

REVIEW OF QUEUEING THEORY

Notes prepared for EE 6345

by

Professor Cyrus D. Cantrell

May–July 2002

WHY LEARN QUEUEING ANALYSIS?

- Networks, and computers running multiuser, multitasking operating systems, can be viewed as interconnected queueing systems
 - ▷ Uses of queueing analysis:
 - Analyze and understand system behavior after the fact (*i.e.*, using real data),
 - Project from an existing system to a future system
 - Develop an analytic model for use in designing a system
 - Create a simulation that models a system
 - ▷ Queueing theory can be used to analyze the performance of:
 - Computer systems
 - Networks
 - Medical facilities, transportation systems, etc. ...

THE DOCTOR'S FALLACY

- It's "reasonable" to assume that if the mean service time in a doctor's office is T_s minutes, then the office can handle an arrival rate of

$$\lambda_{\max} = \frac{1}{T_s}$$

patients per minute

- Unfortunately, accepting λ_{\max} patients per minute leads to a **saturated system**

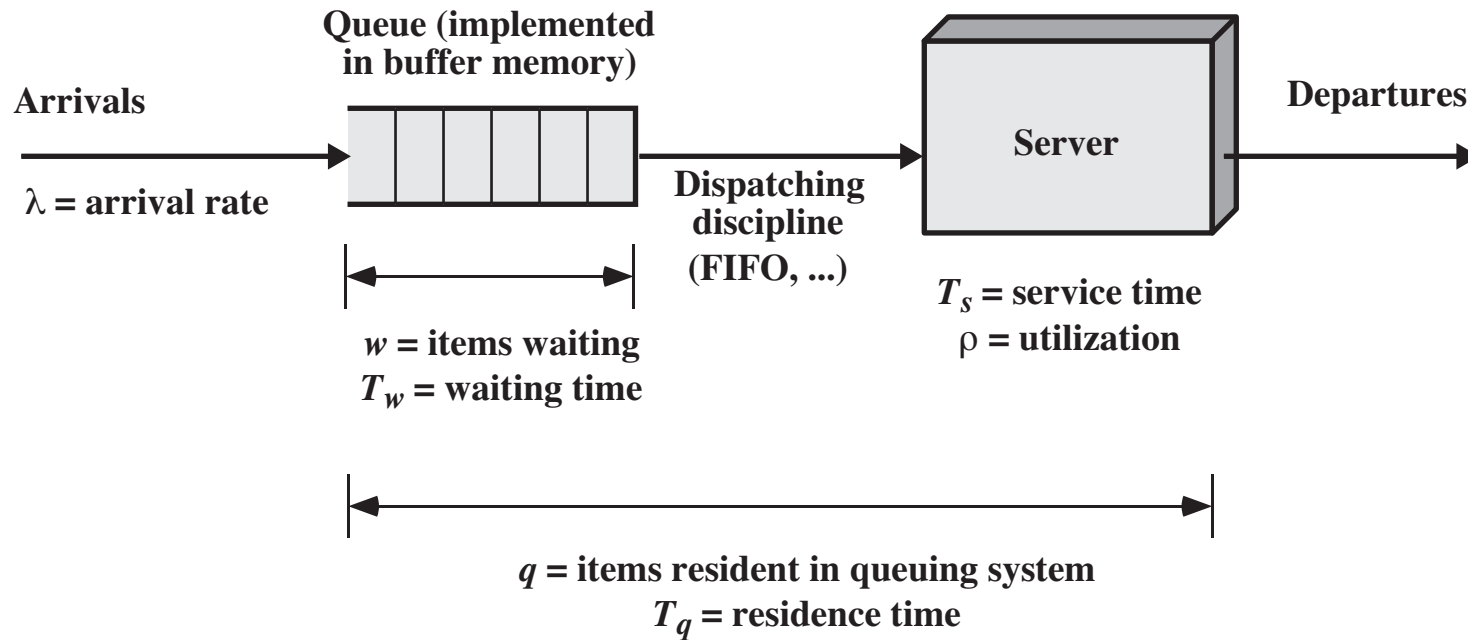
- ▷ The utilization $\rho = \lambda T_s = \frac{\lambda}{\lambda_{\max}} = 1$

- ▷ The mean waiting time $T_w = \infty$

- ▷ Patients get *very* unhappy...

