Queuing Theory in Network Congestion Control

CPE 553: Advanced Networking

Ian Jackson | 04/24/2025

Introduction

- TCP is the backbone of internet data transfer
- Congestion control in TCP is critical for maintaining stability and performance
- Understanding queue behavior helps analyze throughput, latency, and loss [1]
- **Objective:**
  - Use queueing theory to model and analyze TCP congestion control behavior
  - Use M/M/1 queue

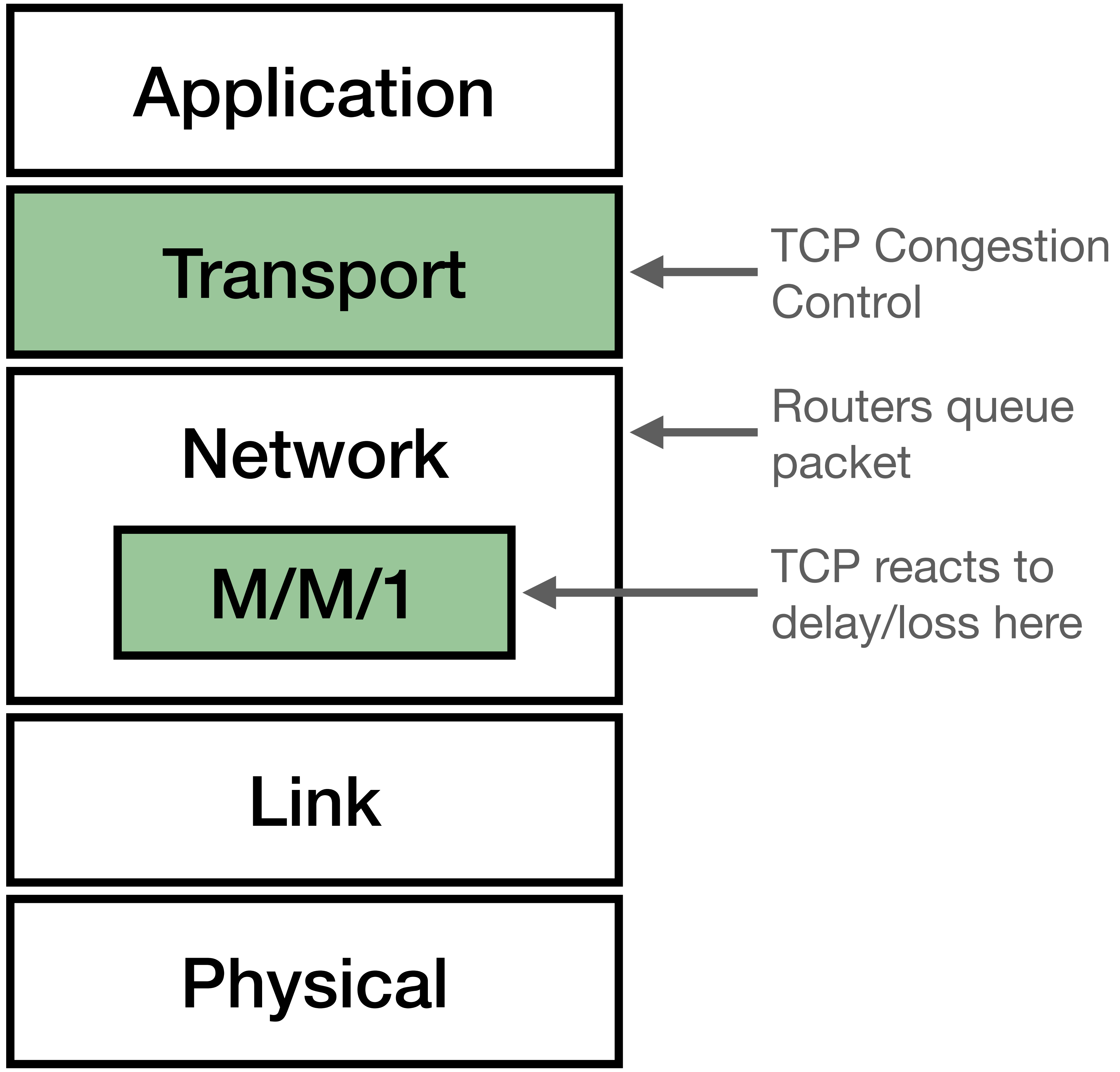


Figure 1. TCP congestion control and queuing theory in the five-layer internet stack

Background

TCP Congestion Control

- Ensures the sender does not overwhelm the network
- Operates through several distinct states
  - *Slow Start*: `cwnd` grows exponentially until loss or `ssthresh` is reached
  - *Congestion Avoidance*: `cwnd` grows linearly, until three duplicate ACKs or loss
  - *Fast Recovery*: pulls back on `cwnd` and `ssthresh`, returns to congestion avoidance [2]
- Multiple flavors: TCP Tahoe, Reno, CUBIC, …

