Lecture Notes 1
EE 549 — Queueing Theory
Instructor: Michael Neely

I. INTRODUCTION TO QUEUEING SYSTEMS

A modern data communication system consists of a network of information processing stations or nodes interconnected by data transmission channels or links. Information flows from node to node in packetized data units. These packets can originate from a variety of sources, and the paths over which they traverse the network can intersect, so that multiple packets are often intended for delivery over the same transmission link. When the number of packets in a particular node exceeds the service capabilities of the outgoing links of the node, some packets must either be dropped (resulting in a loss of information) or placed in a storage buffer or queue for future service. Queueing theory is the study of congestion and delay in such systems.

The dynamics of queueing systems are event driven, that is, the state of a queue changes based on discrete events such as packet arrivals or departures. Such systems are fundamentally different from the linear or nonlinear systems studied in classical control theory, where inputs are continuous functions of time (such as voltage or current waveforms) and where state dynamics are described by differential equations. Queueing systems are also very different from the point-to-point transmission models of physical layer communication, where inputs consist of band-limited signals that are corrupted by continuous background processes such as additive Gaussian noise. Thus, queueing systems require a different and unique set of mathematical tools. An understanding of these tools is essential in the analysis, design, and control of data networks.

In this course, we study queueing systems from both a deterministic and probabilistic perspective. In our deterministic analysis, we illuminate several simple properties and conservation laws exhibited by queueing systems with any arbitrary set of arrival times and packet sizes. We also develop several fundamental bounds on the worst case congestion and delay in networks when traffic is deterministically controlled so that the rate and burstiness of packet arrivals conforms to specified constraints. This deterministic analysis sets the stage for a probabilistic description of queueing systems. In this context, the arrival streams and service processes are assumed to be driven by simple probability laws, so that exact performance metrics can be calculated, such as the average delay in the network, or the probability distribution for the number of packets in a particular queue. The probabilistic study of event driven systems involves the theory of discrete stochastic processes. We shall find that several of the key results of discrete stochastic process theory, such as the ergodicity properties of Markovian networks as well as Little’s Theorem for average delay, establish a fundamental relationship between deterministic and probabilistic analysis of queueing networks.

II. A SINGLE SERVER QUEUE

Consider the simple queueing system of Fig. 1. The system is composed of a server and a storage buffer (often referred to as the queue). The server can hold a single packet, and “serves” the packet by performing whatever operation is necessary to process and transmit the packet to its recipient. The server can only accept a new packet when all previous packets have either completed their service (and thus have departed the system) or have had their service time preempted by a packet of higher priority (according to some pre-specified priority rule). The simplest priority rule, and the one that we shall mainly consider, is the First-In-First-Out (FIFO) priority rule. Under the FIFO rule, packets are served in the order in which they arrive, with no preemption. Packets that arrive when the server is busy are placed in the storage buffer to await service. The buffer and server together comprise the queueing station. For simplicity, we shall often shorten the term “queueing station” and refer to the system of buffer and server as “the queue.”

The dynamics of the queue are determined by the input stream $X(t)$ and the instantaneous server rate $\mu(t)$. Specifically, we have the following state variables:
that intervals. Note that the continuity property of \( Y \) place at time \( t \)
the amount of bits arriving during the half-open interval \((t_1, t_2]\)
the total bits is given by the sum of the bit lengths of all packets arriving during this interval. Let \( X \) work at a given time \( t \). A Basic Equalities and the Concept of
1 function increases continuously.

that is busy. Such intervals are called busy periods. A queue is defined to be work conserving if its service policy is to always serve at the full transmission rate whenever there is data present. Thus, in a work conserving queue, the server is idle if and only if the system is completely empty \((U(t) = 0)\). We say that \( \mu(t) \) is the potential processing rate at time \( t \) because the actual output rate of bits is zero during idle times. Queueing systems that are not work conserving have service policies that could leave the server idle even when \( U(t) > 0 \). Although we shall mainly consider work conserving policies, non-work conserving policies are also important as they arise in examples where it is desirable to save server energy, or to limit the output rate of the system for downstream nodes.

