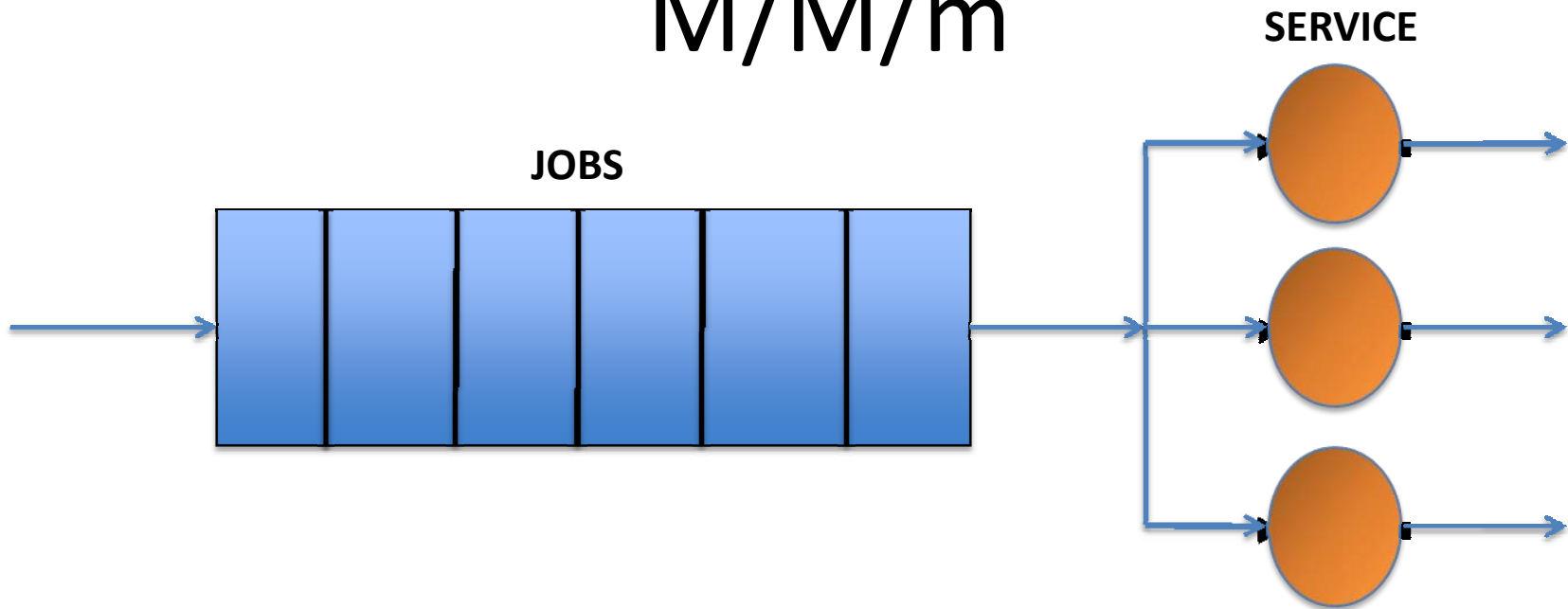


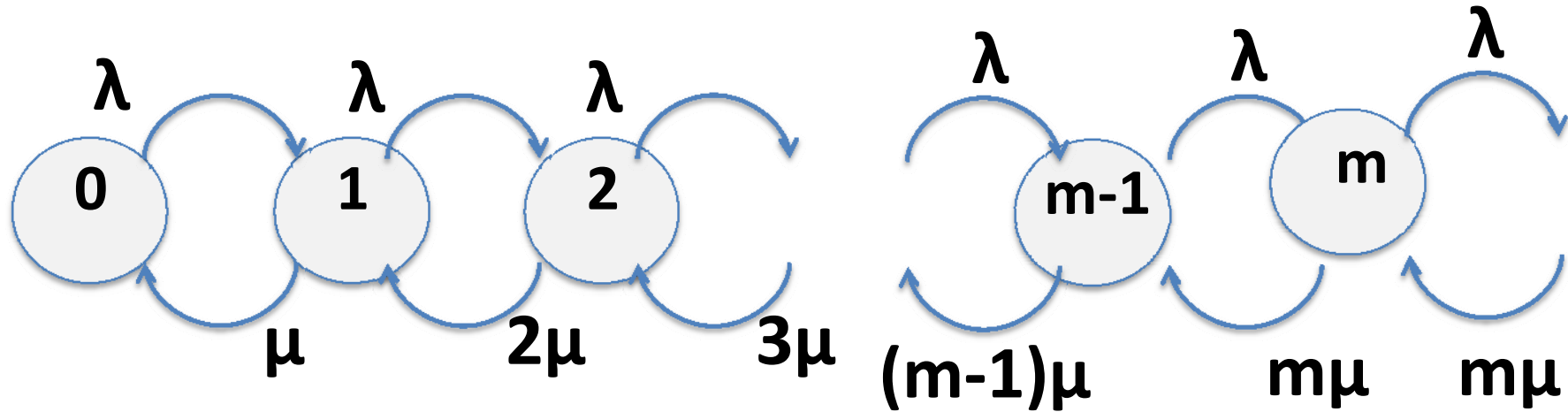
M/M/m and M/M/m/B

M/M/m



- Each server serves μ jobs per unit of time
- Jobs get service right away if less than m jobs in system, otherwise they wait in queue.

Probabilities in M/M/m



- Number of jobs in a M/M/m system is a birth death process. Hence:

$$P_n = \frac{\lambda_0 \lambda_1 \dots \lambda_{n-1}}{\mu_1 \mu_2 \dots \mu_n} P_0$$

Resolving for M/M/m

$$P_n = \frac{\lambda^n}{n! \mu^n} P_0 \quad \text{for } n = 0, 1, 2, \dots, m-1$$

$$P_n = \frac{\lambda^n}{m! m^{n-m} \mu^n} P_0 \quad \text{for } n = m, m+1, \dots, \infty$$

With traffic intensity

- $\rho = \lambda / (m\mu)$

$$P_n = \frac{(m\rho)^n}{n!} P_0 \text{ for } n = 0, 1, 2, \dots, m-1$$

$$P_n = \frac{\rho^n m^m}{m!} P_0 \text{ for } n = m, m+1, \dots, \infty$$

M/M/m queue

- The rest of the results for this system can be derived from the state probabilities (see in the text book)

m times M/M/1 vs M/M/m

- What is better?
 - m queues of the form M/M/1 with arrival rate λ/m
 - a single system of the form M/M/m with arrival rate λ
- In general, M/M/m will be better because it leads to less waiting time (jobs waiting in a queue do not benefit if a server in another queue is free)

M/M/m/B

- The queuing system has limited buffer capacity. After B buffers are full, jobs are no longer admitted
- State transition diagram is similar to that of M/M/m but it finishes with B (as opposed to having ∞ states)
- As before

