QoS IN PACKET NETWORKS

Kun I. Park
The MITRE Corporation USA

Springer

QOS IN PACKET NETWORKS

THE KLUWER INTERNATIONAL SERIES IN ENGINEERING AND COMPUTER SCIENCE

QOS IN PACKET NETWORKS

by

Kun I. Park, Ph.D.
The MITRE Corporation USA

Springer

eBook ISBN: 0-387-23390-3
Print ISBN: 0-387-23389-X

Created in the United States of America

Visit Springer's eBookstore at: http://ebooks.kluweronline.com
and the Springer Global Website Online at: http://www.springeronline.com

Dedication

For Meyeon and Kyunja.

Contents

Preface

QoS is an important subject that takes a central place in overall packet network technologies. It is a complex subject and its analysis involves such mathematical disciplines as probability, random variables, stochastic processes, and queuing. These mathematical subjects are abstract and are not easy to grasp for uninitiated persons.

This book is written with two objectives. The first objective is to explain the fundamental mathematical concepts used in QoS analysis in layman's terms and as plainly as possible so that the reader can have a better appreciation of the subject of QoS treated in this book. Second, this book explains in plain language the various parts of QoS in packet networks so that the reader can have a complete view of this complex and dynamic area of communications networking technology.

Kun I. Park
Holmdel, New Jersey

Chapter 1

INTRODUCTION

1. NEED FOR QOS

In recent years, the importance of Quality of Service (QoS) technologies for packet networks has increased rapidly. Today, QoS is undoubtedly one of the central pieces of the overall packet network technologies. How has QoS come to take such an important place in packet networks? This section reviews the recent history of telecommunications network evolution to put this fundamental question underpinning this book in perspective.

Referring to Figure 1.1, in the beginning of telecommunications, there were in general two separate networks, one for voice and one for data. Each network started with a simple goal of transporting a specific type of information. The telephone network, which was introduced with the invention of telephone by Alexander Graham Bell some hundred years ago, was designed to carry voice. The IP network, on the other hand, was designed to carry data.

In the early telephone network, the terminal device was a simple telephone set, which was nothing more than an analog transducer designed to produce an electrical current fluctuating with the speaker's acoustic pressure. For all practical purposes, this was all the function that the terminal device had to perform. The network itself, on the other hand, was more complex than the terminal, and was provided with "intelligence" necessary for providing various types of voice services.

A telephone connection is dedicated to a call during the entire period. Once the call is complete, the circuits are used to set up other calls. The circuits used to set up calls are referred to as trunks as opposed to "loops,"

which are the lines permanently dedicated to individual end users' telephone sets.

In the early telephone network, there were two key measures of service quality. The first was the probability of call blocking, that is, the probability that a call attempt would be blocked because of unavailability of a trunk circuit. Once a call attempt was successful and a connection was established for the call, the next measure of quality was voice quality. Voice quality depended on the transmission quality of the end-to-end connection during a call such as transmission loss, circuit noise, echo, etc.

The original telephone network, therefore, was designed with two main objectives. The first was to make sure that enough trunk circuits were provided to render call blocking probability reasonable, e.g., 1%. The

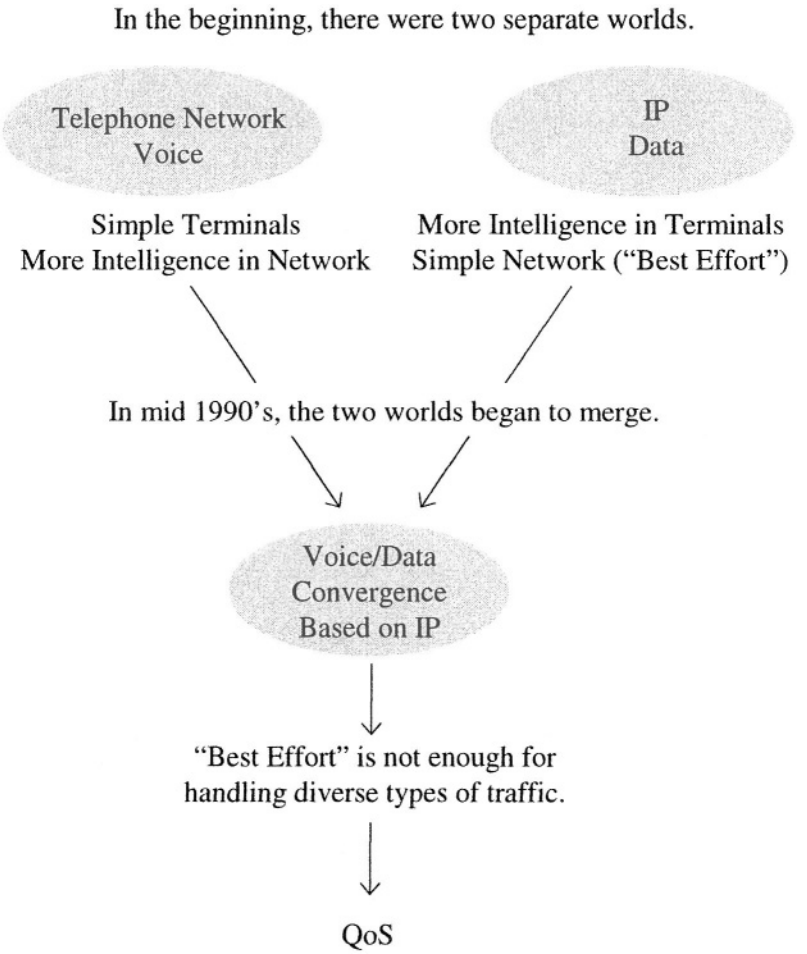


Figure 1-1. Telecommunications network evolution.

second was to design the end to end network with a transmission plan optimized for voice so that the network impairments such as loss, noise, echo, and delay were reasonable. Voice was – and still is – a real time communications service, and there were no queues in the original telephone network to store voice signals for later delivery.

The early IP network was a completely different type of network from the telephone network. First of all, the IP network was designed to carry data. Unlike voice, data was – and still mostly is – a non-real time service. Data could be stored in the network and delivered later. If the data was delivered with error, it could be retransmitted. The data service was sometimes referred to as a "store-and-forward" service.

Since the information carried by the IP network was different from that of the telephone network, the design philosophy used for the IP network was also different from that used for the telephone network.

First, in the original IP network, the network *per se* was designed to be as simple as possible. The main function of the network was to forward packets from one node to the next. All packets were treated the same way and stored in a single buffer and forwarded in a first-in, first-out order.

Second, most of intelligence was placed in the terminal device, which was typically a host computer. For example, if a packet arrived at its destination with error, the receiving terminal would send the sending terminal a negative acknowledgement and the sending terminal would retransmit the packet. The capability of retransmitting lost or errored packets was placed in the terminal, while the network was unaware of the errored packet.

Because the early IP network carried basically one type of information, "store and forward," non-real time data, the network could be designed to operate in the "best effort" mode treating all packets equally, and, as a result, the simple design paradigm described above was possible. The main design objective of the IP network was to make sure that the end user terminal had the appropriate protocols and intelligence to ensure reliable data transmission so that the network could operate as simply as possible.

Although voice and data have distinctly different traffic characteristics and different performance requirements, since the two types of traffic were carried by two separate networks, it was possible to design the networks in the way best suited for the respective payload. In mid 1990's, however, the two separate networks started to merge. A buzz word around this time was "voice and data convergence." The idea was to create a single network to carry both voice and data. Carriers started to plan to consolidate their hodgepodge of separate networks into single "converged" networks for more efficient and economical operation.

At the time, this idea of creating a single converged network for voice and data seemed no more than an engineer's abstract concept. Today, no one can doubt the reality of converged networks for voice and data.

With this convergence, however, a new technical challenge has emerged. In the converged network, the best effort operation of the earlier IP network is no longer good enough to meet diverse performance requirements, often times conflicting, of various types of information carried by the network. QoS is the technology that provides solutions to this technical problem.

2. DEFINITION OF QOS

Figure 1-2 shows an end-to-end network, defines QoS, and the relationships between the various QoS topics treated in this book. The end user represents the terminal devices such as a telephone set, a host computer and other end user communications device. It also represents the human beings who use these terminal devices. The network is a packet network that connects the two end users.

Referring to Figure 1-2, QoS is defined from two points of view: QoS experienced by the end user and the QoS from the point of view of the network. From the end user's perspective, QoS is the end user's perception of the quality that he receives from the network provider for the particular service or application that he subscribes to, e.g., voice, video, and data.

From the network's perspective, the term "QoS" refers to the network's capabilities to provide the QoS perceived by the end user as defined above. Two types of network capabilities are needed to provide QoS in packet networks.

First, to provide QoS, a packet network must be able to differentiate between classes of traffic so that the end users can treat one or more classes of traffic differently than others. Second, once the network differentiates between the traffic classes, it must then be able to treat these classes distinctly by providing resource assurance and service differentiation within the network.

The end user perception of the quality is determined by subjective testing as a function of the network impairments such as delay, jitter, packet loss, and blocking probability. The amount of impairment introduced by a packet network depends on the particular QoS mechanism implemented in the network.

Since a network typically carries a mix of traffic types with different performance requirements, one type of impairment important to a particular service or application may not be as important to other types of service or application and *vice versa.* A QoS mechanism implemented in a network

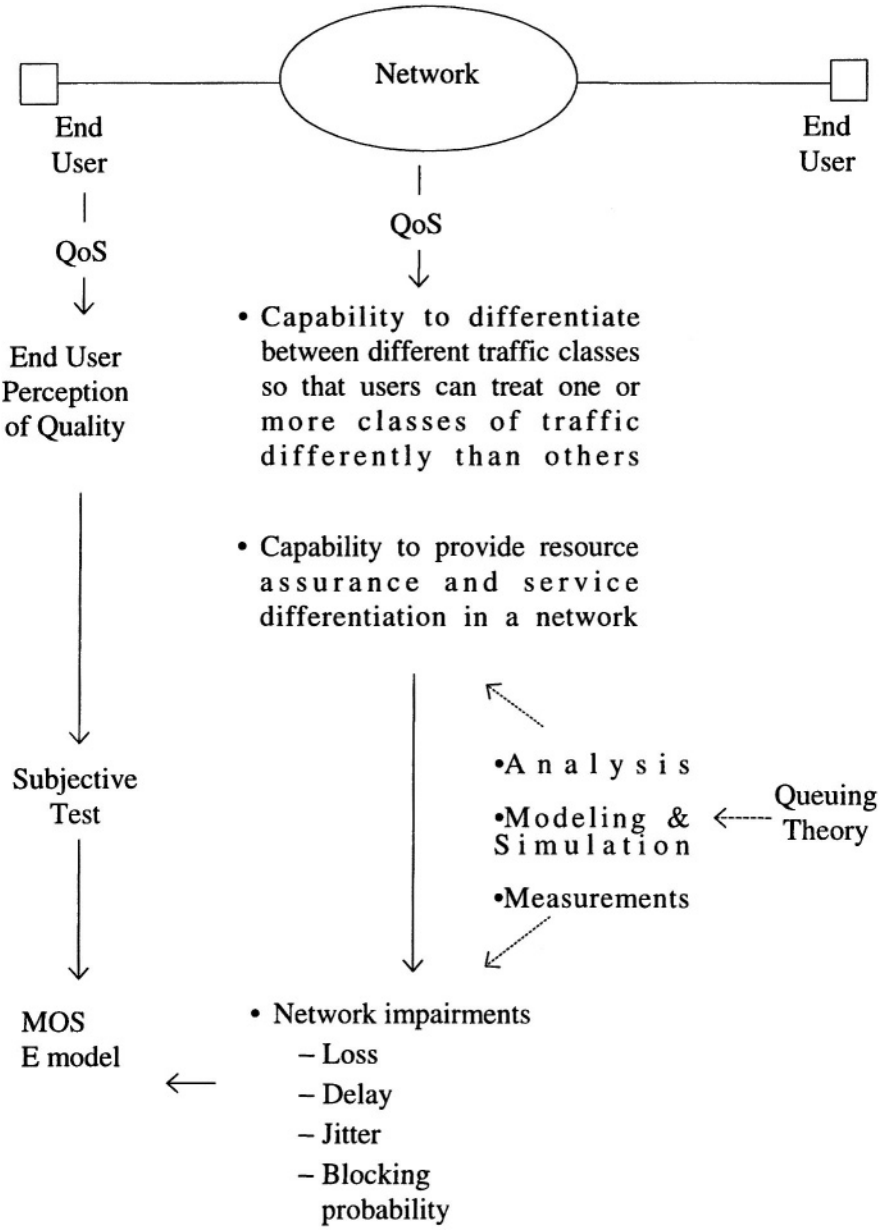


Figure 1-2. Definition of QoS.

must therefore consider various conflicting performance requirements and optimize the trade-off between the impairments.

3. ORGANIZATION OF THE BOOK

Figure 1-2 also serves as a roadmap for this book. As shown in the figure, designing QoS mechanisms for a packet network involves analysis, modeling, simulation, and measurements of network performance. The fundamental mathematical disciplines employed in QoS studies include probability theory, random variables, stochastic processes, and queuing theory. A basic understanding of these mathematical topics, at least at a conceptual level, will help the reader to gain a better appreciation of the QoS topics treated in this book.

The book appropriately begins with a concise treatment of these concepts. The main focus of Chapter 2 is to explain these concepts in plain terms without necessarily involving rigorous mathematics. Throughout the book, application of the mathematics discussed in this chapter will be discussed when appropriate.

Chapter 3 discusses the performance metrics used for QoS from the points of view of the end user and the network. This chapter examines the basic elements of digital communications systems and packet networks and the various types of network impairments generated by the networks. This chapter also discusses subjective testing and the Erlang B and Erlang C models for calculating blocking probability of connection setup attempts.

Chapter 4 and Chapter 5 deal with IP QoS. Chapter 4 explores the generic functional capabilities required in IP networks to provide QoS. It discusses packet marking, packet classification, traffic policing and shaping, traffic metering and coloring, Active Queue Management (AQM), and packet scheduling. Specific topics in this chapter include the single rate three color marker (srTCM) and the two rate three color marker (trTCM); the Random Early Discarding (RED) and the Weighted RED (WRED); the Explicit Congestion Notification (ECN) method of AQM; and various types of packet scheduling including the Priority Queuing (PQ), the Fair Queuing (FQ), the Weighted Fair Queuing (WFQ), and the Class-Based WFQ.

Chapter 5 examines two specific IP QoS mechanisms referred to as the Integrated Services (IntServ) and the Differentiated Services (DiffServ). It discusses briefly the reservation protocol (RSVP) used for IntServ. For DiffServ, the DiffServ Code Points (DSCP's), the Per Hop Behavior, the Expedited Forwarding (EF), and the Assured Forwarding PHB are discussed.

Chapter 6 explains QoS in the Asynchronous Transfer Mode (ATM) network. It discusses various types of ATM virtual connections such as the Virtual Path Connection (VPC) and the Virtual Channel Connection (VCC), ATM service classes such as the Constant Bit Rate (CBR) and the variable Bit Rate (VBR) services, and Connection Admission Control (CAC) methods.

Finally, Chapter 7 discusses Multi-Protocol Label Switching (MPLS). The discussion includes the architecture, implementation and operation of MPLS as well as how MPLS and DiffServ can be used together.

Chapter 2

BASIC MATHEMATICS FOR QOS

To understand QoS in packet networks, it is important to understand not only the mechanism of providing QoS but also the performance behavior that is produced by the QoS mechanism. This chapter reviews some of the basic mathematics that is needed in the analysis of QoS performance in packet networks. The following topics are reviewed in this chapter:

- probability
- random variables
- stochastic processes
- queuing theory

From the author's experience of teaching, students generally considered the mathematical concepts and disciplines such as probability theory, random variables and stochastic processes to be too abstract and hard to apply to real problems.[1-3] One of the purposes of this chapter is to explain the abstract concepts in layman's terms as much as possible so that they can be applied to real problems such as QoS.

1. PROBABILITY THEORY

1.1 Random experiments, outcomes and events

A random experiment is an experiment that produces random outcomes. For example, throwing a die is a random experiment in which each trial produces a random outcome from six possible outcomes, i.e., faces with one through six spots. The word "experiment" implies that the random situation

under consideration is controlled. However, the word may also be used in a broad sense to mean any random situation that produces random outcomes, let us say, a nature's experiment.

A trial is a single instantiation of a random experiment. If a die is thrown ten times, there would be ten trials. The key concept to note here is that each trial produces exactly one outcome.

Another term frequently used in probability is a random "event." A random event is a higher level outcome that may depend on multiple experiments and multiple outcomes of the experiments. For example, consider a game consisting of two random experiments, "throwing a die" and "throwing a coin." A player is to throw the die twice and the coin once. A player who gets the face with one spot in both die-throwings and a "head" in the coin-throwing wins the grand prize. In this game, the random "event" of interest is "winning the grand prize." This event would "occur," if the trials produce the following outcomes: one spot in both of the die-throwings and a "head" in the coin-throwing. In this example, the event depends on multiple experiments and multiple outcomes.

In set theory, a set is defined by the elements contained in the set, e.g., a set of all integers, a set of all even integers, and a set of positive numbers. Using set theory, an event is defined as a set containing the outcomes that make the event happen. For example, in the die-throwing experiment, an event called "face with an even number of spots" may be defined by a set denoted by say E as follows: E= *{"two," "four," "six"},* where *"two" "four"* and *"six"* denote the number of spots on the face of the die.

A random event defined by a set containing a single outcome is referred to as an "elementary event." For example, in the die throwing example, there are six possible random outcomes: *"one," "two," "three," "four," "five,"* and *"six"*. If each of these possible outcomes is defined to be an event, the six possible outcomes produce six elementary events: *{"one"}, {"two"}, {"three"}, {"four"}, {"five"},* and *{"six"}.*

The distinction between the outcome, e.g., *"one,"* and the event, e.g., *{"one"},* is significant and fundamental in the construct of probability theory because, as we shall see in Section 1.3, probability is defined for an event given the probabilities of the underlying random outcomes. *"One"* is an element of a set, whereas *{"one"}* is a set containing one element, *"one."* The probabilities of elementary events would then be equal to the probabilities of the random outcomes.

1.2 Definition of probability

What is probability? Mathematicians attempted to define this seemingly simple term without much success in reaching a consensus for a long time

until Kolmogorov presented his celebrated theory referred to as the "axiomatic approach." The power of the axiomatic approach is in its simplicity.

First, consider the debate that went on before Kolmogorov. A probability was defined as a frequency of occurrence. Consider 1,000 trials in the coin throwing experiment. If the head shows up 400 times, it is concluded that the "probability" of a head is 0.4. The dilemma of this definition of probability is that unless the coin is thrown many times and the outcomes are observed, there is no way of telling the probability.

Some would say that the probability of head should be 0.5 but then others would argue that, unless the coin is minted "perfectly" with identical sides, no one can say that its probability is 0.5 even though it may be "close," etc., etc. Mathematicians had difficulty overcoming the arguments such as this and, as a result, probability theory could not be developed into a useful discipline that could be applied to practical problems.

Most reasonable persons could agree, deep in their hearts, that it should be good enough to take the probability of, for example, a particular face in die throwing is 1/6 and move on to solve other probability problems associated with die throwing. If the 1/6 probability for a face is accepted, then one can find, for example, the probability of a face with an even number of spots, which would be 0.5, etc. With the frequency definition of probability, this simple solution would not be possible. Such an approach is possible because human beings are given this innate capability of *a priori* reasoning.

Kolmogorov presented this simple idea based on *a priori* reasoning that freed everyone interested in probability from the endless arguments. His approach is referred to as the "axiomatic probability theory" and is based on set theory and measure theory. His idea was that there was no need to determine whether a coin was minted perfectly to discuss its probability. He simply turned the table around and asserted that one could "assign" probabilities to the outcomes based on the *a priori* knowledge of the outcomes and let the probabilities initially assigned be the starting point for developing more complex probability theory just like accepting 1/6 as the probability of a face in die throwing.

The key concept is in the word "assign." In this approach, probability "begins" with the assignment of it based on one's own judgment about the likelihood of the outcome. In the axiomatic approach, one can start with "assigning" 1/6 each as the probability of a face in the die-throwing experiment. Once this initial assignment of probability is "accepted" (as an axiom, so to speak), it is now possible to solve all kinds of complex and interesting probability problems associated with die-throwing.

For example, what is the probability of getting an even number of spots? Since the 1/6 probability is "accepted," one can proceed to find its answer, which is 0.5. What is the probability of getting a face with more than four spots? Since either five or six spots would make this event happen, the answer would be 2/6.

1.3 Axiomatic approach to probability

A mathematical system, e.g., linear algebra, set theory, and group theory, is simply an artifact that is useful because it provides a structure for drawing meaningful inferences. The axiomatic probability theory is such a mathematical system.

Consider a random experiment with n possible outcomes, $\xi_1, \xi_2, \ldots, \xi_n$. The probability space S is defined as the set of all possible random outcomes of a random experiment as follows:

$$S = \{ \xi_1, \xi_2, \ldots\ldots, \xi_n \} \tag{2-1}$$

A "measure" is "assigned" to each outcome, ξ_i. This measure is referred to as "probability." Denote this measure by p_i. The measure chosen is a real number between 0 and 1 as follows:

$$0 \le p_i \le 1 \tag{2-2}$$

$$p_i = P(\xi_i) = \textit{probability of random outcome } \xi_i \tag{2-3}$$

The word "probability" was difficult to define because of the attempts to define its meaning semantically and in some instances philosophically. In the axiomatic probability theory, its definition is simply a "measure" that is assigned to an outcome. In fact, this measure does not have to be a number between 0 and 1. It can be a number between 0 and 100 or any number for that matter without changing the axiomatic theory. It is conventional though to use a number between 0 and 1 as a probability measure.

An axiom is a statement accepted as a truth or a rule as a basis of inference. Given the probability space S of (2-1) and the probability measures of the random outcomes of the experiment of (2-3), the axiomatic probability theory is based on the following three simple axioms:

$$\textit{Axiom I} \quad P(A) \ge 0 \tag{2-4}$$

$$\textit{Axiom II} \quad P(S) = 1 \tag{2-5}$$

$$\textit{Axiom III} \quad \textit{If } A \cap B = \{\phi\}, P(A \cup B) = P(A) + P(B) \tag{2-6}$$

In the above equations, S is a set referred to as the probability space defined earlier. A and B are subsets of S and define the random events of interest. Since A and B define the events, they are sometimes simply referred to as "events." S is also a set and, as such, also an event. Since S includes all possible outcomes, any outcome will make S happen and so S is referred to as a certain event. Similarly, $\{\phi\}$ is a set that contains no element. No outcome will make $\{\phi\}$ happen, and $\{\phi\}$ is referred to as an impossible event. Two set operations are used in these axioms. $A \cap B$ is an intersection of A and B, a set of elements belonging to both A and B. $A \cup B$ is a union of A and B, a set of elements belonging to either A or B.

Axiom I states that any event defined in the probability space is assigned a non-negative measure or probability. This is simply an agreement to start the theory. It is entirely possible in the axiomatic theory to use negative numbers for probability as long as that is agreed to at the beginning of the framework because probability is simply nothing more than a numerical measure in the axiomatic theory. However, it would be cumbersome to think in negative numbers when one considers probability.

Axiom I defines the starting point of development of a probabilistic framework of a random experiment under consideration. First, define the elementary events $\{\xi_i\}$ and assign probabilities to them, $P(\{\xi_i\})$. Note the distinction between $P(\{\xi_i\})$ and $P(\xi_i)$. The former is the probability of the elementary event $\{\xi_i\}$ and the latter, that of a random outcome ξ_i. It is important to note that the starting point of the axiomatic framework, i.e., Axiom I, is $P(\{\xi_i\})$ and not $P(\xi_i)$.

Axiom II states that the probability of the space S is one. The space S is a set that contains all possible outcomes under consideration and it would be reasonable to accept as a basic truth that the probability of all possible outcomes is one.

In effect, Axiom II simply states that the probability of certainty is one. One may then ask what about the probability of impossibility, i.e., a null event. Don't we need an axiom, say Axiom IIa that states $P(\{\phi\}) = 0$? It can be shown that the three axioms cover this axiom and adding it would be superfluous because it can be derived from Axioms II and III as follows.

From set theory, the union of the space S and the null set $\{\phi\}$ is the space S and the intersection of the space S and the null set $\{\phi\}$ is the null set $\{\phi\}$:

$$S \cup \{\phi\} = S \tag{2-7}$$

$$S \cap \{\phi\} = \{\phi\} \tag{2-8}$$

From Equation (2-7), it follows that:

$$P(S) = P(S \cup \{\phi\}) \tag{2-9}$$

Equation (2-8) satisfies the condition for Axiom III. Hence, from Axiom III and Equation (2-9), it follows that:

$$P(S) = P(S \cup \{\phi\}) = P(S) + P(\{\phi\}) \tag{2-10}$$

From Axiom II and Equation (2-10), it follows that:

$$P(S) = P(S \cup \{\phi\}) = P(S) + P(\{\phi\}) = 1 \tag{2-11}$$

Finally, from Equation (2-11), it follows that:

$$P(\{\phi\}) = 1 - P(S) = 0 \tag{2-12}$$

Note that Axiom I states $P(A) \geq 0$ but it does not include $P(A) \leq 1$. Once again, the reason is because it can be derived from other axioms and including $P(A) \leq 1$ would be superfluous.

Example 1

A box contains a total of 10 balls of different colors as follows: two white balls, three red balls and five black balls. A player is to withdraw a ball, and, if the ball withdrawn is either red or black, the player wins a piece of candy. What is the probability of winning a piece of candy by playing this game?

Solution

There are eight red or black balls out of a total of 10 balls, and so the probability of winning the grand prize is 0.8. This is a simple problem and one can get the answer quickly in the head without going through the rigor of axiomatic formulation.

However, we shall formulate and solve this problem using the axiomatic approach to illustrate how a probability problem can be formulated and solved systematically. For more complex problems, the disciplined way of dealing with the problem using the axiomatic approach is helpful.

First define the random experiment. There are two alternative ways of defining the space and random outcomes for this problem. Either method should yield the same answer.

Formulation 1. A more direct way of formulation is to define the outcomes of ball drawing like the outcomes of die throwing. Imagine that the individual balls can be distinguished (e.g., by numbering them) as the faces of a die are distinguished. Then there are ten possible outcomes with an equal probability as follows:

$$S = \{ \xi_1, \xi_2, \xi_3, \xi_4, \xi_5, \xi_6, \xi_7, \xi_8, \xi_9, \xi_{10} \} \tag{2-13}$$

$$p_i = P(\xi_i) = 1/10; \qquad i = 1, \ldots, 10 \tag{2-14}$$

where ξ_1 and ξ_2 are drawing a white ball, ξ_3, ξ_4 and ξ_5, a red ball and ξ_6 through ξ_{10}, a black ball.

The next step is to define the event. The event of interest is "winning a candy" and is defined as a set denoted by *W*. In set theory, a set is defined by its members or a member is "qualified" to be included in the event set, if it makes that event happen. *W* in turn depends on the following two events:

$$R = \text{"ball withdrawn is red"} = \{ \xi_3, \xi_4, \xi_5 \} \tag{2-15}$$

$$B = \text{"ball withdrawn is black"} = \{ \xi_6, \xi_7, \xi_8, \xi_9, \xi_{10} \} \tag{2-16}$$

Since $\{\xi_i\}$'s are mutually exclusive, i.e., $\{\xi_i\} \cap \{\xi_j\} = \{\phi\}$ *for i, j = 3 - 8,* it follows that:

$$R = \{ \xi_3, \xi_4, \xi_5 \} = \{ \xi_3 \} \cup \{ \xi_4 \} \cup \{ \xi_5 \}$$

$$= [\{ \xi_3 \} \cup \{ \xi_4 \}] \cup \{ \xi_5 \} \tag{2-17}$$

Applying Axiom III twice, it follows that:

$$P(R) = P(\{ \xi_3, \xi_4, \xi_5 \}) = P([\{ \xi_3 \} \cup \{ \xi_4 \}] + P(\{ \xi_5 \})$$

$$= P(\{\xi_3\}) + P(\{\xi_4\}) + P(\{\xi_5\}) = 0.3 \tag{2-18}$$

Similarly,

$$P(B) = P(\{\{\xi_6, \xi_7, \xi_8, \xi_9, \xi_{10}\}\}) = 0.5 \tag{2-19}$$

W would occur if the ball withdrawn is either red or black: W would occur if either R or B occurs. Since R and B are mutually exclusive events, it follows that:

$$R \cap B = \{\phi\} \tag{2-20}$$

$$W = R \cup B \tag{2-21}$$

Hence, from Axiom III, it follows that:

$$P\{W\} = P\{R \cup B\} = P\{R\} + P\{B\} = 0.3 + 0.5 = 0.8 \tag{2-22}$$

Formulation 2. As long as the axiomatic approach is followed, different definitions of outcomes are possible. The above formulation can be simplified by defining the experimental outcomes as the colors of the balls as follows:

$$S = \{\xi_w, \xi_r, \xi_b\} \tag{2-23}$$

where ξ_w, ξ_r and ξ_b are random outcomes of white, red and black color.

Then from the problem, the probabilities of the random outcomes can be assigned as follows:

$$P(\xi_w) = 0.2; \quad P(\xi_r) = 0.3; \quad P(\xi_b) = 0.5 \tag{2-24}$$

W would occur if ξ_r or ξ_b shows up. Hence,

$$W = \{\xi_r, \xi_b\} = \{\xi_r\} \cup \{\xi_b\} \tag{2-25}$$

Since $\{\xi_r\} \cap \{\xi_r\} = \{\phi\}$, from Axiom III and Equations (2-24) and (2-25), it follows that:

$$P\{W\} = P(\{\xi_r, \xi_b\}) = P(\{\xi_r\} \cup \{\xi_b\}) = P(\{\xi_r\}) + P(\{\xi_b\})$$

$$= 0.3 + 0.5 = 0.8 \qquad (2\text{-}26)$$

2. RANDOM VARIABLES

2.1 Definition

It is conventional to denote a random variable by a bold letter and a deterministic variable or a fixed value by a regular letter. For example, a random variable may be denoted by $\boldsymbol{x}$ and a fixed value that $\boldsymbol{x}$ can take, by x.

A random variable (RV) $\boldsymbol{x}$ is a function of a random outcome $\boldsymbol{\xi}$ of a random experiment that maps a random outcome to a real value: $\boldsymbol{x}(\boldsymbol{\xi})$. As discussed in Section 1, random outcomes could be any objects. It can be the faces of a die in die throwing, the colors of the balls in the ball drawing, etc. Random outcomes could also be real numbers, discrete or continuous. A number can just be an object of a set. A term "real line" is used to denote the set of all real numbers, i.e., the continuum, from $-\infty$ to $+\infty$. Since the real line is a continuum, discrete points are also included in the set. Figure 2-1 illustrates the mapping from $\boldsymbol{\xi}$ to x on the real line.

In the die throwing experiment, the space is a set of six possible outcomes:

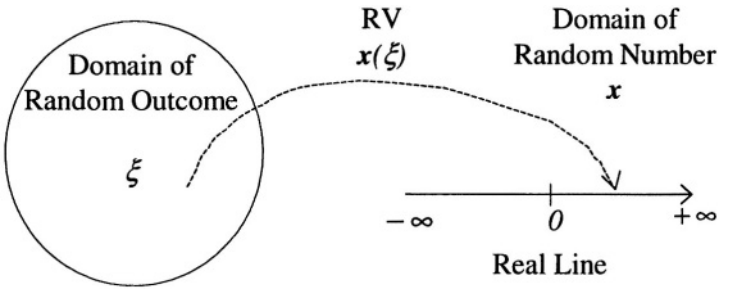


Figure 2-1. Mapping from $\boldsymbol{\xi}$ to x on the real line by $\boldsymbol{x}(x)$.

$$S = \{\xi_1, \xi_2, \xi_3, \xi_4, \xi_5, \xi_6\} \tag{2-27}$$

These six outcomes are not numbers; they are simply objects that constitute the set *S*. An RV is a function that relates these objects to real numbers. An RV must first be defined just as a function must be defined. Let us now define a random variable $\boldsymbol{x}$ that maps the six objects of *S* to a set of real numbers, i.e., onto the real line. To illustrate the concept, suppose that a player is paid \$1 to \$6 depending on the number of spots on the face as follows:

Random Outcome, ξ	Payoff $\boldsymbol{x}(\xi)$
One spot	\$1
Two spots	\$2
Three spots	\$3
Four spots	\$4
Five spots	\$5
Six spots	\$6

In this example, the RV $\boldsymbol{x}(\xi)$ maps the six outcomes to real numbers representing payoffs. Since, in this case, the random outcomes are (non-numerical) objects, there is no convenient way of expressing the functional relationship $\boldsymbol{x}(\xi)$. The best way of "defining" the RV is by a table such as the one above.

Having introduced this basic concept of RV, we now extend the concept to a little more abstract situation. Suppose now that the space of random outcomes *S* is itself the real line:

$$S = \{x \mid x \varepsilon (-\infty, +\infty)\} \tag{2-28}$$

In set theory, the above expression is read as follows: "*S* is a set of *x,* where *x is* a member (as denoted by $\boldsymbol{\varepsilon}$) of an interval of real numbers from $-\infty$ to $+\infty$." It can also be specified that *x* is an integer. In that case, *S* is a set of all integers from $-\infty$ to $+\infty$." An RV $\boldsymbol{x}$ can now be defined as a function on *S* that maps *x of S* to *x*, $\boldsymbol{x}(x) = x$. This is illustrated in Figure 2-2.

It could be less confusing, if the real numbers of *S* were denoted by a different symbol such as *y*; however, this would be even more confusing because then *y* and *x* can take on different values. For now, consider $\boldsymbol{x}(x) = x$ to read as follows: "RV $\boldsymbol{x}$ maps *x* of *S* to itself *x."* In most situations of random variables that we are familiar with, this is the definition tacitly used.

For example, when we say that the temperature in a certain area is a random variable $\boldsymbol{x}$, we cannot possibly mean that the random temperature is a result of multitudes of random outcomes of the nature. It may be possible

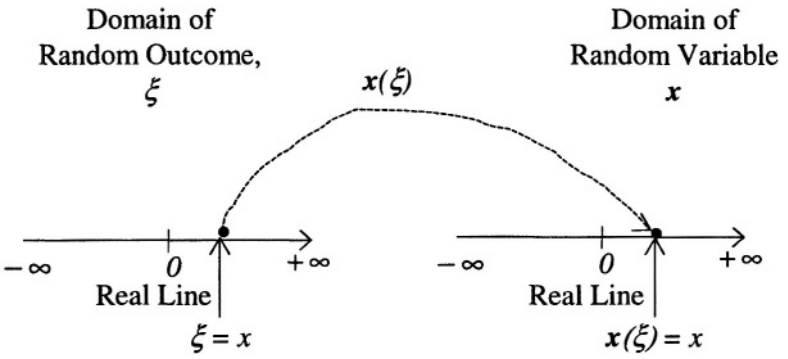


Figure 2-2. Mapping from $\xi=x$ on real line to x on real line by $x(\xi)$.

to do so in certain circumstances. In most cases, however, the way we interpret the random temperature x is as follows. We measure the temperature, i.e., perform a "trial," and take its reading as a random outcome. We then take this random outcome as the value of the random variable, i.e., $x(x) = x$.

Suppose now that the domain of random outcomes ξ is the real line, a continuum. A continuous random variable x is a function of random outcome ξ that maps the specific value x of the random outcome ξ from the continuum of the real line $(-\infty, +\infty)$ (i.e., $\xi = x$) to itself x. Figure 2-2 illustrates this definition: the continuous random variable x maps random outcomes, i.e., $\xi = x$, on the real line to the same value x on the real line, i.e., mapping from x to x by $x(\xi)$.

Finally, an RV may be defined to map multiple outcomes to a single number, i.e., many to one; however, an RV cannot map a single outcome to multiple numbers.

2.2 CDF and pdf

Let x be a random variable (RV). Its cumulative distribution function (CDF) is defined as follows:

$$F_x(x) = P\{x \le x\} \tag{2-29}$$

$P\{x \le x\}$ reads: "the probability that the RV x will be less than a value x."

The probability density function (pdf) of x is defined as follows:

$$f_x(x) = \frac{dF_x(x)}{dx} \tag{2-30}$$

From the above definition, $F(x)$ can also be given by the following integral:

$$F(x) = \int_{-\infty}^{x} f(z)dz \tag{2-31}$$

Conceptually, it is easier to interpret the pdf in the following way. Consider the probability that the random variable x will lie in the small interval between x and $x+\Delta x$. From the definition of the CDF $F(x)$, this probability is obtained as follows:

$$P\{x \prec x \leq x + \Delta x\} = P\{x \leq x + \Delta x\} - P\{x \leq x\}$$

$$= F(x + \Delta x) - F(x) = \int_{-\infty}^{x+\Delta x} f(z)dz - \int_{-\infty}^{x} f(z)dz = \int_{x}^{x+\Delta x} f(z)dz \approx f(x)\Delta x \tag{2-32}$$

From the above, we have:

$$f(x) \approx \frac{P\{x \prec x \leq x + \Delta x\}}{\Delta x} \tag{2-33}$$

Taking the limit, we have:

$$f(x) = \lim_{\Delta x \to 0} \frac{P\{x \prec x \leq x + \Delta x\}}{\Delta x} \tag{2-34}$$

From the above, we see that the pdf $f(x)$ is the probability that x will lie in a small interval of length Δx divided by the interval length Δx as Δx becomes infinitesimally small. This is illustrated in Figure 2-3.

The word "density" refers to the fact that the small probability $P\{x < x \leq x + \Delta x\}$ is normalized by the interval length Δx.

For a discrete random variable x, the pdf is given by:

$$f(x) = \sum_i P\{x = x_i\}\delta(x - x_i) = \sum_i p_i \delta(x - x_i) \tag{2-35}$$

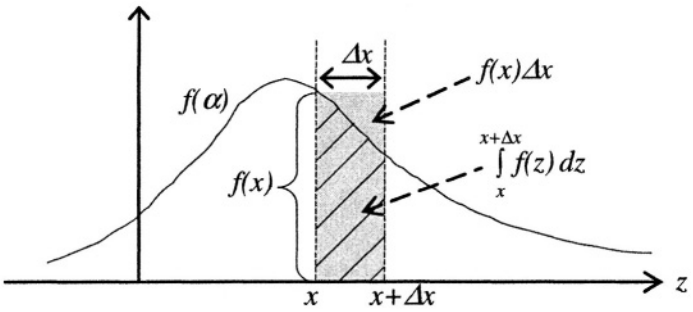


Figure 2-3. Definition of the probability density function (pdf).

$$\text{where } p_i = P\{ x = x_i \} \quad i \varepsilon \{integers\} \tag{2-36}$$

$$\int_a^b \delta(x - x_i)dx = 1 \quad if\ a \prec x_i \leq b$$

$$= 0 \quad if\ b \prec x_i\ or\ x_i \leq a \tag{2-37}$$

The impulse function, $\delta(x)$, as defined above, has the following property. It produces a value when it is integrated, and, without the integration, $\delta(x)$ is undefined. If the integration interval $a \sim b$ includes x_i, the integration of $\delta(x - x_i)$ over this interval is *1*; if x_i lies outside of the integration interval, the integration of $\delta(x - x_i)$ over the interval is zero.

The impulse function is a mathematical artifact convenient for expressing mathematically the pdf of a discrete random variable x, as given by Equation (2-35). For the pdf $f(x)$ of a discrete random variable x defined in terms of the impulse function $\delta(x)$, it is possible to express the CDF $F(x)$ as the integral of $f(x)$ as follows:

$$F(x) = \int_{-\infty}^{x} f(z)dz = \int_{-\infty}^{x} \sum_i p_i \delta(z - x_i)dz$$

$$= \sum_i \{ \int_{-\infty}^{x} p_i \delta(z - x_i)dz \} = \sum_{i=i_{min}}^{i=i_{max}} p_i \tag{2-38}$$

where $p_{i_{min}} = P\{x = x_{i_{min}}\}$ (2-39)

$x_{i_{min}}$ = *smallest discrete value that* x *can take, which is* $\leq x$ (2-40)

$p_{i_{min}} = P\{x = x_{i_{max}}\}$ (2-41)

$x_{i_{max}}$ = *largest discrete value that* x *can take, which is* $\leq x$ (2-42)

2.3 Mean and variance

Consider the ball drawing game of Example 1 discussed earlier. Define an RV x as the payoffs of the game as follows: \$10 if a white ball is drawn, \$20 for a red ball, and \$30 for a black ball. What is the amount of money a player can expect to win by playing this game?

To answer this question, the probabilities of drawing the three colors need to be determined as follows:

$$P\{white\} = \frac{2}{10} = 0.2;\ P\{red\} = \frac{3}{10} = 0.3;\ P\{black\} = \frac{5}{10} = 0.5 \quad (2\text{-}43)$$

The expected amount of payoff is calculated by:

$$E\{x\} = (\$10 \times 0.2) + (\$20 \times 0.3) + (\$30 \times 0.5) = \$23 \quad (2\text{-}44)$$

The expected value is also referred to as two other common terms, "mean" and "average." The term average is used because if the player plays the game long enough performing many "trials," then the average winning, which is determined by dividing the total amount of money won by the number of trials, should approach the expected value, i.e.,

$$\frac{x_1 + x_2 + \ldots + x_N}{N} \rightarrow \$23 \text{ as } N \rightarrow \infty \quad (2\text{-}45)$$

where N is the number of times of playing, and x_i is the i^{th} payoff. In general, the expected value of a discrete random variable x taking on the values of x_i with the probability p_i, $i = 1, 2, \ldots, N$ is:

$$E\{x\} = \sum_{i=1}^{N} x_i p_i \tag{2-46}$$

$$\text{where } p_i = P\{x = x_i\} \quad i = 1, 2, \ldots, N \tag{2-47}$$

To extend the above concept to a continuous random variable x as defined in Figure 2-2, imagine a similar game in which a player takes a measurement from the real line $(-\infty, +\infty)$ and receives a payoff equal to the measurement: payoff x, is x, i.e., $x(\xi = x) = x$. Now consider a small interval of width Δx from x to $x+\Delta x$ on the real line of x domain and a random payoff x falling in this interval as shown in Figure 2-4. The value of $x(\xi)$ in this interval is somewhere between x and $x+\Delta x$, i.e., $x \le x(\xi) \le x+\Delta x$, and so is approximately equal to x, if Δx is small enough. In fact, Δx can be made as small as necessary to make the value of $x(\xi)$ as close to x as possible.

The expected value of the payoff for this small interval is therefore approximately equal to x times the probability that x will fall in this interval as follows:

$$E\{\text{payoff of } x \text{ falling in } (x, x+\Delta x)\} \approx xP\{x \prec x \le x + \Delta x\} \tag{2-48}$$

$$\approx xf(x)\Delta x. \tag{2-49}$$

By taking the limit,

$$E\{\text{payoff of } x \text{ falling in } (x, x+\Delta x)\} = \lim_{\Delta x \to 0} xf(x)\Delta x \tag{2-50}$$

Hence,

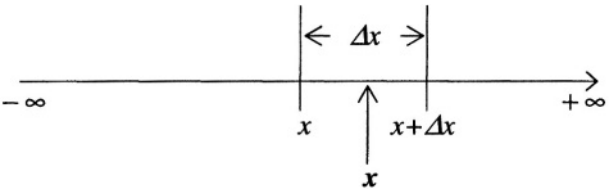


Figure 2-4. Random variable x falling in Dx.

$$E\{x\} = \lim_{\Delta x \to 0} \sum xf(x)\Delta x = \int_{-\infty}^{+\infty} xf_x(x)dx = \eta_x \tag{2-51}$$

The variance of a random variable x is a measure of the variability of x around its mean, η_x. It is the expected value of the square of the difference between the random variable x and its mean η_x as follows:

$$\sigma_x^2 = E\{(x - \eta_x)^2\} = \int_{-\infty}^{+\infty} (x - \eta_x)^2 f_x(x)dx \tag{2-52}$$

The difference is squared because the magnitude of the variation rather than its direction is of primary interest. From the above, the following equation is derived:

$$\sigma_x^2 = E\{x^2\} - \eta_x^2 \tag{2-53}$$

The square root of the variance is the standard deviation:

$$\sigma_x = \sqrt{\sigma_x^2} \tag{2-54}$$

2.4 The normal distribution

Two of the most widely used and important distributions are the normal or Gaussian distribution and the Poisson distribution. The normal random variable x is a continuous random variable with the following pdf:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\eta)^2/2\sigma^2} \tag{2-55}$$

where η is the mean of x and σ is the standard deviation of x.

The CDF of a normal random variable x is the integral of $f(x)$ as follows:

$$F(x) = \int_{-\infty}^{x} f(z)dz = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} e^{-(\alpha-\eta)^2/2\sigma^2} dz \tag{2-56}$$

The normal CDF given by the above integral is tabulated in mathematical tables. It can be shown that the mean and variance of the normal random variable x with the above pdf is η and σ^2. It is significant that, if an RV x is

normal, its pdf can be completely determined by two parameters, mean and variance.

2.5 The Poisson distribution

A Poisson random variable x is a discrete random variable with the following pdf:

$$f(x) = \sum_{k=0}^{\infty} p_k \delta(x - k) \tag{2-57}$$

$$\text{where } p_k = P\{ x = k \} = e^{-\lambda} \frac{\lambda^k}{k!} \tag{2-58}$$

In Equation (2-58), k is an integer taking on a value from *0* to infinity. Putting Equations (2-57) and (2-58) together, the Poisson pdf is given by:

$$f(x) = \sum_{k=0}^{\infty} e^{-\lambda} \frac{\lambda^k}{k!} \delta(x - k) = e^{-\lambda} \sum_{k=0}^{\infty} \frac{\lambda^k}{k!} \delta(x - k) \tag{2-59}$$

The mean and variance of the Poisson random variable x with the parameter λ are both found to be λ as follows:

$$\eta = E\{ x \} = \lambda \qquad \sigma^2 = E\{ x^2 \} - \eta^2 = \lambda \tag{2-60}$$

The Poisson pdf is defined by a single parameter λ. It is significant that, if an RV x is Poisson, its pdf can be completely determined by a single parameter, λ. More will be discussed on the Poisson pdf and the parameter λ later in this chapter.

3. STOCHASTIC PROCESSES

3.1 Definition of a stochastic process

A random variable x is a static variable defined on random outcomes, "static" in the sense that time is fixed for an RV: an RV is a function of random outcome ξ, $x(\xi)$, but time is not an argument of this function.

A stochastic process $x(t)$ is the random variable x extended into another dimension t. To be rigorous in notation, we might write a stochastic process as $x(\xi, t)$, where x is a function of two variables, time t and random outcome ξ. For simplicity, however, $x(\xi, t)$ is written $x(t)$ as the random variable $x(\xi)$ is written x.

To define it in more general terms, a stochastic process is a set of random variables arranged in time as follows:

$$x(t) = \{ x(t) \mid t \varepsilon (-\infty, +\infty) \} \tag{2-61}$$

$$x(t) = \{ x(t) \mid t \varepsilon (t_1, t_2, ..., t_N) \} \tag{2-62}$$

$$x(t) = \{ x(t) \mid t \varepsilon (t_a, t_b) \} \tag{2-63}$$

If the interval (t_a, t_b) is a continuum of time, the stochastic process is referred to as a continuous process; if the interval (t_a, t_b) is a set of discrete time points, t_i's, the stochastic process is referred to as a discrete process.

3.2 CDF and pdf of stochastic process

A useful concept to remember is that, once the time t is fixed at a specific value, say t^*, the stochastic process yields a random variable; that is, $x(t^*)$ is a random variable. Consider a stochastic process $x(t)$. Let us fix the time t to t^* and consider the stochastic process at the instant t^*. At time t^*, $x(t^*)$ is a random variable. Consider the CDF defined earlier for this random variable $x(t^*)$:

$$F(x) = P\{ x(t^*) \leq x \} \tag{2-64}$$

Now let us take a leap and fix time t to an arbitrary value, say t, and write the above equations as follows:

$$F(x) = P\{ x(t) \leq x \} \tag{2-65}$$

Once we fix time t to an arbitrary value t, the first-order CDF of $x(t)$ is a function of time t as follows:

$$F(x,t) = P\{ x(t) \leq x \} \tag{2-66}$$

The first-order refers to the fact that one random variable is considered, i.e., random variable defined at one time point. The first order pdf of $x(t)$ is given by:

$$f(x,t) = \frac{\partial F(x,t)}{\partial x} \tag{2-67}$$

The mean of $x(t)$ is

$$\eta(t) = E\{x(t)\} = \int_{-\infty}^{+\infty} x f(x,t) dx \tag{2-68}$$

Now consider two time points t_1 and t_2 and the two random variables defined for these time points, $x(t_1)$ and $x(t_2)$. The statistics considered for these two random variables is referred to as the second-order statistics, and the joint CDF and joint pdf for $x(t_1)$ and $x(t_2)$ are as follows:

$$F(x_1, x_2; t_1, t_2) = P\{x(t_1) \le x_1, x(t_2) \le x_2\} \tag{2-69}$$

$$f(x_1, x_2; t_1, t_2) = \frac{\partial^2 F(x_1, x_2; t_1, t_2)}{\partial x_1 \partial x_2} \tag{2-70}$$

3.3 Autocorrelation and cross-correlation

An important concept in stochastic processes is the autocorrelation function. It is a measure of the correlation between two random variables defined at two time points for the same stochastic process. The pre-fix "auto" signifies that the correlation is considered for the same process. Later, the cross-correlation function defines the same between two different processes.

For a real process $x(t)$, consider the two real random variables defined for two time points, $x(t_1)$ and $x(t_2)$. The autocorrelation function of $x(t)$, denoted by $R(t_1, t_2)$, is defined as the expected value of the product of $x(t_1)$ and $x(t_2)$, where t_1 and t_2 are variables:

$$R_{xx}(t_1, t_2) = E\{x(t_1)x(t_2)\} = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x_1 x_2 f(x_1, x_2; t_1, t_2) dx_1 dx_2 \tag{2-71}$$

For $t_1 = t_2 = t$, we have

$$R_{xx}(t,t) = E\{x(t)x(t)\} = E\{x^2(t)\} \tag{2-72}$$

In general, for a complex process $x(t)$, the autocorrelation of $x(t)$ is given by:

$$R_{xx}(t,t) = E\{x(t)x^*(t)\} \tag{2-73}$$

where $x^*(t)$ is the complex conjugate of $x(t)$. The autocovariance of a real process $x(t)$, denoted by $C_{xx}(t_1, t_2)$, is given by:

$$C_{xx}(t_1,t_2) = E\{[(x(t_1) - \eta_x(t_1))][(x(t_2) - \eta_x(t_2))]\}$$

$$= R_{xx}(t_1,t_2) - \eta_x(t_1)\eta_x(t_2) \tag{2-74}$$

The variance of $x(t)$ is then given by:

$$var\{x(t)\} = \sigma^2_{x(t)} = E\{[(x(t) - \eta_x(t))]^2\} \tag{2-75}$$

$$= C_{xx}(t,t) = R_{xx}(t,t) - \eta_x(t)^2 = E\{x^2(t)\} - \eta(t)^2 \tag{2-76}$$

In general, for a complex process $x(t)$, the autocovariance of $x(t)$ is given by:

$$C_{xx}(t_1,t_2) = E\{[(x(t_1) - \eta_x(t_1))][(x^*(t_2) - \eta^*_x(t_2))]\} \tag{2-77}$$

$$= R_{xx}(t_1,t_2) - \eta_x(t_1)\eta^*_x(t_2) \tag{2-78}$$

The correlation coefficient $r(t_1, t_2)$ of a process $x(t)$ *is* given by:

$$r(t_1,t_2) = \frac{C_{xx}(t_1,t_2)}{\sqrt{C_{xx}(t_1,t_1)C_{xx}(t_2,t_2)}} \tag{2-79}$$

The cross-correlation is a measure of the correlation between two random variables defined at two time points from two different stochastic processes, $x(t)$ and $y(t)$. Consider the two random variables defined for two time points, t_1 for the real process $x(t)$ and t_2 for the real process $y(t)$: $x(t_1)$ and $y(t_2)$. The cross-correlation of real processes $x(t)$ and $y(t)$, denoted by $R_{xy}(t_1, t_2)$, is defined as the expected value of the product of $x(t_1)$ and $y(t_2)$, where t_1 and t_2 are variables:

$$R_{xy}(t_1,t_2) = E[x(t_1)y(t_2)] = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x_1 y_2 f_{xy}(x_1, y_2; t_1, t_2)dx_1 dy_2 \quad (2\text{-}80)$$

For $t_1 = t_2 = t$, we have

$$R_{xy}(t,t) = E[x(t)y(t)] \quad (2\text{-}81)$$

In general, for two complex processes $x(t)$ and $y(t)$, the cross-correlation of $x(t)$ and $y(t)$ is given by:

$$R_{xy}(t_1,t_2) = E[x(t_1)y^*(t_2)] \quad (2\text{-}82)$$

The cross-covariance of real processes $x(t)$ and $y(t)$, denoted by $C_{xy}(t_1, t_2)$, is defined by:

$$C_{xy}(t_1,t_2) = E\{[(x(t_1) - \eta_x(t_1))][(y(t_2) - \eta_y(t_2))]\} \quad (2\text{-}83)$$

$$= R_{xy}(t_1,t_2) - \eta_x(t_1)\eta_y(t_2) \quad (2\text{-}84)$$

In general, for complex processes $x(t)$ and $y(t)$, their cross-covariance is given by:

$$C_{xy}(t_1,t_2) = E\{[(x(t_1) - \eta_x(t_1))][(y^*(t_2) - \eta_y^*(t_2))]\} \quad (2\text{-}85)$$

$$= R_{xy}(t_1,t_2) - \eta_x(t_1)\eta_y^*(t_2) \quad (2\text{-}86)$$

3.4 The normal process

A stochastic process $x(t)$ is normal, if the n random variables defined for n arbitrarily selected time points, $x(t_1), x(t_2), \ldots, x(t_n)$, are jointly normal for any n. For $n = 1$, the first-order statistics of the normal process is:

$$f(x,t) = \frac{1}{\sigma(t)\sqrt{2\pi}} e^{-(x-\eta(t))^2/2\sigma^2} \tag{2-87}$$

For $n = 2$, the second-order statistics of the normal process is:

$$f(x_1, x_2; t_1, t_2) = \frac{1}{2\pi\sigma_1(t)\sigma_2(t)\sqrt{1 - r^2(t_1, t_2)}}$$

$$\times exp[-\frac{1}{2[1 - r^2(t_1, t_2)]}(\frac{x_1^2}{\sigma_1^2(t)} - 2r(t_1, t_2)\frac{x_1 x_2}{\sigma_1(t)\sigma_2(t)} + \frac{x_2^2}{\sigma_2^2(t)})] \tag{2-88}$$

3.5 Statistical characterization of a stochastic process

How does one go about characterizing a stochastic process $x(t)$ statistically? How does one know that a stochastic process $x(t)$ is "completely" characterized statistically? What statistical information or data characterizes a stochastic process $x(t)$ completely? These are the same question phrased differently. The key word is "completely." Unless the ultimate, i.e., "complete," statistical information is defined, it would be hard to determine what to search for and when to stop collecting data.