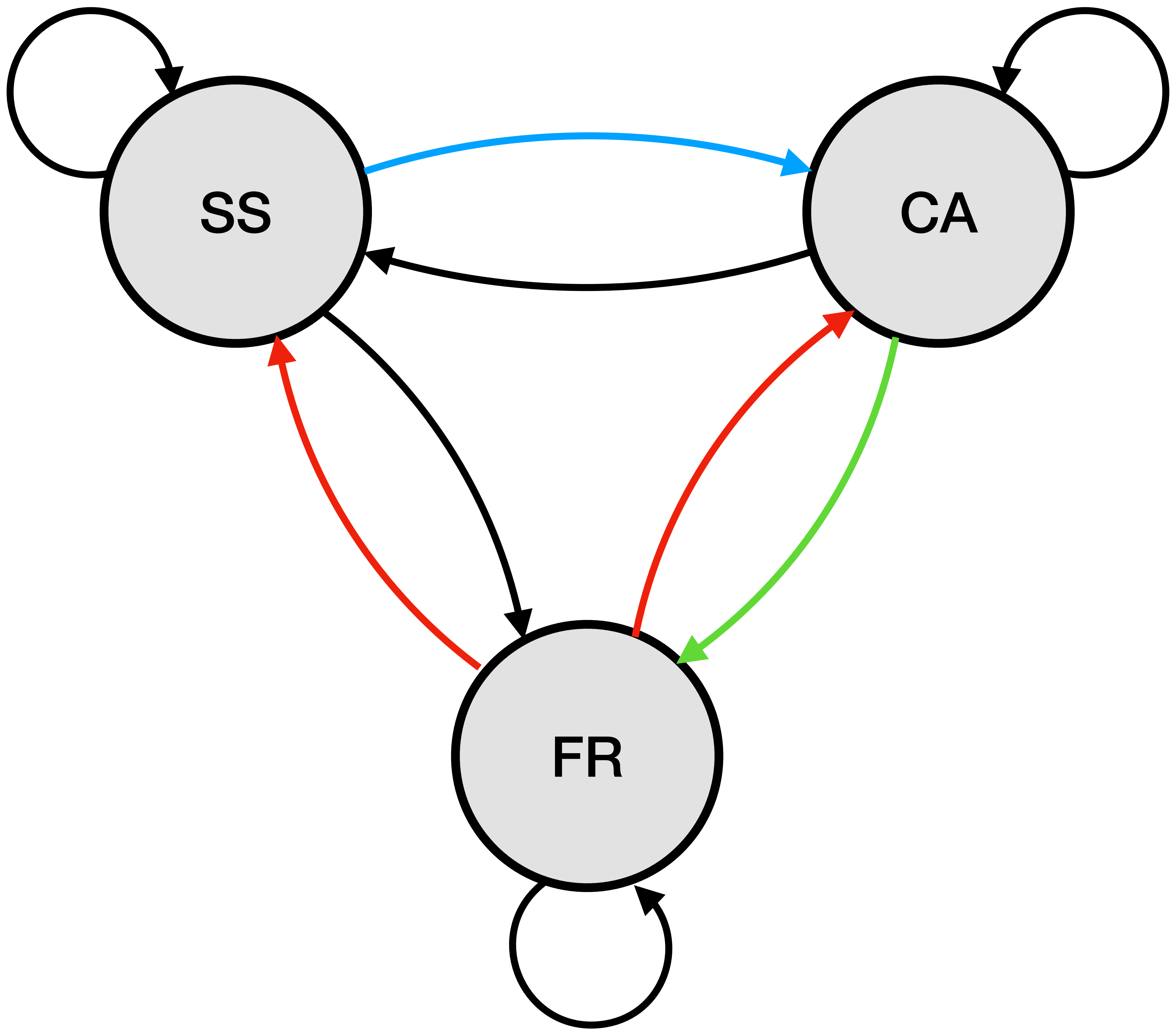


Figure 2. Simplified State Transition Diagram for TCP Congestion Control

Background

TCP Congestion Control

TCP Tahoe	TCP Reno	TCP CUBIC

• Upon packet loss:
  • `cwnd` = 1
  • `ssthresh` = `cwnd` / 2
• Enters slow start after loss
• No fast recover mechanism
• Lost packet immediately retransmitted
[2]
	• Adds fast recovery state
	• On 3 duplicate ACKs
	  • Retransmit lost segment
	  • `ssthresh` = `cwnd` / 2
	  • `cwnd` = `ssthresh`
	• Performs *better* than Tahoe under light/moderate packet loss
	• Struggles with multiple losses per window
	[3]
		• Replaced AIMD with *cubic* growth function
		• `cwnd` increases rapidly when far from previous max
		• `cwnd` slows down near max value
		• Better utilization of high-bandwidth, high-latency networks
		• More aggressive than Reno
		[4,5]

