

Queuing Theory

- The system is modeled by a *stationary stochastic process*
- Jobs are *stochastically independent*
- Job steps from device to device follow a *Markov chain*
- The system is in *stochastic equilibrium*
- The service time requirements at each device conform to an *exponential distribution*
- The system is *ergodic* – i.e., long term time averages converge to the values computed for stochastic equilibrium

Theory

- The theory of queuing networks based on these assumptions is usually called “Markovian queuing network theory”
- Italicized words in the previous slide illustrate concepts that the analyst must understand to be able to deploy the models
- Some concepts are difficult
- Some, such as “equilibrium” or “stationary,” cannot be proved to hold by observing the system in a finite time period
- Most can be disproved empirically – For example, parameters change over time, jobs are dependent, device-to-device transitions do not follow Markov chains, systems are observable only for short periods, and service distributions are seldom exponential

Applicability

- It is a surprise that these models apply so well to systems which violate so many assumptions of the analysis

Validation

- When applying or validating the results of Markovian queuing network theory, analysts substitute operational (i.e., directly measured) values for stochastic parameters in the equations
- The repeated successes of validations led us to investigate whether the traditional equations of Markovian queuing network theory might also be relations among operational variables, and, if so, whether they can be derived using different assumptions that can be directly verified and that are likely to hold in actual systems

Operational Principles

- All quantities should be defined so as to be precisely measurable, and all assumptions stated so as to be directly testable. The validity of results should depend only on assumptions which can be tested by observing a real system for a finite period of time
- The system must be flow balanced – i.e., the number of arrivals at a given device must be (almost) the same as the number of departures
- The devices must be homogeneous – i.e., the routing of jobs must be independent of local queue lengths, and the mean time between service completions at a given device must not depend on the queue lengths of other devices

Stochastic Hypothesis

The behavior of the real system during a given period of time is characterized by the probability distributions of a stochastic process

Supplementary Hypotheses

- Supplementary hypotheses are usually also made
- They typically introduce concepts such as state, ergodicity, independence, and the distributions of specific random variables
- These hypotheses constitute a stochastic model

Operational Variables, Laws, and Theorems

- Hypotheses whose veracity can be established beyond doubt by measurement will be called operationally testable
- Operational analysis provides a rigorous mathematical discipline for studying computer system performance based solely on operationally testable hypotheses

Operational Analysis Components

- A system that can be real or hypothetical
- A time period, which may be past, present, or future
- The objective of an analysis is equations relating quantities measurable in the system during the given time period

Observation Period

- A finite time period in which a system is observed
- An operational variable is a formal symbol that stands for the value of some quantity which is measurable during the observation period
- It has a single, specific value for each observation period

Operational Variables

- Operational variables are either:
 - Basic quantities, which are directly measured during the observation period
 - Derived quantities which are computed from the basic quantities

Basic Quantities

- T – the length of the observation period
- A – the number of arrivals occurring during the observation period
- B – the total amount of time during which the system is busy during the observation period ($B \leq T$)
- C – the number of completions occurring during the observation period

Basic quantities (A, B, C) are typical of “raw data” collected during an observation

Derived Quantities

$\lambda = A/T$ the arrival rate (jobs/second)

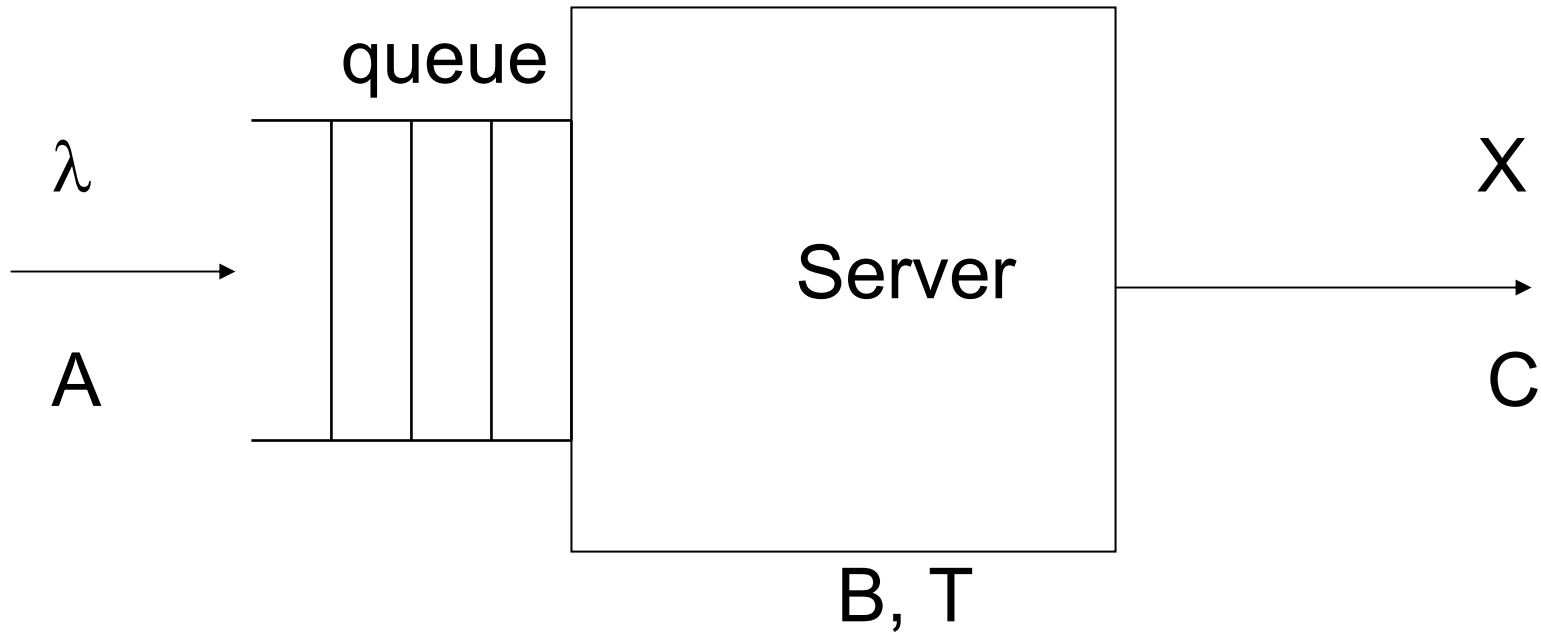
$X = C/T$, the output rate (jobs/second)

$U = B/T$, the utilization (fraction of time system is busy)

$S = B/C$, the mean service time per completed jobs

Derived quantities (λ , X , U , S) are typical of “performance measures” – they are variables that may change from one observation period to another

Single Server Queuing System



Operational Law

$$U = XS$$

Proof:

Since $S = B/C$, $B = S * C$;

and since $X = C/T$, $T = C/X$

Therefore, $U = B/T$ implies $U = S * C * X/C$

so, $U = SX = XS$

Thus, if the system is completing 3 jobs/second, and each job requires 0.1 second of service, the utilization of the system is .3 or 30%

Job Flow Balance

$$A = C$$

- The number of arrivals is equal to the number of completions
- This is called job flow balance because it implies $\lambda = X$
- Job flow balance holds only in some observation periods; however, it is often a very good approximation especially if the observation period is long
- Job flow balance is an example of an operationally testable assumption – it need not hold in every observation period, but we can test whether or not it does

