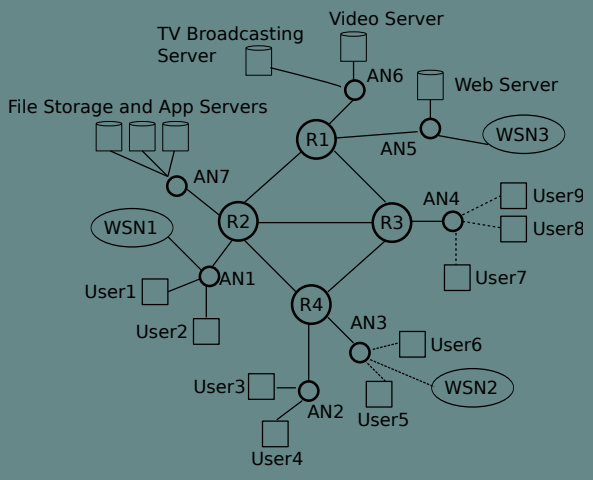
Universitat Pompeu Fabra Barcelona

Analysis of Packet Queueing in Telecommunication Networks

Course Notes. (BSc.) Network Engineering, 2nd year



Boris Bellalta and Simon Oechsner

.

Work in progress!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Modern telecommunication systems influence our daily lives to a large degree. They do not only enable us to reach each other or to access information virtually anywhere, but are also an enabler for a significant part of modern economics. IT networking infrastructures increase the productivity of companies and impact not only the flow of information, but are used to manage the flow of physical products as well. It is therefore worthwhile to study and understand how the infrastructure underlying these systems works.

This course wants to provide understanding of a specific topic in telecommunication systems, namely the analysis of data packet flows. This knowledge is very useful especially for traffic and network engineers, because it allows them to describe existing or potential future networks, to identify problems or to dimension a system.

Specifically, the focus of the course will be on simple analytical tools originating from the field of queueing theory that have an application specifically in modern telecommunication networks. Therefore, although the theoretical part could be applied to any kind of queues (such as customers waiting at a check-out counter), we will always establish a connection to specific features of telecommunication networks and give examples for the application of the presented concepts in this field. These should allow a student to train the use of these methods, and later to apply them

to other problems from practice.

However, we would like to add two disclaimers at this point. The first is that these notes are not meant to be a comprehensive discussion of queueing theory. Very good and exhaustive literature exists on this topic, such as the suggested in next section, so that we do not need to add redundant information. Instead, we use the concepts from these works that are most relevant to modern telecommunication networks and apply them in this context.

Second, we do not claim that the methods presented here are the only tool needed by a network engineer or planner. We view them more as one item in a larger toolbox. Each of the tools in this box has its use and excels for specific jobs, while others might be better in different situations. Thus, methodologies such as measurements of live networks or simulation should be seen as complementing to the analytical approach we focus on here.

For example, while network measurements and the evaluation of system logs can provide very detailed information from real systems, it is often very resource-consuming to test a large number of different configurations. Real equipment has to be set up and configured for each measurement run, making it relatively costly to obtain results. On the other hand, analytical formulas might provide results for a large set of different scenarios and parameter settings in a very short time, allowing to explore a solution space much quicker. However, analytical methods often make simplifying and partially unrealistic assumptions, and thus produce results that might not be seen exactly like this in a real system.

1.2 Recommended Books

These are the books we recommend:

- Bertsekas, Dimitri P., Robert G. Gallager, and Pierre Humblet. **Data networks**. Vol. 2. New Jersey: Prentice-Hall International, 1992.
- Gross, Donald. **Fundamentals of Queueing Theory**. John Wiley & Sons, 2008.
- Kleinrock, Leonard. "**Queueing Systems, volume I: theory.**" (1975).

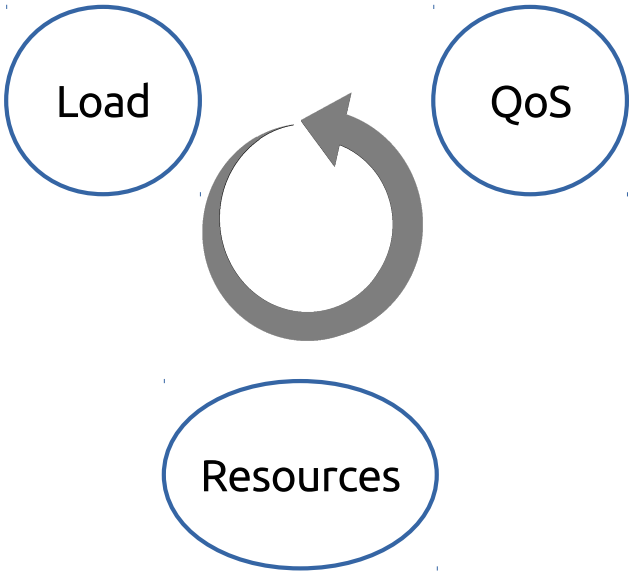


Figure 1.1: The three related pillars in system dimensioning

Part I

Preliminaries

Chapter 2

Random Phenomena in Packet Networks

2.1 Introduction

A phenomenon is 'random' when we do not know how it will behave in the future. For instance, to toss a coin is a random phenomenon as we can not know in advance the result of it. However, we can try to characterize it by estimating those aspects that do not change with the time. For instance, in the case we toss a coin, we can estimate the probability to obtain a head and a cross by simply tossing it several times, and counting the number of occurrences of each type.

Another example is the data exchange in communication networks. When and how we will communicate with others is unknown, as well as the data we will send to the other side. Random phenomena is common in the communication process and a key phenomena in communication networks. Examples of random phenomena in telecommunications networks are:

- The instant of time that packets are generated at the transmitter, which depends on the user's activity (i.e. clicking links on a web page, talking in a voice call, etc.).
- The size of the packets that are transmitted over a link, which depend on the specific contents exchanged at that moment.

- The capacity of a wireless link when nodes move.
- The number of users that transmit data at the same time over the same link.

Therefore, we need to understand the 'randomness' present in telecommunication networks to be able to characterize their performance.

The random variables we will consider in this text will change with the time (i.e., take new values). In that case, the stochastic process behind the random variable is called a *stochastic process*. In addition, we will assume that all the stochastic processes considered are *stationary*.

2.2 Characterizing a random variable

Let X be a random variable that takes values from the set $\mathcal{X}$, which is called the state space of X. Each possible value $x \in \mathcal{X}$ has assigned a probability, and will be referred as $\mathrm{P}\{X = x\}$. For an stationary stochastic process that is stationary, the $\mathrm{P}\{X = x\}$ does not change with the time.

A first consideration about X is related with $\mathcal{X}$:

- If the range of values that X can take is finite, we say that X is a random variable with a **discrete state space**.
- If the range of values that X can take is infinite, we say that X is a random variable with a **continuous state space**.

A second consideration is when the random variable X takes a new value.

- If X can take a new value at any arbitrary time, we say that X is a **continuous time** random variable.
- If X can take a new value only at specific instants of time, we say that X is a **discrete time** random variable.

Finally, a third consideration is about the dependence between present, past and future values.

- We will say that the stochastic process that generates X is an **independent stochastic process**.

- If there are some dependencies between the present and past or future values, we will refer to it as a **dependent stochastic process**.

In all cases, to characterize the random variable X we will focus on the following metrics:

- Histogram
- Expected Value
- Variance
- Coefficient of Variation
- Moments

2.2.1 Histogram

The histogram is a function that given a value of X, x (discrete), or a range of values of X, $[x_1, x_2]$ (continuous), gives the probability that the x value, or a value inside the chosen range of values, appears. In detail:

- If X has a discrete state space, the histogram of X is the $P\{X = x\}$, $\forall x \in \mathcal{X}$.
- If X has an infinite state space, the histogram of X is the $P\{x_1 < X \leq x_2\}$, $\forall x_1,\ x_2 \in \mathcal{X}$.

Example: In Figure 4.1, let us assume that the Video Server contains 1020 videos encoded in AVI and 200 videos encoded in MPEG4. Let X be the random variable that models the next video to be requested. Assuming that all videos have the same probability to be requested, write the histogram of X.

Solution: The state space of X, and therefore also the histogram, contains two values: AVI and MPEG4. As the histogram is the probability that the next video requested belongs to the AVI or MPEG4 category, and as all videos have the same probability to be requested, the histogram is simply the probability that an AVI or an MPEG4 video is requested. Therefore,

$$P\{X = \text{AVI}\} = \frac{1020}{1220} = 0.836$$
$$P\{X = \text{MPEG4}\} = \frac{200}{1220} = 0.164$$

Exercise: Find the new histogram if the video server now includes 500 H.264 encoded videos.

Example: The time between two packets arriving to R1 from R2 is a random variable τ that follows an exponential distribution with parameter $\lambda = 4$ packets/second, $f_\tau(t) = \lambda e^{-\lambda t}$. What is the probability that τ takes a value between 1 and 1.25 seconds?

Solution: To obtain the requested probability, we can simply solve next integral:

$$P\{1 < X \leq 1.25\} = \int_1^{1.25} 4e^{-4t}dt \tag{2.1}$$

For that, we can use the cumulative distribution function (cdf) of the exponential distribution, as previous probability can be written in terms of the cdf as follows:

$$P\{1 < X \leq 1.25\} = F_\tau(1.25) - F_\tau(1) \tag{2.2}$$

where $F_\tau(t_1) = P\{X \leq t_1\} = 1 - e^{-\lambda t}$. Therefore,

$$P\{1 < X \leq 1.25\} = F_\tau(1.25) - F_\tau(1) = e^{-4\cdot 1} - e^{-4\cdot 1.25} = 0.011578 \tag{2.3}$$

Exercise: For the previous example, compute the histogram if the intervals are $\{(0, 2], (2, 4], (4, 6], (6, \infty)\}$.

2.2.2 Expected Value

If the random variable is discrete, we can compute its expected value, $E[X]$, as follows:

$$E[X] = \sum_{\forall x \in \mathcal{X}} xP\{X = x\} \tag{2.4}$$

In case the random variable is continuous, we have to use its pdf:

$$\mathrm{E}[X] = \int_{-\infty}^{+\infty} x f_X(x) dx \tag{2.5}$$

In both cases, the expected value is an indicator of the mean value that the random variable can take.

2.2.3 Variance

The variance is an indicator of the dispersion of the values that X can take. If the random variable X is discrete, the variance can be calculated as follows

$$\mathrm{V}[X] = \sum_{\forall x \in \mathcal{X}} P\{X = x\} \left(x - \mathrm{E}[X]\right)^2 \tag{2.6}$$

In case the random variable X is continuous, the variance is:

$$\mathrm{V}[X] = \int_{-\infty}^{+\infty} f_X(x) \left(x - \mathrm{E}[X]\right)^2 dx \tag{2.7}$$

2.2.4 Coefficient of Variation

The coefficient of variation is a metric that depends on the both the variance and the expected value. It is computed as:

$$\mathrm{CV}[X] = \frac{\sqrt{\mathrm{V}[X]}}{\mathrm{E}[X]} \tag{2.8}$$

Note that $\sqrt{\mathrm{V}[X]}$ is known as the standard deviation of X.

2.2.5 Moments

The mth moment of a random variable X is defined as:

$$\mathrm{E}[X^m] = \sum_{\forall x \in \mathcal{X}} x^m P\{X = x\} \tag{2.9}$$

in case it is discrete. If it is continuous, it is defined as:

$$\mathrm{E}[X^m] = \int_{-\infty}^{+\infty} x^m f_X(x) dx \tag{2.10}$$

Note that the first moment is the expected value and that the variance can be easily obtained from the first and second moments:

$$\mathrm{V}[X] = \mathrm{E}[X^2] - \mathrm{E}^2[X] \tag{2.11}$$

2.3 Stochastic Processes with Independent and Dependent Outcomes

In this document we only consider stationary processes. For stationary processes we refer to those stochastic processes in which their parameters are constant during the time (i.e. the state space does not change, and the probability of each possible value is constant with the time).

We can classify the stochastic stationary processes in two groups:

Independent outcomes : Given a certain outcome, it is completely independent from past or future outcomes. For example, each time we toss a coin we get an independent outcome. In other words, the fact that we got a cross in last attempt does not influence the result in next attempt.

Dependent outcomes : Given a certain outcome, it depends on previous or future outcomes. For example, if we capture the packets transmitted over a link, after seeing a SYN TCP packet, the probability that one of the next packets is a SYN ACK TCP packet is very high, as the later is always transmitted after the first one (i.e., there is a dependence).

One class of stochastic process with dependent outcomes are the Markov processes. A Markov process is a process that satisfies the Markov property, which simply states that next outcome will only depend on the present outcome. Formally,

$$Pr\{X_n = x_n | X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \ldots, X_0 = x_0\} = Pr\{X_n = x_n | X_{n-1} = x_{n-1}\} \tag{2.12}$$

As we will explain in next chapter, Markov processes are suitable to model telecommunications systems.

We will refer to a stochastic process as $X(t)$, with t showing the dependence on the time.

2.4 Examples

Example 1: Number of packets received by Router R1 in intervals of Δ seconds

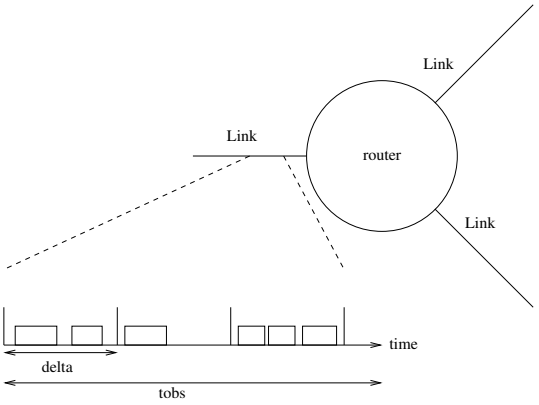


Figure 2.1: Measuring the transmitted packets between R2 and R1.

Consider the router R1 in Figure 4.1, depicted also in Figure 2.1. We measure the number of packets that arrive to R1 from R2, and count the number of packets that arrive to R1 in intervals of Δ seconds. The observation time is $T_{\text{obs}} = 10\Delta$, and the collected data is shown in Table 2.1.

Interval (Δ)	Packets (X)	Interval	Packets (X)
1	5	6	2
2	4	7	1
3	4	8	5
4	3	9	3
5	1	10	1

Table 2.1: Results from the experiment

As it can be observed, the number of packets that arrive at each interval Δ is a random variable, and we call it X. Now, we are going to characterize X by computing its *histogram*, *expected value* and *variance*.

Solution:

To obtain the histogram we need to compute the probability $\mathrm{P}\{X = x\}$, for all possible values of x, i.e., $\forall x \in \mathcal{X}$.

$$\mathrm{P}\{X = x\} = \frac{\text{Number of samples equal to } x}{N} \tag{2.13}$$

where N is the total number of samples. In our case, $N = 10$. We can observe that X only takes five different values: $\{1,\ 2,\ 3,\ 4,\ 5\}$. The resulting histogram is depicted in Table 2.2.

x	Appearances	Probability
1	3	3/10
2	1	1/10
3	2	2/10
4	2	2/10
5	2	2/10

Table 2.2: Histogram

To compute the expected value of X, we use (2.4).

$$E[X] = \sum_{\forall x \in \mathcal{X}} x\mathrm{P}\{X = x\} = \tag{2.14}$$

$$= \frac{3}{10} + 2\frac{1}{10} + 3\frac{2}{10} + 4\frac{2}{10} + 5\frac{2}{10} = \frac{29}{10} = 2.9 \text{ packets} \tag{2.15}$$

Now, we compute the variance using (2.6).

$$\mathrm{Var}[X] = \sum_{\forall x \in \mathcal{X}} \mathrm{P}\{X = x\}(x - E[X])^2 = \tag{2.16}$$

$$= (1 - 2.9)^2\frac{3}{10} + (2 - 2.9)^2\frac{1}{10} + (3 - 2.9)^2\frac{2}{10} + \tag{2.17}$$

$$+ (3 - 2.9)^2\frac{2}{10} + (3 - 2.9)^2\frac{2}{10} = 2.29 \text{ packets}^2 \tag{2.18}$$

From the variance and the expected value, we compute the coefficient of variation:

$$\mathrm{CV}[X] = \frac{\sqrt{\mathrm{Var}[X]}}{E[X]} = 0.5218 \tag{2.19}$$

Observe that the variance can be computed using the first and second moment,

$$\mathrm{Var}[X] = E[X^2] - E^2[X] \tag{2.20}$$

where the second moment is

$$\mathrm{E}[X^2] = \sum_{\forall x \in \mathcal{X}} x^2 \mathrm{P}\{X = x\} = \tag{2.21}$$

$$= \frac{3}{10} + 2^2\frac{1}{10} + 3^2\frac{2}{10} + 4^2\frac{2}{10} + 5^2\frac{2}{10} = 10.7 \text{ packets}^2 \tag{2.22}$$

Example 2: Distribution of the time between two consecutive packets and distribution of the packet sizes.

In this second case, we focus on the time elapsed between two consecutive packets that arrive to R1 from R2. This time is represented by a random variable T, and shown in Figure 2.3, where τ are specific outcomes of T. As it is observed T is a continuous random variable that can take any value in the range $[0, \infty)$. Additionally, it can be observed in Figure 2.3 that the packets have different packet sizes (l) as the packet size also follows a random variable (L).

First, we capture 100 packets from the link that connects R1 and R2. The time between two consecutive packets received at R2 is shown in Figure 2.2(a). In Figure 2.2(b) we plot the histogram computed from the measured data, as well as the theoretical histogram assuming that the time between two consecutive packets follows and exponential distribution. As it can be observed, both histograms match very well, confirming that assumption: that the time between two packets follows and exponential distribution. The measured expected time between two packets is $\mathrm{E}[T] = 0.1$ seconds. In that case, the variance of the time between two consecutive packets is:

$$\mathrm{V}[T] = (\mathrm{E}[T])^2 = 0.01 \ seconds^2 \tag{2.23}$$

as we know that the variance of a random variable that follows and exponential distribution is the square of the average. In addition, the coefficient of variation is

$$\mathrm{CV}[L] = \frac{\sqrt{\mathrm{V}[T]}}{\mathrm{E}[T]} = 1 \tag{2.24}$$

In all cases, if a continuous random variable follows an exponential probability density function, its coefficient of variation is 1.

Regarding the packet sizes, we guess that L is a random variable that follows an uniform distribution with $L_{\min} = 8000$ bits and $L_{\max} = 128$ bits as the maximum and minimum values it can take. In the case our guess is true, and L follows an uniform distribution, the average value of L is

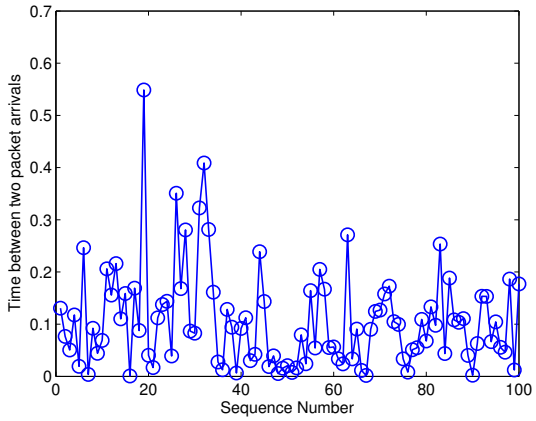
$$\mathrm{E}[L] = \frac{L_{\min} + L_{\max}}{2} = 4064 \text{ bits} \tag{2.25}$$

and the variance

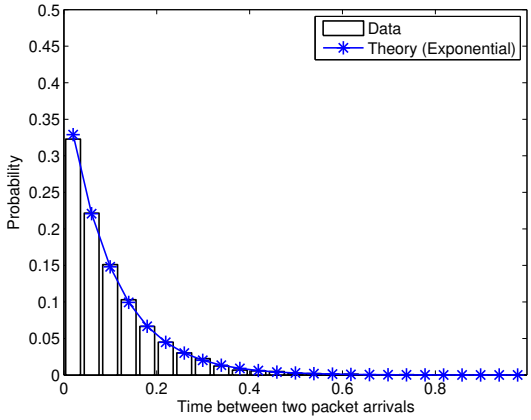
$$\mathrm{V}[L] = \frac{(L_{\max} - L_{\min})^2}{12} = 5164032 \text{ bits}^2 \tag{2.26}$$

Finallly, the coefficient of variation is:

$$\mathrm{CV}[L] = \frac{\sqrt{\mathrm{V}[L]}}{\mathrm{E}[L]} = 0.55917 \tag{2.27}$$



(a)



(b) $N = 1000$ samples

Figure 2.2: Time between two packets: Temporal series and Histogram

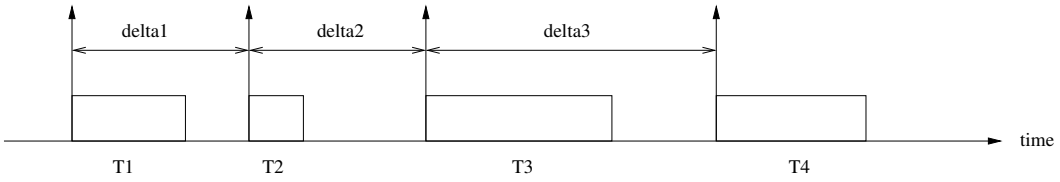


Figure 2.3: Time between two packets (τ). The values of l represent the packet sizes.

