ECE 333: Introduction to Communication Networks
Fall 2002

Lecture 11: Delay models I

- **Components of Network delay**
- **Little's theorem**

We are going to spend the next two lectures discussing ***delay*** in networks. Delay refers to the time required for data to be sent from its origin to its destination. As we saw in lecture 1, low delay is a key service requirement for many applications. For this reason it is a basic performance measure used to evaluate networks. Also, when we study protocols for higher network layers, delay considerations have a strong influence on the choice of algorithms.

Our focus is on the delay experienced by a packet being sent between two points in the network. This quantity will generally vary over time, depending on the other traffic, the errors that occur, etc. Thus the delay for a given packets is often modeled as a random variable. Let T_n denote the delay experienced by the nth packet sent between two points. Several measures of delay are commonly used. One quantity is the ***maximum delay***, i.e. the smallest d such that $\Pr(T_n \le d) = 1$ for all n. Another common measure is the ***average delay,*** given by

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} T_n$$

In some cases, the ***delay jitter,*** or the variance in the delay is also important. Which delay measure is appropriate depends on the application. For example, good quality (real-time) voice requires a maximum delay less

than 90 msec and a small delay jitter. For many data applications a small average delay may be sufficient.
In a network, the main sources of delay include the following:

1. Transmission delay
2. Propagation delay
3. Retransmission delay
4. Processing delay
5. Queueing delay

We have considered the first two factors in the previous lectures as well as in the homework. These depend primarily on the physical channel and the transmission technique used. Recall that on a link, the propagation delay is generally fixed for every packet, while the transmission delay depends on packet size.

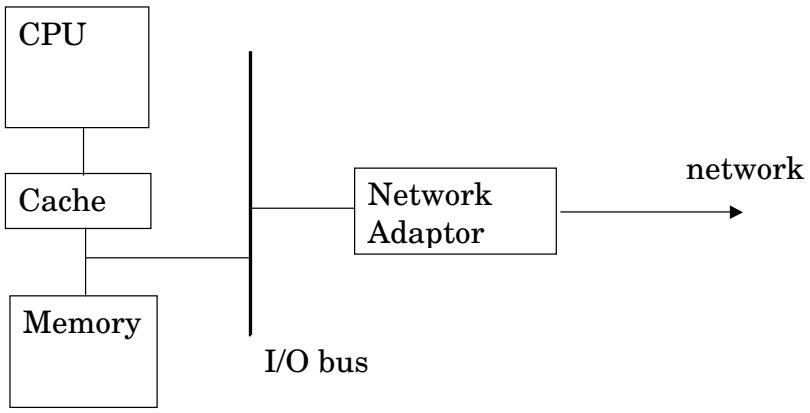
The third factor, the retransmission delay, depends on the ARQ strategy. When a network does not provide a reliable service, this may be zero. Also in networks with very low error rates, this can be minor (with a well-designed ARQ protocol).

Next we will briefly discuss processing delays. Then we will focus on queueing delay for the remainder of this lecture and the next.

Processing delay -

Processing delays refers to the time required for any processing of a packet before it is transmitted. Processing delays may be incurred at both the transmitter and receiver. The nature of these delays strongly depends on the protocols used and the computer architecture of a host.

The figure below shows a high level example of an architecture for a workstation connected to a network.



The physical layer and data link layer will generally be implemented in hardware on the network adaptor card; this card is generally connected to the system I/O bus. Data is transferred to and from the host's main memory over the I/0 bus.

When a packet arrives from the network, before it is delivered to an application running on the CPU, various sources of delay are incurred, including:

- Number of clock cycles required for implementing the required protocols.
- Buffering in the network adapter card to match line speeds.
- Time to acquire the I/O bus.
- Transmission time on the I/O bus.
- Type of memory access - e.g. DMA (direct memory access) or PIO (programmed I/O).
- Interrupt processing time (time for CPU to respond to interrupt generated by network adapter)
- Number of times the message must be read across CPU bus.