Operational Theorem

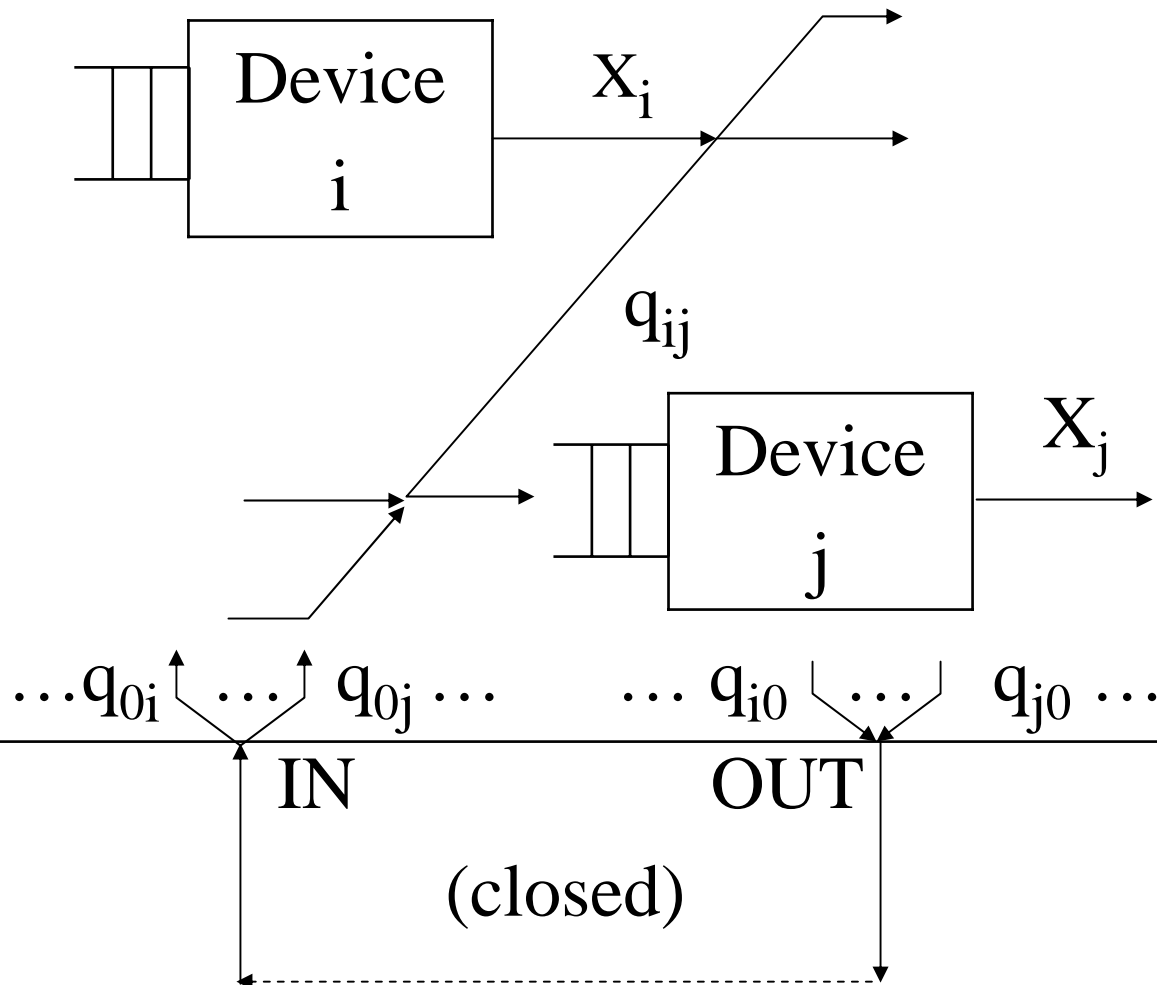
$$U = \lambda S$$

- In a job flow balanced system, utilization equals the arrival rate times the mean service time
- This is an example of an operational theorem – a proposition derived from operational quantities with the help of operational testable assumptions

K-Device Queuing Network

K Devices

N Jobs



K-Device System

- The previous slide shows two of K devices in a multiple-resource network
- A job enters at IN, circulates around the network waiting in queues and having service request processed at various devices, and exits at OUT
- The network is operationally connected – each device is visited at least once by some job during the observation period
- There are no job overlaps regarding use of devices
- Device is busy if request is pending

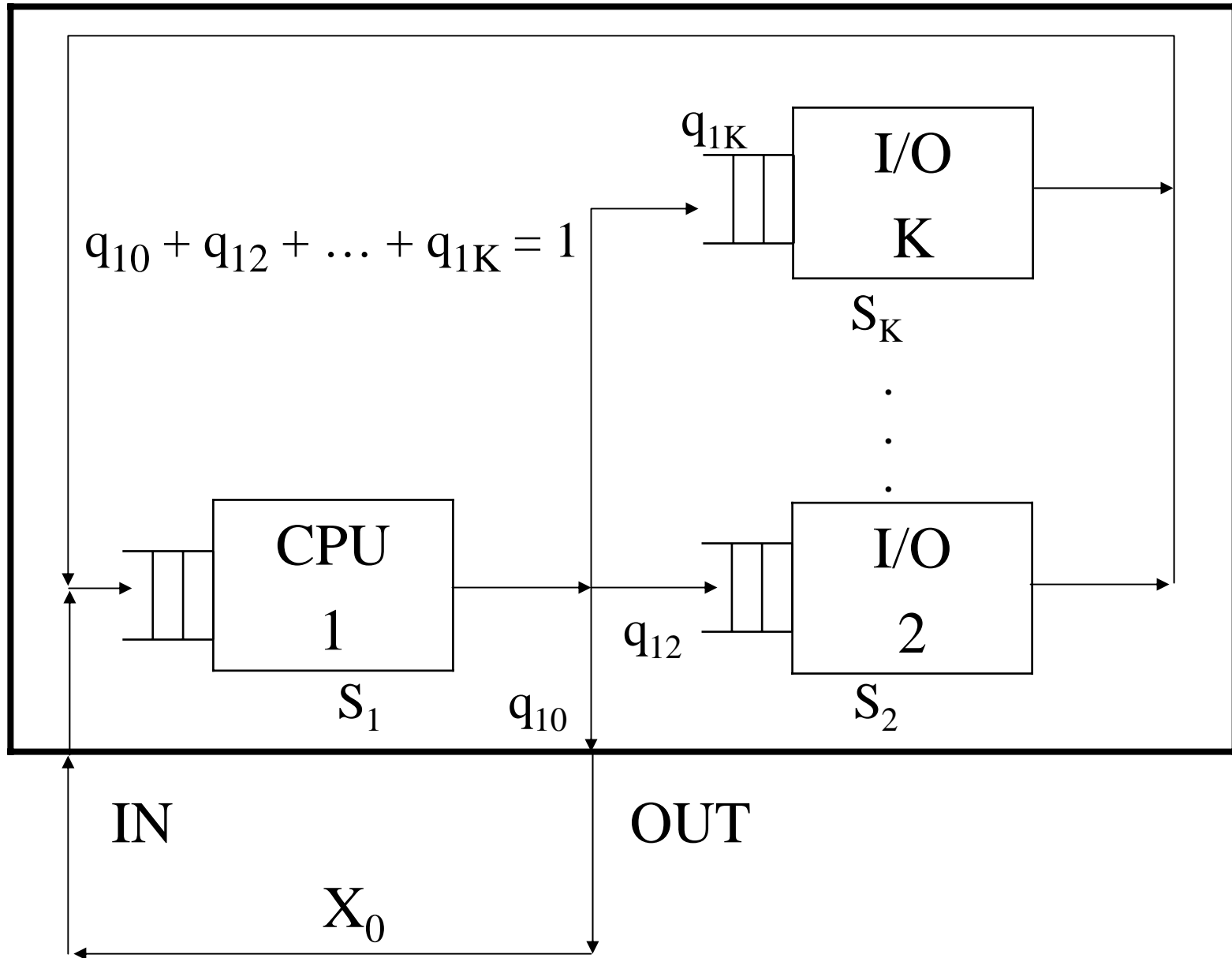
K-Device System - Queues

- Job is in a queue if it is waiting for or receiving service there
- Let n_i denote the number of jobs in the queue at device i
- $N = n_1 + \dots + n_k$ is the total number of jobs in the system

Two-Device System – open vs closed

- The system output rate X_0 is the number of jobs per second leaving the system
- If the system is open, X_0 is known and N varies as jobs enter or leave the system
- If the system is closed, the number of jobs N is fixed – This is modeled in our system by connecting the output back to the input
- An open system assumes that X_0 is known and seeks to characterize the distribution of N
- An analysis of a closed system begins with N given and seeks to determine the resulting X_0 along the OUT/IN path