$$P_n = \frac{\lambda_0 \lambda_1 \dots \lambda_{n-1}}{\mu_1 \mu_2 \dots \mu_n} P_0$$

State probabilities

$$P_n = \frac{\lambda^n}{n! \mu^n} P_0 = \frac{(m\rho)^n}{n!} P_0$$

for $n < m$

$$P_n = \frac{\lambda^n}{m! m^{n-m} \mu^n} P_0 = \frac{\rho^n m^m}{m!} P_0$$

for $n = m, m+1, \dots, B$

M/M/m/B

- All the other parameters can be computed from these probabilities (see the textbook)
- Effective arrival rate:
 - Arrival rate λ
 - After B jobs, no more jobs enter the system
 - $\lambda' = \lambda (1-p_B)$ effective arrival rate
 - $\lambda - \lambda' = \lambda p_B$ packet loss rate
- Apply effective arrival rate to Little's Law

Operational Laws

Operational laws

- Operational laws are relationships that apply to certain measurable parameters in a computer system
- They are independent of the distribution of arrival times or service rates
- The parameters are observed for a finite time T_i , yielding operational quantities:
 - Number of arrivals A_i
 - Number of completions C_i
 - Busy time B_i

Job flow balance

- The job flow balance assumption implies:
- Number of arrivals A_i (jobs arriving during T_i)
- Number of completions C_i (jobs completed during T_i)

$$A_i = C_i$$

- Correct use heavily depends on semantics of “jobs completed”

Derivations

- Arrival rate $\lambda_i = A_i / T_i$
- Throughput $X_i = C_i / T_i$
- Utilization $U_i = B_i / T_i$
- Mean Service Time $S_i = B_i / C_i$
- These are variables that change from observational period to observational period
- Some relationships hold for each observational period