To discuss this concept, let us start with a simple random variable x. The complete statistical information of x is its CDF or pdf. The CDF or the pdf of x represents all the data one can possibly have statistically for x. If you have the pdf of x, you can derive the mean, the variance, and the higher order moments of x. However, the converse is in general not true unless the random variable is known or assumed to be a certain kind, e.g., normal, Poisson, etc. The statistical moments of x are not in general sufficient to derive the CDF or pdf of x. For certain types of random variables, e.g., the normal random variable x, however, the mean and variance of x are sufficient to determine the pdf of x.

Let us now carry this discussion to the stochastic process $x(t)$. Recall that $x(t)$ is a set or collection of random variables in time t. To characterize $x(t)$, proceed as follows. First, pick n arbitrary time points, $t_1, t_2, \ldots, t_n$. For

these n time points, we now have n random variables: $x(t_1), x(t_2), \ldots\ldots, x(t_n)$. For these n random variables, consider the n^{th}-order statistics, i.e., n^{th}-order joint CDF or joint pdf, as follows:

$$F(x_1, x_2, \ldots\ldots, x_n; t_1, t_2, \ldots\ldots, t_n)$$

$$= P\{x(t_1) \le x_1, x(t_2) \le x_2, \ldots\ldots, x(t_n) \le x_n\} \qquad (2\text{-}89)$$

$$f(x_1, x_2, \ldots\ldots, x_n; t_1, t_2, \ldots\ldots, t_n)$$

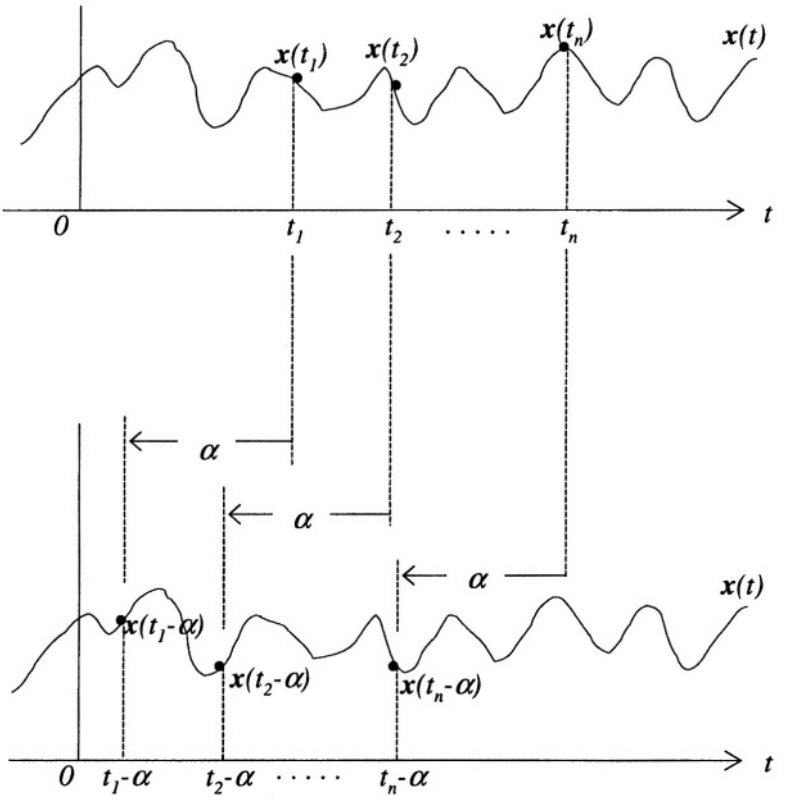


Figure 2-5. Strict sense stationary (SSS).

$$= \frac{\partial^n F(x_1, x_2, \ldots\ldots, x_n; t_1, t_2, \ldots\ldots, t_n)}{\partial x_1 \partial x_2 \ldots\ldots\ldots \partial x_n} \qquad (2\text{-}90)$$

To define a stochastic process $x(t)$ completely statistically, one must

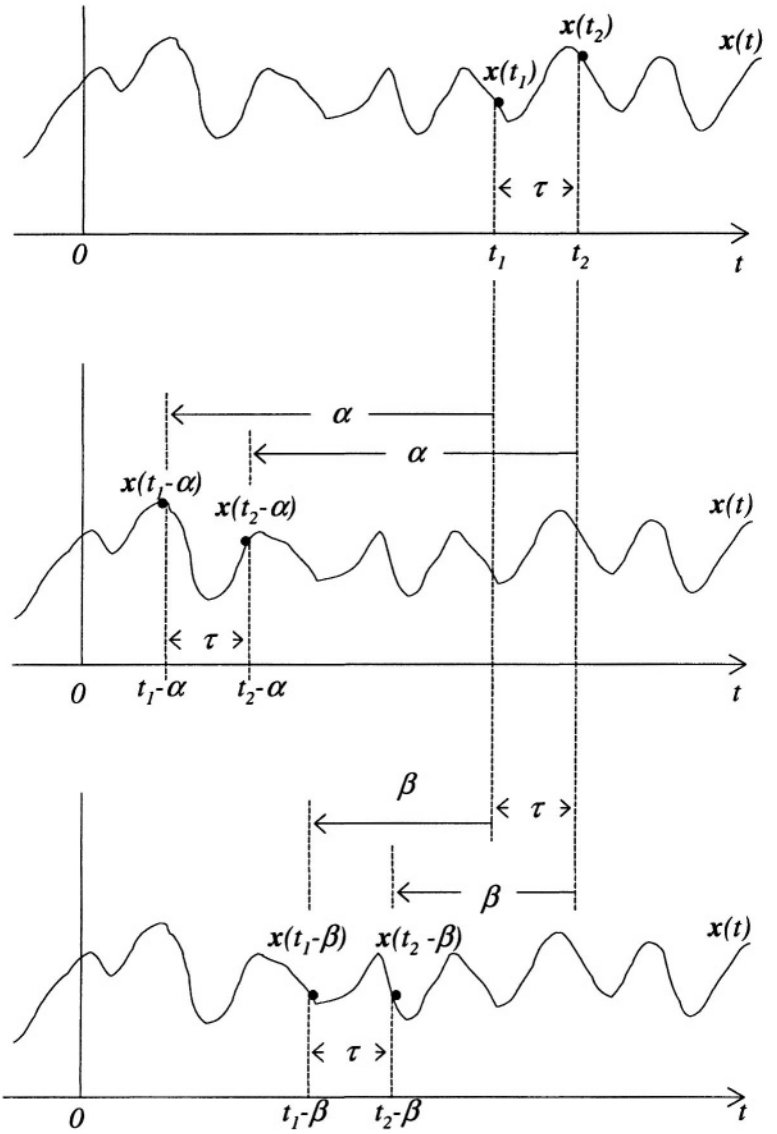


Figure 2-6. Strict sense stationarity.

obtain the ***n*th-order** joint CDF or pdf defined above for an arbitrary n and for an arbitrary set of n time points. For a continuous process $x(t)$, one can see that it is not possible to complete the process of data collection to satisfy this definition. First of all, n can be indefinitely large. Secondly, for a given n, there are an infinite number of possibilities of choosing the n time points.

Nevertheless, this ideal definition of "statistical characterization" of a stochastic process $x(t)$ provides a framework for statistical investigators of a random process: i) to determine how to go about collecting data and ii) to determine when to stop collecting data, i.e., how much data is enough.

3.6 Stationarity

A very important concept in stochastic processes is *stationarity*. In practice, unless the stochastic process under consideration is stationary, it is in general intractable for analysis or simulation. If a process is non-stationary, therefore, it can be divided into a number of sub-processes defined over smaller sub-time intervals over which the sub-processes are either stationary or approximately stationary. The analysis of the original non-stationary process can then be performed by analyzing the sub-processes and synthesizing the results of the sub-processes. Two types of stationarity are defined: strict sense stationarity (SSS) and wide sense stationarity (WSS). SSS is stronger (or harder to meet) than WSS.

3.6.1 Strict sense stationarity (SSS)

A stochastic process $x(t)$ is strict sense stationary (SSS) if its statistical characterization defined in Section 3.5 is invariant to a shift of the origin. This simple definition has the following profound implication. Suppose that,

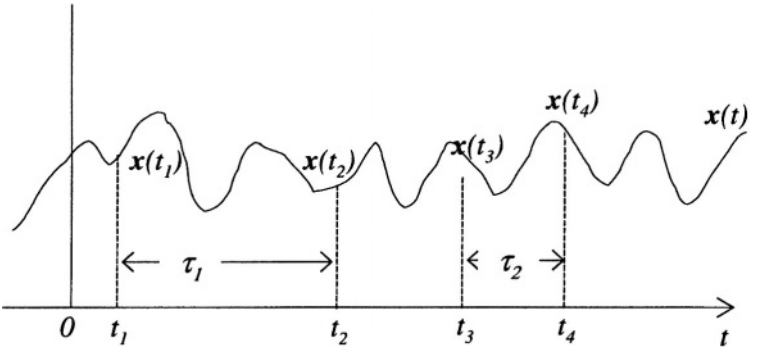


Figure 2-7. SSS.

to characterize $x(t)$, a set of n time points and the corresponding n random variables are selected: $x(t_1), x(t_2), \ldots, x(t_n)$. Next, shift the original n random variable to the left in time by the same amount α and consider the resulting n random variables: $x(t_1 - \alpha), x(t_2 - \alpha), \ldots, x(t_n - \alpha)$. Figure 2-5 shows the original n RV's and the n shifted RV's

To satisfy the above definition of SSS, the two sets of n random variables must have the same n^{th}-order joint pdf as follows:

$$f(x_1, x_2, \ldots, x_n; t_1, t_2, \ldots, t_n)$$

$$= f(x_1, x_2, \ldots, x_n; t_1 - \alpha, t_2 - \alpha, \ldots, t_n - \alpha) \tag{2-91}$$

Just as the complete statistical characterization of $x(t)$ is empirically not possible, establishing strict sense stationarity empirically is not impossible because the above equality must be established for any n and for any arbitrary n time points, and, finally, for any value of α. If $x(t)$ is SSS satisfying the above general equation, the following properties can be derived. For all values of α:

$$f(x;t) = f(x;t-\alpha) \tag{2-92}$$

The above equation indicates that the 1st order pdf of SSS $x(t)$ does not change as time t is varied. This means that $f(x;t)$ of an SSS $x(t)$ is independent of t: that is, for all values of t, $x(t)$ has the same pdf.

$$f(x;t) = f(x) \tag{2-93}$$

This means that an SSS $x(t)$ has a constant mean and a constant variance:

$$\eta_x(t) = \eta_x; \qquad \sigma_x^2(t) = \sigma_x^2 \tag{2-94}$$

For an SSS $x(t)$, the following is true for all values of α:

$$f(x_1, x_2; t_1, t_2) = f(x_1, x_2; t_1 - \alpha, t_2 - \alpha) \tag{2-95}$$

In the above equation, let $\tau = t_2 - t_1$, and rewrite it as follows:

$$f(x_1, x_2; t_1, t_2) = f(x_1, x_2; t_1 - \alpha, t_2 + t_1 - t_1 - \alpha)$$

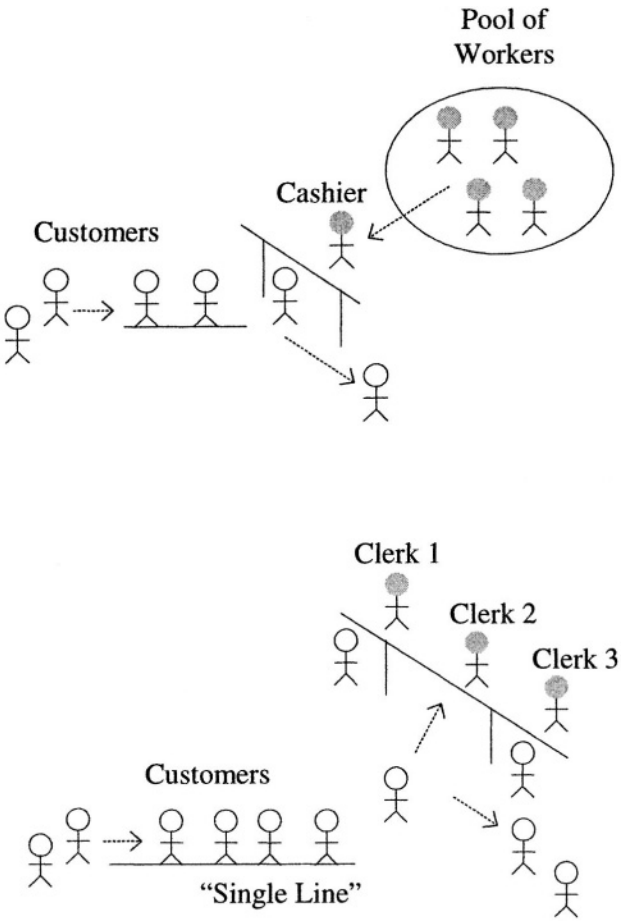


Figure 2-8. Real-life queuing example 1.

$$= f\{x_1, x_2; t_1 - \alpha, t_1 + (t_2 - t_1) - \alpha\}$$

$$= f(x_1, x_2; t_1 - \alpha, t_1 - \alpha + \tau) = f(x_1, x_2; \tau) \qquad (2\text{-}96)$$

In Figure 2-6, $x(t_1)$ and $x(t_2)$ are shifted to the left by α and β, respectively, keeping the distance between t_2 and t_1 at a constant τ. In this case, the 2nd-order pdf stays the same for both α and β; that is, as long as the distance between the two time points τ is constant, the amount of shift does

not change the 2nd-order pdf: the 2nd-order pdf of $\{x(t_1+\alpha), x(t_2+\alpha)\}$ and that of $\{x(t_1+\beta), x(t_2+\beta)\}$ are the same.

Figure 2-7 shows two pairs of time points and the random variables defined for those four time points: $\{x(t_1), x(t_2)\}$; $\{x(t_3), x(t_4)\}$. The distances between t_1 and t_2 and between t_2 and t_3 are τ_1 and τ_2, respectively. If $\tau_1 \neq \tau_2$, the 2nd-order pdf's of the two pairs of random variables are in general not equal as follows:

$$f(x_1, x_2; t_1, t_2) \neq f(x_3, x_4; t_3, t_4) \quad \text{if } \tau_1 \neq \tau_2 \tag{2-97}$$

where $\tau_1 = t_2 - t_1; \tau_2 = t_4 - t_3$.

3.6.2 Wide sense stationarity (WSS)

A stochastic process $x(t)$ is wide sense stationary (WSS) if the following two conditions are met. First, the mean of $x(t)$ is constant, i.e., independent of t:

$$\eta_x(t) = E\{x(t)\} = \eta_x \tag{2-98}$$

The second condition for WSS is that the autocorrelation function $R(t_1, t_2)$ is a function of the difference between t_2 and t_1, i.e., τ, only:

$$R_{xx}(t_1, t_2) = E\{x(t_1)x^*(t_2)\} = R_{xx}(t_1 - t_2) = R_{xx}(\tau) \tag{2-99}$$

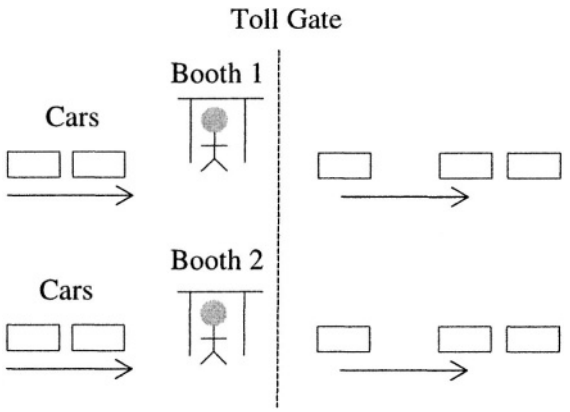


Figure 2-9. Toll gate example.

4. QUEUING THEORY BASICS

The QoS mechanisms implemented in packet routers and switches use various types of queuing discipline. Some basic understanding of queuing theory will help the reader appreciate the QoS performance analysis. This section reviews the following topics:

- Real-life examples of queuing
- Random arrivals
- Random service times
- Utilization factor
- Queuing system metrics
- *M/M/1* queue

4.1 Real-life examples of queuing

Consider several examples of queuing situations that we all experience in our everyday lives. In Figure 2-8, two real life examples of queuing are shown. A customer comes to a place, say, a fast food restaurant, to be served, joins the queue, and, when his turn comes, receives the service and leaves the place. The customer is served by a cashier. Behind the cashier, however, a team of workers help provide the service. How fast the service is provided can be controlled by controlling the number of workers in the pool. If the service is too slow, the restaurant manager can hire more workers and add them to the pool; if the manager considers that operation is too expensive, the manager can reduce the work force. In the former case, the customer service would improve; in the latter case, it would deteriorate.

The second example of Figure 2-8 is a typical bank example. There is a single line of customers and multiple tellers are serving the line. Whenever a teller becomes available the customer at the Head of Queue (HoQ) moves forward to the teller, receives the service and leaves.

The two cases shown in Figure 2-8 are similar and can be modeled by a single line single server queue. In the bank example, the multiple tellers can be considered a single server, i.e., a single pool of tellers, serving the single line. Once again, how fast the service can be provided can be controlled by the number of tellers serving the line.

Figure 2-9 shows another example that we experience at toll gates. A car comes to a toll gate and "randomly" selects one of the lanes for a toll booth, pays the toll and leaves. If all of the booths are assumed to be equal in service rates and other characteristics and that the cars select a booth randomly, the toll gate example can also be modeled as a single-line single-server queue.

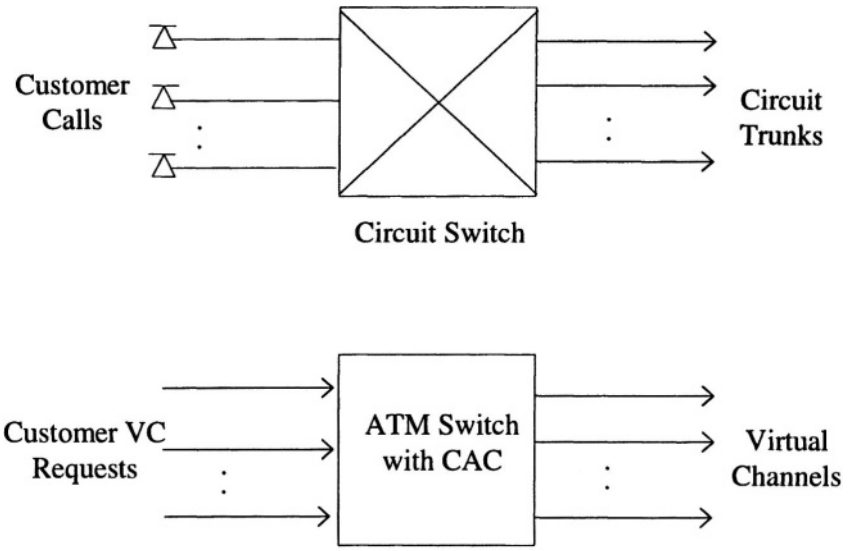


Figure 2-10. Network switching examples.

Figure 2-10 shows two switching examples: telephone calls served by trunks at a circuit switch and virtual connection requests served by an ATM switch. Customers' telephone lines are served by a central office. When a customer attempts to make a call going outside of the serving central office, the switch first tries to find an idle trunk for the call. If an idle trunk is available, the call is place on that trunk. If no trunk is available, the customer gets an "all trunks busy" signal.

In the ATM example in the figure, a virtual connection request arrives at

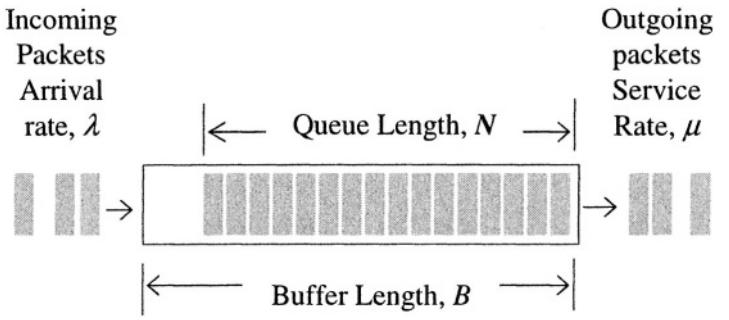


Figure 2-11. Packets arriving at a packet switch.

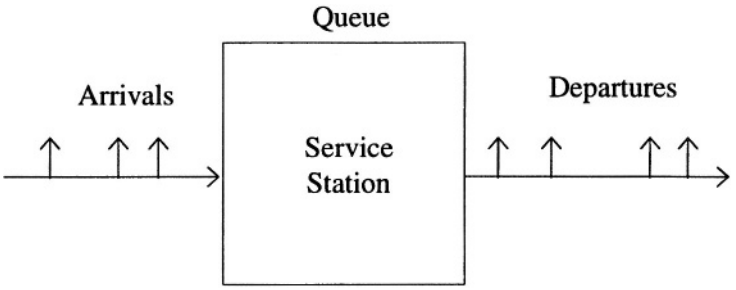


Figure 2-12. Queuing system.

the ATM switch. A Connection Admission Control (CAC) algorithm implemented in the switch determines whether there is enough bandwidth available at the output port to accept the request. If the answer is affirmative, the virtual connection request is accepted; otherwise, it is rejected. This type of queuing system is analyzed by the Erlang B and C systems. The Erlang systems will be discussed in detail in Chapter 3 and also in Chapter 6 for the ATM CAC.

Finally, Figure 2-11 shows the incoming packets that are put into a buffer in an IP router. The packet scheduler determines which packets are sent out from the output port. Various types of packet schedulers are treated in detail in Chapter 4.

The following are some of the typical questions that would be of interest in various types of queuing situations:

- How long would the customer line be?
- How long would a customer wait?
- How quick would the service be?

Queuing theory is a mathematical discipline that addresses this type of questions.

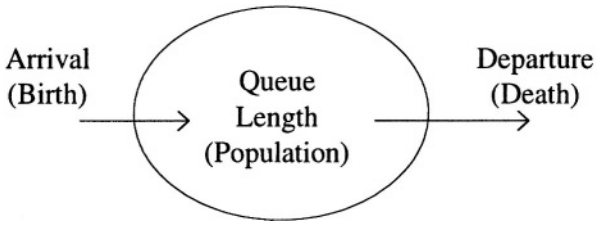


Figure 2-13. Birth-death process model.

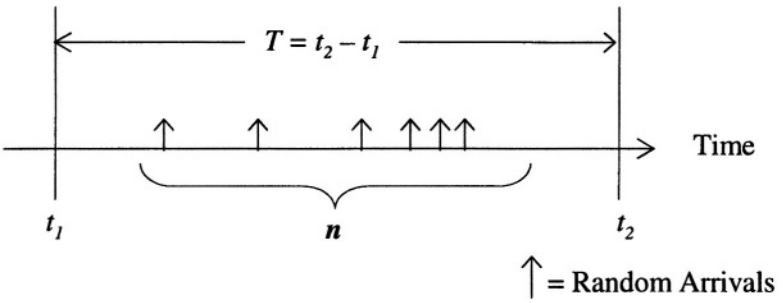


Figure 2-14. Packets arriving at a packet switch.

4.2 Definition of queuing system

In queuing theory, a place or "entity" where some kind of service is rendered is generically referred to as a "service station;" the word "customer" refers to any object that arrives at a service station to get a service. For example, a packet in Figure 2-11 is a "customer" in this sense.

In the examples described in Section 4.1, the following common features may be noted. A customer comes to a "service station," joins a line or "queue" and waits for his/her turn. When the turn comes, the customer receives the service and departs the service station. Figure 2-12 shows a queuing system. A queuing system is a mathematical model of the situation where customers arrive randomly at a service station, get service and depart the service station. A queuing system is defined by probabilistic characterizations of:

- Arrival pattern
- Service mechanism
  - When the service is available
  - How many customers can be served at a time
  - How long the service takes
- The queue-discipline
- The method by which a customer is selected for service

4.3 Birth-death process model

Consider the following examples:

- People being born and dying
- People arriving and leaving in a park

In these examples, the world and the park may be considered a queue; a baby's birth and a person's entrance into the park, an arrival; a person's death and a person's leaving the park, a service completion and departure; the size of the population of the world or the park, the queue length ***N***; and the amount of time a person spends in the world, i.e., age, or in the park, queuing delay ***d***, etc.

Under certain conditions of random arrivals and departures (i.e., services), e.g., Markov chain conditions, a queuing system can be modeled as a special class called the "birth-death" process. The birth-death process is used commonly in population studies, biology, etc. The birth-death process is considered to be most analytically tractable. An important class of queue referred to as the *M/M/1* queue is the "birth-death" process and will be discussed later.

4.4 Arrival rate

4.4.1 Definition

Figure 2-13 illustrates random arrivals. Consider random arrivals in the time interval from t_1 to t_2. The interval length is $T = t_2 - t_1$. The arrival rate is a long term average of the number of arrivals per unit time. Its mathematical symbol is λ and its unit is time^{-1}, and is given by the following equation, where *n* is the number of arrivals in the interval of length *T*.

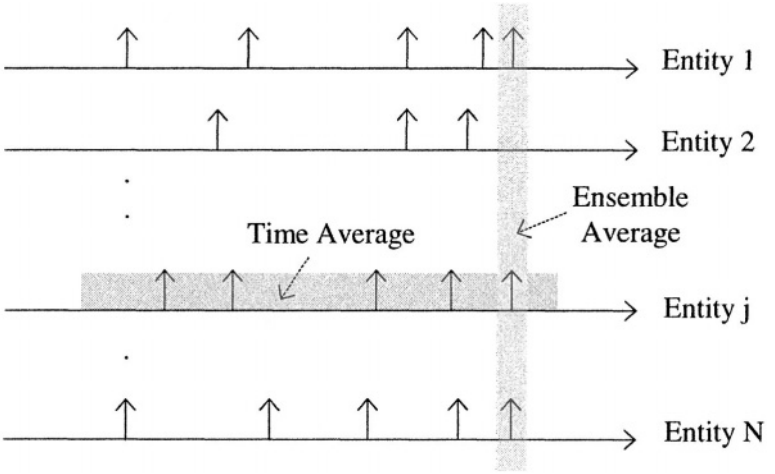


Figure 2-15. Time average and ensemble average.

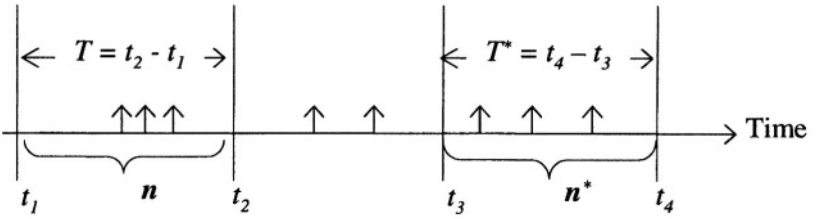


Figure 2-16. Illustration of stationarity.

$$\lambda = \frac{n}{T}$$

Example 2

The number of packets arriving at a packet switch during the one-minute period of from 9:00 *a.m.* to 9:01 *a.m.* has been counted over 30 days. The total count is 42 million packets. What is the packet arrival rate during this period?

Solution

$$\lambda = \frac{42 \times 10^6}{30 \times 1\,min} = 1.4 \times 10^6 / min$$

4.4.2 Empirical determination of arrival rate

There are two types of statistical averages:

- "Time average"
- "Ensemble average"

Figure 2-15 illustrates the two types of averages. The time average is the average taken over a period of time for the same physical entity. The ensemble average is the average taken over a number of physical entities (of the same kind) at a fixed time.

To illustrate these two types of averages, consider the toll gate example of Figure 2-9. Suppose that there are 10 lanes (or toll booths) at a toll plaza. Cars arrive at the toll plaza randomly and select a lane (or toll booth) randomly.

In this example, a time average of the arrival rate is obtained by selecting one toll booth, say Lane 1, counting the number of cars arriving at that

particular toll booth over a long period of time, say from *09:00 a.m.* to *10:00 a.m.,* and dividing the total count by the total measurement interval. This is a time average of arrivals per hour. Dividing this by 60, one can get a time average of arrivals per minute.

An ensemble average of the arrival rate is obtained by fixing a short interval, say one minute interval from a fixed time, say *09:00 a.m. – 09:01 a.m.,* counting the number of cars arriving at all 10 lanes, and dividing the total count by 10. This gives an ensemble average of arrivals per minute for 9:00 a.m.

It is important to understand the following two key concepts for measurement conditions:

- Stationarity
- Ergodicity

4.4.3 Stationarity

Section 3.6 defines Stationarity. To apply the concept of Stationarity to the measurements of arrival rate, consider two non-overlapping time intervals of equal length, T and T^*, shown in Figure 2-16. The random numbers of arrivals in these two intervals are denoted by $\boldsymbol{n}$ and $\boldsymbol{n}.^*$ If the arrival process is SSS, the two random variable $\boldsymbol{n}$ and $\boldsymbol{n}^*$ have the same statistics, e.g., same CDF and pdf. If the process is WSS, $\boldsymbol{n}$ and $\boldsymbol{n}^*$ may not have the same pdf, but they have the same mean: $\eta_n = \eta_{n^*}$.

In general, Stationarity would be harder to assume for a longer measurement interval. Examples of stationary arrivals may be car traffic during a one-hour rush hour period and telephone call traffic during a one-hour busy hour period. Examples of non-stationary arrivals may be traffic

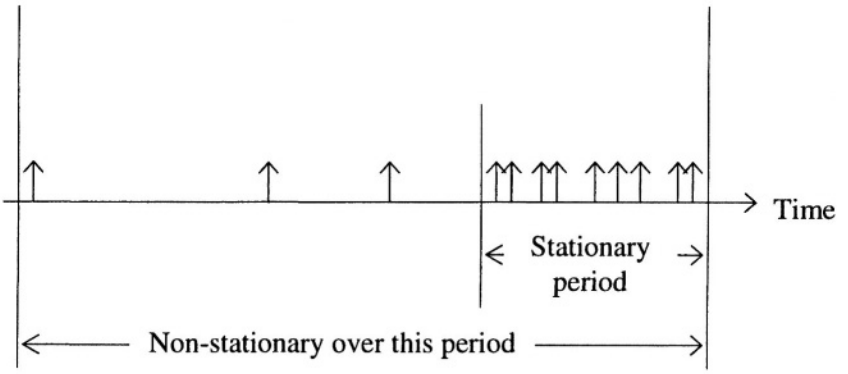


Figure 2-17. Example of stationary and non-stationary arrivals.

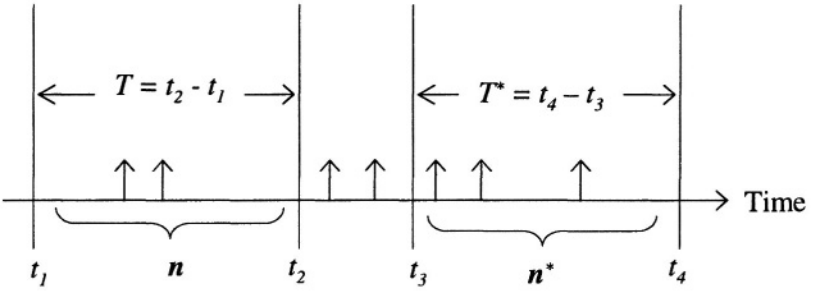


Figure 2-18. Poisson arrival.

over a 24-hour period and telephone call traffic over a 24 hour period. Figure 2-17 illustrates these examples.

4.4.4 Ergodicity

A complete definition and discussion of ergodicity are beyond the scope of this book. In this section, however, ergodicity is discussed for the mean or "average." A random process is "mean ergodic" if its time average is equal to its ensemble average. For example, to empirically determine the telephone call arrival rate λ for a central office in an area, consider the two types of averages determined as follows.

Suppose that the ensemble average is determined by taking the average across, say 10 randomly selected central offices in the area for a short fixed time interval, say, a busy hour from *12:00 p.m. – 01:00 p.m.* The time average is determined by taking the average at a single randomly selected central office, say office 7, over a longer time interval, say 24 hours. If mean ergodicity holds, the two methods should produce the same average. The ergodicity cannot hold for a non-stationary arrival process: stationarity is a necessary condition for ergodicity.

In the example above, if telephone traffic is non-stationary over the 24-hour period, it would be unreasonable to expect that the average over the 10 offices taken over a short one-hour interval and that taken over the 24-hour interval would be the same. The telephone office traffic engineering is based on busy hour statistics.

4.4.5 The Poisson Arrival

A Poisson arrival process is a random process in which the probabilities of the number of random arrivals in two non-overlapping intervals T and T^*,

$n(T)$ and $n(T^*)$, are independent. In a Poisson arrival process, the probability of future arrivals is not affected by previous arrivals.

A Poisson arrival process must satisfy the following conditions:

- The reservoir of arrivals is infinite.
- Stationarity.
- Ergodicity.

Poisson arrivals are often referred to as "pure random arrivals." A Poisson arrival process is an idealized mathematical model. The applicability of a Poisson model must be evaluated for each application under consideration for reasonableness.

The Poisson distribution is defined by a single parameter λ. Given λ, the probability of k arrivals in a time interval of length T is given as a function of T by the following equation:

$$P(k \text{ in } T) = e^{-\lambda T} \frac{(\lambda T)^k}{k!} \tag{2-100}$$

The unit of λ and T must be consistent in time. For example, if $\lambda = 120/\text{hour}$ in the above equation, T needs to be expressed in hours. If T is expressed in minutes, λ must be converted to 120/60 minutes, which yields 2/minute.

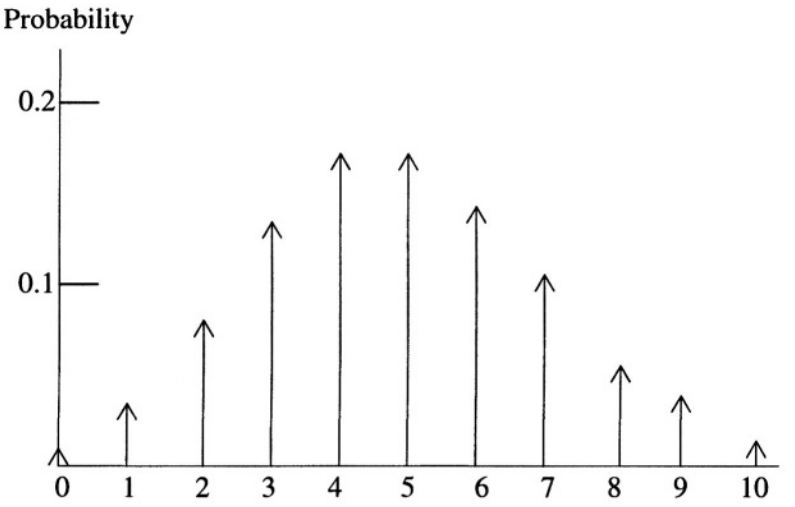


Figure 2-19. Poisson probability distribution.

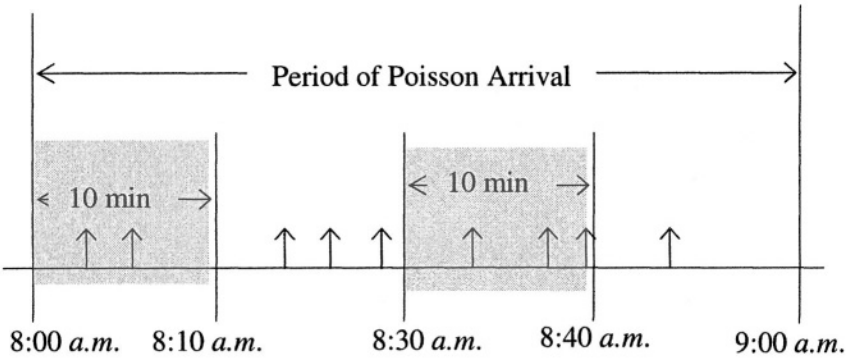


Figure 2-20. Poisson arrival.

Example 3

Consider packets arriving at a packet switch at the following arrival rate: $\lambda = 6 \times 10^6 / min$. Assuming a Poisson arrival, what is the probability that two packets will arrive in a 10-μsec interval?

Solution

Use the Poisson pdf with the following values:

$$\lambda = 6 \times 10^6 \ /min = 1 \times 10^5 \ /sec = 0.1 / \mu \ sec . \qquad T = 10 \ \mu sec$$

Hence $\lambda T = 0.1 \times 10 = 1$. For $k = 3$, the Poisson pdf yields

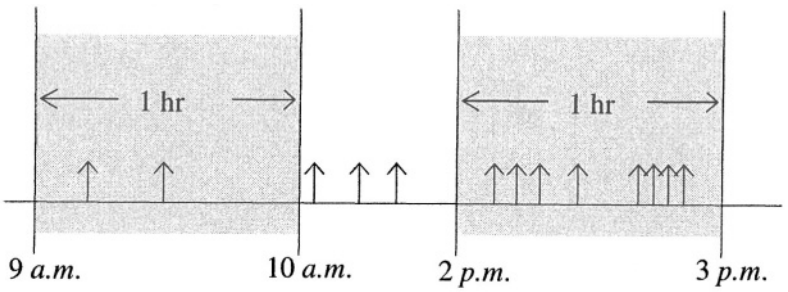


Figure 2-21. Non-Poisson arrival.

$$P\{3 \text{ in } 0.1\ \mu sec\} = e^{-1}\frac{(1)^3}{3!} = 0.06$$

Suppose that the arrival pattern is Poisson for the time interval from *8:00 a.m. – 9:00 a.m.* Under this assumption, probability distributions over the 10 minute period, e.g., from *08:00 - 08:10 a.m.* and that between *08:30 – 08:40 a.m.* are assumed to be independent and identically distributed (i.i.d). This is illustrated in Figure 2-20. On the other hand, a 10-minute interval taken from 8 – 9 *a.m.* and a 10-minute interval taken from, say, 2 – 3 *p.m.* may not have the same arrival characteristics, and stationarity would not hold over this stretched time period. In this case, the Poisson model would not apply. This is illustrated in Figure 2-21.

Consider the following two examples:

- People arriving at a bus station in a large city with millions of people
- People arriving at a bus station in a small community (with a few people)

Consider the significance of the difference between the "large" city and the "small" community. In the first case, the Poisson model may be applied because there is an infinite reservoir of arrivals. In the latter case, the Poisson model would not apply. For example, suppose that there are only five people in the community who take the bus. If five people have already arrived in T^*, the probability of an arrival in T is obviously zero, i.e., n and n^* are not independent. This violates one of the Poisson assumptions that the reservoir of arrivals is infinite.

Figure 2-22 illustrates inter-arrival times. The inter-arrival time, t, is a random variable and is defined as the time between two consecutive arrivals.

The inter-arrival times of Poisson arrivals are exponentially distributed with the following CDF and pdf, where λ is the arrival rate:

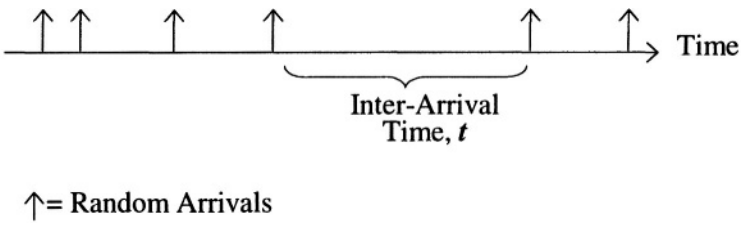


Figure 2-22. Inter-arrival time

$$CDF \quad F(t) = 1 - e^{-\lambda t} \qquad pdf \quad f(t) = \lambda e^{-\lambda t} \qquad (2\text{-}101)$$

By taking the expected value of t, the mean of the inter-arrival time is found to be $1/\lambda$ as follows:

$$\eta_t = E\{t\} = \int_{-\infty}^{+\infty} t f_t(t)dt = \int_{-\infty}^{+\infty} t(\lambda e^{-\lambda t})dt = \frac{1}{\lambda} \qquad (2\text{-}102)$$

Example 4

Assume a Poisson arrival with the arrival rate $\lambda = 10/\mu sec$. Find the probability that the inter-arrival time between consecutive arrivals is greater than $0.1\ \mu sec$.

Solution

$$P\{t > 0.1\ \mu sec\} = 1 - P\{t \le 0.1\ \mu sec\} = 1 - F(0.1)$$

$$= 1 - (1 - e^{-10 \times 0.1}) = e^{-1} = 0.37$$

4.4.6 Markov Modulated Poisson Process (MMPP)

The MMPP process $x(t)$ is a non-stationary process, which is composed of n separate Poisson processes, PP_i, $i = 1, \ldots, n$. While the process $x(t)$ is in "*state i*," $x(t)$ is Poisson process PP_i, with parameter λ_i. The process $x(t)$ moves or "makes transitions," between states at discrete points in time, t_m's. The state transitions are assumed to follow the Markov chain model. In a Markov chain, given that the process is in *state i* at time t_m, the probability that the process will transition into *state j* at the next time instant, t_{m+1}, is referred to as the transition probability p_{ij}. For an n-state MMPP, there are $n \times n$ transition probabilities, including the probability of staying in the current state, p_{ii}.

The $n \times n$ matrix of the transition probabilities is referred to as the transition probability matrix, Π, as follows:

$$\Pi = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1j} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2j} & \cdots & p_{2n} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ p_{i1} & p_{i2} & \cdots & p_{ij} & \cdots & p_{in} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nj} & \cdots & p_{nn} \end{bmatrix} \quad (2\text{-}103)$$

where the transition probability from *state i* to *state j*, p_{ij}, is the conditional probability defined as:

$$p_{ij} = P\{ x(t_{m+1}) \text{ in State } j \mid x(t_m) \text{ in State } i\} \quad (2\text{-}104)$$

Figure 2-23 shows an example of three-state MMPP. Figure 2-24 shows the three-state MMPP making transitions over time.

4.5 Service rate

The service rate is defined as the number of customers served in unit time. Its mathematical symbol is μ. Its unit is 1/time, i.e. time^{-1}.

$$\mu = \frac{m}{T} \quad (2\text{-}105)$$

where m is the number of customers served in the interval of length T.

The inverse of service rate, $1/\mu$, is the service time, which is the time expended to serve one customer. Assuming that the customers leave instantly after getting service, the departure rate is equal to the service rate.

Example 5

Referring to Figure 2-25, a packet switch has a single incoming port and five processors. Incoming packets can be routed to any idle processor. Each processor can serve on the average 1×10^6 packets per minute. What is the average service rate of the packet switch processor operation?

Solution

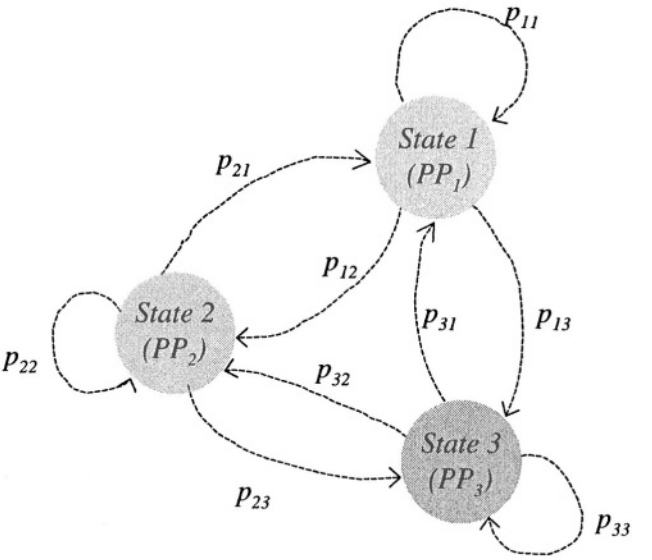


Figure 2-23. Three state Markov Modulated Poisson Process (MMPP).

$$\mu = \frac{1 \times 10^6 \times 5}{1\,min} = 5 \times 10^6 / min$$

Example 6

New processors are to be added to increase the service rate of the operation to $6x10^6$/min. Assume that the new processor can serve $0.5x10^6$ packets per minute. How many new processors need to be added?

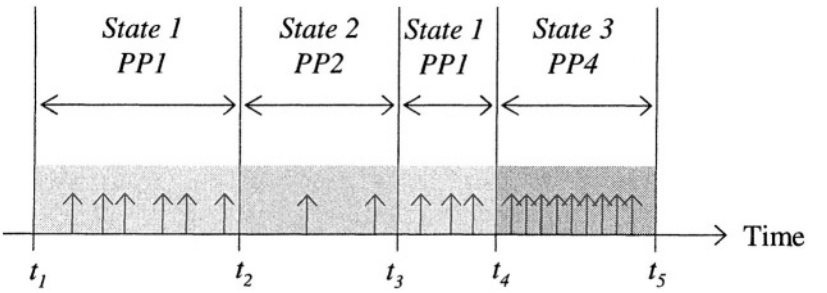


Figure 2-24. An example of MMPP in time.

Solution

Let the unknown number of new processors be *x*.

$$\mu = [(1 \times 10^6 \times 5) + (0.5 \times 10^6 \times x)]/_{min} = 6 \times 10^6/min$$

Solving for *x*, $x = 2$.

4.6 Utilization factor

Utilization factor is a measure of how fully the resource is used to meet the customer need. It is defined as the ratio of arrival rate to service rate. Its mathematical symbol is ρ, and ρ is dimensionless.

$$\rho = \frac{\lambda}{\mu} \tag{2-106}$$

In order for the queue to be stable, the service station should be able to serve customers at a faster rate than the customer arrival rate:

$$\mu \succ \lambda \tag{2-107}$$

In other words, in order for the queue to be stable, the utilization factor ρ should be less than one, i.e., less than 100% utilization of resources:

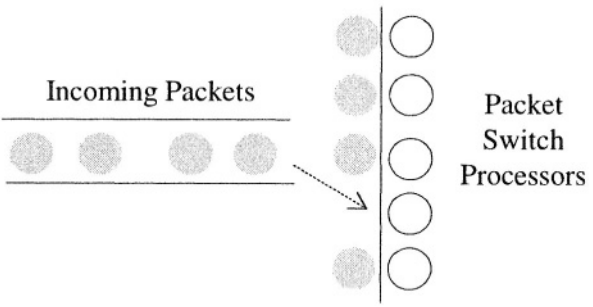


Figure 2-25. Example 5.

$$\rho = \frac{\lambda}{\mu} < 1 \tag{2-108}$$

In fact, the lower the utilization factor, the better the service to the customers. The lower the utilization factor, the shorter the queue length and the shorter the service delay (i.e., customer waiting).

Example 7

Consider a packet switch with the service rate $\mu = 5 \times 10^6$ *packets/min*. Assume that packets arrive at the packet switch at an arrival rate of $\lambda = 5 \times 10^4$ *packets/sec*. What is the utilization factor of the operation?

Solution

$$\rho = \frac{\lambda}{\mu} = \frac{5 \times 10^4 \times 60}{5 \times 10^6} = 0.6$$

4.7 Queuing system performance metrics

The metrics used for queuing system performance include:

- Queue length, *N*
- Delay or "waiting time" *d*

Note that both *N* and *d* are RV's.

4.7.1 Little's Theorem

A useful theorem on the relationship between the average queue length and the average queue delay is given by the Little's theorem as follows:

$$\eta_N = \lambda \times \eta_d \tag{2-109}$$

where λ is the arrival rate, $\eta_N = E\{N\}$ = *mean queue length* and $\eta_d = E\{d\}$ = *mean delay.*

A rigorous mathematical proof of the Little's theorem is beyond the scope of this book. In fact, only recently, this theorem was proven mathematically for arbitrary arrival and service time distributions.

A heuristic proof of the Little's theorem is illustrated in Figure 2-26. In the figure, suppose that Customer A arrives at the tail of queue (ToQ) at time

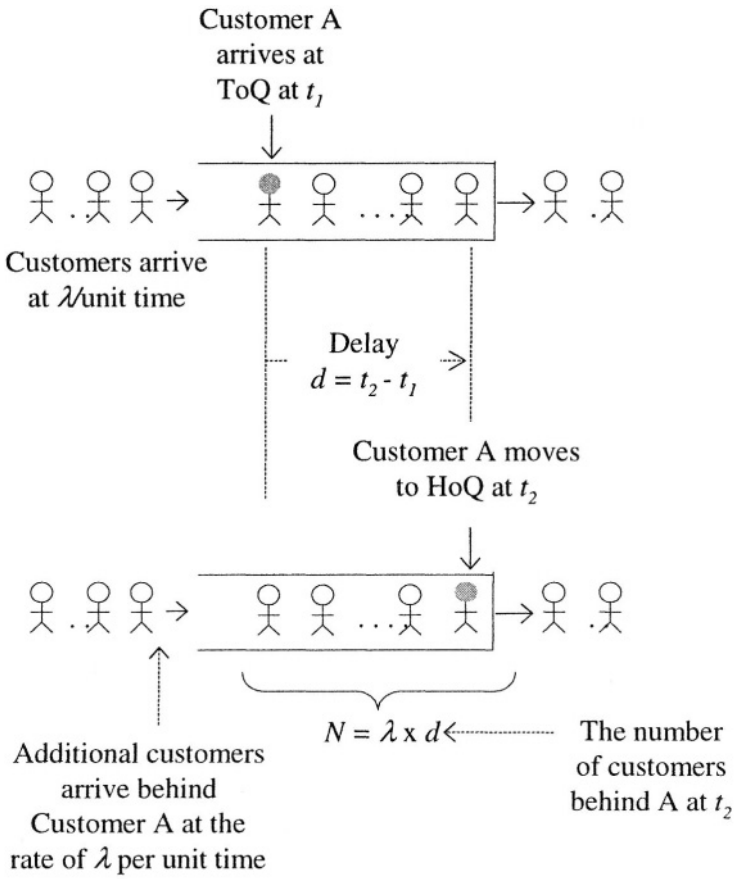


Figure 2-26. Heuristic proof of the Little's theorem.

t_1. Customer A moves to the head of the queue (HoQ) at time t_2. The amount of time Customer A spends in the queue is $d = t_2 - t_1$.

If customers behind Customer A arrive at the ToQ at the rate of λ, if Customer A turns around at the HoQ and counts the customers standing behind, he would see on the average λ x η_d customers. That is equal to the average queue length η_N.

4.8 *M/M/1* queue

The *M/M/1* queue is a single server queue with Poisson arrivals with an arrival rate λ and an exponential service time of a service rate μ. The CDF and pdf of the service time t are given below:

$$CDF \quad F(t) = 1 - e^{-\mu t} \qquad pdf \quad f(t) = \mu e^{-\mu t} \tag{2-110}$$

The *M/M/1* queue satisfies the "birth-death" process and is analytically tractable. The *M/M/1* queue is considered an idealized queuing model.

The probability that there will be k customers in the queue in a steady state is given by the following equation:

$$p_k = (1-\rho)\rho^k \qquad k = 0, 1, 2, 3, \ldots \tag{2-111}$$

where

$$p_k = \text{probability of } k \text{ customers in the queue} = P\{N = k\} \tag{2-112}$$

$$\rho = \frac{\lambda}{\mu}; \qquad \mu \succ \lambda .$$

Table 2-1 tabulates p_k given by the above equation. Observe that p_k depends on λ and μ only through their ratio ρ, i.e., λ and μ are not independent variables for p_k.