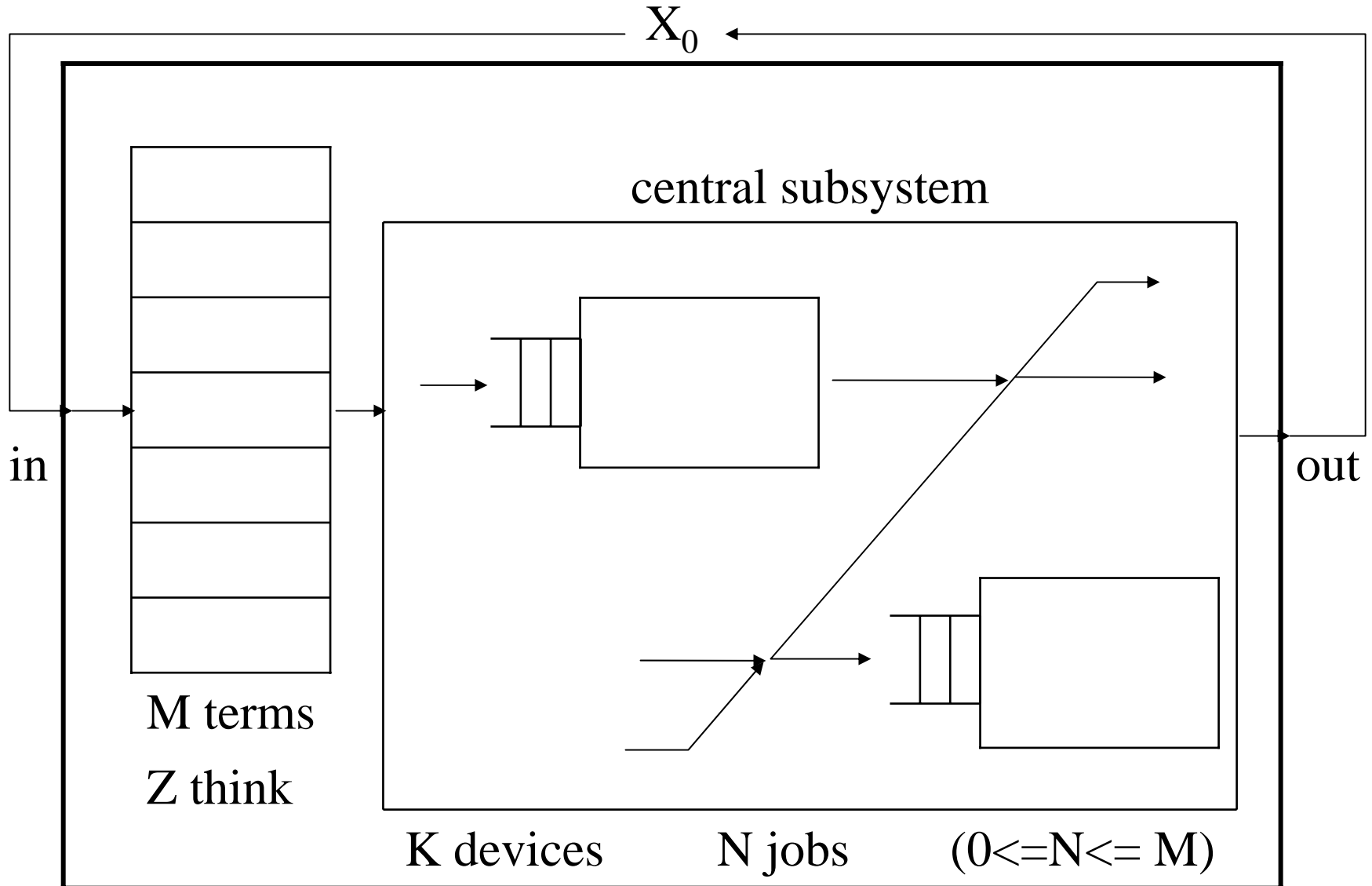
Central Server Network



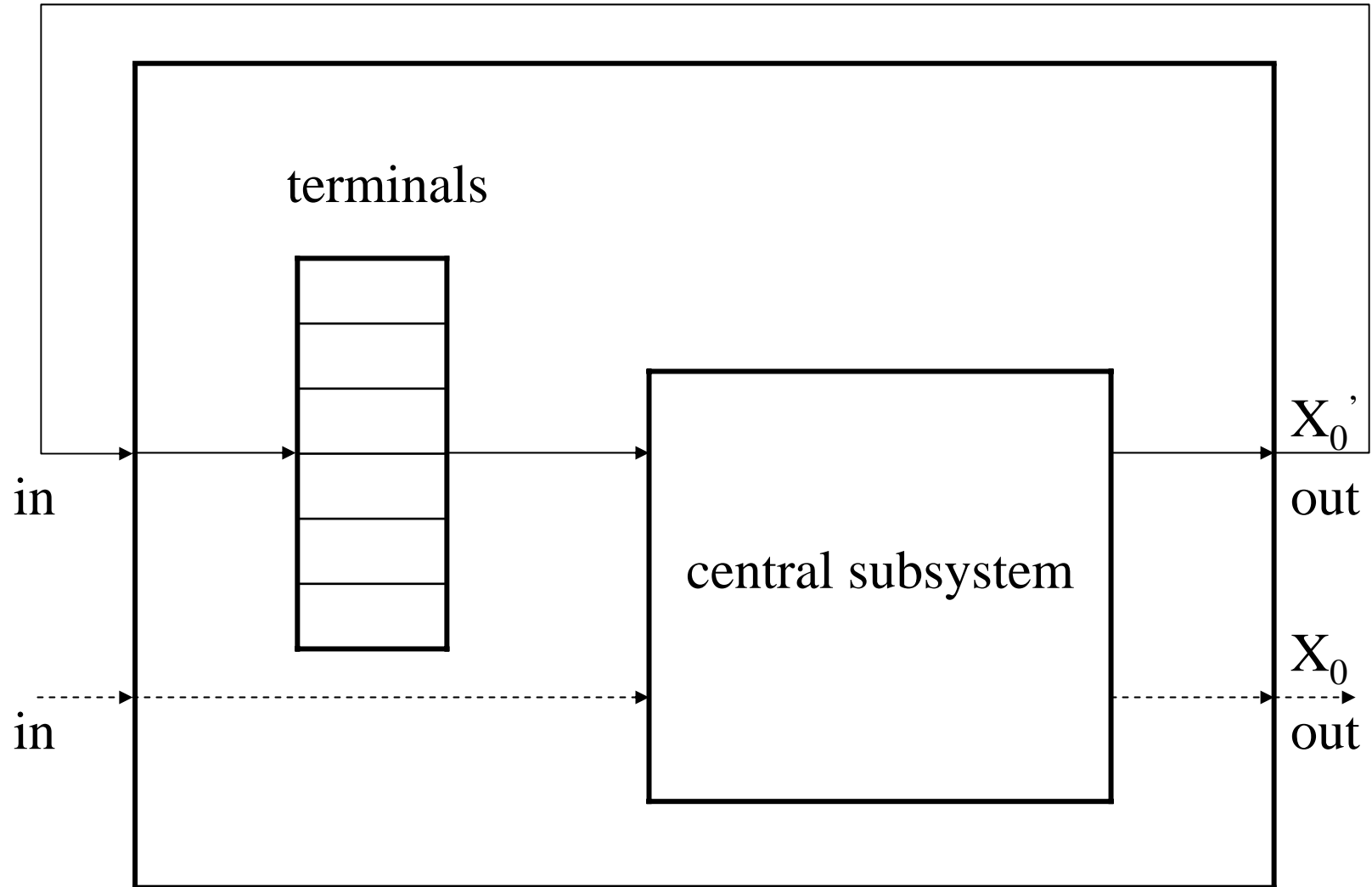
Central Server Network

- This is a common type of network
- Device 1 is the CPU, devices 2 ...K are I/O stations
- A Job begins with CPU service (burst) and continues with zero or more I/O service intervals (bursts) which alternate with further CPU bursts
- The quantities q_{1i} are called “routing frequencies” and the S_i are the mean service times
- A new job enters the system as soon as an active job terminates – This models behavior of a batch OS operating under a backlog
- The throughput of the system under these conditions is X_0

Terminal Driven System



Mixed System



—————→ interactive workload
- - - - -→ batch workload

Basic Operational Quantities

Observation period is T seconds and the following data is collected for each device $i=1, \dots, K$

A_i – Number of Arrivals

B_i – Total busy time (time during which $n_i > 0$)

C_{ij} – Number of times a job requests service at device j immediately after completing a service request at device i

Outside World

If we treat the outside world as device “0”, we can also define:

A_{0j} – Number of jobs whose first service request is for device j

C_{i0} – number of jobs whose last service request is for device i

Possible Combinations

- Assume that $C_{00} = 0$ because otherwise jobs would use no resources before leaving
- However, $C_{ij} > 0$ is possible since a job could request another burst of service from a device which has just completed a service request for that job

Arrivals and Departures

Number of completions at device i :

$$C_i = \sum_{j=0}^K C_{ij}, \quad i = 1, \dots, K$$

Number of arrivals to and departures from
the
system:

$$A_0 = \sum_{j=1}^K A_{0j}$$

$$C_0 = \sum_{i=1}^K C_{i0}$$

Equilibrium

In a closed system $A_0 = C_0$

Operational Quantity Definitions

$$U_i = \text{utilization of device } i \\ = B_i / T$$

$$S_i = \text{mean service time between completions} \\ \text{of requests at device } i \\ = B_i / C_i$$

$$X_i = \text{output rate of requests from device } i \\ = C_i / T$$

$$q_{ij} = \text{routing frequency, the fraction of jobs} \\ \text{proceeding next to device } j \text{ on completing a} \\ \text{service request at device } i \\ = C_{ij} / C_i, \text{ if } i = 1, \dots, K \\ \text{or} \\ = A_{0j} / A_0, \text{ if } i = 0$$

Routing Frequencies

For any i , $q_{i0} + q_{i1} + \dots + q_{iK} = 1$

Note that q_{i0} is an output routing frequency (fractions of completions from device i corresponding to the final service request of some job and q_{0j} is an input routing frequency (fraction of arrivals to the system which proceed first to device j)

System Output Rate

$$\begin{aligned} X_0 &= \text{system output rate} \\ &= C_0 / T \end{aligned}$$

Output Flow Law

$$X_0 = \sum_{i=1}^K X_i q_{i0}$$

Note that X_0, X_1, \dots, X_K cannot be interpreted as “throughputs” because no assumption of job flow balance has been made (yet).