(Similar delays occur when a host is transmitting a packet.)

Computer architectures and protocols can be optimized to minimize these delays.

> *E.g*. design protocol to limit number of times full message needs to be read across the CPU bus - pass pointer instead of messages - use dedicated hardware, etc.

For routers and servers in high-speed networks, these are important considerations.

Addressing such concerns require an understanding of computer architecture and operating systems that is beyond the requirements of this class. The important point to remember is that these types of effect can have an important impact on the overall delay.

In the following, we will usually consider processing delay to be fixed for every packet and often include it with the propagation time of the packet.

Queueing Delay:

Queuing delay accounts for the time a packet waits in memory before it is processed and transmitted. The reason for delay may be because another packet is currently being transmitted, either by the same source, or in the case of a broadcast network, another source.

Example:

Consider a transmission line with a rate of 1 Mbps.

Suppose that packets of size 1000 bits are to be transmitted over the line. Thus it takes 1 msec to transmit each packet.

If the time between packet arrivals is larger than 1 msec, no queueing will take place.

Suppose, Packet 1 arrives at t=0 , Packet 2 arrives at t=.4 msec.

Packet 1 takes 1 msec to be transmitted.

Thus, Packet 2 must wait in buffer for .6 msec before it is transmitted. This is the ***queuing delay*** of Packet 2.

Queueing

Some simplified models of queueing in networks can be analyzed mathematically. The study of such models is the subject of ***Queueing Theory***. We will look at some very basic results and give some intuitive derivations - the fine mathematical details will not be stressed.

(If you are interested in seeing more of this type of analysis - take ECE 454.)

Queueing theory (History)

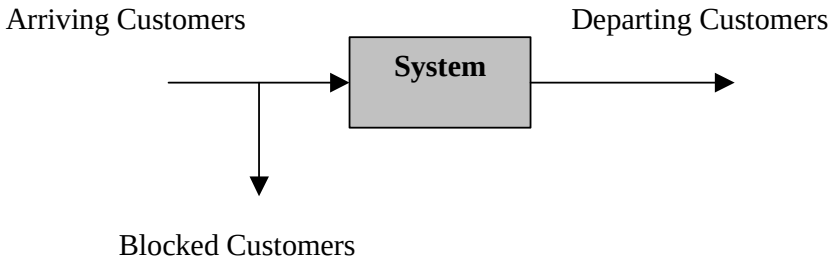
- Developed in early 1900's by Erlang.
- Motivated by telephone network applications. (Erlang worked as an engineer for the Copenhagen Telephone Company)
- Developed throughout 1900's - applied in many areas including manufacturing systems, customer service facilities, transportation systems, and computer memory.

Uses of Queueing Models in Networks:

- *Performance analysis* (However, simplifying assumptions are generally made - a detailed performance analysis usually requires simulations and actual data measurements in addition to queueing analysis.)
- *Provides intuition and understanding of basic trade-offs.*

Basic Queueing Model

"Customers" arrive at random times to a "system". Each customer requires a random amount of "service" before it can leave. Some customers may be ***blocked*** (turned away) and unable to enter the system.



Typically, for us:

Customers = packets, bits, sessions, etc.

System = Network, Buffer, etc.

Service = transmission time (depends on packet length)

To have a mathematical model of a queueing system we need to specify the following: (i) how customers arrive, (ii) how much service each customer requires, (iii) how they are served and (iv) if and when they are blocked (not given sevice).

The arrival of each customer will be modeled as being random. Specifically, the time between when one customer arrives and when the next customer arrives will be modeled as a random variable, called the ***inter-arrival time***. We denote the inter-arrival time between customer *n-1* and customer *n*, by A_n. By convention, A_1 denotes the arrival time of the first customer, thus the arrival time of the nth customer is given by $A_1 + \ldots + A_n$.