2.5 Formulation of independent and dependent processes

- Independent processes: from each state we can move to any other state. The transition probabilities are exactly the same regardless the state, and corre-

$X(t_i)$	$X(t_{i+1})$	Probabilities
0	0	p_0
	1	p_1
	2	p_2
1	0	p_0
	1	p_1
	2	p_2
2	0	p_0
	1	p_1
	2	p_2

Table 2.3: Independent process formulation

$X(t_i)$	$X(t_{i+1})$	Probabilities
0	0	1-p
	0	p
1	0	q
	1	1-q-p
	2	p
2	1	q
	2	1-q

Table 2.4: Dependent process formulation

spond to the stationary probability distribution. Example in Table 2.3.

- Dependent processes: transition probabilities indicate the dependence. They do not correspond to the transitio probability. Example in Table 2.4.

Chapter 3

Markov Chains

3.1 Introduction and Basic Properties

A Markov process is a dependent and stationary stochastic process $X(t)$ characterized by:

- A state space of $X(t)$, called $\mathcal{X}$.
- The time at which $X(t)$ changes, which can be at specific time instants (time-discrete) or at any arbitrary time (time-continuous).
- A transition matrix, which is called $\boldsymbol{P}$ if the Markov process is time-discrete, called **probability transition matrix**, or $\boldsymbol{Q}$ if the Markov process is time-continuous, called **rate transition matrix**. In both cases, the transition matrix represents the possible transitions from any state i to any state j.
- The equilibrium distribution (if exists), $\boldsymbol{\pi}$.

A Markov chain is a tool to represent a Markov process. It is composed by 'circles' and 'arrows', with circles representing $\mathcal{X}$, i.e., the possible outcomes of the random variable $X(t)$, and the arrows represent the transitions, i.e., the dependences between states.

A Markov Chain, as a representation of a Markov process satisfies the Markov property, which states that next state only depends of the current state.

$$Pr\{X_n = x_n | X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \ldots, X_0 = x_0\} = Pr\{X_n = x_n | X_{n-1} = x_{n-1}\} \tag{3.1}$$

Markov chains can be classified in two types depending on when $X(t)$ changes:

- A transition between two states happens only at specific instants of time. We will refer to these Markov Chains as Discrete Time Markov Chains (DTMC). For DTMC the transition matrix is called $\boldsymbol{P}$ and contains the probabilities to move from any state i to any state j.
- A transition between two states can be done at any arbitrary time. We will refer to these Markov Chains as Continuous Time Markov Chains (CTMC). For CTMC the transition matrix is called $\boldsymbol{Q}$ and contains the rates to move from any state i to any state j.

Some important properties of Markov chains are the next ones:

Irreducibility A Markov chain is called irreducible if the system can move from any state i to any state j, regardless the number of required transitions.

Aperiodic A Markov chain is called aperiodic if, after departing from state i, the system can return to it through following different paths.

Positive Recurrent A Markov chain is called positive recurrent if there is a non-zero probability that after departing from state i, the process will return to it in a finite time.

Ergodicity A Makov chain is Ergodic if all its states are **Aperiodic** and **Positive Recurrent**.

Besides that, in this text we only consider **time-homogeneous** Markov chains, which means that the transitions between states are time-independent. In this circumstances, given that the Markov chain is **irreducible**, its states are **positive recurrent** and it is **aperiodic**, we will be able to compute the stationary distribution of the Markov chain, i.e., the probability that the system is in each state at any arbitrary time, and it will be unique. In that case, the stationary distribution is also known as **equilibrium distribution**.

Finally, a Markov Chain is said to be **reversible** if the next condition is satisfied for all its states:

$$\pi_i \mathrm{P}\{X_n = j | X_{n-1} = i\} = \pi_j \mathrm{P}\{X_n = i | X_{n-1} = j\} \tag{3.2}$$

which results in the local balance equations, as we will see later. In plain words, a Markov chain is reversible if the Markov process moves from one state to another the same number of times as in the reverse direction. For example, if the Markov chain represents the probability that a person, which can be inside or outside a room, is inside or outside the room, in the long term, the person will move the same number of times from outside to inside the room as from inside to outside, as otherwise the person would not be able to enter/depart again.

As Markov chains are a useful tool to model telecommunication systems and networks, here we focus on describing how we can obtain their equilibrium distribution, from where several performance metrics of the system, such as packet losses and delay, can be derived.

3.2 Discrete Time Markov Chains: DTMC

We will use DTMC to model systems in which $X(t)$ only takes new values at specific time instants. Examples of cases that can be modelled by DTMC are

- Measuring the traffic on a link and checking its characteristics (i.e., number of packets) only at specifics intervals.
- In Server Farms, a load balancing algorithm that assigns a specific server to each incoming request. The system only changes at specific time instants, that are the instants when new request arrives.

Therefore, the main considerations for DTMCs are:

- In DTMC, transitions between states occur at specific time instants, t_1, t_2, t_3, t_4, t_5, t_6, etc. At each time instant, a transition from state i to state j happens with probability $p_{i,j}$.
- The transition probabilities from any state i to any state j define the matrix $\boldsymbol{P}$, called probability transition matrix.

One condition that we impose is that the matrix $\boldsymbol{P}$ has to be constant, i.e., the transition probabilities must be the same in all instants in which the system changes. If this is satisfied, the Markov chain is time-homogeneous.

3.2.1 Equilibrium Distribution

The Equilibrium distribution is obtained by solving the Balance Equations (Global or Local, although using the Local ones is generally simpler, although the local balance equations only exist when the Markov process is reversible). In both cases, the normalization condition must be considered.

$$\sum_{\forall i \in \mathcal{S}} \pi_i = 1 \tag{3.3}$$

Global Balance Equations

In equilibrium, for any state i, the next condition is satisfied:

$$\pi_i \sum_{\forall j \neq i} p_{i,j} = \sum_{\forall j \neq i} \pi_j p_{j,i} \tag{3.4}$$

which are the global balance equations. If the transition probabilities $p_{i,j}$ are known, the global balance equations provide as many equations as variables, and therefore, solving the system of equations, the equilibrium distribution of the Markov chain can be found.

Local Balance Equations

If the DTMC is reversible, the local balance equations can be formulated and used.

$$\pi_i p_{i,j} = \pi_j p_{j,i} \tag{3.5}$$

which together with the normalization condition, $\sum_{\forall i} \pi_i = 1$, allows to solve the system of equations and find the equilibrium distribution.

3.3 Continuous Time Markov Chains

When the system we have to model can change at any arbitrary time, and the time that the system remains in a certain state is important, we will use a CTMC to model it. Examples of stochastic processes that can be modelled with CTMCs are:

- The number of packets waiting in a queue, which depends on the time packets arrive and depart from it.
- The number of persons with active phone conversations in a cell.

Similary to DTMCs, CTMCs will be characterized by a set of states, $\mathcal{X}$, and a matrix containing the transition rates from one state to the other, $\boldsymbol{Q}$, known as infinitesimal generator or rate transition matrix.

To move from a DTMC to a CTMC, we assume that the time is divided in very small time intervals of size δ, in a way that changes seem as continuous (see Figure 3.1). For instance, when we watch the Television, it seems that the images are continuous, but this is just an illusion: the images are static and change every few msecs.

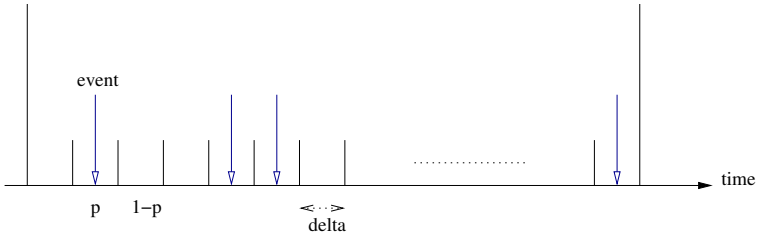


Figure 3.1: The Binomial to Poisson Distribution.

One of the mandatory requirements for these small intervals is that each one only can contain a single event. The probability that one period of time contains an event is $p = q\delta$, where q is the average rate (i.e. frequency) in which events happen (events / second). A second requirement is that all the periods of duration δ must have the same probability to contain or not an event, which means that the probability p must remain always constant. For example, if we have that $q = 10$ events / second, and define $\delta = 0.05$ seconds, the probability that a given period of duration δ contains an event is $p = q\delta = 0.5$.

To satisfy the Markov property, the time spent in each state of the Markov chain must have no memory (i.e., the past is not important and next change in the Markov chain will depend only on the present state). It means that one the Markov chain is in a certain state, the time that it will remain there must depend only on the current state. For instance, consider a router that is transmitting a packet. When the transmission of previous packet is near to be completed, a new packet arrives, which is placed in the queue as the transmitter is in use. At that point, the knowledge about the remaining time from the packet in transmission must be forgotten, and the system must act as if the remaining transmission time for the packet that is in transmission is again its average. This will be only true if the holding times are exponentially distributed, as we will see later.

3.3.1 The exponential distribution in CTMCs

Let us assume that we observe a system during T seconds, and divide T in N intervals of duration $\delta = T/N$.

If the probability that one interval contains a single event is p, the probability to have m events in N intervals is:

$$P\{m|N\} = \frac{N!}{m!(N-m)!} p^m (1-p)^{N-m} \tag{3.6}$$

Replacing p by $\frac{qT}{N}$, we obtain:

$$P\{m|N\} = \frac{N!}{m!(N-m)!} \left(\frac{qT}{N}\right)^m \left(1 - \frac{qT}{N}\right)^{N-m} \tag{3.7}$$

Developing the factorial, and rearranging some terms:

$$P\{m|N\} = \frac{N(N-1)\dots(N-m+1)(N-m)\dots 1}{m!(N-m)(N-m-1)\dots 1} \left(\frac{qT}{N}\right)^m \frac{\left(1-\frac{qT}{N}\right)^N}{\left(1-\frac{qT}{N}\right)^{-m}} \tag{3.8}$$

we obtain:

$$P\{m|N\} = \frac{N(N-1)\dots(N-m+1)}{N^m}\frac{(qT)^m}{m!}\frac{\left(1-\frac{qT}{N}\right)^N}{\left(1-\frac{qT}{N}\right)^{-m}} \tag{3.9}$$

Now, taking into account these three approximations that hold as N grows and $\frac{qT}{N}$ goes small:

$$\left(1-\frac{qT}{N}\right)^N \approx e^{-qT} \tag{3.10}$$

$$\frac{N(N-1)\dots(N-m+1)}{N^m} \approx 1 \tag{3.11}$$

$$\left(1-\frac{qT}{N}\right)^m \approx 1 \tag{3.12}$$

allows us to obtain:

$$P\{m|N\} \approx \frac{(qT)^m}{m!}e^{-qT} \tag{3.13}$$

which tends to be exact as N increases, or equivalently, δ becomes smaller.

This result is very important, and has huge implications. It says that the time between two events follows an exponential distribution. For example, what is the probability that in T seconds there are 0 events?

$$P\{0|T\} = e^{-qT} \tag{3.14}$$

and therefore, what is the probability that next event appears after T seconds?

$$1 - P\{0|T\} = 1 - e^{-qT} = F_\tau(t) \tag{3.15}$$

which is the Cumulative Probability Distribution of an Exponential Distribution for τ, the time between two events.

The conclusion of this observation is that in CTMCs, the time between two events are exponentially distributed.

3.3.2 Memoryless property of the Exponential Distribution

One of the characteristics of the exponential distribution is that it is **memoryless**. What does it mean? Well, basically that past information does not give us any useful information about what will happen in the future.

Example: The time between two consecutive packets that arrive to a router is exponentially distributed. Consider that at $t = 0$ the first packet arrives. What is the probability that we have to wait more than τ seconds to see the arrival of the second packet?

$$\mathrm{P}\{t > \tau\} = 1 - \mathrm{P}\{t \leq \tau\} = e^{-q\tau} \tag{3.16}$$

Now we have been waiting for τ_0. What is the probability that next packet arrives at $t > \tau_0 + \tau$ if we know that during the first τ_0 seconds it has not arrived?

$$\mathrm{P}\{t > \tau_0 + \tau | t > \tau_0\} = \frac{\mathrm{P}\{(t > \tau_0 + \tau) \cap (t > \tau_0)\}}{\mathrm{P}\{t > \tau_0\}} = \frac{\mathrm{P}\{t > \tau_0 + \tau\}}{P\{t > \tau_0\}} = \tag{3.17}$$

$$= \frac{1 - \mathrm{P}\{t \leq \tau_0 + \tau\}}{1 - \mathrm{P}\{t \leq \tau_0\}} = \frac{e^{-q(\tau_0+\tau)}}{e^{-q\tau_0}} = \frac{e^{-q\tau_0}e^{-q\tau}}{e^{-q\tau_0}} = e^{-q\tau} \tag{3.18}$$

$$\tag{3.19}$$

which says that our previous information has not give us any insight on what would happen in the future, as we could have been expected a lower probability in this second case.

Note that previous demonstration is based on the the conditional probability, that states:

$$\mathrm{P}\{A|B\} = \frac{\mathrm{P}\{A \cap B\}}{\mathrm{P}\{B\}} \tag{3.20}$$

and $\mathrm{P}\{(t > \tau_0 + \tau) \cap (t > \tau_0)\} = \mathrm{P}\{t > \tau_0 + \tau\}$. Intuitively the intersection between $t > \tau_0 + \tau$ and $t > \tau_0$ is $t > \tau_0 + \tau$.

3.3.3 Equilibrium Distribution

As observed, we have defined the probability that there is a change as $p = \lambda\delta$, where δ is an arbitrary small interval of time which can only contain one event.

Then, if we replace the transition probabilities in the global and local balance equations of a DTMC by previous definition, we obtain:

$$\pi_i \sum_{\forall j \neq i} q_{i,j}\delta = \sum_{\forall j \neq i} \pi_j q_{j,i}\delta \tag{3.21}$$

where $q_{i,j}$ is the transition rate between states i and j.

From previous equation is clear that δ does not depend on i nor j and therefore can be removed from both sides, resulting in

$$\pi_i \sum_{\forall j \neq i} q_{i,j} = \sum_{\forall j \neq i} \pi_j q_{j,i} \tag{3.22}$$

which are the global balance equations for a CTMC.

Equivalently, the local balance equations are.

$$\pi_i q_{i,j} = \pi_j q_{j,i} \tag{3.23}$$

.

3.4 Examples

3.4.1 Load Balancing in a Farm Server

In this example, we consider the File Storage and App Servers placed in the AN7. Each time that a new request arrives to them, it will be processed by only one of the three servers. In terms of processing power (i.e., CPU and memory), the first server has more resources than the second, and the second than the third one.

Then, when a new service request is received, if last request was served by server i, the server to process the incoming request will be selected following Table 3.1.

	Prob. Next Server		
Current Server	Server 1	Server 2	Server 3
1	5/7	2/7	-
2	2/7	2/7	3/7
3	-	4/7	3/7

Table 3.1: Probability Transition Matrix for the Load Balancing Algorithm

The DTMC that models the Load Balancing Algorithm is shown in Figure 3.2.

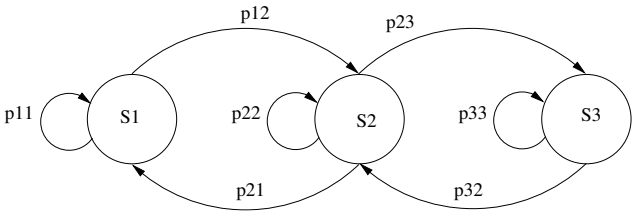


Figure 3.2: Load Balancing Algorithm for the Farm of Servers

Question: What is the stationary probability that a request is processed by Server 1, 2 and 4?

Solution: To solve this problem, as the Markov chain depicted in Figure 3.2 is reversible, we can use the local balance equations. We have two local balance equations:

$$p_{12}\pi_1 = p_{21}\pi_2 \tag{3.24}$$

$$p_{23}\pi_2 = p_{32}\pi_3 \tag{3.25}$$

$$\tag{3.26}$$

and using the normalization condition that states that $\pi_1 + \pi_2 + \pi_3 = 1$, we can solve the system of equations, obtaining:

$$\pi_1 = \frac{1}{1 + \frac{p_{12}}{p_{21}} + \frac{p_{23}}{p_{32}}\frac{p_{12}}{p_{21}}} = 0.36364, \quad \pi_2 = \frac{p_{12}}{p_{21}}\pi_1 = 0.36364, \quad \pi_3 = \frac{p_{23}}{p_{32}}\frac{p_{12}}{p_{21}}\pi_1 = 0.2727 \tag{3.27}$$

which gives us the equilibrium distribution of the stochastic process modeled by the Markov chain shown in Figure 3.2, and the stationary probability that at a new request will be assigned to Server 1, 2, and 3.

3.4.2 Performance Analysis of a Video Server

In this example, we model the **Video Server** that is placed in the AN6. Let's assume that the video server is only able to send a single video at each time. However, it can store up to two video requests. When there are two video requests waiting for service, all new arriving requests to the Video Server are discarded.

We model the number of requests in the Video Server using the stochastic process $X(t)$. The state space of $X(t)$ is § $= \{0, 1, 2, 3\}$, representing the number of requests that are in service and waiting in the Video Server.

Therefore, $X(t)$ can be described by a Markov chain with 4 states (Figure 3.3):

- State 0: There are 0 video requests in the video server.
- State 1: There is 1 video request in the video server, and it is being served.
- State 2: There are 2 video requests in the video server, the first one is being served and the second one is waiting.

- State 3: There are 3 video requests in the video server, one being served and two waiting.

The video requests arrive with a rate equal to $\lambda = 0.01$ video/second, and the duration of the requested videos is exponentially distributed with an expected duration of $1/\mu$ seconds. $\mu = 0.004$ is the departure rate in videos / second, and δ is a period of time arbitrarily small.

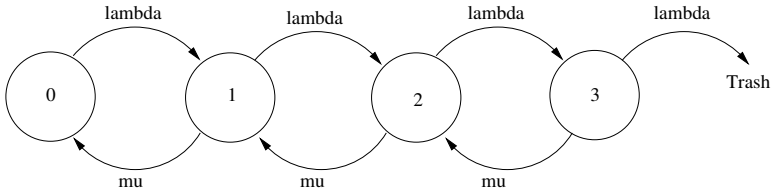


Figure 3.3: Continuous Markov Chain to Model the Video Server Operation

Question: What are the stationary probabilities of states 0, 1, 2 and 4?

Solution: To solve this problem, as the Markov chain depicted in Figure 3.3 is reversible, we can use the local balance equations.

$$\lambda\delta\pi_0 = \mu\delta\pi_1 \tag{3.28}$$

$$\lambda\delta\pi_1 = \mu\delta\pi_2 \tag{3.29}$$

$$\lambda\delta\pi_2 = \mu\delta\pi_3 \tag{3.30}$$

$$\tag{3.31}$$

and

$$\pi_1 = \frac{\lambda}{\mu}\pi_0 \tag{3.32}$$

$$\pi_2 = \frac{\lambda}{\mu}\pi_1 = \left(\frac{\lambda}{\mu}\right)^2 \pi_0 \tag{3.33}$$

$$\pi_3 = \frac{\lambda}{\mu}\pi_2 = \left(\frac{\lambda}{\mu}\right)^3 \pi_0 \tag{3.34}$$

$$\sum_{i=0}^{3} \pi_i = 1 \tag{3.35}$$

If we call $a = \lambda/\mu = 2.5$ and solve the system of equations we get:

$$\pi_0 = \frac{1}{1 + a + a^2 + a^3} = 0.039409 \tag{3.36}$$

$$\pi_1 = a\pi_0 = 0.098522 \tag{3.37}$$

$$\pi_2 = a^2\pi_0 = 0.24631 \tag{3.38}$$

$$\pi_3 = a^3\pi_0 = 0.61576 \tag{3.39}$$

$$\sum_{i=0}^{3} \pi_i = 1 \tag{3.40}$$

From the equilibrium distribution, we can compute some parameters of the system under study at any arbitrary time:

- The expected number of videos in the system: $E[N] = \sum_{i=0}^{3} i\pi_i =$
- The expected number of videos waiting: $E[N_\mathrm{q}] = \sum_{i=2}^{3} (i-1)\pi_i =$
- The probability that the video server is empty (not transmitting any video): $P_\mathrm{e} = \pi_0$
- The probability that the video server is not empty:

$$\sum_{i=1}^{3} \pi_i = 1 - \pi_0 = 1 - P_\mathrm{e}$$

- The probability that the video server has a single video: π_1
- The probability that the video server has a two videos: π_2
- The probability that the video server has a three videos: π_3

Part II

Modeling the Internet

Chapter 4

Delays in Communication Networks

4.1 A network of queues

As our goal is to analyze modern telecommunication networks, we first need to have at least a basic understanding of the relevant features of these networks in order to model them. To start, we observe that, with the exception of mobile telephone networks, most data networks use a datagram- or packet-switching paradigm (**TODO: reformulate to make it true**). This means that to transmit data from one endpoint of the network to another, this data is 'packetized', i.e., partitioned into smaller segments. Then, these packets are sent, each on its own, into the network, which forwards them towards their destination (quite similar to packets in the post).