Setting $k = 0$ in the above equation, the probability that the queue will be empty is given by:

$$p_0 = (1-\rho)\rho^0 = 1-\rho. \tag{2-113}$$

The probability that there will be at least k customers in the queue is given by

$$P\{N \geq k\} = \rho^k \qquad k = 0, 1, 2, 3, \ldots$$

Table 2-1. Probability of k customers in the queue as a function of ρ

ρ	0.5	0.6	0.7	0.8	0.9
k					
0	0.50	0.40	0.30	0.20	0.10
1	0.25	0.20	0.21	0.16	0.09
2	0.13	0.10	0.15	0.13	0.08
3	0.06	0.10	0.10	0.10	0.07
4	0.03	0.10	0.07	0.08	0.07
5	0.02	0.00	0.05	0.07	0.06

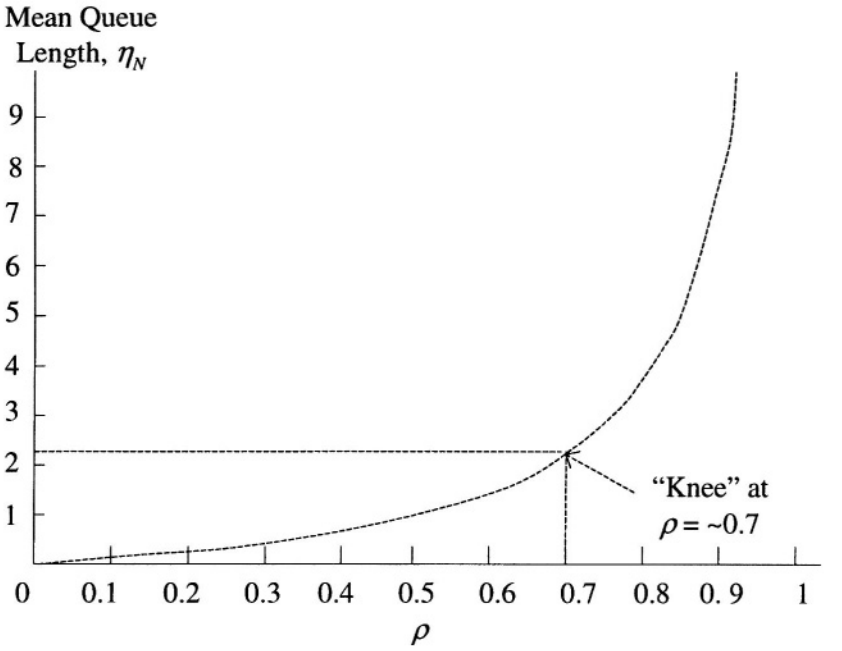


Figure 2-27. Mean delay,N, as a function of utilization factor, ρ

Since $P\{N \geq k\} + P\{N \prec k\} = 1$, the probability that there will be less than k customers in the queue is given by:

$$P\{N \prec k\} = 1 - \rho^k \qquad k = 0, 1, 2, 3, \ldots \tag{2-114}$$

The mean of the queue length, N, is given by:

$$\eta_N = \frac{\rho}{1-\rho}, \qquad \rho \prec 1 \tag{2-115}$$

For $\rho \geq 1$, $\eta_N \rightarrow \infty$. The variance of the queue length, N, is given by:

$$\sigma_N^2 = \frac{\rho}{(1-\rho)^2}, \qquad \rho \prec 1 \tag{2-116}$$

$$\sigma_N^2 = \frac{\rho}{(1-\rho)^2}, \qquad \rho < 1 \tag{2-116}$$

Observe that the variance of the queue length depends on λ and μ only through their ratio ρ, i.e., λ and μ are not independent variables.

Figure 2-27 plots the mean queue length, η_N, as a function of the utilization factor ρ. The mean queue length monotonically increases as the utilization factor increases, and goes to infinity at $\rho = 1$. The curve shows that, as ρ passes about 0.7, the mean queue length suddenly increases rapidly. There is a "knee" of the curve at about $\rho = 0.7$. A smart operator would be able to optimize the performance-to-cost tradeoff by recognizing the knee. For example, if the current operating point is slightly above the knee, say $\rho = 0.8$, the performance can be dramatically improved by bringing down ρ below 0.7 by adding a little more resources.

For example, suppose that a packet switch currently operates at a utilization factor $\rho = 0.9$. At this value of ρ, the average queue length would be nine packets. By decreasing ρ from 0.9 to 0.7, the average queue length can be dramatically cut down to 2.3, which will reduce the packet loss ratio. To improve the packet switch performance this way, more processors need to be added to reduce the utilization factor ρ.

To find the mean delay, η_d, consider the Little's theorem. Combining the Little's theorem of Equation (2-109) and Equation (2-115) yields:

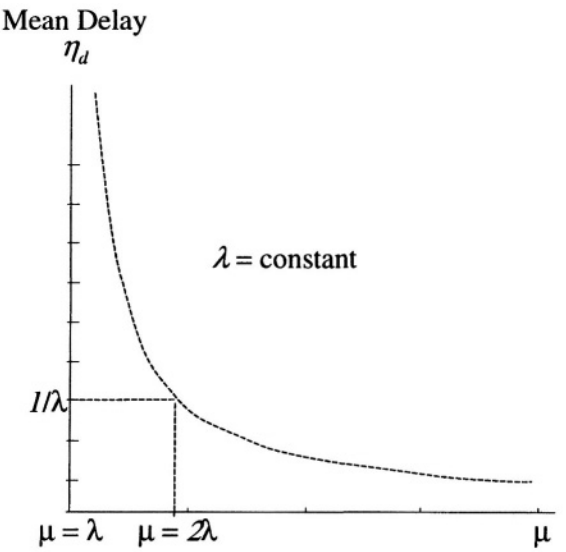


Figure 2-28. Mean delay, η_d, as a function of service rate, μ, for a fixed λ.

$$\eta_d = \frac{1/\mu}{1-\rho} = \frac{1}{\mu - \lambda} \qquad (2\text{-}117)$$

Figure 2-28 plots the mean delay, η_d, as a function of μ for a constant value of λ. The figure shows that the mean delay is infinite if μ is equal to λ. The mean delay is $1/\lambda$ at μ equal to 2λ.

5. EXERCISES

5.1 Problems

1. A player is to throw a die, and, if the face with an even number of spots shows up, the player wins the prize. What is the probability of winning the prize? Formulate the problem and solve it using the axiomatic approach.

2. The number of switched virtual circuit requests arriving at an ATM switch during the period of from *9:00 a.m.* to *10:00 a.m.* has been counted over *30* days. The total count is 900 requests. What is the arrival rate during the *9:00 - 10:00 a.m.* period?

3. ATM cells arrive at an ATM switch at the rate of λ = 600 cells/minute. Assuming that the arrivals are Poisson, what is the probability that two cells will arrive in a 0.1-second interval? (Use the Poisson distribution; use $e \cong 2.72$; $e^{-1} \cong 0.37$.)

4. Packets arrive at a packet switch at the rate of 10×10^6 packets/sec from 8:00 *a.m.* to 9:00 *a.m.* and 5×10^6 packets/sec from 9:00 *a.m.* to 10:00 *a.m.* Would it be reasonable to assume that the arrival pattern over the 8:00 *a.m.*–10 *a.m.* period is Poisson and why?

5. Is the following statement true or false?

   - An arrival pattern of passengers at a train station over a certain period has been found to be non-stationary and ergodic.
   - A Poisson process is stationary and ergodic.

6. A packet switch has a single input port and five processors. Packets go to any available processor. Each processor can process 1.2×10^6 packets per second. What is the service rate of the packet switch operation?

7. Cars arrive at a toll booth at an average rate of five cars per minute. The cashier at a toll booth serves cars at the rate of 600 cars per hour. What is the utilization factor of the toll booth operation? What percentage of the time would the cashier be idle?

8. For an $M/M/1$ queue with $\rho = 0.7$, find the probability that the queue length will be between zero and two inclusively, i.e. $P\{0 \leq N \leq 2\}$; and the probability that the queue will be empty.

9. For an $M/M/1$ queue of $\rho = 0.4$ and $\lambda = 10/\text{hour}$, determine the following:
   - mean queue length
   - mean delay through the queue
   - service rate

   Suppose that the arrival rate doubles but the queue operates with the same service rate. Determine the delay.

5.2 Solutions

1. Die throwing experiment

$$S = \{\xi_1, \xi_2, \xi_3, \xi_4, \xi_5, \xi_6\}$$

$$p_i = P(\xi_i) = 1/6; \qquad i = 1, \ldots, 6$$

The event of interest is "winning the grand prize" and is defined as a set denoted by W as follows:

$$W = \text{"even number of spots"} = \{\xi_2, \xi_4, \xi_6\}$$

Since $\{\xi_2\}$, $\{\xi_4\}$, and $\{\xi_6\}$ are mutually exclusive, i.e., $\{\xi_2\} \cap \{\xi_4\} = \{\phi\}$, $\{\xi_4\} \cap \{\xi_6\} = \{\phi\}$, and $\{\xi_2\} \cap \{\xi_6\} = \{\phi\}$, it follows that $D = \{\xi_2\} \cup \{\xi_4\} \cup \{\xi_6\}$. From Axiom III, it follows that:

$$P(W) = P(\{\xi_4, \xi_5, \xi_6\}) = P(\{\xi_4\} \cup \{\xi_5\} \cup \{\xi_6\})$$

$$= P([\{\xi_4\} \cup \{\xi_5\}] \cup \{\xi_6\}) = P([\{\xi_4\} \cup \{\xi_5\}]) + P(\{\xi_6\})$$

$$= P(\{\xi_4\}) + P\{\xi_5\} + P\{\xi_6\} = 1/6 \times 3 = 0.5$$

2. 900/30 = 30/hr.

3. $k = 2$; $T = 0.1$ sec.; $\lambda = 600/\text{min} = 600/60 = 10/\text{sec}$.
$\lambda T = 10 \times 0.1 = 1$
$P\{2 \text{ in } 0.1 \text{ sec}\} = (e^{-1})[(1)^2/(2!)] = (0.37)/[(2)(1)] = 0.185$.

4. No, because non-stationary.

5. False; True.

6 $\mu = 1.2 \times 10^6 \times 5/\text{sec} = 6 \times 10^6/\text{sec}$.

7.

$\lambda = 5/\text{min}$; $\mu = 600/\text{hr} = 600/60 \text{ min} = 10/\text{min}$; $\rho = \lambda/\mu = 5/10 = 0.5$. 50% idle.

8.
From the table, $P\{0 \le N \le 2\} = P\{0\} + P\{1\} + P\{2\} = 0.3 + 0.21 + 0.15 = 0.66$.; $P\{\text{empty}\} = P\{0\} = 0.3$.

9. Mean queue length

$$\eta_N = \frac{\rho}{1-\rho} = \frac{0.4}{1-0.4} = (0.4/0.6) = 0.67$$

$$\text{Mean delay } \eta_D = \frac{N}{\lambda} = \frac{0.4/0.6}{10/60} = (0.4/0.6) \times (60/10) = 4 \text{ min}$$

$$\text{Service rate: } \textit{Given } \rho = \frac{\lambda}{\mu} = 0.4; \quad \textit{Hence } \mu = \frac{\lambda}{0.4} = \frac{10}{0.4} = 25/\text{hr}$$

Same service rate : $\mu' = \mu$; *double arrival rate :* $\lambda' = 2\lambda$

Delay: $\eta_{D'} = \frac{1}{\mu' - \lambda'} = \frac{1}{\mu - 2\lambda} = \frac{1}{25 - 20} = \frac{1}{5} hr = 12 min$

Chapter 3

QOS METRICS

This chapter discusses the following topics:

- Components of digital communications system
- Network impairments and their sources
- Subjective testing
- Voice quality
- Codec performance
- Blocking probability for connection-oriented packet services

1. NETWORK TYPES

In general, telecommunications networks are divided into circuit-switched and packet-switched networks. Packet-switched networks are further divided into connection-oriented and connectionless packet networks. Figure 3-1 shows this classification.

1.1 Connection-oriented packet network services

A connection-oriented packet network service consists of the following three phases:

- Connection establishment
- Data transfer
- Connection tear-down

In the connection establishment phase, the source and the destination perform "handshaking" to establish a logical connection between the two peer entities. If the two end systems are connected by a dedicated line, the

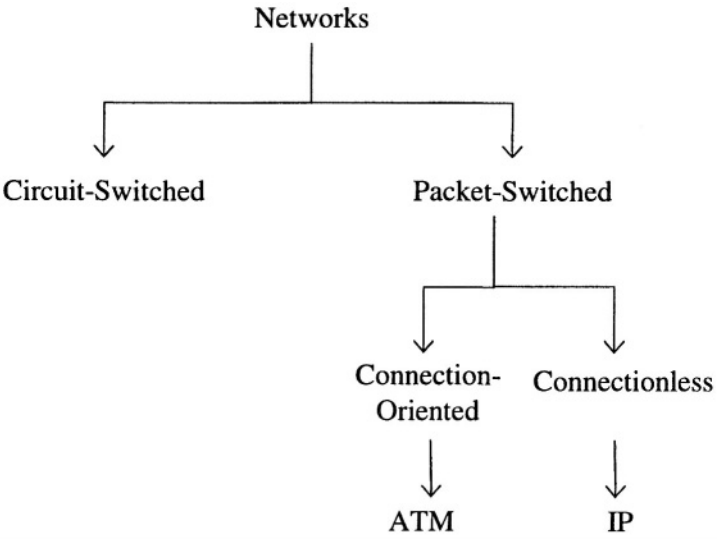


Figure 3-1. Network types.

handshaking could be a simple exchange of a "hello" and a response to initiate a communications session. On the other hand, if the two ends are connected by a packet network offered by a public network service provider, the protocol exchange for connection set-up process could be more involved. The connection establishment is based on the destination address.

Once the connection establishment phase is complete, the data transfer phase begins. Each packet contains an identifier for the virtual connection used for the data transfer. At each packet switching or forwarding node, the packets are switched (or forwarded) based on this identifier; that is, routing decisions for the packets are based on the identifier associated with the virtual connection instead of the destination address. Therefore, the individual packets of connection-oriented services do not contain destination addresses. As compared to the connectionless service discussed next, the packets in connection-oriented services can be forwarded more efficiently and quickly. Connection-oriented services deliver packets in order using sequential numbers; and typically, provide flow control and error control.

Finally, at the conclusion of data transfer, the virtual connection must be "torn down;" otherwise, it would tie up the network resources allocated for the virtual connection unnecessarily. The ATM network discussed in Chapter 6 is a connection-oriented packet network.

1.2 Connectionless packet network services

As the term implies, in connectionless packet network services, no logical connection is set up for exchanging data between end systems. In fact, the connectionless packet network service is similar to the U.S. Postal service. The sender of a postcard drops the postcard at a mail drop, and the postcard gets delivered to the receiver. The sender does not have to call the receiver first before sending the postcard. The connectionless packet network service is also referred to as the "datagram" service.

Since, in the connectionless service, there is no connection establishment, there is no tear-down phase either; the connectionless packet network service has only one phase: data transfer phase.

In the connectionless service, since there is no logical connection associated with the packets, each packet is treated independently of previous or subsequent packets. Furthermore, each packet must contain all of the information necessary for its transmission and delivery such as the destination address. Each packet is forwarded based on the destination address. There is no guarantee that packets will be delivered in sequential order. Typically, flow control and error control are not provided. For this reason, the connectionless service is often referred to as a *"send and pray"* service. The IP network discussed in Chapter 4 is a connectionless packet network.

2. DIGITAL COMMUNICATIONS SYSTEM

To understand QoS metrics and network impairments that affect the user perception of QoS, we need to understand the basic components of the digital communications system. Figure 3-2 shows a block diagram of the digital communications system. To carry analog signals such as voice and video over packet networks based on the digital communications system, the analog signals must go through the following processes at the sending and receiving ends:

- Source coding/decoding
- Packetization/depacketization
- Channel coding/decoding
- Modulation/demodulation

2.1 Source coding

For digital communications over packet networks, the source analog signal must first be digitized. This process of digitizing the source signal is

referred to as source coding. At the receiving end, the reverse process, i.e., source decoding, is performed to produce the original source analog signal. The device that performs the source coding and decoding is referred to as the codec.

There are two types of source coding technique: waveform coding and linear predictive coding (LPC).

2.1.1 Waveform coding

The waveform coding is a fairly straightforward method in which the analog waveform is sampled at discrete time points and each sample is represented by binary bits.

The mathematical basis of the waveform coding is the celebrated theorem by Nyquist. According to the Nyquist theorem, if an analog signal is band-limited by B kHz, the analog waveform can be sampled at the rate twice the bandwidth B and the original analog waveform can be reconstructed "perfectly" from the discrete amplitudes so sampled.

$$\textit{Nyquist sampling rate} = 2 \times \textit{bandwidth of source analog signal} \quad (3\text{-}1)$$

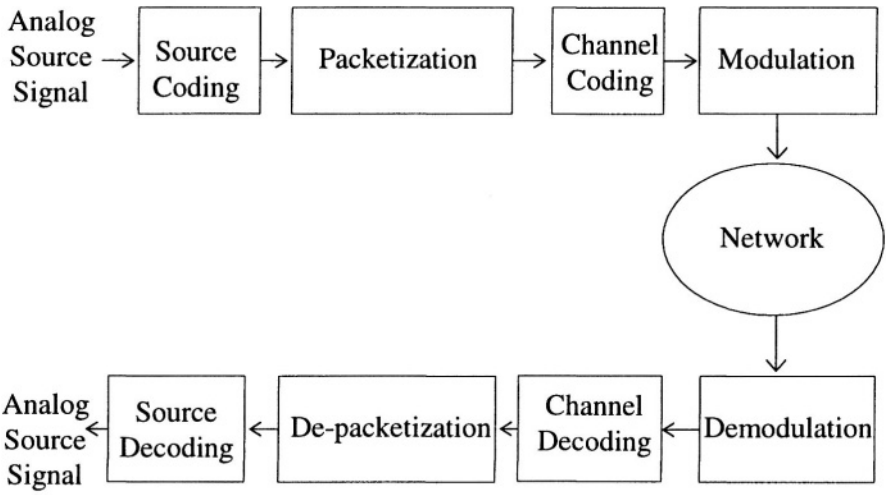


Figure 3-2. Digital communications.

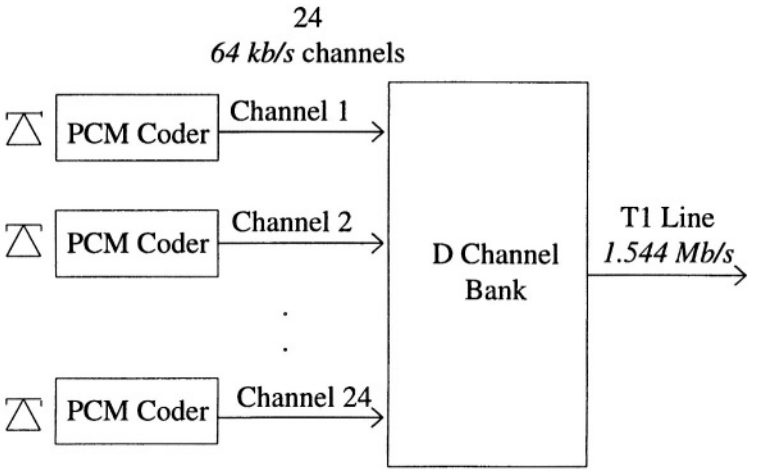


Figure 3-3. D channel bank, DS-1 rate and T-1 line.

The Pulse Code Modulation (PCM). The Pulse Code Modulation (PCM) and the Adaptive Differential Pulse Code Modulation (ADPCM) are two chief examples of waveform coding. The PCM is the first speech coder originally introduced by AT&T Bell laboratories and is most widely used. The PCM was standardized by ITU-T Recommendation G.711[4] in 1988.

Normal human speech is band-limited by *4 kHz.* In the PCM, therefore, the Nyquist theorem suggests that the source speech signal be sampled at the rate of *8 kHz,* i.e., twice the *4-kHz* bandwidth yielding 8000 samples per second. Each discrete sample is coded into eight binary bits. As a result, the PCM coding of speech produces a *64 kb/s* digital bit stream representing the source speech signal:

$$(8000\ samples/sec) \times 8\ bits/sample = 64\ kb/s \qquad (3\text{-}2)$$

The PCM has the frame length of eight bits, or one byte and the frame duration of *0.125 ms.*

The *64 kb/s* bit rate of PCM serves as the basic building block of the Time Division Multiplexing (TDM)-based digital hierarchy. For example, in North America, the D-channel bank multiplexes 24 PCM voice channels, each operating at *64 kb/s,* into the Digital Signal-1 (DS-1) frame with *8 kb/s* added per frame as framing bits as shown in Figure 3-3.

As a result, the output of the D-channel bank, i.e., the DS-1 rate, is *1.544 Mb/s*:

$$(64\ kb/s \times 24) + 8\ kb/s = 1.544\ Mb/s \tag{3-3}$$

The transmission system operating at this rate is commonly known as the "T1" line. In Europe, instead of multiplexing 24 voice channels, 30 voice channels are multiplexed. The resulting rate is higher than the T1 rate, and is referred to as the E1 rate. In waveform coding, therefore, it is convenient to choose the bit rates that are either multiples of *64 kb/s* or even sub-rates of *64 kb/s,* e.g., *32 kb/s* or *16 kb/s.*

The Adaptive Differential Pulse Code Modulation (ADPCM). Since the PCM was introduced, engineers have continually tried to reduce the bit rate to save the transmission system bandwidth. The next major waveform coding technique introduced after the PCM is the Adaptive Differential Pulse Code Modulation (ADPCM).

The basic mechanism of the ADPCM is still the same as the PCM as the last part of the name, "PCM," implies. The first two words, "adaptive" and "differential," show the modifications to the original PCM. First, consider the idea of "differential" coding. In the PCM, the absolute amplitude of each sample is coded into binary bits. In the ADPCM, at each sampling instant, t_N, the past $(N-1)$ samples are used to predict the N^{th} sample. When the actual N^{th} sample is taken at time t_N, the difference between the predicted value and the actual sample is computed:

$$a_N^* = \sum_{i=1}^{N-1} \alpha_i a_i \tag{3-4}$$

$$\Delta_N = a_N^* - a_N \tag{3-5}$$

where

a_i = amplitude of the actual i^{th} sample;
a^*_N = predicted amplitude of the N^{th} sample;
α_i = coefficients of a linear prediction algorithm;
Δ_N = difference between the predicted and actual amplitudes.

In the PCM, each absolute amplitude, i.e., a_i, is coded into binary bits. Therefore, the PCM does not need any prediction algorithm and is rather simple and straightforward. In the ADPCM, the algorithm is "differential" in that the difference between the predicted and actual sample values, i.e., Δ_i, is coded into binary bits. In the ADPCM, the prediction algorithm is "adaptive" to the actual speech samples, i.e., the last N-1 samples are used to predict the N^{th} sample.

Since the adjacent speech samples tend to be similar and their amplitudes do not vary widely, the amplitude difference between the predicted and actual samples tends to be much smaller than the absolute amplitude of the actual sample. Since a numerically smaller value would require a less number of bits to achieve a comparable quantization noise, the differential coding technique used in the ADPCM can reduce the bit rate from the PCM rate.

Like the PCM, the ADPCM samples speech at the same 8,000 samples/sec rate. However, in the ADPCM, each sample is coded into four bits instead of eight bits yielding *32 kb/s*. With the ADPCM, therefore, the bandwidth requirement is halved as compared to the PCM. This bandwidth gain is obtained at the expense of codec complexity: the ADPCM uses a prediction algorithm, which is not needed in the PCM. As electronics advances further, this tradeoff between the transmission and terminal device resources will become more significant. The ADPCM has the frame length of four bits, or ½ byte, and the frame duration of *0.125 ms*.

Table 3-1 summarizes the characteristics of the PCM, the ADPCM and the wideband codec.

2.1.2 Linear Predictive Coding (LPC)

The Linear Predictive Coding (LPC) uses a drastically different approach than waveform coding. Rather than sampling speech waveform and

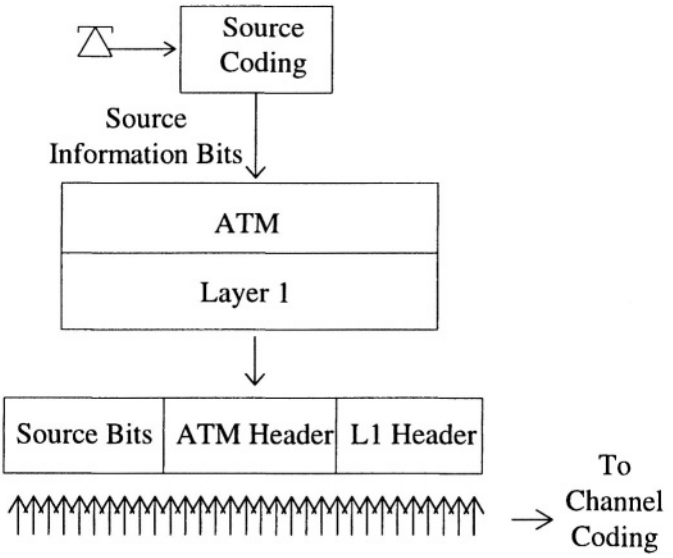


Figure 3-4. Packetization for the ATM network

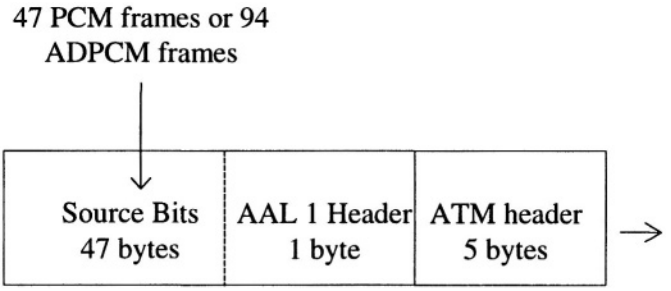


Figure 3-5. ATM cell carrying voice frames using AAL1.

reconstructing the original speech waveform from the sampled amplitudes, the LPC employs models of human vocal tract. It learns the individual speaker's vocal tract characteristics based on speech samples and estimates the model parameters. It transmits the estimated parameters to the receiver decoder. The decoder in the receiver uses the parameters received from the far end in the same internal model, which is a replica of the model used at the sending end, to synthesize the speaker's voice. The speech produced by the decoder is not the original waveform reconstructed based on its digital representation. Rather, it is speech locally manufactured based on the model parameters. It is analogous to computer voice recognition and computer-generated voice.

With the LPC, drastically lower bit rates have been accomplished. Unlike waveform coding, with the LPC, all types of odd bit rates have been introduced, e.g., *13 kb/s, 7.95 kb/s,* etc.

Table 3-1 summarizes the characteristics of some of the LPC codecs.

The Conjugate-Structure Algebraic Code –Excited Linear Predictive (CS-ACELP). A class of LPC coding techniques referred to as the Code – Excited Linear Predictive (CELP) coding uses code books to vector-quantize the excitation signal.

The CELP family of codec includes the Conjugate-Structure Algebraic Code –Excited Linear Predictive (CS-ACELP) and the LD-CELP. The CS-ACELP codec was standardized by the ITU-T G.729 Annex A[5] in 1996.

The CS-ACELP codec operates at *8 kb/s* at the sampling rate of 8000 samples/sec. Each frame consists of 80 samples yielding the frame duration of *10 ms.* When a frame of 80 speech samples is complete, the CS-ACELP coder estimates the parameters of the underlying LPC model based on the analysis of the speech frame. The coder then encodes the estimated parameters and transmits them to the receiver via subsequent processes such as channel coding, etc.

Table 3-1. ITU speech coding standards.

Standard	Bit rate	Frame size	Year finished
G.711 PCM	64 kb/s	0.125 ms	1972
G. 726 [G.721,G.723], G.727 ADPCM	16, 24, 32, 40 kb/s	0.125 ms	1990 [1988, 1988], 1990
G.722 Wideband Coder	48, 56, 64 kb/s	0.125 ms	1988
G.728 LD-CELP	16 kb/s	0.625 ms	1992, 1994
G.729 CS-ACELP	8 kb/s	10 ms	1995
G.723.1 MPC-MLQ	5.3 & 6.4 kb/s	30 ms	1995
G.729 CS-ACELP Annex A	8 kb/s	10 ms	1996

Source: Reference 6.

2.2 Packetization

The bit stream from source coding or the user computer is packetized for transmission over packet networks such as the ATM and the IP networks.

2.2.1 Voice over ATM packetization

We will use packetization of voice for the ATM network as an example. We will then use voice over ATM as a reference to discuss packetization of voice for the IP network. Chapter 6 will be devoted to the ATM QoS. However, in this section, we discuss ATM packetization briefly.

Figure 3-4 shows voice over ATM packetization. Voice frames are carried in ATM cells.

A Continuous Bit Rate (CBR) service such as voice uses the ATM Adaptation Layer 1 (AAL1). The ATM cell format has a five byte header and 48 byte payload. One byte of the 48 byte payload field is used for AAL1 header. Hence, each ATM cell has a space for 47 bytes.

Since a PCM frame is eight bits or one byte long, an ATM cell with AAL1 can carry 47 PCM frames. Similarly, an ADPCM frame is four bits, or ½ byte, long, and an ATM cell can carry 94 ADPCM frames. Encapsulation of voice frames in the ATM AAL1 cell is shown in Figure 3-5.

The overhead of ATM packetization of voice is six bytes (five byte ATM header plus one byte AAL1 header) out of 53 bytes, i.e.:

$$\textit{Overhead of voice packetization for ATM} = \frac{6\ bytes}{53\ bytes} = 11.3\% \qquad (3\text{-}6)$$

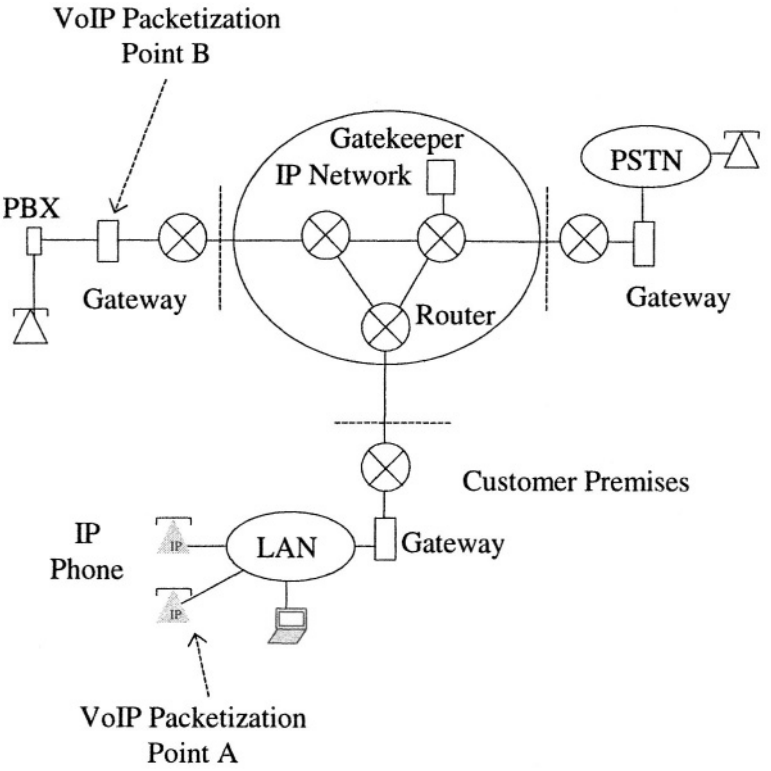


Figure 3-6. VoIP architecture.

2.2.2 Voice over IP packetization

Figure 3-6 shows a general VoIP architecture. Packetization for VoIP may be performed at the end user's telephone set if VoIP begins from the end user using an IP telephone as shown by "Packetization Point A" in the figure; VoIP packetization may also begin at a gateway device into a VoIP network as shown by "Packetization Point B" in the figure.

Figure 3-7 shows encapsulation of voice frames in IP packets using UDP. For the same number of PCM and ADPCM frames, packetization for the IP networks incur more overhead as follows:

$$\textit{Overhead of voice packetization for IPv4} = \frac{28\ \textit{bytes}}{75\ \textit{bytes}} = 37.3\% \qquad (3\text{-}7)$$

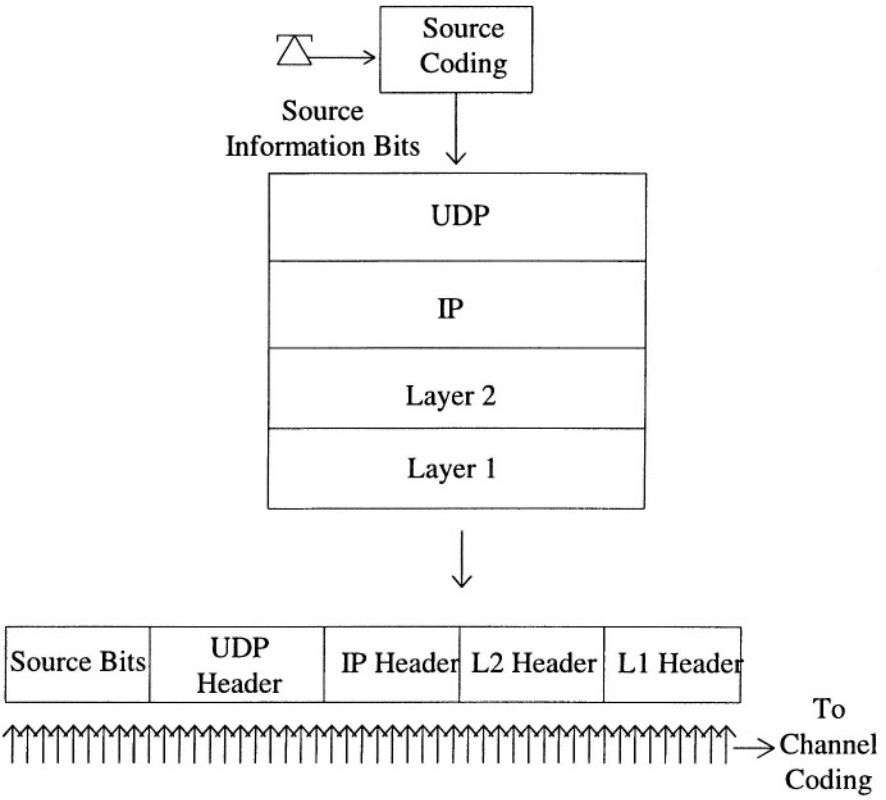


Figure 3-7. Packetization for the IP network.

$$Overhead\ of\ voice\ packetization\ for\ IPv6 = \frac{56\ bytes}{103\ bytes} = 54.4\% \quad (3\text{-}8)$$

2.3 Channel coding

After the source coding and packetization processes, we now have a stream of information carrying bits to be transmitted. However, this bit stream is not quite ready for transmission: it must go through channel coding before transmission. The main purpose of channel coding is to provide error protection and error correction to the information-carrying bit stream. Channel coding makes the bit stream more robust in the presence of transmission impairments.

Channel coding involves two processes, interleaving and error correction coding. Interleaving does not add extra bits to the information-carrying bit

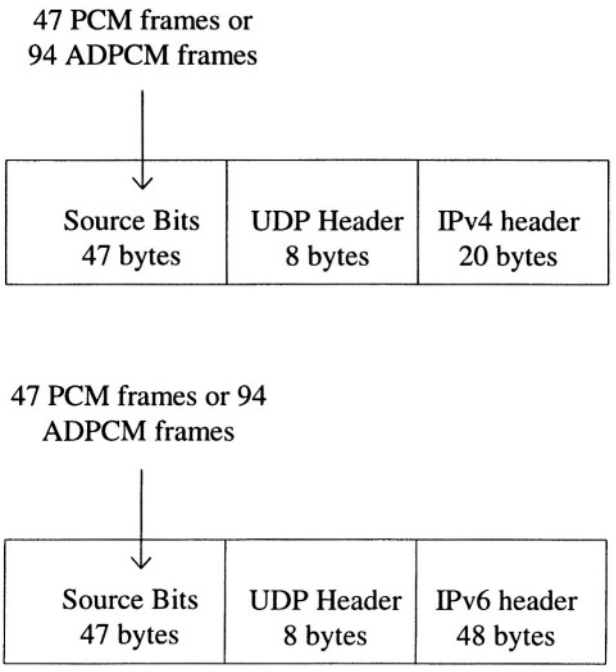


Figure 3-8. VoIP packetization.

stream. The error correction coding adds extra bits to the information-carrying bit stream. Both interleaving and error correction coding add delays.

2.3.1 Interleaving

Interleaving is one of the main techniques used to provide robustness to errors of digitized signals.

Error correction devices are designed to handle primarily random errors occurring singly. Error correction devices are less capable of handling errors occurring consecutively in bursts.

Figure 3-9 shows interleaving at the sending end. The figure illustrates interleaving in blocks of five bits. The natural order of bit transmission is shown from 1 to 25. The interleaver takes the first bit from each of the five blocks to construct the first block out into the transmission path; the second bit from each block to construct the second block out, etc.

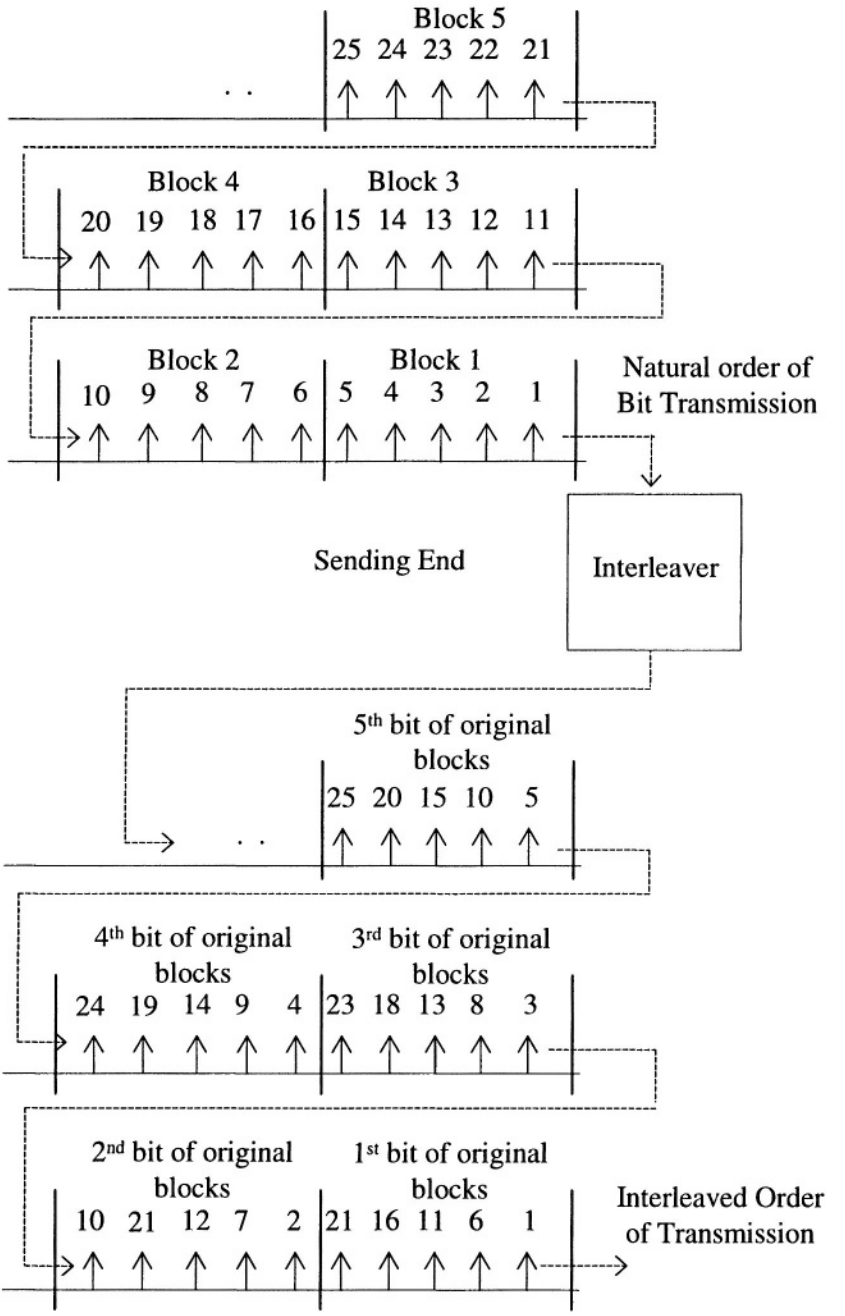


Figure 3-9. Interleaving at the sending end.

In this manner, therefore, to provide interleaving in blocks of five bits,

the interleaving cycle needs to buffer five blocks of five bits; that is, the interleaver must buffer 25 bits at the sending end, and, similarly, 25 bits at the receiving end for the de-interleaver to recover the original order of bits. In general, to provide interleaving in blocks of N bits, N blocks of N bits need to be stored in the interleaving buffer:

$$\textit{Interleaver Buffer Size} = N^2 \tag{3-9}$$

Figure 3-10 shows de-interleaving at the receiving end. The de-interleaver puts the "randomized" order of bits into the original natural order. Suppose that, while in transit, all five bits in the first block of randomized order are hit by a burst of errors as shown by "x" in the figure. Although these five errors are consecutive in transit, after the de-interleaver at the receiving end puts them in the original natural order, theses errors are separated by four intervening correct bits rendering the bursty errors to single errors.

The interleaver "reads in" the bits and stores them row by row in an NxN matrix in a buffer, and "writes out" the bits column by column from the matrix in the buffer. Figure 3-11 shows the operation of the interleaving buffer at the sending end. By storing the bits of the original order row by row and writing them out by column by column, the interleaver at the sending end "randomizes" the bits in blocks of N bits.

Figure 3-12 illustrates the reverse process at the receiving end and shows how the original order of bits is recovered.

2.3.2 Error correction

Another important function of channel coding is to provide additional coding of the source bit stream to provide forward error correction.

To illustrate forward error correction, consider one bit of the source signal. Instead of sending this one bit, the sending end may send the one desired bit plus four additional duplicated bits, a total of five bits. At the receiving end, the received bits are processed in blocks of five bits.

Taking the majority rule algorithm, each block of five bits is interpreted as one or zero depending on the number of ones and zeros in the block. This is the simplest forward error correction method. In this example, one bit is coded into five bits. The tradeoff is between the bandwidth and accuracy. The more the duplicated bits, the more accurate the error correction. There are various forward error correction coding algorithms based on algebra.

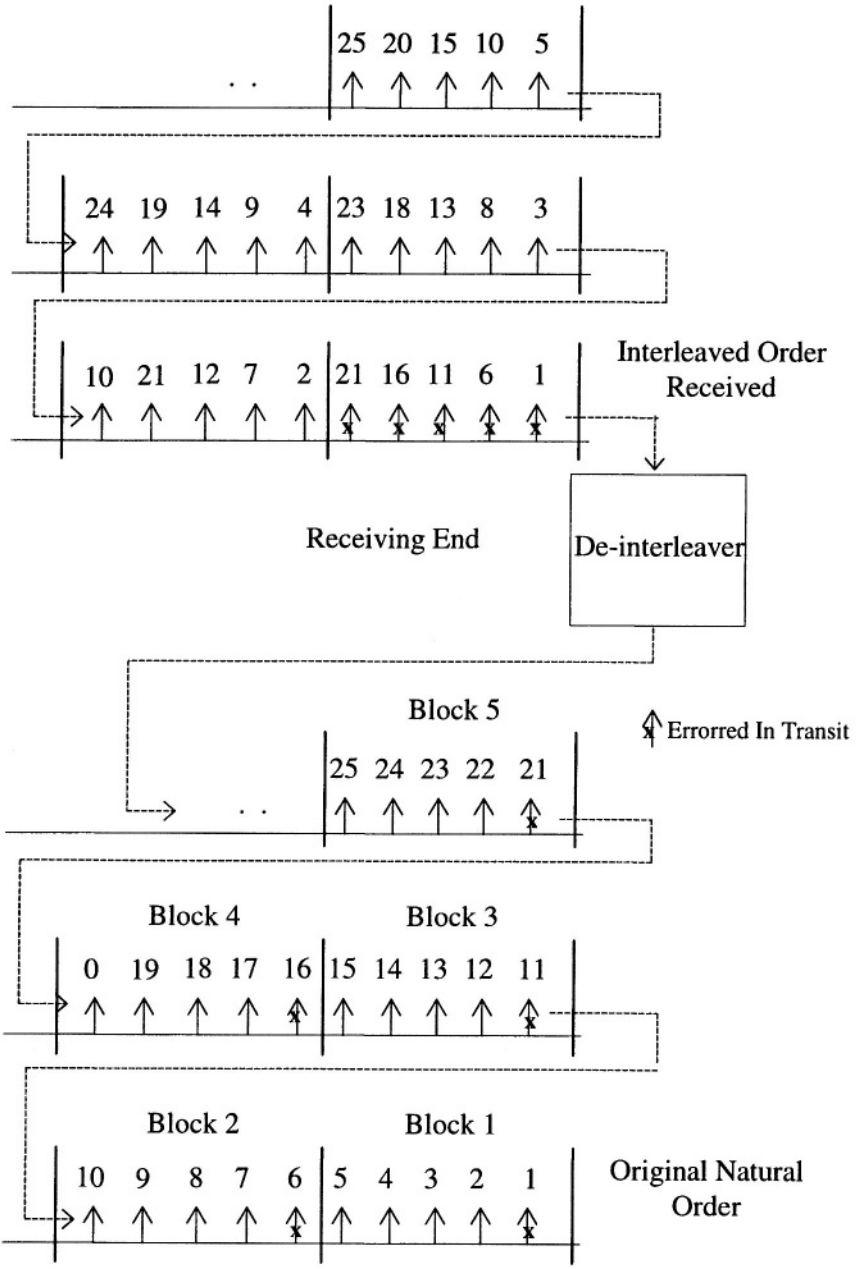


Figure 3-10. Interleaving at the receiving end.

2.3.3 Modulation

Modulation is the final step in the digital communications system before a signal is transmitted over a transmission medium. Information is represented by binary bits, i.e., 1's and 0's. These binary bits are carried from the source to the destination by a signal over a transmission medium. The signal is a sinusoidal wave and is referred to as a carrier. The information is communicated by varying the carrier as the bits change from 1 to 0 or from 0 to 1. The process of changing the carrier is modulation.

Very often, there is a misconception that a digital communications system is a system that uses a "discontinuous signal." If so, how can a communications system using a continuous sine wave as a carrier be considered a digital communications system? The following discussion can clarify this confusion. Consider a carrier or a constant sine wave referred to as a "tone." Even though a sine wave is continuously present between the source and the destination, there is no information conveyed from the source to the destination as long as the tone stays unchanged. "Communications" of information is accomplished only when the bits change and the carrier signal changes according to the modulation method used to reflect the changes in the bit stream. A digital communications system may be defined as a communications system in which communications occurs at discrete time points; that is, the bits change at discrete time points even though the carrier signal is continuous.

3. QOS OF REAL TIME SERVICES

The user's perception of QoS of real time services such as voice and video is determined by subjective testing. Each layer of the communications protocol stack introduces impairments. The factors affecting the quality of real time services over packet networks include:

- Quantization noise
- Bit error ratio
- Delay
- Delay variation or "jitter"
- Packet loss
- The choice of codec
- Echo control
- The design of the network
- Blocking probability

3.1 Quantization noise

3.1.1 Source of quantization noise

Source coding involves sampling and quantization processes. Quantization noise is associated with the latter and should not be confused with sampling noise, which is associated with the former. The Nyquist theorem assures that, if a band-limited analog source signal is sampled at a rate at least twice the bandwidth, the original analog signal can be

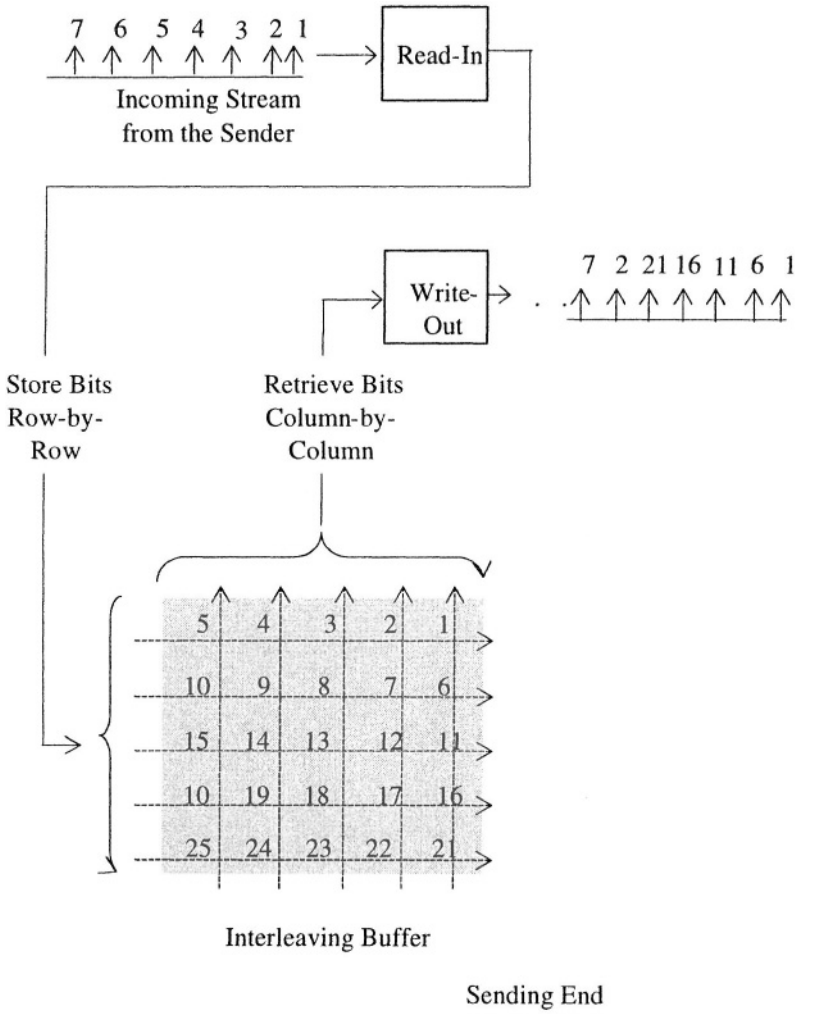


Figure 3-11. Interleaving Buffer read-in and write-out operations at the sending end.

reconstructed from the sampled amplitudes "perfectly." The key word in this theorem is the word "perfectly." For this reason, sampling process is referred to as information-preserving process and error-free process. As long as sampling is performed at the Nyquist rate, there should be no loss of information due to sampling and there is no error or "noise" introduced by sampling. Sampling error would be introduced if the sampling rate is lower than the Nyquist rate or the signal is not perfectly band-limited. Figure 3-13 shows sampling and quantization in sequence.

One condition that is tacitly implied in the Nyquist theorem is that the reconstruction of the analog signal is based on the actual sampled

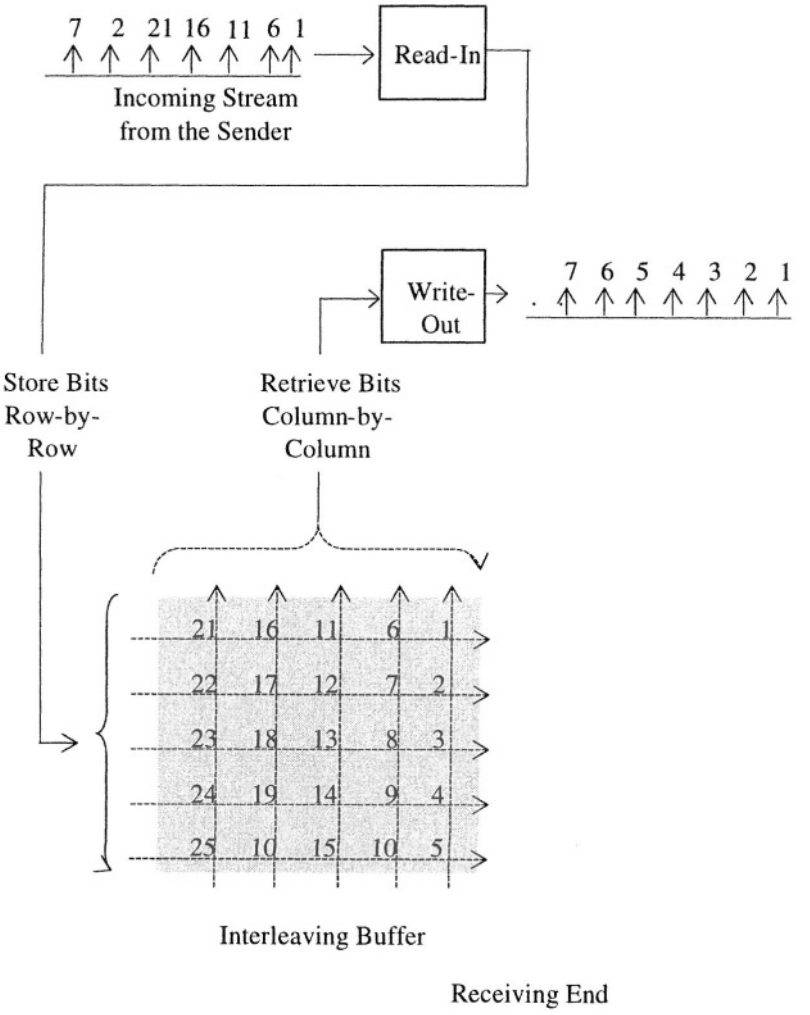


Figure 3-12. Interleaving buffer read-in and write-out operations at the receiving end.

amplitudes. Since the analog signal is continuous both in time and in amplitude, there are an infinite number of possible values of sampled amplitudes. Since, in digital communications, the sampled signal must be represented by binary bits, it is not possible to represent all possible sampled amplitudes exactly.