Background

Queuing Theory - M/M/1

- M/M/1 queue is a single-server queuing model
- **M**emoryless inter-arrival time / **M**emoryless service time / **1** server [6]
    - Arrival rate (λ): customers arrive according to Poisson process with average rate λ

$$T_a \sim \lambda e^{-\lambda t};\ \mathbb{E}[T_a] = \frac{1}{\lambda}$$

    - Service rate (μ): service times are exponential distributed with average rate μ

$$T_s \sim \mu e^{-\mu t};\ \mathbb{E}[T_s] = \frac{1}{\mu}$$

    - FCFS
- TCP reacts to this by reducing sending rate (`cwnd`); like controlling λ

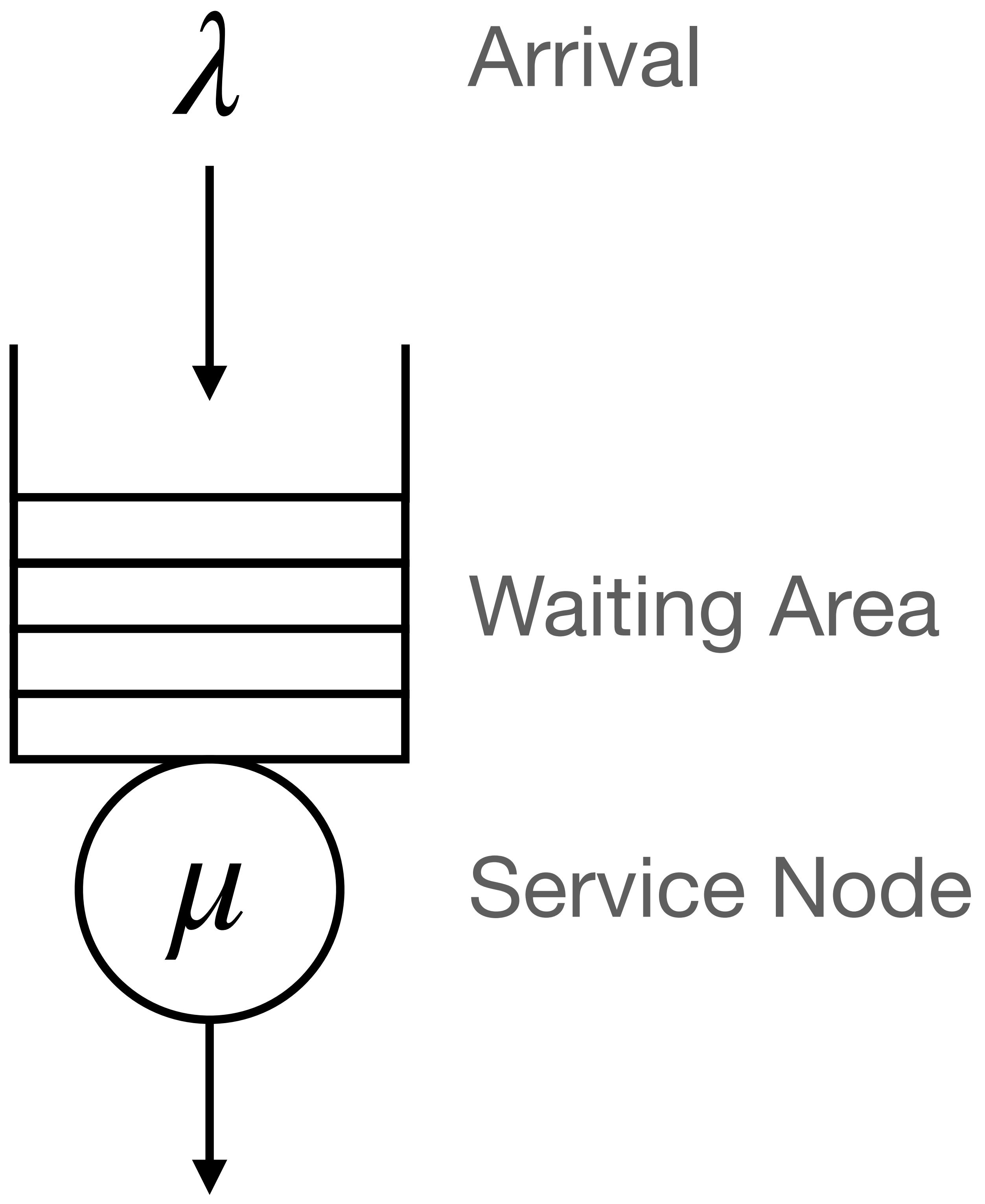


Figure 3. M/M/1 Queuing Model

System Modeling

- Event-driven simulation
- Arrival generate based on λ, controlled by TCP sending logic
- Service time for each packet based on μ, FIFO discipline at queue
- Dropped packet invokes TCP congestion control logic
- Metrics tracked: throughput, avg. latency, drop rate