This forwarding process consists in a number of intermediate steps or hops. Packets are transferred between networking devices such as end hosts, routers or access networks. Each intermediate device, i.e., a device that is not an end host, receives packets from incoming links, decides where to forward them, and queues them for transmission at the buffer of the according outgoing network interface card. The packet is transmitted over this outgoing link towards the next device along the path, and the step (also called 'store-and-forwarding') is repeated until the packet reaches its destination (or enters a different type of network).

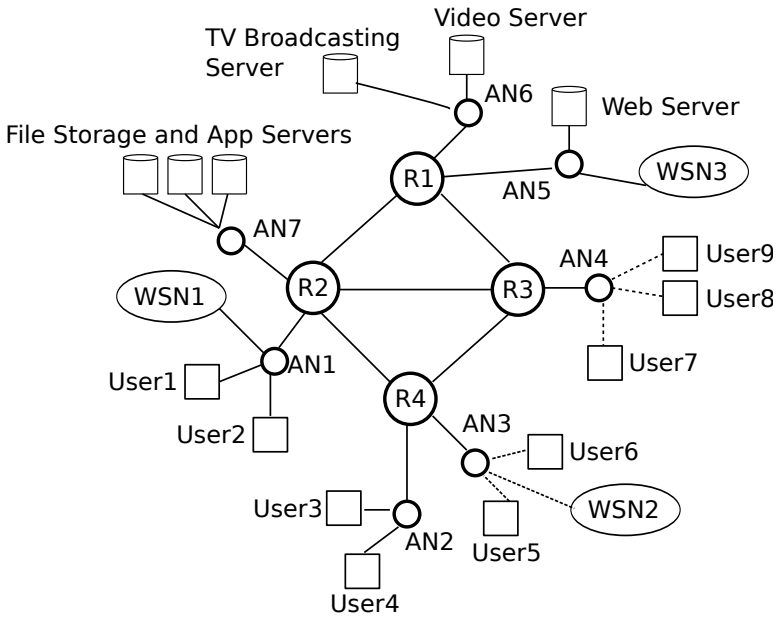


Figure 4.1: Basic Network

The breaking up of the transmission path into hops allows for a great flexibility in the network organization, and has contributed greatly to the success of packet-switched networks. Networks can be organized and handled more efficiently this way, and local changes can be made easily without having to change a lot in the global network.

However, the store-and-forward principle also means that packets have to be received fully in order to be forwarded. For a large amount of data, it is therefore better to transmit several small packets than a single large one, since a part of the total data can thus already be sent along while the rest is still being received. Still, each individual packet will experience a delay at each device, which we will explain in the following for each of the principal types of devices involved.

To use a common instance for such a multi-hop network that does not exceed the scope of this text, all the examples considered in the following will refer to the network presented in Figure 4.1. It has all three types of components: hosts, routers and access networks. Wireless Sensor Networks (WSNs) will be considered as an special type of host.

4.2 Types of Delay

Packets travel from the source host to the destination host. At each hop, a packet can suffer several delays:

1. **Transmission Delay ($D_\mathbf{s}$)**: The time required to transmit a packet, of size L bits, when a transmission rate of R bits / second is used. In general, $D_\mathrm{s} = \frac{L}{R}$. This delay is the time needed by the outgoing network interface to send all the bits of a packet over the outgoing link.

2. **Propagation Delay ($D_\mathbf{p}$)**: The time required for a packet to travel through the medium from the transmitter to the receiver. It depends on the distance between the transmitter and the receiver, d, and the medium characteristics (v). In general, $D_\mathrm{s} = \frac{d}{v}$. This delay stems exclusively from the spread of information along the medium used for transmission between elements. For instance, using optical cables, the information travels with the speed of light across the distance between the connected devices. In other media, such as copper, the speed is typically a significant fraction of the speed of light.

3. **Queueing Delay or Waiting Time ($D_\mathbf{q}$)**: The time that a packet spends in the queue waiting for transmission. This delay is of specific interest in the context of our considerations, because it is not simply derived from some basic parameters of the networking equipment, but depends strongly on the amount of traffic that needs to be handled, i.e., the number of packets that are transmitted over time.

4. **Processing Delay ($D_\mathbf{c}$)**: The time that a node needs to analyze a received packet. Aspects such as checking if the packet contains errors or not, and the packet's final destination are part of the Processing Delay. Among other things, like the used hardware, this depends on the size of the routing table in the device.

5. **Total Delay in a Node**: The overall time that a packet spends in a node, which is the sum of the processing, queueing and transmission delay. Note that the propagation delay is not included. Then,

$$D_\mathrm{node} = D_\mathrm{c} + D_\mathrm{q} + D_\mathrm{s}$$

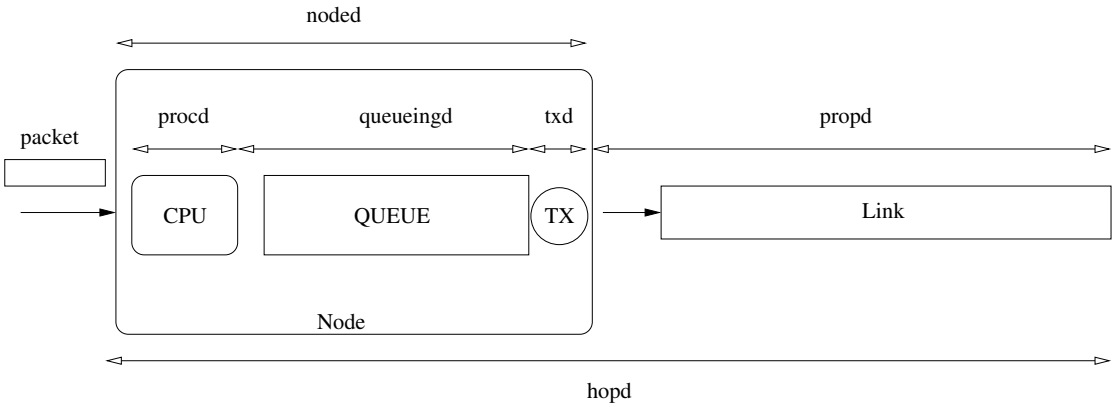


Figure 4.2: Packet delays at each hop

6. **Total Delay in a Hop**: The overall time that a packet spends in a hop, which is the sum of the total delay in a node and the propagation delay. We will refer to the total delay in a node by the variable D_{hop}, with

$$D_{\text{hop}} = D_{\text{node}} + D_{\text{p}}$$

7. **End-to-End Delay**: The total time since the packet is generated by the source until it is received at its destination. It depends on the number of hops between the source and the destination, as well as the characteristics and conditions of each hop (i.e. traffic load, distance, etc.). We will refer to the end-to-end delay by the variable D_{e2e}, with

$$D_{\text{e2e}} = \sum_{\forall i} D_{\text{hop},i}$$

Considering all of the different forms of delay described above, we can observe that the transmission delay, the propagation delay and to a large degree also the processing delay are static, i.e., they do not change much over time. The queueing delay is therefore typically the main source for variance in the end-to-end delay of packets of the same flow. While this variance does not affect a file download or an email transfer negatively, more recent and very popular applications are less lucky in that regard.

For instance, a video streaming application might be able to handle delay quite well

if it is the same for all packets: the stream just starts playing back after an amount of time equal to this delay (the situation is a bit different for interactive streaming, i.e., VoIP or video chat).

If the delay varies, however, it is not sure that the next packets that are needed for the playout process arrive in time. Therefore, video and audio streaming applications have to use buffers to handle the variance in the delay, which is in some works also called *jitter*.

Other applications might have even stronger demands on the delay. The aforementioned VoIP is typically said to be able to support an absolute one-way delay up to 150-400 milliseconds before the pauses in the conversation become too long to sustain it. In addition, one might, in the near future, imagine a remote control for important equipment over a network, e.g., for robots or for medical equipment.

In these cases, it might be a good idea to limit the delay of the packets belonging to these applications as much as possible. This can be done by giving them priority over other packets, which do not mind being treated in the best-effort way that is standard in the Internet. Thus, the so-called Future Internet might consist of a number of different classes of traffic, that are treated differently by the network (today, although many people are talking about it, nobody really knows what the Future Internet will look like).

Since the queueing delay has such a large influence on the function and quality of networking applications, it is very useful to be able to describe it, based on assumptions about the amount and type of traffic in the network. To this end, we will use queueing theory to analyze first and foremost the packet queue all of the aforementioned network devices share, i.e., between network layer and medium access layer. We will, step by step, extend this analysis to allow for more and more realistic cases, giving examples taken from actual applications on the way.

Chapter 5

Modeling a Network Element

In this chapter we will use Markov chains to model network elements (i.e., hosts, routers, and access networks). For the reader that is not familiar with this concept, Appendix 3 contains an introduction into this topic that is sufficient to understand the following steps.

We will characterize network elements in terms of the number of packets they contain over the time ($N(t)$), which depends on how many packets arrive (λ) and depart (μ) from them. This time-dependence is an important concept to realize, since it is different from, e.g., calculating the static transmission delay of a packet. A packet of a given size that is sent over a link with a given rate will always have the same transmission delay D_s. In contrast, the waiting time of a packet in the buffer does not only depend on the packet itself, but on the number of packets before it in the buffer, i.e., the buffer's state. A packet arriving one second later than another might find the buffer less filled and will therefore experience a shorter queueing delay.

A consequence of these considerations is that we will treat values like the waiting time of packets D_q or their number in the buffer as random variables, since they vary over time. Appendix 2 provides the necessary background in case the reader is not familiar with the basic concepts.

In general, we will not focus on the complete system but will model only single network interfaces, assuming that they are independent. Using these models, we will be able to compute several performance metrics related to the operation of

those network elements, such as the waiting and transmission delays for each packet in them, as well as the probability that a network element loses a packet due to buffer overflow.

Before we arrive at these values, however, we will first have to define the building blocks of our analytical model. Most importantly, these include the arrival process of packets at a network interface and how much data each packet contains, i.e., the characterization of the traffic that is being sent over that interface.

5.1 Modeling a Network Interface

A Network interface can be abstracted by assuming it contains:

- A single buffer and a single transmitter (usual).
- A single buffer and multiple transmitters (rare). However, it is a more general case containing the previous one.
- Multiple buffers and multiple transmitters, one for each buffer. An example of this case would be an Ethernet switch. However, in this case, we will simply analyze each queue independently of the others, returning to the first case.

Figure 5.1 shows a network interface characterized by a buffer of size Q and S transmitters. Its key parameters are:

- The number S transmitters or servers.
- Maximum number K of packets in the system (buffer + servers). A real-life buffer is limited in size, although it might be large. This is modeled by the maximum number of packets it can hold.
- Maximum number $Q = K - S$ of packets in the buffer.
- Transmission rate of each transmitter: R bps. We assume the same transmission rate for all transmitters, i.e., the same transmission technology is used in the device.
- Aggregate packet arrival rate: λ packets/second. This is the number of packets that arrives at the buffer per time unit (typically seconds), averaged over a

long time scale. Bursts in the packet arrival process, i.e., periods in which more than average packets arrive in a short time, are modeled by the variance of the packet interarrival time distribution, which is in the following typically assumed to be exponential.

- Packet Length: L bits. This is a random variable describing the distribution of the number of bits over the arriving packets.
- Service/Transmission time: $E[D_s] = \frac{E[L]}{R}$ seconds. Since we assume R to be constant, the variation of the transmission time therefore depends on the variation of the packet length distribution.
- Packet Departure rate: $\lambda(1 - P_b)$ packets/second.
- Maximum Packet Departure rate (assuming that all the transmitters are always busy): $S\mu = \frac{S}{E[D_s]}$ packets / second.
- Offered traffic or traffic intensity: $a = \frac{\lambda}{\mu}$ [Erlangs] (note that this value is independent of the number of servers).
- System utilization: $\rho = 1 - P_e$ (assuming that the system is working always when it is not empty).

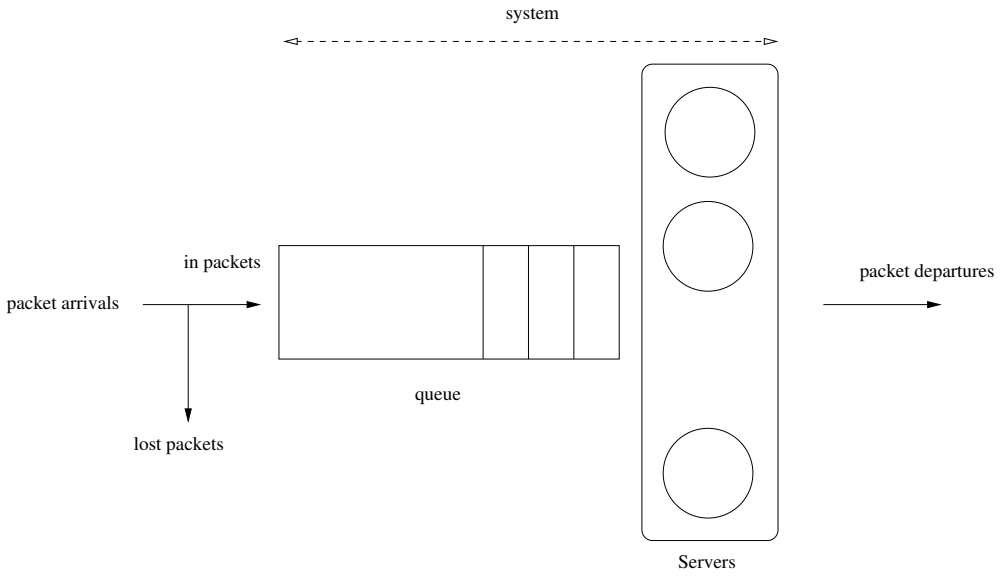


Figure 5.1: Model of a Network Interface

5.2 Erlang Notation

Due to the different configurations that queueing systems can have, there is a standardized procedure to refer to them, and it is called the "Erlang notation". It basically uses a combination of letters as follows:

$$A/B/S/K : SP$$

where

- **A** refers to the packet arrival process. If the packet arrival process follows a Poisson distribution, we will use the letter M.
- **B** refers to the packet service process. If the packet service time is exponentially distributed, we will use the letter M.
- **S** refers to the number of servers (i.e. transmitters or processors).
- **K** refers to the total number of packets that can be stored in the queueing system, including those in the buffer and those in service. If it is omitted, we assume that the buffer has an infinite capacity.
- **SP** refers to the scheduling policy. If it is not given explicitly, we assume that a FIFO (First-In First-Out) policy is applied.

Some examples are:

- **M/M/1/K**: Poisson arrival process, exponentially distributed service times, 1 server and a total system capacity of K packets, with $Q = K - 1$. A FIFO policy is considered.
- **M/M/3/K**: Poisson arrival process, exponentially distributed service times, 3 servers and a total system capacity of K packets, with $Q = K - 3$. A FIFO policy is considered.
- **M/M/1**: Poisson arrival process, exponentially distributed service times, 1 server and a total system capacity of ∞ packets. A FIFO policy is considered.

- **M/D/2**: Poisson arrival process, deterministically distributed service times, 2 servers and a total system capacity of ∞ packets. A FIFO policy is considered.
- **D/D/2**: Deterministic arrival process, deterministically distributed service times, 2 servers and a total system capacity of ∞ packets. A FIFO policy is considered.
- **M/G/1**: Poisson arrival process, generally distributed service times, 1 server and a total system capacity of ∞ packets. A FIFO policy is considered. The general distribution is usually characterized by both the expected value and the coefficient of variation.

5.3 Stability

A queueing system is stable if it is able to process all the packets that enter the system, over a long timescale. In other words, a queueing system is stable if any packet that has been placed inside the system will depart in a finite period of time. Note that blocked packets are not considered.

By default, all systems with a finite buffer are stable, since any packet that enters the system will depart from it. However, queues with infinite buffers are not stable if the offered traffic is higher than the maximum departure rate ($S\mu$), as the buffer backlog will increase until infinite and therefore, there will be packets that never will depart the system in a finite time.

In general, a queue is stable if:

$$\lambda(1 - P_{\mathrm{b}}) < S\mu \quad \rightarrow \quad \frac{\lambda(1 - P_{\mathrm{b}})}{S\mu} < 1 \quad \rightarrow \quad \frac{a(1 - P_{\mathrm{b}})}{S} < 1 \tag{5.1}$$

For $K = \infty$, as $P_{\mathrm{b}} = 0$, we have to guarantee that $a < S$. In other words, the average arrival rate of packets to the system needs to be lower than the maximum departure rate of the system.

5.4 Stationarity

We assume the traffic arrivals are stationary, i.e., the distribution of the values that model the number of packets that arrive to the system do not change along the time.

5.5 Poisson Arrivals

All the system models covered in this book assume a Poisson arrival process. This arrival process has important characteristics which have an impact on the analysis of these systems. In the following, we cover these characteristics.

5.5.1 PASTA

PASTA is the acronym for **P**oisson **A**rrivals **S**ee **T**ime **A**verages, which is one of the key properties of a Poisson arrival process.

First, what is a Time Average? It refers to the state distribution in the equilibrium. For a total observation time that is sufficiently long, the probability that the system is at state i in equilibrium can be formulated as:

$$\pi_i = \frac{\text{Time the system is observed at state } i}{\text{Total Observation Time}} \tag{5.2}$$

The PASTA property says that when a new packet arrives, the probability that it finds i packets in the system, i.e., $P_a\{N(t) = i\}$, is equal to π_i:

$$P_a\{N(t) = i\} = \pi_i.$$

The Poisson distribution appeared when we moved from DTMCs to CTMCs. We defined small intervals of duration λ, where each interval could contain only one arrival with probability p, that had to be constant and the same for any interval. In those conditions, the probability that one arrival observes i packets in the system

$$P_{\mathrm{a}}\{N(t) = i\} = \frac{\text{Number of arrivals that see the system in state } i}{\text{Total Number of Arrivals}} \tag{5.3}$$

is proportional to the time the system is in that state, as intuitively, if the system remains in a certain state for more time than in others, the probability that a packet arrives when the system is in that state is also higher. Why? Because, the probability that a packet arrives in a δ period is constant over the time.

Clearly, if the assumption that p is not constant and the same for all δ intervals is not true, PASTA does not hold.

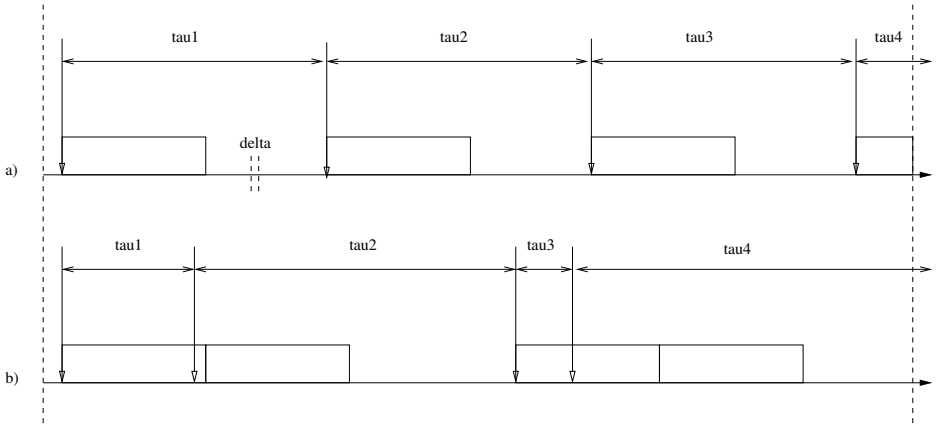


Figure 5.2: Example of the PASTA property. From the Figure, we can see that $\pi_0 = 0.5$ and $\pi_1 = 0.5$. In case a), the interarrival time is deterministic, and all packet arrivals find the system in the empty state. In case b), the interarrival time is exponentially distributed, and 2 packet arrivals find the system in state 0, and two in state 1. As we have 4 arrivals, the probability that an arrival observes the system in state i is the same as the equilibrium probability that the system is in state i (i.e. π_i)

A example-based proof

Consider the temporal evolution of a system with a single server and without any buffer space, i.e. $K = 1$ (Figure 5.2). λ packets/second arrive to this system, and the packets have a deterministic length, which results in a deterministic service time D_s. We observe the system for T_{obs} seconds, dividing the time in several intervals of duration δ. We assume that in each δ there can be only one arrival.

In this system (Figure 5.2), the equilibrium distribution over the time can be computed by dividing the time that the system has been in each state by T_{obs}, i.e.,

$$\pi_0 = \frac{\text{Time the system has been in state 0}}{T_{\text{obs}}} \approx 0.5 \tag{5.4}$$

$$\pi_1 = \frac{\text{Time the system has been in state 1}}{T_{\text{obs}}} \approx 0.5 \tag{5.5}$$

$$\tag{5.6}$$

- In the first case the time between two packet arrivals follows a deterministic

distribution, and its value is τ. Let us assume that $\tau > D_s$. What is the probability that an arrival sees the system empty? The answer is easy, it is 1. In all cases, when a packet arrives the system is empty. Therefore, the PASTA property does not hold in this case, as it states that the probability that a new arriving packet sees i packets in the system is equal to the probability that there are i packets in the system at any arbitrary time (i.e., the equilibrium probability). In detail, $P_a(0) = 1 \neq \pi_0$.