Quantization is a process of translating a sampled amplitude to one of a finite number of pre-determined quantized amplitudes that is closest to the actual sampled value. Figure 3-14 shows the quantization process using an example of four-level quantization. The dynamic range of the analog source signal, i.e., the interval between the minimum and maximum amplitudes, a_{min} and a_{max}, is divided into four discrete values and the sampled amplitude is approximated by one of the four discrete values. The quantized amplitudes, i.e., the outputs of the quantizer, are represented by two binary bits, 00, 01, 10, or 11.

In the PCM, eight bits are used to represent sampled amplitudes. The dynamic range of the PCM codec is therefore quantized into 256 discrete values (i.e., $2^8 = 256$). In the ADPCM, sampled amplitudes are coded by four bits and thus the dynamic range of the ADPCM codec is quantized into 16 discrete values. Notice the drastic decrease in the number of quantized values form 256 to 16

Unlike the sampling process, the quantization process introduces errors because the actual sampled amplitude is approximated by one of the finite values.

Figure 3-14 illustrates quantization error. Quantization error is the difference between the actual sampled amplitude and the quantized amplitude.

3.1.2 Effect of quantization noise

Quantization error manifests itself as noise in the reconstructed analog signal. The noise due to quantization error is referred to as quantization noise. Quantization error is one of the most basic digital impairments that is inherent to digital communications systems and is therefore unavoidable.

There is a tradeoff between the channel bandwidth and quantization error. The more bits are used to represent the sampled amplitudes, the less quantization noise is produced. The more bits are used to represent the sampled amplitudes, the higher bit transmission rate and the more bandwidth are required.

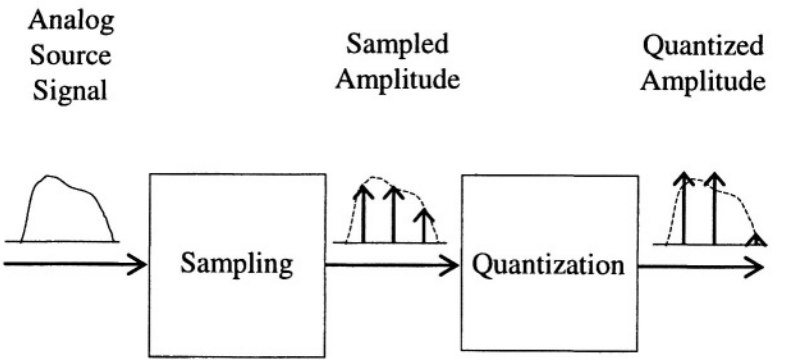


Figure 3-13. Sampling and quantization process.

Delay is the time delay incurred in the source signal, e.g., speech and video. Delay affects primarily real time applications such as voice or video. One-way delay is the amount of time measured from the moment the speaker utters a sound until the listener hears the sound. The round trip delay is the sum of the two one-way delays.

The major sources of delay include:

- Source coding
  - Delay due to A/D and D/A
  - Frame delay
- Packetization delay
- Channel coding
  - Error detection and correction
  - Interleaving
- Jitter buffer delay
- Packet queuing delay
- Propagation delay, or "speed of light" delay

The delays due to source coding, packetization, channel coding and jitter buffer are contributed at the codec; the packet queuing delay and the propagation delay are contributed by the network.

3.2.1 Frame delay

Since speech samples are processed frame-by-frame, the frame duration is a key delay component introduced by source coding. The frame duration can be calculated by the following equation:

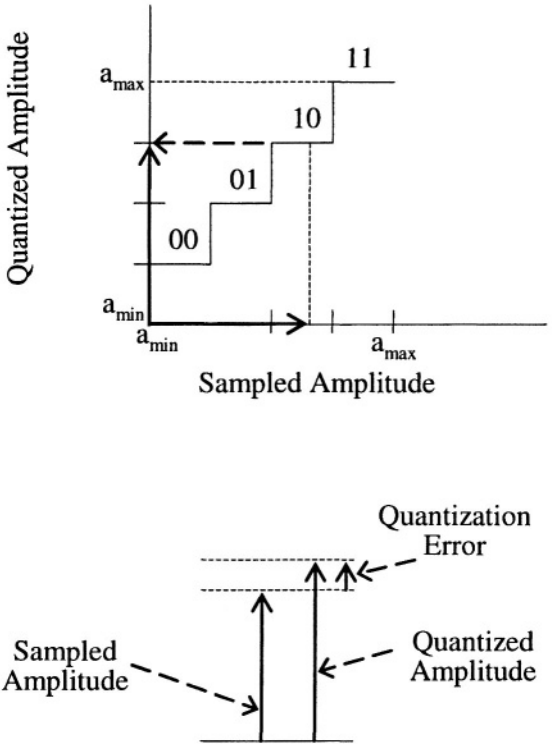


Figure 3-14. Quantization error.

$$Frame\ duration = \frac{n_s}{n_f} \tag{3-10}$$

where

$$n_s = no.of\ samples/frame \tag{3-11}$$

$$n_f = sampling\ frequency = no.of\ samples/second \tag{3-12}$$

The frame length of the PCM recommended by ITU-T G.711[4] is one sample length.

Since the sampling frequency of the PCM is *8,000 samples/second,* the frame duration of the PCM is *0.125 ms* as follows:

$$PCM\ frame\ duration = \frac{1\ sample}{8000\ samples/sec} = 0.125\ ms \qquad (3\text{-}13)$$

Since each PCM sample is coded in eight bits and each PCM frame has one sample, one PCM frame has eight bits.

The ADPCM was standardized by ITU-T Recommendation G.726[7] in 1990. The frame duration of the ADPCM codec recommended by G.726 is same as that of the PCM, i.e., *0.125 ms.* Since the ADPCM uses the same sampling frequency as the PCM, i.e., *8,000 samples/sec* and since the recommended frame duration is same as that of the PCM, an ADPCM frame contains one sample. Since an ADPCM sample is coded in four bits, an ADPCM frame has four bits.

ITU-T G.722[8] standardized the wideband codec in 1988 for three bit rates: *48, 56,* and *64 kb/s.* The frame duration of G.722 codec is *0.125 ms.* Among all of the standard codecs, ITU-T G.711 (PCM), G.726 (ADPCM) and G.722 (Wideband Codec) provide the lowest frame duration, *0.125 ms.*

The frame duration of the Conjugate-Structure Algebraic Code –Excited Linear Predictive (CS-ACELP) is *10 ms.*

3.2.2 Packetization delay

Packetization delay includes the delay due to formatting the bits into packets such as placing headers. However, the main delay of packetization comes from buffering the bits to fill the packet payload field. For example, if voice samples are carried by ATM cells, an ATM cell can be formed only when the ATM payload field is filled by the bits. Using the ATM and IP packetization example of Section 2.2, the following packetization delay can be calculated for voice over ATM:

$$ATM\ packetization\ delay = \frac{47\ bytes}{64\ kb/sec} = \frac{47 \times 8}{64}\ ms = 5.88\ ms \qquad (3\text{-}14)$$

Assuming that the main delay of packetization comes from buffering the bits to fill the packet payload field, for the same amount of payload, VoIP packetization would take about the same amount of packetization delay as that for the VoATM calculated above.

The de-packetization at the receiving end will incur about the same amount of delay. The total delay for packetization and de-packetization is, therefore, about *11.8 ms.*

3.2.3 Interleaving delay

Interleaving delay is directly proportional to the number of bits to randomize. To interleave one PCM frame, which is eight bits long, the interleaving buffer size would be 64 bits. Ignoring the interleaving process delay, this would correspond to *0.125 ms* of interleaving delay.

Example 1

Suppose that an interleaver is needed to provide burst error protection for four speech frames. Consider the ADPCM. What would be the delay introduced by the interlever?

Solution

An ADPCM frame contains one speech sample. Each ADPCM speech sample corresponds to four bits. To interleave four ADPCM speech frames, therefore, the interleaver must randomize the bits based on 16-bit blocks (i.e., *4* x *4*), and the interleaver must buffer *256* bits. Ignoring the processing time, the buffer delay alone is:

$$\frac{1\ second}{32 \times 10^3} \times 256\ bits = 8 \times 10^{-3}\ sec = 8\ ms$$

The delay due to interleaving and de-interleaving for this example is therefore *16 ms.*

To generalize the calculation of interleaving delay,

$$interleving\ block\ size = (n \times m \times l)\ bits \tag{3-15}$$

$$interleving\ buffer\ size = (n \times m \times l)^2\ bits \tag{3-16}$$

$$d = \frac{buffer\ size}{r} = \frac{(n \times m \times l)^2}{r}\ ms \tag{3-17}$$

where

n = number of speech frames interleaved
m = number of speech samples contained in one speech frame

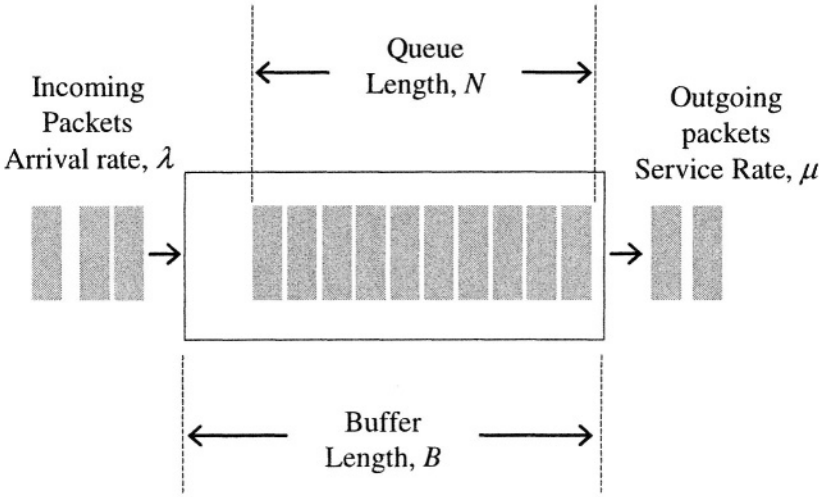


Figure 3-15. Queuing delay.

l = number of bits per sample
r = codec bit rate in *kb/s*
d = interleaving delay in *ms*

The total interleaving and de-interleaving delay is twice the delay d of Equation (3-17).

3.2.4 Error correction coding delay

The delay due to error correction coding depends on the type of coding method used. To illustrate the delay, for the block coding with five bits per information bit, the delay will be five times the bit speed.

3.2.5 Jitter buffer delay

Jitter or delay variation affects real time services such as voice and video. Jitter will be discussed further in Section 3.3. To reduce jitter, incoming packets at the receiving end are buffered in a jitter buffer and "played" out at a constant rate. About *60 ms* is a typical value for the jitter buffer delay.

3.2.6 Packet queuing delay

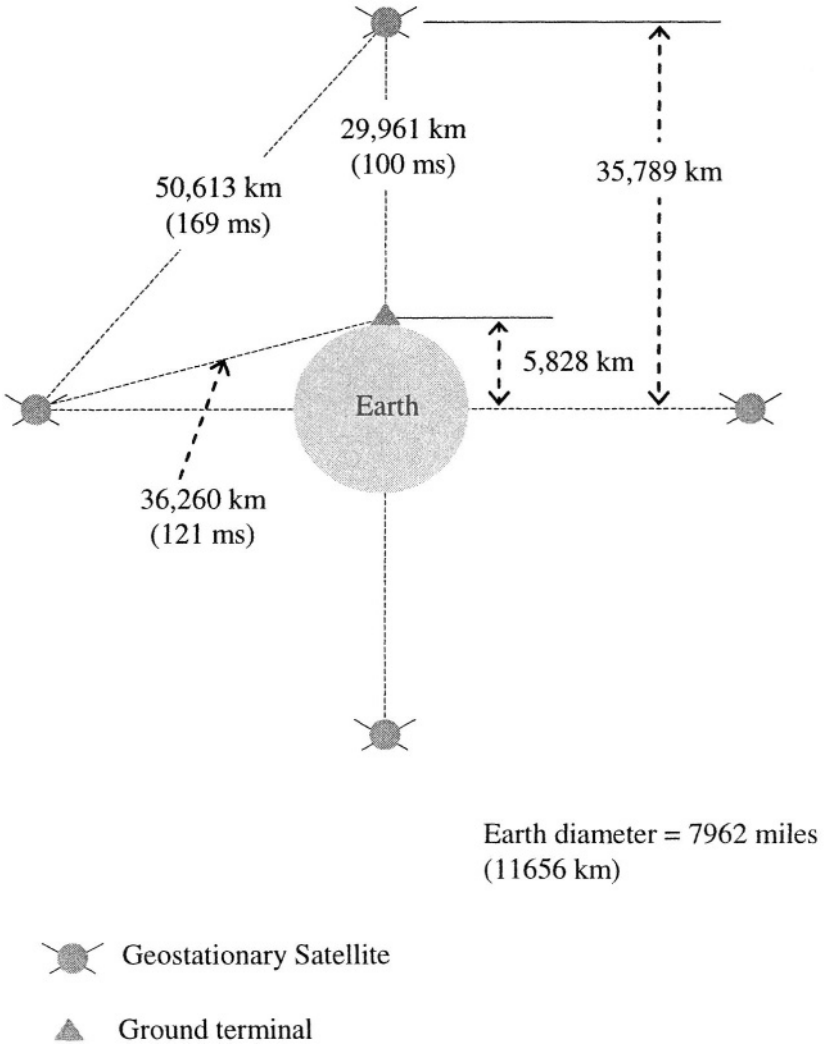


Figure 3-16. Propagation delays of geosynchronous satellites.

The delay sources discussed so far are in the end user codecs. As the packets traverse the network, additional delays are introduced: packet queuing delay and propagation delay. In packet networks, packets are queued in a buffer for processing. Queuing delay depends on the packet scheduling algorithm, the buffer size, and other factors. In Chapter 4, packet scheduling algorithms will be discussed in depth for the IP networks.

Figure 3-15 shows packet queuing delay. Using the result obtained in Chapter 2 for the *M/M/1* queue, the following result may be used as a figure

of merit for the mean packet queuing delay. Equation (3-18) applies only if $\rho < 1$.

$$\eta_d \propto \frac{1}{\mu - \lambda} = \frac{1}{\lambda} \frac{\rho}{1 - \rho} \quad if\ \rho \prec 1 \tag{3-18}$$

For $\rho \geq 1$, the mean delay is infinite: $d \rightarrow \infty, \rho \geq 1$.

Given a packet arrival rate λ, the queuing delay can be controlled by adjusting packet servicing rate μ, or ρ.

3.2.7 Propagation delay

Propagation delay is the time expended for the signal to travel the transmission distance as follows:

$$d = \frac{L}{c} \tag{3-19}$$

where L is the transmission distance in kilometers and c is the speed of light in vacuum $c = 3 \times 10^8$ *m/sec.*

This delay is sometimes referred to as the "speed of light delay." Equation (3-19) gives the "free-space" delay. The speed of light in a transmission medium is obtained by multiplying the speed of light in vacuum by a small factor specific to the medium.

The cross-continental long-haul microwave transmission distance across the United States is ~2,500 miles or 3,676 km. The one-way free space propagation delay corresponding to this distance is *12.25 ms.*

As another example, consider geostationary satellite link propagation delays. A geostationary or geosynchronous satellite rotates around the earth with the same speed as the earth's rotation speed, and, as a result, its relative position above the earth is fixed or stationary. The Newton's law on gravitational force yields a unique distance in the sky from the earth's center where the satellite's speed and the earth's speed are synchronized. That distance is *29,961 km.*

Since the relative position between the earth and the satellite is fixed, four geostationary satellites can cover the earth surface 24 hours a day. At any distance lower than this distance, the satellite must rotate at a faster speed than the earth's speed to maintain the orbit and more than four satellites would be needed to provide the 24-hour coverage of the earth surface. The Low Earth Orbit (LEO) and the Medium Earth Orbit (MEO) satellites are the satellites at the lower distances.

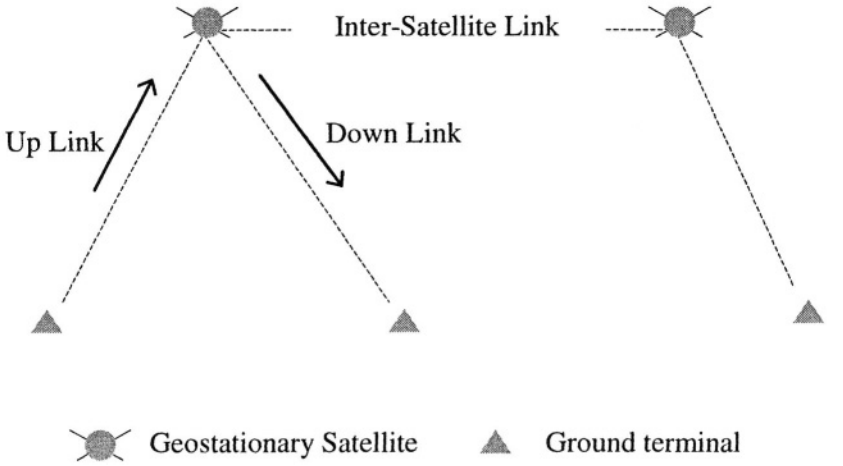


Figure 3-17. Satellites delays.

Using the geometry shown in Figure 3-16, the minimum propagation delay from the ground terminal to the geostationary satellite is about *100 ms* and the maximum delay is about *121 ms.* The inter-satellite delay is about *169 ms.* The ground terminal-to-ground terminal one-way delay is about *200 ~ 242 ms* without including the inter-satellite delay. With the inter-satellite delay, the ground terminal to ground terminal one way delay is about *369 ~ 442 ms.*

3.2.8 Effect of delay

Delay aggravates the effect of echo. Echo is the talker's own signal heard by the talker as a result of the signal reflection at the impedance mismatches of the hybrid devices that convert two-wire circuits to four-wire circuits.

As the round trip delay increases, the effect of echo becomes more pronounced and more annoying to the user. Echo cancellers are used to solve this problem.

3.2.9 End-to-end delay objectives

The following one-way end-to-end delay values have been recommended for voice by ITU-T G.114:[9]

- *0 – 150 ms*: acceptable for most user applications.

- *150 – 400 ms*: acceptable for international connections.
- *400 ms*: unacceptable for general network planning purposes; however, it is recognized that in some exceptional cases this limit will be exceeded.

3.3 Delay Variation or "Jitter"

Delay variation or "jitter" affects real time services, e.g., voice, video. Jitter is removed by a buffer in the receiving device. If jitter exceeds the size of the jitter buffer, the buffer will overflow and packet loss will occur.

3.3.1 Source of delay variation

All of the delay sources listed in Section 3.2 have a fixed value except the packet queuing delay. As packets go through the packet networks, they are queued at routers in the case of IP networks and at ATM switches in the

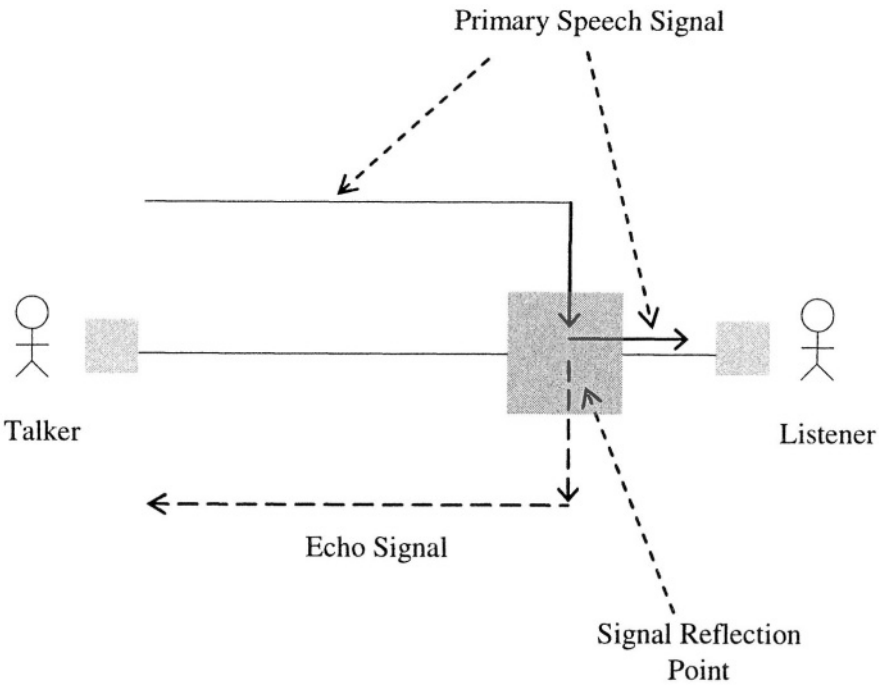


Figure 3-18. Effect of delay on echo.

case of ATM networks.

Consider a constant packet arrival rate λ and the Little's theorem Equation 92-109):

$$\eta_d = \frac{\eta_N}{\lambda} \tag{3-20}$$

The variance of d would be proportional to the variance of N:

$$\sigma_d^2 \propto \sigma_N^2 \tag{3-21}$$

Since from Equation (2-116),

$$\sigma_N^2 = \frac{\rho}{(1-\rho)^2} \tag{3-22}$$

the following measure may be used as a figure of merit for jitter:

$$\sigma_d^2 \propto \frac{\rho}{(1-\rho)^2} \tag{3-23}$$

Observe that, simply by reducing ρ from, for example, *0.8* to *0.7,* jitter can be improved dramatically.

3.4 Packet loss probability

Poisson's "law of large numbers" allows the packet loss probability to be interpreted as a long term value of the ratio of the number of lost packets to the total number of packets transmitted, and the packet loss probability is often referred to as the packet loss ratio as follows. As $n \to \infty$, the packet loss ratio:

$$\frac{k}{n} \to p \tag{3-24}$$

where

k = number of packet losses
n = total number of packets
p = constant = packet loss probability.

For real time signals, e.g., voice and video, packet loss manifests itself as noise in the decoded signal, which results in voice clippings and skips, reduced speech intelligibility, and video quality degradation.

The following are some of the main sources of packet loss:

- Bit errors due to transmission line impairments, e.g., "fading," circuit noise
- Link layer packet collisions
- Network layer processing errors
- Network layer buffer overflows
- Network layer random packet discarding

Considering the earlier results of the probability that at least k customers will be in the queue in Chapter 2, Equation (2-111) and, setting k to the buffer size B, the following equation is obtained for the packet loss ratio due to the buffer overflow in a steady state operation over a long period of time:

$$P\{N \geq B\} = \rho^B \qquad B = 0, 1, 2, 3, \ldots. \tag{3-25}$$

Observe that, as the buffer size B increases, the packet loss probability decreases; as the utilization factor ρ increases, the packet loss probability increases. Observe that, simply by increasing B from 15 to 20 for $\rho = 0.7$, the packet loss probability can be decreased from *0.005* to *0.0008*.

3.5 Subjective testing

3.5.1 Mean Opinion Score (MOS)

End-user perception of QoS is determined by subjective testing as a function of network impairments. The test conditions representative of the factors that the end-users are concerned about are presented to a sample of test subjects. For each test condition, the subjects are asked to rate it on a five-point rating scale of "excellent," "good," fair," "poor," and "unsatisfactory." Numerical scores are then assigned to the subjects' responses as follows:

Verbal rating	Numerical score
Excellent	5
Good	4
Fair	3
Poor	2
Unsatisfactory	1

MOS is the mean of the numerical scores given by the subjects and is calculated as follows:

$$MOS = \frac{(N_E \times 5) + (N_G \times 4) + (N_F \times 3) + (N_P \times 2) + (N_U \times 1)}{N} \quad (3\text{-}26)$$

where N_E, N_G, N_F, N_P and N_U are the numbers of the subjects who have rated the test conditions excellent, good, fair, poor and unsatisfactory, respectively; and N is the total number of subjects:

$$N = N_E + N_G + N_F + N_P + N_U \quad (3\text{-}27)$$

%GoB, which reads "Percent Good or Better," is the percentage of the subjects who rate the test conditions either good or excellent, that is, better than "good," and is calculated as follows:

$$\%GoB = \frac{N_E + N_G}{N} \times 100 \quad (3\text{-}28)$$

%PoW, which reads "Percent Poor or Worse," is the percentage of the subjects who rate the test conditions either poor or unsatisfactory, that is, worse than poor, and is calculated as follows:

$$\%PoW = \frac{N_P + N_U}{N} \times 100 \quad (3\text{-}29)$$

The Rhyme testing is a subjective testing used to determine speech intelligibility.

Example 2

The results of a subjective testing performed with 100 subjects are as follows:

Excellent	30
Good	20
Fair	10
Poor	20
Unsatisfactory	20

Determine the MOS, *%GoB* and *%PoW.*

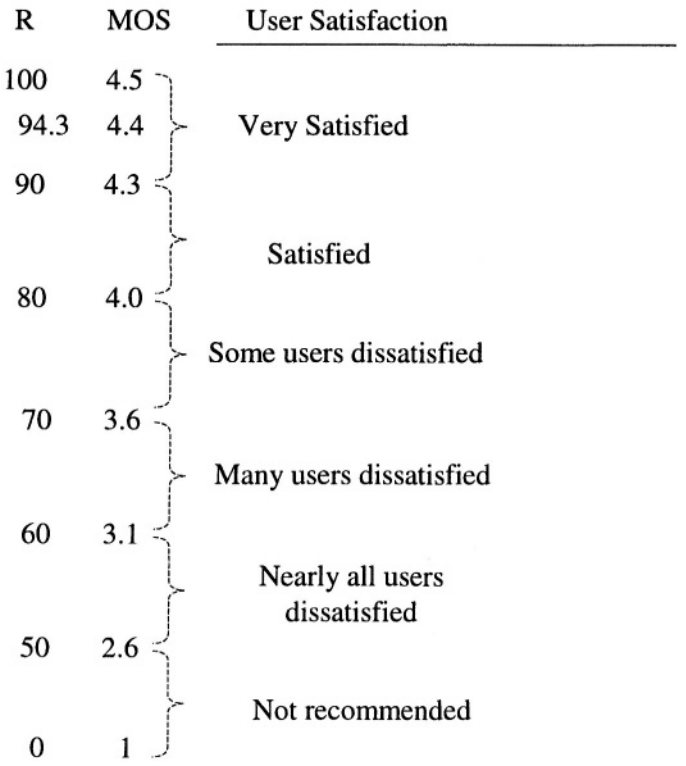


Figure 3-18. R to MOS mapping

Solution

$$MOS = \frac{(30\times 5)+(20\times 4)+(10\times 3)+(20\times 2)+(20\times 1)}{100} = 3.2$$

$$\%GoB = \frac{30+20}{100}\times 100 = 50\,\%$$

$$\%PoW = \frac{2+2}{10}\times 100 = 40\%$$

3.5.2 The Emodel

The Emodel is defined by ITU-T Recommendation G.107.[10] It is a computational model designed to produce the *MOS* without conducting subjective testing.

Subjective testing is costly and time-consuming. The Emodel is a computational model that can be substituted for subjective testing.

To use the Emodel, the effects of delay, jitter, packet loss, and other relevant impairments are combined into a single objective parameter R that ranges from *0* to *100*. For voice quality, Figure 3-19 shows the mapping of the R value to the MOS.

3.5.3 Codec performance

The performance of codec is measured by subjective testing. For waveform codecs, one of the main factors that affect the codec performance is the quantization noise.

Examples of delay budget associated with a typical ITU-T G.729 codec are shown below:[11]

Delay source	Delay budget *(ms)*
Device sample capture	0.1
Encoding delay (algorithmic+processing)	17.5
Packetization/depacketization delay	20
Move to output queue/queue delay	0.5
Access (up) Link transmission delay	10
Backbone network transmission delay	variable
Access (down) Link transmission delay	10
Input queue to application	0.5
Jitter buffer	60
Decoder processing delay	2
Device play out delay	0.5

Using the above values, the end-to-end one-way delay for a cross-continental terrestrial delay and a geostationary satellite link are:

$$12.25 + 121.1 = 133.35\ ms \prec 150\ ms\ (cf.\ ITU\ -\ T\ limit)$$

$$242 + 121.1 = 363.1\ ms \prec 400\ ms\ (cf.\ ITU\ -\ T\ limit)$$

QoS of non-real time services such as file transfer and email is determined by user requirements.

4. BLOCKING PROBABILITY

For connection-oriented packet services, the blocking probability is a key QoS measure. In this subsection, we discuss general mathematical models for determining blocking probabilities: Erlang B and Erlang C systems. Later in Chapter 6, the Erlang systems will be used in the examples of ATM virtual channel (VC) Connection Admission Control (CAC).

4.1 "Trunked Channel" systems

Figure 3-20 shows a "trunked channel" system. Connection requests (i.e., call setup requests or virtual channel requests) come to the system for a resource, i.e., communications channel to be assigned to the request. "Trunks" are shared by multiple users. Each trunk is occupied during the duration of a connection and is reassigned to another connection request when the current call is finished ("loop").

4.1.1 Offered traffic load

The amount of traffic load offered to a trunked channel system is given by the following equations:

$$L_u = \lambda H \tag{3-30}$$

$$L = L_u M \tag{3-31}$$

where
L_u = Traffic load generated by a single user
L = Total offered traffic load
λ = Arrival rate of connection requests per user
(i.e., number of connection requests placed by a user per unit time)
H = Average call duration per call, i.e., "call holding time"
M = Number of users served by the system.

4.1.2 Units of traffic load

$$ccs = \frac{s}{100} \tag{3-32}$$

where

s = number of busy seconds in one hour

ccs = "Hundred Call Second;" number of busy seconds during a one-hour period divided by 100

1 erlang = amount of traffic load that makes one trunk circuit busy for one hour.

Example 3

How many *ccs*'s is one *erlang* equal to?

Solution

By definition, one *erlang* is the amount of traffic load that makes one trunk circuit busy for one hour. Since there are 3,600 seconds in one hour,

$$\frac{60 \times 60}{100} = 36\ ccs$$

Hence,

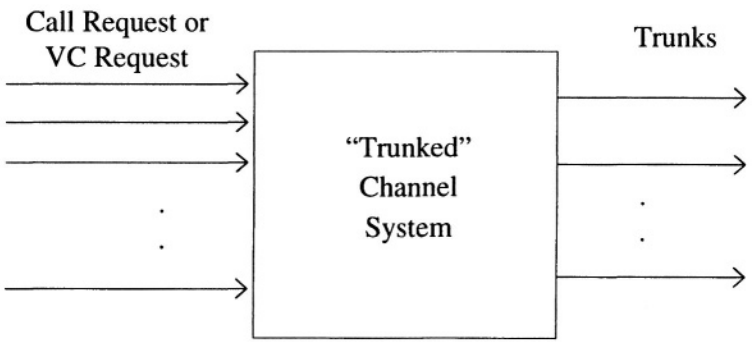


Figure 3-20. A general model of a trunked channel system.

$$1\ erlang = 36\ ccs \tag{3-33}$$

Example 4

A customer line is busy for five minutes during a one-hour period. How much traffic in *ccs* does this line generate during this one-hour period?

Solution

$$\frac{5 \times 60\ seconds}{100} = 3\ ccs$$

4.1.3 Trunk utilization factor

From the definition of utilization factor ρ introduced in Chapter 2, the utilization factor of a trunked channel system, or trunk utilization factor, can be expressed as follows:

$$\rho = \frac{L}{N} \tag{3-34}$$

where L = offered load in *erlangs* and N = number of trunks.
The following statistics provide typical busy-hour statistics:

- Loops
  - Residential *2 ccs* (*200* sec/hr, i.e., *5.6 %* of time)
  - Business *5 ccs*
  - Average *3.5 ccs*
- Trunks *20 ccs* (*67 %* utilization)

4.2 Erlang B system

Two types of trunked channel systems are discussed: the Erlang B system and the Erlang C system. Erlang B system is discussed first in this section; and Erlang C system, in Section 5.3.

The Erlang B system is based on the following assumptions:

- Call arrivals, i.e., the random arrivals at the trunked channel system, are assumed to be Poisson. This implies that the reservoir of arrivals is infinite, i.e., the number of end-users generating traffic is infinite. The Erlang B formula provides a conservative

estimate of probability of blocking because in reality the number of users is finite.

- Call holding times are exponentially distributed.
- There are a finite number of channels available in the trunking pool.
- *"Blocked calls cleared"* system. If no channel is available (i.e., "all trunks are busy") at the time of connection setup request, the connection attempt is blocked and cleared from the system, i.e., there is no queue for the blocked calls. Hence, the Erlang B system may be thought of as a queuing system with zero buffer length.

The QoS measure used for the Erlang B system is the blocking probability: a call or connection request either gets accepted or blocked. The blocking probability, P_B, of the Erlang B system is a function of two variables – the offered traffic load in *erlangs, L,* and the number of trunks, *N* – and is given by the following equation:

$$P_B = \frac{\frac{L^N}{N!}}{\sum_{k=0}^{N} \frac{L^K}{k!}} \qquad (3\text{-}35)$$

The above equation is tabulated in the Erlang B Table in terms of the following three parameters:

- Blocking probability (P_B)
- Number of channels (*N*)
- Offered load in *erlangs (L)*

Given two of the three parameters, the value of the third parameter can be found from the Erlang B table, i.e.,

- Given P_B and *L*, *N* can be found.
- Given P_B and *N, L* can be found.
- Given *L* and *N,* P_B can be found.

Example 5- Erlang B: Finding N

Assume:

- Number of user hosts, *M = 1000*

- Number of connection requests per host during busy hour, $\lambda = 2$
- Virtual connection holding time, $H = 18$ *seconds*

What is the number of virtual connections required to keep the busy hour virtual connection blocking probability P_B at 2% (i.e., *0.02*)?

Solution

$$L = l \times H \times M = 2 \times 0.3 \times 60 \times 1{,}000$$

$$= 36{,}000 \text{ busy seconds in 1 hour}$$

$$= \frac{36{,}000}{100} = 360 \text{ ccs} = 10 \text{ erlangs}$$

From the Erlang B table, the required number of channels is $N = 17$.

Example 6 - Erlang B: Finding M

Assume:

- Number of virtual connections, $N=20$
- Number of virtual connection requests per host during busy hour, $\lambda = 2$
- Virtual connection holding time, $H = 18$ *seconds*

What is the number of hosts that could be supported by the system if the required busy hour virtual connection blocking probability P_B is 2% (i.e., *0.02*)?

Solution

First find the total offered load L supportable by the system and derive the number of users corresponding to that amount of load based on call holding time and the number of calls generated by one user per unit time.

For $N = 20$ and $P_B = 0.02$, the Erlang B table shows $L = 13.2$ *erlangs.* The amount of traffic generated by a single user is given by the following:

$$L_u = 2 \times 0.3 \text{ min} = 2 \times 0.3 \text{ min} \times 60 \text{ sec}$$

$$= \frac{2 \times 0.3 \times 60}{100} = 0.36 \text{ ccs} = 0.01 \text{ erlangs}$$

The offered traffic load is determined as follows:

$$L = L_u \times M = 0.01 \times M \text{ erlangs} = 13.2 \text{ erlangs}$$

Solving for M, the number of users supported by the system $M = 1,320$.

Example 7 - Erlang B: Finding P_B

Assume:

- Number of users $M = 1200$
- Number of virtual connections, $N = 20$
- Number of virtual connection requests per host during busy hour, $\lambda = 2$
- Virtual connection holding time, $H = 18$ seconds

What is the busy hour virtual connection blocking probability P_B during the busy hour?

Solution

From the earlier problem,

$$L = \frac{18 \times 2 \times 1,200}{100 \times 36} = 12 \text{ erlangs}$$

From the Erlang B table, $P_B = 1.0\ \% = 0.01$.

4.3 Erlang C system

The same assumptions as those for the Erlang B system apply to the Erlang C system except that, in the Erlang C system, the blocked calls are queued instead of being cleared. The Erlang C system is called the "Blocked Calls Delayed" system. An example of the Erlang C system is the call attendant service, where a blocked call is put in a queue with a recoded message playing in the background, *"your call is important to us ..."* The initial blocking probability is given by the following equation, which is tabulated in mathematical tables:

$$P_B = P[delay \succ 0] = \frac{L^N}{L^N + N!(1 - \frac{L}{N})\sum_{k=0}^{N-1}\frac{L^K}{k!}} \quad (3\text{-}36)$$

In the Erlang C system, there is only delay and no blocking. The probability that delay will exceed a value t is given by the following equation, which is the product of the initial blocking probability of Equation (3-36) and the second term of the exponential function:

$$P[delay \succ t] = P[delay \succ 0]e^{\frac{(N-L)t}{H}} \qquad for\ N \succ L \quad (3\text{-}37)$$

where

L = total offered load in *erlangs*
N = number of trunked channels
t = time in seconds
H = average holding time in seconds.

The mean delay is given by the following equation:

$$\eta_D = P[delay \succ 0]\frac{H}{N - L} \qquad for\ N \succ L \quad (3\text{-}38)$$

Example 8 - Erlang C: Finding the mean delay

Assume:

- Number of users, M = 1000
- Number of call requests per host during busy hour, $\lambda = 2$
- Call holding time, H = 1.5 minutes
- Number of virtual connections = 60

What is the mean delay of call?

Solution

Total offered load

$$L = \frac{2 \times 1.5 \times 1{,}000}{100 \times 36} = 50\ erlangs$$

From the Erlang C table, $P_B = 0.1$. Hence,

$$\eta_D = P_B \frac{H}{N-L} = 0.1 \times \frac{1.5 \times 60}{60-50} = 0.9\ sec$$

5. EXERCISES

5.1 Problems

1. Assume:
   $\lambda = 5.4 \times 10^5$ packets/sec;
   $\mu = 6 \times 10^5$ packets/sec;
   $B = 50$ packets.

Determine the packet loss ratio due to the buffer overflow.

2. Responses from 200 subjects on a VoIP quality are as follows:

Rating	Number of Subjects
Excellent	20
Good	40
Fair	80
Poor	40
Unsatisfactory	20

Determine the MOS, *%GoB* and *%PoW*.

3. Using the delay budget table for G.729 modem, determine the end to end delay of connection via an airplane six *km* above ground. See Figure 3-21.

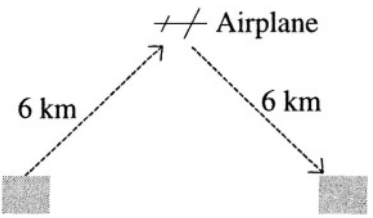


Figure 3-21. Exercise 3.

4. *Erlang B System – Finding N*
Assume:
Number of user hosts, $M = 500$
Number of virtual connection requests per host during busy hour, $\lambda = 4$
Virtual connection holding time, $H = 18$ seconds
What is the number of virtual connections required to keep the busy hour virtual connection blocking probability P_B at 1% (i.e., 0.01)?

5. *Erlang B System – Finding M* Assume:
Number of virtual connections, $N = 10$
Number of virtual connection requests per host during busy hour, $\lambda = 1.2$
Virtual connections holding time, $H = 30$ seconds
Find the number of hosts that could be supported by the system if the required busy hour virtual connections blocking probability P_B is 2% (i.e., 0.02).

6. *Erlang B System – Finding P_B*
Assume:
Number of users $M = 1400$
Number of virtual connections, $N = 20$
Number of virtual connection requests per host during busy hour, $\lambda = 3$
Virtual connection holding time, $H = 12$ seconds
What is the busy hour virtual connection blocking probability P_B during the busy hour?
From Erlang B table, $P_B = 3\%$.

7. *Erlang C Mean Delay* Assume:
Number of users, $M = 1{,}000$
Number of call requests per host during busy hour, $\lambda = 2$
Call holding time, $H = 1.5$ minutes
Number of virtual connections = 60
What is the mean delay of call?

5.2 Solutions

1. $\rho = \frac{\lambda}{\mu} = \frac{5.4 \times 10^5}{6 \times 10^5} = 0.9.$

From the packet loss ratio equation, $p = 0.005$.

2.

$$\frac{(20\times 5)+(40\times 4)+(80\times 3)+(40\times 2)+(20\times 1)}{200}=\frac{600}{200}=3$$

$$\%GoB\ \ \frac{60}{200}=30\%$$

$$\%PoW\ \ \frac{60}{200}=30\%$$

3.

$$Propagation\ delay=\frac{6\times 2}{3\times 10^{8}\times 10^{-3}}=4\times 10^{-5}\ sec$$

$$=4\times 10^{-2}\ ms=0.04\ ms\ .$$

$$End\text{ - }to\text{ - }end\ delay=121.1+0.04=4\times 10^{-2}\ ms=121.14\ ms$$

4.

$$L=4\times 18\times 500=\frac{36{,}000}{100}=360\ ccs=10\ erlangs$$

From the Erlang B table, the required number of channels, $N = 18$.

5. For $N = 10$ and $P_B = 0.02$, the total offered load that can be supported is (from Erlang B table):

   $L = 5.08$ *erlangs.*

   Let the unknown number of hosts be x. Then,

   $(1.2 \times 30 \times x)/(100 \times 36) = 5.08.$

$$\frac{1.2 \times 30 \times x}{100 \times 36} = 5.08.$$

Solving for x, $x = 508$.

6.

$$L = \frac{3 \times 12 \times 1{,}400}{100 \times 36} = 14 \text{ erlangs}$$

From Erlang B table, $P_B = 3\%$.

7. Total offered load

$$L = \frac{2 \times 1.5 \times 1{,}000}{100 \times 36} = 50 \text{ erlangs}$$

From the Erlang C table, $P_B = 0.1$.
Hence,

$$D = P_B \frac{H}{N - L} = 0.1 \times \frac{1.5 \times 60}{60 - 50} = 0.9 \text{ sec}$$

Chapter 4

IP QOS GENERIC FUNCTIONAL REQUIREMENTS

1. INTRODUCTION

Without a QoS mechanism, an IP network provides the "best effort" service. In the best effort service, all packets are indistinguishable and are given the same forwarding treatment. A QoS mechanism in the IP network provides a means of distinguishing the packets and treating them differently. Two main QoS mechanisms available for the IP network are the Integrated Services (IntServ) and the Differentiated Services (DiffServ).

Figure 4-1 illustrates the best effort service, the IntServ and the DiffServ. In this illustration, the term "traffic flow" is used in a loose sense and represents the source of traffic. In the best effort service, all packets are lumped into a single mass regardless of the source of the traffic. In IntServ, individual flows are distinguished on an end-to-end basis. In DiffServ, individual flows are not identified end-to-end. Rather, they are aggregated into a smaller number of classes. Furthermore, these classes of traffic are given differential treatment on a per hop basis and there is no end-to-end treatment of these traffic classes.

This chapter discusses the generic functional requirements in providing QoS over the IP network. The two specific IP QoS mechanisms, IntServ and DiffServ, are then discussed in Chapter 5.

To provide QoS over the IP network, the network must perform the following two basic tasks:

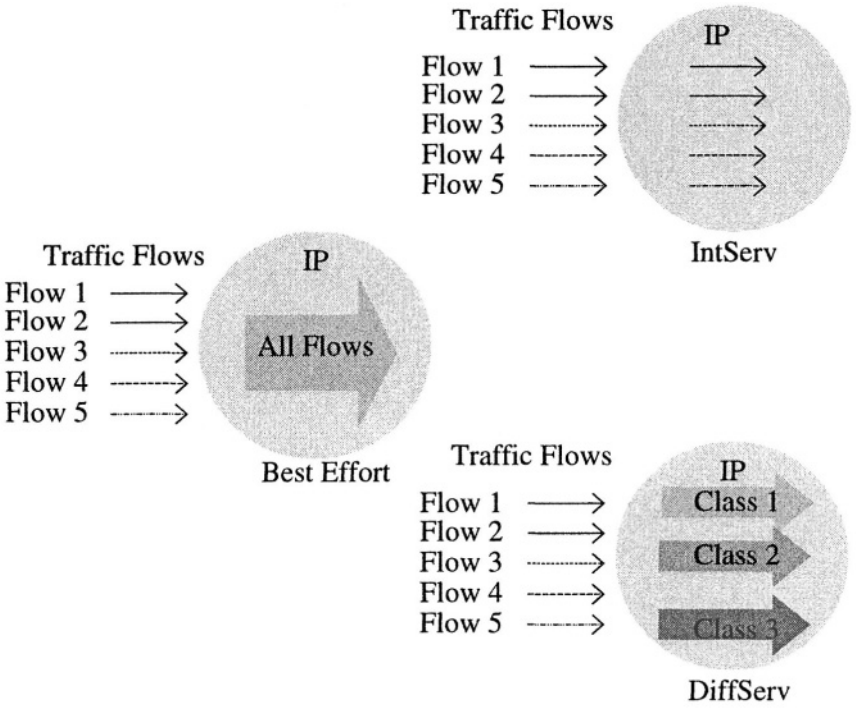


Figure 4-1. Best effort, IntServ, and DiffServ.

- Task 1 - differentiate between traffic or service types so that users can treat one or more classes of traffic differently than other types
- Task 2 - treat the different classes of traffic differently by providing resource assurance and service differentiation in a network

The generic functions required in the IP network to perform these two tasks are summarized in Figure 4-2. Task 1 is typically performed at user-network and network-network interfaces; and Task 2, by the network. Task 2 is implementation-dependent. The network capability to perform Task 2 is a key technology differentiator of manufacturers' product.

Figure 4-3 shows the requirements listed in Figure 4-2 in functional blocks in an IP router. The sequence of the functional blocks shown in the figure is applied to the pair of an input port and an output port, that is, for the packets arriving at a particular input port and leaving from a particular output port.

An incoming packet first goes through the packet marking and packet classification modules. These two modules perform Task 1 defined above.

The packet then goes through the packet conditioning and packet servicing modules. These latter two modules perform Task 2 defined above.

2. PACKET MARKING

In the general sense of the words, packet marking refers to setting the binary bits of appropriate fields in the IP header to specific values for the purpose of distinguishing one type of IP packet from another. For example, a packet may be distinguished by its source address, its destination address, or a combination of both. Another example is setting the DiffServ Code Point (DSCP) of the DiffServ field of a packet to a specific value. The DSCP marking will be discussed in detail in a later section.

An incoming packet arriving at an input port of a router is either marked or unmarked. If it is already marked, it may be remarked if, for example, the packet is subject to traffic policing and the outcome of the policing shows

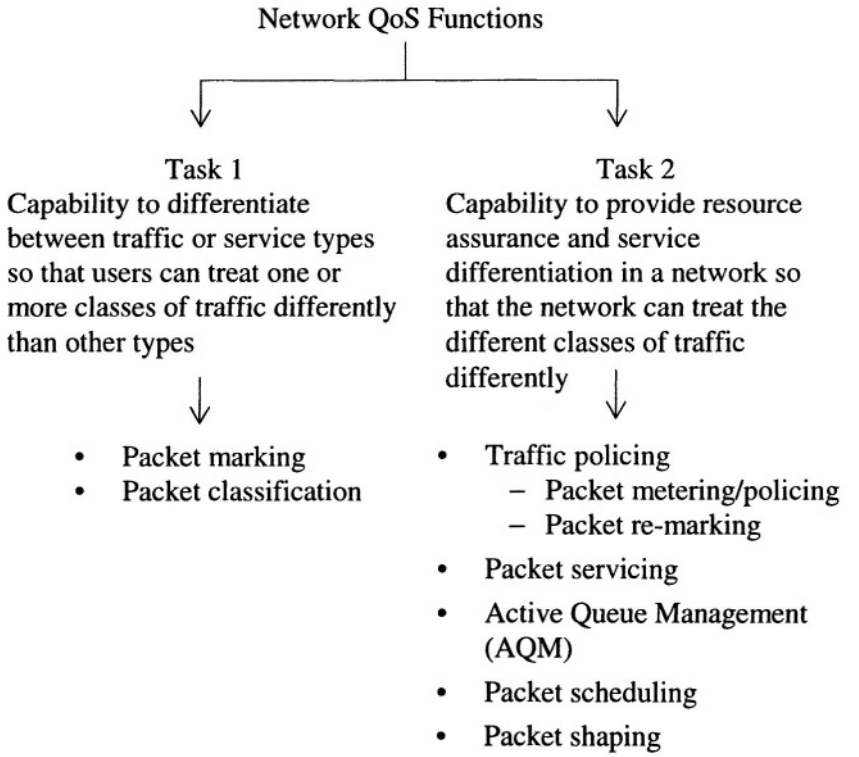


Figure 4-2. IP QoS generic functional requirements.

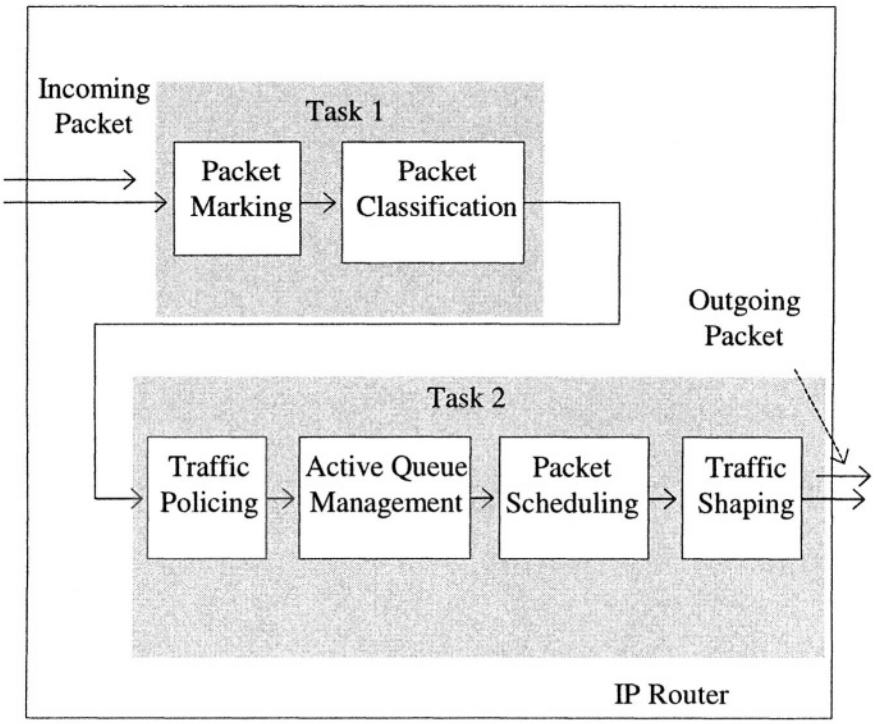


Figure 4-3. A block diagram of functional requirements in an IP router.

that the packet has violated the rule. If a packet traverses multiple DS domains, the packet marked in one DS domain may need to be re-marked as it enters another DS domain depending on the SLA between the two domains.

If a packet arrives at a router unmarked, it may be marked, if the router is the appropriate place where the packet is initially marked. The network management policy determines where the packet is marked.

3. PACKET CLASSIFICATION

Packet classification is to group packets according to a classification rule. Although packet marking and packet classification are related concepts and seem to be the same, they are quite different. The following analogy clarifies the distinction. A class of 20 students can be marked by numbers 1 through 20. This is analogous to packet marking. The students can then be classified into two groups: one group of students numbered 1 through 10,

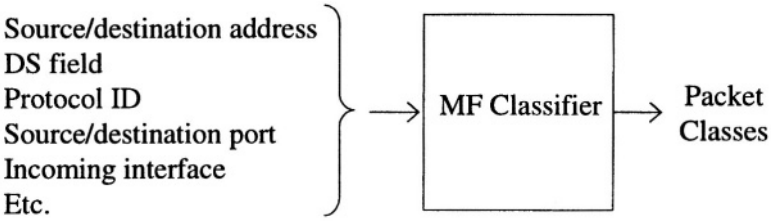


Figure 4-4. Multi-Field (MF) packet classification.

and another group of students numbered 11 through 20. This is analogous to packet classification.

The initial point of traffic classification may be at the end user. In the network, packets are selected based on the fields of the packet header that have been used for packet marking. There are two types of packet classification methods:

- Multi-Field (MF) classification
- Behavior Aggregate (BA) classification

The Multi-Field (MF) classification method is illustrated in Figure 4-4. In the MF classification method, packets are classified based on a combination of the values of one or more header fields. In addition to the header field, other parameters, e.g., incoming interface identification, may be used for classification purposes as well.

The BA classification is shown in Figure 4-5. In the BA classification, packets are classified based on the DiffServ Code Point (DSCP) values only. More will be discussed later in Chapter 5 when DiffServ will be discussed.

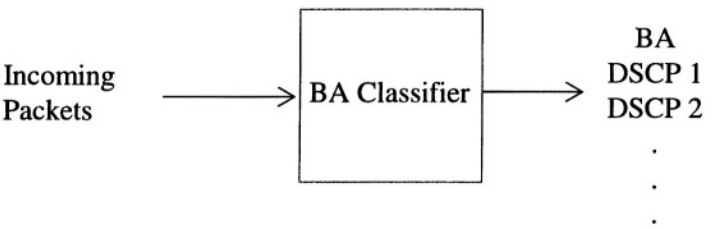


Figure 4-5. Behavior Aggregate (BA) classification.

4. TRAFFIC POLICING

Traffic policing is to check whether the incoming traffic at an input port conforms to the traffic rates that have been agreed upon between the customer and the IP network service provider. Traffic policing consists of metering the traffic according to preset traffic rates and marking or re-marking the packets based on the outcome of the metering. Packets may need to be dropped depending on the traffic policing. Figure 4-6 shows the traffic policing module and its two sub-modules, the traffic meter and the packet marker/re-marker.

Typically, traffic policing checks the rate of the incoming traffic with respect to either a single rate referred to as the Committed Information Rate (CIR) or two rates, the CIR and the Peak Information Rate (PIR). To "police" the CIR and the PIR, traffic policing uses three additional auxiliary parameters: the Peak Burst Size (PBS), the Committed Burst Size (CBS) and the Excess Burst Size (EBS).

