CSC 2229 – Software-Defined Networking

Handout # 8: Rethinking Congestion Control in Software-Defined Networks

Professor Yashar Ganjali

Department of Computer Science

University of Toronto

yganjali@cs.toronto.edu

http://www.cs.toronto.edu/~yganjali

Joint work with Monia Ghobadi, Soheil Hassas Yeganeh

Challenges

Previous lectures

- Scalability, and performance limitations
  - Controller (network OS and applications)
  - Switches
- Standardization
  - Defining Appropriate APIs, programming models, …
  - SDN-specific hardware, …
- Human/business factors
  - Personnel adaptation, and training
  - Infrastructure transition cost, and risk

Business Opportunities

- Worldwide SDN market
  - 2013: $1.5B
  - 2016: $14.8B
  - 2018: $35.6B

- Percentage of network spending on SDN
  - 2013: 2%
  - 2016: 18%
  - 2018: 40%

Reference: SDxCentral SDN Infographic

Technical Opportunities

- Tremendous opportunity for *new apps*
  - Simplified network management, energy saving, ...
- A new set of simpler and more powerful *tools*
  - Management, integration, monitoring, debugging, ...
- New possibilities for *optimization*
  - Through network and end-host coordination

Example: Optimize TCP

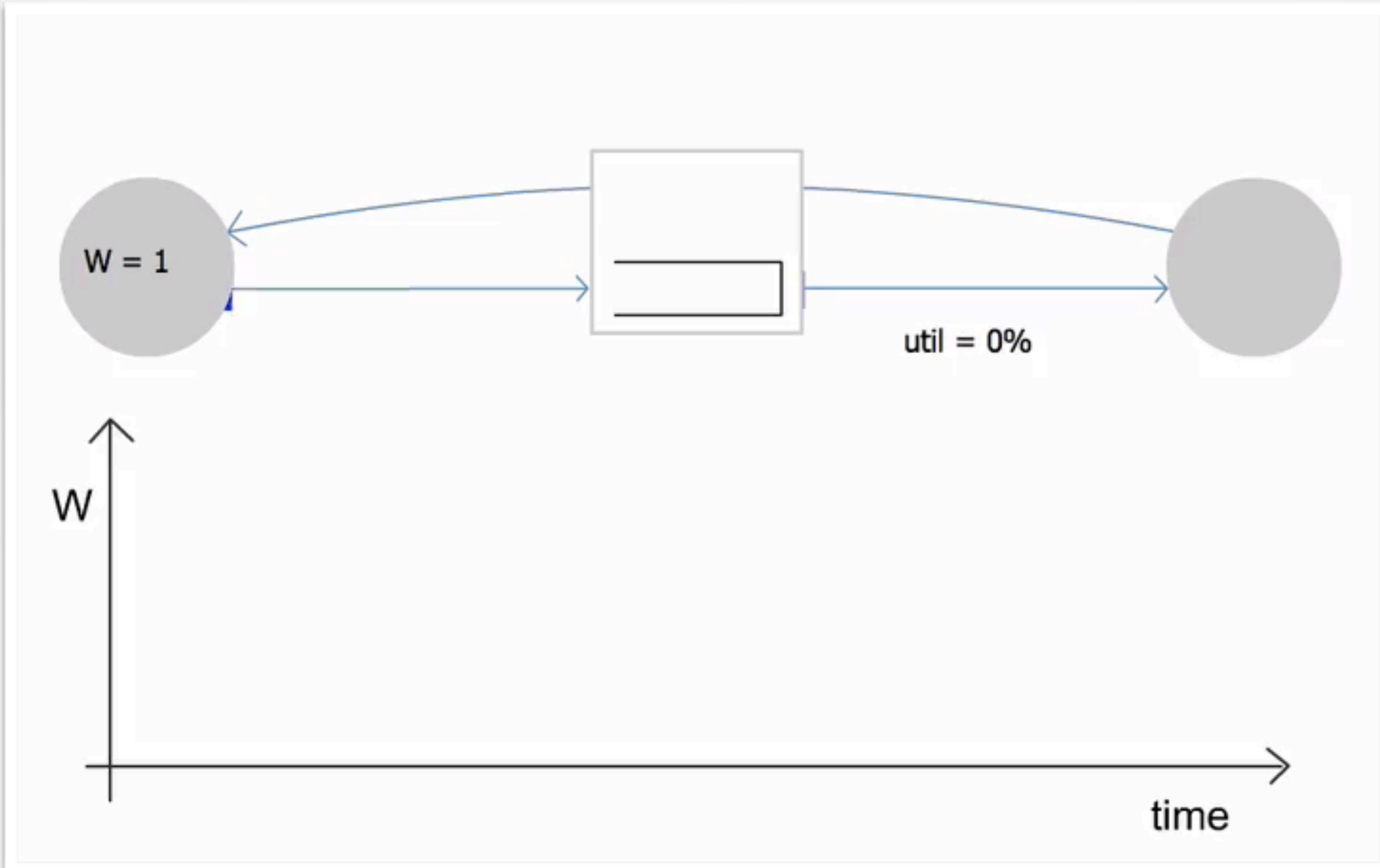
- TCP is an end-to-end protocol
  - Trying to find the *appropriate transmission rate*
  - For a given source and destination pair
- TCP blindly probes the network
  - Additive increase
  - Multiplicative decrease