- In the second case, packets arrive following a Poisson process with rate $\lambda = 1/\tau$. As we can see, the value of λ will be the same as in previous case. However, now τ is a random variable exponentially distributed. In this conditions, the probability that an event contains one arrival is the same for all the intervals. In this case, the packet arrivals will sample all possible cases with equal probability, and will satisfy the PASTA property.

A formal proof

A more formal definition is as follows, taking into account the assumption that the probability of an arrival in an interval δ is equal in all of them and therefore constant.

$$
\begin{aligned}
P_{\mathrm{a}}(i) &= \lim_{t\to\infty}\lim_{\delta\to 0} P\{N(t) = i | \text{arrival at } (t, t+\delta)\} && (5.7)\\
&= \lim_{t\to\infty}\lim_{\delta\to 0} \frac{P\{N(t) = i \cap \text{arrival at } (t, t+\delta)\}\}}{P\{\text{arrival at } (t, t+\delta)\}\}} && (5.8)\\
&= \lim_{t\to\infty}\lim_{\delta\to 0} \frac{P\{N(t) = i\}P\{\text{arrival at } (t, t+\delta) | N(t) = i\}\}}{P\{\text{arrival at } (t, t+\delta)\}\}} && (5.9)\\
&= \lim_{t\to\infty}\lim_{\delta\to 0} \frac{P\{N(t) = i\}P\{\text{arrival at } (t, t+\delta)\}}{P\{\text{arrival at } (t, t+\delta)\}} && (5.10)\\
&= \lim_{t\to\infty}\lim_{\delta\to 0} P\{N(t) = k\} = \pi_i && (5.11)\\
& && (5.12)
\end{aligned}
$$

Observe here that we have used the definition of the conditional probability $P\{A \cap B\} = P\{B\}P\{A|B\}$ twice, and the fact that $P\{N(t) = i \text{ and arrival at } (t, t+\delta)\}$

are independent events.

PASTA also implies that in equilibrium, an arrival observes $\mathrm{E}[N] = \mathrm{E}[N_q] + \mathrm{E}[N_s]$ packets in the system.

5.5.2 Aggregation and Division of Poisson processes

Two very interesting properties of Poisson processes are:

- **Aggregation of Poisson processes**: the aggregation of several Poisson processes results in a new Poisson process with a rate $\lambda_{\text{aggregate}}$ which is the sum of the rates λ_i of the individual Poisson processes aggregated.

$$\lambda_{\text{aggregate}} = \sum_i \lambda_i \tag{5.13}$$

- **Splitting a Poisson process** in several other processes selecting packets randomly with constant probability over the time causes the resulting processes to be also Poisson. For example, if we split a Poisson process in two Poisson processes, we obtain:

$$\lambda_1 = \alpha_1 \lambda_{\text{aggregate}} \tag{5.14}$$

$$\lambda_2 = \alpha_2 \lambda_{\text{aggregate}} \tag{5.15}$$

$$1 = \alpha_1 + \alpha_2 \tag{5.16}$$

$$\tag{5.17}$$

  with α_1 being the probability that a packet belongs to the resulting Poisson process 1, and α_2 the opposite. Note that to obtain several Poisson processes from a single Poisson process, the assignation of a packet to the resulting process must be independent of previous decisions (i.e., stochastic).

5.6 Exponential Packet length and Residual Service Time

One of the requirements of CTMC is that the time in a state has to follow an exponential duration. This can be deduced from the previous chapter, but here we will give a more behavioral explanation.

Notice that when a Markov chain changes from one state to another, the Markov chain only knows that it is in the new state, as it does not store any information from the past. For example, we can move to state i from state $i-1$ or from state $i+1$, though the Markov chain does not care about which of these alternatives actually happened. In other words, at every change of state, the Markov chains forgets the past, which is the same as to say it is memoryless. Therefore, the time in each state must be exponentially distributed:

- If the queue is empty, it is guaranteed by the fact that Poisson arrivals have exponentially distributed interarrival times.
- If the queue is not empty, the time in each state must be also exponentially distributed, which means that the service times must be exponentially distributed, with expected value $E[D_s] = \frac{1}{\mu}$ and μ the departure rate.

Therefore, in those conditions, the probability density function of the service time is:

$$f_{D_s}(t) = \mu e^{-\mu t} \tag{5.18}$$

and its cumulative probability distribution

$$F_{D_s}(t) = 1 - e^{-\mu t} \tag{5.19}$$

5.6.1 Residual Service Times in Markov Chains

Consider a system in which the packet sizes are exponentially distributed, with expected packet size $E[L]$. The system transmits packets at a constant transmission rate R. In this case, the service time also follows an exponential distribution, as it only depends on the distribution of L, and the expected service time is $E[D_s]$. Note that the parameter of the exponential distribution is $\mu = \frac{1}{E[D_s]}$.

Now, consider that a packet starts to be transmitted at $t = 0$. If another packet arrives at the system also at time 0, what is the probability that the residual time it observes is larger than T? Since we know that the service time is exponentially distributed, to compute this probability is straightforward:

$$P\{t > T\} = 1 - P\{t \leq T\} = e^{-\mu T} \tag{5.20}$$

Now, we assume that the second packet arrives T_0 units of time after the packet in service has started. In this case, what is the probability that the residual service time observed by this packet is also T? Notice that it means that the service time of the packet finishes at the time instant $T_0 + T$.

$$\begin{aligned} P\{t > T + T_0 | t > T_0\} &= \frac{P\{t > T + T_0 \cap t > T_0\}}{P\{t > T_0\}} \\ &= \frac{P\{t > T + T_0\}}{P\{t > T_0\}} = \frac{1 - P\{t \leq T + T_0\}}{1 - P\{t \leq T_0\}} \\ &= \frac{e^{-\mu(T+T_0)}}{e^{-\mu T_0}} = e^{-\mu T} \end{aligned} \tag{5.21}$$

As we can see, we obtain the same result. What does this mean? It means that the information that the packet has started T_0 seconds before our arrival does not have any impact on the future (i.e., it is useless). This is known as the memoryless property of the exponential distribution.

Thus, the residual time when the service time is exponentially distributed satisfies:

$$E[D_r] = E[D_s] \tag{5.22}$$

and it is also exponentially distributed with parameter μ.

5.7 Little's Law

Little's law, applied to our use-case says that the long-term average number of packets in a system is the average arrival rate of packets to that system multiplied by the average time it spends in the system. This can be applied to just the number of waiting packets (if just the buffer is interpreted as the system), just the number of packets in the servers (with just the transmitters being the system), or the complete system. Formally, this means

$$\mathrm{E}[D] = \frac{\mathrm{E}[N]}{\lambda(1-P_\mathrm{b})}, \qquad \mathrm{E}[D_q] = \frac{\mathrm{E}[N_q]}{\lambda(1-P_\mathrm{b})}, \qquad \mathrm{E}[D_s] = \frac{\mathrm{E}[N_s]}{\lambda(1-P_\mathrm{b})} \tag{5.23}$$

Why? Let us assume that we are the last packet that has arrived at the queue. In equilibrium, we find on average $E[N]$ packets before us, and we will depart from the queue after $\mathrm{E}[D]$ seconds. How many packets will we see on average in the queue at the moment we depart? In equilibrium, we will find on average $E[N]$ packets. Therefore, it means that during the time we have been in the queue, there have been $E[N]$ arrivals. Then,

$$\lambda(1-P_\mathrm{b})\mathrm{E}[D] = \mathrm{E}[N] \tag{5.24}$$

with $\lambda(1-P_\mathrm{b})\mathrm{E}[D]$ the number of packets that arrive and enter the system on average, during the time we are in it.

5.8 Performance Metrics

To evaluate the performance of a queueing system, we are interested in several performance metrics (parameters which characterize the system performance). These metrics describe, in different forms, how well the system is dimensioned, and how good the 'service' is that the packets experience. In terms of a packet network, the two most important so-called *Quality of Service* metrics are the packet loss probability and the delay experienced by the packets. Some helpful results from the analysis are therefore:

- Probability that the system is empty at any arbitrary time: $P_e = \pi_0$.
- Probability that there are i packets in the system at any arbitrary time: π_i.
- Blocking or packet loss probability: $P_b = \pi_K$.
- System utilization: the fraction of time that the system is active (i.e., transmitting packets), given by: $\rho = 1 - \pi_0$.
- Expected number of packets in the queue: $E[Q] = \sum_{i=S+1}^{K} (i - S) \cdot \pi_i = E[N] - E[N_s]$.
- Expected number of packets in service: $E[N_s] = \sum_{i=0}^{K} \min(i, S) \cdot \pi_i$.
- Expected number of packets in the system: $E[N] = \sum_{i=0}^{K} i \cdot \pi_i$.
- Expected delay of a packet in the buffer:

$$E[D_q] = \frac{E[N_q]}{\lambda(1 - P_b)}.$$

- Expected delay of a packet in the server:

$$E[D_s] = \frac{E[N_s]}{\lambda(1 - P_b)}.$$

- Expected delay of a packet in the system:

$$E[D] = \frac{E[N]}{\lambda(1 - P_b)}.$$

- Probability to find the system working (transmitting packets): $1-\pi_0 = \rho$ (this has to be understood as follows: *always when the system contains packets, there is at least one packet in service*).

5.9 Basic Queueing Systems

In this section we present the three basic queueing systems that are based on the assumption of a Poisson arrival process and an exponentially distributed service time.

In all the systems considered, we assume that the packet arrival rate λ is time-homogeneous and independent of the system state. In the same way, for the cases where multiple servers are considered, we assume that all the servers are equal, with a service rate equal to μ. In addition, $a = \frac{\lambda}{\mu}$ is the offered traffic or traffic intensity.

5.9.1 The M/M/S/K queue

The M/M/S/K system has been widely used in the past to plan telephone networks, where the number of servers S models the number of calls that can be active simultaneously in a link or a cell (i.e., lines or channels).

In today's packet based networks, in general, all the links have a single transmitter, which is modeled using an M/M/1/K system, which is an specific case of the M/M/S/K system with a single server ($S = 1$). However, there are also examples where $S > 1$. For instance, a M/M/S/K queueing system can be applied to model a server with multiple processors but a single shared queue for all the arriving requests. We therefore cover the analysis of such a system for completeness' sake.

The M/M/S/K queueing system implicitly assumes Poisson arrivals with rate λ and exponentially distributed service times with average $E[D_s] = 1/\mu$.

Markov chain

In Figure 5.3 the Markov chain for the M/M/S/K queueing system is depicted.

Note that the service rate depends on the system state (more busy servers mean a higher service rate), while the arrival rate does not. Once all S servers are busy, this rate cannot increase even if more (waiting) packets are in the system. The number

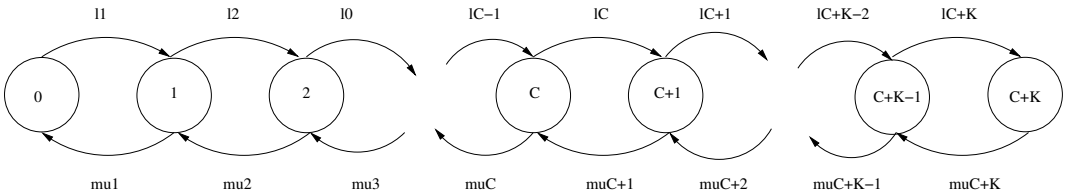


Figure 5.3: Markov Chain for the M/M/S/K queue

of states is the number of packets possible to have in the system. Since the system can also be empty, we get $K + 1$ states.

Local Balance Equations

The balance equations for the M/M/S/K system are:

$$\begin{aligned}
\pi_0\lambda &= \pi_1\mu \\
\pi_1\lambda &= \pi_2 2\mu \\
&\dots \\
\pi_{i-1}\lambda &= \pi_i i\mu, \qquad i \leq S \qquad (5.25)\\
&\dots \\
\pi_{i-1}\lambda &= \pi_i S\mu, \qquad i \geq S \qquad (5.26)\\
&\dots \\
\pi_{K-1}\lambda &= \pi_K S\mu
\end{aligned}$$

This, together with the normalization condition:

$$\sum_{i=0}^{K} \pi_i = 1, \qquad (5.27)$$

will allow us to obtain the equilibrium distribution for the M/M/S/K system.

Note that we can write the previous equations as follows:

$$\pi_i = \frac{\lambda}{i\mu}\pi_{i-1} = \frac{a^i}{i!}\pi_0, \qquad i \leq S \tag{5.28}$$

$$\ldots$$

$$\pi_i = \frac{\lambda}{S\mu}\pi_{i-1} = \frac{a^{i-S}}{S^{i-S}}\pi_S = \frac{a^{i-S}}{S^{i-S}}\frac{a^S}{S!}\pi_0 = \frac{a^i}{S^{i-S}S!}\pi_0, \qquad i > S \tag{5.29}$$

Equilibrium Distribution

The equilibrium probability for the 0-th state is given by

$$\pi_0 = \frac{1}{\sum_{j=0}^{S}\frac{a^j}{j!} + \sum_{j=S+1}^{K}\frac{1}{S^{j-S}}\frac{a^j}{S!}}. \tag{5.30}$$

With π_0 known, all the other π_i can be computed as well. Thus, the state probabilities are known, allowing to derive all other performance metrics. This will be done in more detail for the M/M/1/K system.

Performance Metrics

Refer to Section 5.8.

5.9.2 M/M/1/K queue

As explained above, the M/M/1/K system is a specific case of a M/M/S/K system with a single server ($S = 1$). However, as it is the most common queueing system in today's Internet we will consider it in detail, deriving closed-form expressions when possible, specifically for the performance metrics.

Markov chain

Again, the basis for our analysis is using a Markov chain with the number of packets in the system as the states. In Figure 5.4 this Markov chain for the M/M/1/K queue is depicted.

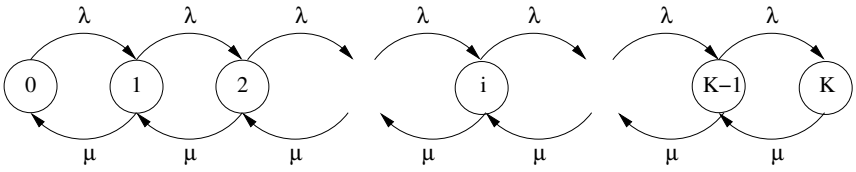


Figure 5.4: Markov Chain for the M/M/1/K queue

In contrast to the Markov chain for the M/M/S/K system, the service rate remains constant, since there is only one server. This also simplifies the load balance equations a bit.

Local Balance Equations

$$\begin{aligned} \pi_0\lambda &= \pi_1\mu \\ \pi_1\lambda &= \pi_2\mu \\ &\dots \\ \pi_{i-1}\lambda = \pi_i\mu \rightarrow \pi_i = \left(\frac{\lambda}{\mu}\right)\pi_{i-1} &= a\pi_{i-1} \rightarrow \pi_i = a^i\pi_0 \\ &\dots \\ \pi_{K-1}\lambda &= \pi_K\mu \end{aligned} \tag{5.31}$$

Equilibrium Distribution

Using again the normalization condition (the sum of all state probabilities equals one), the stationary probability for the 0-th state is

$$\pi_0 = \frac{1}{\sum_{j=0}^{K} a^j} = \frac{1}{\frac{1-a^{K+1}}{1-a}} = \frac{1-a}{1-a^{K+1}}. \tag{5.32}$$

Knowing the state probability for state 0, the stationary probability for the i-th state then is

$$\pi_i = a^i\pi_0 = \frac{(1-a)a^i}{1-a^{K+1}}. \tag{5.33}$$

Now we will derive the other performance metrics from these state probabilities.

Performance Metrics

Packet Losses

The probability that a packet is lost due to buffer overflow is equal to the probability that an arriving packet sees a full system, i.e., the system in state K. Thus, it can be computed by applying PASTA:

$$P_b = \pi_K = \frac{(1-a)a^K}{1-a^{K+1}} \tag{5.34}$$

Queue Occupancy

The state probabilities essentially give us the distribution for the number of packets in the system. Therefore, to calculate the average number of packets, are looking for the average of that distribution.

If $a \neq 1$, the system occupation is computed as follows:

$$
\begin{aligned}
E[N] &= \sum_{q=0}^{K} \pi_q q = \sum_{q=0}^{K} \frac{(1-a)a^q}{1-a^{K+1}} q \\
&= \frac{(1-a)}{1-a^{K+1}} \sum_{q=0}^{K} q a^q = \frac{a(1-a)}{1-a^{K+1}} \sum_{q=0}^{K} q a^{q-1} \\
&= \frac{a(1-a)}{1-a^{K+1}} \sum_{q=0}^{K} \frac{d}{da} a^q = \frac{a(1-a)}{1-a^{K+1}} \frac{d}{da} \sum_{q=0}^{K} a^q \\
&= \frac{a(1-a)}{1-a^{K+1}} \frac{d}{da} \left(\frac{1-a^{K+1}}{1-a} \right) = \\
&= \frac{a(1-(K+1)a^K + Ka^{K+1})}{(1-a)(1-a^{K+1})} = \\
&= \frac{a(1-(K+1)a^K + (K+1)a^{K+1} - a^{K+1})}{(1-a)(1-a^{K+1})} = \\
&= \frac{a(1-a^{K+1} - (K+1)a^K(1-a))}{(1-a)(1-a^{K+1})} = \\
&= \frac{a}{1-a} - \frac{(K+1)a^{K+1}}{1-a^{K+1}}
\end{aligned} \tag{5.35}
$$

In 5.35 we can observe that the system occupancy has two terms. The first term describes the queue occupancy for an infinite queue, and the second term is the queue occupancy that is lost due to the finite buffer size. We observe that, if K increases, the second term tends to 0 if $a < 1$.

For the specific case $a = 1$:

$$E[N] = \frac{K}{2}$$

To find $E[N_q]$, we can use the relation

$$E[N_q] = E[N] - E[N_s] \tag{5.36}$$

where

$$E[N_s] = 1 - \pi_0 \tag{5.37}$$

Alternatively, $E[N_q]$ can be computed in a similar fashion to $E[N]$, as $E[N_q] = \sum_{q=0}^{K} \pi_q(q-1)$.

Delay

The parameters related to the delay can be obtained by simply applying Little's formula:

$$E[D_s] = \frac{E[N_s]}{\lambda(1-P_b)} = \frac{1-\pi_0}{\lambda(1-P_b)} = \frac{E[L]}{R} \tag{5.38}$$

$$E[D] = \frac{E[N]}{\lambda(1-P_b)} \tag{5.39}$$

$$E[D_q] = \frac{E[N_q]}{\lambda(1-P_b)} = E[D] - E[D_s] \tag{5.40}$$

Note that this equality is true $\frac{1-\pi_0}{\lambda(1-P_b)} = \frac{E[L]}{R}$ since $\lambda(1-P_b) = \mu(1-\pi_0)$ (from the stability requirements). Therefore,

$$\frac{\lambda}{\mu} = \frac{(1-\pi_0)}{(1-P_b)}$$

Transforming this relation slightly, we get that

$$\frac{1-\pi_0}{\lambda(1-P_b)} = \frac{1}{\mu} \tag{5.41}$$

$$1-\pi_0 = \frac{\lambda(1-P_b)}{\mu} \tag{5.42}$$

$$\pi_0 = 1 - \frac{\lambda(1-P_b)}{\mu} = 1 - a(1-P_b) \tag{5.43}$$

$$E[N_s] = a(1-P_b) \tag{5.44}$$

which expresses that the expected number of packets in the server is equal to the fraction of traffic intensity that enters in the queue.

5.9.3 M/M/1 queue

The M/M/1 queueing system assumes that the system buffer size is infinite (i.e. $K \to \infty$), which is obviously not realistic. However, when the queue size is large, and the traffic intensity is low, it provides a very accurate model and allows to estimate the different expected delays using simpler expressions than the used for the M/M/1/K queue. It should therefore be seen as a complementary tool that can provide useful results faster, if the aforementioned prerequisites are met.

The analysis of a M/M/1 queue can be only done if the system is stable. Since the queue is infinite here, we require that $a < 1$. Note that for the M/M/1 queue, $P_b = 0$, since new packets can always be stored in the buffer.

Markov chain

In Figure 5.5 the Markov chain for the M/M/1 queue is shown. Observe that the Markov chain now has an infinite number of states.