4.1 Traffic rates

To understand the traffic rate parameters, consider the packet forwarding of an IP router for a moment as illustrated in Figure 4-7. Incoming packets arrive at the input ports of an IP router over physical transmission lines or links. The packets are "routed" to the appropriate output ports within the router based on the packets' destination addresses. Each output port is associated with, and connected to, a next router or device by an outgoing physical transmission line or link. Therefore, packet forwarding from an input port to an output port is tantamount to packet forwarding to a next hop. A packet forwarding table is pre-stored in the router, which maps an incoming packet from its input port to an output port.

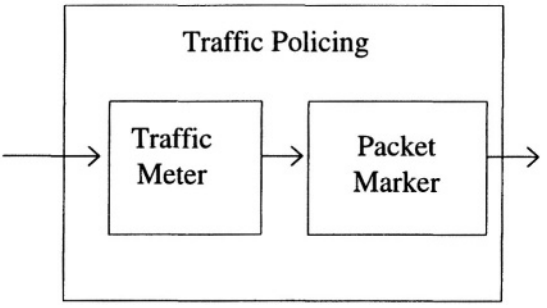


Figure 4-6. Traffic policing module.

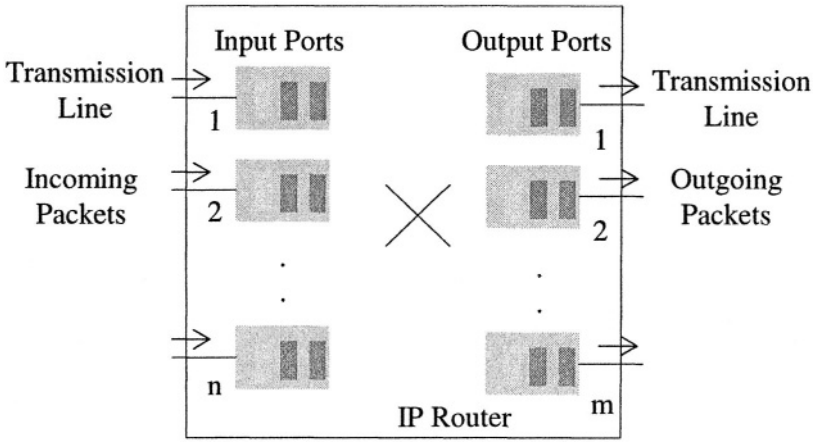


Figure 4-7. Input and output ports and the transmission lines.

4.1.1 Line rate

At an edge router, an incoming transmission line could be an end user line coming directly from the user traffic source. It could also be a line from an end user's Local Area Network (LAN) with an aggregated traffic from the LAN. At a core router, an incoming transmission line is typically a "backbone" high speed line and carries an aggregated traffic from lower speed end user lines.

Figure 4-8 illustrates traffic aggregation from lower speed end user lines

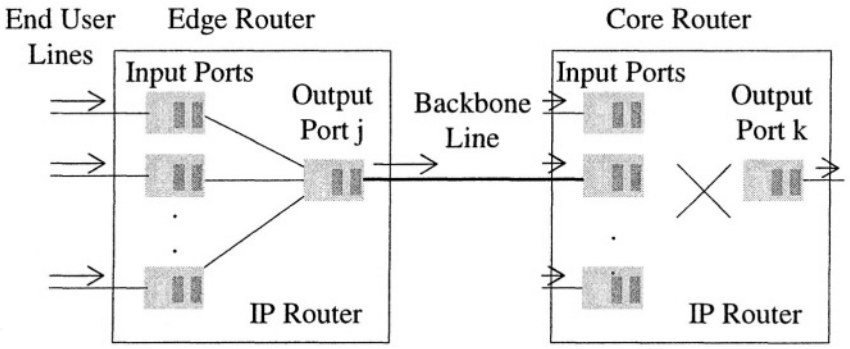


Figure 4-8. Traffic aggregation from user lines to a backbone line.

to a higher speed backbone line. In any event, the line rate at an input port places the upper limit of the incoming traffic rate at that port. The total amount of offered packet load for an output port j is the sum of the packets from all of the input ports that are forwarded to output port j as follows:

$$L_j = \sum_{i=1}^{n} L_{ij} \qquad (4\text{-}1)$$

where L_{ij} is the amount of load from input port i to output port j and L_j is the total amount of packets received at output port j. Both L_{ij} and L_j may be expressed in bytes/second. Other units are also possible.

The line rate is the bit transmission rate of a transmission line and is expressed in bits/second. Digital transmission lines are precisely "clocked" at regular intervals and bits are transmitted at these discrete time points. The time points at which bits can be transmitted are referred to as the "bit positions." The line rate is expressed in the number of bit positions per second.

The bit positions of a digital transmission system are fixed and therefore bits cannot be transmitted at a faster rate than the line rate of the system. The interval between two consecutive bit positions on a transmission line is the inverse of the line rate and is referred to as the inter-bit position time.

Digital transmission lines operate in bits. On the other hand, packets are transmitted in units of eight bits, i.e., bytes, and the packet transmission rate

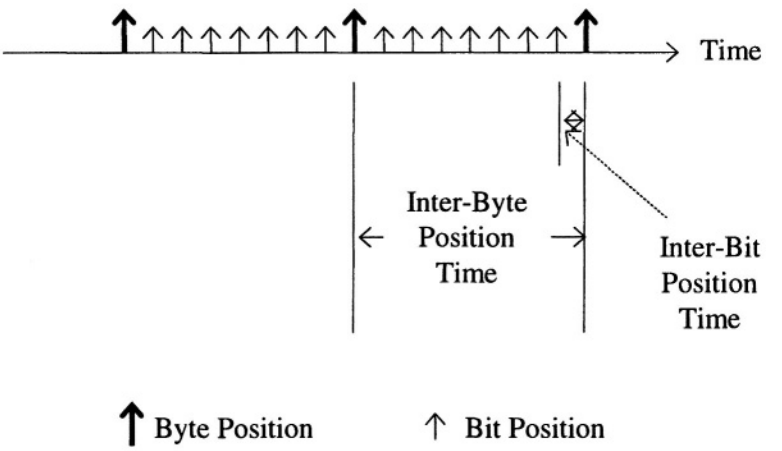


Figure 4-9. Line rate bit positions and byte positions

is expressed in bytes/second. Therefore, it is convenient to consider byte positions by normalizing the bit positions by eight bits in analyzing packet transmission. Every eighth bit position may be considered a byte position, and eight times the inter-bit interval, inter-byte time. Figure 4-9 shows the digital transmission line bit positions, the inter-bit time, the byte positions and the inter-byte time. Some examples of line rate are: ISDN B channel rate 64 kb/s, DS-1 rate 1.544 Mb/s, OC-48 rate of 2.5 Gb/s and OC-192 rate of 10 Gb/s.

4.1.2 Peak Information Rate (PIR)

The Peak Information Rate (PIR) is the maximum bit emission rate of a customer that is agreed upon with the service provider by, for example, an SLA. For a particular customer, the maximum emission rate is physically limited by the line rate of the customer. The PIR of the customer, therefore, cannot be greater than the customer's line rate. If the PIR is denoted by λ_{max}, the inverse of PIR is the theoretical minimum inter-arrival time of the packets: $1/\lambda_{max}$.

The PIR is specified in bytes per second. The PIR measures the IP packet transmission rate and so, in counting the number of bytes of an IP packet, the entire packet including the IP header is considered; the lower layer headers, e,g, Layer 2 and other physical line specific overheads, are not included in counting the rate.

4.1.3 Committed Information Rate (CIR)

The Committed Information Rate (CIR) is the "long term" average traffic rate that the network service provider is committed to honor by an agreement with the customer. The CIR is measured in bytes per second. In counting the number of bytes of an IP packet for the CIR, the entire packet including the IP header is considered. However, like the PIR, only the IP layer is considered in counting the CIR and the lower layer headers, e,g, Layer 2 and other physical line specific overheads, are not included in counting the rate.

Typically, packet transmission is a series of bursts of packets intervened by quiet intervals. While packets arrive in bursts, they are transmitted at the maximum rate, i.e., the PIR. Because of the intermittent quiet intervals, the average rate over a long period of time is less than the PIR. Hence, the CIR is in general less than the PIR.

4.1.4 Burst sizes

There are three burst size parameters used in traffic policing as auxiliary parameters: the Committed Burst Size (CBS), the Excess Burst Size (EBS) and the Peak Burst Size (PBS). The CBS is the maximum burst size that the network is committed to honor and specifies the maximum number of packets in bytes that can be transmitted by the source at the PIR while still complying with the negotiated CIR. The EBS is another threshold for a burst size that exceeds the CBS, and CBS < EBS. Packets exceeding the EBS are marked red. The CBS and the EBS are used in conjunction with the CIR. The PBS is a burst size parameter similar to the CBS, which is defined in conjunction with the PIR. More will be discussed on these burst size parameters as part of traffic metering and coloring. The CBS, EBS and PBS are expressed in bytes.

4.2 Traffic metering and coloring

There are two types of traffic metering and coloring mechanisms:

- Single Rate Three Color Marker
- Two Rate Three Color Marker

4.2.1 Single Rate Three Color Marker (srTCM)

The single rate three color marker (srTCM) is specified by RFC 2697.[13] As the name implies, the srTCM is used to police a single rate, the CIR. It meters the traffic rate and, based on the result of metering, marks or remarks the packets by three "colors" or grades. The three colors, "green," "yellow," and "red," indicate the degree of conformance in the descending order.

The srTCM has two modes of operation: color-blind mode and color-

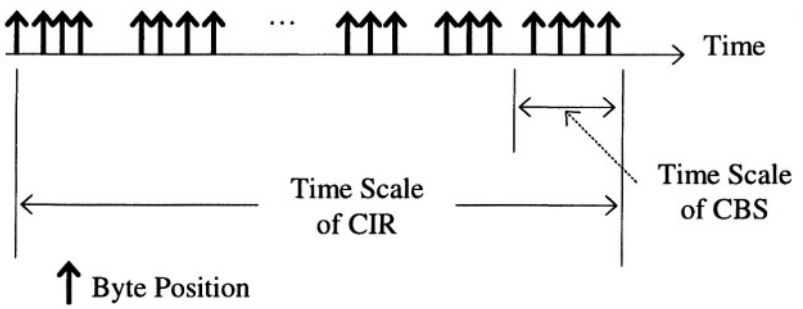


Figure 4-10. Time scales of CSB and CIR.

aware-mode. In the color-blind mode, the srTCM assumes that packets come to the meter uncolored; and in the color-aware mode, the srTCM assumes that packets come to the meter colored by some preceding entity. The srTCM is configured by setting the mode of operation and the CIR, CBS, and EBS to specific values.

The purpose of the srTCM is to ensure that the user's long term average traffic rate is within the CIR. What is the time scale of the long term average? Although not specified, it should be long enough to include the user's typical application session duration. This long term time scale is obviously not appropriate for the time scale for policing because the purpose of policing is to detect the traffic flows that violate the pre-agreed rates and mark them appropriately as they pass by: any misbehaving traffic needs to be detected and marked in "real time" since the packets move on and cannot be stored in a router for a long time until the CIR is determined based on the long term average.

Therefore, a mechanism is needed to police the CIR based on a shorter time scale. For this purpose, the two auxiliary parameters, the CBS and the EBS, are used, which are defined for shorter time scales. Figure 4-10 shows the two time scales: the CSB time scale and the CIR time scale.

The srTCM employs two types of token buckets, Token Bucket C and Token Bucket E. Figure 4-11 shows these two buckets.

The depth, or the maximum size, of Bucket C is the CBS. Bucket C is initially set full with the token count $T_c = \mathbf{CBS}$. The depth of Bucket E is the

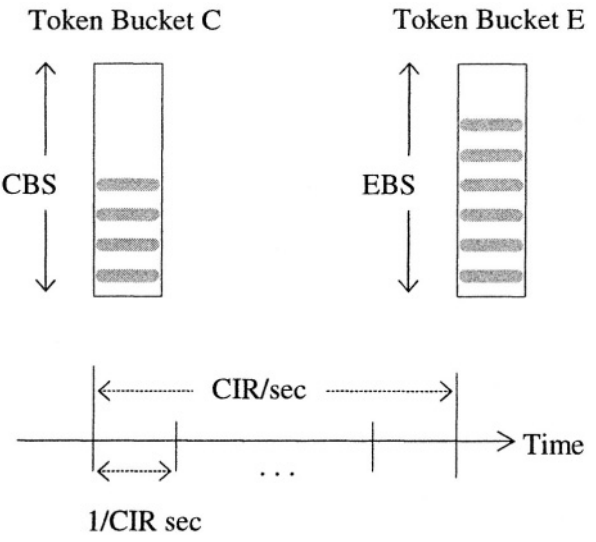


Figure 4-11. Token Buckets C and E for the srTCM.

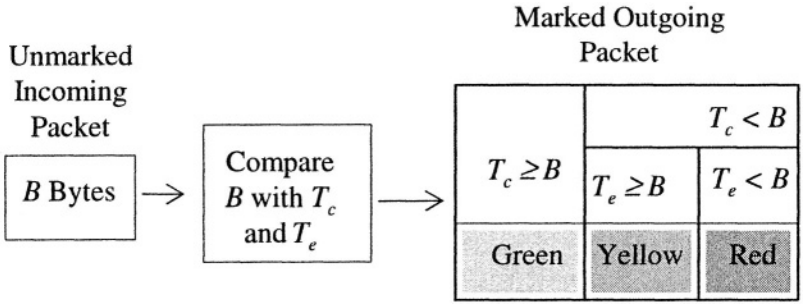


Figure 4-12. Color-blind mode operation of the srTCM.

EBS. Bucket E is initially set full with the token count T_e = EBS. Both token counts T_c and T_e are updated at the rate of the CIR, i.e., at every 1/CIR second.

The updating algorithm of the two buckets is as follows. At each update time, i.e., at 1/CIR second, if Bucket C is not full (i.e., $T_C < CBS$), T_C is incremented by one: $T_C = T_C + 1$. If Bucket C is full but Bucket E is not full (T_C = CBS and $T_e <$ EBS), T_C is unchanged and T_e is incremented by one: $T_e = T_e + 1$. If both Bucket C and Bucket E are full (T_C = CBS and T_e = EBS), both T_C and T_e are unchanged.

Figure 4-12 shows the color-blind mode of operation of the srTCM. An uncolored packet of size B bytes arrives at time t at the meter. First, the meter compares the packet size B with the current token count of Bucket C, T_c. If Bucket C has enough "credit," i.e., the packet size is within the token count, $B \leq T_c$, the packet is marked "green." T_c is then decremented by B: $T_c = T_c - B$.

If there is not enough credit in Bucket C, i.e., $B > T_c$, the meter checks the second bucket, Bucket E. If Bucket E has enough credit, i.e., $B \leq T_e$, the packet is marked "yellow," and $T_e = T_e - B$. Since Bucket C is not used in this case, T_c is left unchanged.

Finally, if Bucket E does not have enough credit either, i.e., $B > T_e$, the packet is marked "red." Since both buckets are not used in this case, both T_c and T_e are left unchanged.

Figure 4-13 shows the color-blind mode of operation of the srTCM in a more graphical way. The figure shows the packet size B with respect T_c and T_e of Bucket C and Bucket E and illustrates how the packets are marked depending on B, T_c and T_e. The figure shows that the packet "meets" Bucket C first and then Bucket E. In Case 1, B is lower than T_c in Bucket C (i.e., Bucket C has enough credit) and the packet "uses" B bytes of the credit in Bucket C. Thus, T_c is decremented by B after the packet is marked green.

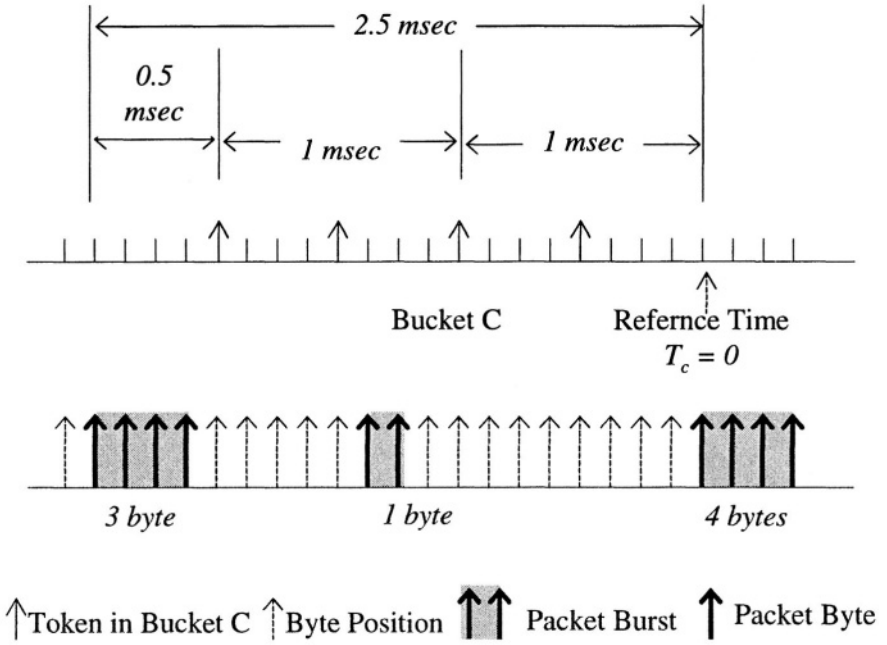


Figure 4-14. Example 2: Bursty packet stream conforming to the CIR.

In Case 2, *B* is higher than T_c of Bucket C but lower than T_e of Bucket E and the packet uses *B* bytes of Bucket *E.* Thus, T_e is decremented by *B* after the packet is marked yellow. Since the packet did not touch Bucket C, no change is made to T_c. Finally, in Case 3, *B* is lower than both T_c and T_e, and so the packet is marked red and no changes are made to T_c and T_e.

The following two examples illustrate how the srTCM makes sure that a well behaving traffic flow is assured for the CIR.

Example 1 Continuous stream conforming to the CIR

Suppose that a traffic source sends a continuous stream of packets at the CIR rate. In this case, the srTCM operation will assure that all of the packets will be marked green because, since Bucket C is filled at the CIR rate and the packets use Bucket C at a constant rate (i.e., no bursts) of the CIR, Bucket C will always have a token to send a packet marked green.

Example 2 Bursty stream conforming to the CIR

Next, take an example of bursty flow. Assume the following:

- Line rate = *64 kb/s = 8 kbytes/sec = 8 bytes/msec*
- PIR = Line rate = *8 bytes/msec = 8 byte positions/msec*
- CIR = ¼ PIR = *2 bytes/msec = 2 byte positions/msec*
- CBS = *4 bytes*

Figure 4-14 shows this example. To initialize the example, suppose that a packet burst of the size equal to the CBS, i.e., 4 bytes, has just been transmitted, depleting Bucket C and thus, at the reference time, the token count of Bucket C is 0, $T_c = 0$.

Now consider the packet transmission over the next *2.5 msec* from the reference time. As shown in the figure, two tokens are accumulated in Bucket C after *1 msec.* A little after 1 *msec* (at *1.25 msec*), a packet of one byte is transmitted spending one token. At this point, one token is left in Bucket C.

At the *2 msec* point, another two tokens are added to make the total of three tokens in Bucket C. A packet of three bytes is sent in the next *0.5 msec* interval, spending all three tokens. Since the packet sizes are all within the CBS of four bytes and since the packets are within the token counts of Bucket C, all of the packets are marked green in this example.

Now let us see what the long term average packet transmission rate is for this example. Over the period of *2.5 msec,* a total of four bytes have been

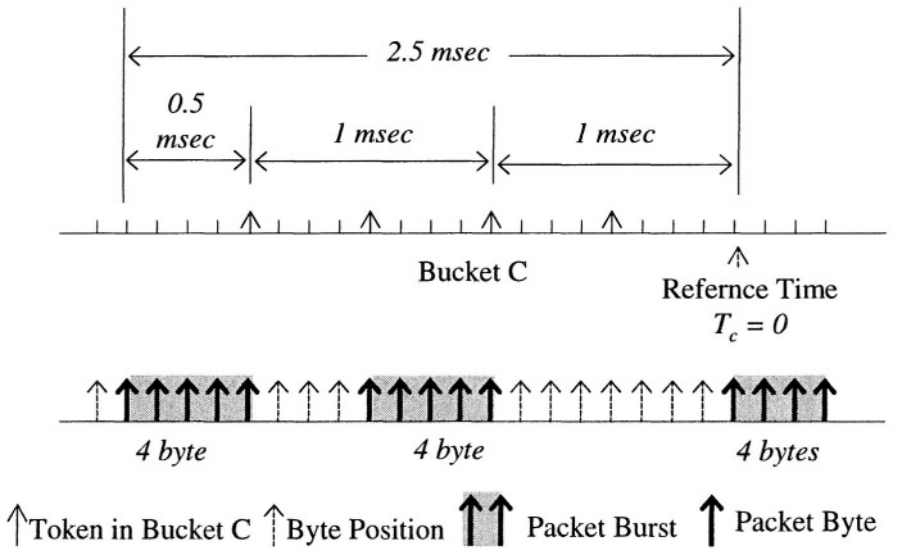


Figure 4-15. Example 3: packet stream non-conforming to the CIR.

Pre-color	Post-color		
		$T_c < B$	
	$T_c \geq B$	$T_e \geq B$	$T_e < B$
Red	Red	Red	Red
Yellow	Yellow	Yellow	Red
Green	Green	Yellow	Red

Figure 4-16. Color-aware mode of operation of the srTCM.

transmitted, yielding *1.6 kbytes/sec,* which is less than the CIR of *2 kbytes/sec,* conforming to the CIR

Example 3 Bursty stream nonconforming to the CIR

Let the two packets in Example 2 be both 4 bytes as shown in Figure 4-15, violating Bucket C. In this case, the long term average packet transmission is eight bytes in *2.5 msec,* which is *3.2 kbytes/sec.* This exceeds the CIR of *2 kbytes/sec.*

Figure 4-16 illustrates the color-aware mode of operation of the srTCM. It is similar to the color-blind mode operation. In fact, for a pre-colored green packet, the color-aware mode works the same way as the color-blind mode. A green packet of size *B* bytes arrives at time *t.* It stays green if $T_c \geq B$ and T_c is decremented by *B,* $T_c = T_c - B$, and no change is made to T_e; it is remarked yellow if $T_c < B \leq T_e$ and T_e is decremented by *B,* $T_e = T_e - B$, and no change is made to T_c; and re-marked red if $T_e < B$ and no changes are made to T_c and T_e.

In the color-aware mode, a yellow packet either stays yellow or may be re-marked red, but it can never be re-marked green. A yellow packet of size *B* stays yellow if $B \leq T_c$ or $T_c < B \leq T_e$. In the former case, T_c is decremented by *B,* $T_c = T_c - B$, and T_e is left unchanged; in the latter case, T_c is left unchanged and T_e is decremented by *B,* $T_e = T_e - B$. In this case, both token counts are left unchanged. A red packet always remains red: it can never be "promoted" to higher grades. For red packets, therefore, the token buckets are irrelevant.

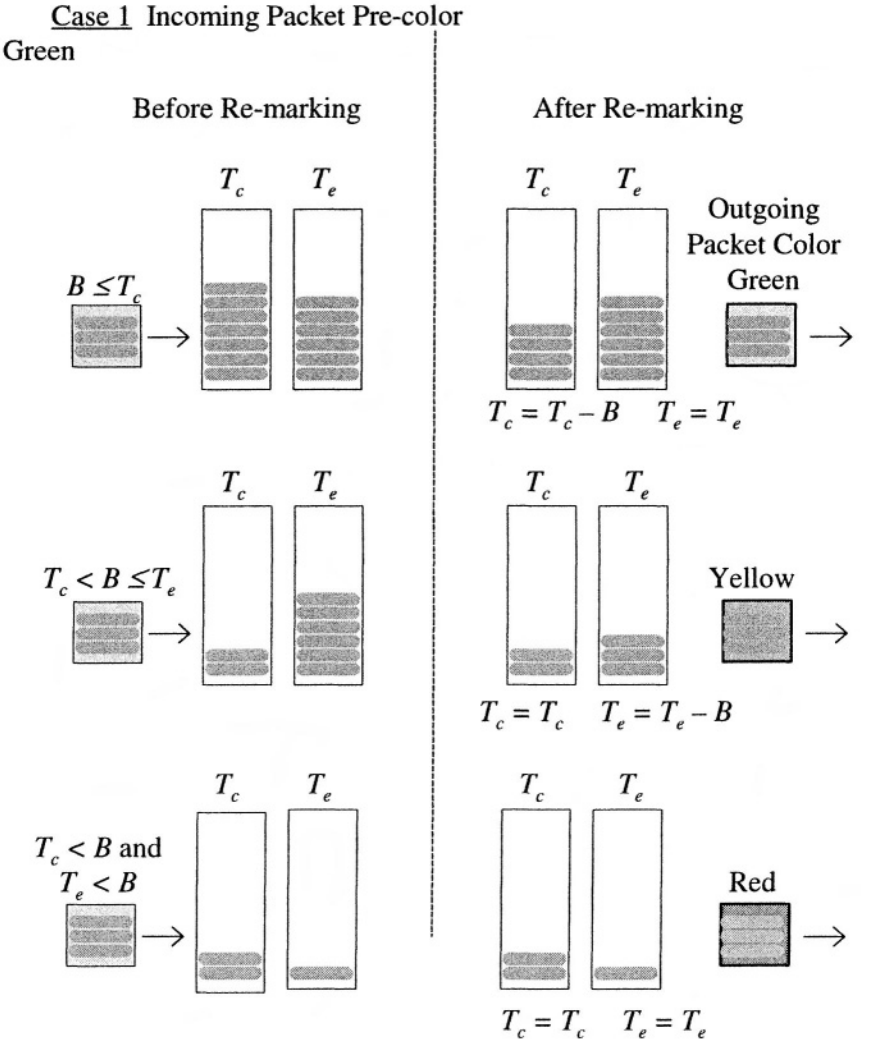


Figure 4-17. Color-aware mode of srTCM with pre-color green.

A yellow packet is re-marked (or downgraded) red if both buckets do not have enough credit, i.e., $B > T_c$ and $B > T_e$.

Figures 4-17 and 4-18 show the color-aware mode of operation of the srTCM for green and yellow packets in a more graphical way. These figures are similar to Figure 4-13 for the color-blind mode operation.

Example 4

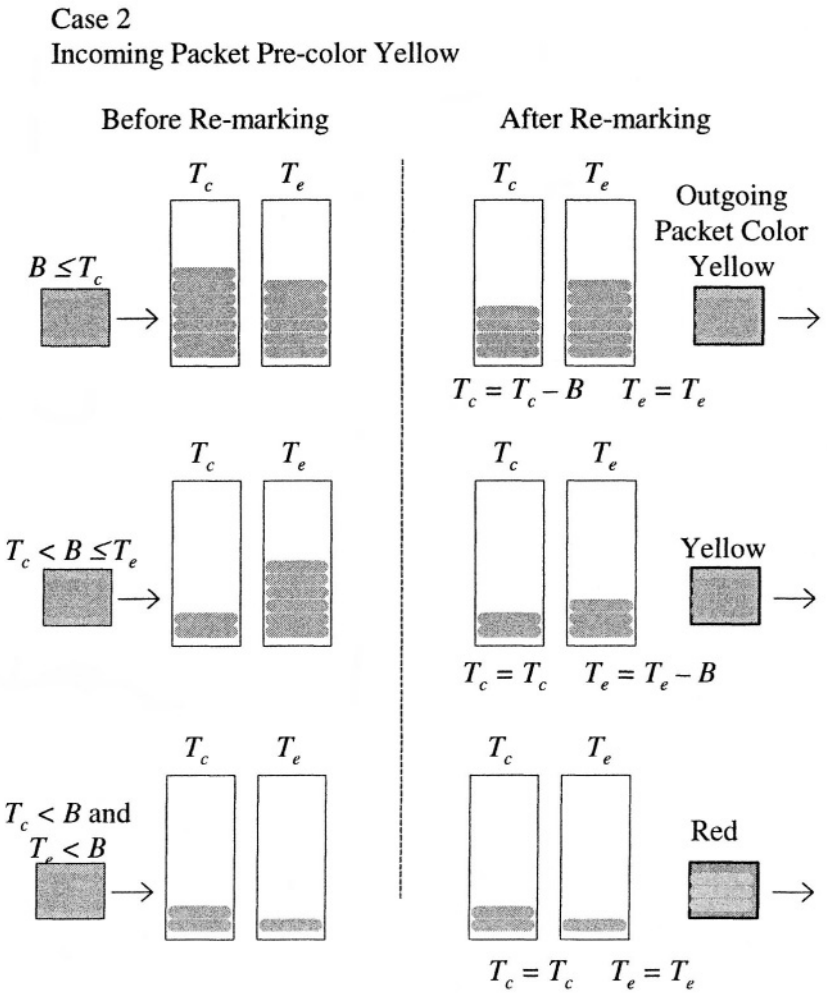


Figure 4-18. Color-aware mode of srTCM with pre-color yellow.

Referring to Figure 4-19, assume:

- Color-blind mode operation
- CIR = 1,000 bytes/sec
- CBS = 50 tokens
- EBS = 400 tokens
- At time *t*, $T_c = 10$ tokens; $T_e = 70$ tokens

At time *t*, a packet of five bytes arrives. What is the packet color and what are T_c and T_e after processing the packet?

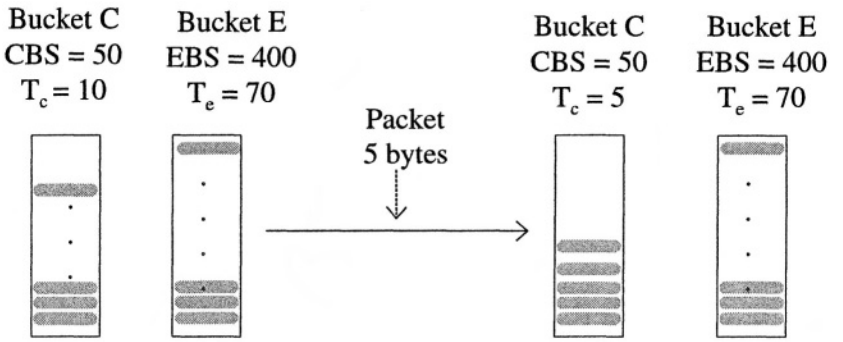


Figure 4-19. Example 4: srTCM.

Solution

Since $T_c(t) > 5$, the packet is marked "green." $T_c = T_c - B = 10 - 5 = 5$.

Example 5

Referring to Figure 4-25, assume:

- Color-blind mode
- CIR = *1,000* bytes/sec
- CBS = *50* tokens
- EBS = *400* tokens
- At time *t*, $T_c = 10$ tokens; $T_e = 70$ tokens

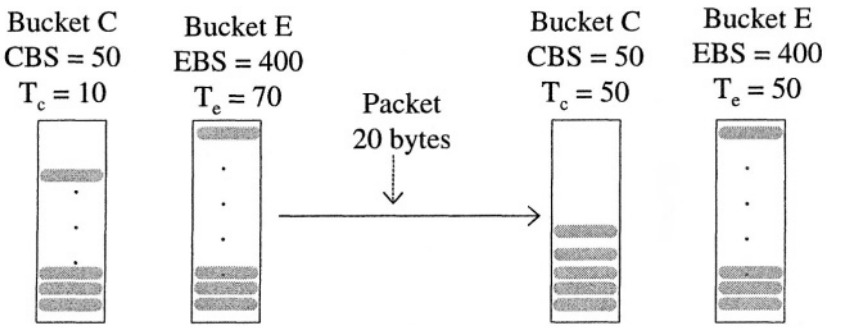


Figure 4-20. Example 5: srTCM.

At time *t*, a *20*-byte packet arrives. What are the packet color and T_c and T_e after processing the packet?

Solution

Since $T_c(t) < 20$, check $T_e(t)$. Since $T_e(t) > 20$, the packet is marked "yellow." Set $T_e = T_e - B = 70 - 20 = 50$.

Example 6

Referring to Figure 4-21, assume:

- Color-blind mode
- CIR = 100 bytes/sec
- CBS = 50 tokens
- EBS = 400 tokens

At time t, $T_c(t) = 10$ and $T_e(t) = 50$. For the next token bucket update time, the token count is updated as follows: t' = t + (1/100). For the next two seconds, no packets arrive. What are the states of Buckets C and E after two seconds?

Solution

The period of two seconds corresponds to *200* updates. Two hundred updates without spending a token are enough to fill bucket C with CBS = 50. Hence $T_c(t') = 50$. Now, determine the number of tokens needed to fill bucket C: $CBS - T_c(t) = 50 - 10 = 40$. The number of tokens left after

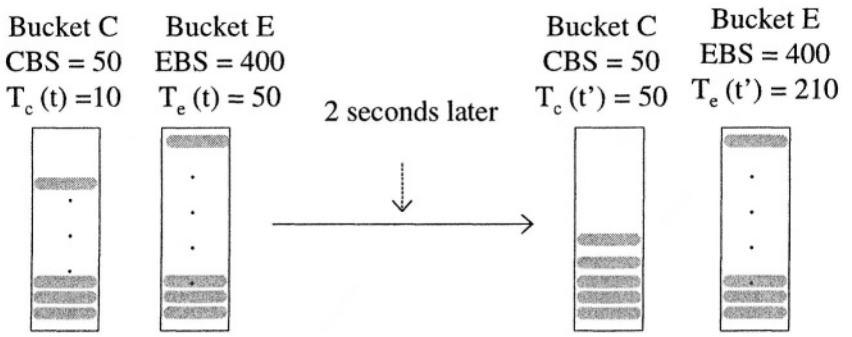


Figure 4-21. Example 6: the Single Rate Three Color marker.

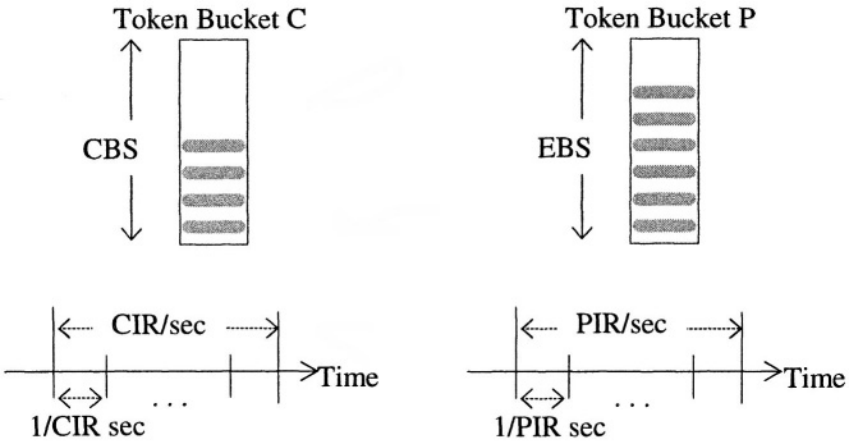


Figure 4-22. Token Bucket C and Token Bucket P for the trTCM.

filling bucket C: *200 – 40 = 160.* Hence: $T_c(t') = T_c(t) + 160 = 50 + 160 = 210$.

4.2.2 Two Rate Three Color Marker (trTCM)

The two rate three color marker (trTCM) is specified by RFC 2698.[14] The trTCM is used to police both the PIR and the CIR separately. Like the srTCM, the trTCM also has two modes of operation: color-blind mode and color-aware mode. The trTCM is configured by setting the mode and the PIR, the CIR, the PBS and the CBS to specific values.

The trTCM operates with two token buckets: Token Bucket C and Token Bucket P. Token Bucket C is used to monitor the CIR and Token Bucket P, the PIR. Figure 4-22 shows Token Bucket C and Token Bucket P for the trTCM. Token Bucket C of the trTCM is same as Token Bucket C of the srTCM. As in the srTCM, the depth of Bucket C is equal to the CBS. Its token count, T_c, is updated at the CIR rate. Token Bucket P is new with the trTCM. Its depth is equal to the PBS. Its token count, T_p, is initially set to the PBS, and is updated at the PIR rate, i.e., at every 1/PIR second.

The color-blind mode operation of the trTCM is shown in Figure 4-23. Suppose that an uncolored packet of size B bytes arrives at time t. The packet size B is compared with the token count of Bucket P, T_p, first. If Bucket P does not have enough credit, i.e., $B > T_p$, the packet is marked red regardless of Bucket C, and no changes are made to T_c and T_p.

If Bucket P has enough credit, i.e., $T_p \geq B$, the packet size B is compared with the token count of Bucket C, T_c. If, although $T_c \geq B$, Bucket C does not

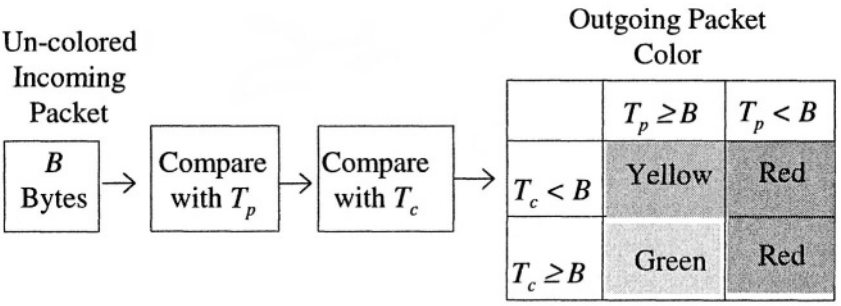


Figure 4-23. Color-blind mode of the trTCM.

have enough credit, i.e., $T_c < B$, the packet is marked yellow and $T_p = T_p - B$. If, in addition to $T_c \geq B$, Bucket P has enough credit, i.e., $T_p \geq B$, the packet is marked green and $T_p = T_p - B$ and $T_c = T_c - B$.

The color-aware mode operation of the trTCM is shown in Figure 4-24. As in the srTCM operation, packet colors never improve: they are either left the same or remarked to a lower grade. Suppose that a pre-colored packet of size *B* bytes arrives at time *t*. If the packet is red, it stays red and Buckets C and P are irrelevant. If the packet is yellow, it is re-marked red if $B \leq T_p$ and T_p is decremented by *B*: $T_p = T_p - B$. If the packet is green, it is re-marked as follows: red if $T_p < B$; yellow if $T_c < B \leq T_p$ and decrement $T_p = T_p - B$; and green if $T_c \geq B$ and $T_p \geq B$ and decrement $T_c = T_c - B$ and $T_p = T_p - B$.

Pre-color	$T_p \geq B$		$T_p < B$
	$T_c \geq B$	$T_c < B$	
Red	Red	Red	Red
Yellow	Yellow	Yellow	Red
Green	Green	Yellow	Red

Figure 4-24. Color-aware mode operation of the trTCM.

5. ACTIVE QUEUE MANAGEMENT

5.1 Tail drop method and TCP global synchronization

In the absence of an Active Queue Management mechanism in a router, the default mechanism is the tail drop method. Figure 4-25 shows the tail drop queue management. It is a passive queue management technique, whereby overflowing packets are discarded automatically when the queue is completely full. The main advantage of the tail drop method is its simplicity. However, the tail drop method causes the phenomenon referred to as the TCP global synchronization.

The TCP global synchronization occurs as follows. When the TCP sending host receives a negative acknowledgement (NAK) indicating that a TCP packet is lost while traversing a network, it assumes that the packet had been lost because of congestion in the network. To help improve the network congestion, the TCP automatically slows down the packet transmission rate.

With the tail drop method, when the buffer is full, all packets arriving at the buffer are dropped indiscriminately. If these packets are TCP packets, all the TCP sessions associated with the dropped packets would slow down simultaneously and come back with transmission at about the same time.

Since the affected TCP sessions react in a synchronized manner, the congested situation tends to go into oscillation between peaked congestion and off-congestion. This phenomenon is referred to as a "global TCP synchronization" and causes inefficient utilization of the network resources, e.g., the output port bandwidth, buffer space.

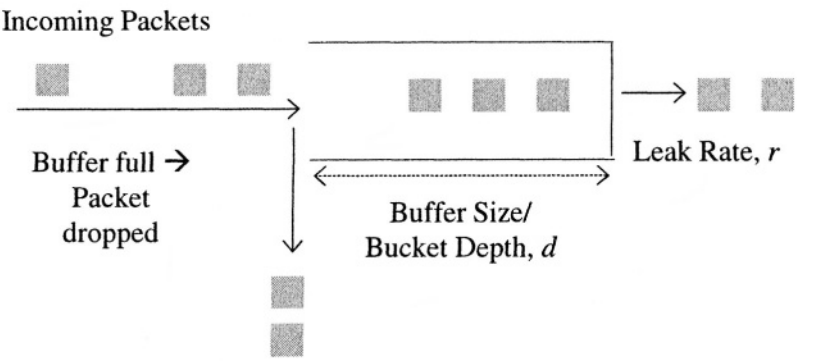


Figure 4-25. Tail drop queue management.

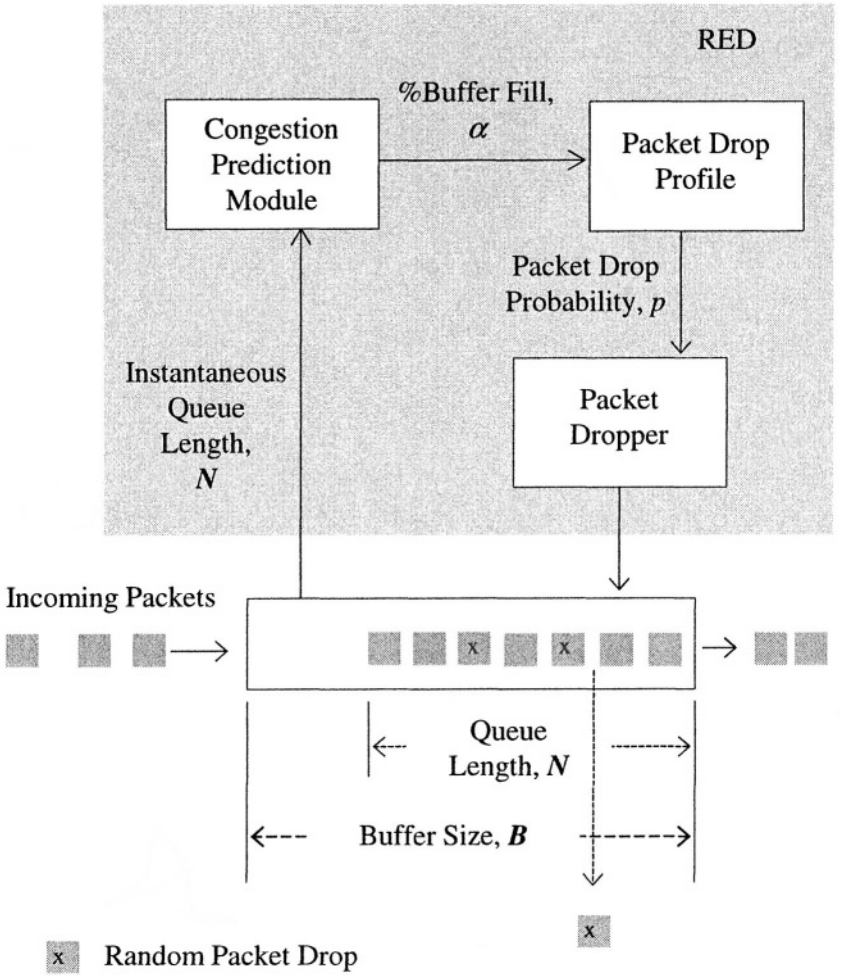


Figure 4-26. RED operation.

The Active Queue Management (AQM) is a congestion control mechanism and its main objective is to prevent the TCP synchronization. The main idea of the AQM is to anticipate an onset of congestion and take action to prevent or mitigate the effect of the congestion. The following three AQM methods are discussed in this section:

- Random Early Discarding (RED)
- Weighted Random Early Discarding (WRED)
- Explicit Congestion Notification (ECN)

The RED and the WRED involve taking action on the queue dropping packets and do not directly involve the end user hosts; the ECN takes a fundamentally different approach that involves direct participation of the end user hosts.

5.2 Random Early Discarding (RED)

The RED detects an onset of congestion and randomly drops packets from the buffer. Figure 4-26 presents a conceptual diagram that shows how the Random Early Discarding AQM operates. As shown in the figure, the RED employs a congestion prediction algorithm and a packet drop profile as the central components.

The main function of the congestion prediction module is to assess how the traffic in the buffer behaves over time and detect any buildup of congestion. As an analogy, the congestion prediction task is similar to predicting how a stock market behaves. The simplest approach would be to look at the instantaneous queue length, N, and determine the state of congestion based on how full the queue is as compared to the buffer size, B, very much like making a stock trading decision on the basis of the stock's current price.

A more sophisticated congestion prediction algorithm involves calculating a weighted time average of the queue length, very much like using a time series analysis for the stock market. The output of the congestion prediction module is the weighted mean queue length, η_N. Although it reflects the current queue length, η_N is not the actual queue length at the moment unless the default method discussed above is used. In this sense, η_N can be regarded as a measure of congestion buildup. Define the percentage fill of the buffer denoted by α as follows:

$$\%buffer\ fill \quad \alpha = \frac{\eta_N}{B} \tag{4-2}$$

where η_N is the weighted mean queue length and B is the buffer size. Just as η_N is in general a long term measure of the queue length, α is a long term measure of the buffer fill and not necessarily an instantaneous buffer fill unless the default method is used.

The next module in the RED is a packet drop profile. Given the calculated measure of congestion buildup expressed in percent buffer fill, α, a packet drop probability is determined based on the "packet drop profile." Figure 4-27 illustrates the concept of a packet drop profile. The abscissa is the percent buffer fill, α, and the ordinate is the packet drop probability, p. Packets are not dropped as long as α stays below a pre-assigned minimum

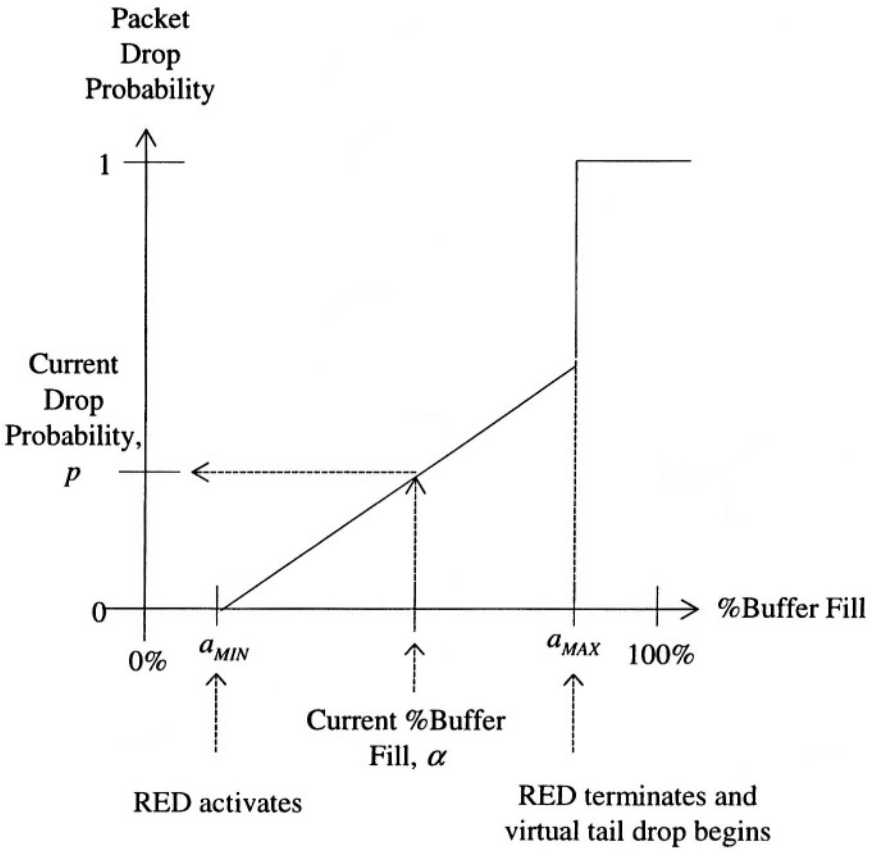


Figure 4-27. RED Packet Drop Profile.

value, α_{MIN}. The RED is activated as α exceeds α_{MIN}. If α exceeds beyond α_{MAX}, the queue begins to operate in a virtual tail drop mode. The tail drop mode is "virtual" in the sense that the queue behaves as though the buffer is full even though the buffer may still have room because α is not the actual percent buffer fill as defined above. For α between α_{MIN} and α_{MAX}, the packet drop probability, *p,* is determined by a pre-defined function such as a linear function shown in the figure. Although the function shown in the figure is linear, in general, it can be in any form.

The packet drop profile is pre-configured by specifying three factors: α_{MIN} and α_{MAX}, and drop probability function $p = f(\alpha)$. The degree of aggressiveness of packet dropping can be controlled by these three factors. The lower the α_{MIN} is set, the more aggressive the RED is; the lower the

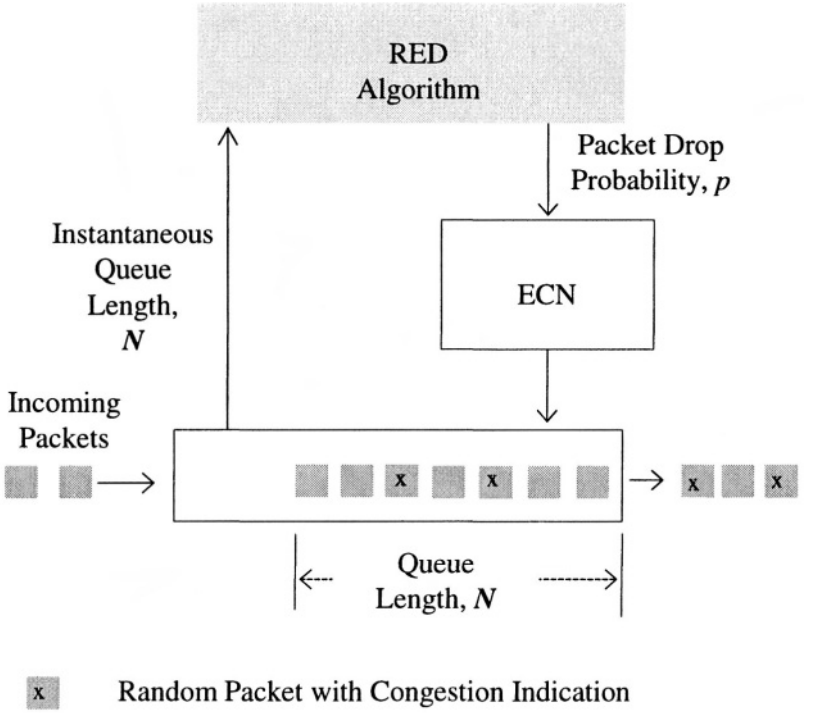


Figure 4-28. Concept of the ECN.

α_{MAX}, the more aggressive; and the steeper the slope of $f(\alpha)$, the more aggressive.

The main challenge of the RED is developing algorithms. This involves developing a mathematical algorithm for calculating α and setting a drop profile to predict a congestion buildup accurately.

By randomly dropping packets ahead of congestion, the RED prevents the TCP global synchronization. Since the UDP packets are oblivious to packet dropping, the RED has no effect on UDP traffic flows. In fact, if the traffic consists of UDP only or at least a majority of UDP, the RED should not be used because packet dropping would have no effect on UDP traffic and would waste packets unnecessarily.

Use of the RED requires care in configuring the RED factors and normally requires accumulated experience with regard to the traffic behavior of the network for which the RED is being considered. As one can imagine, a wrong RED algorithm for α and a wrong drop profile configuration that is too aggressive in packet dropping would lead to excessive dropping of

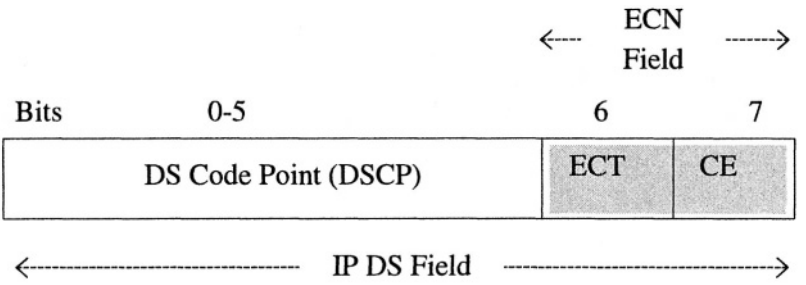


Figure 4-29. DS field.

packets, unnecessarily wasting network resources. The situation would be similar to a wrong prediction of a bull market.

5.3 Weighted Random Early Discarding (WRED)

The Weighted Random Early Discarding (WRED) is the RED with multiple drop profiles. In the WRED, rather than using a single drop profile for all queues, different drop profiles may be defined for individual queues. In addition, multiple drop profiles may be defined within a single queue.