An example set of arrival, service rate, and unfinished work functions is provided in Fig. 2. Packets arrive at time instants \( \{t_1, t_2, \ldots\} \) and have corresponding bit lengths \( \{b_1, b_2, \ldots\} \). Note that the \( X(t) \) function is non-decreasing and changes in discontinuous jumps at packet arrival times. The height of the jump is equal to the length of the arriving packet. The \( Y(t) \) function is also non-decreasing. However, assuming that \( \mu(t) < \infty \) for all \( t \), the \( Y(t) \) function increases continuously.¹

A. Basic Equalities and the Concept of \( t^+ \) and \( t^- \)

Let \( X[t_1, t_2] \) represent the amount of bits arriving to a queueing system during the closed interval \([t_1, t_2]\), where the total bits is given by the sum of the bit lengths of all packets arriving during this interval. Let \( X(t_1, t_2) \) represent the amount of bits arriving during the half-open interval \((t_1, t_2]\), which does not include any arrivals that may take place at time \( t_1 \). Likewise, let \( Y[t_1, t_2] \) and \( Y(t_1, t_2) \) represent the corresponding bit departures during the given intervals. Note that the continuity property of \( Y(t) \) implies that \( Y[t_1, t_2] = Y(t_1, t_2) \) for any times \( t_1 \) and \( t_2 \) such that \( t_1 < t_2 \).

For any queueing system (potentially non-work conserving), the following basic equality relates the unfinished work at a given time \( t_1 \) to the unfinished work at a future time \( t_2 \):

\[
U(t_2) = U(t_1) + X(t_1, t_2) - Y[t_1, t_2]
\]

That is, the unfinished work at time \( t_2 \) is equal to the unfinished work at time \( t_1 \) plus the difference between bit arrivals and departures during the interval between \( t_1 \) and \( t_2 \). It is often convenient to re-write the above equality

¹Generalizations to \( \mu(t) \) functions that contain delta impulses can also be treated, in which case a fixed amount of data can be delivered at the instant of the impulse. Such generalizations do not considerably alter the analysis, but for simplicity of exposition they shall not be treated.
in terms of $X[t_1, t_2]$ by considering the unfinished work in the system just before time $t_1$:
\[
U(t_2) = U(t_1^-) + X[t_1, t_2] - Y[t_1, t_2]
\]

As suggested by the above equality, the notation $t^-$ is used to represent a time instant “just before” time $t$. Similarly, the notation $t^+$ represents a time instant “just after” time $t$. Formal definitions of $U(t^-)$ and $U(t^+)$ are given below:
\[
U(t^-) = \lim_{\delta \to 0^+} U(t - \delta)
\]
\[
U(t^+) = \lim_{\delta \to 0^+} U(t + \delta)
\]

The limiting notation above means that the limit is taken for $\delta > 0$, and therefore $\delta$ approaches 0 from the right hand side of real number line.

Note that the basic equality (2) also holds for any data storage system with inputs and outputs, such as multi-server queues or multi-queue networks. Specifically, if $X_{general}(t)$ and $Y_{general}(t)$ represent the aggregate bit arrival and bit departure functions of the general system, and if $U_{general}(t)$ represents the sum of the unprocessed bits in the entire system, then we again have for any times $t_1 < t_2$:
\[
U_{general}(t_2) = U_{general}(t_1^-) + X_{general}[t_1, t_2] - Y_{general}[t_1, t_2]
\]

**B. A Simple Inequality for $Y[t_1, t_2]$**

For any single server queueing system with server process $\mu(t)$ and for any two times $t_1, t_2$ such that $t_1 < t_2$, we necessarily have:
\[
Y[t_1, t_2] \leq \int_{t_1}^{t_2} \mu(t)dt
\]

That is, the total bits processed during any interval $[t_1, t_2]$ must be less than or equal to the integral of the transmission rate over this interval. The inequality above is given by an equality whenever the queueing system is work conserving and the system is busy for the full duration of the interval $[t_1, t_2]$, because in such a case the queue processes bits at the full transmission rate for the entire interval.