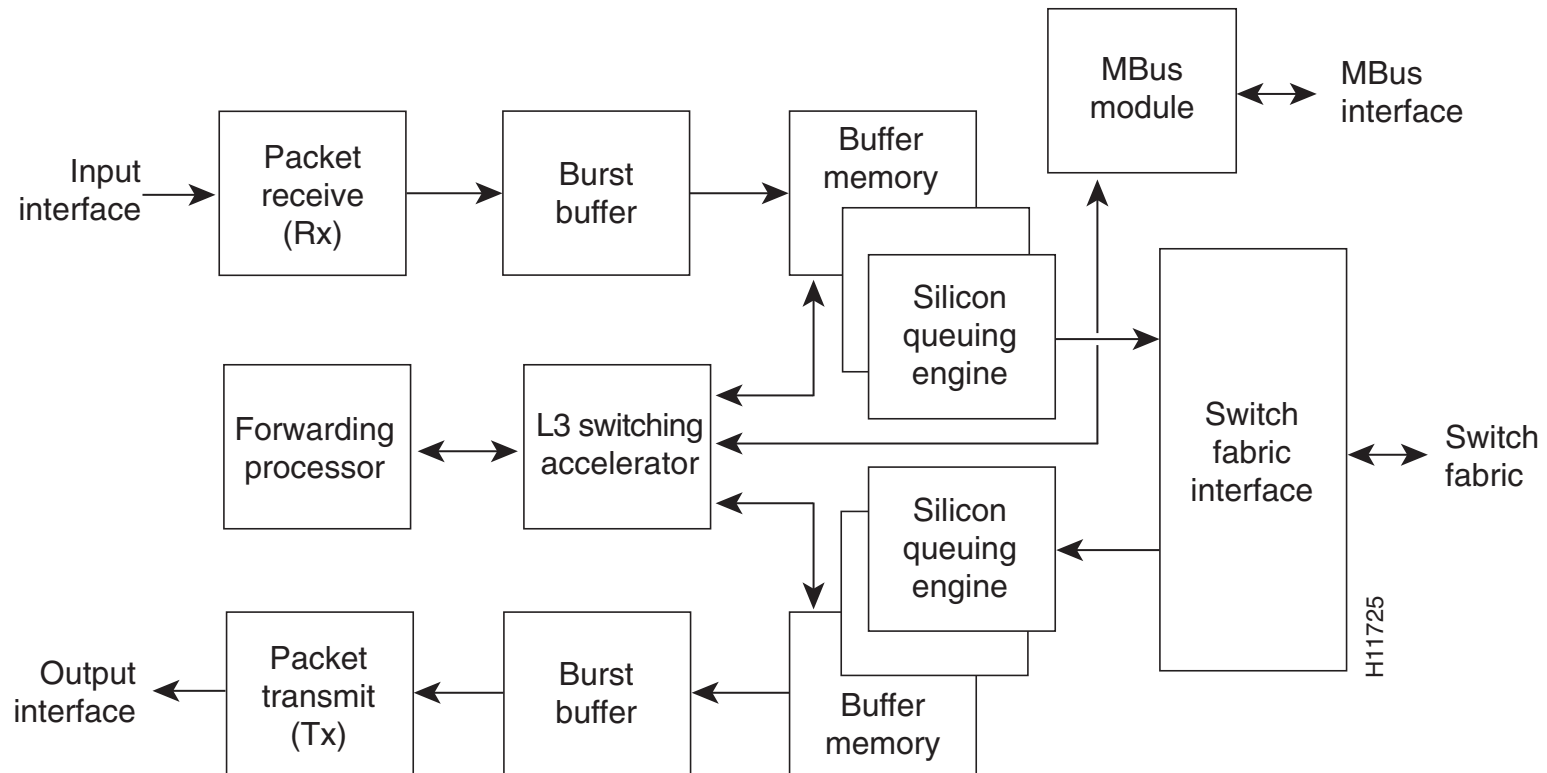
- **Near saturation, the waiting time changes exponentially in response to changes in utilization**

A SINGLE-SERVER QUEUE



QUEUEING IN CISCO OC-48 LINE CARD

Block Diagram of Cisco's OC-48c/STM-16c POS Line Card



BASIC QUEUEING RELATIONS

● General:

▷ Little's formula:

$$q = \lambda T_q$$

$$w = \lambda T_w$$

▷ Time:

$$T_q = T_w + T_s$$

● Single server

▷ Utilization:

$$\rho = \lambda T_s$$

▷ Mean number of items:

$$q = w + \rho$$

● Multiple servers

▷ Utilization:

$$\rho = \frac{\lambda T_s}{N}$$

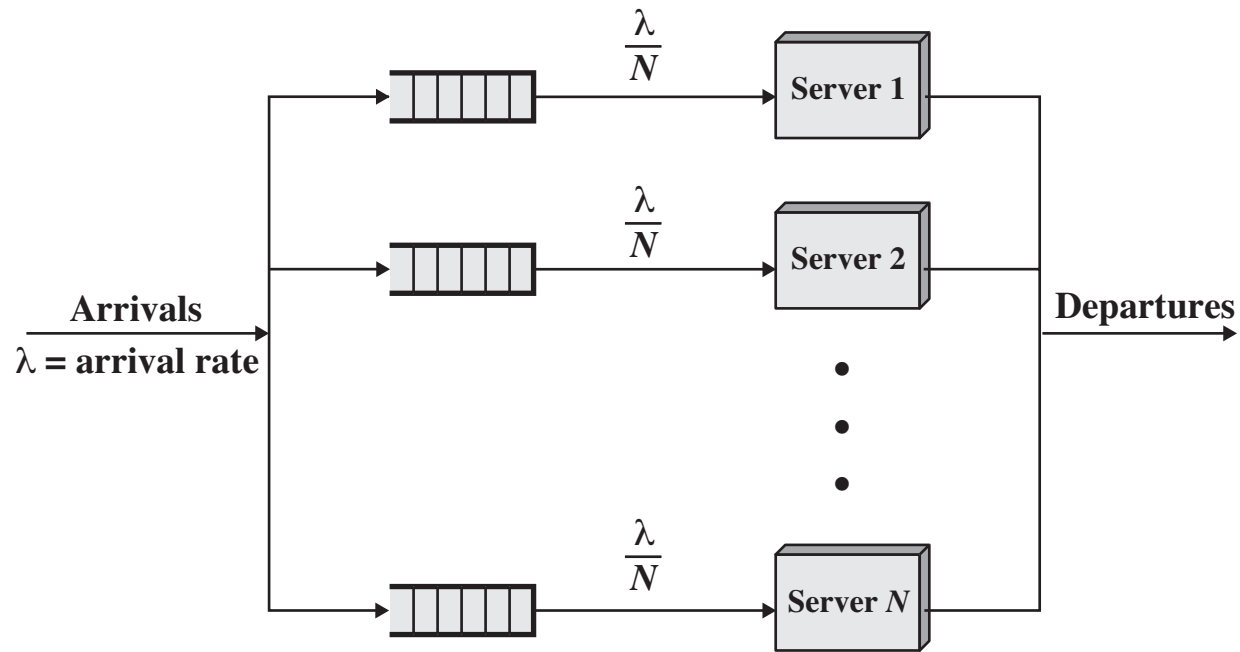
▷ Mean number of items:

$$q = w + N\rho$$

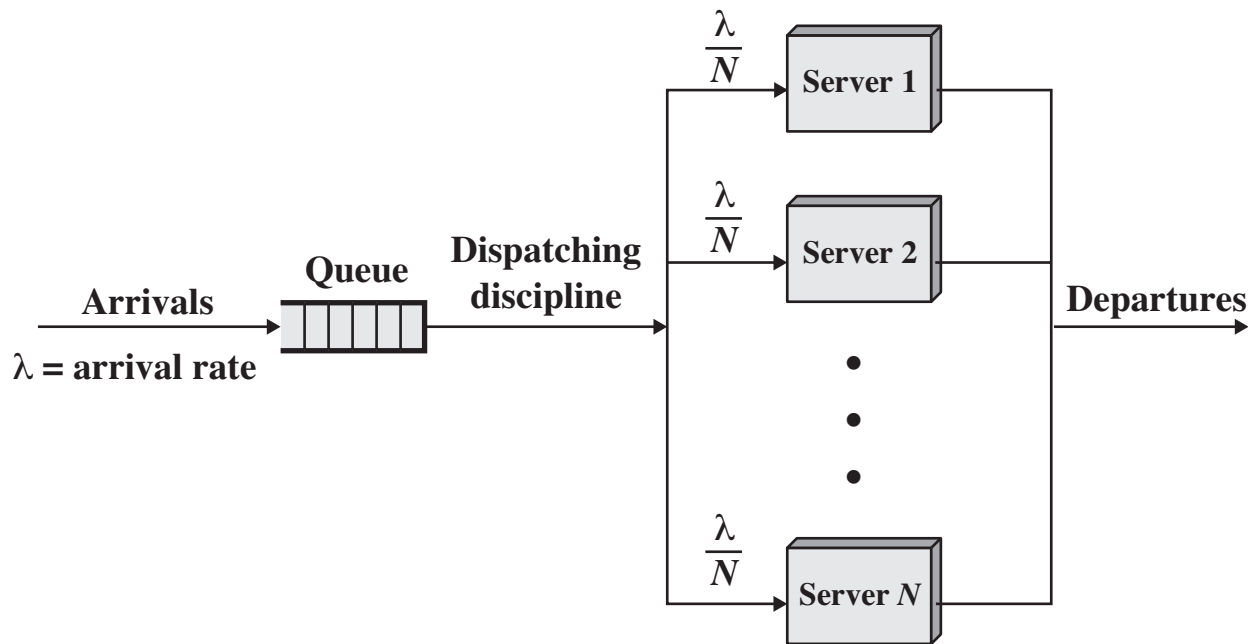
▷ Traffic intensity:

$$u = \lambda T_s = \rho N$$

MULTIPLE SINGLE-SERVER QUEUES



A MULTIPLE-SERVER QUEUE



ASSUMPTIONS COMMONLY USED IN QUEUEING MODELS

- Item population
 - ▷ There is an infinite supply of items (mean arrival rate is constant)
- Infinite queue size
 - ▷ Real queues are finite because they are implemented in buffer memory
- Dispatching discipline (how to decide which item to take care of)
 - ▷ Most common is FIFO
- Arrival-time-interval and service-time statistics
 - ▷ G – general independent arrivals or service times
 - ▷ M – (negative) exponential distribution
 - Equivalent to Poisson distribution of arrivals in a fixed time T
 - ▷ D – deterministic arrivals or constant service time