The ***service time*** required by a customer will also be modeled as a random variable (Recall, for a packet transmitted at a fixed rate, the packet length is proportional to the service time). We will denote the service time of the nth customer by X_n. After the nth customer arrives, it will wait in the systems for some time W_n, before receiving service, this is referred to as the ***queueing time*** of the customer. Thus, the customer will leave the system $T_n = X_n + W_n$ seconds after arriving, where T_n is the total customer delay.

How long the customer has to wait in the system depends on the other arrivals as well as the ***queueing discipline.*** This is the rule the system uses to decide who gets served next

Some examples of queuing disciplines:

One packet at a time:

- First come first serve (FCFS)
- Last come first serve (LCFS)
- Round robin
- Priorities (on session basis, time elapsed, etc.)

Multiple packets at a time:

- FCFS
- Separate queues/separate servers.

One can also specify a blocking rule, a common one is simply to discard users when the system is full. Alternatives include randomly dropping customers, and blocking customers from a certain class. In most of the following, we will ignore blocking; we assume all customers are admitted and that the system can hold an arbitrary number of customers.

We next define some quantities that will be useful in analyzing a queueing system. Assume that at time 0 the system is empty and let $\alpha(t)$ denote the number of arrivals to the system between time 0 and t. This will be an increasing, piecewise constant function of t. The ***average arrival rate*** to the system is given by

$$\lambda = \lim_{t\to\infty} \frac{\alpha(t)}{t} \text{ customers/sec.}$$

Let $\beta(t)$ denote the number of customers to depart from the system between time 0 and t. The number of customers in the system at time t is then given by:

$$N(t) = \alpha(t) - \beta(t).$$

The ***average number of customers*** in the system is given by

$$N = \lim_{t\to\infty} \frac{1}{t}\int_0^t N(\tau)d\tau \text{ customers.}$$

Suppose that the nth customer stays in the system for T_n seconds, i.e. this is the delay experienced by the nth customer. Then the ***average customer delay*** is given by

$$T = \lim_{N\to\infty} \frac{1}{N}\sum_{n=1}^{N} T_n .$$

LITTLE'S THEOREM

Under very broad conditions, Little's theorem says that the following relationship holds:

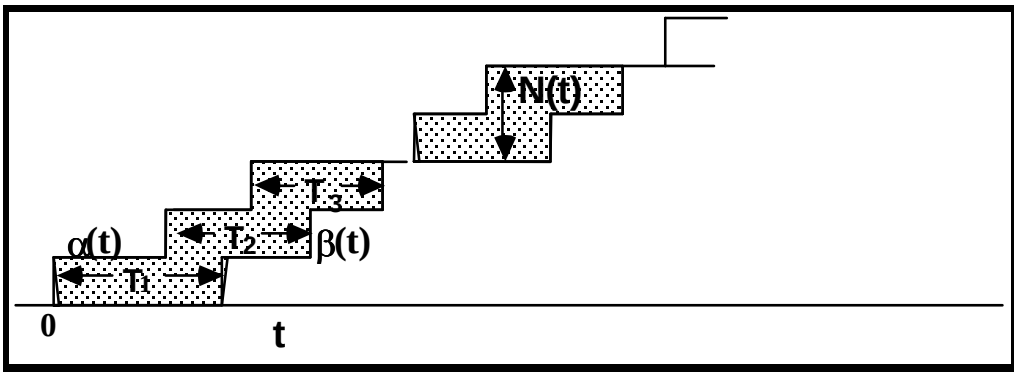
Average number of customers in system =
Customer arrival rate x Average customer delay

i.e. $\boldsymbol{N = \lambda T}$.

The "system" in this theorem can be many things including a queue, queue plus server, the entire network, server alone, etc.

Example: assume packets arrive at a node at 100 packets/sec, and there are on average 10 packets waiting in the node, then Little's theorem says that the average delay per packet is 0.1 sec.