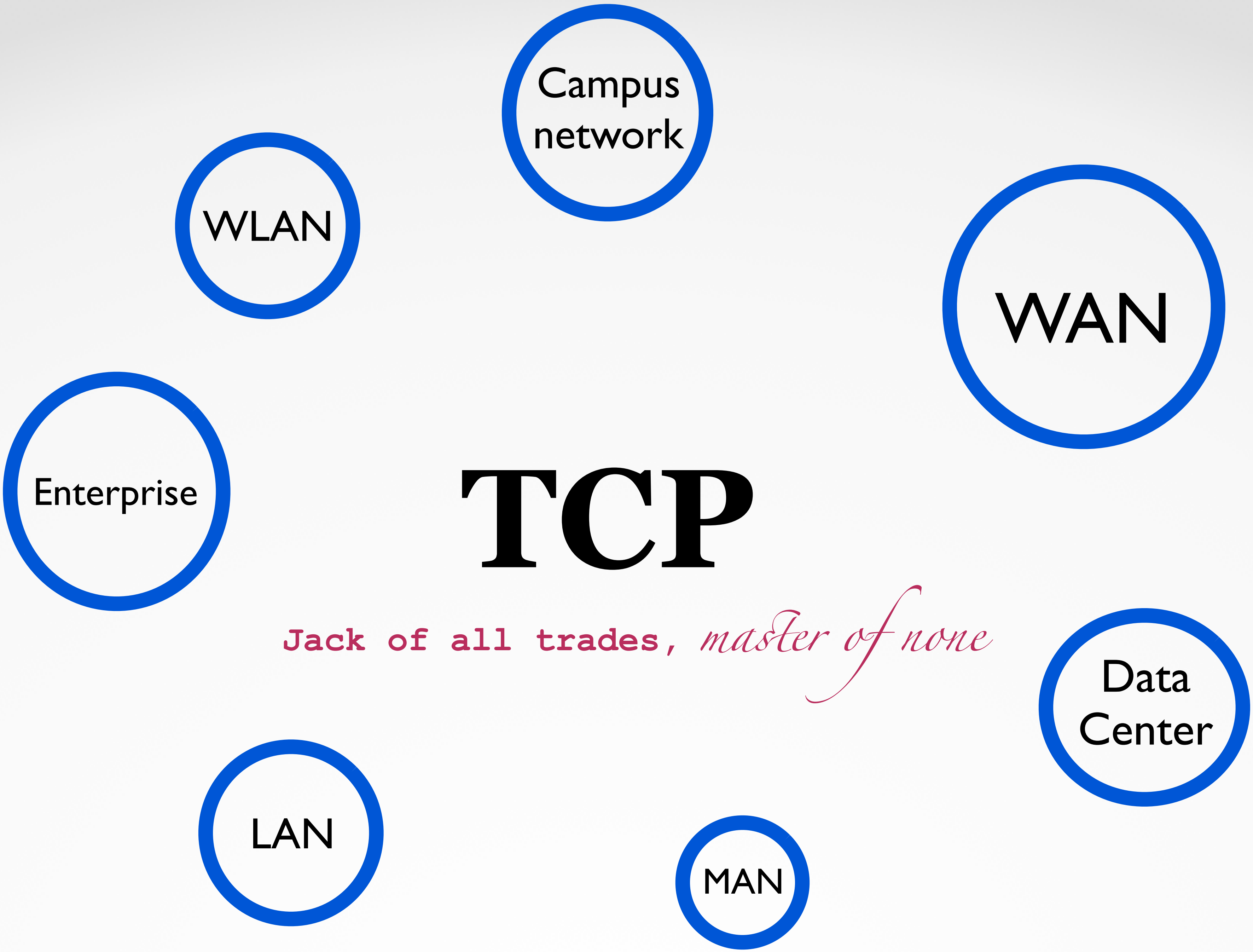
TCP Congestion Control

- Window based
  - Send W packets out
  - Wait for acknowledgement

- Adjust W to change rate
  - ACK received: W <— W+1
  - ACK dropped: W <— W/2

TCP Window Evolution




Campus network
WLAN
WAN
Enterprise
TCP
Jack of all trades, *master of none*
Data Center
LAN
MAN

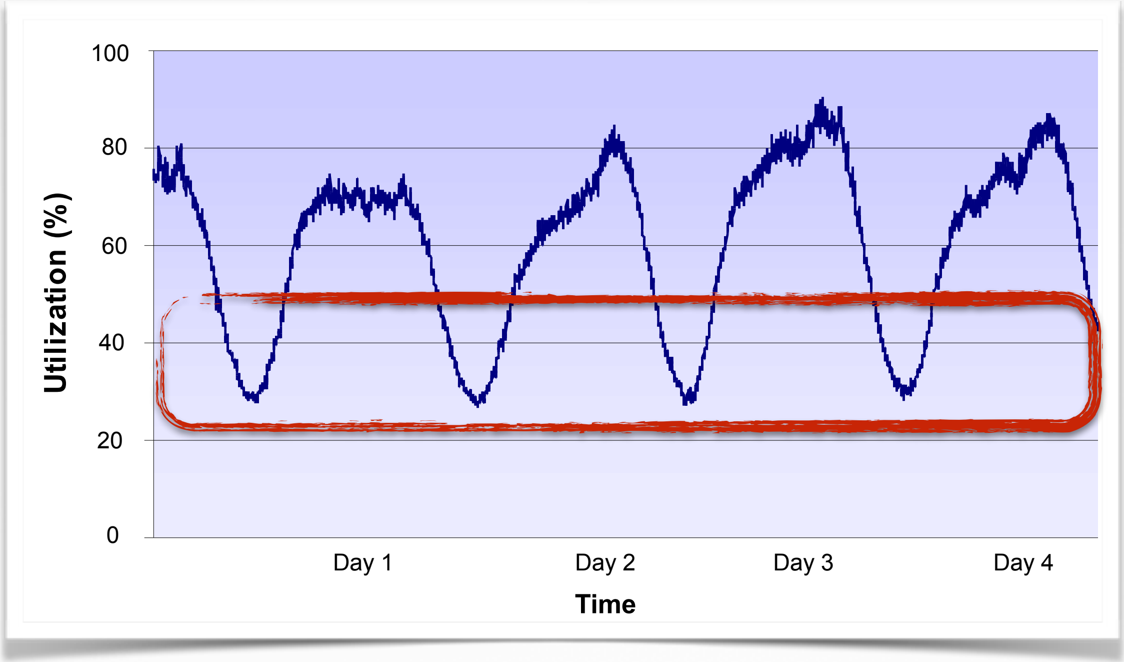
Spatial Diversity

Data Center

- Limit TCP to specific kind of network.
- Take advantage of characteristics of the network and traffic:
  - Small RTT
  - Single administrative domain
- Example: DCTCP [SIGCOMM'10]

Opportunity 1:
Adjust TCP to gain better performance in specific networks.