**C. Work Conserving Queues and Busy Periods**

Here we consider work conserving queues. For these queues, busy periods correspond to intervals when the queue is not empty. Consider any time $t$ such that $U(t) > 0$, and let $t_b$ represent the starting time of the current busy period. It follows that the system was empty just before time $t_b$, and hence from (2) we have:
\[
U(t) = X[t_b, t] - Y[t_b, t]
\]

However, because the system is busy during the entire interval $[t_b, t]$, the bit departures during this interval are equal to the integral of the service rate. We thus have the following identity that holds at any time $t$ such that $U(t) > 0$:
\[
U(t) = X[t_b, t] - \int_{t_b}^{t} \mu(\tau)d\tau
\]

The unfinished work function $U(t)$ in a work conserving queue depends only on the input function $X(t)$ and the server rate function $\mu(t)$. In particular, it is invariant for all work conserving service disciplines, regardless of the order in which packets are served.

We note that the process $Y(t)$ does not necessarily describe the output process of the queue as seen by the destination. Indeed, it is reasonable to define the time at which a packet reaches its destination as the time in which the last bit of that packet is served. To illustrate this point, we define the following additional state variables:

- $N(t) =$ Number of packet arrivals during the interval $[0, t]$
- $D(t) =$ Number of packet departures during the interval $[0, t]$
- $L(t) = N(t) - D(t) =$ Number of packets in the queueing system at time $t$
The $N(t)$ function is integer valued and increases by 1 at packet arrival times. The $D(t)$ function is also integer valued. If the arriving packets have different lengths, then $D(t)$ is not completely determined by the arrival function and the service rate function, as it also depends on the priority policy determining the order in which packets are served. Thus, the First-In-First-Out (FIFO) policy may produce a different $D(t)$ function than the Last-In-First-Out (LIFO) service discipline.

**D. The Dual Nature of the Queueing Processes**

Up to this point, we have treated $X(t)$ and $\mu(t)$ as given functions of time. This approach does not specify how these functions are generated, and says nothing about the “typical” behavior of these functions. In the case when these functions are generated according to a given probability law, then they can be viewed as stochastic processes, that is, random functions of time. In this light, the resulting unfinished work function $U(t)$ can also be viewed as a stochastic process. Indeed, its joint probability distribution $p(u_1, u_2, \ldots, u_k)$ at any specified set of times $(t_1, t_2, \ldots, t_k)$ is determined by the stochastics of the $X(t)$ and the $\mu(t)$ processes.

In contrast to the deterministic setting, this stochastic setting allows probabilistic quantities to be defined. For example, one can talk about $\mathbb{E}\{X(t)\}$, the expected number of bits to arrive during $[0, t]$, or $\Pr[U(t) > 15]$, the probability that the unfinished work at time $t$ is greater than 15. These quantities have no meaning in the deterministic context. However, note that any particular realization, or sample path, of the stochastic processes $X(t)$ and $\mu(t)$ are in fact ordinary, deterministic functions of time. Thus, every property associated with $X(t)$, $\mu(t)$, and $U(t)$ when viewed as deterministic functions of time must be inherited by every sample path of $X(t)$, $\mu(t)$, and $U(t)$ when viewed as stochastic processes. For example, if we know that all sample path functions $U(t)$ of a certain queueing system satisfy $U(t) \leq 10$ kilobits for all time $t$, then we must also have $\mathbb{E}\{U(t)\} \leq 10$ kbits for all time $t$. A simple example of this is a FIFO queue with a constant processing rate $\mu(t) = \mu$, and which
receives fixed length packets from a preliminary queue that also serves at a constant rate \( \mu \). In this case, there is never more than one packet in the downstream queue at any instant of time.