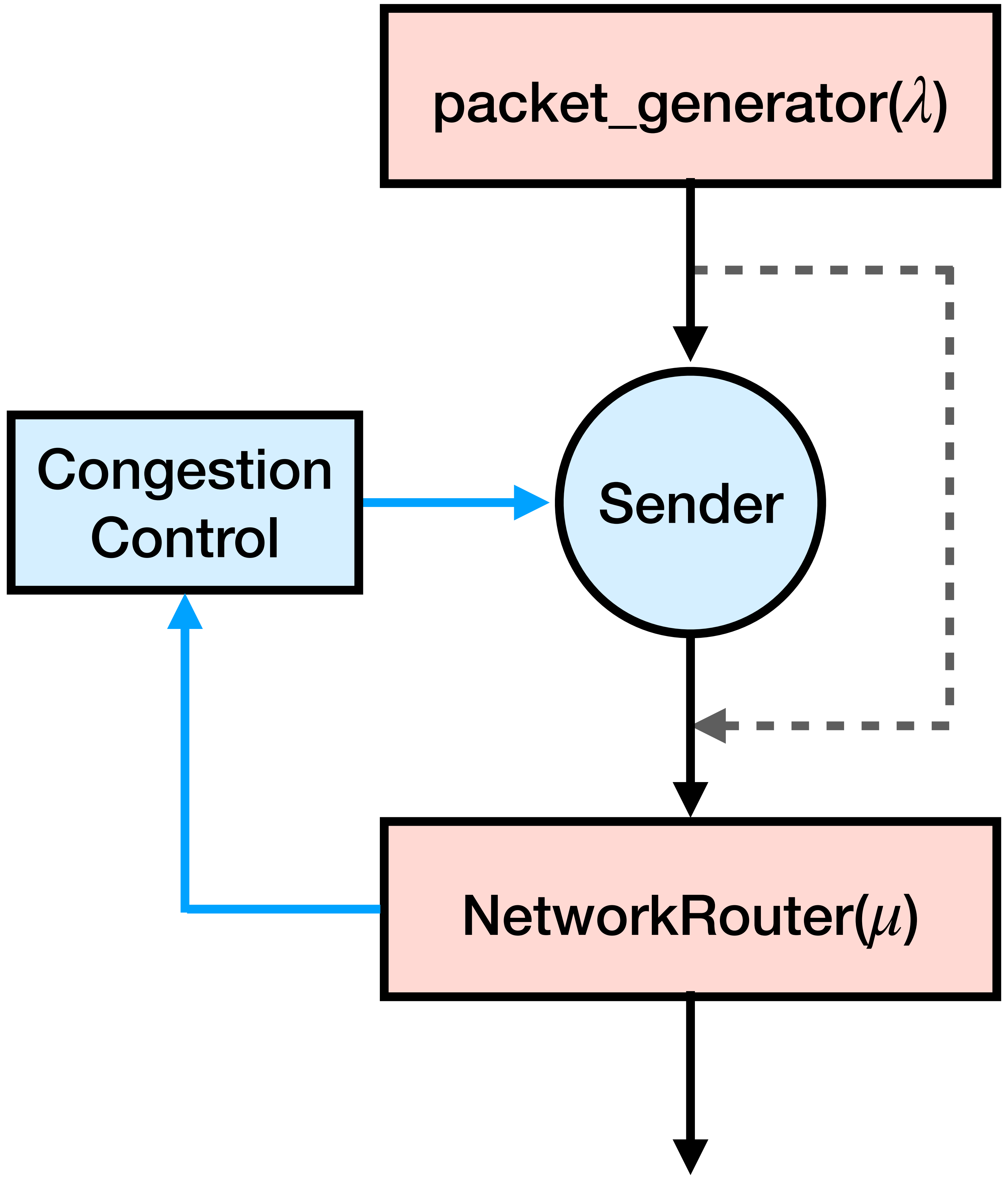


Figure 4. System Modeling Diagram

Experiment Design

- Variables:

λ := Arrival rate $\in [5,20]$

μ := Service rate $= 10$

K := Queue size $\in [1,10]$

θ := Deadline $\in [0.1,10]$

N := Packets sent $\in [50,500]$

CC := Congestion control algorithm

- Sweep one variable, hold rest constant
- Measure: throughput, average delay, loss rate
- All runs performed with same random seed