Utilization Law

$$U_i = \frac{B_i}{T_i} = \frac{C_i}{T_i} \times \frac{B_i}{C_i} = X_i \cdot S_i$$

The utilization law is simply based on the observation that the service time indicates how long the server needs to complete a job. Hence, the utilization of the server is the number of jobs completed multiplied the time that it takes to complete each one of them

Example Utilization Law

- A disk serves 40 requests per second. Each request requires 0.0225 seconds
- Utilization: $40 \times 0.0225 = 90\%$
- Instrumentation shows that a disk is busy 80% of the time when an application runs. We know that the application produces 20 requests per second
- The time taken per request is 40 ms

Forced Flow Law

- Assume a closed system several devices are connected as a queuing network. The number of completed jobs is C_0
- Assume job flow balance
- If each job makes V_i (visit ratio) requests of the i_{th} device, then $C_i = C_0 V_i$

$$X_i = \frac{C_i}{T_i} = \frac{C_i}{C_0} \times \frac{C_0}{T_i} = X V_i$$

Bottleneck device

- Utilization of the device

$$U_i = X_i S_i = X V_i S_i$$

With the demand for the device $D_i = V_i S_i$

$$U_i = X D_i$$

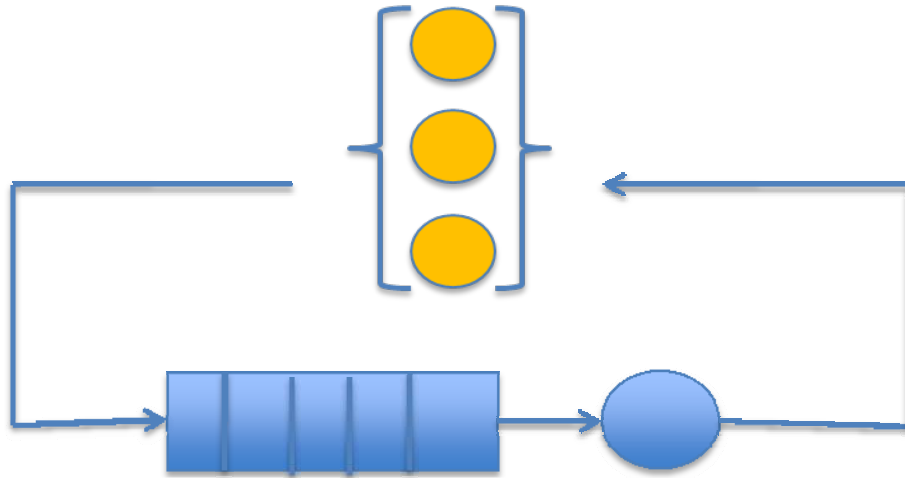
The device with the highest demand is the bottleneck device

Little's Law

- Assuming job flow balance:
- R_i is the response time of the device
- Q_i is the number of jobs in the device
- $\lambda_i = X_i$ (arrival rate equals throughput)

$$Q_i = \lambda_i R_i = X_i R_i$$

Interactive Response Time Law



- Users submit a request, when they get a response, they think for a time Z , and submit the next request

