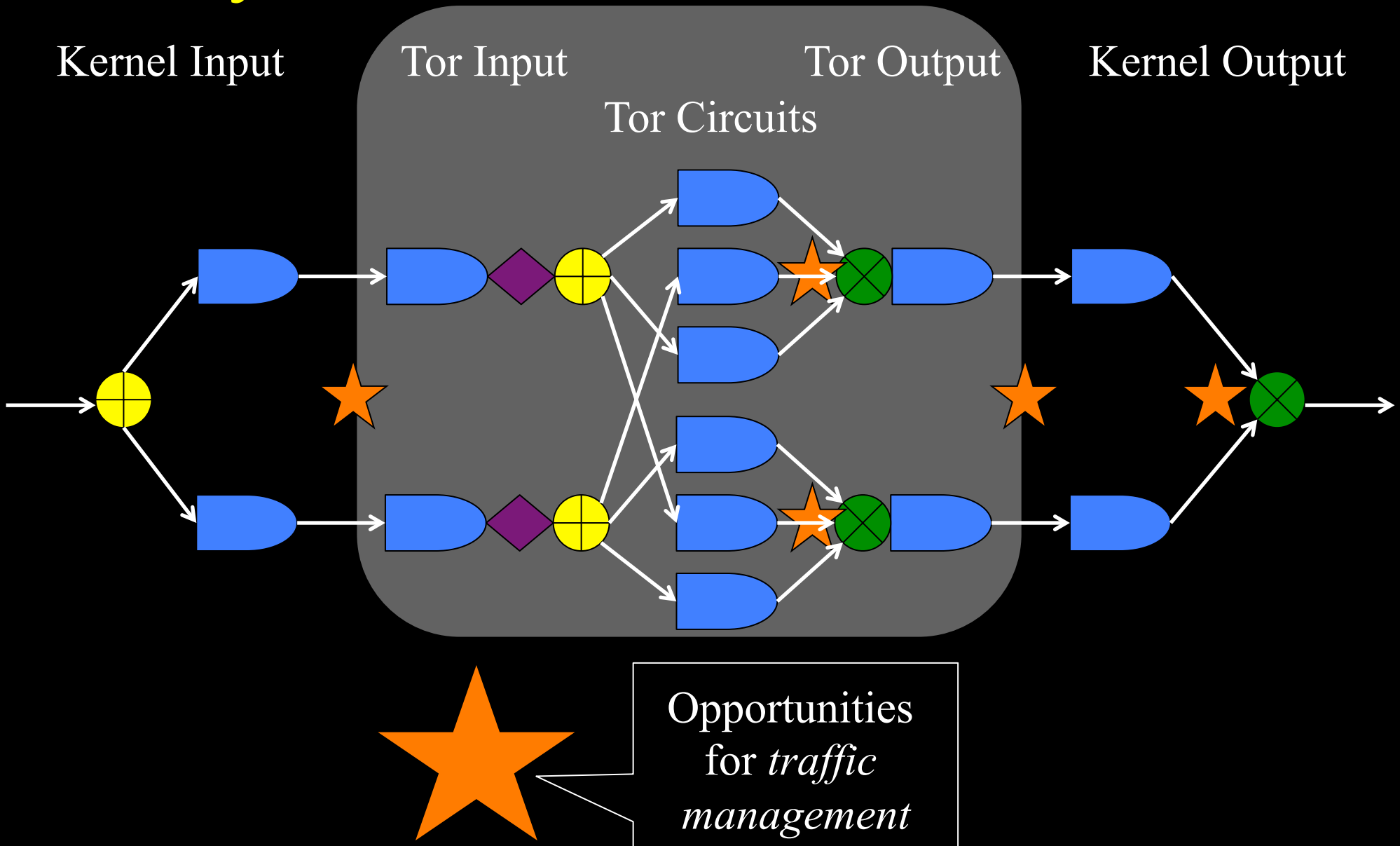


Write data from Tor into sockets

Relay Internals



Relay Internals



KIST Improves Network Throughput