NOTATION FOR ARRIVAL AND SERVICE STATISTICS

- The notation for a queueing model is $X/Y/N$, where
 - ▷ X denotes the arrival-time-interval distribution
 - ▷ Y denotes the service-time distribution
 - ▷ N denotes the number of servers
- Example:
 - ▷ $M/M/1$ denotes a 1-server queueing model with Poisson arrival statistics and an exponential distribution of service times
- The distribution of voice traffic arrival times is close to exponential
- The distribution of arrival times in a data network is non-exponential
 - ▷ Traffic is bursty
 - ▷ Actual distributions are heavy-tailed and self-similar (more on this later)
- **Voice traffic models are not valid for data networks**

POISSON DISTRIBUTION (1)

- Probability of n events in time T , assuming a constant arrival rate λ and independent arrivals:

$$p_n = \lim_{N \rightarrow \infty} \binom{N}{n} \left(1 - \frac{\lambda T}{N}\right)^{N-n} \left(\frac{\lambda T}{N}\right)^n = \frac{e^{-\lambda T}}{n!} (\lambda T)^n$$

(probability of no events in $N - n$ time intervals of length T/N , times probability of one event in each of n intervals)

- Time interval distribution (probability of zero events in time T) is exponential:

$$p_0 = e^{-\lambda T}$$

POISSON DISTRIBUTION (2)

- Generating function:

$$\mathcal{G}(s) = \sum_{n=0}^{\infty} p_n (1-s)^n = e^{-\lambda T} \sum_{n=0}^{\infty} \frac{(\lambda T)^n}{n!} (1-s)^n = e^{-\lambda T s}$$

- ▷ Factorial moments (easier to calculate than the usual moments):

$$E[n(n-1)\cdots(n-m+1)] = (-1)^m \left. \frac{d^m \mathcal{G}}{ds^m} \right|_{s=0} = (\lambda T)^m$$

- ▷ Mean:

$$E[n] = \lambda T$$

- ▷ Variance:

$$\sigma^2 = E[n^2] - (E[n])^2 = E[n(n-1)] + E[n] - (E[n])^2 = \lambda T$$

FORMULAS FOR $M/M/1$ QUEUES

- Mean number of items in system or waiting:

$$q = \frac{\rho}{1 - \rho} \qquad w = \frac{\rho^2}{1 - \rho}$$

- Time in queue or waiting time:

$$T_q = \frac{T_s}{1 - \rho} \qquad T_w = \frac{\rho T_s}{1 - \rho}$$

- Standard deviations:

$$\sigma_q = \frac{\sqrt{\rho}}{1 - \rho} \qquad \sigma_{T_q} = \frac{T_s}{1 - \rho} \qquad \sigma_{T_s} = T_s$$