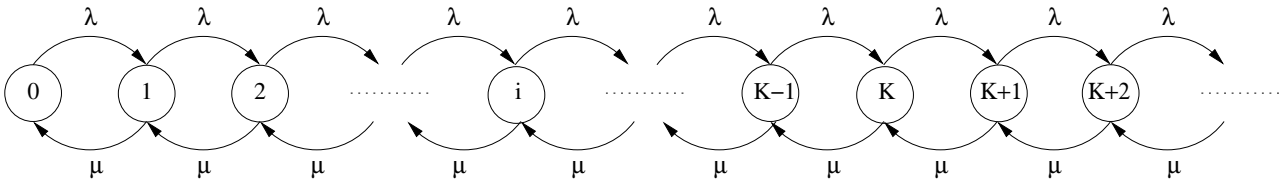


Figure 5.5: Markov Chain for the M/M/1 queue

Local Balance Equations

The balance equations for the M/M/1 system take the same form as in the M/M/1/K system, although now we have an infinite number of them.

$$\begin{aligned} \pi_0\lambda &= \pi_1\mu \\ \pi_1\lambda &= \pi_2\mu \\ &\dots \\ \pi_{i-1}\lambda = \pi_i\mu \rightarrow \pi_i = \left(\frac{\lambda}{\mu}\right)\pi_{i-1} &= a\pi_{i-1} \rightarrow \pi_i = a^i\pi_0 \\ &\dots \end{aligned} \tag{5.45}$$

Equilibrium Distribution

The equilibrium distribution for the M/M/1 system is the same as for the M/M/1/K system, just considering that $K \rightarrow \infty$, which leads to $a^{K+1} = 0$ if $a < 1$. Then, the stationary probability for the 0-th state is

$$\pi_0 = \frac{1}{\sum_{j=0}^{\infty} a^j} = \frac{1}{\frac{1}{1-a}} = 1 - a \tag{5.46}$$

The stationary probability for the i-th state is therefore

$$\pi_i = a^i\pi_0 = (1-a)a^i \tag{5.47}$$

Distribution of the time a packet spends inside the system, D

For the M/M/1 the distribution of the time that a packet spends inside the system is given by:

$$f_D(t) = (\mu - \lambda)e^{-(\mu-\lambda)t} \tag{5.48}$$

for $t > 0$. We can observe that the time that a packet spends inside the system also follows an exponential distribution with rate $\mu - \lambda$. For instance, what is the probability that a packet is inside the system less or T seconds?

$$Pr\{t \leq T\} = F_D(t) = 1 - e^{-(\mu-\lambda)T} \tag{5.49}$$

Performance Metrics

As we will see in the following, we can obtain very simple formulas for the performance metrics of the M/M/1 queue.

Packet Losses

As stated before, the probability that a packet is lost due to buffer overflow is

$$P_b = 0 \tag{5.50}$$

Queue Occupancy

If $a < 1$, the system occupation is computed as follows:

$$E[N] = \sum_{q=0}^{\infty} \pi_q q = \frac{a}{1-a} \text{ packets.} \tag{5.51}$$

To find $E[N_q]$, we will use the relations

$$E[N_s] = 1 - \pi_0 = a \tag{5.52}$$

and

$$E[N_q] = E[N] - E[N_s] \tag{5.53}$$

$$= \frac{a}{1-a} - a = \frac{a - a + a^2}{1-a} = \frac{a^2}{1-a} \text{ packets.} \tag{5.54}$$

Delay

The parameters related to the delay are obtained by applying Little's formula:

$$E[D_s] = \frac{E[N_s]}{\lambda(1-P_b)} = \frac{a}{\lambda} = \frac{1}{\mu} \text{ seconds} \tag{5.55}$$

$$E[D] = \frac{E[N]}{\lambda(1-P_b)} = \frac{1}{\mu(1-a)} = \frac{1}{\mu-\lambda} \text{ seconds} \tag{5.56}$$

$$E[D_q] = \frac{E[N_q]}{\lambda(1-P_b)} = \frac{a^2}{\lambda(1-a)} = \frac{a}{\mu(1-a)} = \frac{a}{\mu-\lambda} \text{ seconds} \tag{5.57}$$

$$\tag{5.58}$$

We can reach the same expression for $E[D_q]$ by considering that

$$E[D_q] = E[D] - E[D_s] = \frac{1}{\mu-\lambda} - \frac{1}{\mu} = \frac{\lambda}{\mu(\mu-\lambda)} = \frac{a}{\mu-\lambda} \text{ seconds} \tag{5.59}$$

5.10 Examples

5.10.1 Example - Multiple Links sharing a buffer

Motivation: This example shows that sharing a single buffer between multiple transmitters is a better approach than individual buffers for each transmitter, given that the sum of buffer space is exactly the same.

Let us consider the link between R1 and R3 consists of three independent cables of capacity C each one. Let us consider two cases:

1. Each virtual link (i.e., each transmitter) has an individual buffer of size Q.
2. All virtual links share a single buffer of size 3Q.

To compare which is the best situation, we compute the expected system delay $E[D]$ for both cases. In the first case, we have three independent M/M/1/Q+1 system, and in the second case, we have a single M/M/3/3Q+3 system. To obtain the results,

we will consider that Q=4 packets, that packet arrival rate is $\lambda = 20$ packets/second, and each transmitter can transmit packets at a rate $\mu = 25$ packets/second.

Solution: In the first case, we have a M/M/1/5 system. In such a system, the delay is given by:

$$E[D] = \frac{E[N]}{\lambda(1 - P_b)} \tag{5.60}$$

where

$$P_b = \frac{0.8^5(1 - 0.8)}{1 - 0.8^6} \tag{5.61}$$

$$E[N] = \frac{0.8}{1 - 0.8} - \frac{6 \cdot 0.8^6}{1 - 0.8^6} \tag{5.62}$$

5.10.2 Example - A Network Interface

[To revise!]

User7 from Figure 4.1 is viewing online TV on his computer. The packets with the streaming data come from the *TV broadcasting server* and pass through the network elements AN6, R1, R3 and AN4. In this example, we will evaluate the delay that packets suffer in the AN4, which is a WLAN. We assume that User 8 and User 9 are not connected to the WLAN, so all the WLAN bandwidth is used by User7.

The AP sends packets to User 7 at a transmission rate of $R = 22$ Mbps. We assume that the service time, D_s, for each packet follows an exponential distribution, with average

$$D_s = DIFS + E[\text{BO}] + \frac{L_h + E[L]}{R} + SIFS + \frac{L_{\text{ACK}}}{R} \tag{5.63}$$

where DIFS= 34 μs, SIFS= 16 μs, $E[\text{BO}] = 90$ μs are parameters of the WLAN. $L_{\text{ACK}} = 112$ bits is the length of MAC-layer ACK which is sent by the receiver after receiving a packet and $L_h = 230$ bits is the length of the MAC header which is added to each TV data packet. The packets have an average size of $E[L] = 4000$ bits.

If the *TV broadcasting server* send packets to User7 at a rate $\lambda = \frac{B_{\text{stream}}}{E[L]}$ following a Poisson process, with $B_{\text{stream}} = 8$ Mbps the bandwidth required by the TV flow, and the queue size at AN4 is $K = 10$ packets, compute $E[D_s]$, $E[D_q]$ and $E[D]$.

Solution: The AP can be modelled by a M/M/1/K queue, with $K = 10$, $\lambda = 500$ packets, $E[D_s] = 3.342$ ms and $\mu = 1/E[D_s] = 2992$ packets. The offered traffic is $a = \frac{\lambda}{\mu} = 0.668$ Erlangs.

First, we compute the Equilibrium Distribution. The results are shown in Table 5.1.

State	Value	State	Value
π_0	0.3356266	π_5	0.0447631
π_1	0.2243206	π_6	0.0299180
π_2	0.1499277	π_7	0.0199961
π_3	0.1002062	π_8	0.0133647
π_4	0.0669742	π_9	0.0089325
-	-	π_{10}	0.0059701

Table 5.1: Equilibrium Distribution for the WLAN Exercise

The blocking probability is

$$P_b = \pi_K = \pi_{10} = 5.9 \cdot 10^{-3}$$

.

The expected system occupation is

$$E[N] = \sum_{k=0}^{K} \pi_k k = 1.8830 \text{ packets}$$

and the expected system delay can be obtained by applying the Little's Law

$$E[D] = \frac{E[N]}{\lambda(1 - P_b)} = 0.947 \cdot 10^{-3} \text{ seconds.}$$

The expected waiting delay is

$$E[D_q] = E[D] - E[D_s] = 0.613 \cdot 10^{-3} \text{ seconds.}$$

Alternatively, it can obtained by computing first $E[N_q] = \sum_{k=2}^{K} \pi_k(k-1) = 1.2186$ packets, and then applying Little's law.

From those results, we can observe that User7 will be able to watch the TV without suffering neither high packet losses nor high delays.

5.10.3 Example - Is $K = \infty$ a good approximation?

Here, we consider the same scenario as in the previous example. The goal now is to evaluate what is the impact of assuming that the buffer size is infinite in the performance metrics that we can obtain.

First, assuming that $K = \infty$, the $E[D]$ and $E[D_q]$ values for different TV stream bandwidth values are:

-	TV stream bandwidth (Mbps)		
Parameter	2	6	10
$E[D_q]$	6.7041e-05	3.3589e-04	0.0016968
$E[D]$	4.0122e-04	6.7007e-04	0.0020309

Table 5.2: $E[D_q]$ and $E[D]$ assuming $K = \infty$

Considering the case with the highest stream bandwidth, $B = 10$ Mbps, what is the value of K that gives similar values for $E[D_q]$ and $E[D]$ when they are compared with the case of $K = \infty$? The results obtained are shown in Table 5.3.

K	P_b	$E[D_q]$	$E[D]$
5	1.0148e-01	5.4986e-04	8.8404e-04
10	0.0316381	0.0010332	0.0013674
15	0.0117574	0.0013343	0.0016685
20	0.0046219	0.0015082	0.0018423
25	0.0018554	0.0016024	0.0019366
30	7.5097e-04	1.6510e-03	1.9851e-03
40	1.2401e-04	1.6867e-03	2.0209e-03
50	2.0534e-05	1.6947e-03	2.0289e-03
60	3.4014e-06	1.6963e-03	2.0305e-03
70	5.6349e-07	1.6967e-03	2.0309e-03

Table 5.3: $E[D_q]$ and $E[D]$ for a TV stream bandwidth value of 10 Mbps

We can observe that, for medium/high traffic loads (10 Mbps, $a = 0.83$ Erlangs), even for small K values, the performance metrics obtained using the M/M/1/K queueing system are close to those obtained when the M/M/1 queueing system is used.

Obviously, when $a \to 1$, the use of the M/M/1 queue will not be accurate. However, even for medium/high values of the offered traffic (a), as we have seen in this example, the assumption that the buffer size is infinite allows us to use the less complex formulas of the M/M/1 queueing system without significantly compromising the accuracy of the results.

5.10.4 WIFI Downlink Model

1. The packet arrival rate per flow is $\{\lambda_1,\ \lambda_2,\ \ldots\ , \lambda_N\}$.
2. The aggregate packet arrival rate is $\lambda = \sum_{n=1}^{N} \lambda_n$
3. The packet transmission time per flow is given by

$$\{T_1(R_1, L_1),\ T_2(R_2, L_2),\ \ldots\ , T_N(R_N, L_N)\},$$

   where R_n is the transmission rate experienced by station n, and L_n is the size (average) of the packets directed to station n.

4. The expected packet transmission time of the system is given by:

$$E[D_s] = \sum_{n=1}^{N} \frac{\lambda_n}{\lambda} T_n(R_n, L_n)$$

5. Normalized traffic load per flow: $a_n = \lambda_n T_n(R_n, L_n)$
6. Normalized aggregate traffic load: $a = \lambda E[D_s] = \sum_{n=1}^{N} a_n$
7. We obtain the stationary probability distribution: $\pi_n = \frac{(1-a)a^n}{1-a^{K+1}},\ \forall\ n$, where $K = Q + 1$ is the AP packets storage capacity (including the packet in transmission).
8. The AP utilization is given by $\rho = 1 - \pi_0 = a(1 - P_b)$

9. The packet blocking probability is $P_b = \pi_K$
10. The expected AP delay for flow n is $E[D]_n = T_n(R_n, L_n) + E[D_q]$. Note that the term $E[D_q]$ is the same for all flows, as the time a packet waits in the buffer depends on the packets that have arrived before it (and are still waiting).
11. The expected AP delay is given by $E[D] = \sum_{n=1}^{N} \frac{\lambda_n}{\lambda} E[D]_n = \frac{E[N]}{\lambda(1-P_b)}$
12. The expected AP buffer delay is given by $E[D_q] = \frac{E[N_q]}{\lambda(1-P_b)}$

Chapter 6

End-to-end Delay

6.1 Queueing Networks

In the previous chapter, we have analyzed a variety of systems using different queueing models. However, all of these models have one thing in common: they describe only a single networking element (or even just a part of one). This is sufficient if the goal is to dimension this part of a network, or to identify isolated problem spots.

However, there may be cases where a complete network path from its source to its destination needs to be considered. For instance, consider again an audio streaming application. As we discussed earlier, the path end-to-end delay and the deviation from its average are very important for the quality of this stream, and for parameterizing application components such as buffers.

Therefore, it would be helpful if we could analyze the sojourn time of a packet going through a sequence of queueing systems instead of just a single one. A simple first approach is to compute the sojourn time for each of the systems individually, and then adding up these values. However, this means that we will ignore the fact that packet stream *leaving* a queueing system does not necessarily have the same stochastic properties as the arrival process. Since the departure process also depends heavily on the service time distribution, it may no longer be Markovian.

One may argue that we have made the same kind of error already in assuming a

Markov arrival process in any of the previous models. We justified this assumption with the large number of individual packet flows that together form the packet arrival process of a network core element. While this assumption has its merits, we will now use a more exact method of describing and analyzing networks of queues, only assuming properties of packet flows entering the network and for the routing of packets between its elements. We will compare the results using this method to the aforementioned simple approach of adding up individual results, and finally also will use a simulation to generate yet another set of values for comparison.

This 'competition' of different performance evaluation approaches will allow us to recognize the limits of our queueing theory approach, thus enabling us to judge how far we can go in using them for system analysis.

6.2 Jackson Networks

Packets traverse multiple hops from their source to their destination. For example, if User4 and User1 have VoIP conversation, their packets can traverse AN1, R2, R4 and AN2.

In this chapter we explain how the average end-to-end packet delay can be computed. To compute the end-to-end delay we have to know the following information:

- The route that the packets follow from their source to their destination. The route that a packet follows is indicated by the set of links $\mathcal{F}$ it traverses.
- The aggregate packet arrival rate to every node in the route.
- The stochastic routing vector $\boldsymbol{\alpha}$ at each network element. The stochastic routing vector indicates the probability that a packet is directed to a certain output network interface.

With this information, the end-to-end delay is the sum of the average times a packet has spent in each hop.

$$E[D_{\mathrm{e2e}}] = \sum_{\forall f \in \mathcal{F}} (E[D_{\mathrm{node}}]_f + E[D_{\mathrm{p}}]_f) \tag{6.1}$$

where $E[D_{\text{node}}]_f$ and $E[D_{\text{p}}]_f$ are the expected delay at node f and the propagation delay at hop f respectively. Usually, we will simply assume that $E[D_{\text{node}}]_f = E[D]_f$.

However, to apply (6.1), we have to make several assumptions:

- At each element, the packets have to arrive following a Poisson process.
- Each network interface has to be modelled assuming $K = \infty$, with the exception of the last hop.
- The offered traffic at each network interface must be $a < 1$, i.e., the queueing system must be stable.
- Stochastic routing must be applied, i.e., the next hop for a given packet is selected randomly.
- The service time has to be exponentially distributed, with the exception of the last hop.
- The service times of the same packet in different queues have to be independent.

These assumptions are based on the Burkes and Jackson's theorems, as well as, the work done by Kleinrock in his PhD thesis.

6.3 Burke and Jackson's theorems

Burke proved that the departure process of a M/M/S queueing system is also a Poisson process with rate λ. Therefore, it allows us to concatenate several M/M/S systems.

Jackson proved that considering the assumptions stated in the previous section, the network state can be computed as a product of the state of all individual network interfaces. In other words, that each network interface behaves independently of the others. For example, the probability that the network contains 0 packets is $\prod_{\forall n \in \text{Network nodes}} \pi_{0,n}$.

6.4 Model of a node in a network

In Figure 6.1 the schematic of a node (router) is depicted. It has M_{in} input links and M_{out} output links. For instance, in Figure 4.1, the R4 has 4 input and 4 output links, as all links are bidirectional.

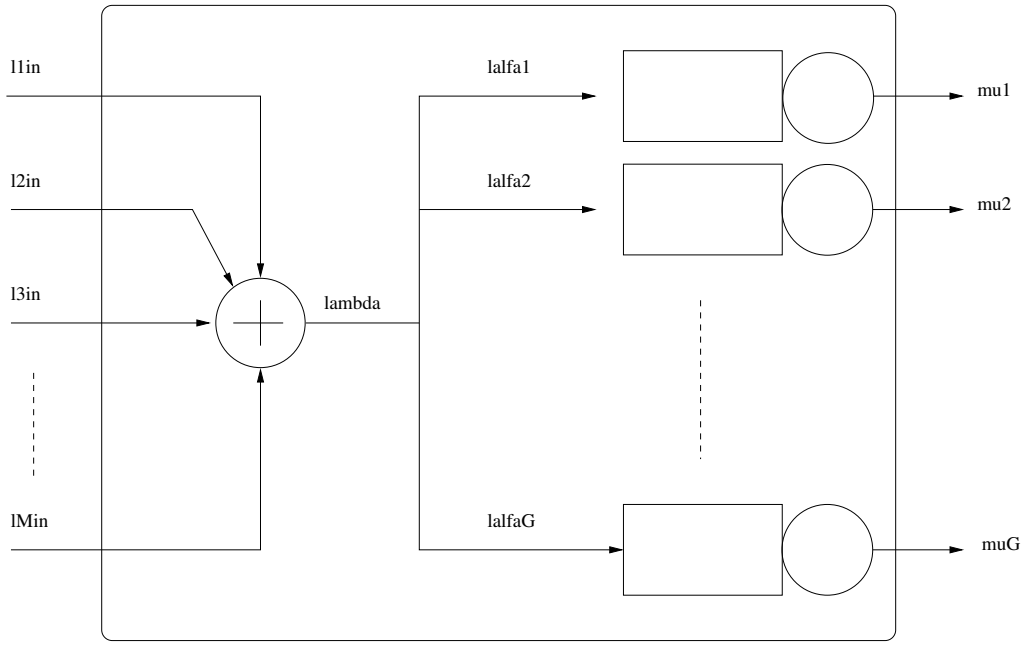


Figure 6.1: Schematic of a Node

6.5 Examples for End-to-end Delay

End to End delay for a video transmission

Consider that User9 is watching a video transmitted by the video server connected to AN6. The route that the packets follow has 5 hops (Video Server, AN6, R1, R3, AN4 and User9). Each hop is modeled by an M/M/1 queue as shown in Figure 6.2.

Given that $\lambda = 200$ packets/second, and $E[L] = 2000$ bits, compute the expected end-to-end packet delay.

Solution: We have to compute the $E[D_{\text{hop}}] = E[D] + E[D_p]$ for each hop:

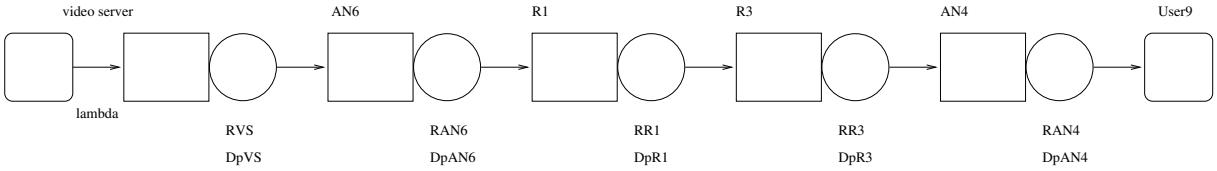


Figure 6.2: Network for Example 1

- $E[D_{\text{hop,VS}}] = \frac{1}{\frac{10E6}{2000} - 200} + 0.001 \cdot 10^{-3} = 2.0933 \cdot 10^{-4}$ seconds.
- $E[D_{\text{hop,AN6}}] = \frac{1}{\frac{100E6}{2000} - 200} + 4 \cdot 10^{-3} = 0.0040201$ seconds.
- $E[D_{\text{hop,R1}}] = \frac{1}{\frac{100E6}{2000} - 200} + 1 \cdot 10^{-3} = 0.0010201$ seconds.
- $E[D_{\text{hop,R3}}] = \frac{1}{\frac{100E6}{2000} - 200} + 0.001 \cdot 10^{-3} = 2.1080 \cdot 10^{-5}$ seconds.
- $E[D_{\text{hop,AN4}}] = \frac{1}{\frac{22E6}{2000} - 200} + 2 \cdot 10^{-3} = 0.0020926$ seconds.