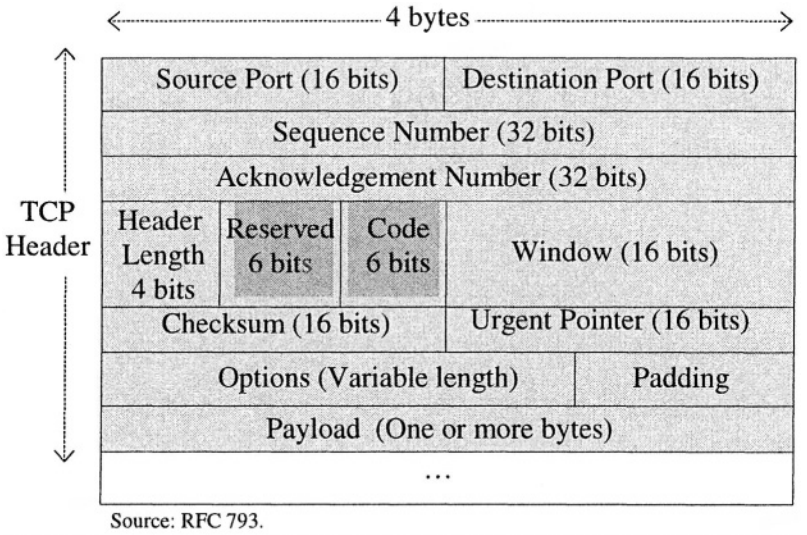


Figure 4-30. TCP header.

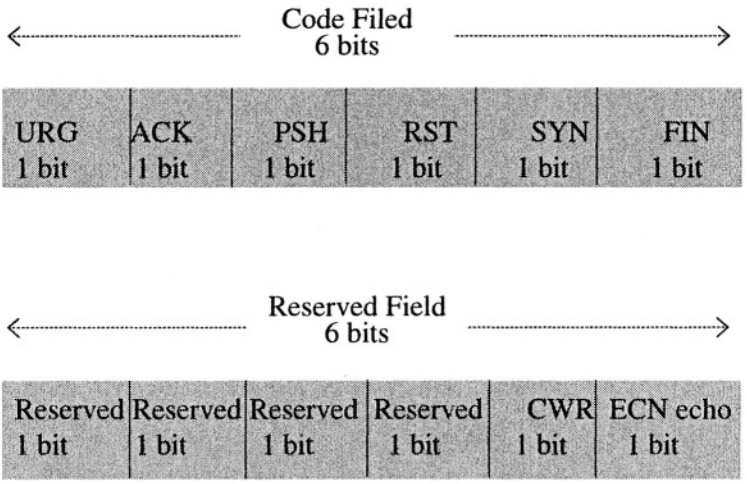


Figure 4-31. The code field and the reserved field of the TCP header.

For example, three different drop profiles may be used for the three different colors of packets: the most aggressive profile for red packets, the least aggressive drop profile for green packets, etc. Various combinations of drop profiles may be possible.

5.4 Explicit Congestion Notification (ECN)

5.4.1 General concept

The Explicit Congestion Notification (ECN) method is a congestion control method applied to the TCP traffic. The ECN method was proposed in 1999 in RFC 2481[15] as an experimental addition to the IP architecture. The ECN method is a fundamentally different approach from the RED and the WRED.

Figure 4-28 illustrates the ECN method. In the ECN, an onset of congestion is communicated to the end systems by marking the appropriate fields in the TCP and IP headers with congestion indication rather than dropping packets. Hence, the main difference between the RED and the ECN is that, in the RED, packets are randomly dropped, whereas, in the ECN, the randomly selected packets are allowed to pass (i.e., not dropped) with the congestion indication mark to notify the end systems. The same algorithm that the RED uses is used to select the packets for congestion indication.

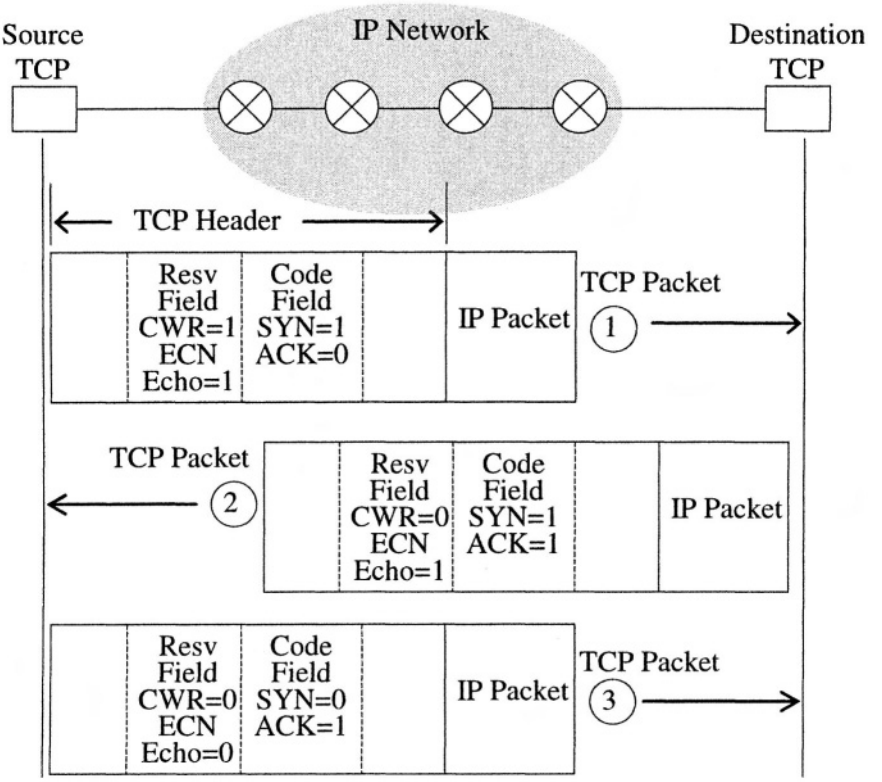


Figure 4-32. ECN handshaking by TCP hosts.

5.4.2 ECN marking in the IP header

The ECN requires making both the IP and the TCP headers. The ECN makes use of two reserved bits in the TCP header and two reserved bits in the IP header. The last two reserved bits in the eight-bit Type of Service (TOS) field in the IPv4 header and the eight-bit Traffic Class field of the IPv6 header are used for marking ECN in IP packets. For DiffServ, the ToS and TC fields are overridden as the DiffServ (DS) field as will be discussed later in Chapter 5. The last two bits of the DiffServ field in the IP header have experimentally been designated as the ECN field.

Figure 4-29 shows the ECT bit and the CE bit used for the ECN. The seventh bit from the left is defined as the ECN-Capable Transport (ECT) bit and the last bit, i.e., the eighth bit from the left, as the Congestion Experienced (CE) bit.

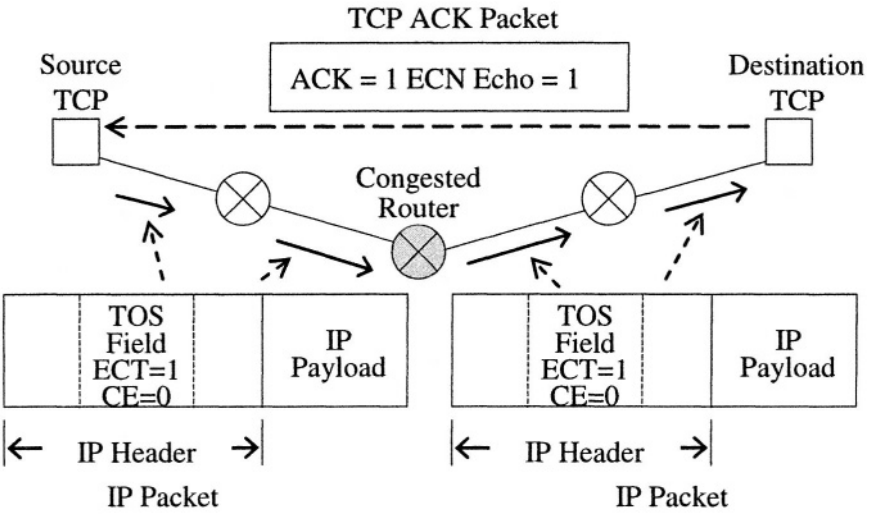


Figure 4-33. ECN operation.

The ECT bit is set by the source TCP application indicating to the routers in the IP network that the packets are eligible for the ECN. When a router anticipates congestion, it sets the CE bit to '1' in the packets with the ECT bit equal to '1' to indicate congestion to the end systems. The routers that have a packet arriving at a full queue drop the packet, as in the RED or the tail drop methods.

5.4.3 ECN marking in the TCP header

The ECN also requires defining two flags using the two bits of the reserved field in the TCP header. These two flags are specified in RFC 2481:[15] the ECN-Echo flag and the Congestion Window Reduced (CWR) flag. Figure 4-30 shows the TCP header specified by RFC 793[16] and shows the six bit reserved field and the six bit code field. Figure 4-31 shows the bit-by-bit specifications of the code filed and the reserved field.

5.4.4 ECN handshaking and operation

The source and destination TCP hosts use the Echo flag and the CWR flag of the reserved field and the SYN and ACK bits of the code field for "handshaking" for the ECN. Figure 4-32 shows the ECN handshaking protocol between the source and destination TCP hosts.

Figure 4-33 shows the ECN operation after the initial handshaking between the TCP hosts shown in Figure 4-32. As shown in the figure, the IP

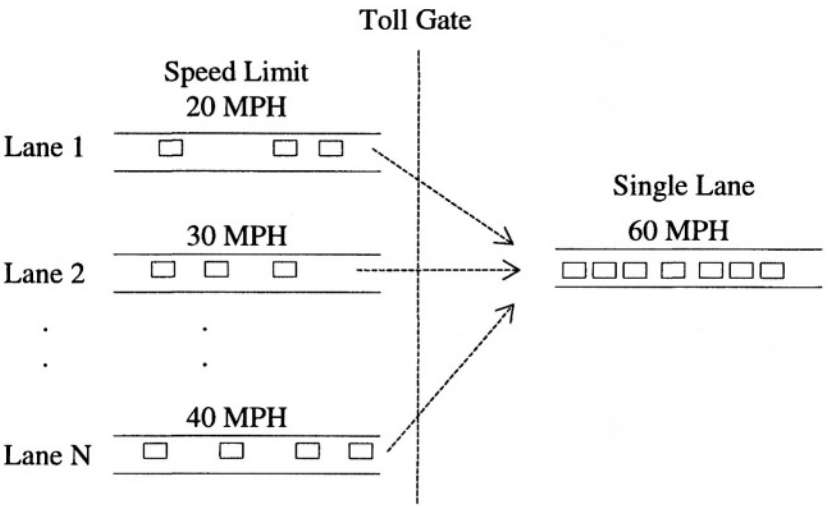


Figure 4-34. An analogy of toll gate.

packet leaves the source host with the ECT bit set to '1' and the CE bit set to 0. The CE bit is flipped to '0' at the router that detects an onset of congestion (by the RED algorithm discussed in Section 5.2). The receiving TCP host flips the echo bit to '1' in the TCP ACK packet to tell the source host that there is congestion in the network.

6. PACKET SCHEDULING

To illustrate packet scheduling, consider a car analogy shown in Figure 4-34. Suppose that a driver on a highway with multiple lanes arrives at a toll plaza. Further suppose that there is only one lane after the toll gate. Only one car can get on the outgoing lane at a time. In general, the outgoing lane should have a higher capacity, or a higher speed limit, than the individual incoming lanes although its capacity could be smaller than the sum of the capacities of all incoming lanes.

The multiple incoming lanes and the lines behind the toll booths are like the queues in a router; the cars, like the packets; the toll gate, like the output port; and the outgoing lane, like the outgoing link. As in the car analogy, in an IP router, one packet can go out of an output port onto the outgoing link at a time. Also, as in the car analogy, an outgoing link typically has a higher speed (i.e., more bandwidth) than the incoming links feeding the queues. Since only one packet can go out at a time, which packet should be allowed

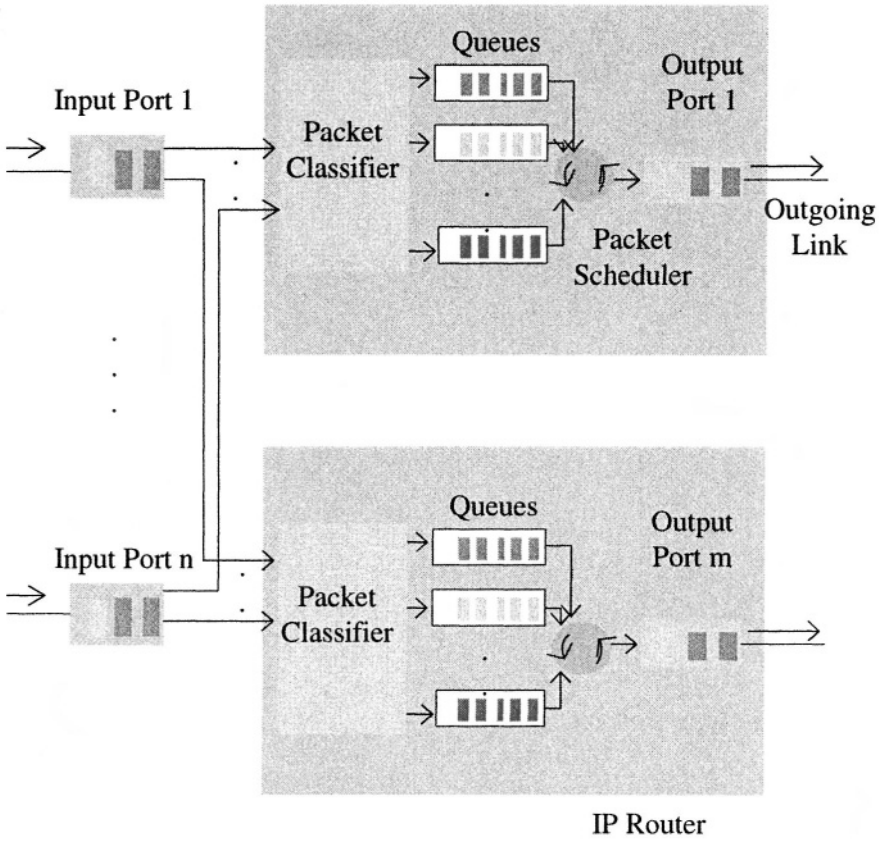


Figure 4-35. Conceptual diagram of a packet scheduler.

to go out next? Packet scheduling is to "schedule" the packets in the queues in such a way that the fixed amount of an output port bandwidth is "equitably" and "optimally" distributed among the competing classes of incoming traffic flows that are routed to that output port.

For example, in the DiffServ mechanism, which will be discussed in Chapter 5, a Per Hop Behavior (PHB) is defined for a traffic class as an externally observable forwarding behavior of that class. A packet scheduling algorithm is designed so that the expected PHB is implemented at a router. Typically, packet scheduling is not "standardized" and is manufacturer-specific. Packet scheduling is at the heart of QoS mechanism and provides a measure of technology differentiation among manufacturers' products.

Figure 4-35 shows packet scheduling: the figure is not intended as the actual router internal structure; rather, it provides a conceptual diagram. The

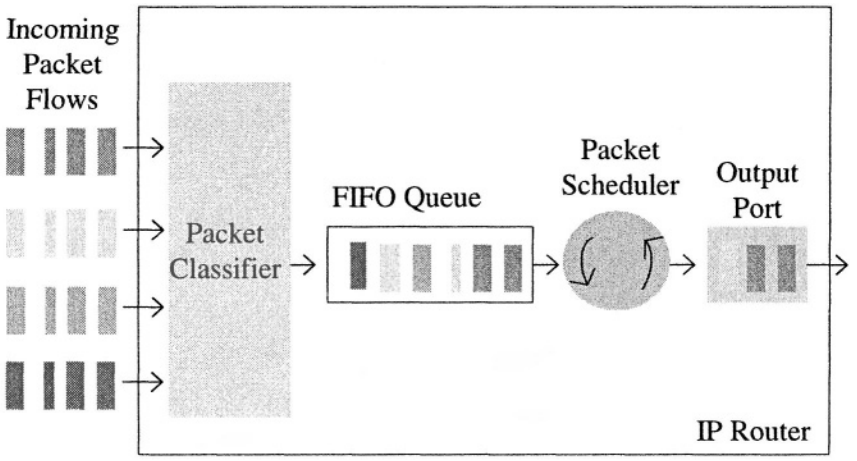


Figure 4-36. The FIFO.

actual implementation of packet scheduling in a router is manufacturer-specific.

As shown in the figure, packet scheduling is applied on a per-output-port-basis. The packets arriving at input ports *1* through *n* are first "routed" to output ports *1* through *m* according to their destination as determined from the routing table in the router. For each output port, the packets are classified and queued. Packet scheduling is applied to these queues destined for a particular output port. The remainder of this section focuses on packet scheduling for one such output port.

The following major types of packet scheduling methods are discussed:

- First-in-first-out (FIFO)
- Priority queuing (PQ)
- Fair-queuing (FQ)
- Weighted Round Robin (WRR)
- Weighted Fair Queuing (WFQ)
- Class-Based WFQ

For each packet scheduling method, its operation, its main objectives and its main drawbacks are discussed.

6.1 FIFO

The first-in first-out (FIFO) is shown in Figure 4-36. The FIFO is a default queuing mechanism in the absence of any specific packet scheduling algorithm. In FIFO, packets are queued in a single queue in the order they arrive and are sent out on the outgoing link in the same order they are

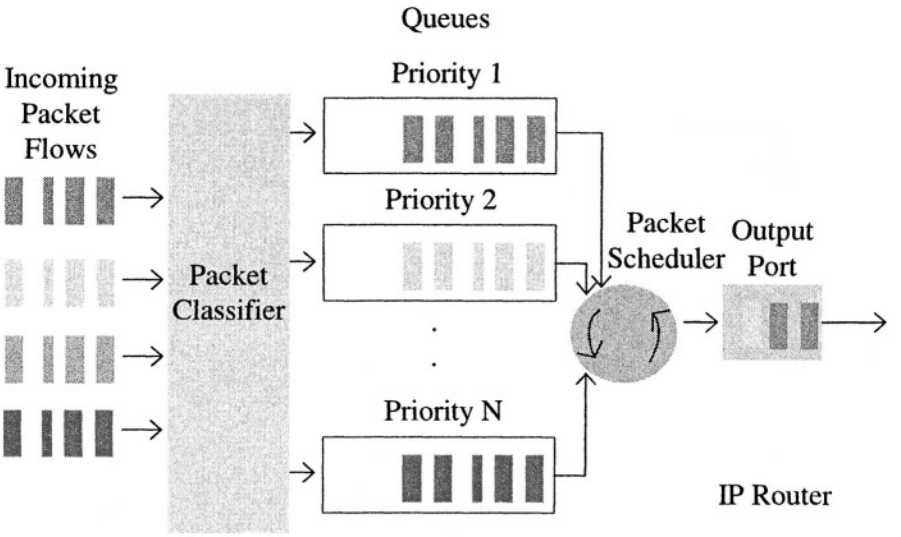


Figure 4-37. Priority Queuing (PQ).

queued. Since the first arriving packet is the first packet to be served, the FIFO queue is also referred to as the first-come-first-served (FCFS) queue.

Just as the main advantage of the default queue management scheme, i.e., the tail drop method, is its simplicity, the main advantage of the default packet scheduling mechanism, the FIFO, is its simplicity. No special algorithm is required to implement the FIFO. All it takes is a single buffer, which can store the incoming packets as they come and send out in the same order.

The FIFO treats all packets equally and, therefore, is best suited for the best effort network. The main drawback of the FIFO, therefore, is that the FIFO does not distinguish (or has a very limited capability to distinguish) traffic classes. Because the FIFO does not provide class differentiation, all traffic flows suffer equally during congestion. The FIFO has some rudimentary capability of distinguishing traffic classes by packet "coloring" as discussed in Section 4. Even in this case, the FIFO has a drawback in that it does not treat a mixed traffic of TCP and UDP packets fairly: the FIFO tends to favor UDP traffic over TCP traffic during congestion because the TCP protocol backs off during congestion.

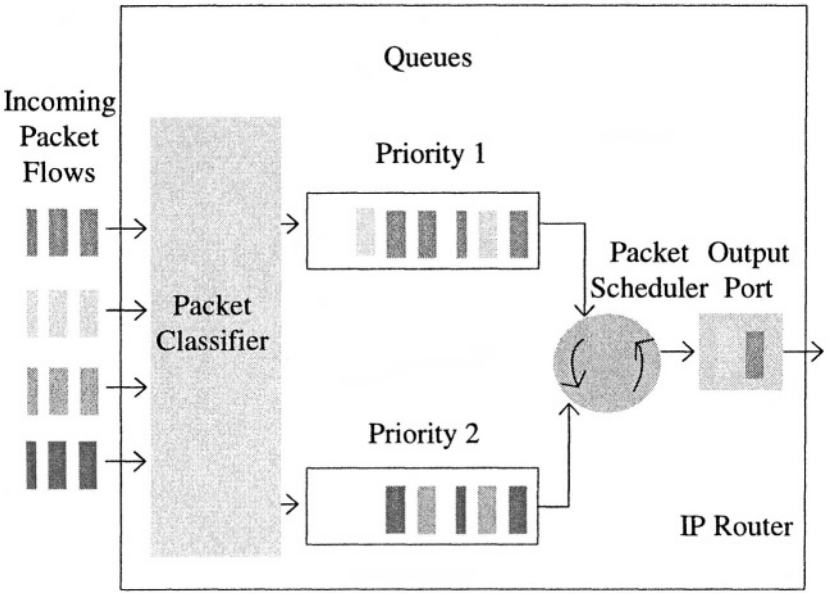


Figure 4-38. Example 7: rate-controled PQ.

6.2 Priority queuing (PQ)

The FIFO puts all packets in a single queue without regard to traffic class distinction. A simple way of creating class distinction is to use the priority queuing (PQ). In the PQ, *N* queues are created as shown in Figure 4-37 with the priority ranking *1* through *N*. The scheduling order is determined by the priority order and by whether there are packets in higher priority queues. The packets in the j^{th} queue are processed only if there are no packets in any of the higher priority queues, i.e., Queues *1* through *(j-1).* For example, if a packet arrives at any of the queues above Queue *j,* say Queue *(j-3),* while the scheduler is at Queue *j,* the scheduler goes to Queue *(j-3),* i.e., there is no preset order such as the round robin order used in other scheduling mechanisms discussed later.

Like the FIFO, the PQ's main advantage is its simplicity: it provides a simple means of creating traffic class distinction. The main drawback of the PQ is that the PQ can cause the phenomenon referred to as "starvation" of low priority queues. As the term implies, if the higher priority queues always have packets to be processed, the lower priority queues may never get a chance to send packets out: the lower priority queues may be deprived

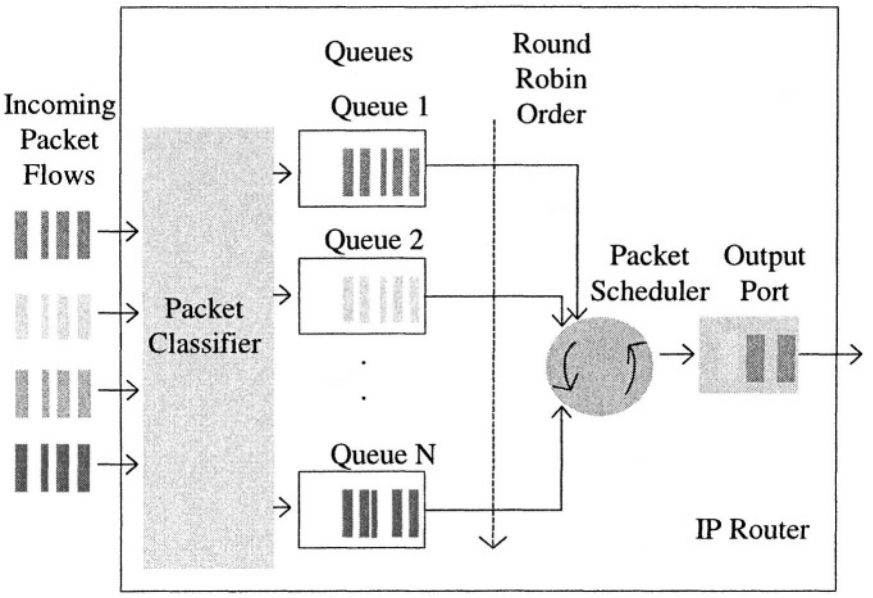


Figure 4-39. Fair Queuing (FQ).

of access to the output port bandwidth completely. Because of the potential starvation problem, care must be taken when the PQ is applied.

The PQ is particularly suitable if the high priority traffic makes up a small fraction of the overall traffic of all the queues. For example, suppose that a tiny fraction of the overall mass of traffic is extremely critical and must always be processed and sent out as quickly as possible. The simplest way of handling this situation is to create a PQ for that traffic leaving the rest of the traffic to other types of queues. Since the priority traffic is small, its effect on the rest of the traffic would be minimal and there is no danger of creating starvation.

The PQ is a convenient and simple means of creating queues dedicated to real time traffic, e.g., voice and video over IP and TDM circuit emulation. The real time traffic such as voice and video typically uses UDP. Using the PQ for TCP traffic requires special care because TCP behavior during congestion may aggravate the starvation problem for the rest of the traffic in other queues.

The operation of the PQ described above is also referred to as the strict PQ. To prevent the starvation problem and ensure a minimum amount of bandwidth for the lower priority queues, the rate-controlled PQ can be used. In the rate controlled PQ, the packets in a high-priority queue are scheduled before the packets in the lower priority queues only if the amount of traffic in the high-priority queue stays below a re-specified threshold level.

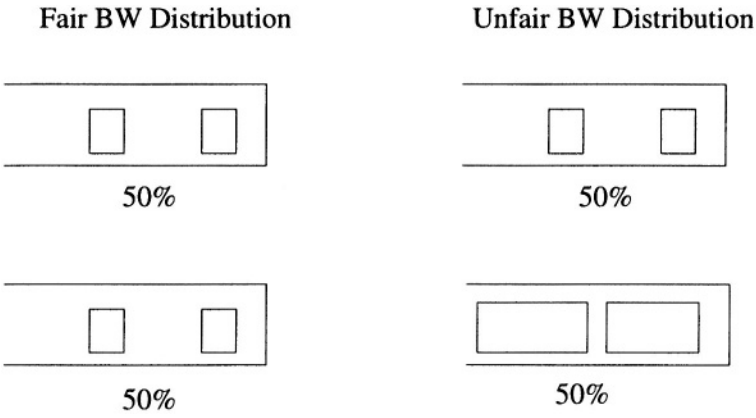


Figure 4-40. Effect of packet size on bandwidth distribution.

Example 7 Rate-controlled PQ

Figure 4-38 shows an example of the rate-controlled PQ. Suppose that the Priority 1 queue is rate limited to 10% of the output port bandwidth. This means that the scheduler operates in the PQ mode as long as the bandwidth consumption by Priority 1 queue stays below 10% of the output port bandwidth. How 10% bandwidth is assured is implementation-specific. For example, the maximum 10% rate of the rate controlled PQ in this example may be realized by allowing the scheduler to spend no more than 10% of the time with Priority 1 queue.

6.3 Fair Queuing (FQ)

Another more general means of creating traffic class distinction is the fair queuing (FQ), which is also referred to as per-flow or flow-based queuing. Figure 4-39 shows the FQ. In the FQ, incoming packets are classified into *N* queues. Each queue is allocated *1/N* of the output port bandwidth. The scheduler visits the queues according to the round robin order skipping empty queues. At each scheduled visit of a queue, one packet is transmitted out of the queue.

The FQ is simple. It does not require a special bandwidth allocation mechanism. If a new queue is added to the existing *N* queues to create a new traffic class, the scheduler automatically adjusts the individual queue bandwidth to *1/(N+1)* of the output port bandwidth. This simplicity is the FQ's main advantage.

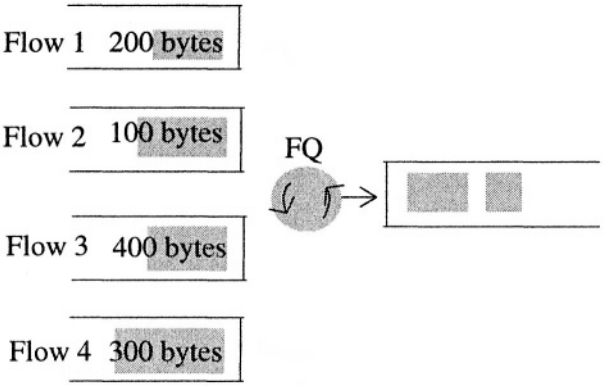


Figure 4-41. Example 8: FQ

The FQ has two major drawbacks. First, since the output port bandwidth is divided to the *N* queues equally by *1/N,* if the input traffic classes have different bandwidth requirements, the FQ will not be able to distribute the output port bandwidth among the input flow classes according to their bandwidth requirements.

Second, since one whole packet is transmitted per each scheduled visit of a queue regardless of the packet size, the packet size will impact the actual bandwidth distribution among the queues even though each queue is visited equally by *1/N.* For example, if a particular queue tends to have bigger size packets than other queues, that queue would get more than the *1/N* share of the output port bandwidth. This is illustrated in Figure 4-40.

Example 8 FQ

Consider a FQ with four queues with Flows 1, 2, 3, and 4 as shown in Figure 4-41. Assuming that the average packet sizes of the four flows are 200, 100, 400 and 300 bytes, respectively, determine the percentage shares of the output port bandwidth used by the four flows.

Solution

Flow 1 $200/1{,}000 = 20\%$ *Flow 2* $100/1{,}000 = 10\%$.

Flow 3 $400/1{,}000 = 40\%$ *Flow 4* $300/1{,}000 = 30\%$.

6.4 Weighted Round Robin (WRR)

The Weighted Round Robin (WRR) queuing addresses the first of the two drawbacks discussed for the FQ in Section 6.3, that is, the FQ's inability to distribute the output port bandwidth to input traffic classes according to their bandwidth requirements. The WRR divides the output port bandwidth to input traffic classes according to their bandwidth requirements. The WRR is also referred to as the class-based queuing (CBQ) or custom queuing.

Figure 4-42 shows the WRR. First, the input traffic flows are grouped into *m* classes and the output port bandwidth is distributed to the *m* classes according to appropriate weights determined by the bandwidth requirements of the *m* classes. The weights should add up to 100%:

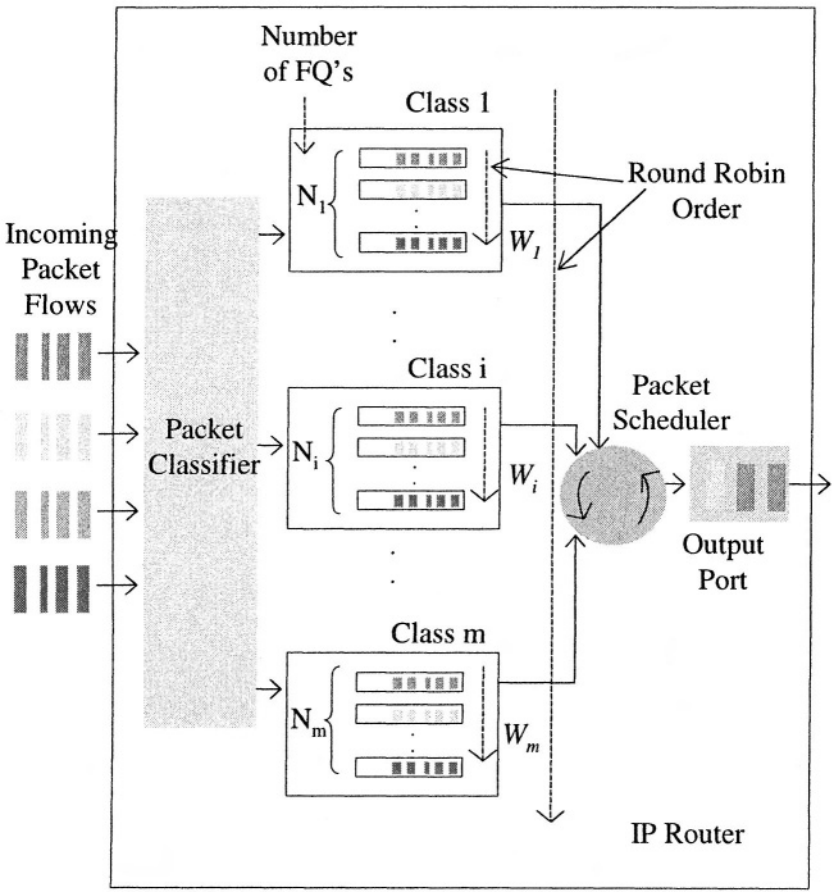


Figure 4-42. Weighted Round Robin (WRR).

$$\sum_{i=1}^{m} W_i = 100\% \tag{4-3}$$

where m is the number of traffic classes and W_i is the percent weight of Class i. Within each class, individual flows are scheduled by the FQ. Denoting the number of FQ's in Class i by N_i, the total number of FQ's in the WRR scheduling is given by the following:

$$Total\ number\ of\ FQ's\ in\ WRR = \sum_{i=1}^{m} N_i \tag{4-4}$$

where m is the total number of traffic classes.

As shown in Figure 4-42, the WRR involves two layers of round robin scheduling. First, Classes *1* through m are visited by the scheduler in the round robin order. Refer to this as the first layer round robin. When the scheduler is with a particular class, the FQ's within that class are visited by the scheduler in the round robin order. Refer to this as the second layer round robin.

The percentage of the output port bandwidth given to Class i, i.e., weight for Class i, W_i, can be realized by specifying the amount of time spent by the scheduler with Class i. For example, suppose that Class i is given 20% of the output port bandwidth, i.e., $W_i = 20\%$. The scheduler must spend 20% of the time during the first layer round robin cycle with Class i. While the scheduler is with Class i, it spends an equal amount of time with each of the N_i FQ's, i.e., $1/N_i$. Hence, the weight allocated to individual FQ's in Class i is given by:

$$W_{ij} = W_i \times (1/N_i) \tag{4-5}$$

where W_i is the weight of Class i, N_i is the number of FQ's in Class i, and W_{ij} is the weight of the j^{th} queue in Class i. The above equation can be written as:

$$W_{ij} = W_i \times w_{ij} \tag{4-6}$$

where w_{ij} is the percentage distribution (i.e., weight) of Class i's bandwidth to the j^{th} queue in Class i, and the FQ gives an equal weight to all queues:

$$w_{ij} = 1/N_i \tag{4-7}$$

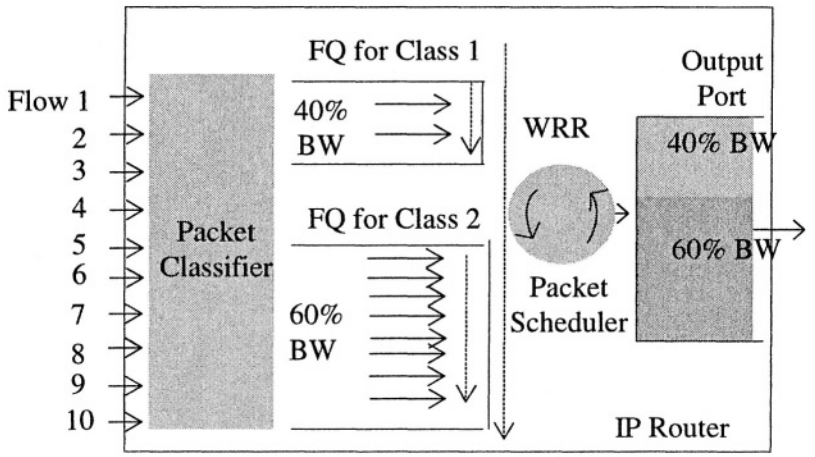


Figure 4-43. Example 9: WRR.

Also the following relation should hold true:

$$W_i = \sum_{j=1}^{N_i} W_{ij} \qquad (4\text{-}8)$$

By using W_i's, rather than the equal division *1/m,* the WRR can create *m* traffic classes with different output port bandwidth requirements, thereby circumventing the first drawback of the FQ discussed in Section 6.3.

Example 9 WRR

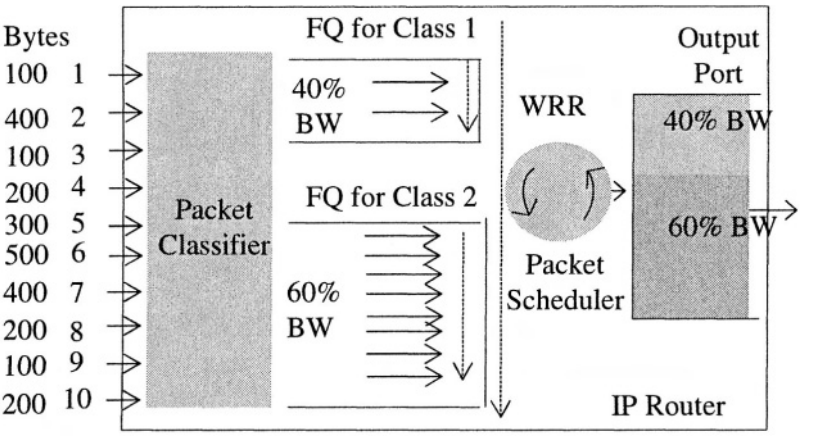


Figure 4-44. Example 10: WRR.

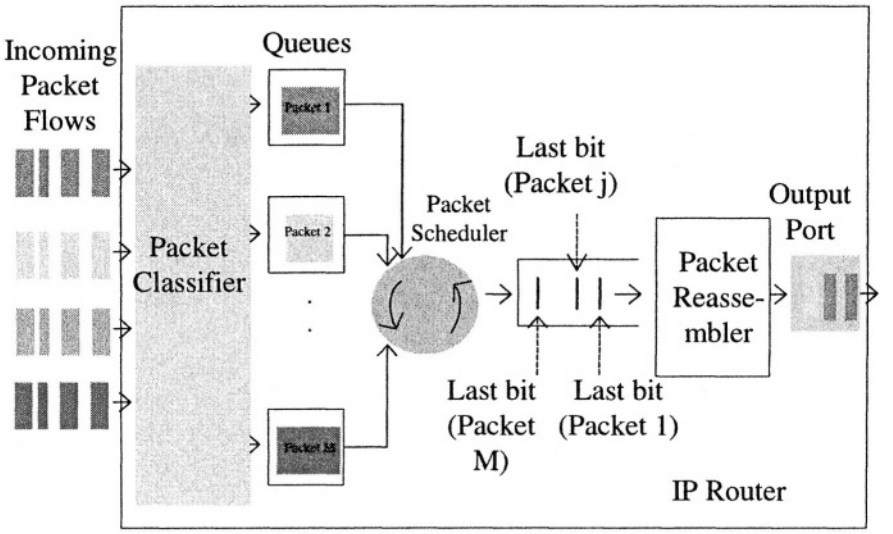


Figure 4-45. Weighted Bit-By-Bit Round Robin scheduler.

Consider a WRR with two classes, Class 1 and Class. The total output port bandwidth is 200 Mb/s as shown in Figure 4-43. Forty percent and 60% of the output port bandwidth are allocated to Class 1 and Class 2, respectively. Class 1 has two flows and Class 2 has eight flows. Assume that all flows have the same packet size. Determine the output port bandwidth consumed by each of the eight flows.

Solution

Class 1 *80 mb/s.* Each of the two flows has *40 mb/s.*
Class 2 *120 mb/s.* Each of the eight flows has *15 mb/s.*

Example 10 WRR

Assume that the ten flows in Example 9 have the following packet sizes, respectively: 100, 400, 100, 200, 300, 500, 400, 200, 100 and 200 bytes, respectively. Determine the output port BW consumed by each of the ten flows.

Solution

Flow 1 20% x 1/5 = 4% *Flow 2 20% x 4/5 = 16%*
Flow 3 80% x 100/2000 = 4% *Flow 4 80% x 200/2000 = 8%*
Flow 5 80% x 300/2000 = 12% *Flow 6 80% x 500/2000 = 20%*
Flow 7 80% x 400/2000 = 16% *Flow 8 80% x 200/2000 = 8%*

Flow 9 80% x 100/2000 = 4% *Flow 10 80% x 200/2000 = 8%*

6.5 Weighted Fair Queuing (WFQ)

Although the WRR addresses the first drawback of the FQ, the WRR does not address the second drawback of the FQ, i.e., the packet size impact on bandwidth share, because the WRR uses the FQ within the classes. The Weighted Fair Queuing (WFQ) addresses this drawback of the FQ. In the WFQ, as in the FQ, the input traffic flows are grouped into *m* queues; however, the output port bandwidth is distributed to the *m* queues according to appropriate weights determined by the bandwidth requirements of the *m* classes rather than equally divided, where the weights add up to 100%:

$$\sum_{i=1}^{m} W_i = 100\% \qquad (4\text{-}9)$$

where *m* is the number of traffic classes in the WFQ and W_i is the percent weight of Class *i*. In the FQ, each queue sends out one whole packet during a scheduled visit. In the WFQ, the scheduler sends out the packets from the queues based on the calculated order of the packet finish time. The WFQ attempts to approximate a theoretical model referred to as the weighted bit-by-bit round robin scheduler shown in Figure 4-45.

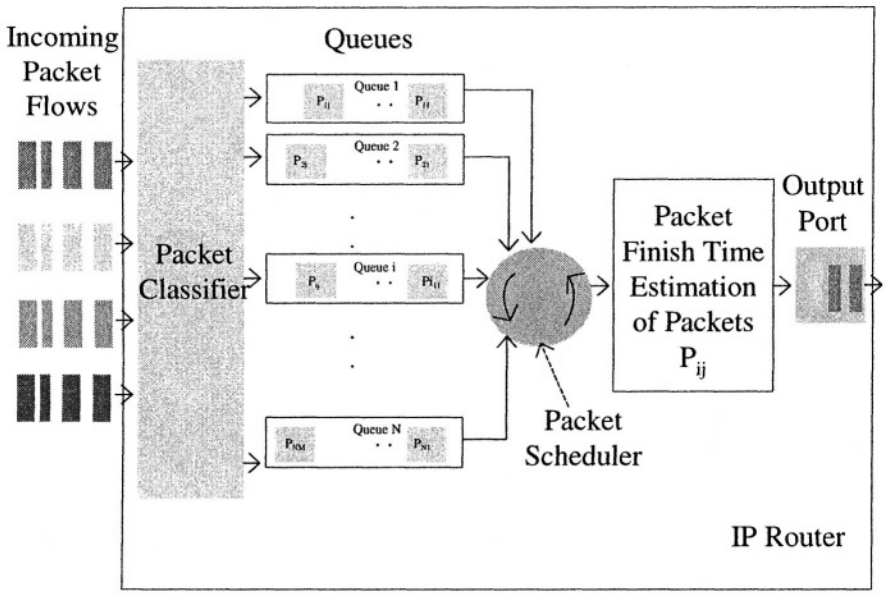


Figure 4-46. Weighted Fair Queuing (WFQ).

As shown in the figure, the weighted bit-by-bit round robin scheduler visits the queues in the round robin order; however, at each scheduled visit, the scheduler takes out one bit from the queue at a time; the packet assembler collects all the bits of a packet, when the packet is reassembled, it is sent out. Hence, a larger size packet must wait longer to be reassembled. This bit-by-bit scheduler is a theoretical model only and is not practical.

Figure 4-46 shows the WFQ. The WFQ calculates the finish times of the packets and sends them out from the output port in the order of the finish times calculated by the scheduler.

6.6 Class-Based WFQ (CB WFQ)

Figure 4-47 shows the Class-Based (CB) WFQ. The CB WFQ is similar

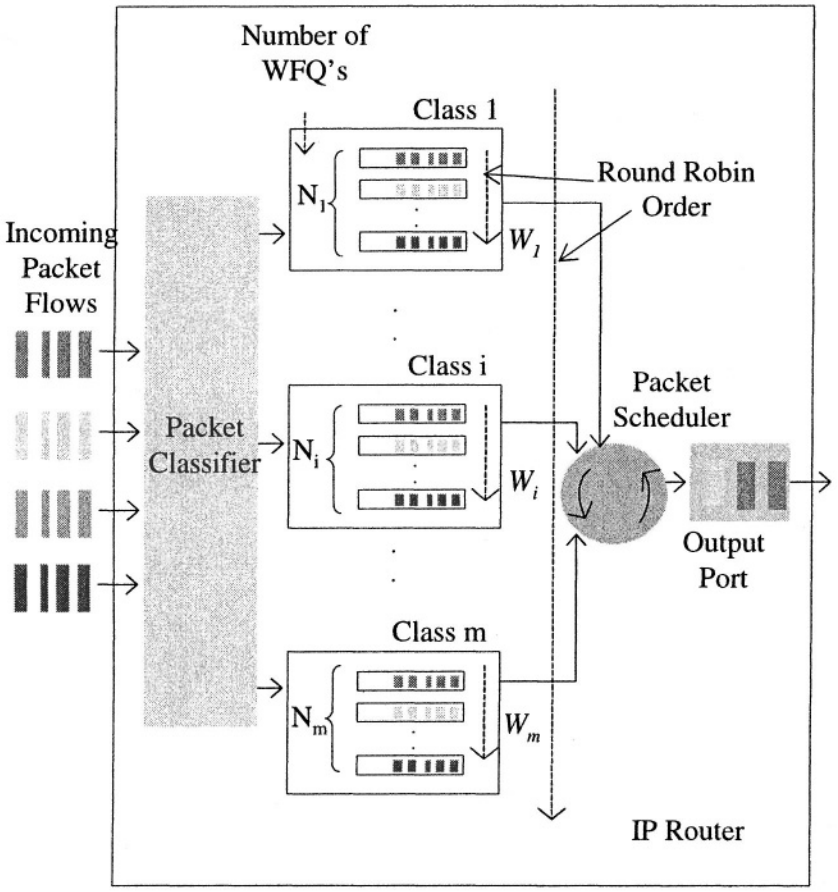


Figure 4-47. Class-Based WFQ.

to the WRR shown in Figure 4-19. In the CB WFQ, as in the WRR, the input traffic flows are grouped into m classes and the output port bandwidth is distributed to the m classes according to the appropriate weights determined by the bandwidth requirements of the m classes, where the weights add up to 100%:

$$\sum_{i=1}^{m} W_i = 100\% \tag{4-10}$$

where m is the number of traffic classes and W_i is the percent weight of Class i. Up to this point, the CB WFQ and the WRR are the same. The difference is within each class. In the CB WFQ, within a class, individual flows are scheduled by the WFQ, whereas in the WRR, they are scheduled by the FQ.

Denote the number of WFQ's in Class i by N_i; then, the total number of WFQ's in the CB WFQ scheduling is given by the following:

$$Total\ umber\ of\ FQ's\ in\ CB\ WFQ = \sum_{i=1}^{m} N_i \tag{4-11}$$

where m is the total number of traffic classes. The bandwidth allocated to Class i is further distributed among the N_i queues within Class i according to appropriate weights, w_{ij}. The weight given to WFQ j in Class i is given by:

$$W_{ij} = W_i \times w_{ij} \tag{4-12}$$

where

W_i = the percentage distribution (i.e., weight) of the output port bandwidth to Class i

w_{ij} = the percentage distribution (i.e., weight) of Class i's bandwidth to the j^{th} queue in Class i

W_{ij} = the percentage distribution of the output port bandwidth to the j^{th} queue in Class i

N_i = the number of queues in Class i

m = the number of classes.

The sum of the weights (of the output port share) of the queues in a class should add up to the weight (of the output port share) of that class:

$$W_i = \sum_{j=1}^{N_i} W_{ij} \tag{4-13}$$

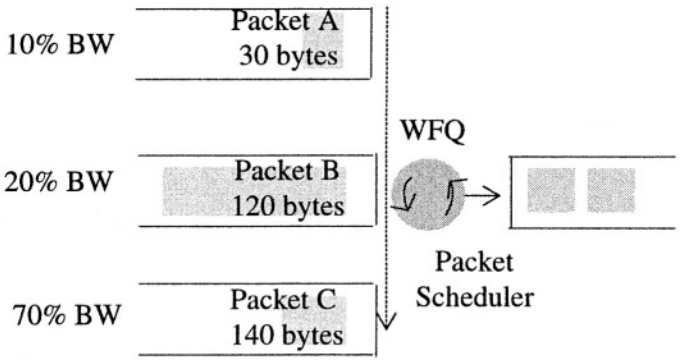


Figure 4-48. Example 11: WFQ

Example 11 WFQ

Consider the three queues with WFQ scheduling with the percentages of output port bandwidth as shown in Figure 4-48. Consider the three packets, A, B, and C, all at the head of Queue (HoQ) with the respective lengths as shown in the figure. Determine the order of transmission of the three packets in the output port.

Solution

At each schedule time, imagine that following bytes are sent, respectively, for the three packets: 10 bytes; 20 bytes; and 70 bytes. Packet A would take three scheduled visits, Packet B, six visits, and Packet C, two visits.

Hence the order of transmission would be C, A and B.

7. TRAFFIC SHAPING

Traffic shaping is to change the rate of incoming traffic flow to regulate the rate in such a way that the outgoing traffic flow behaves more smoothly. If the incoming traffic is highly bursty, it needs to be buffered so that the output of the buffer is less bursty and smoother.

In this way, traffic shaping makes the traffic flow behave more like the predefined traffic profile, e.g., per an SLA. An analogy of traffic shaping is the "stop and go" driving through, for example, Lincoln Tunnel to Manhattan. Drivers are asked to first stop momentarily at the entrance to the

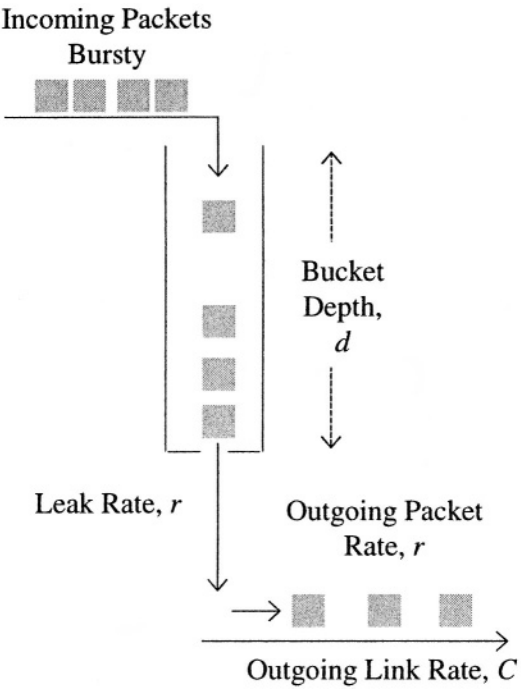


Figure 4-49. The pure traffic shaper.

tunnel and proceed at a certain speed, e.g., 30 MPH. Traffic shaping would introduce a delay through the buffer.

There are two types of traffic shaper: the pure traffic shaper and the token bucket traffic shaper. The latter is sometimes referred to as the leaky bucket traffic shaper.

7.1 Pure traffic shaper

Figure 4-49 shows the pure traffic shaper. Incoming packets are put into a buffer, or a "bucket," with depth *d,* and are sent out on the outgoing link at a constant rate. This constant rate is referred to as the leak rate, *r.*

The pure traffic shaper does not allow bursts on the outgoing stream. Typically, the leak rate, *r,* is much smaller than the link rate, *C.* However, with the pure traffic shaper, the leak rate *r* places the upper limit on the outgoing rate of the traffic flow, because it does not allow bursts on the outgoing link. If the burst size exceeds the bucket depth, *d,* the overflowing packets would be dropped.

7.1.1 Token bucket traffic shaper

Figure 4-50 shows the token bucket traffic shaper. The token bucket traffic shaper uses a token bucket, which is similar to Bucket C used for policing the CIR in the srTCM and the trTCM.

Tokens are put into the token bucket at a constant rate referred to as the token rate, *r*. The token rate *r* is similar to the CIR. The token bucket has a maximum size, which is referred to as the bucket depth, *d*. The bucket depth *d* is similar to the size of Bucket C, the CBS. If the token bucket is full, no more tokens are put into the bucket.

Each token allows the input traffic buffer to send out one byte of packet. When there is no packet in the buffer to send out, the "bottom" of the token bucket is closed and no tokens are taken out. When there are packets in the buffer, the tokens are withdrawn at the outgoing link rate, C, and so the packets are "bursted out" on the outgoing link. If the token bucket is

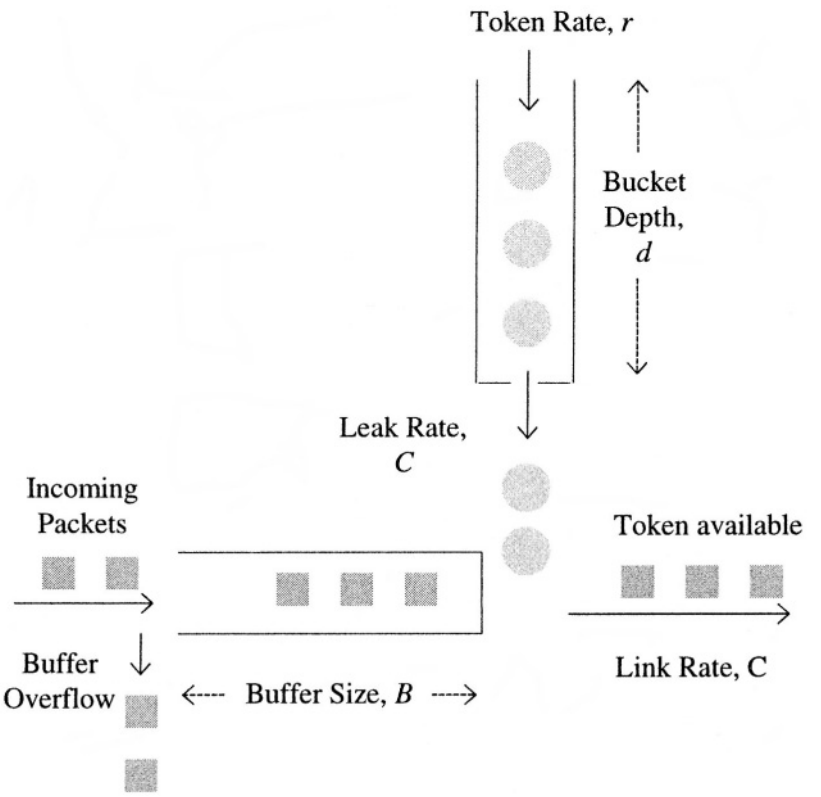


Figure 4-50. The token bucket traffic shaper.

completely depleted with no more tokens left, packets in the buffer must wait for tokens to be put into the bucket.