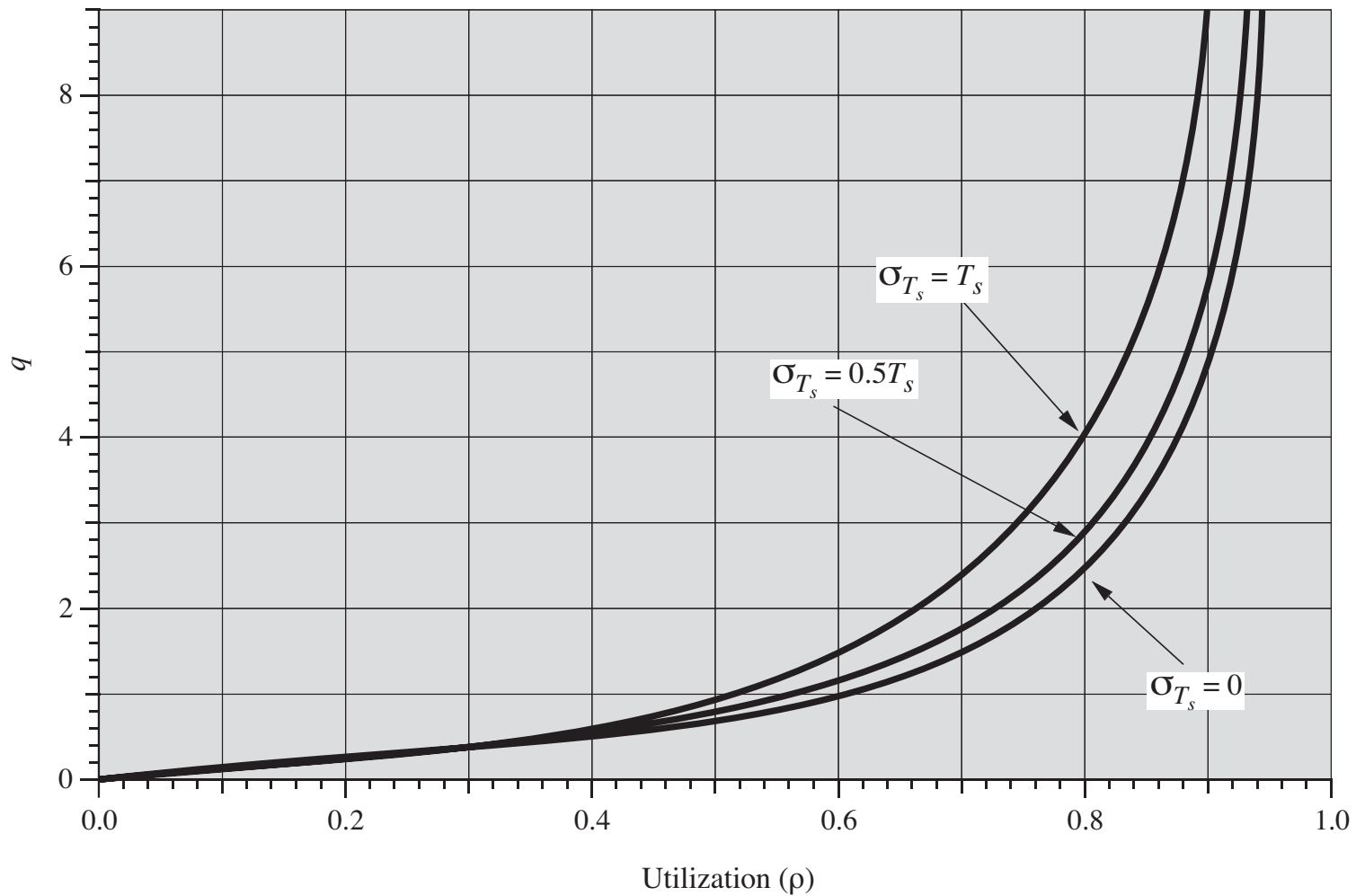
- Probability that the length of the queue, Q , is exactly L :

$$\Pr(Q = L) = (1 - \rho)\rho^L$$

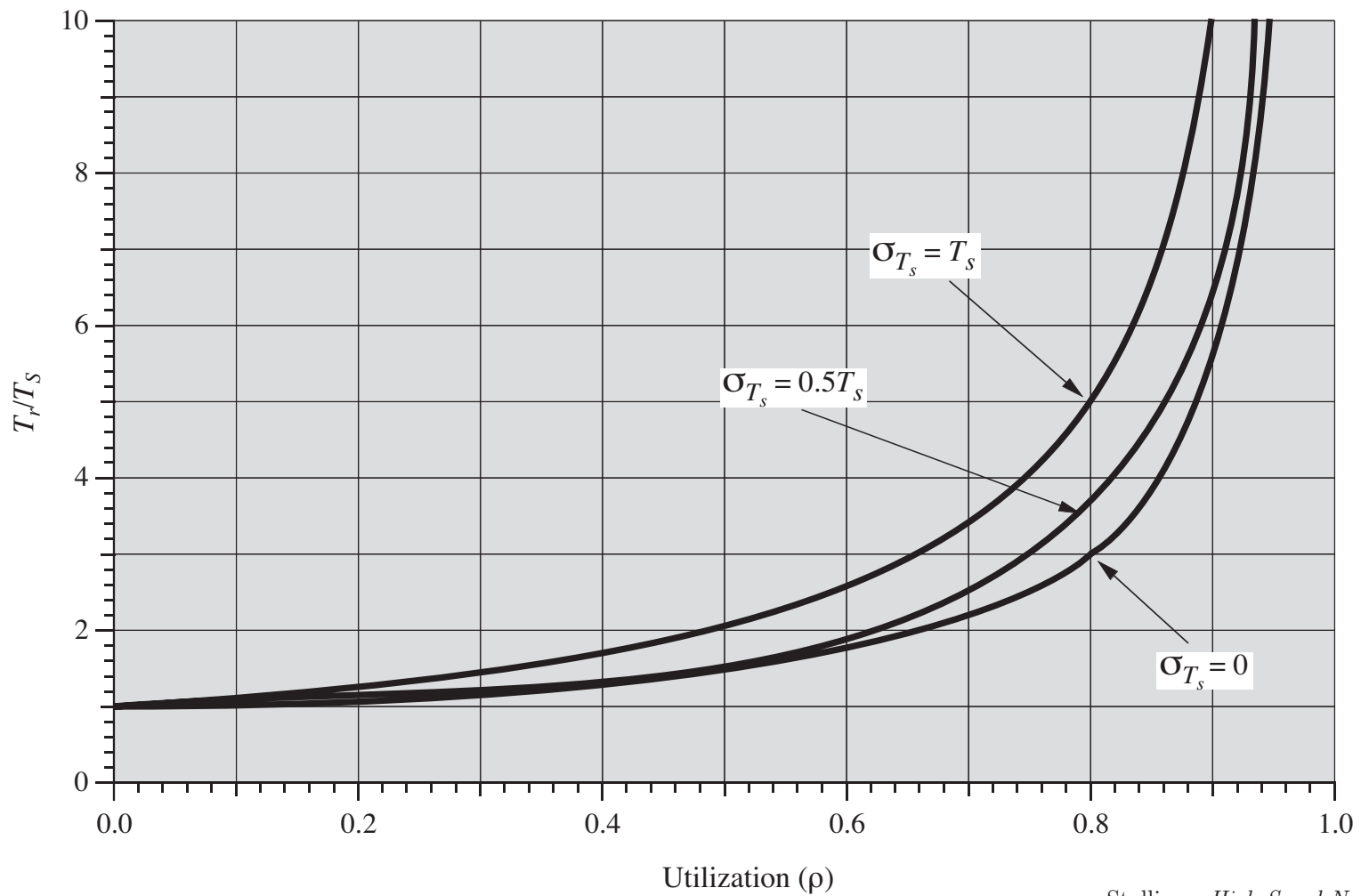
- Probability that Q is greater than the buffer length L (\Rightarrow items dropped):