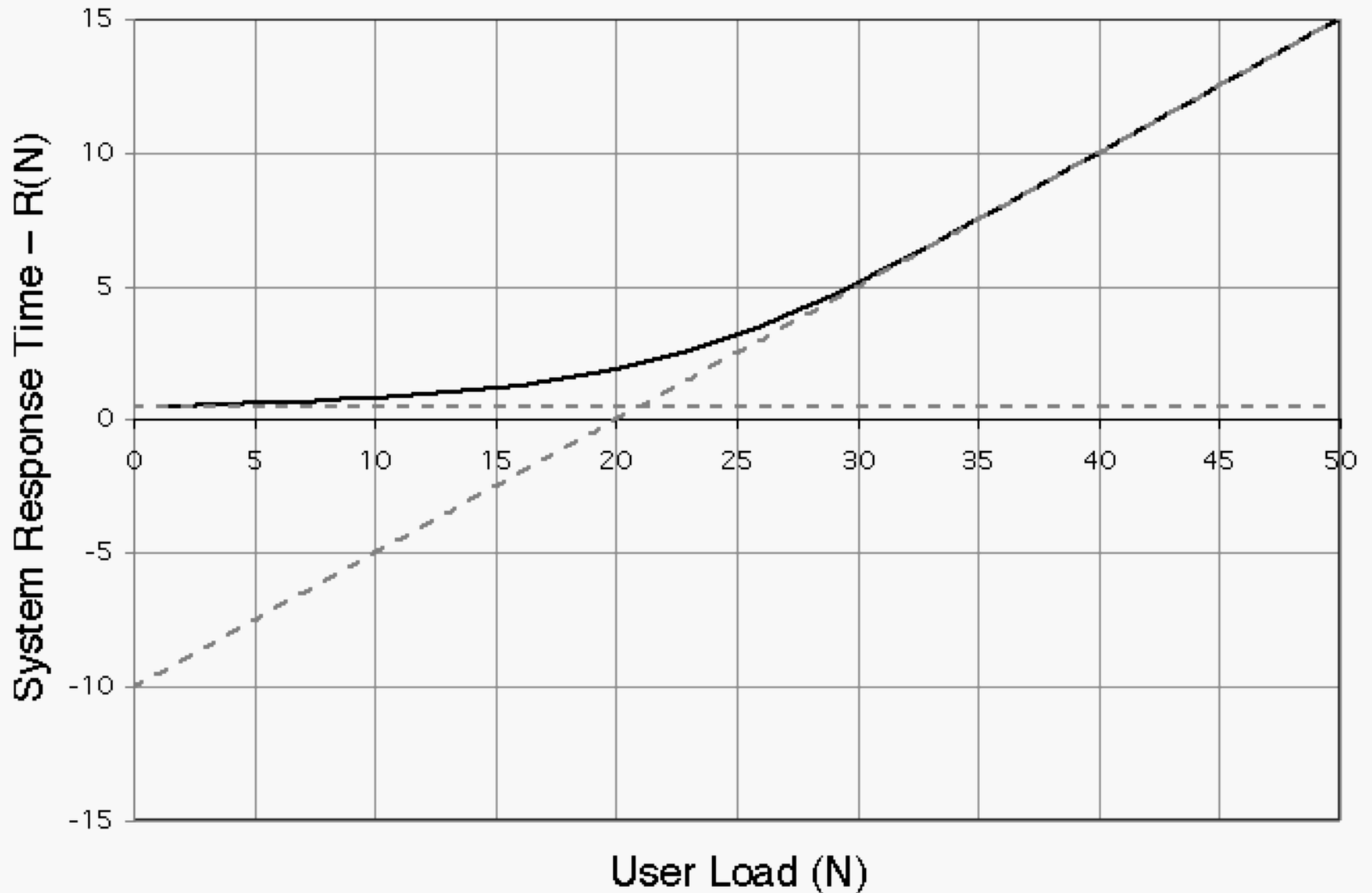
Interactive, closed systems

- Response time R
- Total cycle time $R+Z$
- Each user generates $T/(R+Z)$ requests in time T
- There are N users