The result of this operation is that packet bursts are permitted on the outgoing link at the link speed C. The burst size is limited by the bucket depth, *d.* Since tokens are put into the bucket at the token rate *r,* the long term average rate of the packets on the outgoing link would be *r.* Hence, the token bucket traffic shaper works exactly same as in Bucket C of the srTCM and the trTCM except that the token bucket is applied at the output port, whereas Bucket C is applied at the input port.

8. EXERCISES

8.1 Problems

1. Assume the following:
   - Color-blind mode
   - CIR = 500 bytes/sec
   - CBS = 50 bytes
   - EBS = 100 bytes
   - At time t, $T_c(t) = 30$ tokens; $T_e(t) = 70$ tokens
   - A packet of 40 bytes arrives at time t.

   Determine
   - Color of packet:
   - T_c and T_e after marking the packet:

2. Which of the following entities sets the CE bit to 1 ?

   Source TCP host

   Destination TCP host

   Congested router IP layer

   Which of the following entities sets the ECT bit to 1?

   Source TCP layer

   Destination TCP layer

   Source IP layer

   Destination IP layer

   Which of the following entities informs the source of the congestion in the network?

   Source TCP layer

   Destination TCP layer

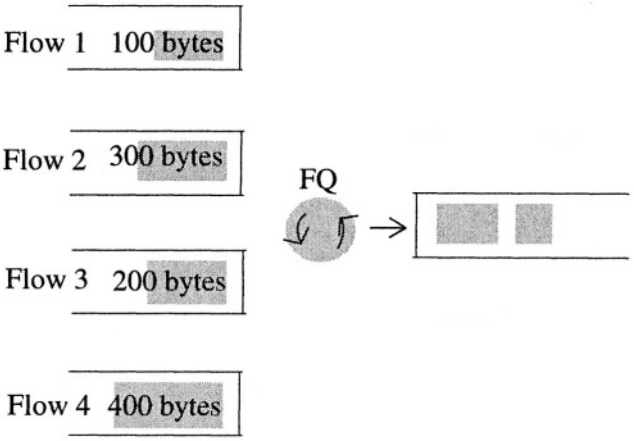


Figure 4-51. Exercise 3.

Source IP layer
Destination IP layer

3. Consider a FQ with four queues with Flows 1, 2, 3, and 4. Assume that the average packet sizes of the four flows are 100, 300, 200 and 400

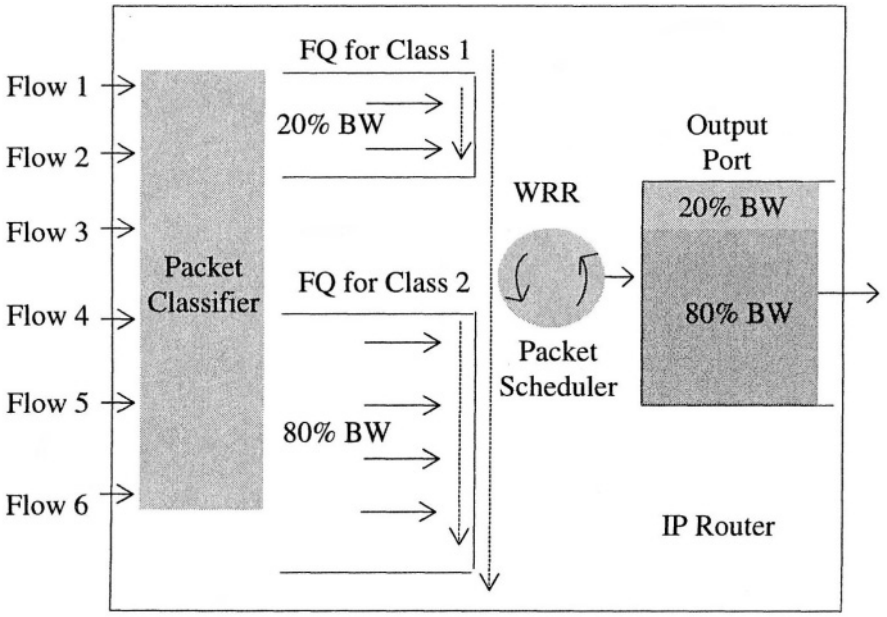


Figure 4-52. Exercise 4.

bytes, respectively. Determine the percentage shares of the output port bandwidth used by the four flows.

4. Consider a WRR with two classes, Class 1 and Class. The total output port bandwidth is 100 Mb/s. Twenty percent and 80% of the output port bandwidth are allocated to Class 1 and Class 2, respectively. Class 1 has two flows and Class 2 has four flows. Assume that all flows have the same packet size. Determine the output port bandwidth consumed by each of the six flows.

5. For the problem of exercise 4, assume that Flows 1 – 6 have the following packet sizes, respectively: 100, 300, 100, 200, 300, and 400 bytes. Determine the output port BW consumed by each of the six flows.

6. Consider the three queues with WFQ scheduling. The three queues have the percentages of output port bandwidth as shown in the figure. Consider the three packets, A, B, and C, all at the HoQ with the respective lengths as shown in the figure. Determine the order of transmission of three packets in the output port.

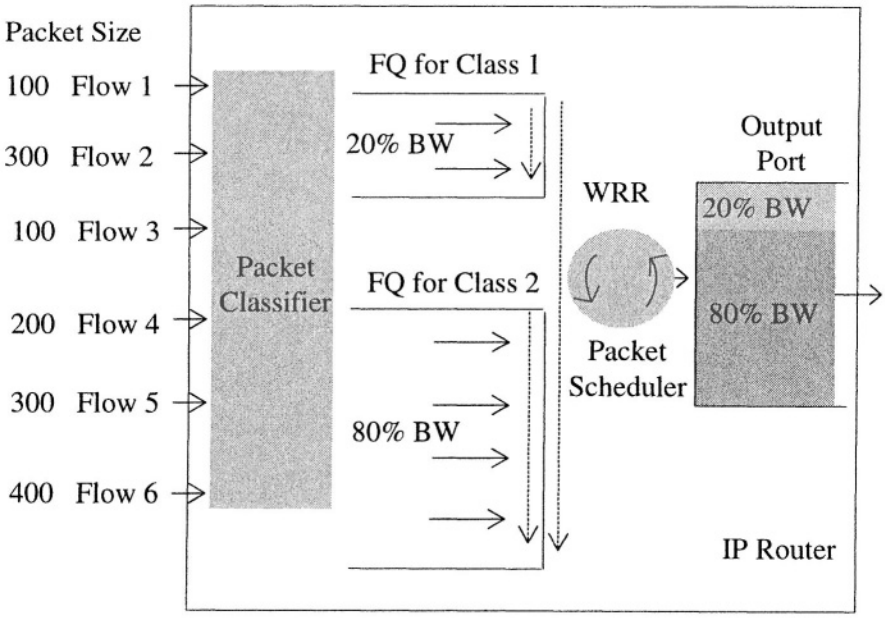


Figure 4-53. Exercise 5.

8.2 Solutions

1. yellow.
   $T_c = 30; T_e = 30.$

2. Congested router IP layer (√)
   Source IP layer (√)
   Destination TCP layer (√)

3.

Flow 1	100/1,000 = 10%
Flow 2	300/1,000 = 30%
Flow 3	200/1,000 = 20%
Flow 4	400/1,000 = 40%

4.

Class 1	20 mb/s	Flow 1	10 mb/s
		Flow 2	10 mb/s
Class 2	80 mb/s	Flow 3	20 mb/s
		Flow 4	20 mb/s
		Flow *5*	20 mb/s
		Flow 6	20 mb/s

5.

Flow 1	20% x(1/4) = 5%
Flow 2	80% x (3/4)= 15%
Flow 3	80% x (100/ 1,000) = 8%
Flow 4	80% x (200/1,000) = 16%
Flow 5	80% x (300/1,000) = 24%
Flow 6	80% x (400/1,000) = 32%

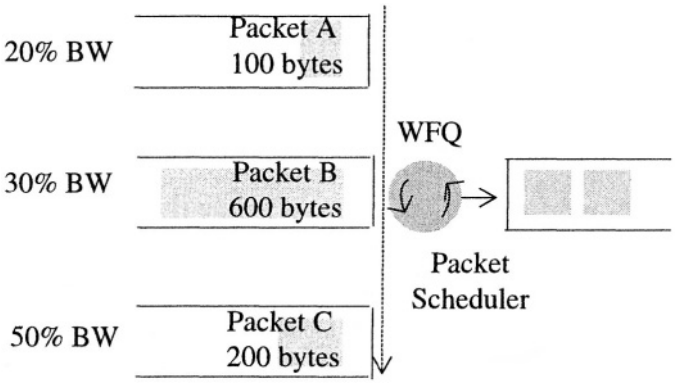


Figure 4-54. Exercise 6.

6. At each schedule time, pretend that following bits are sent: 20 bytes; 30 bytes; 50 bytes. Then, A takes 5 scheduled visits; B, 20 scheduled visits; C, 4 visits.
   Hence, order: C, A, B.

Chapter 5

IP INTEGRATED SERVICES AND DIFFERENTIATED SERVICES

This chapter discusses two specific IP QoS mechanisms: IntServ and DiffServ. Some of the generic IP QoS functional requirements discussed in Chapter 4 will be discussed further as they apply to IntServ and DiffServ, if appropriate.

1. INTEGRATED SERVICES

1.1 IntServ basic functional requirements

In IntServ, an individual IP flow is identified by the following quintuple of parameters:

- Protocol identifier
- Destination IP address
- Destination port address
- Source IP address
- Source port address

To make a resource reservation for a flow, the source application must provide a flow specification. A flow specification consists of a traffic characterization and service requirements for the flow. The traffic characterization includes the peak rate, the average rate, the burst size and the leaky bucket parameters; and the service requirements include the minimum bandwidth required and the performance requirements, e.g., delay,

delay jitter, and packet loss rate. IntServ uses the Resource Reservation Protocol (RSVP) for reserving resources for a flow.

1.2 Resource Reservation Protocol (RSVP)

1.2.1 Overview of RSVP

RSVP is specified in RFC 2205.[17] RSVP is an IP QoS reservation setup protocol. It supports both IPv4 and IPv6 and is applicable for both multicast and unicast mode of IP. In RSVP, resource is reserved in each direction separately.

The source and destination hosts exchange RSVP messages to establish the packet classification and forwarding state at each node. The source initiates the reservation request but the determination of available resources and the actual reservation of the resources begin from the receive end. The "state" of resource reservation at the RSVP nodes is not permanent and is refreshed periodically.

RSVP is not a routing protocol. RSVP messages take the same path that IP packets take, which is determined by the routing tables in the IP routers. RSVP provides several reservation styles. RSVP is a complicated protocol. Because each node on the path must keep the reservation state, for large networks, RSVP becomes impractical because of the scalability problem.

1.2.2 RSVP operation

An RSVP session is normally defined by the following three parameters:

- Destination address

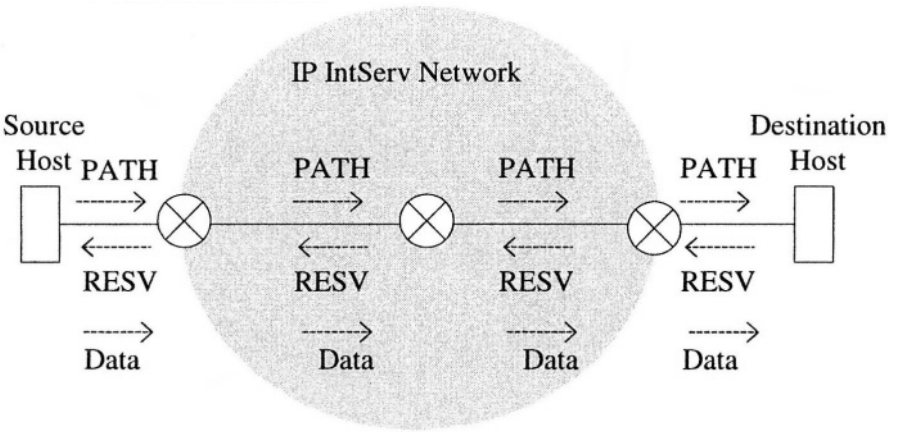


Figure 5-1. RSVP operation.

- Protocol identifier
- Destination port

Figure 5-1 shows the operation of RSVP. The sending host sends a PATH message to the destination host for a flow or a "session." The PATH message contains a flow specification for the flow. As the PATH message passes through the routers on the path, the routers register the flow identification and the flow specification so that, when the corresponding RESV message comes from the receiving host, the routers make the appropriate correlation between the information contained in the PATH and the RESV messages. When the receiving host receives the PATH message, it sends a RESV message. The RESV message carries the resource reservation information. The IP packets of the flow travel in the direction of the PATH message.

1.2.3 RSVP reservation styles

Three types of reservation styles are defined in RFC 2205[17] as shown in Figure 5-2. The sender control controls the selection of senders. Two types of sender control are defined. In the explicit selection style, an "explicit" list of all selected senders is defined. In the wildcard selection, all the senders to the session are selected.

The sharing control controls the treatment of reservations for different senders within a same session. Two types of sharing control are defined. In the distinct reservation style, reservation is made for each upstream sender. In the shared reservation style, a single reservation is shared by multiple upstream senders.

Sender Selection	Reservations	
	Distinct	Shared
Explicit	Fixed-Filter (FF) Style	Shared-Explicit (SE) Style
Wildcard	(None defined)	Wildcard-Filter (WF) Style

Source: IETF RFC 2205.

Figure 5-2. RSVP reservation styles.

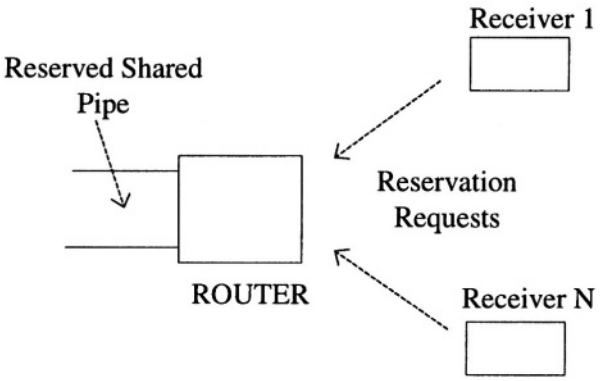


Figure 5-3. Reserved shared pipe.

As shown in Figure 5-2, there are four possible combinations of the sharing control and the sender selection control. However, one of the four combinations is not defined. The remaining three styles are the Fixed-Filter (FF) style, the Shared-Explicit (SE) style and the Wildcard-Filter (WF) style.

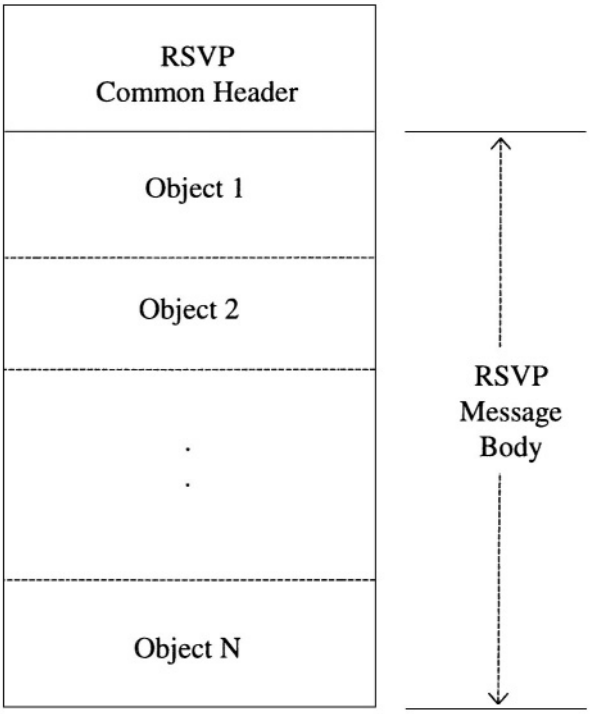


Figure 5-4. RSVP message structure.

Table 5-1. RSVP message types.

Message Type Field	Message Type
1	Path
2	Resv
3	PathErr
4	ResvErr
5	PathTear
6	ResvTear
7	ResvConf

Figure 5-3 shows a bandwidth "pipe" reserved to be shared by multiple senders.

1.2.4 RSVP message format

An RSVP message has a common header followed by a number of "objects" as shown in Figure 5-4. Each object in turn has the object header followed by object contents. The RSVP common header format is shown in Figure 5-5. The total length of the header is eight bytes or 64 bits. It includes four bits for the RSVP protocol version number; four bits for flags; eight bits for the RSVP message type; 16-bit check sum field; eight bits for the send Time to Live (TTL); eight reserved bits; and a 16-bit message length field. The four bit flag field has not yet been defined.

If the check sum field contains all zeros, it means that no check sum was transmitted for the message. The Send_TTL field specifies the IP TTL value with which the message was sent. The RSVP length includes the common header and the variable-length objects that follow.

There are seven types of RSVP messages as shown in Table 5-1. The

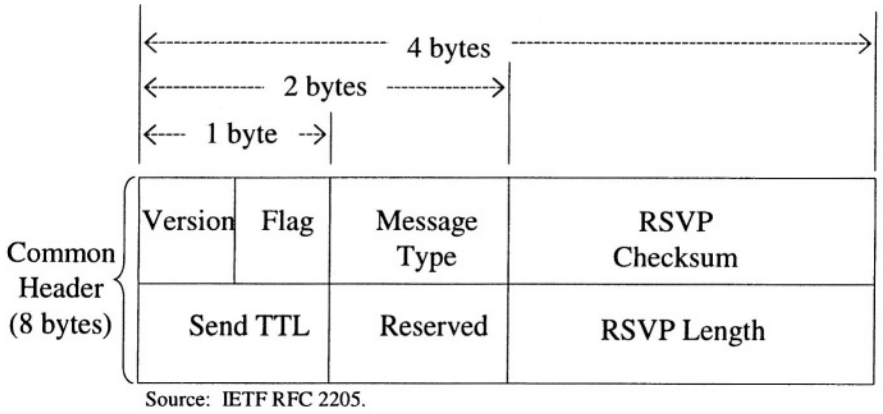


Figure 5-5. RSVP common header format.

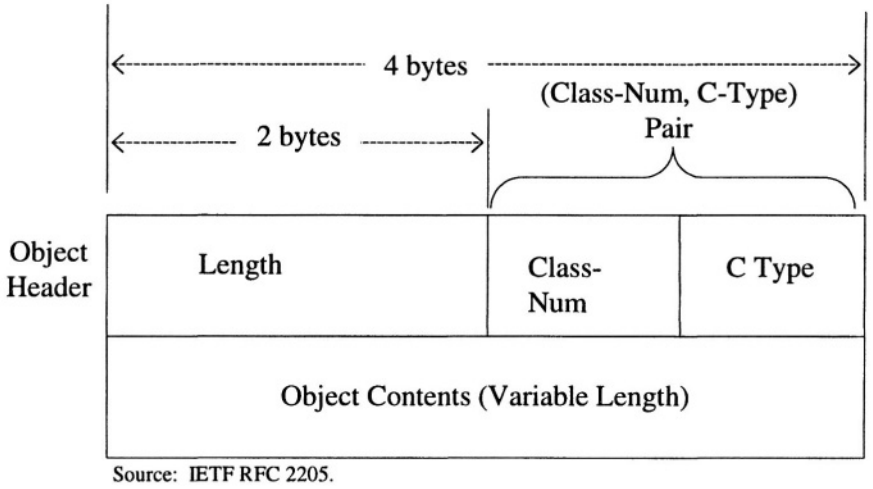


Figure 5-6. RSVP message object format.

PATH and RESV messages are the primary messages used for reserving network resources. The other messages are the error messages, the path tear down and reservation tear down message and a reservation confirmation message

The RSVP object format is shown in Figure 5-6. An RSVP object consists of a 32-bit object header and object contents of a variable length. An object length is in multiples of 32 bits and is up to a maximum of 65,528 bytes.

The RSVP objects are organized by "object class" and "object type." The "Class-Num" field defines the object class and the "C Type" field defines a unique object in the object class as illustrated in Figure 5-7. A "Class-Num, C Type" pair uniquely identifies an RSVP object.

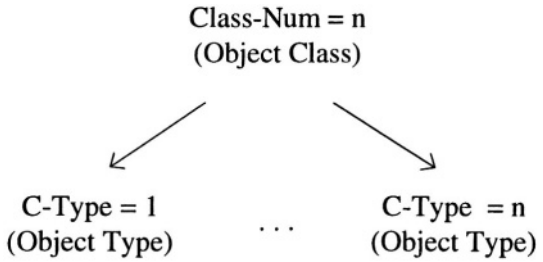


Figure 5-7. Class-Num and C-Type.

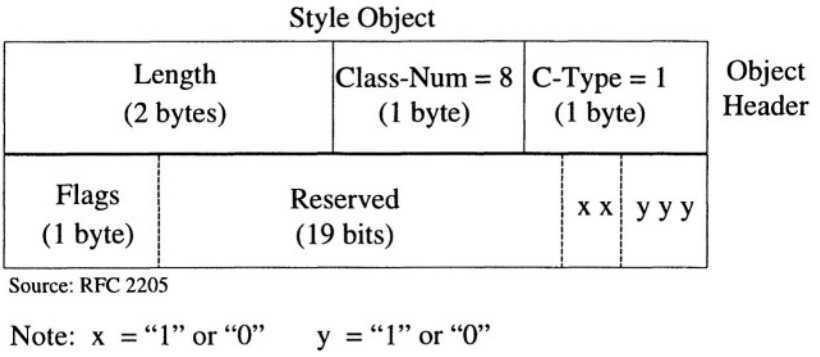


Figure 5-8. Style object.

The following object classes are used in the RSVP messages:

- NULL
- SESSION
- RSVP_HOP
- TIME_VALUES
- STYLE
- FLOWSPEC
- FILTER_SPEC
- SENDER_TEMPLATE

The FLOWSPEC object contains the following parameters for the controlled load service:

- Token bucket rate (receiver's desired reservation)
- Token bucket size (receiver's desired reservation)
- Peak data rate (receiver's desired reservation)
- Etc.

One of the object classes required for the Resv message is the STYLE class with Class-Num = 8. This class has one object, which is the Style Object, with C-Type = 1. The Style Object defines the reservation style.

Table 5-2. Bit settings for the sharing control.

xx bit setting	Sharing Control
00	Reserved
01	Distinct reservation
10	Shared reservation
11	Reserved

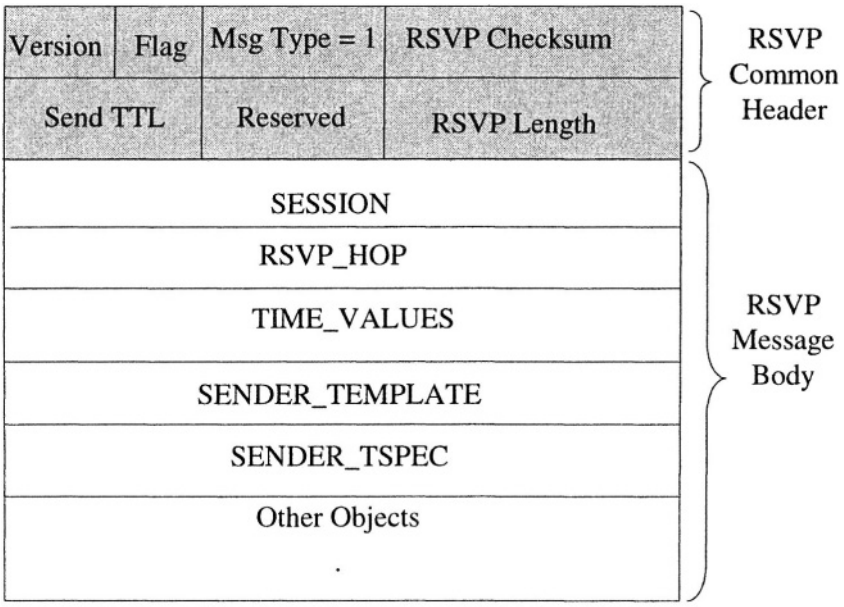


Figure 5-9. RSVP PATH message format.

Figure 5-8 shows the format of the Style Object. The reservation style is defined by the last five bits. The first two bits shown in Figure 5-8 by "xx" define the sharing control, and the last three bits "yyy," the sender selection control.

Tables 5-2 and 5-3 show the "xx" and "yyy" bit settings and corresponding definitions of the sharing control and the sender selection control.

1.2.5 PATH message

Figure 5-9 shows the format of the PATH message. The PATH message is identified by setting the message field of the RSVP common header to one.

Table 5-3. Bit setting for the sender selection control.

yyy bit setting	Sender Selection Control
000	Reserved
001	Wildcard
010	Explicit
011-111	Reserved

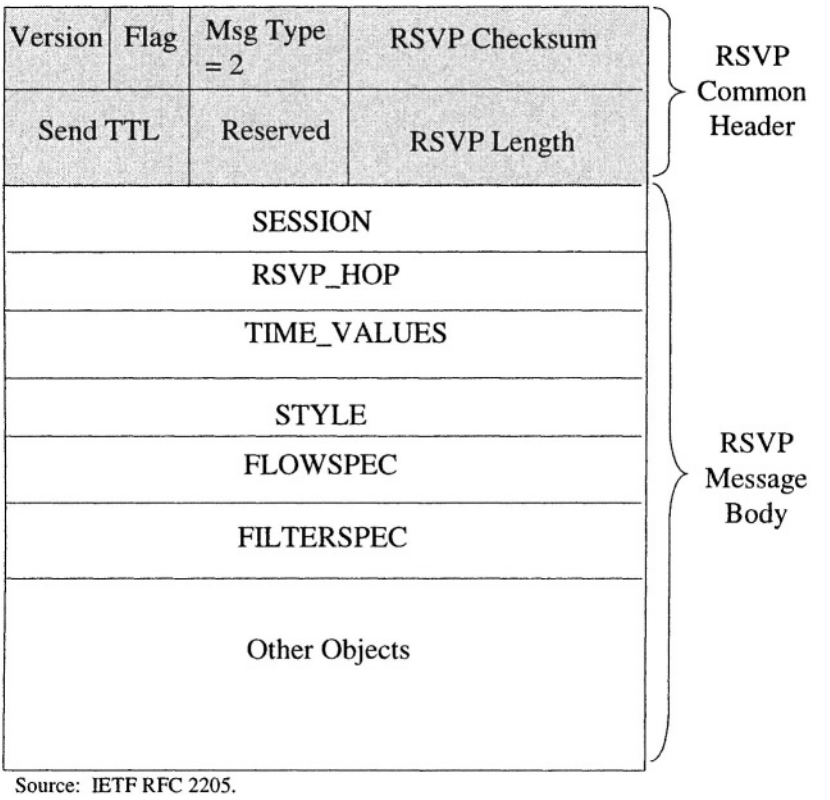


Figure 5-10. RSVP RESV message format.

Each RSVP-capable node along the path captures a PATH message and processes it to create a path state for a sender session defined by the flow specification.

A PATH message carries the sender's IP address as its IP source address and the destination IP address for the session.

1.2.6 RESV Message

Figure 5-10 shows the format of the RESV message. A RESV message carries a reservation request hop-by-hop from the receiver to the sender along the reverse paths of data flows for the session. This path is the same path that the corresponding PATH message has taken.

2. DIFFERENTIATED SERVICES

This section discusses the following topics:

- Differentiated Services (DiffServ) architecture
- DiffServ packet marking
- DiffServ Code Points (DSCP's)
- Per Hop Behaviors (PHB's)

2.1 DiffServ overview

In DiffServ, individual traffic flows are not distinguished and are aggregated into a small number of traffic classes. In DiffServ, bandwidths and other network resources are allocated to the aggregated classes of traffic rather than individual flows. The main focus of DiffServ is placed on a single DS domain rather than the end-to-end paths that packets take.

DiffServ only provides relative "differential" treatments to different traffic classes. Because of the "relative" differentiation, DiffServ alone cannot provide an "absolute" level of QoS. To ensure some level of "absolute" QoS, admission control may be needed at the edges of the DS domain to control the amount of traffic entering the network. Unlike IntServ where the RSVP signaling is used to reserve bandwidths along the path, QoS in DiffServ is provided by provisioning rather than reservation.

The term "DiffServ" describes the overall treatment of a customer's traffic within a service provider's network and defines the service that the

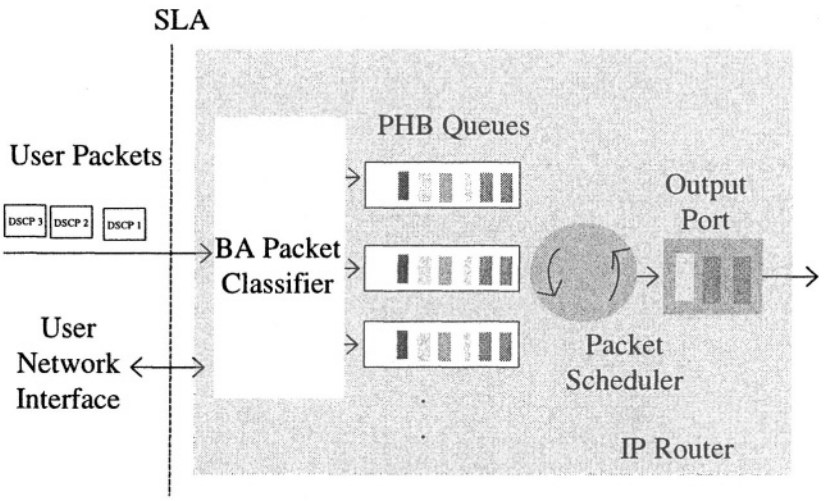


Figure 5-11. DiffServ steps.

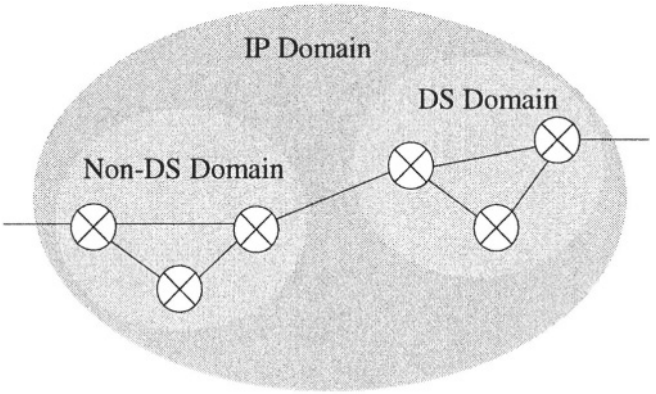


Figure 5-12. IP domain.

customer can expect from the service provider, e.g., an ISP. A DiffServ service is defined in the form of the Service Level Agreement (SLA) between a customer (e.g., a particular customer application such as VoIP, TCP, etc.) and a DiffServ service provider's network.

A DiffServ is defined in terms of the parameters that the customer understands such as the Traffic Conditioning Agreement (TCA), the traffic profiles (e.g., token bucket parameters), the performance metrics (e.g., throughput, delay, packet drop precedence), how non-conformant packets will be treated, and additional marking and shaping of the traffic. Given a DiffServ service definition, it is up to the service provider to design its network to meet the traffic treatment and behavior expected by the customer according to the SLA.

Figure 5-11 shows the basic steps involved in providing the DiffServ services. Steps 2 – 5 in the figure are internal to the service provider's network and are not directly visible to the customer. Customer's packets arrive at the router with DSCP marked (or unmarked). The router examines the DSCP' of the packets and classifies the packets by the Behavior Aggregation (BA) classification method. The packets classified into a particular BA are forwarded according to the Per Hop Behavior (PHB) defined for that BA. Each PHB is represented by a DSCP value and receives its unique packet forwarding treatment. The generic requirements discussed in Chapter 4, e.g., traffic policing, traffic shaping, packet discarding, active queue management, and packet scheduling, are applied as appropriate.

2.2 DiffServ architecture

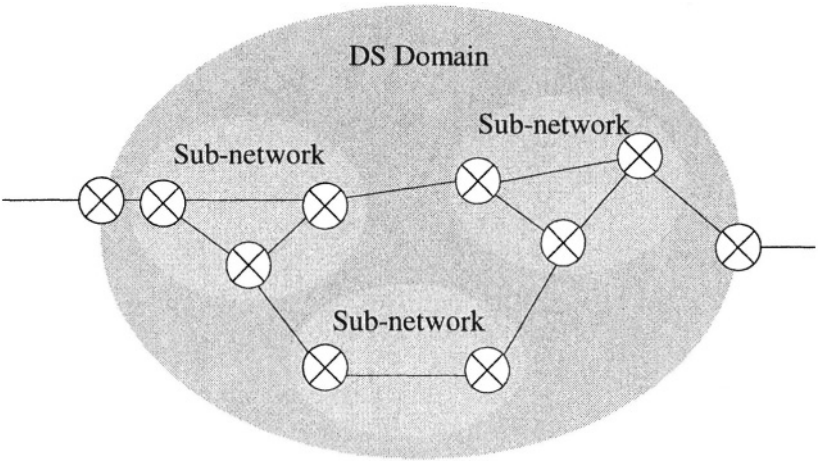


Figure 5-14. A DS domain and sub-networks.

In general, a "domain" in the IP network refers to a geographical area with a boundary over which a certain policy or capability is implemented. An IP network "domain," or an IP domain, is an IP network that is under the control of a single network administration authority. An IP domain may consist of several networks, which are geographically dispersed but under the same authority.

An IP network is considered DS-capable, if it is capable of providing DiffServ. An IP domain may have a DS-capable portion and a non-DS-capable portion. A DS domain is the DS-capable portion of an IP domain. Since a DS domain is a subset of a larger IP domain, which is under a single authority, a DS domain is also under the same single authority. Figure 5-12 illustrates an IP domain that contains both a DS domain and a non-DS domain.

Figure 5-13 shows a DS domain and its key elements. The key terms used in describing the DS architecture are defined in RFC 2475.[18] An IP

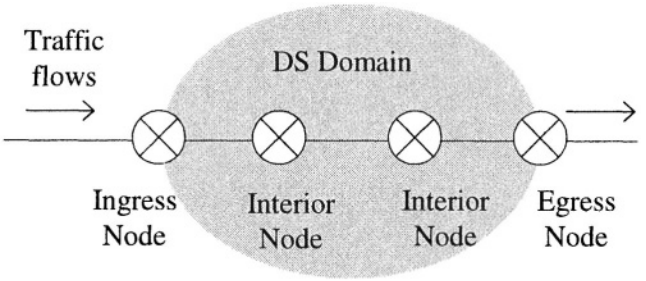


Figure 5-13. DiffServ domain.

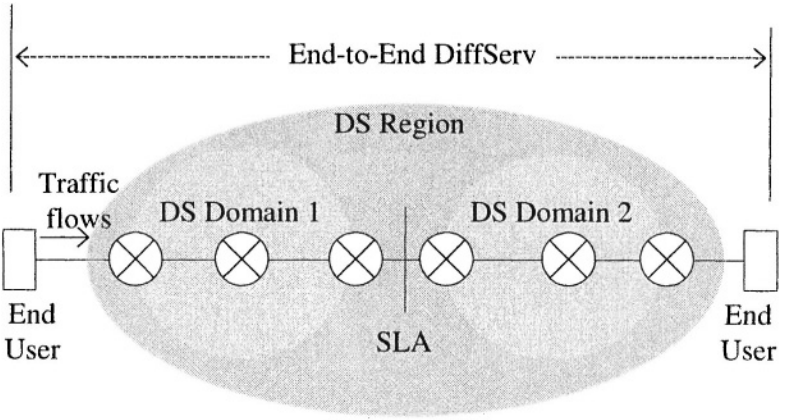


Figure 5-15. DS region.

node or device is said to be "DS-compliant," if it supports DiffServ. A DS node is an IP node that is DS-compliant. As an IP domain has a boundary, a DS domain is demarcated by a DS boundary. A DS node that is located at the DS boundary is referred to as a DS boundary node; and a DS node that is inside a DS domain, a DS interior node. DS boundary nodes perform certain specific functional requirements such as traffic policing.

There are two types of boundary node: the ingress node and the egress node. An ingress node is a DS boundary node that is at an ingress of a DS domain, and an egress node is a DS boundary node that is at an egress of a DS domain. Traffic enters a DS domain via an ingress node and leaves a DS domain via an egress node.

A DS boundary node is connected to either a DS interior node of the same DS domain, a DS boundary node of another DS domain or a node in a non-DS capable domain. A DS interior node is connected to either another DS interior node or a boundary node of the same DS domain: it is not directly connected to a node outside of the DS domain.

There is no requirement that a DS domain must consist of DS nodes only, i.e., DS-compliant nodes only. A DS domain can include non-DS-compliant nodes, and can still be DS-capable simply by making non-DS-compliant nodes unaware of DiffServ features. However, the more non-DS-compliant nodes are mixed in a DS domain, the less the DiffServ becomes effective.

A DS domain may consist of multiple networks under a single network authority as illustrated in Figure 5-14. The single network authority can impose and administer a common service provisioning policy across the sub-networks of the domain. How IP packets are treated in a DS-capable network is defined by the Per Hop Behaviors (PHB's). The PHB's are discussed later. Since a DS domain is under a single network authority, it is

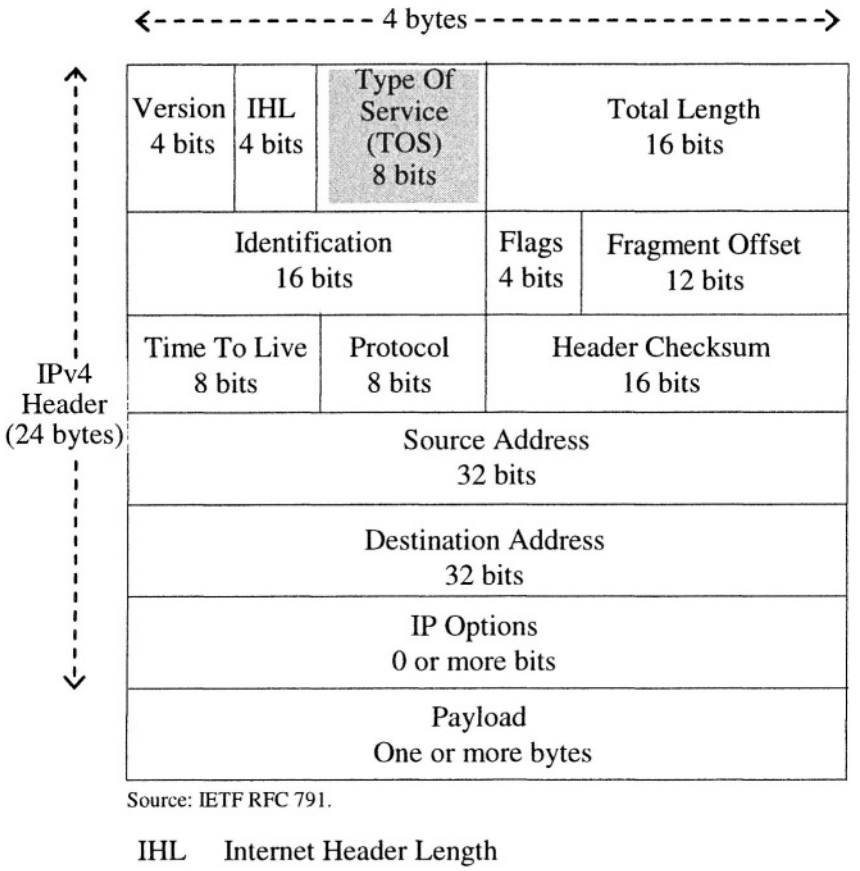


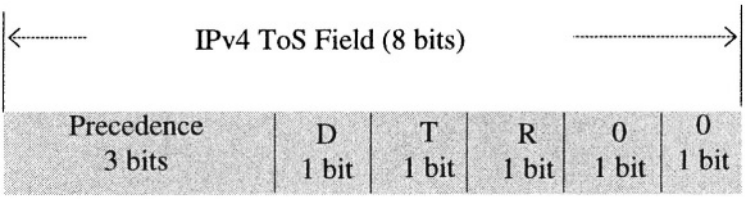
Figure 5-16. IPv4 header.

possible to implement a consistent set of Per Hop Behaviors (PHB's) implemented at the DS nodes.

Figure 5-15 shows a DS region. A DS region consists of one or more contiguous DS domains belonging to different administrative authorities. Therefore, a DS region can provide DiffServ over the IP routes spanning the networks under multiple authorities.

In general, individual DS domains operate with their own policies and Per Hop Behaviors (PHB's). Each DS domain may use its own DiffServ Code Point (DSCP) assignments of traffic types. To provide DiffServ over a DS region, the peering DS domains within the DS region must establish a Service Level Agreement (SLA) at the interface between the DS domains.

As part of the SLA, the authorities must agree on how the DSCP's are interpreted and how the traffic is handled within the respective DS domains and how the traffic is handed over from one DS domain to another. An



Source: IETF RFC 791.

Figure 5-17. Type of Service (TOS) field in the IPv4 header.

alternative approach is for the constituent DS domains of a region to adopt a common service provisioning policy and a common set of PHB and DSCP mappings. Figure 5-15 shows Domain 1 and Domain 2 belonging to two different administrations, e.g., ISPs. To provide DiffServ on an end-to-end basis, an SLA needs to be agreed upon between the administrative authorities.

2.3 DiffServ packet marking

DiffServ uses the Type of Service (ToS) field of the IPv4 header and the Traffic Class (TC) field of the IPv6 header for marking packets. When the IPv4 and IPv6 routers operate in the conventional mode and do not recognize DiffServ, the ToS and the TC field are used as they are originally intended to be used.

When the same IPv4 and the IPv6 routers support DiffServ and operate as a DS node, the ToS and the TC fields are overridden and redefined as the DiffServ (DS) field. In other words, there is no separate DS field defined in the IP headers: the existing fields, i.e., the ToS and TC fields, are used for DiffServ packet marking.

2.3.1 Packet marking in conventional routers

Figure 5-16 shows the IPv4 header, which includes an eight-bit field referred to as the Type of Service (ToS) field. Figure 5-17 shows the ToS field. In a conventional router (i.e., non-DiffServ router), the eight bits of the ToS field are defined per RFC 791[19] as shown in Tables 5-4 and 5-5. The first three bits (bits 0, 1, and 2) of the ToS fields are referred to as the IP precedence bits. Table 5-4 shows the IP precedence bit setting and the corresponding traffic types.

RFC 791[19] specifies that the network control precedence designation (‘111’) be used within a private network only; and that the inter-network control designation (‘110’), by gateway control originators only.

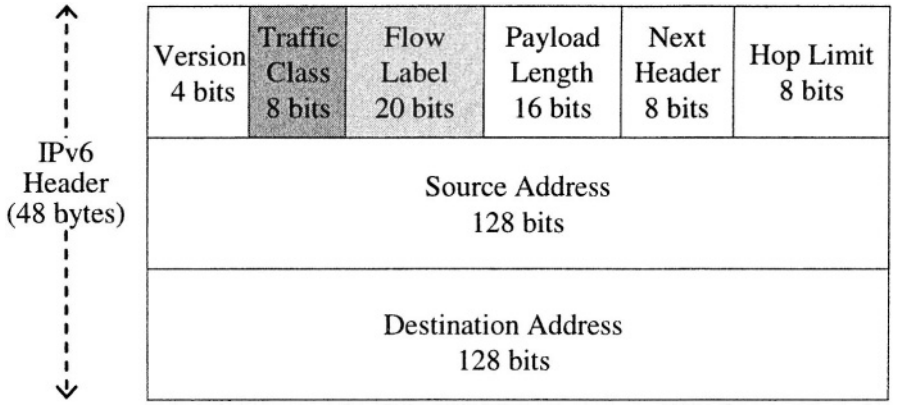


Figure 5-18. IPv6 header.

The next three bits (bits 3, 4 and 5) of the ToS field define the performance characteristics of the service generating the packets. Table 5-5 shows the settings of the D-bit, T-bit and R-bit of the ToS field and the corresponding meanings. The last two bits (bits 6 and 7) of the ToS field are reserved for future use.

Table 5-4. IP precedence bits.

IP precedence bit settings	Traffic Type
111	Network control
110	Inter-network control
101	Critical/emergency
100	Flash override
011	Flash
010	Immediate
001	Priority
000	Routine

Table 5-5. Performance indicators.

Bit Setting	D-bit	T-bit	R-bit
0	Normal delay	Normal throughput	Normal reliability
1	Low delay	High throughput	High reliability

Table 5-6. Blocking of DSCP values.

Pool	Code Point Space	Assignment Policy
1	xxxxx0	Standard Action
2	xxxx11	Experimental/Local use
3	xxxx01	Experimental/Local use*

* These values may be utilized for future Standards Action allocations as necessary. 'x' denotes '0' or '1.' Source: IETF RFC 2474.[20]

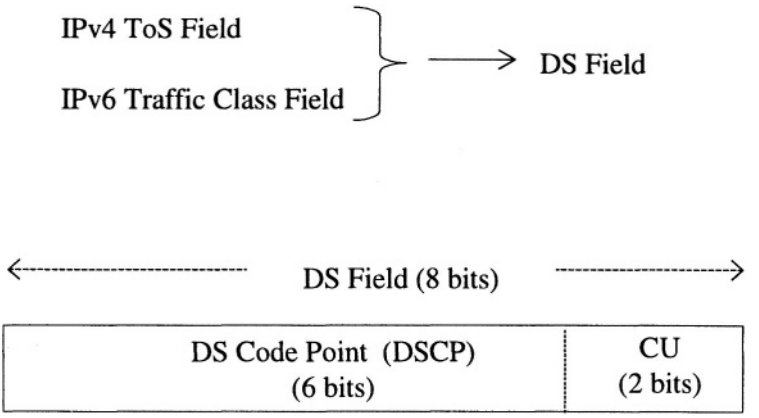


Figure 5-19. DiffServ (DS) field.

Figure 5-18 shows the IPv6 header. It contains the eight-bit Traffic Class (TC) field and the 20-bit Flow Label (FL) field. Both of these fields are relevant to QoS. However, at the time of this writing, no significant applications have been identified for the FL field. The TC field provides the similar capability as the ToS field of the IPv4 header.

2.3.2 DiffServ (DS) field

When a router is used for DiffServ as a DS node, the same eight-bit fields, the ToS field in IPv4 and the TC field in IPv6, are overridden as the DiffServ (DS) field. Figure 5-19 illustrates overriding of the ToS field and the TC field by the DS field and the definition of the DS field.

Of the eight bits of the DS field, the first six bits are used for marking DiffServ packets and the last two bits are reserved for future use. The six bits used for marking DiffServ packets are referred to as the DS Code Points (DSCP's). Packet marking in DiffServ, therefore, is to set the DSCP's.

2.3.3 DiffServ Code Points (DSCP's)

The six bits in the DSCP field provide 64 possible permutations of DSCP value. RFC 2474[20] blocks the 64 possible DSCP values into three groups referred to as "pools" as shown in Table 5-6.

The last bit (i.e., the sixth bit) of a Pool 1 DSCP is fixed at zero. The other five bits of a Pool 1 DSCP can be either '0" or '1.' Hence, there are 32 Pool 1 DSCP's.

The Pool 1 DSCP's require the IETF standard actions and are universally recognized. The last two bits of a Pool 2 DSCP are fixed at '11'. The remaining four bits allow 16 permutations of Pool 2 DSCP's. The Pool 2 DSCP's do not require standard actions and are used for experimental and local purposes. DiffServ packets within a private Intranet can be marked by the Pool 2 DSCP's. The Pool 2 DSCP's have only local significance and are not recognized outside the Intranet.

The Pool 3 DSCP's always end with '01' and there are 16 Pool 3 DSCP's. The Pool 3 DSCP's are similar to the Pool 2 DSCP's in that they are intended for experimental and local use; the difference, however, is that the Pool 3 DSCP's can be taken away for standard actions, if necessary.

There is a need to make the DS nodes backward compatible: compatible with the conventional routers. The conventional routers running IPv4 may support the IP precedence bits of the ToS field as defined by RFC 791.[19] These three IP precedence bits allow eight different traffic types to be recognized by the conventional routers as shown in Table 5-4.

Eight DSCP's from Pool 1 are used to designate the IP precedence traffic types of the conventional routers. The eight Pool 1 DSCP's used for this purpose are referred to as the Class Selector Code Points (CSCP's). The last three bits of a CSCP is fixed at '000'. Hence, the CSCP's have the form 'xxx000', where 'x' is either '0' or '1.'

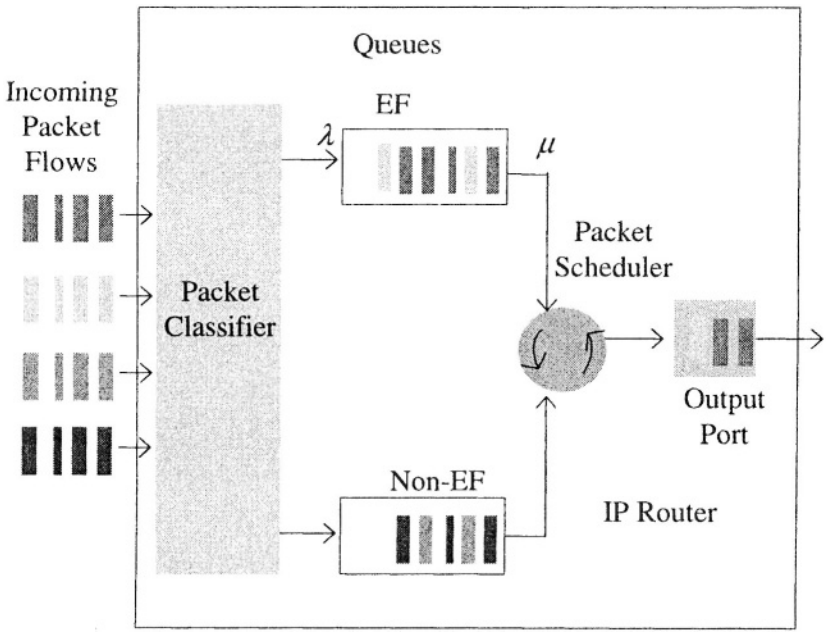


Figure 5-20. EF implementation example.

When there is no predefined packet treatment, a "default" treatment is given to the packets by a DS node. The default DSCP is '000000'. The packets with the default DSCP may be given a best effort treatment.

Of the eight CSCP's, '111000' and '110000' are given preferential treatment over a default treatment of '000000'. These two CSCP's correspond to the IP precedence markings '111' and '110', which are commonly used for network control and inter-network control signaling in the conventional nodes and the preferential treatments of the two CSCP's over a default treatment are to preserve the common use of IP precedence values.

2.4 Per-Hop Behaviors (PHB's)

DiffServ uses the Behavior Aggregate (BA) classification method. In the BA classification method, packets are classified based on their DSCP values only and no other parameters. The packets classified into a BA are given the same forwarding treatment.

How packets are treated at a router involves a definition of the expected packet forwarding behavior that is externally observable and its internal implementation within the router. A PHB is externally observable in the sense that it defines the behavior observed outside of a router. A PHB is a technical description internal to a network and is not observable to an end user. A DiffServ service description discussed earlier is what is observed by the end user, not a PHB.

DiffServ is fundamentally per-hop based and is based on defining PHB's for individual routers. However, to provide DiffServ across a DS domain, PHB's for individual routers may be designed in such a way that the overall end-to-end QoS is best provided. Properly designed PHB's combined with appropriate traffic conditioning, admission control and network provisioning, end-to-end DiffServ may be provided.

A service provider uses a PHB to determine the bandwidths and other network resources required to make sure that the PHB is realized. Therefore, a PHB may be defined in terms of the network resources required (e.g., bandwidth, buffer size), a relative priority of the PHB compared with other PHB's at the same router, and the expected performance in terms of delay, jitter, and packet loss.

To produce the packet forwarding behavior defined by a PHB, the internal mechanism of a router must then implement the appropriate AQM and packet scheduling techniques and other requirements discussed in Chapter 4. A PHB of a router does not specify the internal implementation mechanisms of the router. A variety of different implementation mechanisms may be used to realize a PHB. Therefore, an implementation of

a PHB is typically manufacturer's proprietary mechanism and, in general, is not standardized.

There are two types of standard PHB's: Expedited forwarding (EF) PHB and Assured forwarding (AF) PHB.

2.4.1 Expedited Forwarding (EF) PHB

The Expedited Forwarding (EF) PHB was specified initially by RFC 2598,[21] which has later been replaced by RFC 3246.[22] The DSCP value recommended for the EF PHB is '101110'. With the EF PHB, the packets are forwarded with low loss, low delay and low jitter. The EF PHB requires a guaranteed amount of output port link bandwidth to produce the low loss, low delay, and low jitter behavior.