$$\Pr(Q > L) = (1 - \rho) \sum_{l=L+1}^{\infty} \rho^l = \rho^{L+1}$$

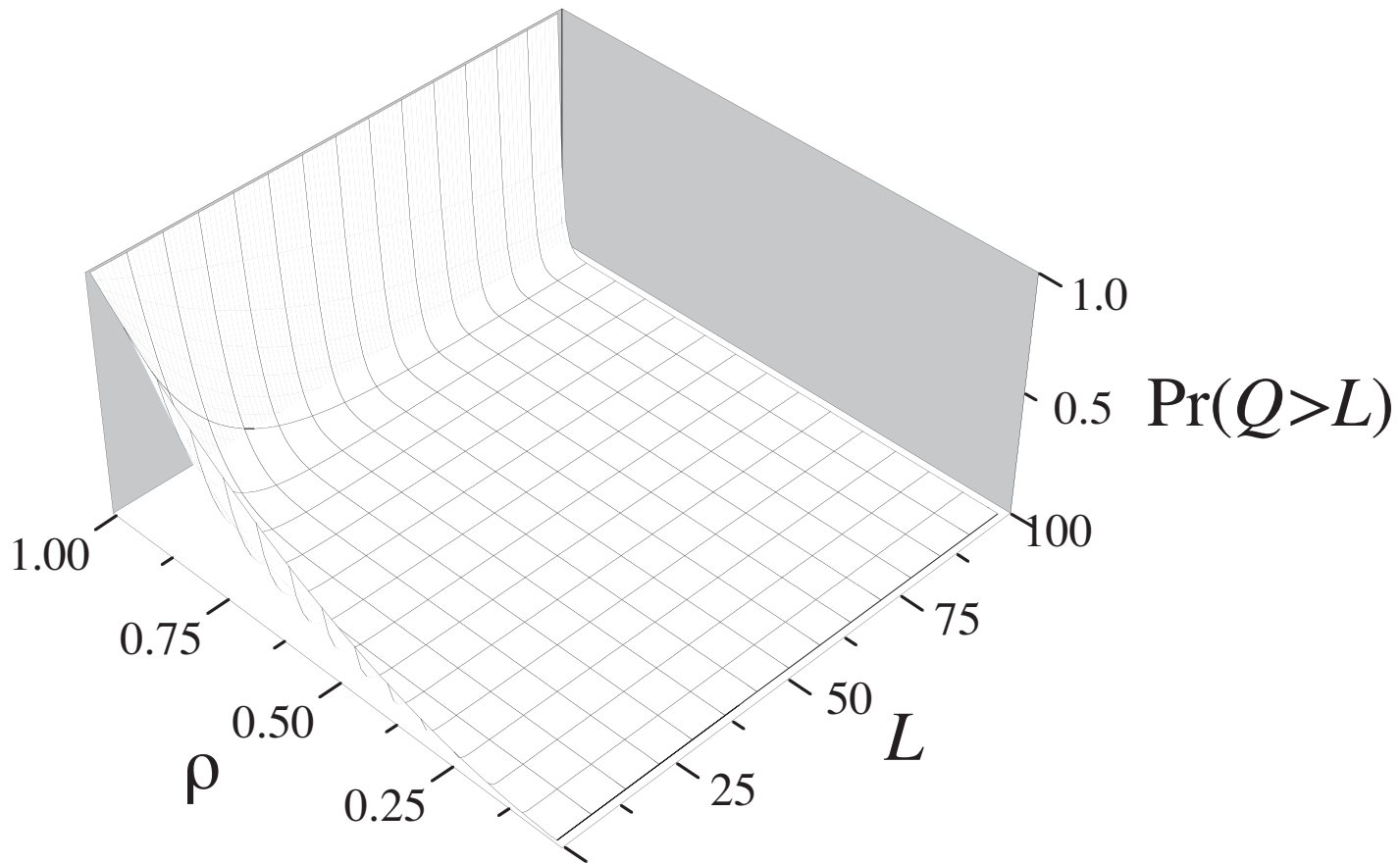
MEAN QUEUE SIZE IN A SINGLE-SERVER QUEUE



