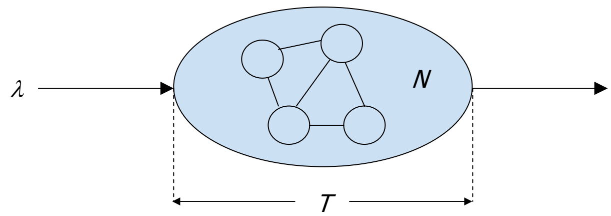
Little's Theorem

Little's Theorem: Introduction

- More interested in long term, steady state than in startup ➔ Arrivals = Departures
  - Black box view of the system
- Little's theorem:
  Mean # of tasks in system = arrival rate x mean reponse time
  $n = \lambda w$
  - Observed by many, Little was first to prove
  - One of the most commonly used theorems in queuing theory
- Applies to any system in equilibrium, as long as nothing in black box is creating or destroying tasks
  - Can be applied even to the systems where jobs can be lost
  - Applied to parts of systems consisting of waiting and service positions as a job is not lost if it finds a buffer position

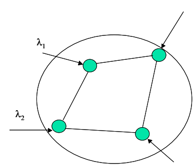
Little's Theorem



- λ: customer arrival rate
- N: average number of customers in system
- T: average delay per customer in system
- <u>Little's Theorem</u>: System in steady-state

$$N = \lambda T$$

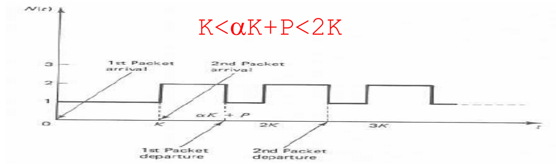
Example # 1: Little's Theorem



- Consider a network of transmission lines
  - Packets arrive at `n` different nodes with corresponding rates λ_1, …, λ_n
  - If `N` is the average total number of packets inside the network, then determine the average delay per packet (`T`), regardless of the packet length distribution and method of routing packet
- Applying Little's theorem: $T = \frac{N}{\sum_{i=1}^{n} \lambda_i}$
- If N_i and T_i are average number in the system and average delay of packets arriving at node `i`, respectively, then

$$N_i = \lambda_i T_i$$

Example # 2



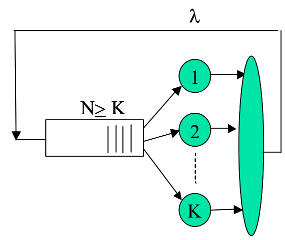
- A packet arrives at a transmission line every K seconds
  - The first packet arriving at time 0
  - All packets have equal length and require αK seconds for transmission where $\alpha < 1$
  - The processing and propagation delay per packet is P seconds
  - Determine average number in the system N
- Solution:
  - The arrival rate here is $\lambda = 1/K$
  - Since packets arrive at regular rate (equal inter-arrival times), there is no delay for queuing ➔ time T a packets spends in the system (including propagation delay) is: $T = \alpha K + P$
- Applying Little's theorem to find time average # in system:

  $N = \lambda T = \alpha + P/K$
  - Here, the # in system N(t) is a deterministic function of time
  - In this case, N(t) does not converge but Little's theorem holds with N viewed as a time average

Example # 3

- Consider a window flow control system
  - Window size is W for each session
  - Arrival rate of packets into the system for each session = λ
  - Apply Little's theorem to analyze impact of W on λ and delay T
- Applying Little's theorem:
  - Since, # of packets in the system is never more than W, therefore $W \geq \lambda T$
  - If congestion builds up in the system → T increases and λ must eventually decrease
  - Next, the network is congested and capable of delivering λ packets per unit time for each session. Assuming delays for ACKs to be negligible relative to forward packets and $W \cong \lambda T$
    - Increasing W in this case only result in increasing delay T without appreciably changing λ

Example # 4



- Consider a K server queuing system
  - Room for at most $N \geq K$ customers in system
  - The system is always full
  - Assume that it starts with N customers and that a departing customer is immediately replaced by a new customer
    - A closed queuing system
  - Average customer service time = $\overline{X}$
  - We want to find the average customer time in the system T
- Applying Little's theorem twice:
  - For the entire system: $N = \lambda T$
  - For servers only: $K = \lambda \overline{X}$ → servers continuously busy
  - By eliminating l in the two relations:

$$T = \frac{N\overline{X}}{K}$$

Example # 4 (Cont'd)

- Now consider same system under different arrival assumptions
  - Customers arrive at rate λ but are blocked (and lost) from the system if they find the system full
  - Then the number of servers that may be busy are less than K
  - Let $\overline{K}$ be the average number of busy servers
  - Let β be the proportion of customers that are blocked from entering the system
- Applying Little's theorem to the servers of the system:
  - Effective arrival rate = $(1 - \beta)\lambda$
  - Then average number of busy servers are given as: $\overline{K} = (1 - \beta)\lambda \overline{X}$
  - Which gives: $\beta = 1 - \frac{\overline{K}}{\lambda \overline{X}}$
  - Since, $\overline{K} \leq K$, we obtain a lower bound on blocking probability as:

$$\beta \geq 1 - \frac{K}{\lambda \overline{X}}$$

Example # 5: A Polling System

- Consider a transmission line:
  - Serves m packet streams (i.e., m users) in round-robin cycles
  - In each cycle, some packets of user 1 are transmitted followed by some packets of user 2, and son on until finally packets of user m are transmitted
  - An overhead period of average length A_i precedes the transmission of the packets of user i in each cycle
  - The arrival rate and average transmission time of the packets of user i are λ_i and $\overline{X}_i$ respectively
  - If $A = A_1 + A_2 + \ldots + A_m$, determine average cycle length L
- Applying Little's theorem:
  - Fraction of time the transmission line is busy transmitting packets of user i is = $\lambda_i \cdot \overline{X}_i$
  - Overhead period of packet i can be viewed as transmission of "packets" with average transmission time of A_i