Finally, the end-to-end delay is computed by adding all the previous delays, and the result is:

$$E[D_{\text{e2e}}] = 7.3632 \quad \text{msecs}$$

Background Traffic

In this second example, there appears background traffic in the same route of the video packets, as shown in Figure 6.3. Consider that the average packet length is $E[L] = 2000$ bits and that

$$\alpha_{R4} = \frac{\lambda_{B1}}{\lambda_{B1} + \lambda} = 0.99593 \qquad \alpha_{User8} = \frac{\lambda_{B2}}{\lambda_{B2} + \lambda} = 0.92308$$

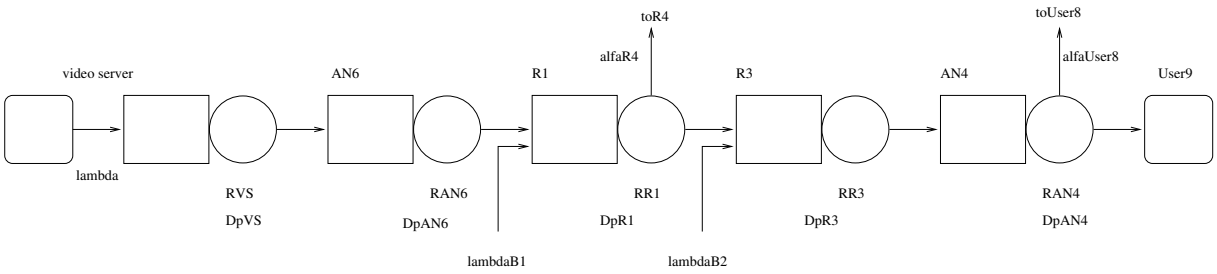


Figure 6.3: Network for Example 2

Solution: We have to compute the $E[D_{\text{hop}}] = E[D] + E[D_p]$ for each hop:

- $E[D_{\text{hop,VS}}] = \frac{1}{\frac{10E6}{2000} - 200} + 0.001 \cdot 10^{-3} = 2.0933 \cdot 10^{-4}$ seconds.
- $E[D_{\text{hop,AN6}}] = \frac{1}{\frac{100E6}{2000} - 200} + 4 \cdot 10^{-3} = 0.0040201$ seconds.
- $E[D_{\text{hop,R1}}] = \frac{1}{\frac{100E6}{2000} - (49000+200)} + 1 \cdot 10^{-3} = 0.0022500$ seconds.
- $E[D_{\text{hop,R3}}] = \frac{1}{\frac{100E6}{2000} - (2400+200)} + 0.001 \cdot 10^{-3} = 2.2097 \cdot 10^{-5}$ seconds.
- $E[D_{\text{hop,AN4}}] = \frac{1}{\frac{22E6}{2000} - (2400+200)} + 2 \cdot 10^{-3} = 0.0021190$ seconds.

Finally, the end-to-end delay is computed by adding all the previous delays, and the result is:

$$E[D_{\text{e2e}}] = 8.6206 \text{ msecs}$$

Part III

Miscellaneous Traffic and Quality of Service

Chapter 7

Heterogeneous Traffic in IP Networks

7.1 Observations about Real Packets

In the previous part, we assumed several things about IP packets in order to be able to model a data link using a Markov chain model. One of these assumptions, namely that packet sizes are exponentially distributed, is especially unrealistic considering real packet-switched networks. To start, it is pretty obvious that packets can only contain a discrete number of bits, whereas the exponential distribution is continuous. Moreover, there typically is a minimum packet size significantly larger than 0, owed to the headers that each packet needs to contain. In the case of IP version 4 packets, not considering lower layer headers, this minimum size would be 20 Byte (40 Byte in version 6), although normally one can assume that each IP packet contains at minimum a transport layer header as well.

Finally, current packet switched networks typically impose a limit on the maximum size of individual packets, the Maximum Transmission Unit (MTU). A typical value for the MTU is 1500 Byte, resulting from the use of Ethernet V2 technology. Packets that are larger than this MTU above the network layer are fragmented by the IP so that the fragments conform to the limit. This means that any realistic packet size distribution should have a finite maximum value, whereas the exponential distribution does not. To illustrate the difference between the Markov model assumption

and the real world, Figure 7.1 shows packet sizes observed on a core network link (available via www.caida.org) and compares the resulting distribution with an exponential distribution with a similar average value. The discrepancies between the two are obvious.

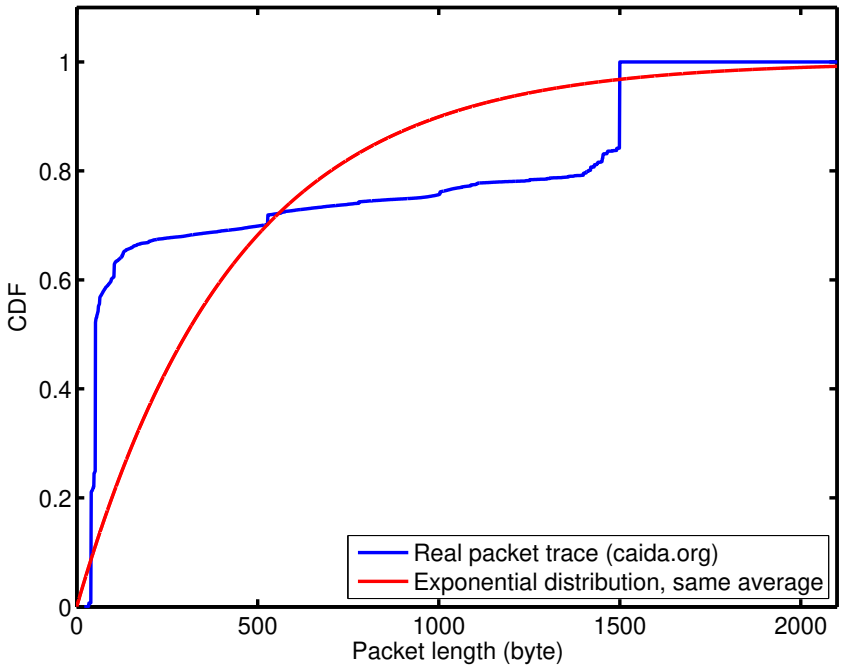


Figure 7.1: Real packet size distribution vs. exponential distribution with same mean

What does this mean for our formulas derived for M/M/S/K queues? Basically, it means that if we use these formulas, we have to keep in mind that they result from a simplification (and maybe oversimplification) of the observed system. This does not mean that they are completely useless, just that they provide only a first, somewhat unaimed shot at analyzing the system in question. However, that does not mean we have to stop there.

In the following, we will provide analytical tools and their derivation for systems with packet size distributions different than an exponential distribution. Better than that, these formulas are valid for a very large class of distributions, namely all packet size distributions which are *general* and *independent*. The first attribute means that the sizes of all packets are determined by the same distribution, and the second that the size of each packet is independent from the size of other packets. This is similar

to rolling a dice, where the outcome of one roll does not depend on the numbers rolled before, but the same distribution, i.e., dice, is used for each roll. Packet sizes in real networks fulfill these requirements rather well, so that this should lead us to a much more exact model than the one from the previous chapter.

7.2 M/G/1 Waiting System

In order to be able to model more realistic packet length distributions, we need to replace the exponential service time distribution with a general, independent distribution, 'G' in the Kendall notation. The resulting M/G/1 waiting system, cf. Figure 7.2, will be analyzed in the following.

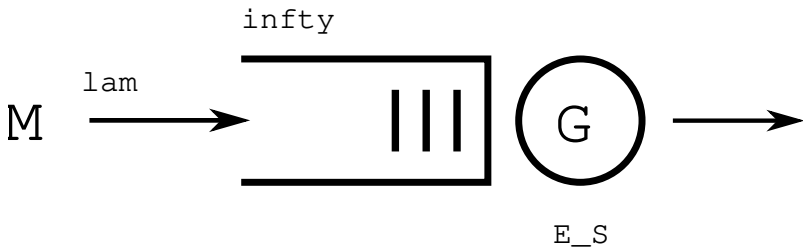


Figure 7.2: A M/G/1 waiting system

In this system, the utilization ρ and the traffic intensity still depend on the average service time $E[D_s]$:

$$\rho = a = \lambda \cdot E[D_s].$$

For the system to be in a stable condition, it still holds that $a < 1$.

We are again interested in the same important system characteristics, such as the average time spent in the queue $E[D_q]$, the average system response or sojourn time $E[D]$, and the average queue length $E[N_q]$. The complete distributions for these values can be derived using a so-called embedded Markov chain. However, since we are now only interested in the average values, we choose a simpler approach. We will analyze the system from the viewpoint of an arriving packet, cf. Figure 7.3.

Under a FIFO queueing strategy, this newly arrived packet will have to wait on

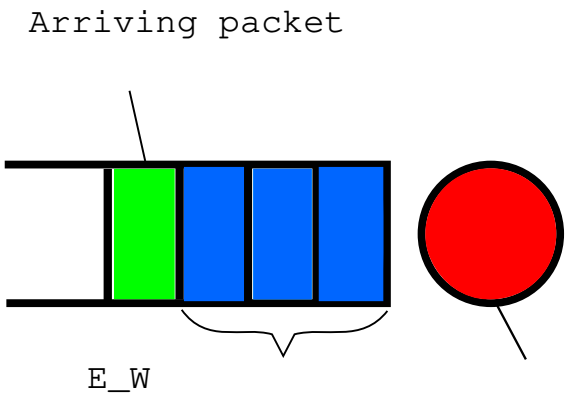


Figure 7.3: Consideration for the average waiting time

average until the packet that is currently being sent is completely transmitted, and then until all the packets before it in the queue are also transmitted. The former is the residual service time $E[D_r]$, while the latter is the product of the average number of packets encountered in the queue by arriving packets $E[N_q^a]$ and the average service time needed for each:

$$E[D_q] = E[N_q^a] \cdot E[D_s] + E[D_r]. \tag{7.1}$$

Since we still assume a Poisson arrival process, the PASTA property still holds. This means that the average number of packets in the queue seen by an arriving packet equals the average number of packets in the queue over time, or $E[N_q^a] = E[N_q]$.

We can use Little's Law to derive $E[N_q]$:

$$E[N_q] = \lambda \cdot E[D_q]. \tag{7.2}$$

Using (7.2) with (7.1) gives us

$$\begin{aligned} E[D_q] &= \lambda \cdot E[D_q] \cdot E[D_s] + E[D_r] \\ E[D_q] &= \rho \cdot E[D_q] + E[D_r] \\ E[D_q](1-\rho) &= E[D_r] \\ E[D_q] &= \frac{E[D_r]}{(1-\rho)} \end{aligned} \tag{7.3}$$

Next, we have to quantify the average residual service time $E[R]$. To give an intuition for this, Figure 7.4 shows a qualitative plot of the residual work over time.

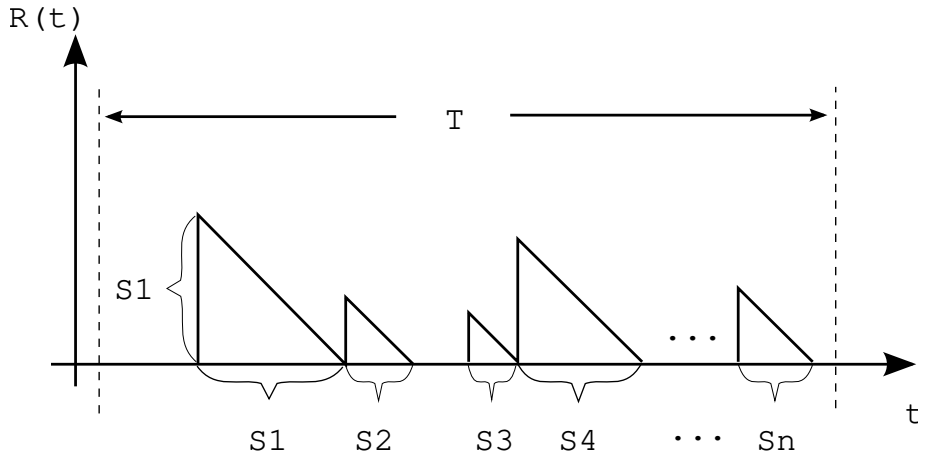


Figure 7.4: Residual service time process

For the average residual service time, we consider the system over a long timespan T. In this interval, we will see on average $\lambda \cdot T = n$ packets arriving. Then,

$$E[D_r] = \frac{1}{T}\int_0^T D_r(t')dt' = \frac{1}{T}\sum_{i=1}^{n}\frac{1}{2}D_{s_i}^2 = \underbrace{\frac{n}{T}}_{\to\lambda} \cdot \underbrace{\frac{1}{n}\cdot\sum_{i=1}^{n}\frac{1}{2}D_{s_i}^2}_{\to\frac{1}{2}E[D_s^2]}.$$

Finally, we get

$$E[D_q] = \frac{E[D_r]}{(1-\rho)} = \frac{\lambda \cdot E[D_s^2]}{2(1-\rho)} \left(= \frac{1+CV[D_s]^2}{2} \cdot \frac{\rho}{1-\rho} \cdot E[D_s]\right). \tag{7.4}$$

Knowing the average waiting time, we can calculate the average system response

time as

$$E[D] = E[D_q] + E[D_s] = \frac{\lambda \cdot E[D_s^2]}{2(1-\rho)} + E[D_s] \tag{7.5}$$

Applying Little's Law again using (7.4) and (7.5), we get the average number of packets in the queue and in the system, respectively:

$$E[N_q] = \lambda \cdot E[D_q] = \frac{\lambda^2 \cdot E[D_s^2]}{2(1-\rho)} = \frac{1 + CV[D_s]^2}{2} \cdot \frac{\rho^2}{1-\rho} \tag{7.6}$$

$$E[N] = \lambda \cdot E[D] = \frac{\lambda^2 \cdot E[D_s^2]}{2(1-\rho)} + \lambda \cdot E[D_s] = \frac{1 + CV[D_s]^2}{2} \cdot \frac{\rho^2}{1-\rho} + \rho \tag{7.7}$$

7.2.1 Averaging

$$\begin{aligned} D_{q,i} &= D_{r,i} + \sum_{n=1}^{N_{q,i}} D_{s,n} \\ D_{q,i} &= D_{r,i} + N_{q,i} \left(\frac{1}{N_{q,i}} \sum_{n=1}^{N_{q,i}} D_{s,n} \right) \\ D_{q,i} &= D_{r,i} + N_{q,i} E[D_s] \\ \rightarrow E[D_q] &= E[D_r] + E[N_q] E[D_s] \end{aligned}$$

Application to M/M/1 waiting systems

Since the M/M/1 waiting system is a special case of the M/G/1 waiting system, we can apply the results for the latter and compare it with the results gained by the Markov chain-based approach. Since $CV[D_s] = 1$ in a M/M/1 system, we get

$$E[D_q] = \frac{1 + CV[D_s]^2}{2} \cdot \frac{\rho}{1-\rho} \cdot E[D_s] = \frac{\rho}{1-\rho} \cdot E[D_s],$$

and

$$E[N] = \frac{1 + CV[D_s]^2}{2} \cdot \frac{\rho^2}{1-\rho} + \rho = \frac{\rho^2}{1-\rho} + \rho = \frac{\rho^2}{1-\rho} + \frac{\rho(1-\rho)}{1-\rho} = \frac{\rho}{1-\rho},$$

which are the known formulas for M/M/1.

[Before showing how the Residual time looks, it's an interesting exercise to compute it as: $\frac{E[D_r]}{1-a} + E[D_s] = \frac{1}{\mu-\lambda}$]

Application to M/D/1 waiting systems

As a second case for a specific class of service time distribution, we apply the M/G/1 analysis to a M/D/1 waiting system. Here, $CV[D_s] = 0$ due to the deterministic service process. Therefore,

$$E[D_q] = \frac{1 + CV[D_s]^2}{2} \cdot \frac{\rho}{1-\rho} \cdot E[D_s] = \frac{1}{2} \cdot \frac{\rho}{1-\rho} \cdot E[D_s],$$

or exactly half the average waiting time of a M/M/1 waiting system with the same average service time and load. Similarly,

$$E[N] = \frac{1 + CV[D_s]^2}{2} \cdot \frac{\rho^2}{1-\rho} + \rho = \frac{1}{2} \cdot \frac{\rho^2}{1-\rho} + \rho.$$

7.2.2 Comments

We have a traffic flow that has two different service time values: $D_{s,1,1}$ and $D_{s,1,2}$, with probability $p_{1,1}$ and $p_{1,2}$, respectively. Then, the second moment is:

$$E[D_{s,1}^2] = p_{1,1} D_{s,1,1}^2 + p_{1,1} D_{s,1,2}^2 \tag{7.8}$$

If we add another traffic flow that has also two different service time values: $D_{s,2,1}$ and $D_{s,2,2}$, with probability $p_{2,1}$ and $p_{2,2}$, and we know that the fraction of arriving packets from flow 1 is ψ_1 and from 2 is ψ_2, the second moment of the mixture

is:

$$\begin{aligned} E[D_s^2] &= \psi_1(p_{1,1}D_{s,1,1}^2 + p_{1,2}D_{s,1,2}^2) + \psi_2(p_{2,1}D_{s,2,1}^2 + p_{2,2}D_{s,2,2}^2) \\ &= \psi_1 E[D_{s,1}^2] + \psi_2 E[D_{s,2}^2] \end{aligned} \tag{7.9}$$

where $\psi_1 = \frac{\lambda_1}{\lambda_1+\lambda_2}$ and $\psi_1 = \frac{\lambda_2}{\lambda_1+\lambda_2}$.

Therefore, with multiple traffic flows, the queueing delay in a M/G/1 queue reduces to:

$$\begin{aligned} E[D_q] &= \frac{\lambda \sum_i \psi_i E[D_{s,i}^2]}{2(1-a)} = \frac{\sum_i \lambda_i E[D_{s,i}^2]}{2(1-a)} = \\ &= \frac{\sum_i \lambda_i E^2[D_{s,i}](1+\mathrm{CV}^2[D_{s,i}])}{2(1-a)} = \\ &= \frac{\sum_i a_i E[D_{s,i}](1+\mathrm{CV}^2[D_{s,i}])}{2(1-a)} = \\ &= \frac{\sum_i a_i \frac{E^2[D_{s,i}](1+\mathrm{CV}^2[D_{s,i}])}{2E[D_{s,i}]}}{(1-a)} = \frac{\sum_i a_i \frac{E[D_{s,i}^2]}{2E[D_{s,i}]}}{(1-a)} = \\ &= \frac{\sum_i E[D_{r,i}]}{(1-a)} \end{aligned} \tag{7.10}$$

7.3 Examples for the Use of M/G/1

Example 1: Impact of the variance of the packet length in the core network

A data flow aggregate sent over a core link, cf. Figure 7.5, contains packets of an average length $E[L] = 800\,\mathrm{bit}$, which arrive following a Poisson arrival process at a link with a queue sufficiently large to be considered infinite. The average arrival rate is $\lambda = 10^7\,\frac{1}{\mathrm{s}}$, and the capacity of the link is $R = 10\,\mathrm{Gbps}$.

We will now compare the waiting times of these packets in the buffer for packet length distributions. These distributions serve as a first and very rough modeling of different application classes.

1. Calculate the average waiting time of a packet if all packets are of fixed length L, such as in a voice streaming application.

2. Compare the previous result to the waiting time for exponentially distributed

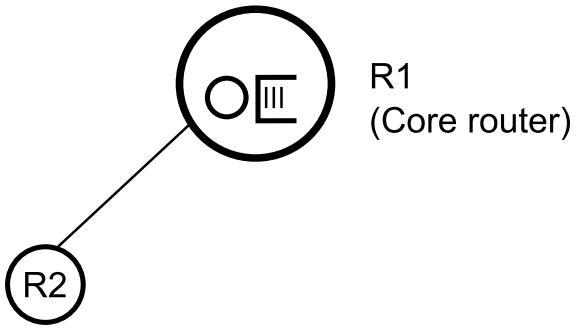


Figure 7.5: Core network link

packet lengths.

3. Compare the previous results to the waiting time if the packet lengths show a coefficient of variation of $CV[L] = 3.2$. What is the conclusion?

Solution:

1. In this first case, the packets have a fixed length. Therefore, the service time distribution is deterministic. Every packet needs the same time to be transmitted:

$$E[D_s] = \frac{E[L]}{R} = \frac{800\,\text{bit}}{10\,\text{Gbps}} = 8 \cdot 10^{-8}\,\text{s} = 8 \cdot 10^{-5}\,\text{ms}$$

$$\rho = \lambda \cdot E[D_s] = 10^7\,\frac{1}{\text{s}} \cdot 8 \cdot 10^{-8}\,\text{s} = 0.8$$

With a deterministic service time distribution, we can directly model the system as a M/D/1 waiting system. For this kind of system, the waiting time is:

$$E[D_q] = \frac{\rho}{1-\rho}\frac{E[D_s]}{2} = \frac{0.8}{0.2}\frac{8 \cdot 10^{-8}\,\text{s}}{2} = 16 \cdot 10^{-8}\,\text{s} = 16 \cdot 10^{-5}\,\text{ms}$$

2. With the change in the packet length distribution, the service time distribution changes as well. The new system is therefore an M/M/1 waiting system, where the average waiting time is:

$$E[D_q] = \frac{\rho}{1-\rho}E[D_s] = \frac{0.8}{0.2} \cdot 8 \cdot 10^{-8}\,\text{s} = 32 \cdot 10^{-8}\,\text{s} = 32 \cdot 10^{-5}\,\text{ms},$$

or double the waiting time of the previous case.