Utilization Law

$$U_i = X_i S_i$$

Proof:

Since $S_i = B_i / C_i$, $B_i = S_i * C_i$;

and since $X_i = C_i / T$, $T = C_i / X_i$;

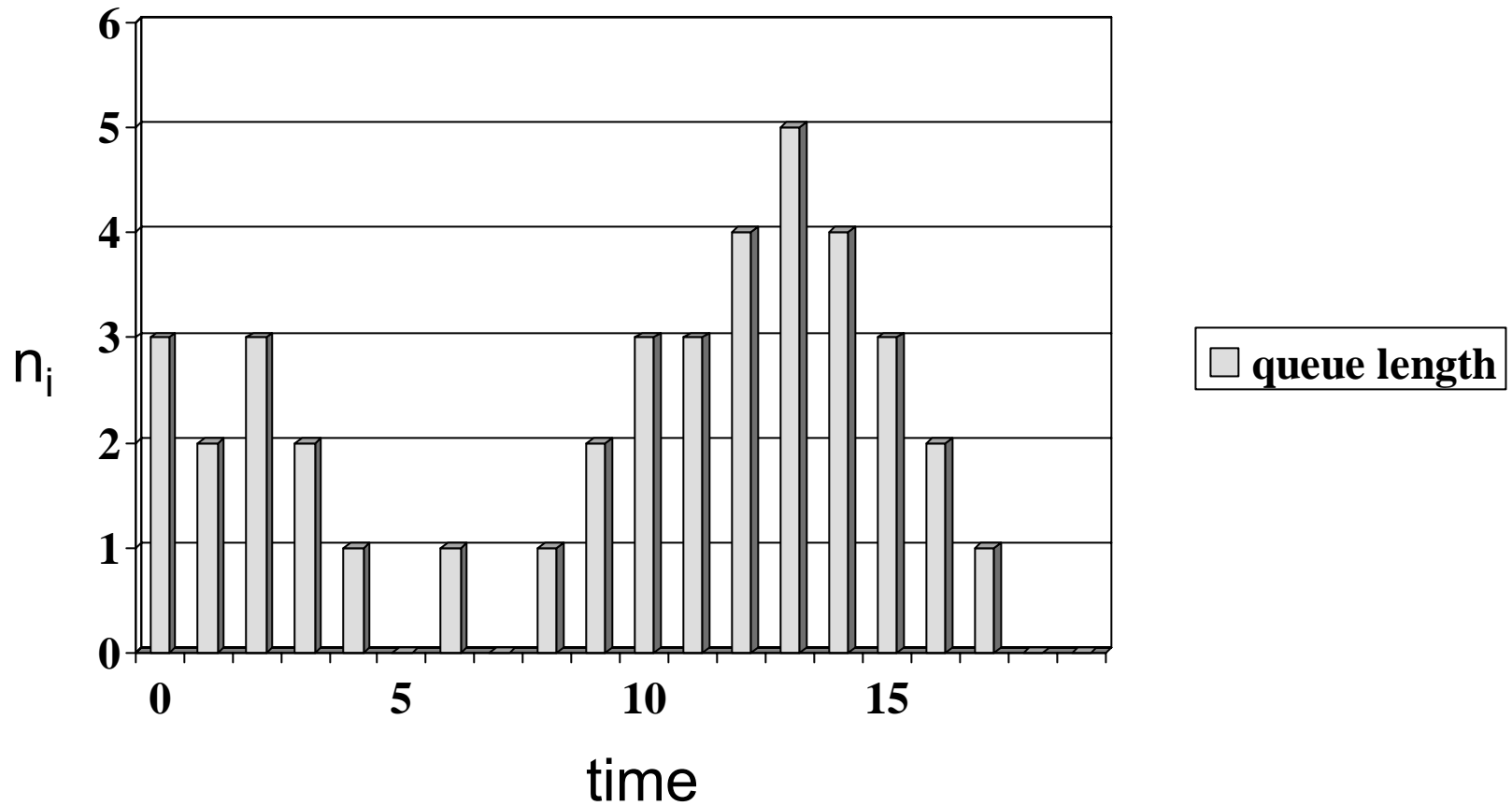
Therefore, $U_i = B_i / T$ implies $U_i = S_i * C_i * X_i / C_i$

so, $U_i = S_i X_i = X_i S_i$

Queues

- n_i is the queue length at device i
- Sometimes we write $n_i(t)$ to make the time dependence explicit
- $n_i(t)$ includes jobs waiting for and receiving service at time t

$n_i(t)$ Example



Mean Queue Length

W_i = area under graph $n_i(t)$ during the observation period – can be interpreted as the total number of “job-seconds” accumulated at device i during the observation period

$$\check{n}_i = W_i / T$$

is the mean queue length of device over time period T (or the average height of the graph for device i)

Response Time

$$R_i = W_i / C_i$$

is the average amount of time
accumulated at
device i per completed request

Little's Law

An immediate consequence of the last two formulae for average queue length and response time is Little's Law:

$$\check{n}_i = X_i R_i$$

$\check{n}_i = W_i / T$ and $R_i = W_i / C_i$ or $W_i = R_i * C_i$ imply

$$\check{n}_i = R_i * C_i / T \text{ or}$$

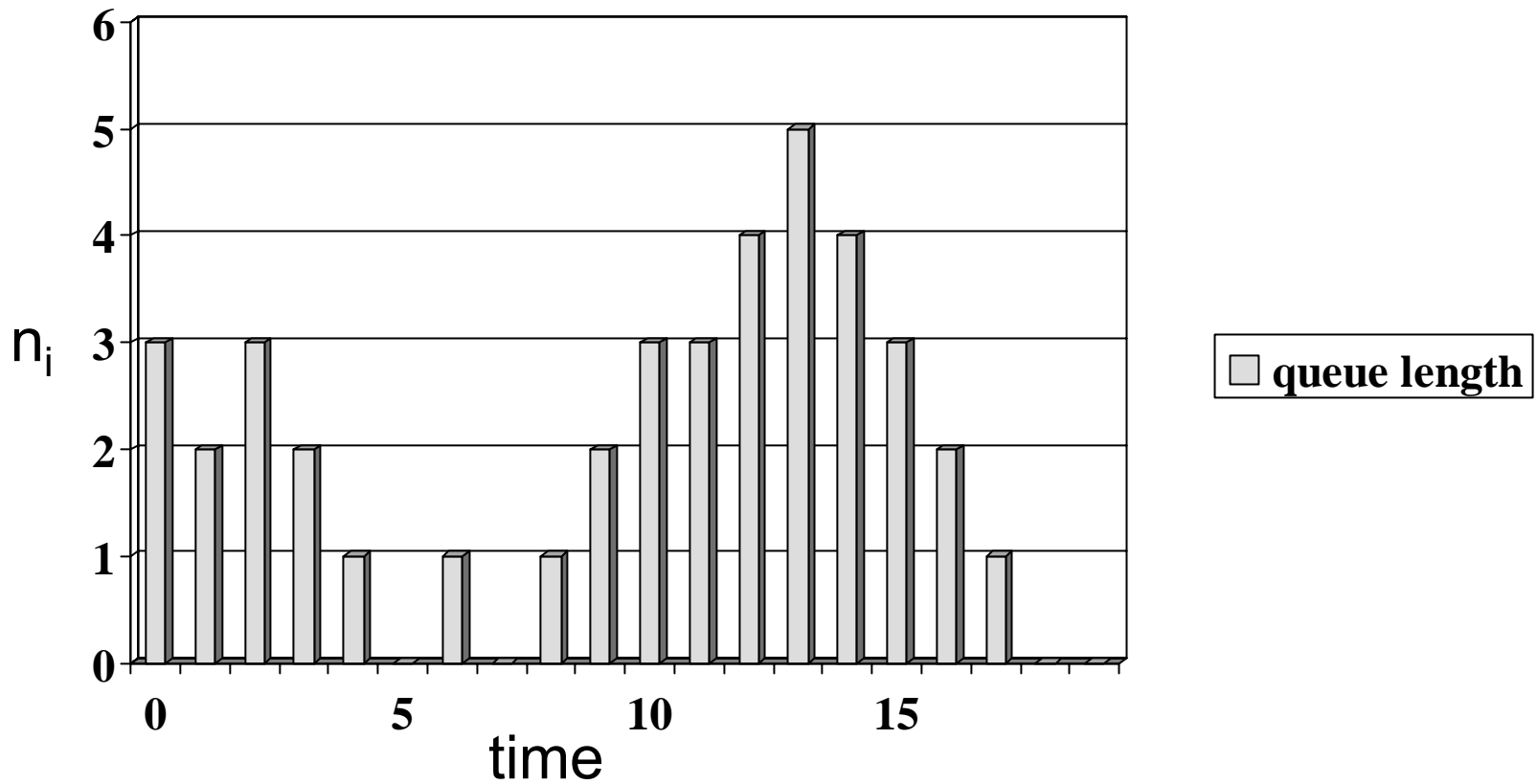
$$\check{n}_i = R_i * X_i$$

Example 1

$n_i(t)$



B_i



Example 1(Cont)

Observation period for the device i queue length is 20 seconds

$A_i = 7$ jobs, $B_i = 16$ seconds, $C_i = 10$ jobs

Since $n_i(0) = 3$, $n_i(20) = n_i(0) + A_i + C_i = 0$

$U_i = B_i / T = 16/20 = .8 = 80\%$ (device utilization)