MEAN QUEUEING TIME IN A SINGLE-SERVER QUEUE



BUFFER SPILL IN AN $M/M/1$ QUEUE



FORMULAS FOR $M/D/1$ QUEUES

- Mean number of items in system or waiting:

$$q = \frac{\rho^2}{2(1-\rho)} + \rho \qquad w = \frac{\rho^2}{2(1-\rho)}$$

- Mean time in queue or waiting time:

$$T_q = \frac{T_s(2-\rho)}{2(1-\rho)} \qquad T_w = \frac{\rho T_s}{2(1-\rho)}$$

- Standard deviations:

$$\sigma_q = \frac{1}{1-\rho} \sqrt{\rho - \frac{3\rho^2}{2} + \frac{5\rho^3}{6} - \frac{\rho^4}{12}} \qquad \sigma_{T_q} = \frac{T_s}{1-\rho} \sqrt{\frac{\rho}{3} - \frac{\rho^2}{12}}$$

FORMULAS FOR $M/M/N$ QUEUES

- Mean number of items in system or waiting:

$$q = C(N, u) \frac{\rho}{1 - \rho} + N\rho \qquad w = C(N, u) \frac{\rho}{1 - \rho}$$

$C(N, u)$ is the Erlang C function (see next slide); $u = N\rho$

- Time in queue or waiting time:

$$T_q = \frac{C(N, u)}{N} \frac{T_s}{1 - \rho} + T_s \qquad T_w = \frac{C(N, u)}{N} \frac{T_s}{1 - \rho}$$

- Standard deviations:

$$\sigma_{T_q} = \frac{T_s}{N(1 - \rho)} \sqrt{C(2 - C) + N^2(1 - \rho)^2}$$

$$\sigma_w = \frac{1}{1 - \rho} \sqrt{C\rho(1 + \rho - C\rho)}$$

ERLANG C FUNCTION

- $C(N, u)$ is the probability that all N servers are busy in a queueing system with traffic intensity $u = \lambda T_s$

- Formula:

$$C(N, u) = \frac{1 - K(u)}{1 - \rho K(u)} \quad \text{where} \quad \rho = \frac{u}{N}$$

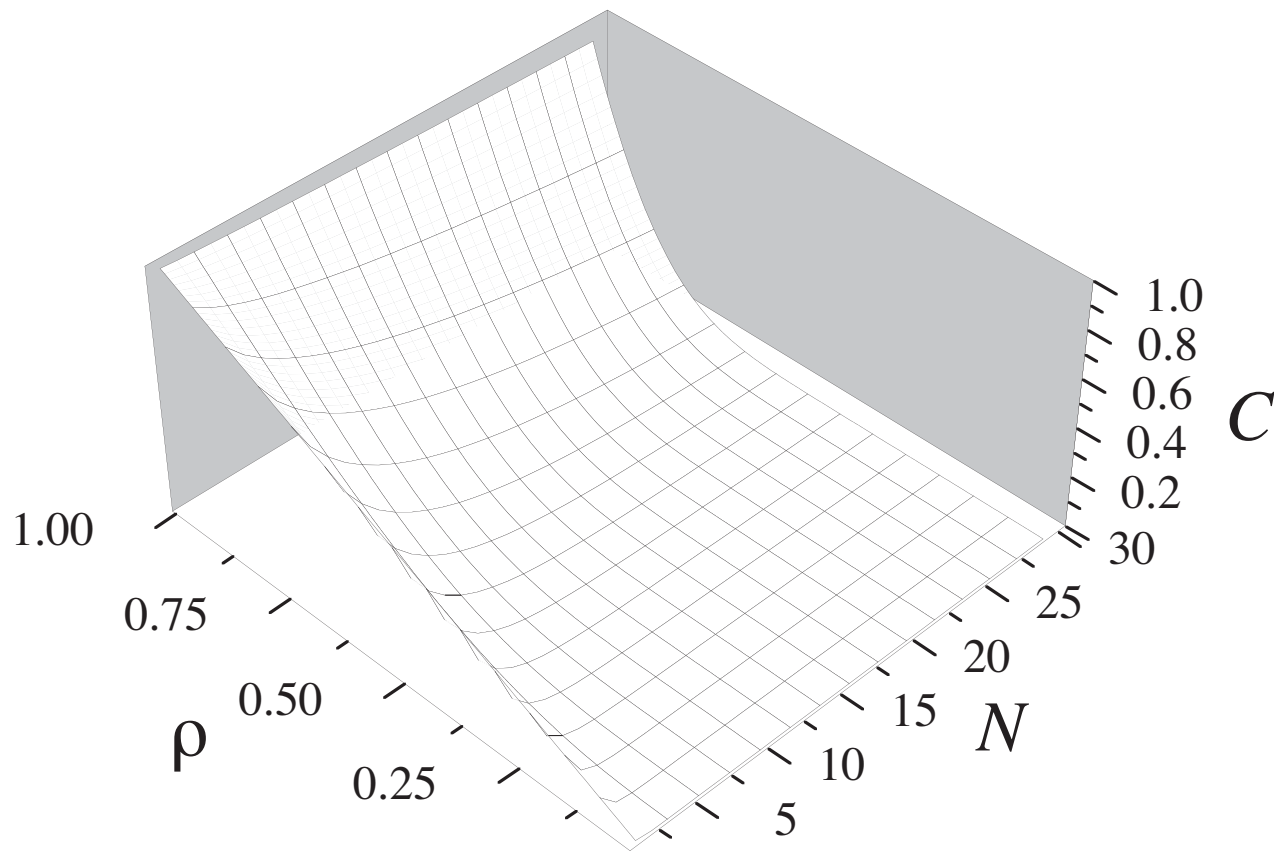
and where K is the Poisson ratio function,