The simplest example of a probabilistic law describing an input process \( X(t) \) is a renewal process, where the inter-arrival times \( T_i \) between successive packets are independent but identically distributed (i.i.d) with distribution function \( p_T(t) \), and the packet lengths \( B_i \) are i.i.d. with distribution function \( p_B(b) \). Specifically, the first arrival occurs at time \( T_1 \), where \( T_1 \) is a positive random variable with distribution function \( p_T(t) \). This packet has size \( B_1 \). The second packet arrives at time \( T_1 + T_2 \) (where \( T_1 \) and \( T_2 \) are i.i.d. random variables) and has size \( B_2 \) (where \( B_1 \) and \( B_2 \) are i.i.d. random variables with distribution \( p_B(b) \)). This is called a “renewal process” because the probability law that determines future arrivals “renews” itself at packet arrival instants.

III. SAMPLE PATH PROPERTIES

Here we develop simple but useful sample path properties that hold for queueing systems with arbitrary input and server processes.

A. An Elementary Monotonicity Property

Consider a queue with input process \( X(t) \) and server process \( \mu_1(t) \). Let \( U_1(t) \) represent the resulting unfinished work process. Let \( U_2(t) \) represent the unfinished work process in another queue with the exact same input \( X(t) \), but with a potentially different server process \( \mu_2(t) \). Both queues are work conserving and are assumed to be empty at time \( t = 0 \).

**Lemma 1:** If \( \mu_1(t) \geq \mu_2(t) \) for all \( t \geq 0 \), then \( U_1(t) \leq U_2(t) \) for all \( t \geq 0 \).

The above lemma is quite intuitive: It states that, all else being equal, the unfinished work in the queue with the smaller server rate will always be at least as much as the unfinished work in the queue with the larger server rate. We can formally prove this lemma by using the sample path equalities of Sections II-A and II-C:

**Proof:** (Lemma 1) Consider any time \( t \geq 0 \). If \( U_1(t) = 0 \), then clearly \( U_1(t) \leq U_2(t) \), and we are done. Now assume \( U_1(t) > 0 \), and let \( t_b \) represent the start of the current busy period in this queue. We thus have:

\[
U_1(t) = X[t_b,t] - \int_{t_b}^t \mu_1(\tau)d\tau
\]

which follows from (4). Using (2), we can also write \( U_2(t) \) in terms of its value at the start of the busy period for the first queue:

\[
U_2(t) = U_2(t_b) + X[t_b,t] - Y_2[t_b,t]
\]

where \( Y_2[t_b,t] \) denotes the bits served in the second queue during the interval \([t_b,t]\). We clearly have \( U_2(t_b) \geq 0 \) and \( Y_2[t_b,t] \leq \int_{t_b}^t \mu_2(\tau)d\tau \), and hence:

\[
U_2(t) \geq X[t_b,t] - \int_{t_b}^t \mu_2(\tau)d\tau \\
\geq X[t_b,t] - \int_{t_b}^t \mu_1(\tau)d\tau \\
= U_1(t)
\]

proving the lemma.

B. Unit Scaling

Consider a queue with input stream \( X(t) \) and server function \( \mu(t) \). Assume the queue is initially empty, and let \( \bar{U}(t) \) represent the unfinished work as a function of time. Fix any positive constant \( V \), and now consider the process \( \bar{U}(t) \) that represents the unfinished work in a new queueing system with input stream \( V X(t) \) and server process \( V \mu(t) \). That is, the bit lengths of all arriving packets to the new system are scaled by a factor of \( V \), as is the transmission rate process.

**Lemma 2:** (Unit Scaling Equality) For any constant \( V > 0 \), we have:

\[
\bar{U}(t) = VU(t) \quad \text{for all time } t \geq 0
\]
The unit scaling property holds because scaling both the input and server process by a constant $V$ scales the $U(t)$ sample path by $V$ at every instant of time (see Fig. 3). For example, in a virtual system where only half of every packet arrives to the queue, and the other half is thrown away, a server function $\tilde{\mu}(t) = \frac{1}{2}\mu(t)$ yields an unfinished work function $\tilde{U}(t) = U(t)/2$. The operation of scaling both $X(t)$ and $\mu(t)$ by the same constant $V > 0$ can be viewed simply as expressing the backlog in units other than bits. This lemma is quite intuitive, and we omit a formal proof.