3. Finally, for the general M/G/1 case, we use Formula 7.4:

$$\begin{aligned} E[D_q] &= \frac{1+CV[D_s]^2}{2}\frac{\rho}{1-\rho}E[D_s] = \frac{1+3.2^2}{2}\frac{0.8}{0.2}\cdot 8\cdot 10^{-8}\,\mathrm{s} \\ &= 179.84\cdot 10^{-8}\,\mathrm{s} = 179.84\cdot 10^{-5}\,\mathrm{ms}. \end{aligned}$$

Comparing the three results, we see that the variance of the service time has a large effect of the waiting time, all other things (and specifically the average service time) being equal. Specifically, a higher variance in the service time process leads to longer waiting times.

Example 2: TCP data traffic

As stated before, the Internet carries packets of different sizes. Measurements show that two very important classes of IP packets are data packets that utilize the full Maximum Transfer Unit (MTU) of Ethernet networks, i.e., 1500 Byte, and TCP acknowledgments, which have a length of 40 Byte using IPv4. Using the M/G/1 model, we can now analyze a link that transports these two types of packets. We will use a DSL uplink of a private user as an example, cf. Figure 7.6.

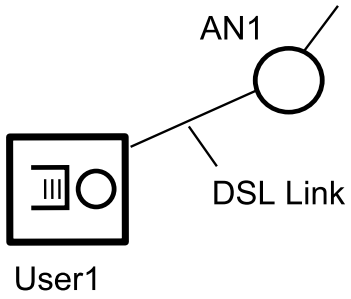


Figure 7.6: DSL uplink

To do so, we model the acknowledgments as packets with a fixed length of $L_{ack} = 40\,\mathrm{Byte}$, and the data packets also with a fixed packet length of $L_{data} = 1500\,\mathrm{Byte}$. The probability for an arriving packet to be an acknowledgment is $p_{ack} = 0.33$ (i.e., we see roughly one acknowledgment for every two data packets), and the total packet arrival rate is $\lambda = 45\,\frac{1}{\mathrm{s}}$. The link capacity is $R = 0.5\,\mathrm{Mbps}$.

We now want to determine the average waiting times and the average number of acknowledgments and data packets (individually and in total) in the system.

Solution:

Since the overall service time distribution is neither deterministic (there is more than one type of packet), nor exponential (in this case, the distribution is even discrete), we have to use the M/G/1 model.

We know that, in order to calculate the waiting time of the packets, we will need the second central moment $E[D_s^2]$ of the waiting time. As well, we need to calculate the average service time $E[D_s]$ to get the system utilization $\rho = \lambda \cdot E[D_s]$. We can do both in a similar fashion. First, we calculate the according values for the individual packet types, and then use a weighted sum based on their probabilities to get the moment of the total distribution:

$$\begin{aligned}
E[D_{s_{ack}}] &= \frac{E[L_{ack}]}{R} = \frac{40 \cdot 8\,\text{bit}}{0.5 \cdot 10^6\,\frac{\text{bit}}{\text{s}}} = 0.64 \cdot 10^{-3}\,\text{s} \\
E[D_{s_{data}}] &= \frac{E[L_{data}]}{R} = 24 \cdot 10^{-3}\,\text{s} \\
E[D_s] &= p_{ack} \cdot E[D_{s_{ack}}] + p_{data} \cdot E[D_{s_{data}}] \\
&= 0.33 \cdot 0.64 \cdot 10^{-3}\,\text{s} + 0.67 \cdot 24 \cdot 10^{-3}\,\text{s} = 16.3 \cdot 10^{-3}\,\text{s}
\end{aligned}$$

$$\begin{aligned}
E[D^2_{s_{ack}}] &= (1 + \overbrace{CV[D_{s_{ack}}]^2}^{=0\text{(det.)}})E[D_{s_{ack}}]^2 = E[D_{s_{ack}}]^2 = 4.1 \cdot 10^{-7}\,\text{s}^2 \\
E[D^2_{s_{data}}] &= (1 + \overbrace{CV[D^2_{s_{data}}]}^{=0\text{(det.)}})E[D_{s_{data}}]^2 = 5.76 \cdot 10^{-4}\,\text{s}^2 \\
E[D_s^2] &= p_{ack} \cdot E[D^2_{s_{ack}}] + p_{data} \cdot E[D^2_{s_{data}}] \\
&= 0.33 \cdot 4.1 \cdot 10^{-7}\,\text{s}^2 + 0.67 \cdot 5.76 \cdot 10^{-4}\,\text{s}^2 = 3.8 \cdot 10^{-4}\,\text{s}^2
\end{aligned}$$

Next, we can calculate the average waiting time of packets, which is also needed for the system response time and therefore for the number of packets in the system (using Little's Law).

The average waiting time is the same for both packet types, since it does not depend on a packet itself, but on the packets that are positioned before it in the buffer:

$$\begin{aligned} E[D_q] &= \frac{\lambda E[D_s^2]}{2(1-\rho)} \\ &= \frac{45\,\frac{1}{\mathrm{s}} \cdot 3.8 \cdot 10^{-4}\,\mathrm{s}^2}{2(1-45\,\frac{1}{\mathrm{s}} \cdot 16.3 \cdot 10^{-3}\,\mathrm{s})} = 3.17 \cdot 10^{-2}\,\mathrm{s} \end{aligned}$$

The average time spent in the system is different for the two packet types, since this is the sum of the average waiting time (which is equal) and the average service time (which is not):

$$E[D_{ack}] = E[D_q] + E[D_{s_{ack}}] = 3.23 \cdot 10^{-2}\,\mathrm{s} \approx 32\,\mathrm{ms}$$

$$E[D_{data}] = E[D_q] + E[D_{s_{data}}] = 5.57 \cdot 10^{-2}\,\mathrm{s} \approx 56\,\mathrm{ms}$$

Finally, knowing the average time a packet spends in the system and the arrival rates of the packets, we can use Little's Law to calculate the average number of these packets in our system:

$$\begin{aligned} E[N_{ack}] &= \lambda_{ack} \cdot E[D_{ack}] = p_{ack} \cdot \lambda \cdot E[D_{ack}] \\ &= 0.33 \cdot 45\,\frac{1}{\mathrm{s}} \cdot 3.81 \cdot 10^{-2}\,\mathrm{s} = 0.48 \end{aligned}$$

$$\begin{aligned} E[N_{data}] &= \lambda_{data} \cdot E[D_{data}] = p_{data} \cdot \lambda \cdot E[D_{data}] \\ &= 0.67 \cdot 45\,\frac{1}{\mathrm{s}} \cdot 5.57 \cdot 10^{-2}\,\mathrm{s} = 1.68 \end{aligned}$$

Example 3: WLAN traffic

Wireless LAN uses CSMA-CA (Carrier Sense Multiple Access - Collision Avoidance) as its medium access protocol. This means that a packet will be sent over the air interface, and if the local WLAN station was the only one transmitting, the transmission was successful. However, if more than one station transmits at the same time, the packets will collide and will have to be retransmitted. This retransmission

is delayed by a random amount of time in order to try to avoid another collision, hence Collision Avoidance.

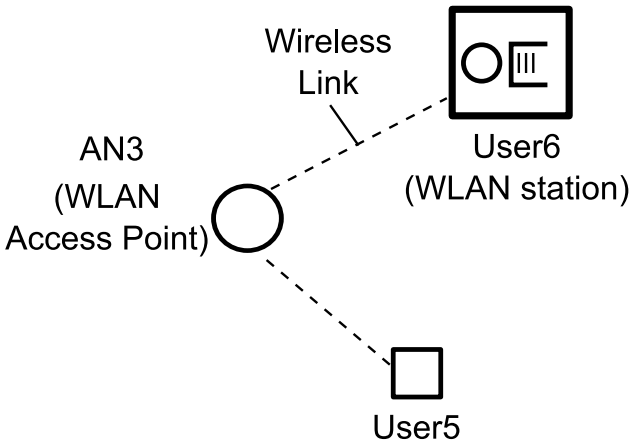


Figure 7.7: A WLAN link

We will in the following show how the M/G/1 model can be used to do a simple analysis of the packet waiting times in a WLAN station, cf. Figure 7.7. To simplify things a bit, we will assume a fixed and independent packet collision probability of $p_{col} = 2\%$ for each packet. In addition, we assume that the backoff time after an unsuccessful transmission is a fixed interval of $D_{bo} = 1\,\text{ms}$, and that a packet is discarded after $Z_{max} = 3$ unsuccessful transmission intents.

We will assume data packets with a fixed size of $L = 8000\,\text{Byte}$ (which is close to the WLAN MTU), and that the wireless network card can transmit packets at $R = 11\,\text{Mbps}$. Packets to be sent over the WLAN are generated at the station following a Poisson process, with an average rate of $\lambda = 100\,\frac{1}{\text{s}}$.

Solution:

Again, the important thing is to characterize the overall service time process for a packet. We can easily calculate the time $E[D_s^*]$ it takes to transmit a packet, regardless of whether it collides with another packet or not:

$$E[D_s^*] = \frac{E[L]}{R} = \frac{8000 \cdot 8\,\text{Bit}}{11 \cdot 10^6\,\frac{\text{bit}}{\text{s}}} = 5.8 \cdot 10^{-3}\,\text{s}$$

The actual service time of a packet, i.e., the time it takes to process the packet, also depends on the number of necessary retransmissions. With probability $1 - p_{col}$, the

packet is successfully transmitted with the first try. With probability $p_{col} \cdot (1-p_{col})$, it is successfully sent after the first retransmission following a collision, and finally, the packet leaves the system in the rest of the cases, either because it was successfully transmitted after two previous collisions, or because it is discarded.

In the first case, we just need $E[D_s^*]$ to transmit the packet, and in the second $E[D_s^*] + D_{bo} + E[D_s^*] = 2E[D_s^*] + D_{bo}$, since even an unsuccessful transmission is completed before the backoff. Similarly, the third case takes $3E[D_s^*] + 2D_{bo}$. Therefore, the total average service time can be calculated as:

$$\begin{aligned} E[D_s] &= (1-p_{col}) \cdot E[D_s^*] + p_{col}(1-p_{col})(2E[D_s^*] + D_{bo}) \\ &\quad + p_{col}^2(3E[D_s^*] + 2D_{bo}) \\ &= 6 \cdot 10^{-3}\,\mathrm{s}. \end{aligned}$$

By now we are familiar with the fact that we need the second moment of the service time as well. Luckily, we can again use the same approach as in the last example, since for each case, the service time is again deterministic:

$$\begin{aligned} E[D_s^2] &= (1-p_{col}) \cdot E[D_s^*]^2 + p_{col}(1-p_{col})(2E[D_s^*] + D_{bo})^2 \\ &\quad + p_{col}^2(3E[D_s^*] + 2D_{bo})^2 \\ &= 3.65 \cdot 10^{-5}\,\mathrm{s} \end{aligned}$$

With $\rho = \lambda \cdot E[D_s] = 0.6$, we again have everything we need to calculate the average waiting time of packets:

$$E[D_q] = \frac{\lambda E[D_s^2]}{2(1-\rho)} = \frac{100\,\frac{1}{\mathrm{s}} \cdot 3.65 \cdot 10^{-5}\,\mathrm{s}^2}{2(1-0.6)} = 4.5 \cdot 10^{-3}\,\mathrm{s}$$

Thus, in total, packets spend on average $E[D] = E[D_q] + E[D_s] = 10.5\,\mathrm{ms}$ in the buffer before they are either successfully transmitted or discarded. The probability for this packet loss is $P_r = p_{col}^3 = 8 \cdot 10^{-6}$, or 0.0008%.

7.4 Heterogeneous flows: Slides M/G/1

The expected packet size is

$$E[L] = p_1(64 \cdot 8) + p_2(800 \cdot 8) + p_3(1500 \cdot 8) = 5710.4 \text{ bits.} \tag{7.11}$$

The expected service time is given by

$$E[D_s] = \frac{E[L]}{R} = p_1 \frac{64 \cdot 8}{10 \cdot 10^6} + p_2 \frac{800 \cdot 8}{10 \cdot 10^6} + p_3 \frac{1500 \cdot 8}{10 \cdot 10^6} = 0.571 \text{ ms.} \tag{7.12}$$

Note that:

$$E[D_s] = p_1 E[D_{s,1}] + p_2 E[D_{s,2}] + p_3 E[D_{s,3}] = p_1 51 \ \mu\text{s} + p_2 0.64 \text{ ms} + p_3 1.2 \text{ ms} \tag{7.13}$$

The second moment of the service time is given by

$$E[D_s^2] = p_1 \left(\frac{64 \cdot 8}{10 \cdot 10^6} \right)^2 + p_2 \left(\frac{800 \cdot 8}{10 \cdot 10^6} \right)^2 + p_3 \left(\frac{1500 \cdot 8}{10 \cdot 10^6} \right)^2 = 0.5871 \ \mu\text{s}^2. \tag{7.14}$$

The expected residual time is then given by

$$E[D_r] = \frac{\lambda E[D_s^2]}{2} = \frac{8 \cdot 10^6}{5710.4} 0.5871 \cdot 10^{-6} = 0.8225 \text{ ms} \tag{7.15}$$

Note that the residual time can be decomposed in the contributions of each packet size:

$$E[D_r] = E[D_{r,1}] + E[D_{r,2}] + E[D_{r,3}] = \frac{\lambda_1 E[D_{s,1}^2]}{2} + \frac{\lambda_2 E[D_{s,2}^2]}{2} + \frac{\lambda_3 E[D_{s,3}^2]}{2} \tag{7.16}$$

where $E[D_{s,i}^2] = \left(\frac{E[L_i]}{R} \right)^2 (1 + CV[D_{s,i}]^2)$.

The expected queueing delay is

$$E[D_q] = \frac{E[D_r]}{1 - \rho} = \frac{0.8225 \cdot 10^{-3}}{1 - \rho} = 0.0021 \text{ s} \tag{7.17}$$

with $\rho = \lambda E[D_s] = \frac{8 \cdot 10^6}{E[L]} E[D_s] = 0.8225$

The expected system delay is

$$E[D] = \frac{E[D_r]}{1-\rho} + E[D_s] = 0.0026 \text{ s} \tag{7.18}$$

The CV of D_s is given by

$$CV[D_s] = \frac{\sqrt{V[D_s]}}{E[D_s]} = \frac{\sqrt{E[D_s^2] - E^2[D_s]}}{E[D_s]} = 0.8947 \tag{7.19}$$

If we compare the obtained delay with the delay of a M/M/1 queue, we obtain:

$$E[D] = \frac{1}{\mu - \lambda} = \frac{1}{\frac{1}{E[D_s]} - \frac{B}{E[L]}} = 0.0029 \text{ s} \tag{7.20}$$

Here, it's exactly the same as before. We first calculate the second moment of each type of size:

$$E[D_{s,i}^2] = E[D_{s,i}]^2 (1 + CV[D_{s,i}]^2) \tag{7.21}$$

where $CV[D_{s,i}] = CV[L_i]$ as $D_{s,i} = \frac{L_i}{R}$, with R a constant.

Then, we can calculate the residual time:

$$E[D_r] = E[D_{r,1}] + E[D_{r,2}] + E[D_{r,3}] = \frac{\lambda_1 E[D_{s,1}^2]}{2} + \frac{\lambda_2 E[D_{s,2}^2]}{2} + \frac{\lambda_3 E[D_{s,3}^2]}{2} \tag{7.22}$$

Note that in previous exercise, since all packet sizes where deterministic, we just considered that $CV[L_i] = 0$ in all cases.

Chapter 8

Traffic Differentiation in IP Networks

8.1 M/G/1 Multiple flows

$$E[D_q] = \sum_{f=1}^{F} E[N_{q,f}]E[D_{s,f}] + \sum_{f=1}^{F} a_f E[D_{r,f}|a_f]$$

$$E[D_q] = \sum_{f=1}^{F} E[N_{q,f}]E[D_{s,f}] + \sum_{f=1}^{F} a_f \frac{E[D_s^2]}{2E[D_s]}$$

$$E[D_q] = \sum_{f=1}^{F} E[N_{q,f}]E[D_{s,f}] + \sum_{f=1}^{F} \lambda_f E[D_{s,f}] \frac{E[D_{s,f}^2]}{2E[D_{s,f}]}$$

$$E[D_q] = \sum_{f=1}^{F} E[N_{q,f}]E[D_{s,f}] + \sum_{f=1}^{F} \lambda_f \frac{E[D_{s,f}^2]}{2}$$

$$E[D_q] = \sum_{f=1}^{F} E[N_{q,f}]E[D_{s,f}] + \sum_{f=1}^{F} E[D_{r,f}]$$

Example with 2 flows:

$$
\begin{aligned}
E[D_q] &= E[N_{q,1}]E[D_{s,1}] + E[N_{q,2}]E[D_{s,2}] + E[D_{r,1}] + E[D_{r,2]} \\
E[D_q] &= \lambda_1 E[D_q]E[D_{s,1}] + \lambda_2 E[D_q]E[D_{s,2}] + E[D_{r,1}] + E[D_{r,2]} \\
E[D_q](1 - a_1 - a_2) &= E[D_{r,1}] + E[D_{r,2]} \\
E[D_q] &= \frac{E[D_{r,1}] + E[D_{r,2}]}{1 - a_1 - a_2} = \frac{\lambda_1 \frac{E[D_{s,1}^2]}{2} + \lambda_2 \frac{E[D_{s,2}^2]}{2}}{1 - a_1 - a_2} \\
E[D_q] &= \frac{\lambda \frac{E[D_s^2]}{2}}{1 - a_1 - a_2} = \frac{\lambda \frac{E[D_s^2]}{2}}{1 - a}
\end{aligned}
$$

Some relationships:

$$
\begin{aligned}
E[N_q] &= \sum_{f=1}^{F} E[N_{q,f}] \\
E[N_{q,f}] &= \frac{\lambda_f}{\lambda} E[N_q] \qquad (8.1)
\end{aligned}
$$

8.2 M/G/1 Waiting Systems with Priorities

Up to now, we have considered pure FIFO queues, i.e., packets do not 'overtake' each other in the queue. In this kind of system, packets are scheduled only based on their arrival time, but otherwise have the same priority. This is a very good model for the 'best effort' Internet, where packets are indeed not treated differently and the arrival sequence of packets in a buffer is the same as the send sequence.

However, nowadays more and more applications with completely different requirements use the same network. Among these are applications that do not depend much on the end-to-end delay of their individual packets, such as file downloads, web traffic or email. The quality of service of these applications is determined by the arrival time of the last packet of the transmission (i.e., the file transfer, the webpage or the mail). In many cases, even a minor delay in the transmission duration is not of a very large consequence.

In contrast, multimedia applications such as video streaming or Voice over IP (VoIP)

depend heavily on the delay of individual packets. Since the transmitted content (video frames or voice samples) needs to be played out continuously on the receiver side, a delay in only a fraction of the packets may lead to a stall and/or errors in the video, or garbling of the received speech signal.

These circumstances have led to the development of mechanisms and architectures such as Differentiated Services, which enable network operators to provide different classes of service to different types of traffic flows. To give an example, an Internet Service Provider (ISP) could treat packets that it can identify as belonging to multimedia traffic flows preferentially, i.e., giving them a higher priority in packet queues or guaranteeing a minimum delay and flow throughput. The following analysis covers one specific type of queueing system that implements the first policy, i.e., handling different classes of traffic with different priorities.

As an example, we can imagine a link carrying file transfer and VoIP traffic, cf. Figure 8.1.

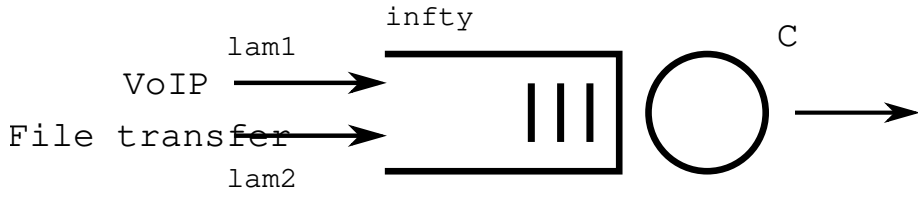


Figure 8.1: Link with two classes of packets

With a normal FIFO scheduling strategy, all packets would experience the same average waiting time, regardless of their type. For example, assume values of $L_1 = 48\,\text{bit}$, $\lambda_1 = 1.21\,\frac{1}{\text{s}}$ and $CV[D_{S_1}] = 0$ for the average packet length, arrival rate, and coefficient of variation for the VoIP packets, respectively, and the according values $L_2 = 960\,\text{bit}$, $\lambda_2 = 4.91\,\frac{1}{\text{s}}$ and $CV[D_{S_2}] = 1$ for the file transfer packets. Then, the average waiting time for all packets equals, using the M/G/1 waiting model, $E[D_q] = 97.64\,\text{ms}$.