Temporal Diversity



Opportunity 2:
Take advantage of low utilization and improve performance.

Example

Static

- Increase TCP's initial congestion window
  - Reduce flow completion times

Dynamic

- Adjust init_cwnd value
  - Network state (flow initiations)
  - Congestion control policies

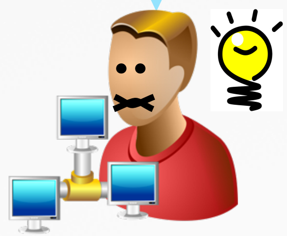
Other Parameters

- Initial congestion window
- Maximum congestion window
- TCP pacing
- TCP variant
- AIMD parameters
- …

Knowledge is Power!

A framework to observe the state and dynamics of a network and adapt TCP.

end-to-end measurements



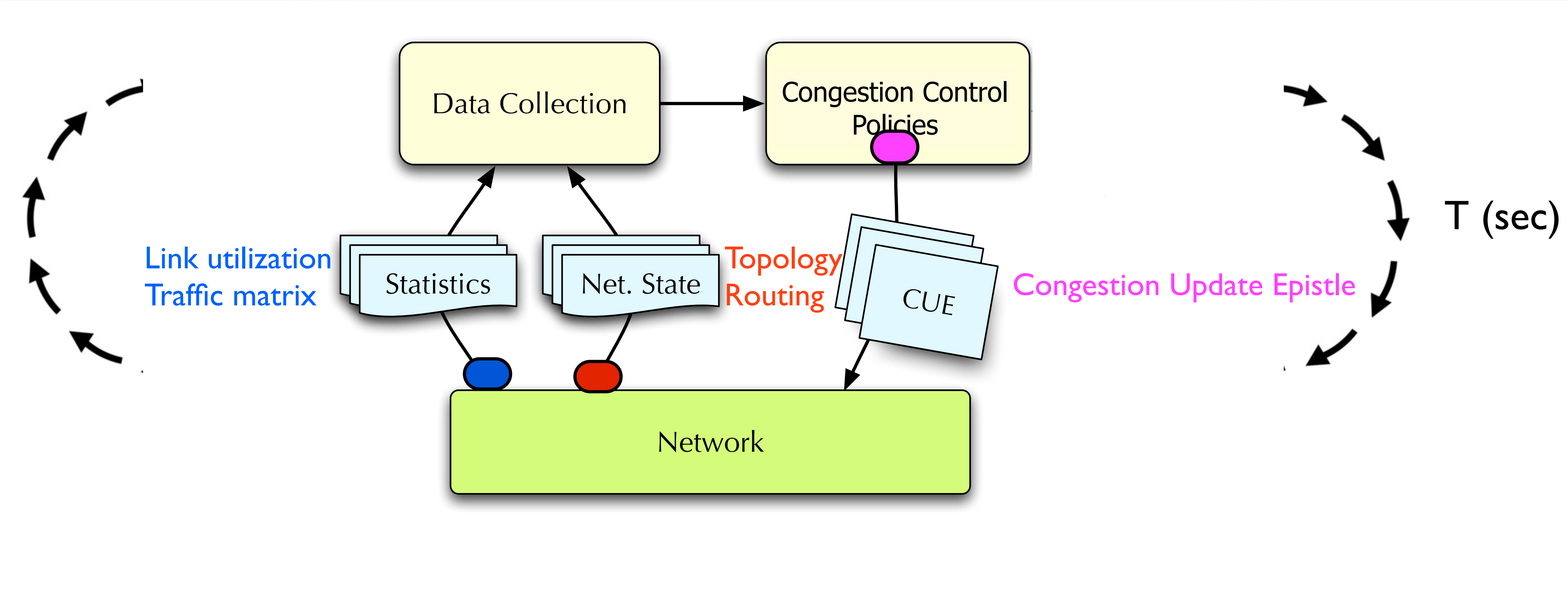
Software-Defined Networks

```
if (bottleneck link utilization < 50%)
    tcp_init_cwnd = 22
else
    tcp_init_cwnd = DEFAULT_INIT_CWND
```

OpenTCP: automated TCP adaptation framework.