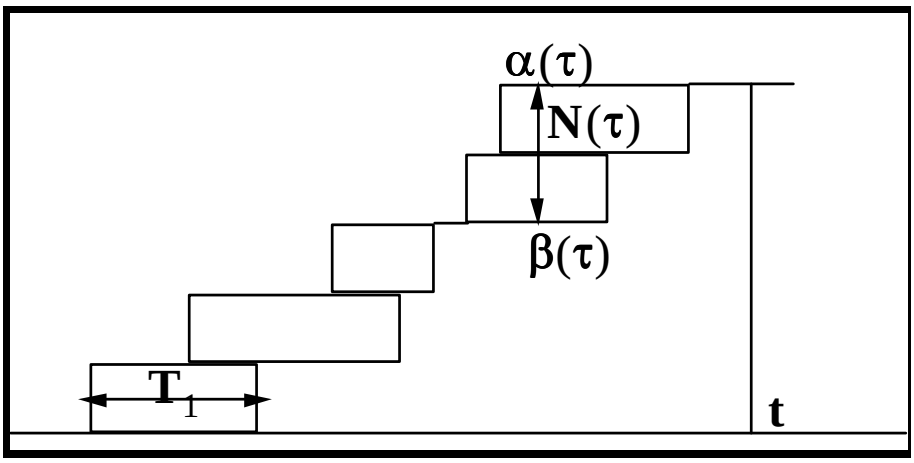
The following graphical argument gives an outline of the proof of Little's theorem. The argument is for a system that serves one customer at a time in FCFS order, however the theorem applies for much more general cases. Assume that the system is empty at time 0. The figure below shows the arrival process, $\alpha(t)$, (the top curve) and the departure process, $\beta(t)$ (the bottom curve). Since $N(t) = \alpha(t) - \beta(t)$, the vertical distance between these curves is the number in the system at any time *t*. Since customers are served FCFS, the delay, T_i, of customer *i* is the vertical distance between the curves as shown.



Consider some time $t>0$ when the systems is again empty. Let N_t be time average number of customers in the system from time 0 to t, *i.e.,*

$$N_t = \frac{1}{t}\int_0^t N(\tau)d\tau$$

The integral $\int_0^t N(\tau)d\tau$ is the area between the two curves in the figure. This area is also given by the sum of the boxes shown. Each box has height 1 and length T_n.



Thus we have

$$N_t = \frac{1}{t}\sum_{i=1}^{\alpha(t)} T_i$$

By multiplying and dividing the right-hand side by α(t) we have

$$N_t = \frac{\alpha(t)}{t}\frac{1}{\alpha(t)}\sum_{i=1}^{\alpha(t)} T_i$$

Now define the time-average arrival rate from 0 to t to be

$$\lambda_t = \frac{\alpha(t)}{t},$$

and the time-average time a customer is in the system as

$$T_t = \sum_{i=0}^{\alpha(t)} \frac{T_i}{\alpha(t)}.$$

Thus $N_t = \lambda_t T_t$ for any t at which system is empty.

Assuming that all three of these approach a limit as $t \to \infty$, we have $\boldsymbol{N = \lambda T}$.

Little's theorem doesn't give us N or T, but allows either to be found if the other can be found.

Examples:

Fast food restaurant (small T) requires small dining area (small N) for given λ.

On a rainy day, people drive more slowly (T is larger in a given area) and thus N is larger (if λ remains same).

We noted that Little's theorem can be applied to a variety of different "systems". We consider several possibilities next:

Consider a single queue followed by a single server.

Let N be the average number in system (queue plus transmission (service) time) and T the average delay in system; then from Little's theorem we have $N = \lambda T$.

Next, let N_Q be average number in queue (not including service) and W be the average delay in queue; then $N_Q = \lambda W$ from Little's theorem.

Finally, if the average transmission time on a link is $\overline{X}$, then the average number of packets under transmission, □□is given by

$$\square = \lambda \overline{X}$$

from Little again.

□ can be viewed as the ***utilization factor*** of a link. If packets are sent one at a time, then the number in transmission is either 0 or 1. The average number in transmission is then the probability that the link is occupied. In other words □ is the fraction of time the link is busy.