However, as we discussed above, in some cases it might improve system performance if some packets could be prioritized, e.g., the (small) VoIP packets of our example. To achieve this, we need to change the scheduling policy in the queue. *Priority scheduling* processes a packet with higher priority before all packets with a lower one, whenever there is one in the queue. This can be done preemptively, i.e., interrupting

the current processing of a low-priority packet for an arriving higher-priority one, or non-preemptive, i.e., finishing sending the packet that is currently being processed even if it has a lower priority than another packet arriving during its service time. Within the same priority class, we still assume a FIFO strategy, i.e., packets of the same priority leave the queue in order of their arrival. This can be visualized as done in Figure 8.2:

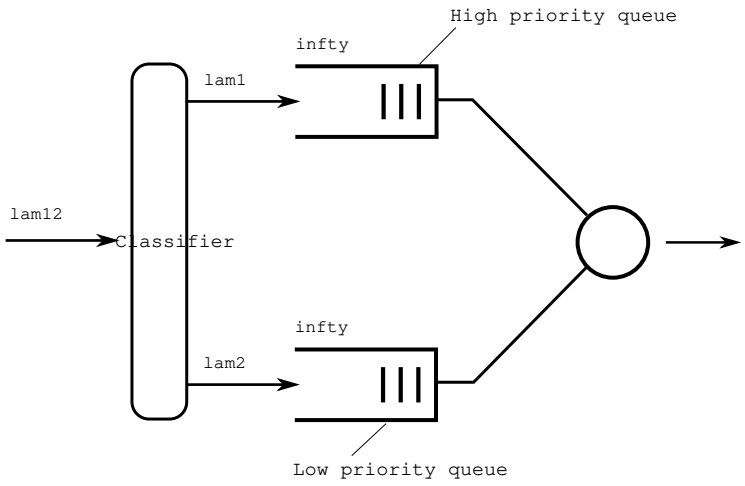


Figure 8.2: System with priority scheduling

We will now adapt our previous M/G/1 analysis to model this new type of system. First, we will derive the average waiting time for the case with two priorities, and then provide the solution for the general case. We assume D_{S_i} to be the service time distribution of packet priority class i, and λ_i the arrival rate of packets of this class. Then, $\rho_i = \lambda_i \cdot E[D_{S_i}]$, with $\rho = \sum_i \rho_i < 1$ necessary for a stable system.

The basic approach is the same as for the standard M/G/1 waiting system. We consider a packet that has just arrived in the system, and the time needed to serve the packets scheduled before it.

We now have to distinguish between two cases, i.e., whether the packet that arrived was a high-priority packet (priority class 1), or a low-priority packet (priority class 2). In the former case, the average waiting time of the packet is

$$\begin{aligned} E[D_{q_1}] &= E[N_{q_1}] \cdot E[D_{S_1}] + E[D_r] \\ &= \lambda_1 \cdot E[D_{q_1}] \cdot E[D_{S_1}] + E[D_r] \\ &= \rho_1 \cdot E[D_{q_1}] + \frac{\lambda}{2} E[D_s^2] \\ \Rightarrow E[D_{q_1}] &= \frac{\lambda E[D_s^2]}{2(1-\rho_1)} \end{aligned}$$

Here, $E[D_r]$ is again the residual service time for the packet being served at the arrival instant. Since we do not preempt this service, it may be a packet of any priority, and we can express $E[D_r]$ in terms of the general service time distribution, like we did for the normal M/G/1 system. However, the number of packets in the queue that have to be processed before the arriving packet is only the number of packets $E[N_{q_1}]$ of the high-priority class, since these take precedence over the lower priority packets. The high-priority packets, in turn, only need $E[D_{s_1}]$ as their average service time, and their number is again found using Little's Law.

For an arriving packet of low priority, the situation is slightly more complex. These packets have to wait until

1. the packet currently being serviced has finished
2. the packets of high priority found in the queue at arrival have been processed
3. the packets of low priority found in the queue at arrival have been processed
4. and finally, the packets of high priority that arrive during the waiting period have been processed as well (since they 'overtake' the packet under consideration).

For the average waiting time of these packets, this results in:

$$
\begin{aligned}
E[D_{q_2}] &= \overbrace{E[N_{q_1}] \cdot E[D_{s_1}]}^{2.} + \overbrace{E[N_{q_2}] \cdot E[D_{s_2}]}^{3.} \\
&\quad + \underbrace{\lambda_1 \cdot E[D_{q_2}] \cdot E[D_{s_1}]}_{4.} + \underbrace{E[D_r]}_{1.} \\
&= \lambda_1 \cdot E[D_{q_1}] \cdot E[D_{s_1}] + \lambda_2 \cdot E[D_{q_2}] \cdot E[D_{s_2}] \\
&\quad + \lambda_1 \cdot E[D_{q_2}] \cdot E[D_{s_1}] + \frac{\lambda}{2} E[D_s^2] \\
&= \rho_1 E[D_{q_1}] + \rho_2 E[D_{q_2}] + \rho_1 E[D_{q_2}] + \frac{\lambda}{2} E[D_s^2] \\
E[D_{q_2}](1 - \rho_2 - \rho_1) &= \rho_1 E[D_{q_1}] + \frac{\lambda}{2} E[D_s^2] \\
E[D_{q_2}] &= \frac{\rho_1 E[D_{q_1}] + \frac{\lambda}{2} E[D_s^2]}{(1 - \rho_2 - \rho_1)} \\
&= \frac{\frac{\rho_1 \frac{\lambda}{2} E[D_s^2]}{(1 - \rho_1)} + \frac{\lambda}{2} E[D_s^2]}{(1 - \rho_2 - \rho_1)} \\
E[D_{q_2}] &= \frac{\lambda E[D_s^2]}{2(1 - \rho_2 - \rho_1)(1 - \rho_1)}
\end{aligned}
$$

This approach can be generalized to the case with C classes of packets, cf. Figure 8.3. Each class has an arrival rate of λ_i, and a service time distribution D_{s_i}, for $i = 1, ..., p$.

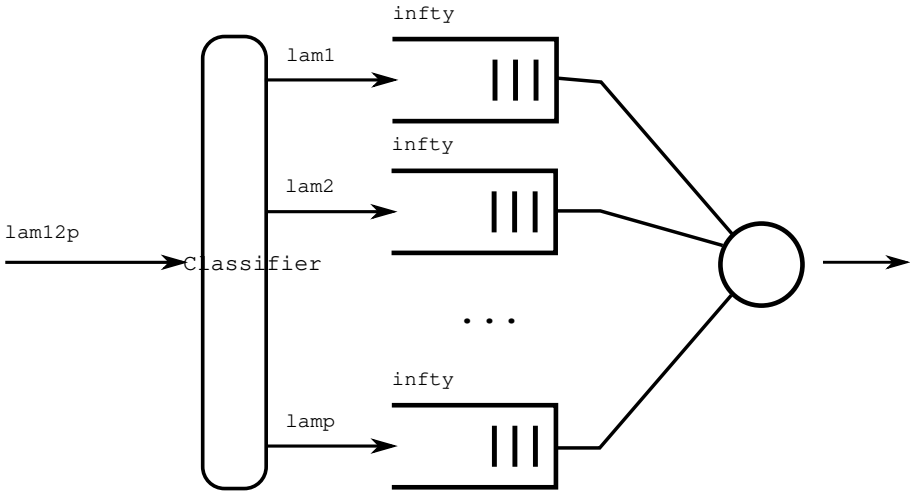


Figure 8.3: Generalized system with priority scheduling

Again, $\rho = \sum_i \rho_i$, with $\rho_i = \lambda_i E[D_{s_i}]$. Then,

$$E[D_{q_i}] = \frac{\lambda E[D_s^2]}{2\prod_{j=i-1}^{i}\left(1-\sum_{z=1}^{j} a_z\right)} \tag{8.2}$$

Using the example from the beginning of the section, we can now calculate the waiting times for the VoIP and file transfer packets if we prioritize the VoIP packets. With the given values, we get $\rho_1 = 0.006$ and $\rho_2 = 0.491$, and thus $E[D_{q_1}] = 49.41\,\text{ms}$ and $E[D_{q_2}] = 98.24\,\text{ms}$. Comparing this with the result for a system without priorities, the VoIP packets, which only make up a small part of the load, now experience much shorter waiting times, while the data packets are affected by only slightly longer waits.

Still, keep in mind that there is no free lunch, i.e., that the much shorter delay of the VoIP packets in this example was achieved only by treating the file transfer packets worse. While the effect in this example may be small due to the chosen values of ρ_1 and ρ_2, this changes when the preferred traffic makes up a significant part of the total. To imagine the worst case, think of a link where there is always a packet of the high priority class to send, i.e., the queue of this traffic class in never empty. In this case, no packet of the lower priorities can be transmitted! Thus, high priority traffic could starve out low priority traffic.

8.3 Examples for M/G/1 with Priorities

Example 1: Voice over IP and background traffic

We consider one router in a network that supports Quality of Service (QoS) for Voice over IP (VoIP) by prioritizing voice traffic over other data traffic (or background traffic), cf. Figure 8.4. We want to calculate the effect this has on the voice packets, by comparing the system with and without the priority policy.

Voice packets have an average length of $E[L_V] = 480\,\text{bit}$, and arrive with an average rate of $\lambda_V = 10\,\frac{1}{\text{s}}$. The average length of data packets is $E[L_D] = 10000\,\text{bit}$, these packets arrive with an average rate of $\lambda_D = 20\,\frac{1}{\text{s}}$. For both types, the interarrival

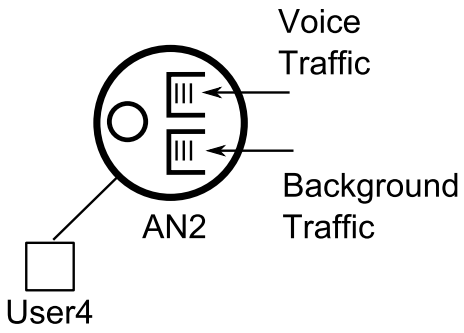


Figure 8.4: An access router supporting QoS

times as well as the packet lengths are distributed exponentially. The outgoing link of the router under consideration has a send rate of $R = 1\,\text{Mbps}$ and its packet queue is large enough to be considered infinite.

1. First, we assume that only voice packets are sent over the link. Calculate the average number of these packets in the system.
2. Now adding the data traffic, calculate the average system response time for voice packets and data packets, respectively.
3. Finally, calculate the number of voice and data packets in the queue and in the system under the condition that the voice packets are prioritized.

Solution:

1. If we just consider the voice traffic, we only have one arrival process with an exponentially distributed interarrival time. Thus, the system is simplified to a M/M/1 waiting system, and we can apply the according formulas for the average waiting time:

$$\begin{aligned} E[D_{s_V}] &= \frac{E[L_V]}{R} = \frac{480\,\text{bit}}{1 \cdot 10^6\,\frac{\text{bit}}{\text{s}}} = 4.8 \cdot 10^{-4}\,\text{s} \\ \rho_V &= \lambda_V \cdot E[D_{s_V}] = 50\,\frac{1}{\text{s}} \cdot 4.8 \cdot 10^{-4}\,\text{s} = 0.024 \\ E[D_q] &= \frac{\rho_V}{1-\rho_V} \cdot E[D_{s_V}] = 1.18 \cdot 10^{-5}\,\text{s} \end{aligned}$$

   To get the number of packets in the system, we use again Little's theorem,

first calculating the average time spent in the system by a packet:

$$\begin{aligned} E[D] &= E[D_q] + E[D_{s_V}] = 4.92 \cdot 10^{-4}\,\mathrm{s} \\ E[N] &= \lambda_V \cdot E[D] = 50\,\frac{1}{\mathrm{s}} \cdot 4.92 \cdot 10^{-4}\,\mathrm{s} = 0.025 \end{aligned}$$

2. We can no longer use the M/M/1 model from the previous subproblem, since the arrival process is no longer Poisson. Therefore, we use the M/G/1 model from now on. We already know some of the necessary values for the voice traffic. We still miss the second central moment, and the moments of the service time distribution of the data packets to calculate the overall second moment of the service time:

$$\begin{aligned} E[D_{s_D}] &= \frac{10000\,\mathrm{bit}}{1 \cdot 10^6\,\frac{\mathrm{bit}}{\mathrm{s}}} = 0.01\,\mathrm{s} \\ \rho &= \rho_V + \rho_D = \lambda_V \cdot E[D_{s_V}] + \lambda_D \cdot E[D_{s_D}] = 0.224 \\ E[D_s] &= p_V \cdot E[D_{s_V}] + p_D \cdot E[D_{s_D}] \\ &= \frac{\lambda_V}{\lambda_V + \lambda_D} \cdot E[D_{s_V}] + \frac{\lambda_D}{\lambda_V + \lambda_D} \cdot E[D_{s_D}] = 3.2 \cdot 10^{-3}\,\mathrm{s} \end{aligned}$$

Since the packet lengths are exponentially distributed, $CV[L_V] = CV[D_{s_V}] = 1$ and $CV[L_D] = CV[D_{s_D}] = 1$, as well. Therefore:

$$\begin{aligned} E[D_{s_V}^2] &= (1 + CV[D_{s_V}]^2) E[D_{s_V}]^2 = (1 + 1^2) \cdot (4.8 \cdot 10^{-4}\,\mathrm{s})^2 = 4.61 \cdot 10^{-7}\,\mathrm{s}^2 \\ E[D_{s_D}^2] &= (1 + CV[D_{s_D}]^2) E[D_{s_D}]^2 = 2 \cdot 10^{-4}\,\mathrm{s}^2 \\ E[D_s^2] &= p_V \cdot E[D_{s_V}^2] + p_D \cdot E[D_{s_D}^2] = 5.75 \cdot 10^{-5}\,\mathrm{s}^2 \end{aligned}$$

(Side note: this means that the coefficient of variation of the service time distribution is $CV[D_s] = 2.15$. The combination of two exponentially distributed service times does not give another exponential distribution!) Now, we can calculate the average waiting time and thus the system response times for the

two packet classes:

$$\begin{aligned} E[D_q] &= \frac{\lambda \cdot E[D_s^2]}{2(1-\rho)} = 2.6 \cdot 10^{-3}\,\mathrm{s} \\ E[D_V] &= E[D_q] + E[D_{s_V}] = 3.08 \cdot 10^{-3}\,\mathrm{s} \\ E[D_D] &= E[D_q] + E[D_{s_D}] = 12.6 \cdot 10^{-3}\,\mathrm{s} \end{aligned}$$

3. The model changes again slightly, to the M/G/1 model with priorities. For the voice packets, we get

$$E[D_{q_V}] = \frac{\lambda \cdot E[D_s^2]}{2(1-\rho_V)} = 2.1 \cdot 10^{-3}\,\mathrm{s}$$

In contrast, the data packets on average have to wait:

$$E[D_{q_D}] = \frac{\lambda \cdot E[D_s^2]}{2(1-\rho_V)(1-\rho_V-\rho_D)} = 2.7 \cdot 10^{-3}\,\mathrm{s}$$

Therefore,

$$\begin{aligned} E[D_V] &= E[D_{q_V}] + E[D_{s_V}] = 2.6 \cdot 10^{-3}\,\mathrm{s} \\ E[D_D] &= E[D_{q_D}] + E[D_{s_D}] = 12.7 \cdot 10^{-3}\,\mathrm{s} \\ E[N_{q_V}] &= \lambda_V \cdot E[D_{q_V}] = 50\,\frac{1}{\mathrm{s}} \cdot 2.1 \cdot 10^{-3}\,\mathrm{s} = 0.11 \\ E[N_{q_D}] &= \lambda_D \cdot E[D_{q_D}] = 20\,\frac{1}{\mathrm{s}} \cdot 2.7 \cdot 10^{-3}\,\mathrm{s} = 0.05 \\ E[N_V] &= \lambda_V \cdot E[D_V] = 50\,\frac{1}{\mathrm{s}} \cdot 2.6 \cdot 10^{-3}\,\mathrm{s} = 0.13 \\ E[N_D] &= \lambda_D \cdot E[D_D] = 20\,\frac{1}{\mathrm{s}} \cdot 12.7 \cdot 10^{-3}\,\mathrm{s} = 0.25 \end{aligned}$$

Example 2: Signaling, Streaming and Best Effort Traffic

In this example, we introduce a third traffic class into a QoS-supporting network, namely signaling traffic. This can be for example VoIP signaling in the form of SIP, or signaling messages from the traditional telephone network carried over an IP network, e.g., using SIGTRAN. Since these messages are critical for their applications, they should have a very high priority.

In addition, we see streaming traffic that consists not only of voice packets but

video content as well. Again, streaming has tighter delay constraints than other data traffic, so it should receive some priority. Finally, the rest of the traffic, for example Peer-to-Peer (P2P) traffic, is considered background traffic with the lowest priority.

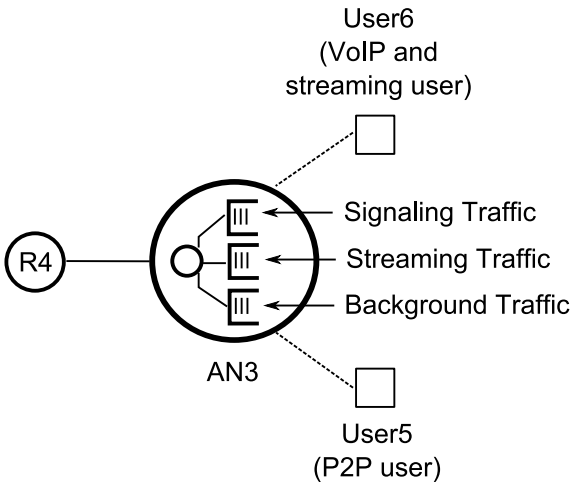


Figure 8.5: An access network with three different traffic classes

We will use the M/G/1 model with priority to analyze a network element carrying these three traffic types, giving priority in the order: signaling, streaming, background traffic.

Signaling packets have an average length of $E[L_{sig}] = 50\,\text{Byte}$, streaming packets of $E[L_{str}] = 1000\,\text{Byte}$ and background data packets of $E[L_{data}] = 1450\,\text{Byte}$. The lengths of the packets are distributed with coefficients of variation $CV[L_{sig}] = 0.5, CV[L_{str}] = 1, CV[L_{data}] = 0.1$, respectively. The link to be modeled has a send rate of $R = 10\,\frac{\text{Mbit}}{\text{s}}$.

The packet arrival process at the link is a Poisson process, with an average packet arrival rate of $\lambda = 500\,\frac{1}{\text{s}}$. Of the arriving packets, $p_{sig} = 5\%$ are signaling packets, $p_{str} = 35\%$ are streaming packets and $p_{data} = 60\%$ background data packets.

We will calculate the average time spent by the different packet types in the buffer of that link.

Solution:

We need the second moment of the service time distribution and the partial system utilizations ρ_i caused by the different traffic classes:

$$
\begin{aligned}
E[D_{s_{sig}}] &= \frac{E[L_{sig}]}{C} = 0.04 \cdot 10^{-3}\,\mathrm{s}, \\
E[D_{s_{str}}] &= 0.8 \cdot 10^{-3}\,\mathrm{s}, E[D_{s_{data}}] = 1.2 \cdot 10^{-3}\,\mathrm{s} \\
\Rightarrow E[D_s] &= p_{sig}E[D_{s_{sig}}] + p_{str}E[D_{s_{str}}] + p_{data}E[D_{s_{data}}] \\
&= 1 \cdot 10^{-3}\,\mathrm{s}
\end{aligned}
$$

$$
\begin{aligned}
E[D^2_{s_{sig}}] &= (1 + CV[D_{s_{sig}}]^2) \cdot E[D_{s_{sig}}]^2 = 2 \cdot 10^{-9}\,\mathrm{s}^2 \\
E[D^2_{s_{str}}] &= 1.28 \cdot 10^{-6}\,\mathrm{s}^2, E[D^2_{s_{data}}] = 1.01 \cdot 10^{-6}\,\mathrm{s}^2 \\
\Rightarrow E[D^2_s] &= p_{sig}E[D^2_{s_{sig}}] + p_{str}E[D^2_{s_{str}}] + p_{data}E[D^2_{s_{data}}] \\
&= 1.05 \cdot 10^{-6}\,\mathrm{s}
\end{aligned}
$$

$$
\begin{aligned}
\rho_{sig} &= p_{sig} \cdot \lambda \cdot E[D_{s_{sig}}] \\
&= 0.001 \\
\rho_{str} &= 0.14, \rho_{data} = 0.36 \\
\Rightarrow \rho &= 0.501
\end{aligned}
$$

With all of this, we can calculate the waiting times according to Formula 3.1:

$$
\begin{aligned}
E[D_{q_{sig}}] &= 0.33 \cdot 10^{-3}\,\mathrm{s} \\
E[D_{q_{str}}] &= 0.38 \cdot 10^{-3}\,\mathrm{s} \\
E[D_{q_{data}}] &= 0.77 \cdot 10^{-3}\,\mathrm{s}
\end{aligned}
$$

We can see the rather strong effect of the scheduling policy on the delay of the background data packets, while the signaling and streaming packets have to wait much

shorter in comparison. Although the signaling packets are short and their number is low, they still have to wait a significant time. This is less due to the queueing than to the probability to encounter one of the longer streaming or data packets being sent over the link when arriving, and having to wait for this transmission to end.

The end!