OpenTCP Design

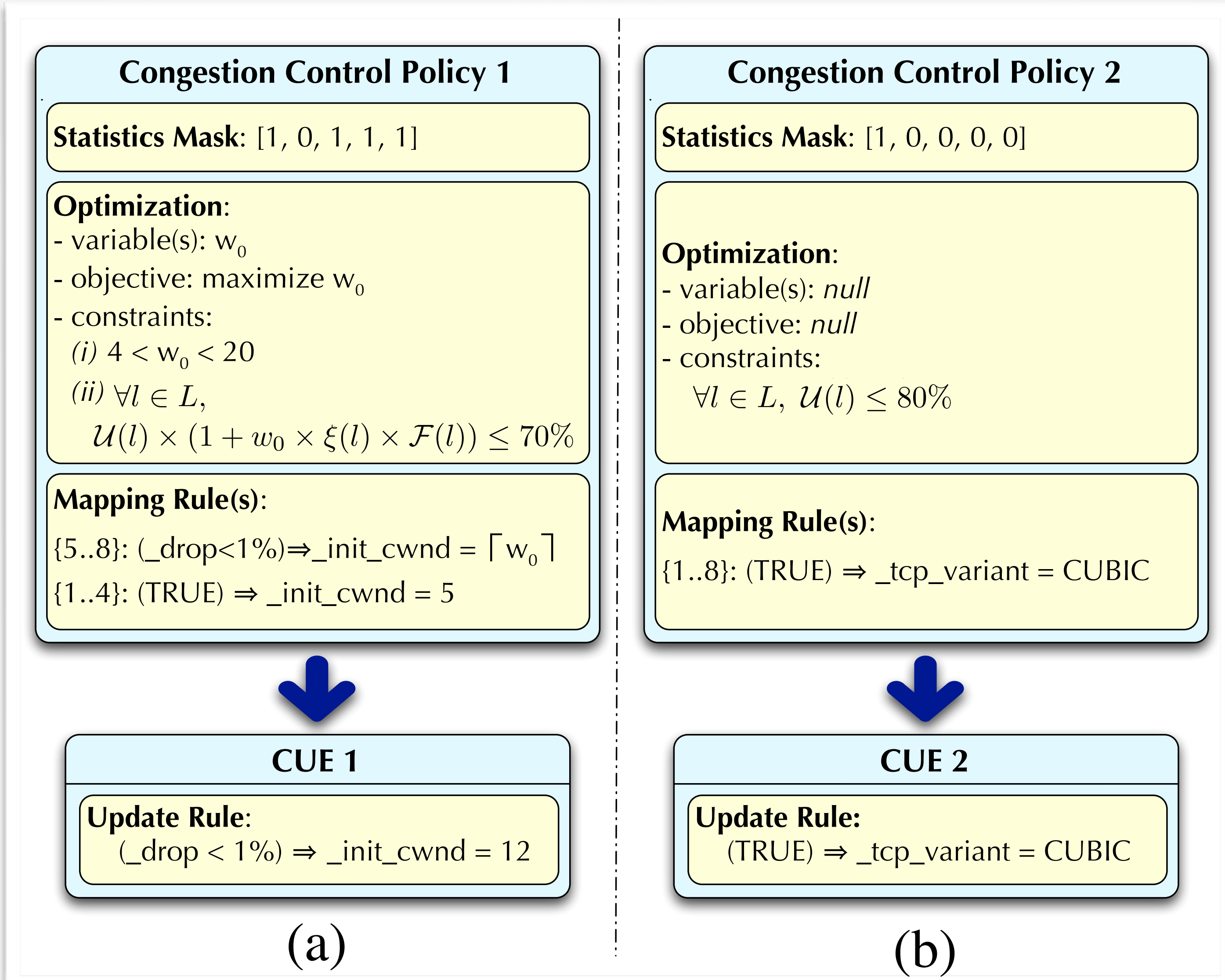
- Cycle of three steps: (i) data collection, (ii) hint generation, (iii) TCP adaptation
- Two-timescale control: T and RTT



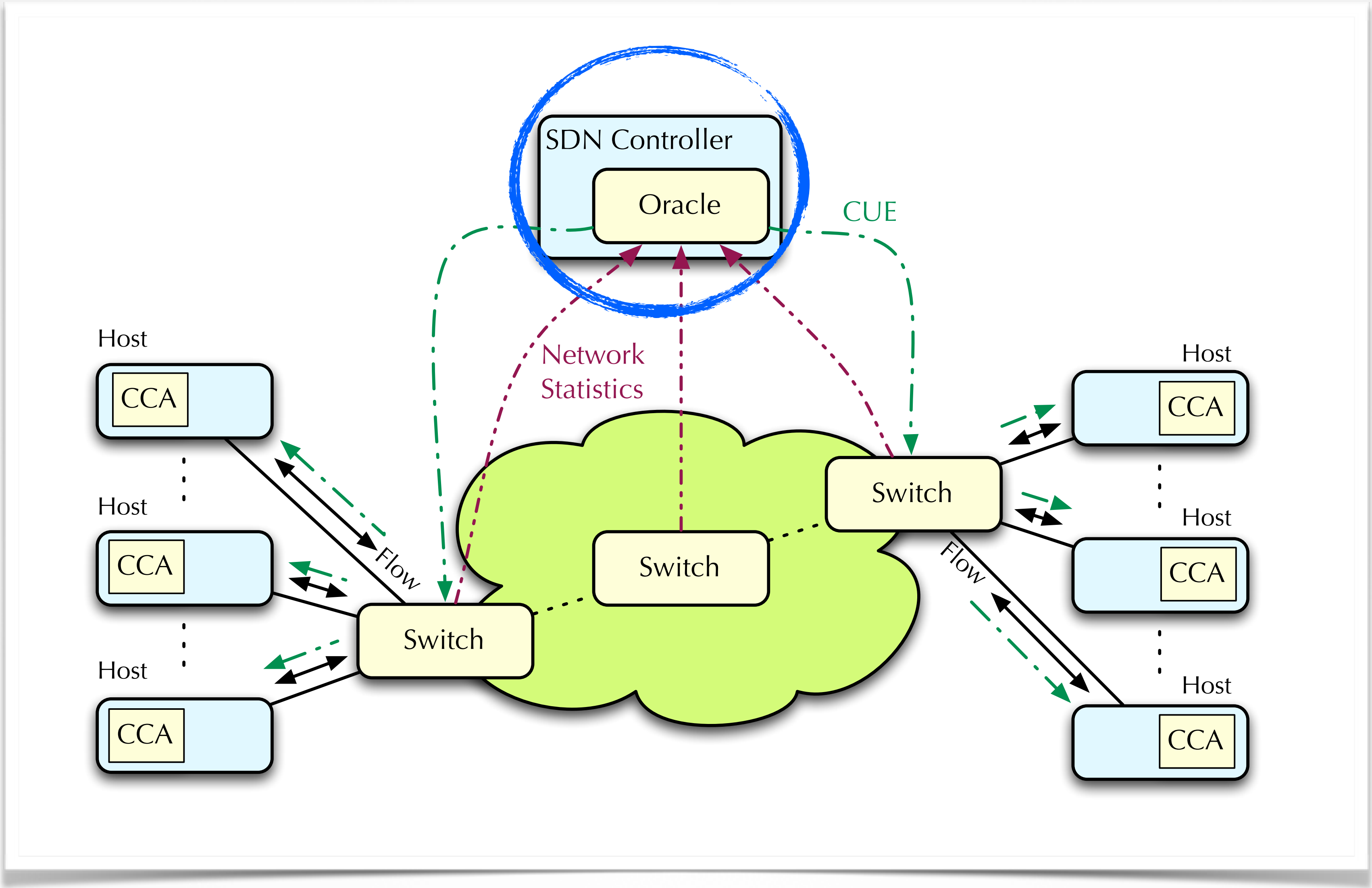
OpenTCP Policies

- Major decisions are made by the network operator
  - Statistics need to be collected
  - Target operational goal
  - Constraints
  - Mapping