Results

Congestion Control Sample

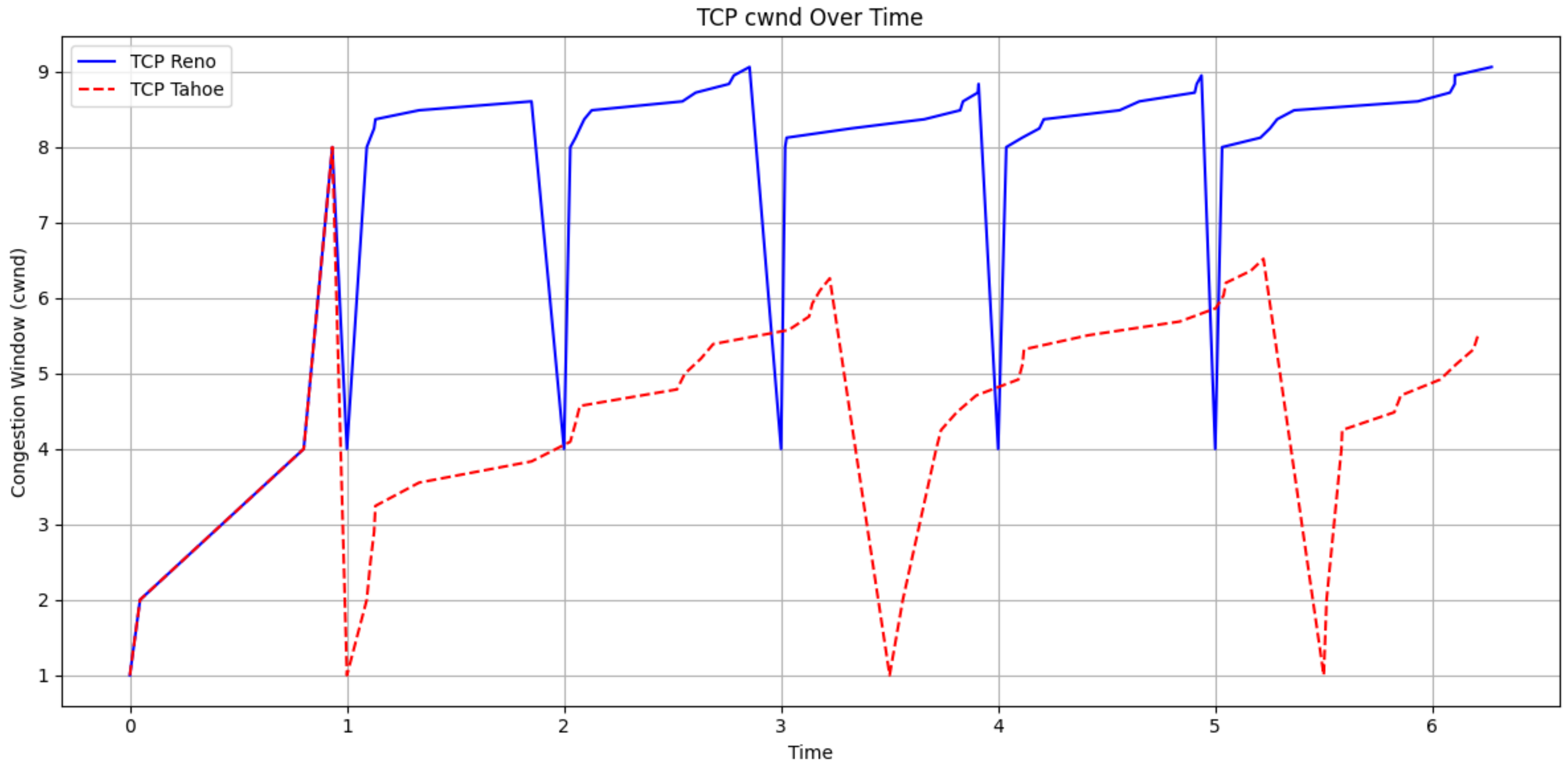


Figure 5. Sample `cwnd` vs time plot for TCP Tahoe and Reno

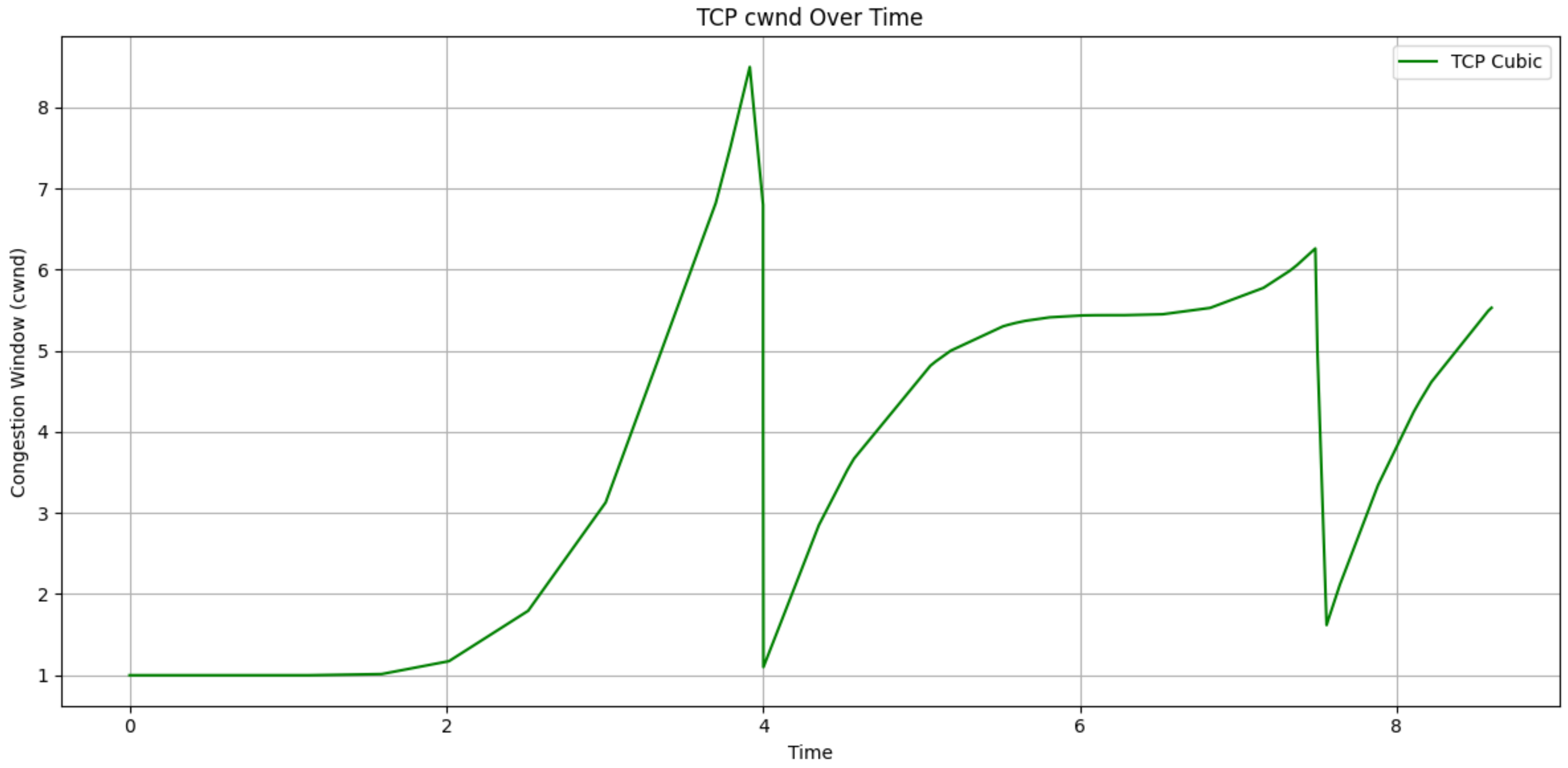


Figure 6. Sample `cwnd` vs time plot for TCP Cubic