C. The Multiplexing Inequality

Here we compare a system of $K$ queues with input processes $X_1(t), \ldots, X_K(t)$ and server rate processes $\mu_1(t), \ldots, \mu_K(t)$ to a single queue with a combination of all $K$ inputs (resulting in a total input process $X(t) = \sum_{k=1}^{K} X_k(t)$), and with a server rate $\mu(t)$ equal to the sum of the individual server rates of the multi-queue system. That is, we have $\mu(t) = \sum_{k=1}^{K} \mu_k(t)$ (see Fig. 4). Both systems are assumed to be initially empty. Packets are processed in the single queue system according to any work conserving service discipline, while the service discipline in the multi-queue system is arbitrary and not necessarily work-conserving. Let $U_{single}(t)$ represent the unfinished work (in bits) in the single server queue, and let $U_{multi}(t)$ represent the total amount of unfinished bits in the multi-queue system.

Lemma 3: (Multiplexing Inequality): For all time $t \geq 0$:

$$U_{single}(t) \leq U_{multi}(t)$$  \hspace{1cm} (5)

Proof: Consider any time $t$.

Case 1: $U_{single}(t) = 0$.
In this case, we have $U_{single}(t) = 0 \leq U_{multi}(t)$, and hence the result trivially holds.

Case 2: $U_{single}(t) > 0$.
In this case, there must be a time $t_b$ that initiated the current busy period in the single queue system, and hence by (4) we have:

$$U_{single}(t) = X[t_b, t] - \int_{t_b}^{t} \mu(\tau)d\tau$$  \hspace{1cm} (6)

However, for the multi-server system at time $t$, we have:

$$U_{multi}(t) = U_{multi}(t_b^{-}) + X[t_b, t] - Y_{multi}[t_b, t]$$  \hspace{1cm} (7)
$$\geq X[t_b, t] - Y_{multi}[t_b, t]$$  \hspace{1cm} (8)
$$\geq X[t_b, t] - \int_{t_b}^{t} \mu(\tau)d\tau$$  \hspace{1cm} (9)

where (7) follows from the basic queueing equality (3), the inequality (8) follows from the fact that the total bits processed by the multi-server system over a particular interval is necessarily less than or equal to the sum of the
This multiplexing inequality demonstrates that, from the perspective of unfinished work, it is always better to multiplex data streams from individual queues to a single queue whose rate is equal to the sum of the individual processing rates. It is useful to consider such a virtual single-server queue to provide a baseline for measuring the performance of routing policies in cases where packets arrive to a set of parallel queues, and must be routed to one of the queues upon arrival.

IV. THE JITTER THEOREM

Here we use the Scaling Equality and the Multiplexing Inequality, to prove the “Jitter Theorem,” which states that, under certain conditions, the expected unfinished work in a queue with a time varying server rate is greater than or equal to the expected unfinished work in a system with the same input process but with the server rate replaced by a constant rate server of rate equal to the average value of the time varying server. To begin, we must define the notion of a stationary stochastic process.

A stochastic process is a random function of time that can be simulated according to a specified probability law. An example of a stochastic arrival process $X(t)$ was given in Section II-D. An example of a stochastic server rate process is provided below.

**Example:** Suppose the $\mu(t)$ server rate holds its value constant on integral intervals of time, and changes randomly on the timeslot boundaries $t = 0, 1, 2, \ldots$. Let $\mu_i$ represent the value of $\mu(t)$ during the interval $[i, i+1)$, and suppose the $\mu_i$ values are i.i.d. Bernoulli random variables, where $\mu_i = 1$ with probability 0.4, and $\mu_i = 0$ with probability 0.6. Then $\mu(t)$ is a stochastic process.

**Definition 1:** A stochastic process $\mu(t)$ is stationary if it is statistically equivalent to the process $\mu(t - T)$ for any constant timeshift $T$. That is, the joint distribution function satisfies:

$$Pr[\mu(t_1) \leq V_1, \ldots, \mu(t_k) \leq V_k] = Pr[\mu(t_1 - T) \leq V_1, \ldots, \mu(t_k - T) \leq V_k]$$

---

This section is optional. Students will not be held responsible for material in this Section IV.
for any finite set of times $t_1, \ldots, t_k$, any real values $V_1, \ldots, V_k$, and for any timeshift $T$.