Example # 5 (Cont'd)

- Application of Little's theorem (cont'd)
  - Arrival rate of these overhead "packets" = $1/L$
  - Fraction of time used for transmission of these overhead "packets" using Little's theorem = A/L
  - Therefore,

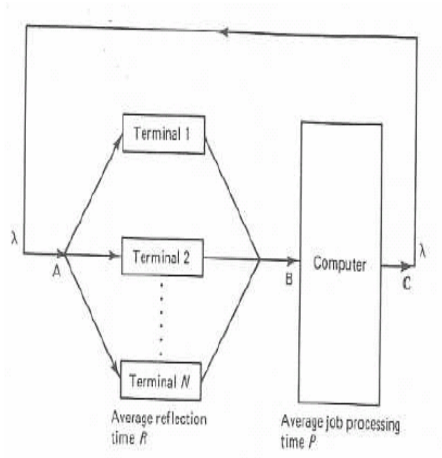
$$1 = \frac{A}{L} + \sum_{i=1}^{m} \lambda_i \overline{X}_i$$

which yields the average cycle length as:

$$L = \frac{A}{1 - \sum_{i=1}^{m} \lambda_i \overline{X}_i}$$

Example # 6: Time-Sharing System

- Time-sharing system with N terminals
  - A user logs into the system through a terminal
  - After an initial period of average length R submits a job that requires an average processing time P at the computer
  - Jobs queue up inside computer and are served by a single CPU according to an unspecified priority or time-sharing rule
- Estimate:
  - Maximum throughput sustainable by the system (in jobs per unit time); and
  - Average delay of a user



Term 032 3-3-11 COE540-Abdul Waheed

Example # 6 (Cont'd)

- Bounds on the attainable system throughput λ
  - Assume number in the system is always N → to get upper bound
    - As soon as a user departs → replaced by another immediately
    - Model: departing user re-enters the system immediately
  - Bounds on N and T can be translated into throughput bounds via Little's theorem: $\lambda = N/T$
- Apply Little's theorem between points A to C:
  - If T is the average time in the system: $\lambda = N/T$
  - $T = R + D$
    - R is the average reflection time before a job is submitted
    - D is the average delay between submitting job until its completion: $P \le D \le NP$ → D varies from no waiting (P) to maximum waiting (NP)
  - Therefore, $R+P \le T \le R+NP$
  - Thus, bounds on λ are given as: $\frac{N}{R+NP} \le \lambda \le \frac{N}{R+P}$

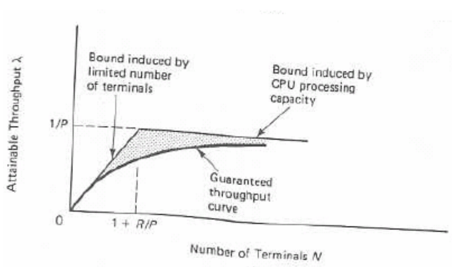
Term 032 3-3-12 COE540-Abdul Waheed

Example # 6 (Cont'd)

- Throughput is also bonded above by processing capacity
  - Execution time of a job is P units on the average
  - Computer cannot process more 1/P jobs per unit time in the long run
  - Therefore, $\lambda \le \frac{1}{P}$
- By combing two results: $\frac{N}{R+NP} \le \lambda \le \min\left\{\frac{1}{P}, \frac{N}{R+P}\right\}$
- Bounds on average delay using T = N/λ:
  - When system is fully loaded

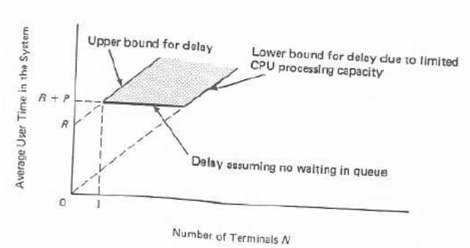
$$\max\{NP, R+P\} \le T \le R+NP$$

Example # 6 (Cont'd)



- Throughput bounds:
  - As # of terminals N increases → throughput reaches up to 1/P
  - When N < 1 + R/P → N becomes throughput bottleneck
  - When N > 1 + R/P → limited processing power is the bottleneck
- These bounds are independent of system parameters
  - This is due to Little's theorem

Example # 6 (Cont'd)



- Bounds on average delay:
  - Delay rises in direct proportion to N
  - Assume fully loaded system

Example # 7

- A monitor on a disk server showed that the average time to satisfy an I/O request was 100 milliseconds. The I/O rate was about 100 requests per second. What was the mean number of requests at the disk server?

- Using Little's theorem:
  Mean number in the disk server = arrival rate x response time
  = (100 reqests/sec)(0.1 sec)
  = 10 requests

Example # 8: The M/M/1 Queue

- Little's Theorem: average time in system

$$T = \frac{N}{\lambda} = \frac{1}{\lambda}\frac{\lambda}{\mu-\lambda} = \frac{1}{\mu-\lambda}$$

- Average waiting time and number of customers in the queue – excluding service

$$W = T - \frac{1}{\mu} = \frac{\rho}{\mu-\lambda} \text{ and } N_Q = \lambda W = \frac{\rho^2}{1-\rho}$$