Results - λ Sweep

Table 1. Loss Rate with Varying Arrival Rate

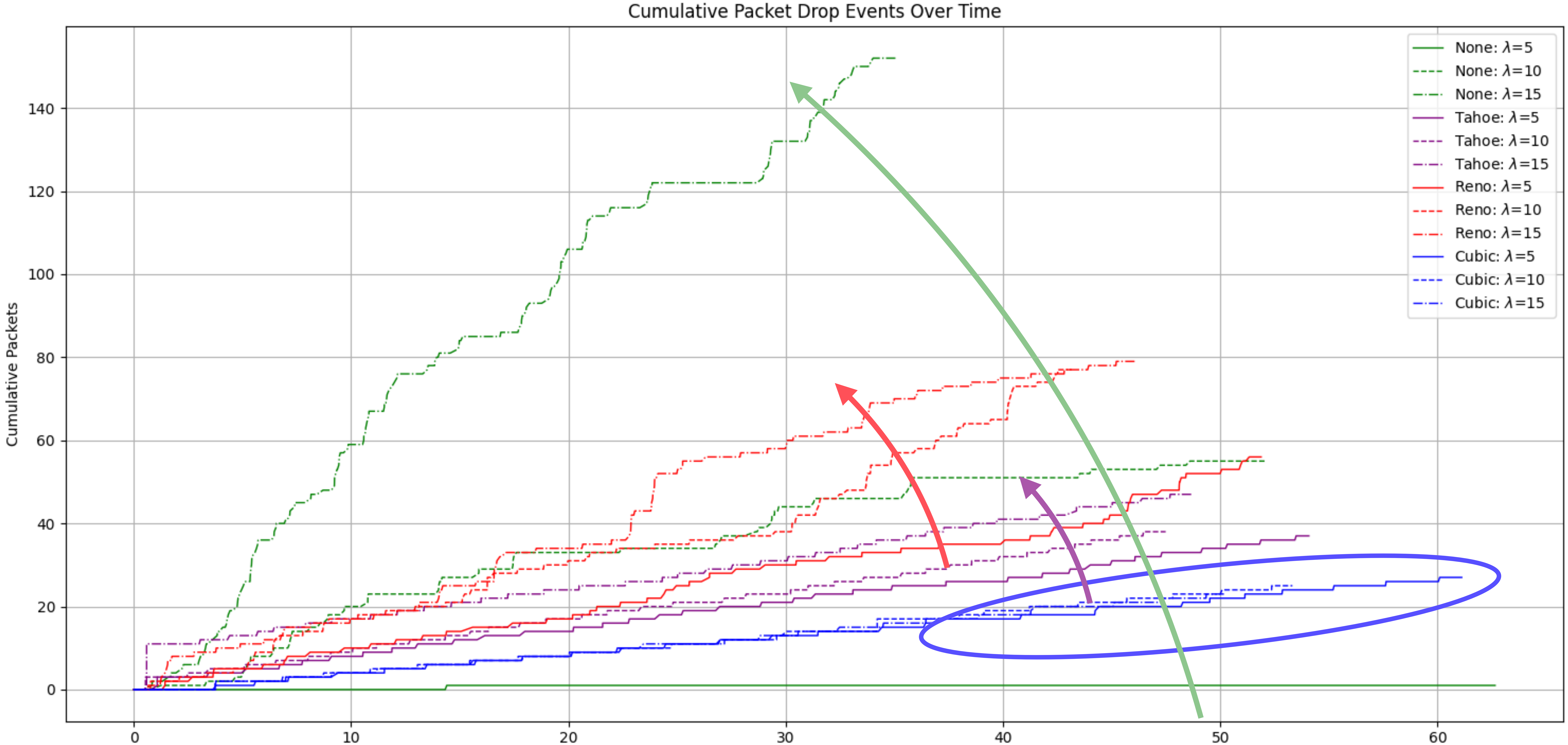
Loss Rate (%)	5	10	15
None	0.20	11.0	30.4
Tahoe	7.40	7.60	9.4
Reno	11.2	15.4	15.8
Cubic	5.40	5.00	4.60