Formalization

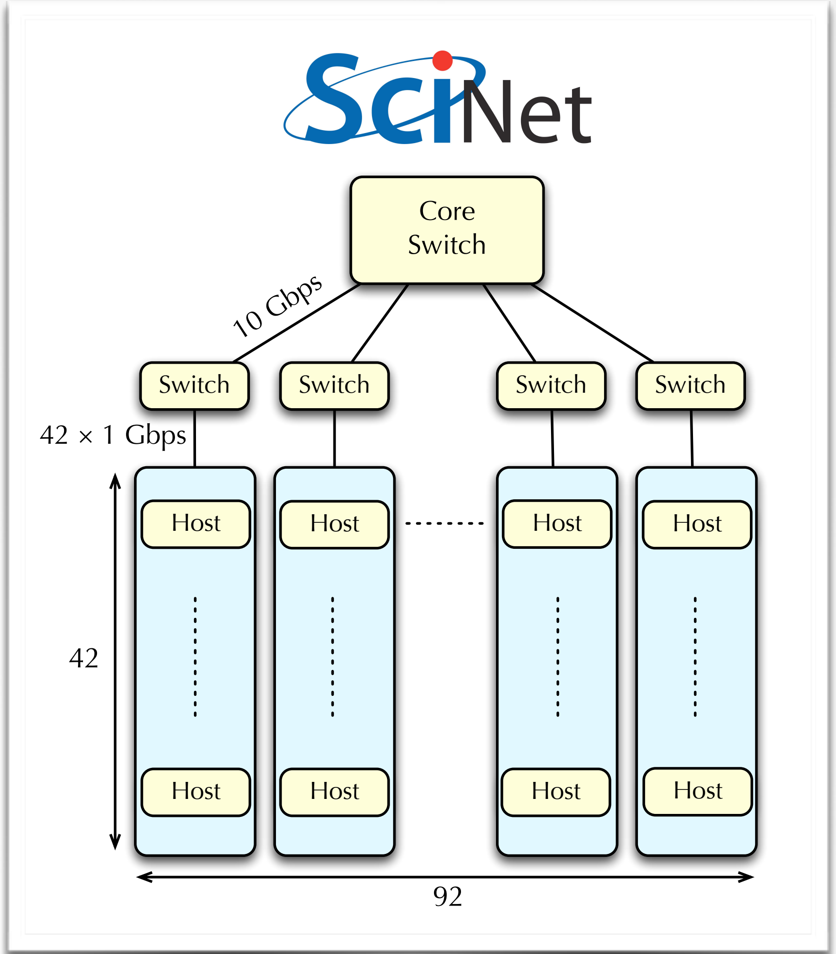


OpenTCP Architecture

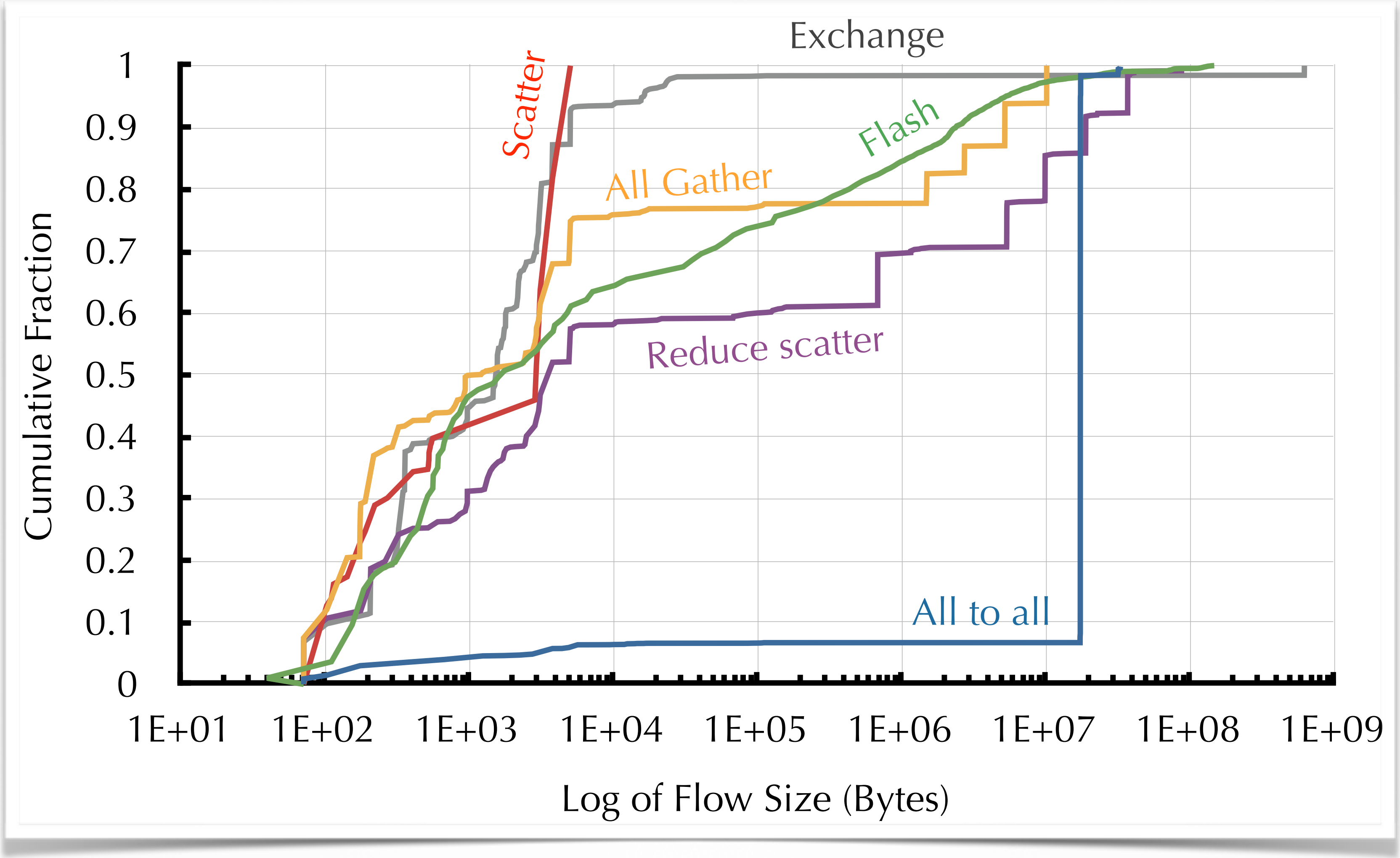


OpenTCP Evaluation

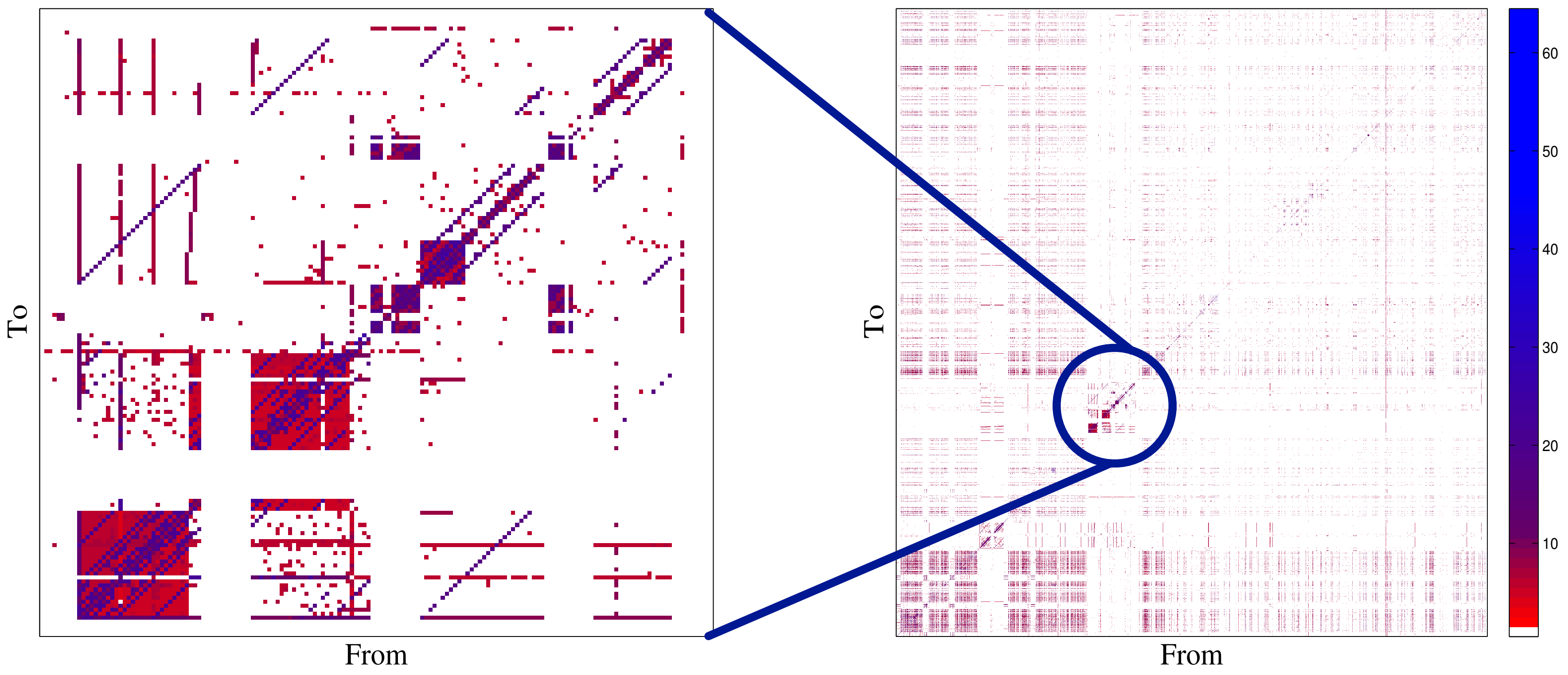
- Deployed in a 4000 node data center
  - 32,000 cores
  - A simple kernel module receiving CUEs from the SDN controller
- Up to 64% improvement in flow completion times
  - And much more...