$S_i = B_i / C_i = 16/10 = 1.6$ seconds (mean service time)

$X_i = C_i / T = 10/20 = .5$ jobs/second (output rate)

$W_i = 40$ job-seconds

$\check{n}_i = W_i / T = 40/20 = 2$ jobs (mean queue length)

$R_i = W_i / C_i = 40/10 = 4$ seconds (response time)

Job Flow Balance

- For each device i , $\lambda_i = X_i$
- This principle will give a good approximation if the observation period is long enough and the difference between the arrivals and completions, $A_i - C_i$, is small compared with C_i
- It will be exact if the initial queue length $n_i(0)$ is the same as the final queue length $n_i(T)$
- When job flow is balanced, we refer to X_i as device throughputs

Job Flow Balance Equations

$$C_j = A_j = \sum_{i=0}^K C_{ij}$$

Since $q_{ij} = C_{ij} / C_i$

$$C_j = \sum_{i=0}^K C_i q_{ij}$$

and since $X_j = C_j / T$

$$X_j = \sum_{i=0}^K X_i q_{ij}, \quad j = 0, \dots, K$$

Visit Ratios

- V_i is the visit ratio of device i , which expresses the mean number of requests per job for a device (the mean number of visits per job to a device)
- $V_i = X_i / X_0$ (the job flow through device i relative to the system's output flow)
- $V_i = C_i / C_0$ (the mean number of completions at device i for each completion from the system)

Forced Flow Law

$$X_i = V_i X_0$$

The Forced Flow Law states that flow in any one part of the system determines the flow everywhere in the system

Example 2

- Jobs generate an average of 5 disk requests
- Disk throughput is measured as 10 requests/second

Question: What is the system throughput

Answer: Assume subscript i refers to the disk, then:

$X_i = V_i X_0$ is the Forced Flow Law

$$X_0 = X_i / V_i = \frac{10 \text{ requests / second}}{5 \text{ requests / job}}$$

$$= 2 \text{ jobs / second}$$

Visit Ratio Equations

On replacing each X_m with $V_m X_0$ in the job flow balance equations,

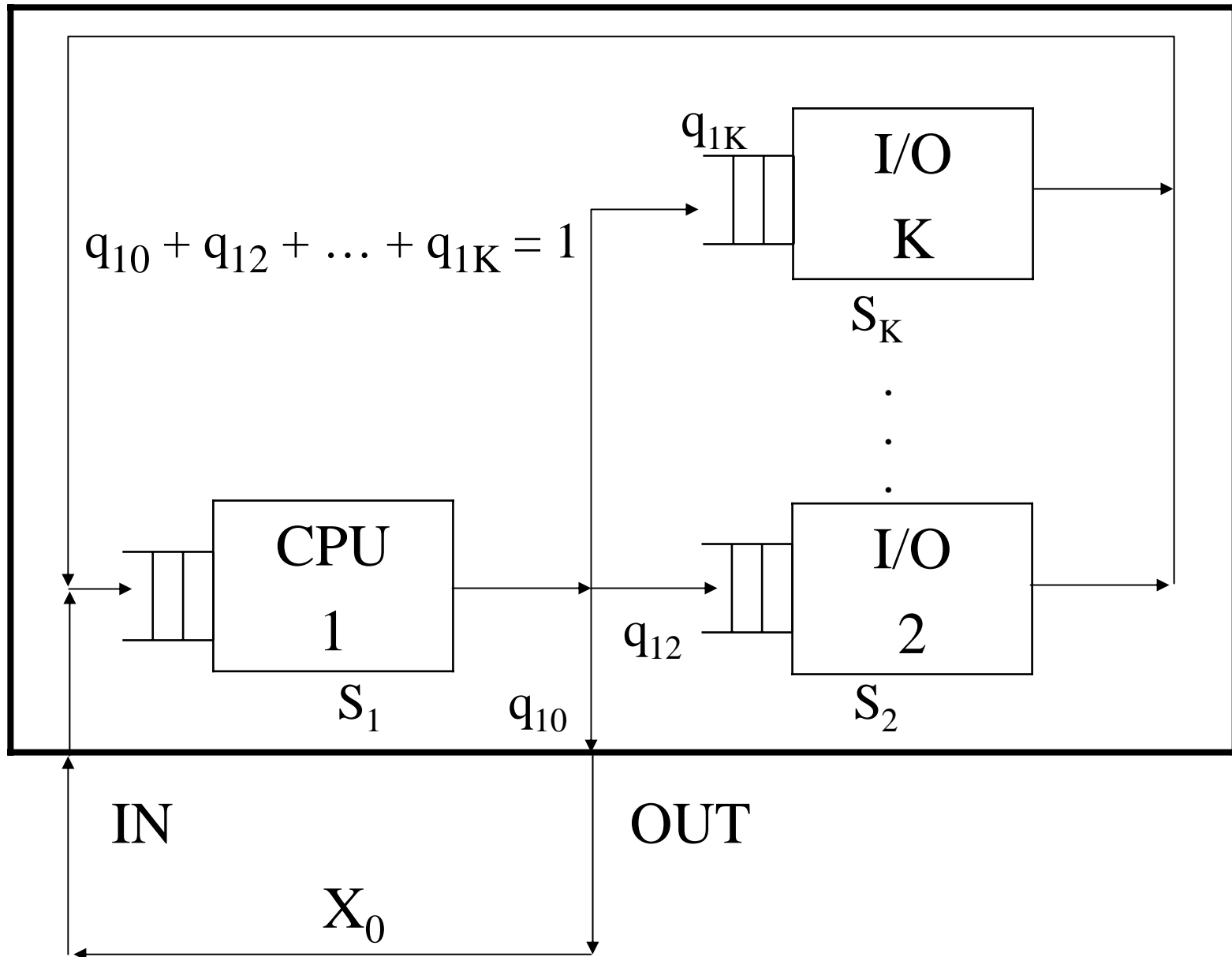
$$X_j = \sum_{i=0}^K X_i q_{ij}, \quad j = 0, \dots, K$$

we obtain the following Visit Ratio Equations

$$V_0 = 1$$

$$V_j = q_{0j} + \sum_{i=1}^K V_i q_{ij}, \quad j = 1, \dots, K$$

Example 3 Central Server



Ex 3 – Job Flow Equations

$$X_0 = X_1 q_{10}$$

$$X_1 = X_0 + X_2 + \dots + X_k$$

$$X_i = X_1 q_{1i}, i = 2, \dots, k$$

Ex 3 – Visit Ratio Equations

By setting $X_m = V_m X_0$, the Job Flow Equations reduce to the Visit Ratio Equations:

$$V_0 X_0 = V_1 X_0 q_{10}$$

$$V_1 X_0 = V_0 X_0 + V_2 X_0 + \dots + V_K X_0$$

$$V_i X_0 = V_1 X_0 q_{1i}, i = 2, \dots, k$$

or

$$V_0 = 1 = V_1 q_{10}$$

$$V_1 = 1 + V_2 + \dots + V_K$$

$$V_i = V_1 q_{1i}, i = 2, \dots, k$$

Ex 3 – Visit Ratio Equations (Cont)

Since $1 = V_1 q_{10}$, it follows that:

$$V_1 = 1/q_{10}$$

Also, if $V_1 = 1/q_{10}$ and $V_i = V_1 q_{1i}$, $i = 2, \dots, k$,
then:

$$V_i = q_{1i}/q_{10}, i = 2, \dots, k$$

Ex 3 – Visit Ratios

- The point is that in a network such as this one, we can solve simultaneous equations for all visit ratios
- Given the visit ratios and mean service times S_i , we can compute all performance quantities
- In practice, an analyst can extract visit ratios directly from workload data, thereby avoiding computing a solution to the visit ratio equations

Response Time

Applying Little's Law to the system as a whole:

$$R = \check{N} / X_0, \text{ where } \check{N} = \check{n}_1 + \dots + \check{n}_K$$

or

$$R = \sum_{i=1}^K \check{n}_i / X_0$$

General Response Time Law

If \check{N} or X_0 are not known, Little's Law:

$$\check{n}_i = X_i R_i$$

and the Forced Flow Law:

$$X_i = V_i X_0$$

imply that $\check{n}_1 / X_0 = V_i R_i$, so:

$$R = \sum_{i=0}^K \check{n}_1 / X_0 = R = \sum_{i=0}^K V_i R_i$$

which is known as the General Response Time Formula

Interactive Response Time

Formula

- If Z = The time a user thinks, $Z + R$ is the mean time for a user to complete a think/wait cycle
- When job flow is balanced, X_0 denotes the rate cycles are completed
- By Little's law, $(Z + R) X_0$ must be the mean number of users which is M , so:

$$M = (Z + R) X_0, \text{ or } R = M / X_0 - Z$$

Operational Equations

Utilization Law	$U_i = X_i S_i$
Little's Law	$\check{n}_i = X_i R_i$
Forced Flow Law	$X_i = V_i X_0$
Output Flow Law	$X_0 = \sum_{i=1}^K X_i q_{i0}$
General Response Time Law	$R = \sum_{i=0}^K V_i R_i$
Interactive Response Time Formula (assumes job flow balance)	$R = M / X_0 - Z$

Example 4 - Conditions

The following data is generated on a time sharing system:

Each job generates 20 disk requests;

The disk utilization is 50%

Mean service time at the disk is 25 milliseconds

There are 25 terminals

Think time is 18 seconds

Q1: What is the system throughput?