Table 2. Throughput with Varying Arrival Rate

Throughput (P/sec)	5	10	15
None	4.793	8.467	9.838
Tahoe	8.429	9.589	9.289
Reno	8.485	9.652	9.109
Cubic	7.649	8.791	9.417

Constants: $\mu = 10, K = 5, \theta = 1$

Results - λ Sweep



Results - K Sweep

Table 1. Loss Rate with Varying Arrival Rate

Loss Rate (%)	1	5	10
None	60.0	30.4	24.8
Tahoe	49.8	9.40	6.00
Reno	84.4	15.8	5.80
Cubic	9.20	4.60	5.80

Table 2. Throughput with Varying Arrival Rate

Throughput (P/sec)	1	5	10
None	5.679	9.838	10.635
Tahoe	6.072	9.289	9.708
Reno	5.769	9.110	9.527
Cubic	5.882	9.417	8.694

Constants: $\mu = 10, \lambda = 15, \theta = 1$

Conclusion

- Queuing theory provides a powerful insight to understand the effects of TCP congestion control, especially through M/M/1 modeling
- TCP congestion control algorithms regulate arrival rate (λ) via `cwnd`, directly influencing queue dynamics (delay, throughput, and loss)
- Simulation results showed:
  - *TCP Tahoe*: conservative and avoids losses but may underutilize the network
  - *TCP Reno*: better throughput at the cost of higher loss under congestion
  - *TCP CUBIC*: most robust performance, maintaining low loss rates and high throughput even at high arrival rates and small buffer sizes.

Conclusion

- Future work:
  - Implement M/M/c modeling to simulate multi-server scenarios
  - Run true random runs, perform statistical analysis
  - Implement more TCP congestion control algorithms

References

[1] J. F. Kurose and K. W. Ross, “TCP Congestion Control,” in *Computer Networking; A Top Down Approach*, 8th ed. Boston, MA, USA: Pearson, 2021, pp. 263-279.

[2] V. Jacobson, “Congestion Avoidance and Control,” *Proc. 1988 ACM SIGCOMM Conference* (Stanford, CA, Aug. 1988), pp. 314–329.

[3] M. Allman, V. Paxson, W. Stevens, “TCP Congestion Control,” RFC 2581, Apr. 1999.

[4] Sangtae Ha, Injong Rhee, and Lisong Xu, "CUBIC: A New TCP-Friendly High-Speed TCP Variant," ACM SIGOPS Operating Systems Review, vol. 42, no. 5, pp. 64–74, July 2008.

[5] I. Rhee, L. Xu, S. Ha, A. Zimmermann, "CUBIC for Fast Long-Distance Networks," RFC 8312, Feb. 2018.

[6] L. Kleinrock, *Queueing Systems, Volume 1: Theory*, New York, NY, USA: Wiley-Interscience, 1975.

Questions?