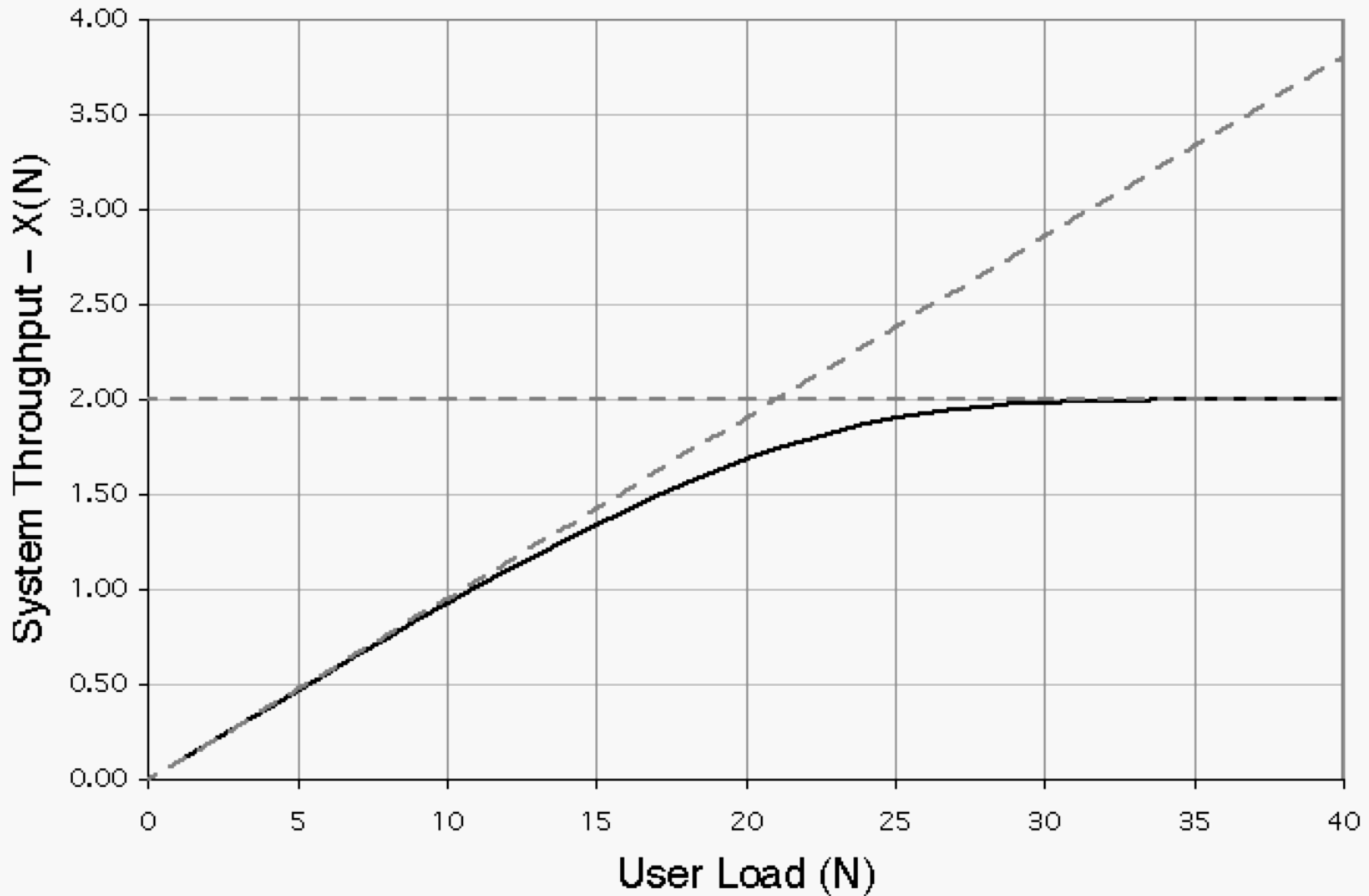
$$X = \frac{\text{Jobs}}{\text{Time}} = \frac{N \frac{T}{R+Z}}{T} = \frac{N}{R+Z}$$