Q2: What is the system response time?

Example 4 - Solution

A1 – The Forced Flow and Utilization Laws imply:

$$X_0 = X_i / V_i = U_i / V_i S_i = .5 / (20) (.025) = 1$$

job/second

A2 – The Interactive Response Time Formula implies:

$$R = M / X_0 - Z = 25 / 1 - 18 = 7 \text{ seconds}$$

Example 5 – Conditions

The following data is collected from a mixed system:

There are 40 terminals

Think time is 15 seconds

Interactive response time is 5 seconds

Disk mean service time is 40 milliseconds

Interactive jobs generate 10 disk requests

Batch jobs generate 5 disk requests

Disk utilization is 90%

Q1 – What is the throughput of the batch system?

Q2 – Estimate a lower bound on the interactive response time if batch throughput is tripled

Ex 5 – Interactive Throughput

$$X_0' = M / (Z + R') = 40 / (15 + 5) = 2$$

jobs/second

Ex 5 – Disk Throughput

Let subscript i refer to the disk

- Disk throughput is the sum of the batch component X_i and the interactive component X_i' , or $X_i + X_i'$

- The Utilization Law implies:

$$X_i + X_i' = U_i / S_i = .9 / .04 = 22.5$$

requests/second

- The Forced Flow Law implies the interactive component is:

$$X_i' = V_i' / X_0' = (10) (2) = 20 \text{ requests/second}$$

- Therefore, the batch component is:

$$X_i = (X_i + X_i') - X_i' = 22.5 - 20 = 2.5$$

Ex 5 – Batch Throughput

A 1 – Using the forced flow law again, the batch throughput is:

$$X_0 = X_i / V_i = 2.5 / 5 = 0.5 \text{ jobs/second}$$

Ex 5 – Triple Batch Throughput

If batch throughput X_0 is tripled from 0.5 to 1.5 without changing V_i , then the batch component of disk throughput X_i would change to:

$$X_i = V_i X_0 = (5) * 1.5 = 7.5 \text{ requests/second}$$

Ex 5 – Maximum Disk Throughput

If the disk mean service time S_i is invariant throughout this change, the Utilization Law implies the maximum disk throughput is:

$$X_i' + X_i =$$

$$U_{i(\max)} / S_i = 1 / S_i = 25 \text{ requests/second}$$

This implies X_i' cannot exceed $25 - 7.5 = 17.5$

Ex 5 – New Interactive Throughput

By the Forced Flow Law, the interactive throughput is changed to:

$$X_0' = X_i' / V_i' \leq 17.5/10 = 1.75 \text{ jobs/second}$$

Ex 5 – New Interactive Response Time

A2 – Using the Interactive Response Time Formula, the new interactive response time is:

$$R' = M / X_0' - Z \geq 40/1.75 - 15 = 7.9 \text{ seconds}$$

This presumes that M , Z , V_i and S_i are invariant under the change of batch throughput – Although reasonable, these assumptions should be checked

Bottleneck Analysis

Ratios

The ratio of the completion rates for any two

devices is equal to their visit ratios, or:

$$X_i / X_j = V_i / V_j$$

and the Utilization Law $U_i = X_i S_i$ implies that:

$$U_i / U_j = V_i S_i / V_j S_j$$

Saturation

- Device i is saturated if its utilization is 100%
- The Utilization Law implies that the following holds for a device in saturation:

$$X_i = 1 / S_i$$

This means the completion rate is one request every S_i seconds

- In every system $U_i \leq 1$ and $X_i \leq 1 / S_i$

Bottleneck Device

- Since the ratio U_i/U_j is fixed as N increases, the device with the largest $V_i S_i$ is the first to achieve saturation
- If b is a subscript to identify the bottleneck device,

$$V_b S_b = \max (V_1 S_1, \dots, V_K S_K)$$

- Bottlenecks are determined by the device and workload parameters

Maximum Throughput

- If N becomes large, $U_b = 1$ and $X_b = 1/S_b$
- Since $X_0/X_b = 1/V_b$ implies that:

$$\max X_0 = 1 / V_b S_b$$

is the maximum possible value for system throughput

Minimum Throughput

- $V_i S_i$ is the total of all service request time per job at device i
- $V_i S_i$ is the minimum time that job requests can be serviced at device i – it does not include any queuing delays
- Therefore minimum system response time is:

$$\min R_0 = V_1 S_1 + \dots + V_K S_K$$

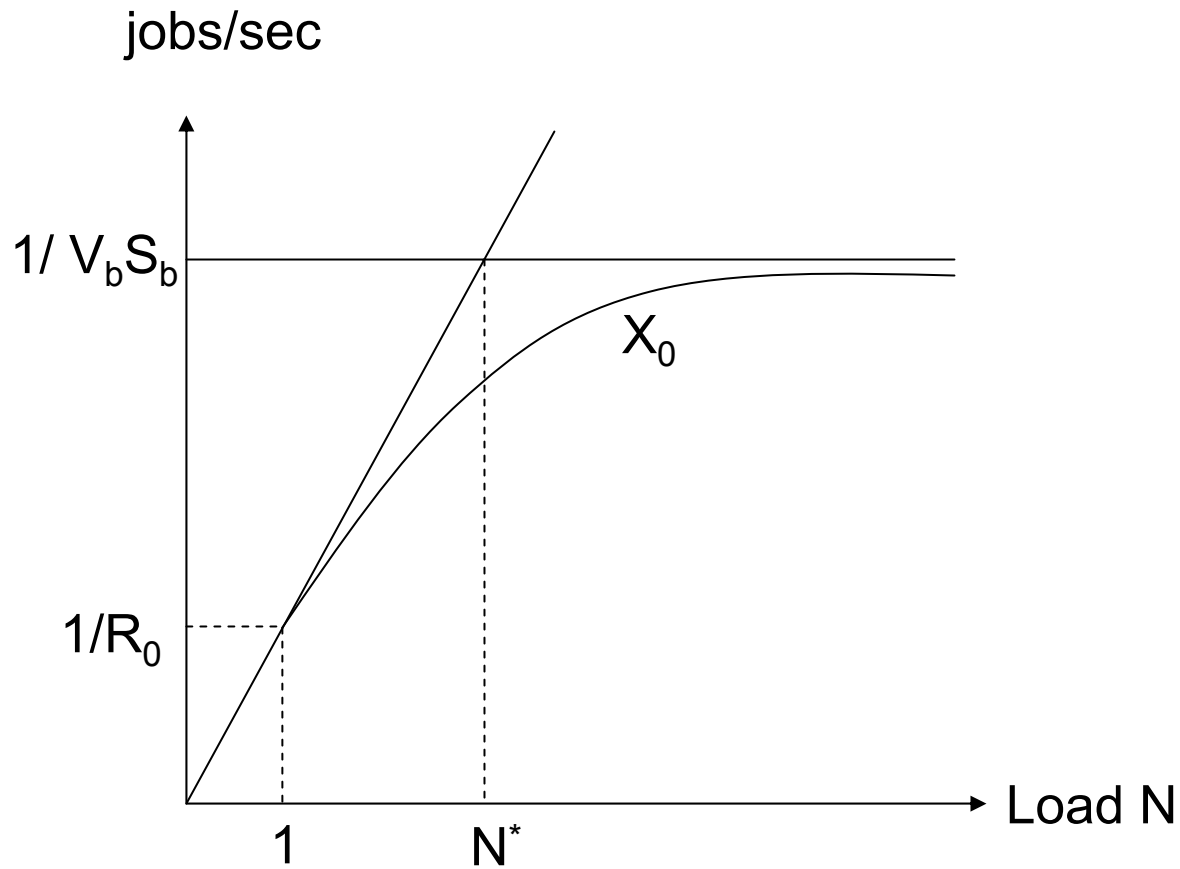
- R_0 is the response time when $N = 1$, so from Little's Law, minimum throughput is:

$$\min X_0 = 1 / R_0$$

Job Interference

- As throughput increases from:
 $1/R_0$ where $N = 1$ to
 k/R_0 where $N = k$
- $k/R_0 \leq 1/V_b S_b$ or
- $k \leq N^* = R_0/V_b S_b$
 $= (V_1 S_1 + \dots + V_K S_K)/V_b S_b \leq K$
- In words, $k > N^*$ implies with certainty that jobs queue somewhere in the system
- N^* is the saturation point of the system

System Throughput



Terminal Driven System

- For M terminals and think time Z ,

$$R = M / X_0 - Z$$

- When $M = 1$, R is $\min R_0$ and X_0 cannot exceed $1/V_b S_b$
- $R \geq M V_b S_b - Z \geq M V_i S_i - Z$, for $i = 1..K$
- As M becomes large, R approaches the asymptote $M V_b S_b - Z$