Backup Slides

TCP CUBIC

- Congestion window function of time, defined as

$$W(t) = C(t - K)^3 + W_{max}$$

  - W_{max} := window size before last loss event
  - C := constant that determines aggressiveness of growth
  - K := time at which window size will reach W_{max} again
- K is computed such that

$$K = \sqrt[3]{\frac{W_{max} \cdot \beta}{C}}$$

  - β := multiplicative decrease factor
- For simulations: $C = 0.4, \beta = 0.2$

Constants: $\mu = 10, K = 5, \theta = 1$

Full Results - λ Sweep

Experiment	TCP Variant	λ (Arrival Rate)	Loss Rate (%)	Throughput	Avg Latency
E1.1.N	None	5	0.20%	4.7926	0.1421
E1.2.N	None	10	11.00%	8.4666	0.2492
E1.3.N	None	15	30.40%	9.8380	0.3336
E1.1.T	Tahoe	5	7.40%	8.4287	0.3417
E1.2.T	Tahoe	10	7.60%	9.5885	0.3495
E1.3.T	Tahoe	15	9.40%	9.2887	0.4055
E1.1.R	Reno	5	11.20%	8.4846	0.3423
E1.2.R	Reno	10	15.40%	9.6523	0.3617
E1.3.R	Reno	15	15.80%	9.1096	0.4348
E1.1.C	Cubic	5	5.40%	7.6487	0.3084
E1.2.C	Cubic	10	5.00%	8.7910	0.3424
E1.3.C	Cubic	15	4.60%	9.4172	0.3381

Constants: $\mu = 10, \lambda = 15, \theta = 1$

Full Results - K Sweep

Experiment	TCP Variant	Queue Size	Loss Rate (%)	Throughput	Avg Latency
E2.1.N	None	1	60.00%	5.6789	0.1024
E2.2.N	None	5	30.40%	9.8380	0.3336
E2.3.N	None	10	24.80%	10.6345	0.5872
E2.1.T	Tahoe	1	49.80%	6.0719	0.0963
E2.2.T	Tahoe	5	9.40%	9.2887	0.4055
E2.3.T	Tahoe	10	6.00%	9.7079	0.5409
E2.1.R	Reno	1	84.40%	5.7688	0.0957
E2.2.R	Reno	5	15.80%	9.1096	0.4348
E2.3.R	Reno	10	5.80%	9.5269	0.6602
E2.1.C	Cubic	1	9.20%	5.8817	0.0986
E2.2.C	Cubic	5	4.60%	9.4172	0.3381
E2.3.C	Cubic	10	5.80%	8.6935	0.4583

Constants: $\mu = 10, \lambda = 12, K = 5$

Full Results - θ Sweep

Experiment	TCP Variant	θ	Loss Rate (%)	Throughput	Avg Latency
E3.1.N	None	0.1	11.00%	8.4666	0.2492
E3.2.N	None	1	11.00%	8.4666	0.2492
E3.3.N	None	10	11.00%	8.4666	0.2492
E3.1.T	Tahoe	0.1	7.60%	9.5885	0.3495
E3.2.T	Tahoe	1	7.60%	9.5885	0.3495
E3.3.T	Tahoe	10	7.60%	9.5885	0.3495
E3.1.R	Reno	0.1	15.40%	9.6523	0.3617
E3.2.R	Reno	1	15.40%	9.6523	0.3617
E3.3.R	Reno	10	15.40%	9.6523	0.3617
E3.1.C	Cubic	0.1	5.00%	8.7910	0.3424
E3.2.C	Cubic	1	5.00%	8.7910	0.3424
E3.3.C	Cubic	10	5.00%	8.7910	0.3424

Full Results - N Sweep

Constants: $\mu = 10, \lambda = 10, K = 5, \theta = 1$

Experiment	TCP Variant	N	Loss Rate (%)	Throughput	Avg Latency
E4.1.N	None	50	6.00%	7.8196	0.2322
E4.2.N	None	300	14.67%	7.9411	0.3060
E4.3.N	None	1000	16.40%	8.1908	0.2975
E4.1.T	Tahoe	50	10.00%	7.3757	0.3213
E4.2.T	Tahoe	300	8.00%	8.8071	0.3743
E4.3.T	Tahoe	1000	7.60%	9.3578	0.3636
E4.1.R	Reno	50	10.00%	7.7302	0.3675
E4.2.R	Reno	300	12.33%	8.6186	0.4079
E4.3.R	Reno	1000	16.00%	9.3526	0.3862
E4.1.C	Cubic	50	4.00%	7.0840	0.2064
E4.2.C	Cubic	300	4.67%	8.2553	0.3378
E4.3.C	Cubic	1000	4.90%	8.8385	0.3412

Constants: $\mu = 10, \lambda = 15, K = 5, \theta = 1$

Full Results - $\theta \sim \text{Exp}(\theta)$

Experiment	TCP Variant	is_exp_drop	Loss Rate (%)	Throughput	Avg Latency
E5.1.N	None	TRUE	39.80%	8.7580	0.3085
E5.2.N	None	FALSE	30.40%	9.8380	0.3336
E5.1.T	Tahoe	TRUE	9.40%	9.2887	0.4055
E5.2.T	Tahoe	FALSE	9.40%	9.2887	0.4055
E5.1.R	Reno	TRUE	15.80%	9.1096	0.4348
E5.2.R	Reno	FALSE	15.80%	9.1096	0.4348
E5.1.C	Cubic	TRUE	4.60%	9.4172	0.3381
E5.2.C	Cubic	FALSE	4.60%	9.4172	0.3381