$$K(u) = \frac{\sum_{l=0}^{N-1} \frac{u^l}{l!}}{\sum_{l=0}^N \frac{u^l}{l!}}$$

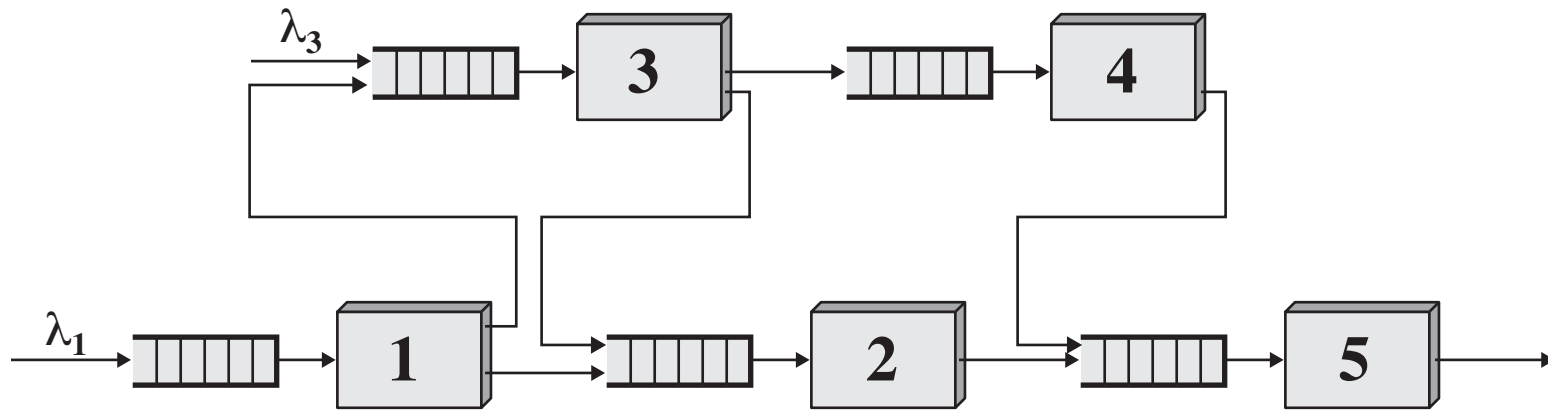
- Single-server queue:

$$C(1, u) = \rho$$

PLOT OF ERLANG C FUNCTION



EXAMPLE OF A NETWORK OF QUEUES



QUEUEING MODELS AND SIMULATIONS

- Professor Ann-Marie Martensson-Pendrill gives a Matlab program to compute the Erlang C function, along with many examples of values of C , time in queue, etc.
- Simulations are useful if one wants to escape from unrealistic assumptions (such as infinite buffer length) or use non-exponential arrival or service time distributions
 - ▷ A standard tool for academic research is *ns*, which was developed at the Lawrence Berkeley Laboratory, based on S. Keshav's **REAL Network Simulator**
 - ▷ Professor Ivor Page's Java applet simulates an $M/M/N$ queue, with adjustable parameters