Flow Sizes of Benchmarks



Traffic Matrix

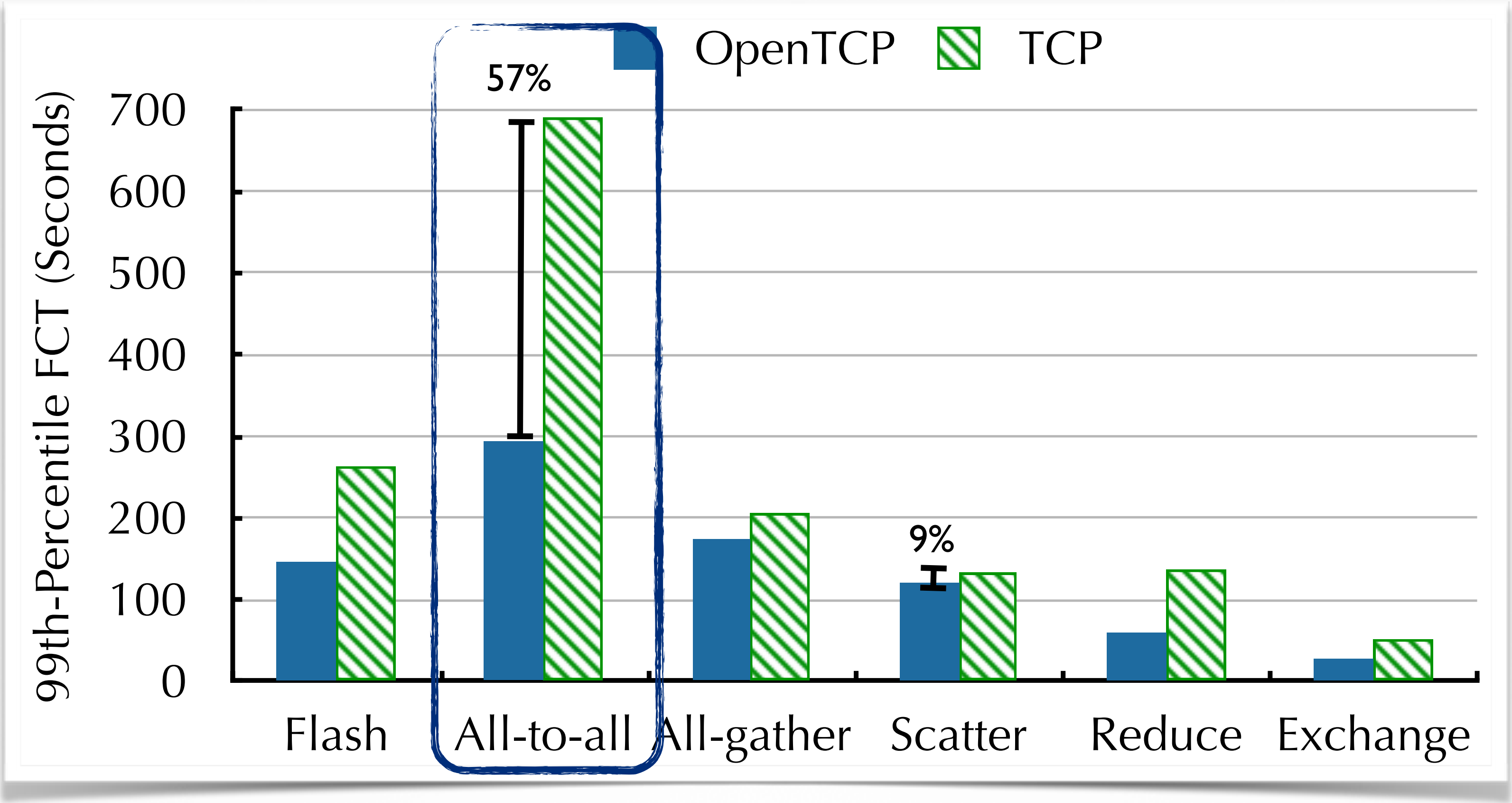


Experiments

- Measure impact on
  - Flow Completion Times (FCTs)
  - Drop rate
- To have an apple-to-apple comparison
  - Split the servers into two sets
  - Change one part
  - Keep the second part the same

Benchmarks

- Intel MPI Benchmarks (IMB)
- Sample user jobs in SciNet



OpenTCP Varients

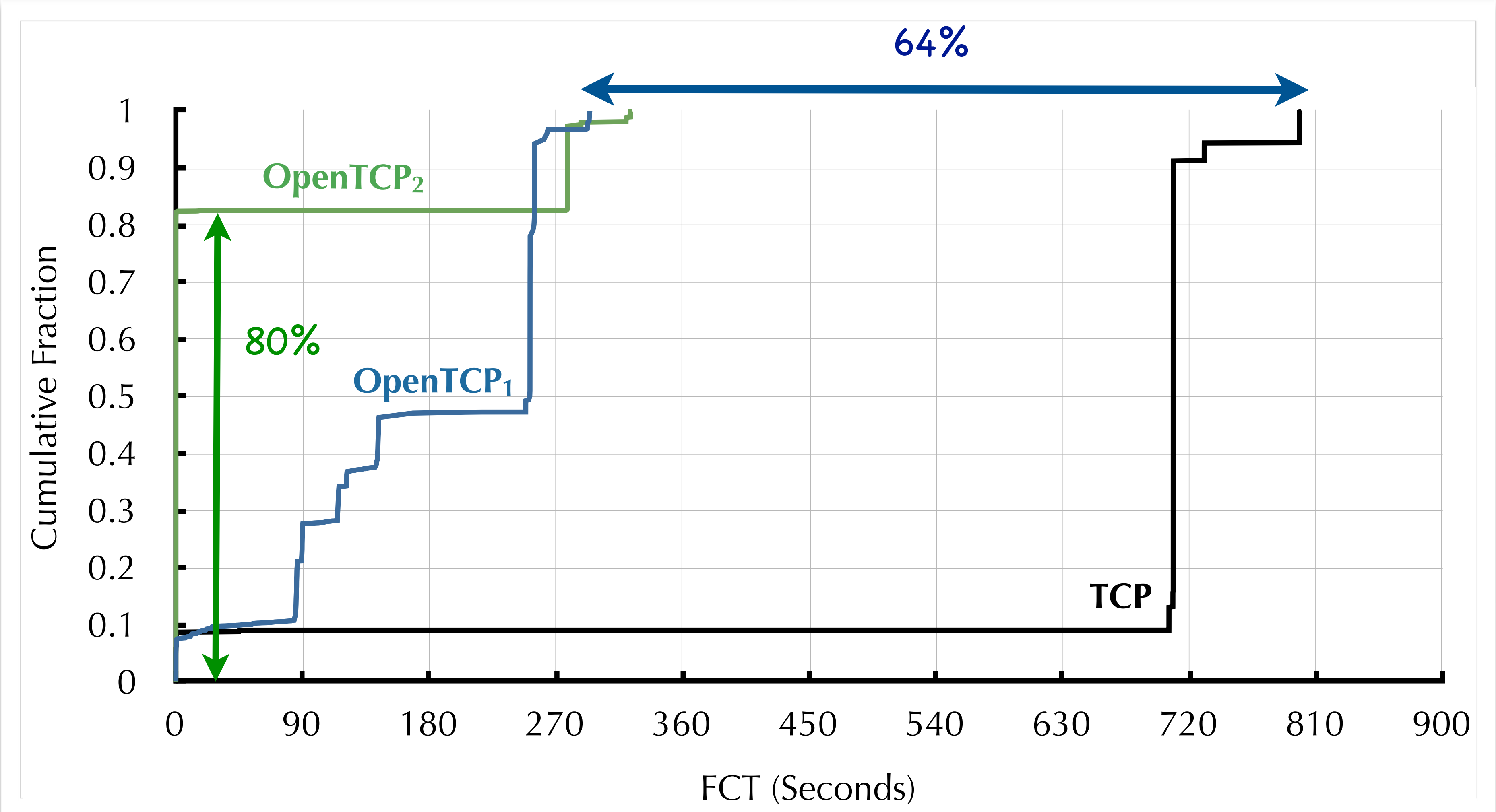
- **OpenTCP1**: Reduce FCTs by updating the initial congestion window size and the Retransmission Timeout (RTO)

```
if util < 70%
  init_cwnd = 20, RTO = 2ms
else
  init_cwnd = DEFAULT, RTO = DEFAULT
```