Terminals in Saturation

- Notice that the response time asymptote intersects the horizontal axis at:

$$M_b = Z/V_b S_b$$

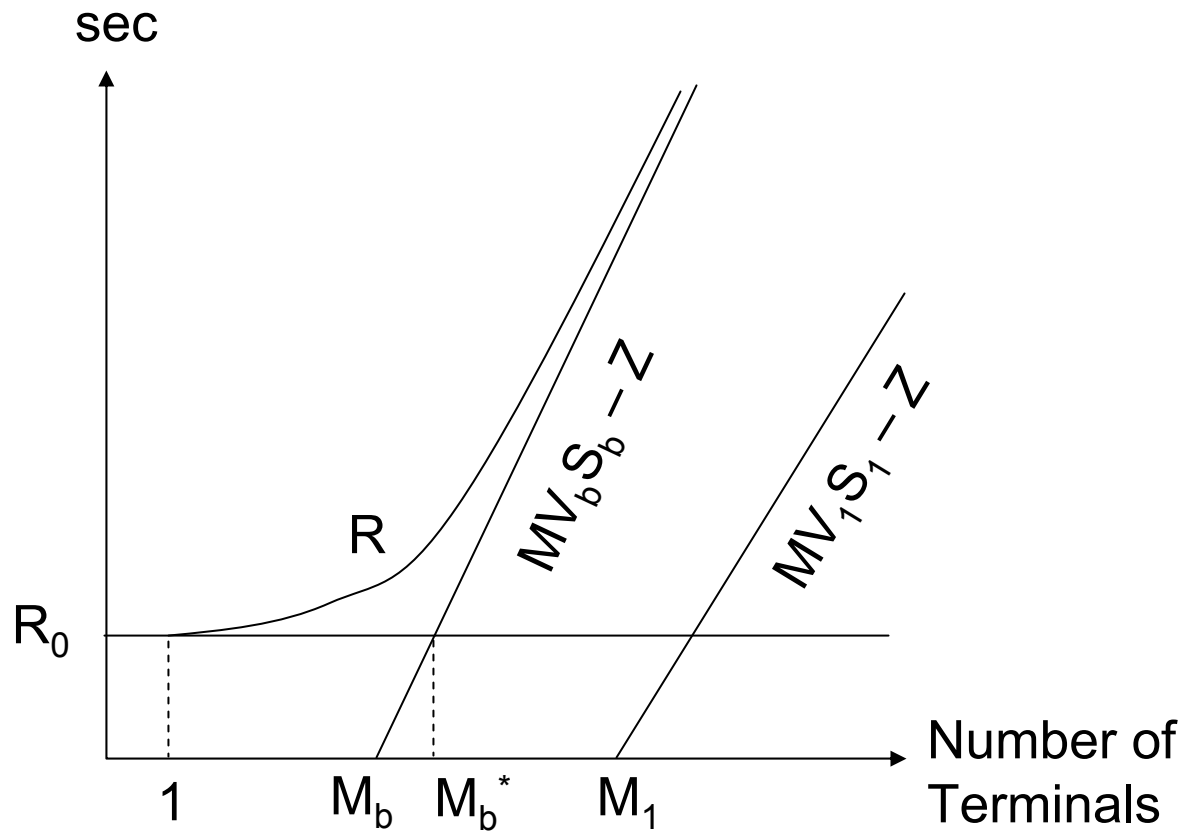
This is the mean number of terminals when the system is in saturation

- The response time asymptote crosses the minimum response time R_o at

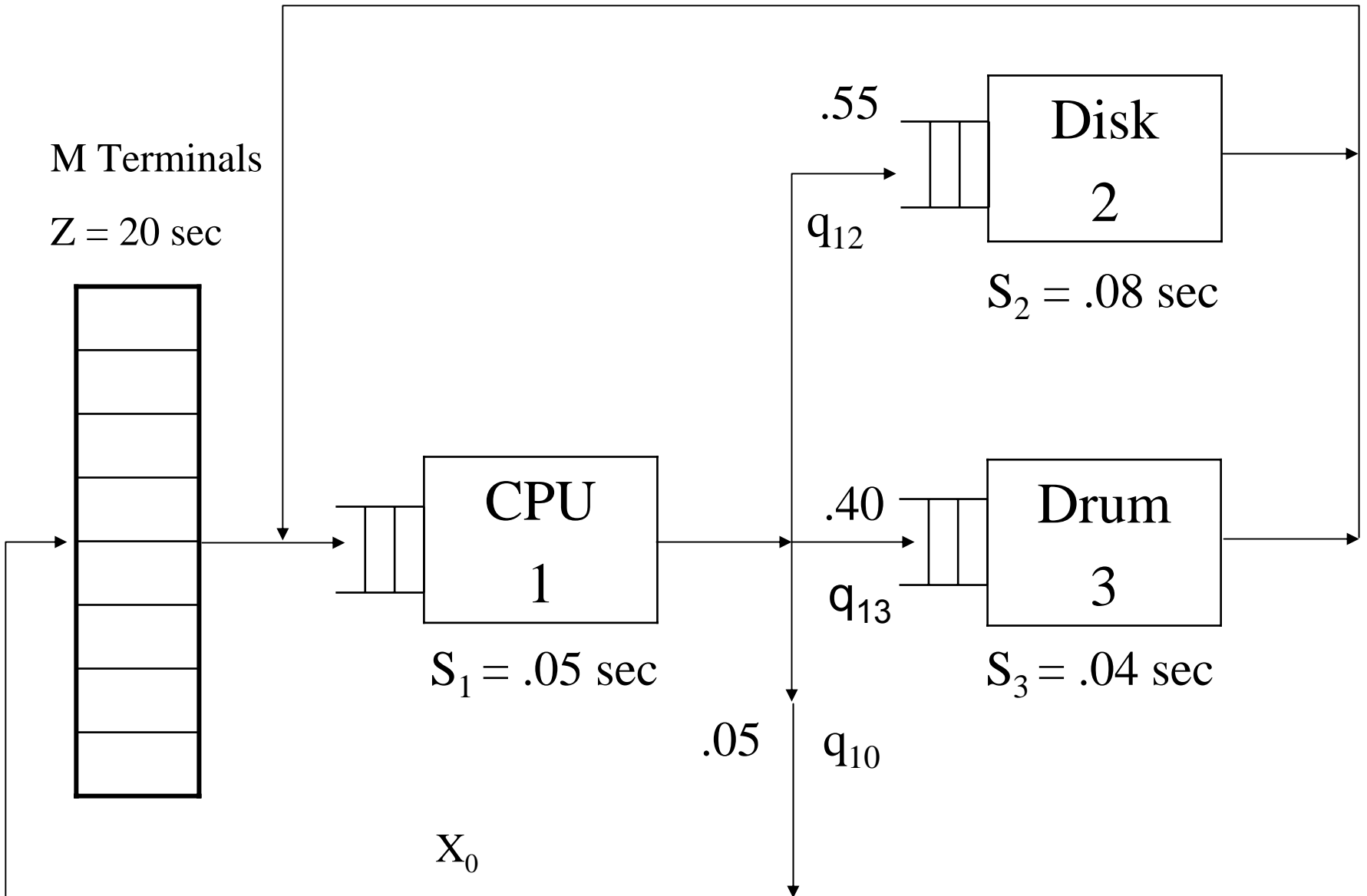
$$M_b^* = (R_o + Z)/V_b S_b = N^* + M_b$$

- M_b^* is the number of terminals where queuing is certain to exist in a central subsystem

Response Time



Interactive System



Visit Ratio Equations

$$V_0 = 1 = .05V_1$$

$$V_1 = V_0 + V_2 + V_3$$

$$V_2 = .55V_1$$

$$V_3 = .40V_1$$

Solution is:

$$V_0 = 1, V_1 = 20, V_2 = 11, V_3 = 8$$

Example 6

Question –

How many terminals are required to saturate the system?