$$R = \frac{N}{X} - Z$$

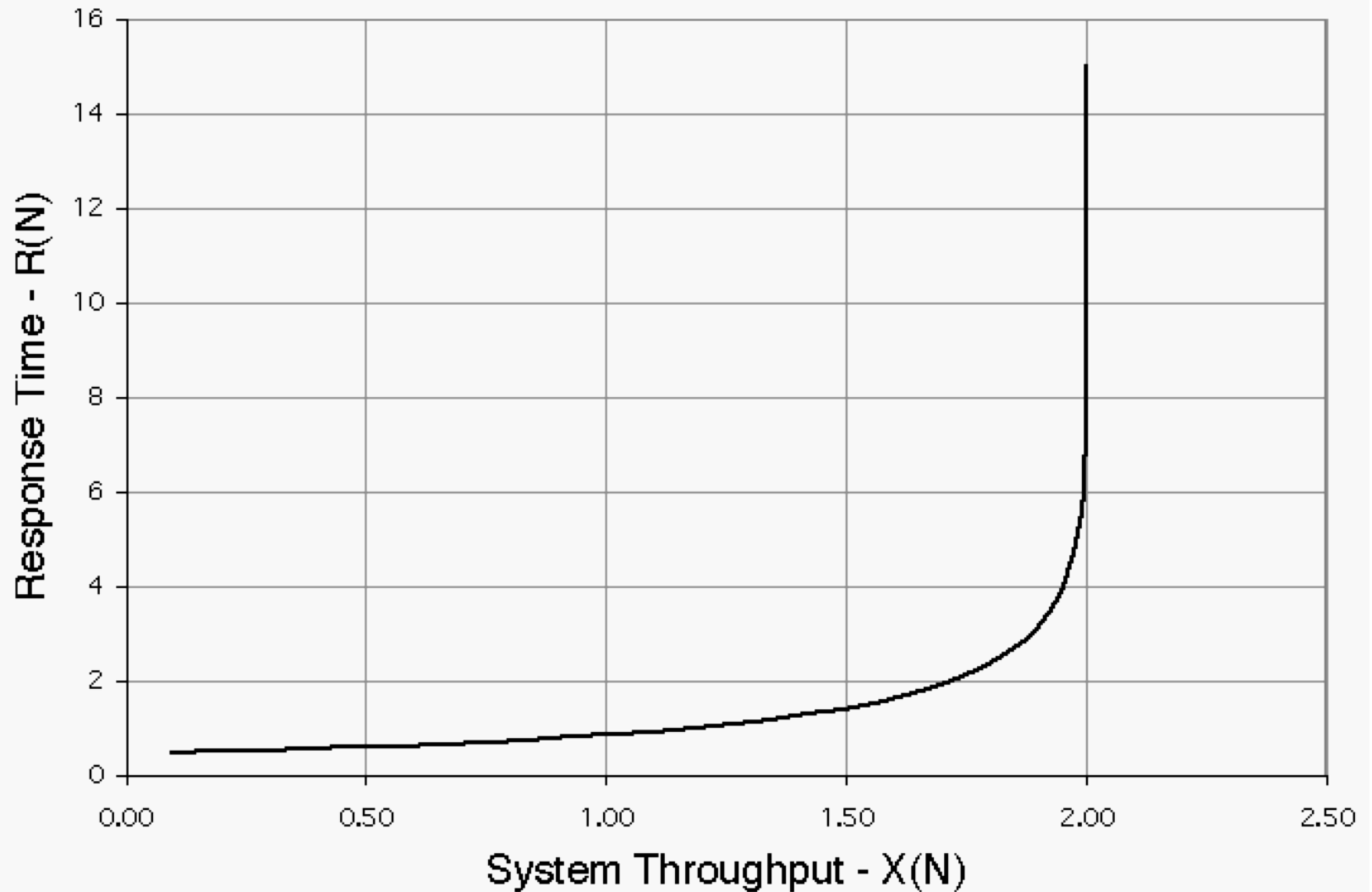
Interactive Law in practice



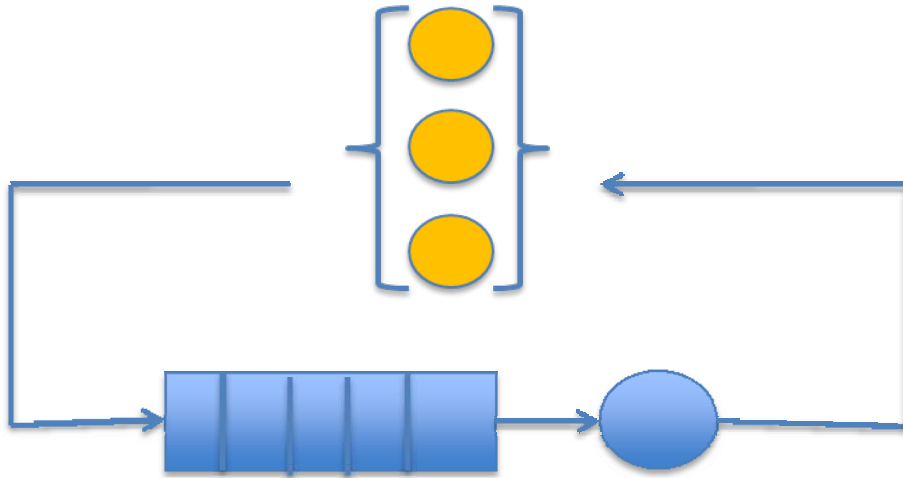
Interactive law in practice



Interactive law in practice



Warning!



- Real systems are not ideal
 - Network effects
 - Processing overhead(not only wait)
 - Exceptions and not “normal” return values