The $\mu(t)$ process given in the example above is not stationary, because its statistical behavior over the interval $[0, 1)$ is dramatically different from its statistical behavior over the interval $\left[ \frac{1}{3}, \frac{3}{2} \right]$. Indeed, it must be a constant over the interval $[0, 1)$, whereas it need not be constant over the interval $\left[ \frac{1}{3}, \frac{3}{2} \right]$. However, it can be shown that shifting this process by a random phase that is uniformly distributed over the unit interval yields a stationary process.

Assume $\mu(t)$ is a stationary random process, and define $\mu_{av} \triangleq \mathbb{E} \{ \mu(0) \}$. Note that, by stationarity, it follows that $\mathbb{E} \{ \mu(t) \} = \mu_{av}$ for any time instant $t$.

Consider a work conserving queue with an input process $X(t)$ and a server process $\mu(t)$ that is stationary and independent of $X(t)$. Let $U(t)$ represent the unfinished work in this queue at time $t$. Let $\hat{U}(t)$ represent the unfinished work in a work conserving queue with the same input process but with a constant server of rate $\mu_{av}$.

**Theorem 1:** (Jitter Theorem) At every instant of time, we have:

$$\mathbb{E} \{ U(t) \} \geq \mathbb{E} \{ \hat{U}(t) \}$$

Thus, any time varying jitter in the linespeed process creates extra queue congestion.

The proof of the Jitter Theorem follows directly from the unit scaling equality and the multiplexing inequality together with the following picture. From the figure, we note by the unit scaling property that the unfinished work $U(t)$ at every instant of time is equal to the sum unfinished work when the system is duplicated $M$ times, with scaled inputs and server rates $\frac{1}{M}X(t)$ and $\frac{1}{M}\mu(t)$. Next, note that the expected unfinished work in each duplicate system $m$ is equal to the expected unfinished work in a modified system $m'$ with the same input process $\frac{1}{M}X(t)$ but with the server process replaced by $\frac{1}{M}\mu_{m}(t)$, where $\mu_{m}(t)$ is an independent but identically distributed version of the original $\mu(t)$ process. This holds because the original $\mu(t)$ process is independent of the input stream $X(t)$. However, applying the multiplexing inequality, we find that the sum unfinished work in all modified queues is greater than or equal to the unfinished work in a single queue with an input stream $X(t) = \frac{1}{M} \sum_{m=1}^{M} X(t)$ and a server process $\frac{1}{M} \sum_{m=1}^{M} \mu_{m}(t)$. Thus, the expected unfinished work in the original system is greater than or equal to the expected unfinished work in a new system with the same input stream but with a server process consisting of a sum of $M$ i.i.d. processes $\mu_{m}(t)$. This holds for all positive integers $M$, and hence we can take limits as $M \to \infty$. Because all processes are stationary and independent, it follows by the law of large numbers (applied to processes) that $\frac{1}{M} \sum_{m=1}^{M} \mu_{m}(t) \to \mu_{av}$.  

---

$^3$It may be of interest to note the manner in which $\frac{1}{M} \sum_{m=1}^{M} \mu_{m}(t) \to \mu_{av}$. Note that for any value of $t$, the random variable $\frac{1}{M} \sum_{m=1}^{M} \mu_{m}(t)$ converges to $\mu_{av}$ with probability 1 by the law of large numbers. Alternatively, assume that at any time $t$, the variance of $\mu(t)$ is equal to $\sigma^2$. It is not difficult to show that for any interval of time $[0, T]$, the value of $\mathbb{E} \left\{ \int_{0}^{T} \left( \frac{1}{M} \sum_{m=1}^{M} \mu_{m}(t) - \mu_{av} \right)^2 dt \right\}$ converges to 0 as $M \to \infty$.  


Fig. 5. An illustrative proof of the Jitter Theorem