Example 6 – $V_i S_i$ Products

- The $V_i S_i$ products are:

$$V_1 S_1 = (20)(.05)$$

$$= 1 \text{ seconds (Total CPU Time)}$$

$$V_2 S_2 = (11)(.08)$$

$$= .88 \text{ seconds (Total disk Time)}$$

$$V_3 S_3 = (8)(.04)$$

$$= .32 \text{ seconds (Total drum Time)}$$

So, bottleneck device is CPU or $b = 1$, and

$$R_0 = 1 + .88 + .32 = 2.2 \text{ seconds}$$

Example 6 – Saturation

- The number of thinking terminals in saturation is:

$$M_b = Z / V_b S_b = 20 \text{ terminals}$$

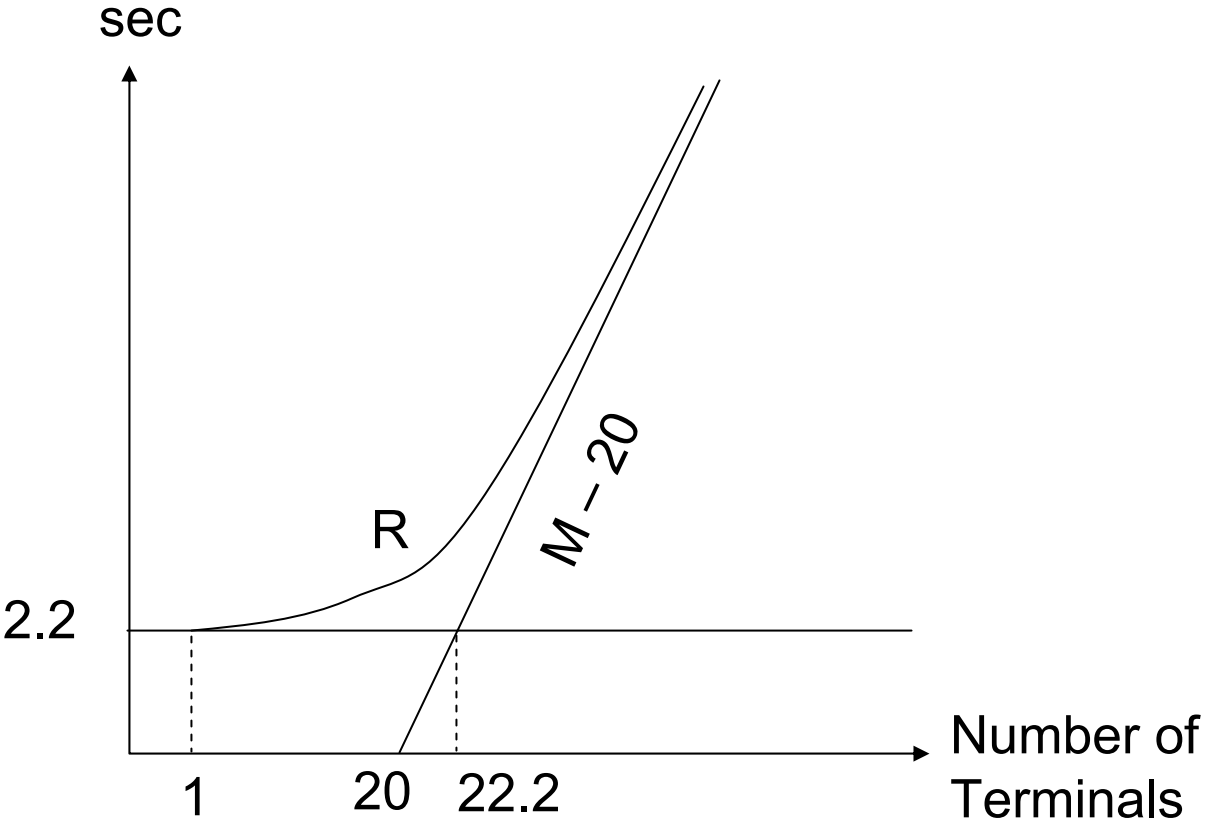
- Saturation point of the central subsystem is

$$N^* = R_0 / V_b S_b = 2.2 \text{ jobs}$$

- Number of terminals required to saturate the entire system is:

$$M_b^* = M_b + N^* = 22.2$$

Response Time Curve



Example 7 – Conditions

- Throughput = 0.715 jobs/second
- Mean Response Time = 5.2 seconds

Question – What is the mean number of users logged in during the observation period?

Example 7 – Solution

Using the interactive response time formula:

$$\begin{aligned} M &= (R + Z)/X_0 \\ &= (5.2 + 20)/(.715) \\ &= 18 \text{ terminals} \end{aligned}$$

Example 8

Question 1

Is an 8 second response time feasible when 30 users are logged in?

Question 2

If not, what minimum CPU speedup is required?

Example 8 – Response Time Asymptote

A1 – For $M = 30$, the response time asymptote requires that:

$$R \geq MV_b S_b - Z = 30 - 20 = 10 \text{ seconds}$$

Therefore, an 8 second response time is not feasible

Example 8 – Faster CPU?

- When considering whether a response time is feasible with an infinitely fast CPU:
 - First, check whether it's greater than the new minimum response time = $V_2S_2 + \dots + V_kS_k$
 - Then, check whether it's greater than the minimum response time for the next highest V_iS_i

$$R \geq M V_i S_i + Z$$

Example 8 – Faster CPU

- To meet an 8 second response time, a faster CPU implies the following:

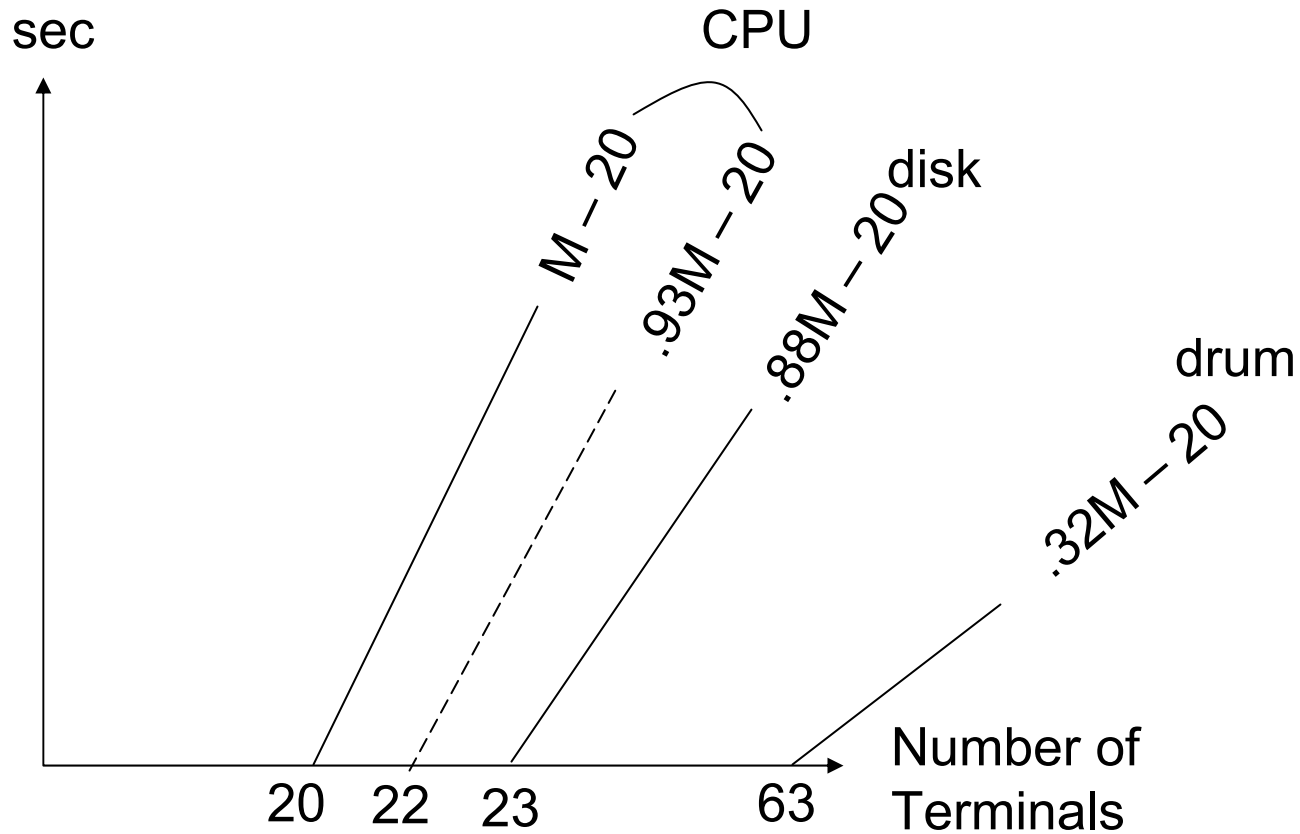
$$MV_1S_1' - Z \leq 8 \text{ seconds}$$

$$\text{or } S_1' \leq 0.047 \text{ seconds}$$

- Speedup factor is $S_1 / S_1' = 1.07$
- New CPU must be 7% faster
- Since $V_1S_1 = 0.93$, CPU is still the bottleneck

A2 – An 8 second response time is feasible with a faster CPU

Response Time Asymptotes



Example 9

Question 1

Is a 10 second response time feasible if 50 users are logged in?

Question 2

If not, what minimum CPU speedup is required to achieve a 10 second response time?

Example 9 – Current CPU

With the current CPU,

$$\begin{aligned} R &\geq MV_1S_1 - Z \\ &= (50)(1.0) - 20 = 30 \text{ seconds} \end{aligned}$$

A1 – No, with the current CPU an 8 second response time is not feasible if 50 users are logged on

Example 9 – Infinitely Fast CPU

- When the CPU is infinitely fast, $S_1 = 0$
- In that case, the disk would be the bottleneck so:

$$\begin{aligned} R &\geq MV_2S_2 - Z \\ &= (50)(.88) - 20 = 24 \text{ seconds} \end{aligned}$$

A2 - No amount of CPU speedup will achieve an 8 second response time with 50 users logged on

Example 9 – 50 Terminals