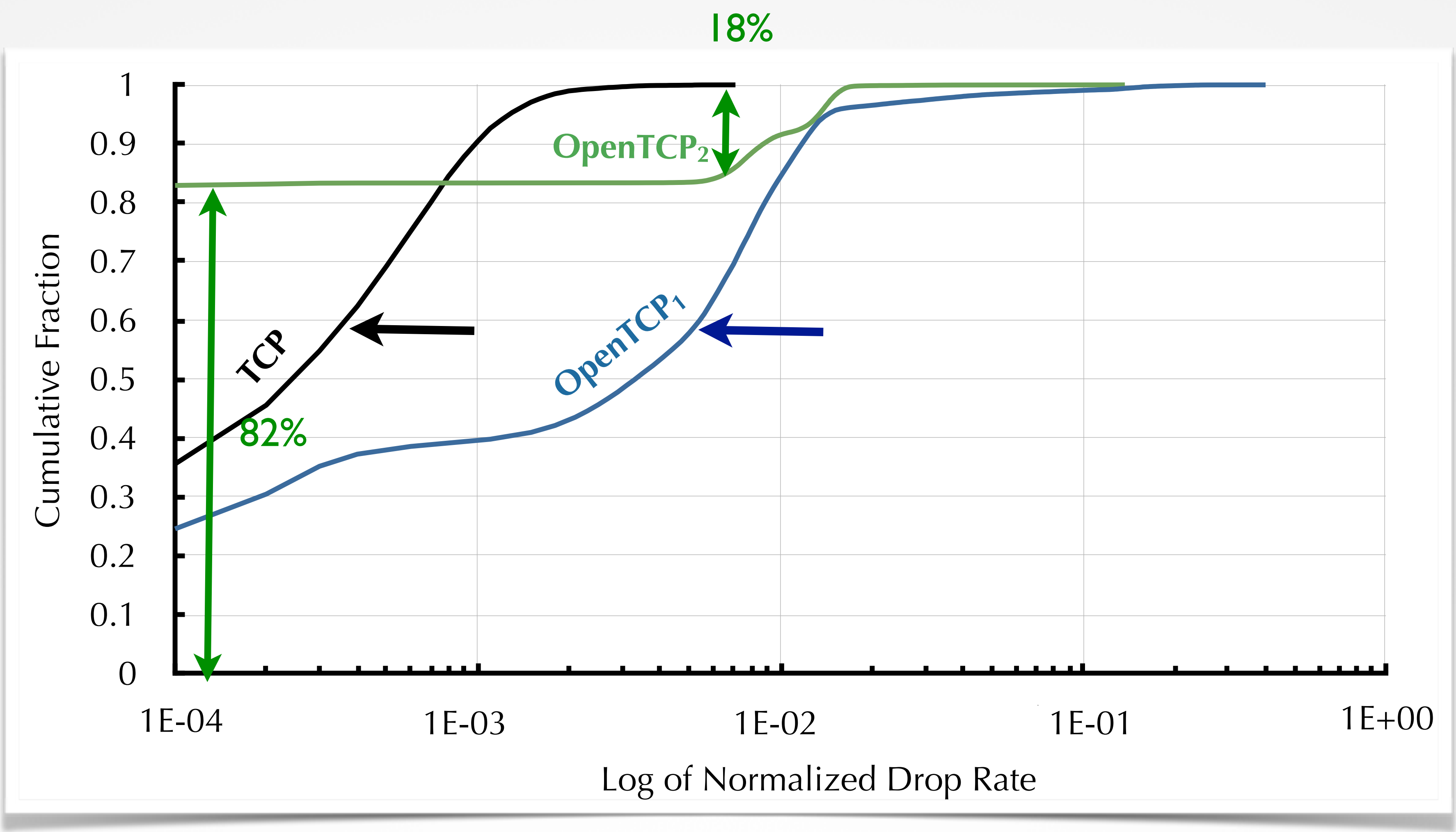
- **OpenTCP2**: Keep the packet drop rate below 0.1%

```
if util < 70% && drop < 0.1%
  init_cwnd = 20, RTO = 2ms
else
  init_cwnd = DEFAULT, RTO = DEFAULT
```

Flow Completion Times



Impact on Packet Drop Rate



OpenTCP Overhead

- 4000 nodes, 100 switches
- Different refresh rates

Overhead \ Oracle refresh cycle	1 min	5 min	10 min
Oracle CPU Overhead (%)	0.9	0.0	0.0
CCA CPU Overhead (%)	0.5	0.1	0.0
Data Collection (Kbps)	453	189	95
CUE Dissemination (Kbps)	530	252	75

OpenTCP and Related work

- OpenTCP is orthogonal to previous works improving TCP's performance
  - It is not meant to be a new variation of TCP
  - It complements previous efforts
- The decisions are made in advance through the congestion control policies
  - Defined by network operators