The EF PHB is possible if the output port link bandwidth plus the buffer size and other network resources dedicated to the EF packets allow the service rate, μ, of the packet scheduler in the router for the EF packets on a given output port to exceed the packet arrival rate, λ, at that port, independently of the traffic load on other non-EF PHB's. The queuing theory basis for this is discussed in Chapter 2. This means that the packets with the EF PHB are treated with a pre-allocated amount of output bandwidth and a priority that will guarantee the minimum loss, minimum delay and minimum jitter forwarding behavior.

The EF PHB is appropriate for circuit emulation, private leased line emulation, and real-time services such as voice and video, which are not tolerant to high values of loss, delay and jitter.

Figure 5-20 shows an example of the EF PHB implementation. It is a simple priority queuing scheduling mechanism. At the edges of a DS domain, the EF packet flows are policed according to the values agreed by an SLA. The EF queue in the figure should be given an enough allocation of the output port bandwidth so that the service rate, μ, of the EF queue exceeds their arrival rate, λ. To provide the EF PBH across a DS domain end-to-end, the bandwidths at the output ports at the core routers should be all pre-allocated to ensure the requirement of $\mu > \lambda$. This is accomplished by a manual provisioning process.

In the figure, the EF packets are placed in the priority queue. As long as the queue can operate with $\mu > \lambda$, the lower priority queue may be visited even if there are packets in the EF queue. In this way, the starvation of the non-EF queue can be avoided. This type of priority queue is a rate limited priority queue.

An alternative to this is to use the strict priority queuing scheduling mechanism. With this method, care must be taken to avoid the starvation

problem of the non-EF queues. Another alternative of implementing the EF PHB is to use a variant of WFQ.

Since the EF is used primarily for real time services such as voice and video and since real time services use UDP instead of TCP, the RED is in general not appropriate for the EF queues. The RED should not be used for the EF queues because the applications using UDP would not respond to the random packet discarding and the RED will drop packets unnecessarily. In the future, an AQM appropriate for UDP may be introduced, an AQM similar to the ECM method used for TCP. Until then, the default strategy, i.e., the tail drop, is in fact a better strategy than any AQM for the EF PHB.

2.4.2 Assured Forwarding (AF) PHB

The Assured Forwarding (AF) PHB is specified by RFC 2597.[23] The objective of the AF PHB is to deliver the packets reliably and therefore delay and jitter are not as important as packet loss. The AF PHB is appropriate for non-real time services such as TCP applications.

The AF PHB is derived from the general class of Active Queue management (AQM) referred to as the RED with the In/Out bit or RIO. However, a PHB does not specify any specific implementation in a router and the AF PHB does not specify a RIO. It simply describes a forwarding treatment characteristic of a RIO. How the AF PHB behavior is produced by a specific implementation is again up to the network provider.

The basic concept of the RIO is to divide packets into "in-profile" and "out-of-profile" packets depending on whether a packet conforms to a policing rule or not and to mark them accordingly. The RED in a router uses these "in/out" markings when it randomly selects packets for dropping: the "in-profile" packets will have lower probabilities of being selected for dropping than "out-of-profile" packets."

The AF PHB is based on an extension of the binary marking of the RIO to three drop precedence levels. The AF PHB first defines four forwarding classes, AF1, AF2, AF3 and AF4. Within each of these AF classes, packets are then classified into three subclasses with three levels of drop precedence, yielding a total of 12 AF subclasses.

Table 5-7 shows the four AF classes and the 12 AF subclasses and the DSCP values for the 12 subclasses assigned by RFC 2597.[23] RFC 2597[23] also allows adding more than three drop precedence levels for local use. However, these additional drop precedence levels will have local significance only.

The AF PHB ensures that packets are forwarded with a high probability of delivery as long as the aggregate AF traffic stays within the limit of the rate agreed to in an SLA. If the AF traffic at an ingress port exceeds the pre-

assigned policing rate, the non-conformant or "out-of-profile, packets are not delivered with as high a probability as the conformant traffic or in-profile packets. When there is network congestion, the out-of-profile packets are dropped before the conformant packets are dropped.

RFC 2597[23] on AF PHB does not specify what kind of service the AF PHB should be used for. However, the AF PBH provides a mechanism by which a service provider can design a DiffServ service with different levels of service grades. One example given in RFC 2597[23] is the "Olympic Service," where the packets are classified into three grades of service, "Gold," "Silver" and "Bronze" using three of the four AF classes, e.g., AF3, AF2, and AF1. The example uses three AF classes as the Olympics uses only three grades of medals. A "Platinum" class may be added by using AF4 and a four-grade service may be designed.

Once the service grades are defined by using the AF classes, the quantitative and qualitative differences between the AF classes can be realized by allocating different amounts of bandwidths and buffer space to the four AF classes.

As with the EF, policing individual flows of an AF class is needed at the edges of a DS domain. The appropriate amounts of output port bandwidths are allocated at the routers across a DS domain by manual provisioning.

Unlike the EF, most of the AF traffic is non-real time traffic using TCP, and the RED is an appropriate AQM to use for the AF PHB. The four classes of the AF PHB can be implemented as four individual queues. The output port bandwidth is divided to the four AF queues. Within each AF queue, the packets are marked by three "colors" or three levels of drop precedence. Figure 5-21 shows an example of AF implementation by the WFQ scheduling mechanism.

Table 5-4. AF DSCP's.

PHB class	PHB subclass	Drop precedence	DSCP
AF4	AF41	Low	100010
	AF42	Medium	100100
	AF43	High	100110
AF3	AF31	Low	011010
	AF32	Medium	011100
	AF33	High	011110
AF2	AF21	Low	010010
	AF22	Medium	010100
	AF23	High	010110
AF1	AF11	Low	001010
	AF12	Medium	001100
	AF13	High	001110

Source: IETF RFC 2597.[23]

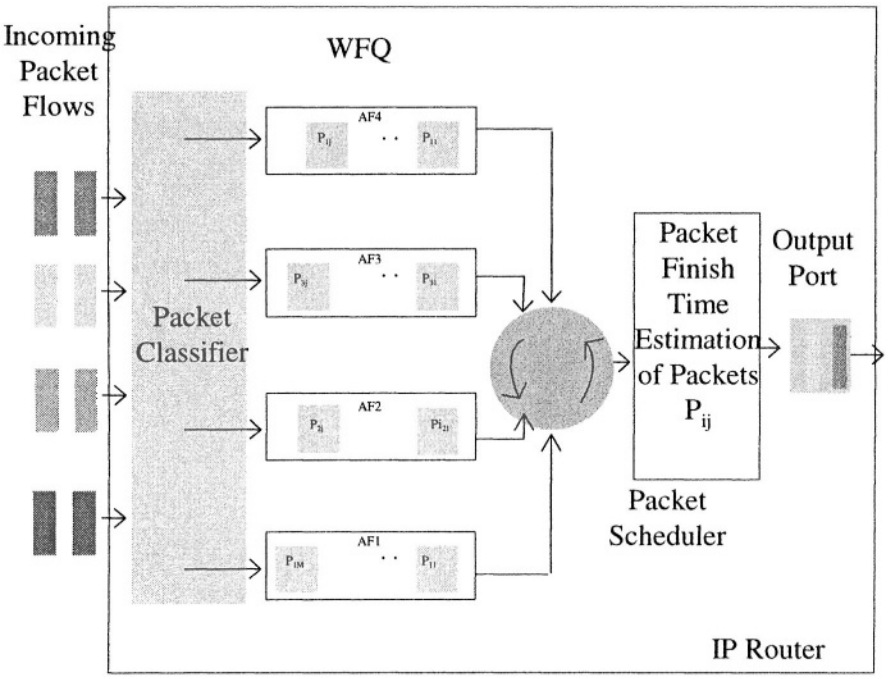


Figure 5-21. AF implementation example.

Out of 32 Pool 1 DSCP's defined in Table 5-6, 21 DSCP's have already been standardized as follows: one for EF PHB, 12 for AF PHBs and eight CSCP's. There are 11 Pool 1 DSCP's that are still available for standard actions.

3. EXERCISES

3.1 Problems

1. Name three object classes in a PATH message.
2. Name three object classes in a RESV message.
3. Name one object type in the FILTER SPEC object class.
4. Which one of the following DSCP values is a Pool 2 DSCP?
   110100 110101 111111

5. Which one of the following DSCP values is a Pool 1 DSCP?
   111001 110111 000000
6. Which one of the following DSCP values is a CSCP?
   111010 000000 111100
7. Why is RED not appropriate for an EF queue?
8. For a TCP flow, which of EF and AF would be appropriate?
9. How many bits are used to define an IP precedence?

3.2 Solutions

1. SESSION, RSVP_HOP, SENDER_TEMPLATE
2. SESSION, RSVP_HOP, STYLE
3. IPv6 FILTER_SPEC object (C-Type = 2)
4. 111111
5. 000000
6. 000000
7. RED has no effect on UDP.
8. AF.
9. 3 bits.

Chapter 6

QOS IN ATM NETWORKS

ATM is a connection-oriented packet network. Because ATM services are provided based on connections, it is, in general, easier to provide QoS in ATM networks than in the connectionless IP networks. In a nutshell, QoS in ATM networks is provided by specifying the performance requirements for the requested logical connections along with the amount of bandwidth needed to meet the pre-specified performance level and administering a Connection Admission Control (CAC) to make sure that the performance of the current connections are not degraded by adding new connections.

After a brief discussion of the genesis of ATM, this chapter discusses the following topics:

- ATM protocols
- ATM virtual connections
- ATM traffic descriptors
- ATM QoS parameters
- ATM service categories
- ATM Connection Admission Control

1. BACKGROUND

1.1 Genesis of ATM

ATM was developed to provide high-speed integrated services with a single infrastructure for data, voice and video. The main driving force behind the ATM development was the deficiency of ISDN in meeting this need. ISDN was considered inadequate to meet the needs of multi-media traffic. For example, ISDN Basic Rate Interface (BRI) operating at 128 kb/s

ATM Chronology

Mid 1980's	Fundamental aspects addressed: fixed-length "cells"
1989	CCITT key standards agreements Cell size Header format Protocol structure, e.g., ATM Adaptation layer
1990's	CCITT (i.e., ITU-T) Mature definition of traffic and congestion controls OAM cell flows Cell transfer performance, call processing performance
ATM Forum	Started about 1995 Defined ATM service classes

and Primary Rate Interface (PRI) operating at 1.5 Mb/s were both insufficient for broadcast-quality video.

ITU-T, previously known as CCITT, standardized B-ISDN using ATM as the basic transmission and networking technology. In mid 1980s, some of the fundamental aspects of ATM were addressed and it was decided that a fixed size packet format, referred to as "cell," be used. The cell size was determined based on the consideration of voice transmission of 64 kb/s. The list above shows some of the key events in the ATM chronology.

ATM is a connection-oriented packet transport service. Packet transmission is based on ATM cells, and packet switching, using virtual circuits. Since an ATM service is a connection-oriented service, the end-user must request a "connection" to the intended receiver. As part of the connection request, the user must specify a set of "traffic descriptors." In ATM, QoS is assured by provisioning sufficient bandwidth per virtual connections (VCs) and exercising the Connection Admission Control (CAC) of VC requests.

1.2 ATM network interfaces

Figure 6-1 shows the interface types in the ATM network. The User-to-Network Interface (UNI) is the interface between the end user and the ATM network that the user subscribes to. The Network-to-Network Interface (NNI) is the interface between one ATM switch and another ATM switch within the same carrier's ATM network. The Broadband ISDN (BISDN)

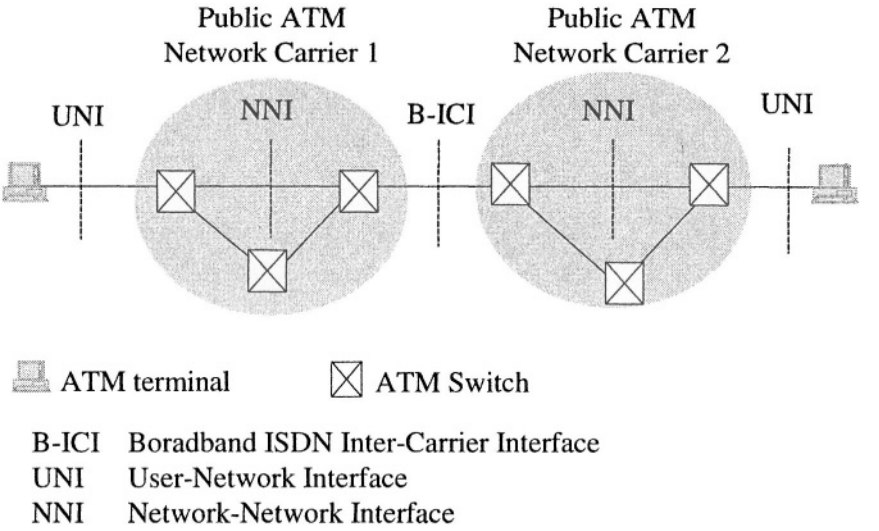


Figure 6-1. ATM interfaces.

Inter Carrier Interface (B-ICI) is the interface between two public ATM carriers.

The ATM signaling protocol is private or public depending on the type of interface over which the signaling is carried out. If the UNI is between an end-user terminal and a public ATM network, the public UNI signaling protocol is used. If the UNI is between an end-user terminal and the end-user's private ATM network, the private UNI signaling protocol is used.

The ATM signaling protocol used between the ATM nodes within the same ATM network is the private NNI signaling protocol, which is referred to as the Private NNI or PNNI protocol. Between the ATM nodes in two different public ATM networks, the public NNI protocol is used. This public NNI signaling protocol is referred to as the B-ICI protocol.

2. ATM PROTOCOLS

Figure 6-2 shows the ATM protocol stack and where the protocols fit in the seven layer OSI protocol stack. As shown in the figure, the ATM protocol layer occupies Layer 2, or the link layer. The ATM layer is further divided into sub-layers: the lower sub-layer is the ATM cell layer; the higher sub-layer, the ATM Adaptation Layer (AAL). The AAL is further divided into the Segmentation and Re-assembly (SAR) layer and the Convergence Sublayer (CS).

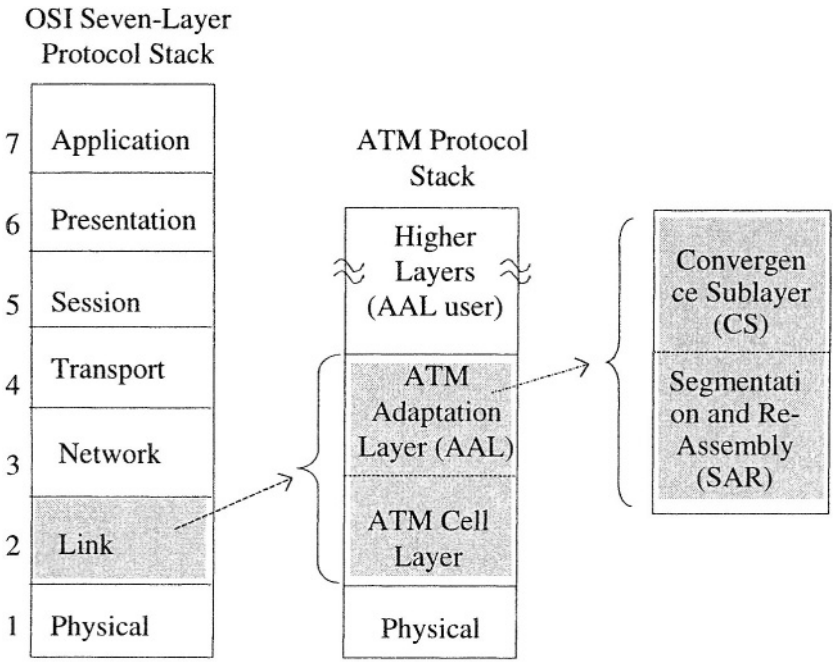


Figure 6-2. ATM protocol stack.

2.1 ATM cell layer

Figure 6-3 defines the ATM cell format of the ATM cell layer. Unlike the IP packet format, in which packets can have a variable length, ATM uses fixed-length packets called "cells." An ATM cell is 53-octet long: a five-octet header plus a 48-octet payload field. One of the main factors in the choice of the 53-octet cell size had been voice performance. As discussed in Chapter 3, the main source of the ATM packetization delay is the time to fill the ATM cell payload, i.e., the longer the cell length, the larger the ATM packetization delay. The 53-octet cell length had been arrived at after a long study and deliberation by the standard body.

As shown in Figure 6-3, the ATM cell format is slightly different between the UNI and the NNI. The four-bit Generic Flow Control (GFC) field is present only at the UNI and is not present at the NNI. The GFC field is used as a place holder to provide the network a capability to control the traffic flow from the user to the network.

Another difference in the ATM cell format between the UNI and the NNI is the length of the VPI field. At the UNI, eight bits are allocated for the VPI while, at the NNI, 16 bits are allocated to the VPI. This is because more

VPI's are needed at the NNI than at the UNI. The VCI field length is the same at the UNI and the NNI.

The eight-bit Header Error Control (HEC) field contains the check sum calculated for the 32 bits of the ATM header information. The three-bit Payload Type Identifier (PTI) field indicates the type of payload carried in

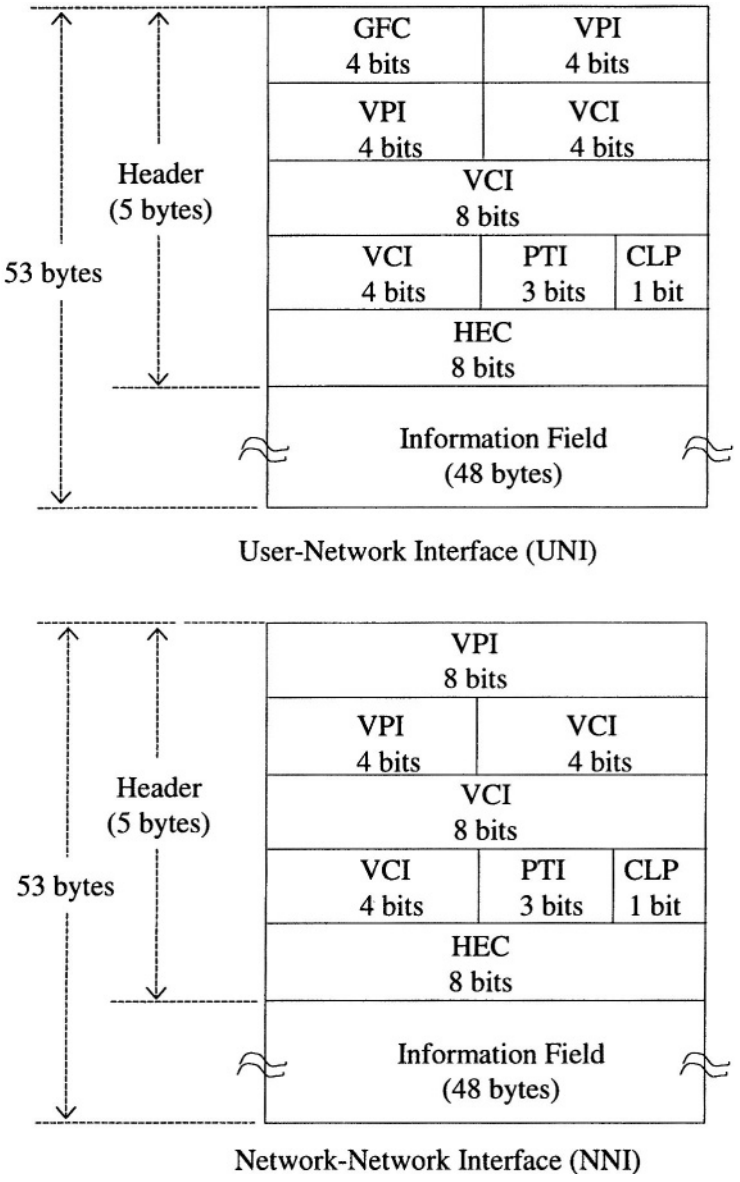


Figure 6-3. ATM cell formats.

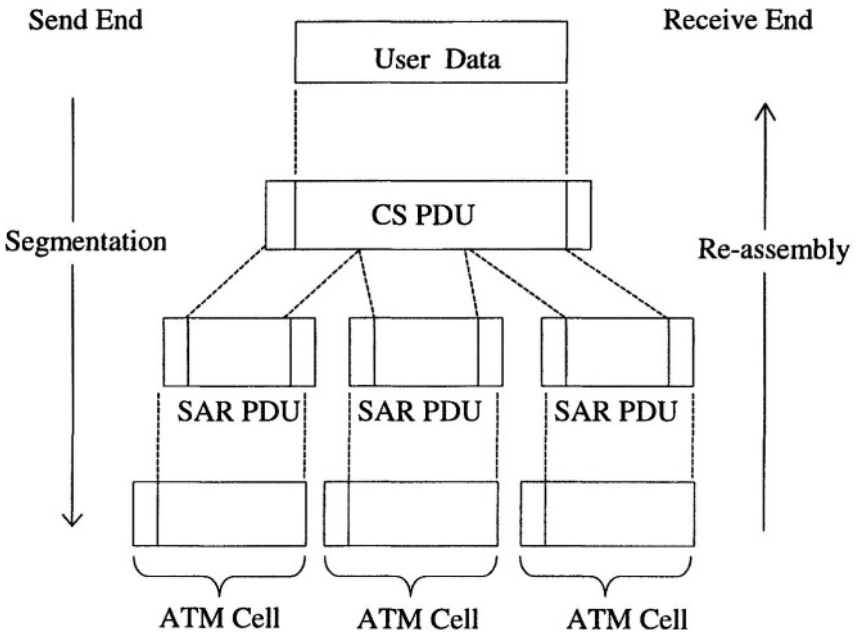


Figure 6-4. ATM cell segmentation and re-assembly.

the ATM cell. The three bits of the PTI field provide eight payload types. Four PTI codes are used for user information cells; and the remaining four PTI codes, for Operations, Administration & Maintenance (OAM) cells.

For the ATM QoS, the Virtual Path Identifier (VPI) and the Virtual Channel Identifier (VCI) fields are important and will be discussed at length as part of the ATM Virtual Connections later in this chapter.

The one-bit Cell Loss Priority (CLP) field is used to mark the cells for cell dropping priorities during congestion. The cells with the CLP bit set to zero are normal cells; those with the CLP bit set to one are the cells eligible for discarding.

2.2 ATM Adaptation Layer (AAL)

As shown in Figure 6-2, the ATM Adaptation Layer (AAL) is divided into two sub-layers: the Convergence Sub-layer (CS) and the Segmentation and Re-assembly (SAR) sub-layer. The Convergence Sub-layer (CS) handles transmission errors, lost and misinserted cells, timing relation between source and receiver, and cell delay variations; and the SAR sub-layer handles the packet segmentation into cells at the send end and the packet re-assembly from cells at the receive end. Figure 6-4 shows the user

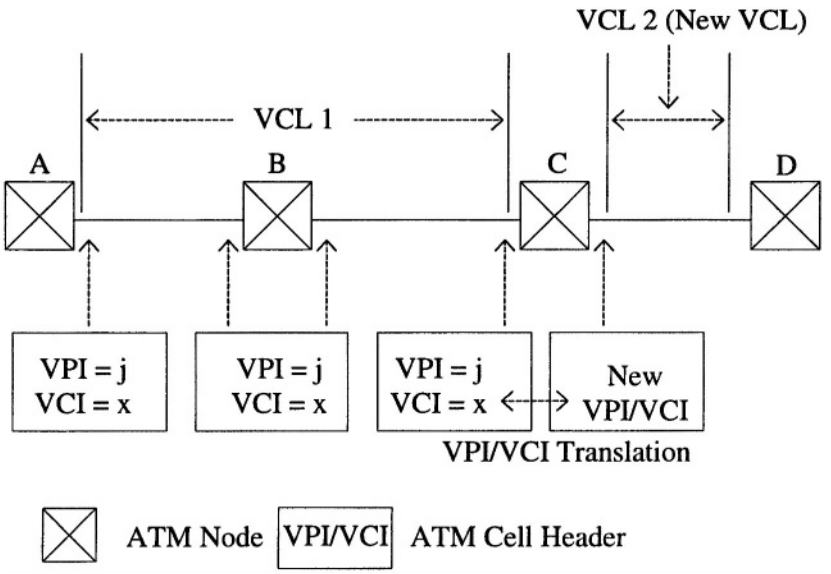


Figure 6-5. Virtual Channel Link (VCL).

data segmentation into ATM cells at the send end and re-assembly of the ATM cells into the user data at the receive end.

3. ATM VIRTUAL CONNECTIONS

A key to understanding QoS in the ATM network is to understand how virtual connections are created in the ATM network and how the bandwidths are allocated and managed for the virtual connections. This section discusses:

- Virtual links
- Virtual connections
- Permanent and switched virtual connections

3.1 The Virtual Channel and the Virtual Path

The Virtual Channel (VC) and the Virtual Path (VP) are defined in ITU-T I.113[24] as follows. The Virtual Channel (VC) is "a concept used to describe unidirectional transport of ATM cells by a common unique identifier value." The Virtual Path (VP) is "a concept used to describe unidirectional transport of ATM cells by a common unique identifier value."

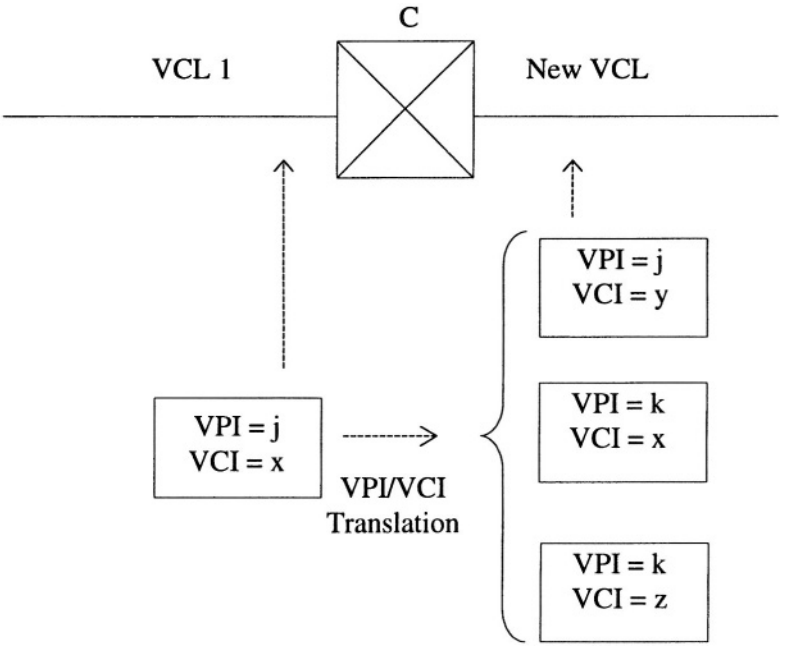


Figure 6-6. VPI/VCI Translation.

A VC is analogous to a trunk circuit in the circuit switched network and a VP, a trunk group. The Virtual Channel Identifier (VCI) and the Virtual Path Identifier (VPI) in the ATM header are used to identify the VC and the VP. Since VC's are in general grouped into VP's, a VC is uniquely identified by the combination of a VPI and a VCI: a VCI alone does not completely define a VC. The VCI and the VPI both have local significance, and, consequently, the VC and the VP also have local significance, i.e., the VC and the VP are locally visible "concepts" for organizing ATM cells into "channels" and "paths."

It is helpful to note the distinction between a link and a "connection." A connection connotes an end-to-end entity, while a link, a component of a connection. Typically, a connection is created by concatenating a number of links.

3.2 Virtual links

There are two types of virtual links: the Virtual Channel Link (VCL) and the Virtual Path Link (VPL). ITU-T I.113[23] defines the VCL and the VPL as follows. The Virtual Channel Link (VCL) is "a means of unidirectional

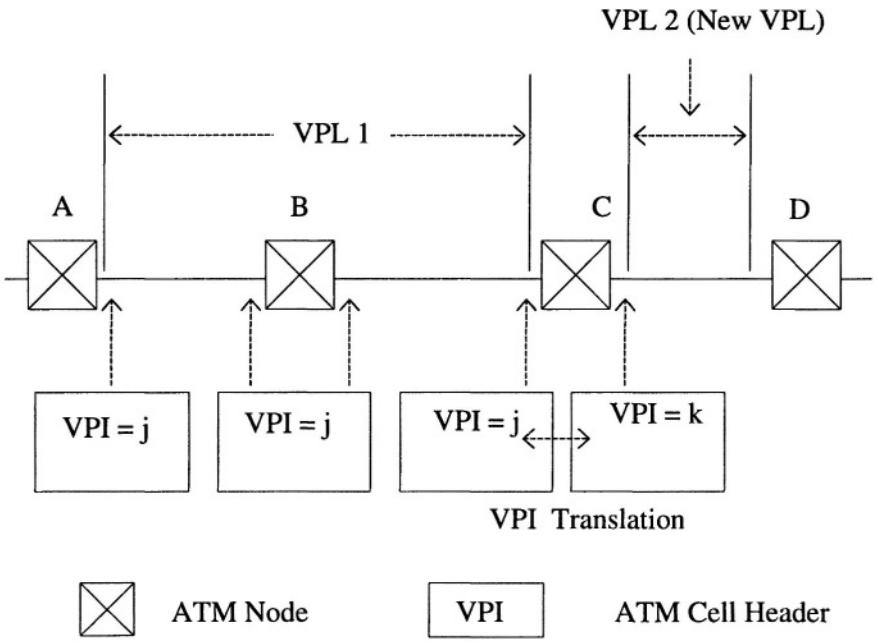


Figure 6-7. Virtual Path Link (VPL).

transport of ATM cells between a point where a Virtual Channel Identifier (VCI) value is assigned and the point where that value is translated or removed." The Virtual Path Link (VPL) is "a group of VCL's, identified by a common value of the Virtual Path Identifier (VPI), between the point where a VPI value is assigned and the point where the VPI value is translated or removed."

Figure 6-5 illustrates the VCL. The VCL shown in the figure carries the ATM cells with VPI = j and VCI = x from ATM Node A to ATM Node C. At the intermediate node B, the VPI/VCI is not changed and Nodes A, B and C are part of the same VCL. At Node C, the VPI/VCI is translated from "j/x" to some other values and Node C is the termination point of the VCL.

The translation of VPI/VCI at Node C can be any of the following three possibilities as shown in Figure 6-6. In the first case, the new VCL is within the same VPL identified by VPI = j. The second and the third case show a new VCL in a different VPL identified by VPL = k. The second case makes a point that, once the VPI changes (from j to k), the VCI does not need to change to define a new VC (i.e., VCI = x) because the combination of VPI/VCI uniquely identifies the VCL.

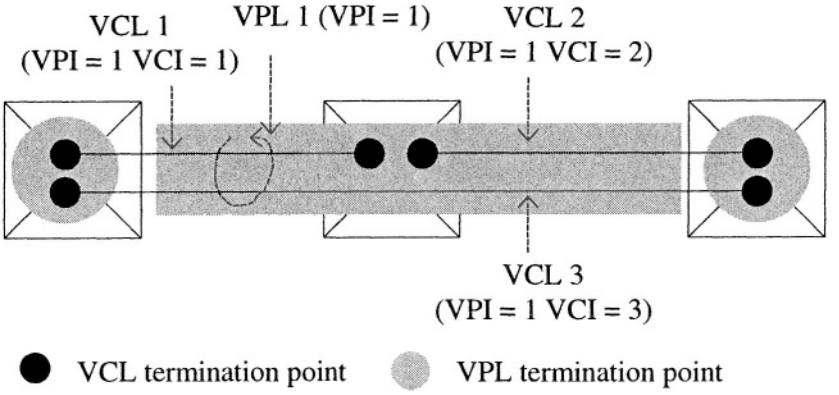


Figure 6-8. Relationship between VCL and VPL.

To identify a VPL, it is sufficient to specify a VPI. All ATM cells with a same VPI share the same VPL. Figure 6-7 shows the VPL. All cells with VPI = j are put on the VPL 1 from Node A to Node C. At Node D, a new VPL is defined by translating the VPI from j to k.

Figure 6-8 illustrates the relationship between the VCL and the VPL. The span of a VCL cannot extend beyond that of a VPL. A VPI change terminates not only the VPL at that point but also the VCL's within the VPL at the same point because the VPI/VCI combination that identifies a VCL changes with a VPI change. On the other hand, a VCL within a VPL can terminate before the VPL terminates because a VCI change with the same VPI terminates the VCL but it does not terminate the VPL.

3.3 Virtual Connections

The Virtual Connections are created by concatenating the Virtual Links. Like the Virtual Links, there are two types of Virtual Connections: the Virtual Path Connection (VPC) and the Virtual Channel Connection (VCC).

3.3.1 Virtual Path Connection (VPC)

A Virtual Path Connection (VPC) is created by concatenating VPL's as illustrated in Figure 6-9. A VPC is identified by a pair of VPI values at the ingress and egress ports. Given the ingress/egress VPI pair, the intermediate nodes make the appropriate translation of incoming VPI's into outgoing VPI's to create the desired VPC between the pair of VPI's. Use of VPC's simplifies the network architecture, increases the network performance and

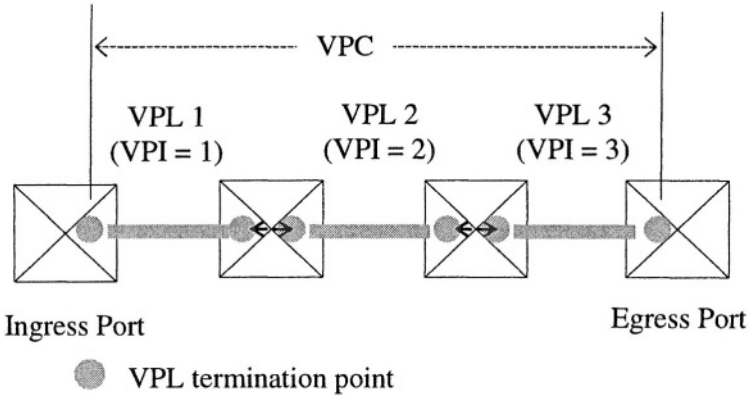


Figure 6-9. Virtual Path Connection (VPC).

reliability, minimizes the connection setup time, and allows capacity engineering.

3.3.2 Virtual Channel Connection (VCC)

A Virtual Channel Connection (VCC) is created by concatenating VCL's. A VCC is identified by a pair of VPI/VCI values at ingress/egress ports. The pair of VPI/VCI values is paired by pre-provisioning through the

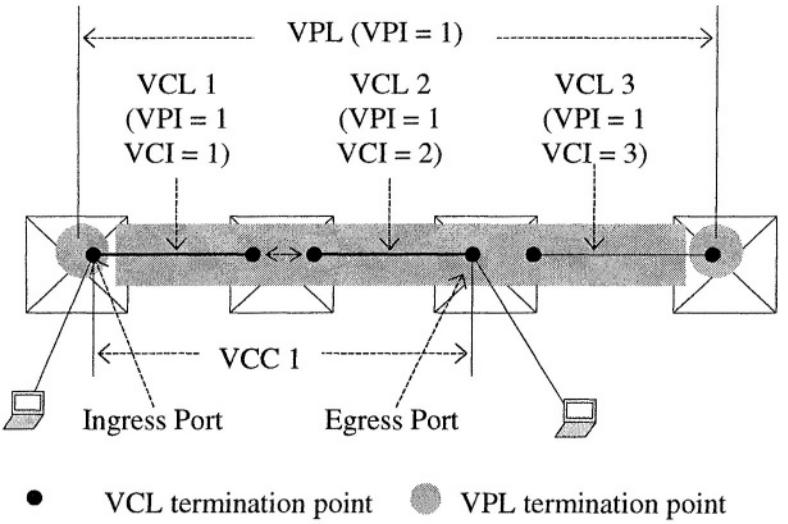


Figure 6-10 Virtual Channel Connection (VCC) within a VPL.

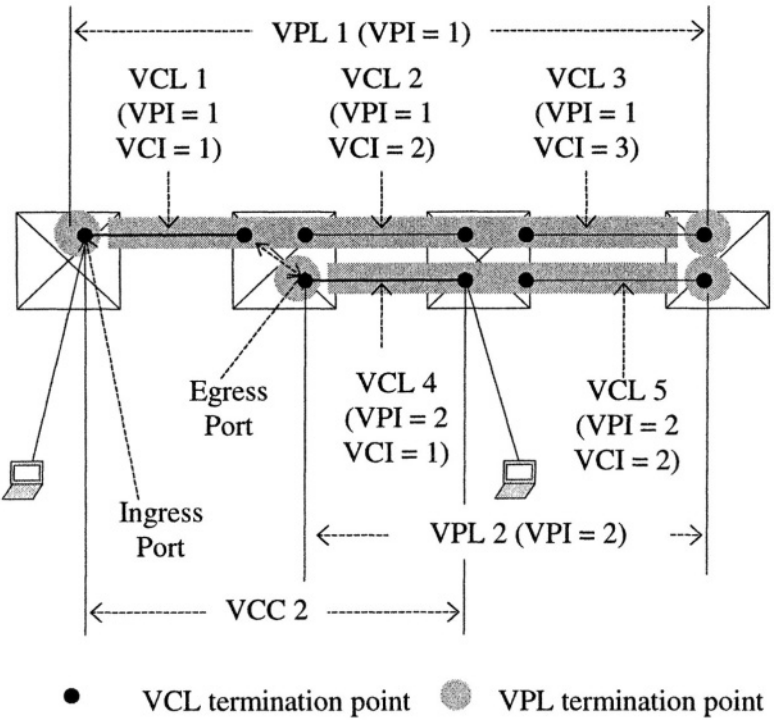


Figure 6-11. VCC created from VCL's from different VPL's.

core network.

Figure 6-10 shows a VCC created by concatenating VCL's within a same VPL: VCL 1 and VCL 2. Note that VCL 1 and VCL 2 have the same VPI = 1. Figure 6-11 shows a VCC created by concatenating VCL's from different VPL's: VCL 1 in VPL 1 and VCL 4 in VPL 2. A VCC extends between two points where an ATM adaptation layer is accessed, i.e., a logical port.

3.4 Permanent Virtual Connection (PVC)

A Permanent Virtual Connection (PVC) is a virtual connection (VC) that lasts for a long period of time. It is established and removed through a provisioning process. A PVC is set up before the connection is used. A PVC is "active" until it is de-provisioned even though it may not be actually used.

As there are two types of VC, which are VCC and VPC, there are two types of PVC:

- Permanent Virtual Channel Connection (PVCC)

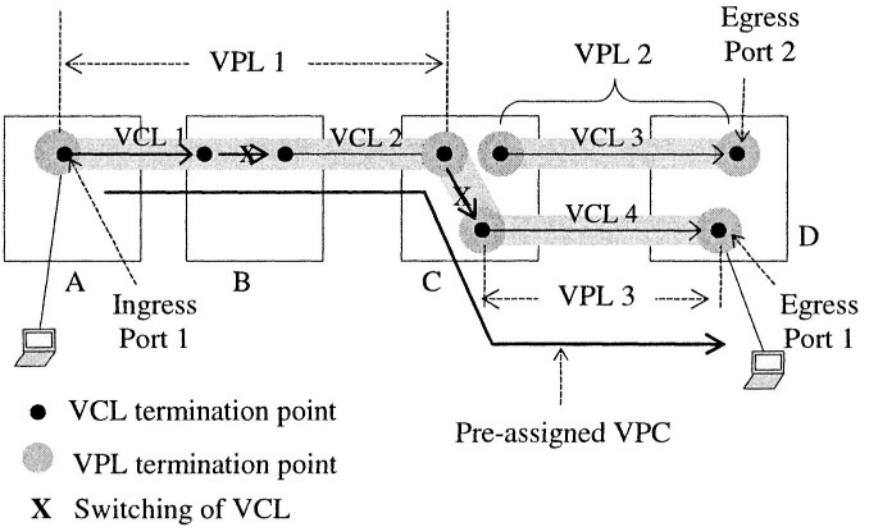


Figure 6-12. SVCC.

- Permanent Virtual Path Connection (PVPC)

3.5 Switched Virtual Connection (SVC)

A Switched Virtual Connection (SVC) is a virtual connection (VC) that is established by switching. An SVC lasts only for the duration of a call. After the call is complete, the SVC is torn down by signaling. Unlike a PVC, an SVC is established in real time initiated by an SVC request of a call. Only VCC's can be established by switching, i.e., only SVCC's exists; VPC's cannot be switched, i.e., SVPC's do not exist yet. Two types of signaling processes are used for setting up SVCC's:

- B-ISDN User Part (B-ISUP)
- Private Network-Network Interface (PNNI) signaling

An SVCC is set up within a pre-designated VPC; that is, an SVCC can be created by concatenating the VCL's within the same pre-designated VPC. Figure 6-12 illustrates setting up of an SVCC by switching. Node A initiates an SVCC from Ingress Port 1 to Egress Port 1 at Node D. This SVCC is possible only if there is a pre-assigned VPC between Ingress Port 1 and Egress Port 1.

This pre-assigned VPC is shown in the figure as the concatenation of VLP 1 from Node A to Node C and VPL 3 from Node C to Node D. The requested SVCC is created by switching VCL 1 to VCL 2 at Node B within VPL 1 and switching VCL 2 to VCL 4 in VPL 3. An SVCC from Ingress

Port 1 at Node A to Egress Port 2 at Node D is not possible because no pre-assigned VPC exists between the ingress and egress ports.

4. ATM QOS PARAMETERS

4.1 Information transfer performance

Network performance is objectively measured between two measurement points in a network without regard to the end-users' subjective opinions. As discussed in Chapter 3, the end-user QoS is determined by user's subjective opinions on satisfaction with an overall telecommunications service and includes the Customer Premises Equipment (CPE). For the ATM network, the following performance parameters are defined by the ATM Forum Technical Committee document af-tm-0056.000[25] as well as ITU-T I.356.[26]

$$Cell\ Loss\ Ratio\ (CLR) = \frac{Number\ of\ lost\ cells}{Total\ number\ of\ cells\ transmitted} \qquad (6\text{-}1)$$

$$Cell\ Error\ Ratio\ (CER)$$

$$= \frac{No.of\ errored\ cells}{No.of\ successfully\ transferred\ cells + No.of\ errored\ cells} \qquad (6\text{-}2)$$

A misinserted cell is a cell that is carried over a VC to which it does not belong. An undetected error in the ATM header causes a misinserted cell.

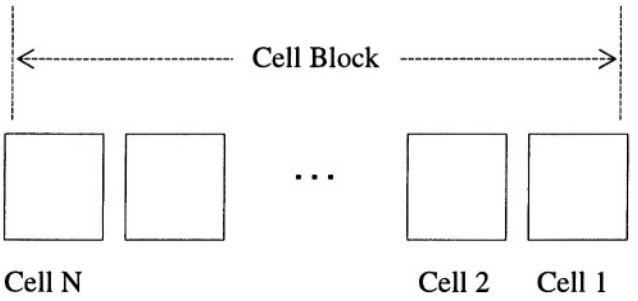


Figure 6-13. Cell block.

$$\textit{Cell Misinsertion Rate (CMR)} = \frac{\textit{Number of misinserted cells}}{\textit{Time interval}} \qquad (6\text{-}3)$$

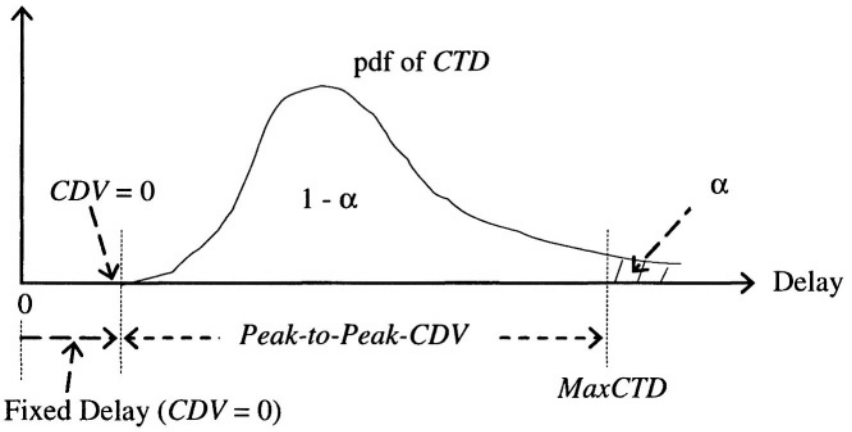
A cell block is a sequence of N consecutive ATM cells on a given connection as shown in Figure 6-13. A severely errored cell block is the cell block with M cells that are either errored, lost (i.e., never arrived) or misinserted (i.e., do not belong to the VC).

$$\textit{Severely Errored Cell Block Ratio (SECBR)}$$

$$= \frac{\textit{No.of severely errored cell blocks}}{\textit{Total no.of cell blocks transmitted}} \qquad (6\text{-}4)$$

The Cell Transfer Delay (CTD) is defined as the elapsed time between the departure time of a cell from the generating end-system and the arrival time at the destination. Figure 6-14 shows the Cell Transfer Delay (CTD) probability density model for real-time service categories. The fixed delay includes propagation delay of the physical media, delays induced by the transmission system and the fixed components of switch processing delay.

The Cell Delay Variation (CDV) is the variable component of the end-to-end delay. The CDV is introduced by buffering and cell scheduling through the network. Obviously, the CDV cannot be negative. Therefore, the



Source: Reference 25.

Figure 6-14. CTD pdf model

ordinate of the pdf of CTD is zero at delay equal to the fixed delay.

The Maximum Cell Transfer Delay (MaxCTD) specified for an ATM connection is defined as the $(1 - \alpha)$ percentile of CTD:

$$P\{CTD \succ MaxCTD\} \prec \alpha \tag{6-5}$$

The peak-to-peak CDV is the value of CDV on the CTD distribution that corresponds to the MaxCTD; that is, the peak-to-peak CDV is the MaxCTD minus the fixed delay:

$$Peak\text{-}to\text{-}Peak\ CDV = MaxCTD - Fixed\ Delay \tag{6-6}$$

ITU-T I.358[27] defines:

- Call setup delays
- Call processing failures due to all causes

4.2 End-to-end performance

The performance parameters expressed in "ratios" are estimates of "probabilities." For example, the *CLR* and the *CER* can be interpreted as the Cell Loss Probability and the Cell Error Probability.

Since the *CLR* and the *CER* are probabilities, their link values accumulate in a multiplicative fashion to yield end-to-end values. To illustrate this point, consider CER_1 and CER_2 on Link 1 and Link 2 and the end-to-end *CER*, CER_{VC} in the two link ATM Virtual Connection (VC) shown in Figure 6-15.

In the analysis of error (or failure) probabilities, a technique commonly used for deriving the end-to-end error (or failure) probability of a serial connection from the component link error (or failure) probabilities is to first find the probability of the error free (or failure free) end-to-end connection and subtract it from one (*1*) as follows:

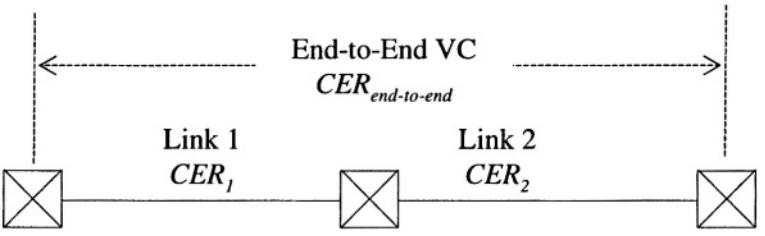


Figure 6-15 End-to-end performance.

$$CER_{end\text{-}to\text{-}end} = probability\ of\ error\ in\ end\text{ - }to\text{ - }end\ VC$$

$$= 1 - probability\ of\ error\ free\ end\text{ - }to\text{ - }end\ VC \quad (6\text{-}7)$$

$$probability\ of\ error\text{ - }free\ end\text{ - }to\text{ - }end\ VC$$

$$= probability\ of\ error\ free\ Link\ 1 \times probability\ of\ error\ free\ Link\ 2$$

$$= (1 - CER_1) \times (1 - CER_2) \quad (6\text{-}8)$$

$$CER_{end\text{-}to\text{-}end} = 1 - \{(1 - CER_1) \times (1 - CER_2)\} \quad (6\text{-}9)$$

Extending the above analysis to an end-to-end connection with N links, it follows that:

$$CER_{end\text{-}to\text{-}end} = 1 - \{(1 - CEL_1) \times (1 - CEL_2) \times \ldots \times (1 - CEL_N)\} \quad (6\text{-}10)$$

$$= 1 - \prod_{i}^{N} (1 - CER_i) \quad (6\text{-}11)$$

By similar analysis,

$$CLR_{end\text{-}to\text{-}end} = 1 - \prod_{i}^{N} (1 - CLR_i) \quad (6\text{-}12)$$

$$SECBR_{end\text{-}to\text{-}end} = 1 - \prod_{i}^{N} (1 - SECBR_i) \quad (6\text{-}13)$$

For small values of *CER* and *CLR,* the higher order terms are negligible, and the above equations reduce to approximately additive equations as follows:

$$CER_{end\text{-}to\text{-}end} \approx \sum_{i}^{N} CER_i \quad (6\text{-}14)$$

$$CLR_{end\text{-}to\text{-}end} \approx \sum_{i}^{N} CLR_i \quad (6\text{-}15)$$

$$SECBR_{end\text{-}to\text{-}end} \approx \sum_{i}^{N} SECBR_i \quad (6\text{-}16)$$

Example 1

Assuming that *CER* on Link 1 is $1x10^{-6}$ and *CER* on Link 2 is $1.5x10,^{-6}$ determine the end-to-end *CER* of a two link VC.

Solution

$$CER_{end\text{-}to\text{-}end} = 1 - \{(1 - 1\times10^{-6})\times(1 - 1.5\times10^{-6})\}$$

$$= 1 - (1 - 1\times10^{-6} - 1.5\times10^{-6} + 1.5\times10^{-12})$$

$$= 1\times10^{-6} + 1.5\times10^{-6} - 1.5\times10^{-12}$$

$$\approx 1\times10^{-6} + 1.5\times10^{-6} \approx 2.5\times10^{-6}$$

4.3 Performance Management Information Base (MIB)

The Management Information Base (MIB) defines the data collection requirements for the ATM network elements. The following are some of the examples of the ATM Network Element (NE) MIB. For each interface on the ATM NE, the MIB provides 15-minute counts of the following parameters:

- Number of cells transmitted
- Number of cells received
- Number of cells discarded due to congestion

For selected VCL's and VPL's, the MIB provides 15-minute counts of the following parameters:

- Number of cells received
- Number of cells transmitted
- Number of cells discarded by policing

For example, based on the raw counts available from the MIB, the following parameters can be derived at the ATM NE interface:

CRL

$$= \frac{\textit{15 - minute count of cell discards due to congestion at interface}}{\textit{15 - minute count of cells transmitted at interface}} \quad (6\text{-}17)$$

Utilization Factor (Incoming) at the ATM NE

$$= \frac{\textit{15 - minute count of cells received at interface}}{\textit{PCR of interface} \times \textit{900 secobds}} \quad (6\text{-}18)$$

Utilization Factor (Outgoing) at the ATM NE

$$= \frac{\textit{15 - minute count of cells transmitted at interface}}{\textit{PCR of interface} \times \textit{900 secobds}} \quad (6\text{-}19)$$

Utilization Factor (Incoming) for a selected VCL

$$= \frac{\textit{15 - minute count of cells received at VCL}}{\textit{PCR of VCL} \times \textit{900 secobds}} \quad (6\text{-}20)$$

Utilization Factor (Outgoing) for a selected VCL

$$= \frac{15 \text{ - minute count of cells transmitted at VCL}}{\text{PCR of VCL} \times 900 \text{ secobds}} \quad (6\text{-}21)$$

5. ATM SERVICE CATEGORIES

5.1 ATM service categories

The following ATM service categories are defined by the ATM Forum Technical Committee:

- Constant Bit Rate (CBR)
- real-time Variable Bit Rate (rt-VBR)
- non-real-time Variable Bit Rate (nrt-VBR)
- Unspecified Bit Rate (UBR)
- Available Bit Rate (ABR)

The Constant Bit Rate (CBR) service provides connections requiring a fixed amount of bandwidth continuously throughout the entire period of the connection. The fixed amount of required bandwidth is the Peak Cell Rate (PCR) of the connection. With the CBR service, the source can emit cells at the PCR throughout the period of the connection. Since the PCR is the maximum possible rate, the source can also emit cells below the PCR. Since CBR services operate at the maximum rate, which is the PCR, statistical multiplexing of connections is not possible. The CBR service may be used for both VPCs and VCCs.

The CBR service is intended to support real-time applications such as voice, video, and circuit emulation. The CBR service provides enough bandwidth for all connections sharing a same physical transmission path to emit cells at their individual maximum rate, the PCRs. The delay and jitter are kept at a minimum level, i.e., there should be no packet buffering in the network.

The real-time Variable Bit Rate (rt-VBR) service provides connections with the bandwidth requirement specified by three parameters: the PCR, the Sustainable Cell Rate (SCR), and the Maximum Burst Size (MBS). Like the CBR service, the rt-VBR is also intended for real-time applications, i.e., those requiring tightly constrained delay and jitter, such as voice and video applications; however, one key difference is that the rt-VBR specifies the required bandwidth taking into consideration the burstiness of the source traffic to be carried by the service.

Unlike the CBR service, therefore, the rt-VBR service is not appropriate for circuit emulation. Since rt-VBR services operate over time at a lower

rate, i.e., the SCR, than the maximum rate required, statistical multiplexing of connections is possible. Some examples of the applications appropriate for the rt-VBR service are compressed video and audio and voice over packet networks with silence suppression.

Like the rt-VBR service, the non-real-time Variable Bit Rate (rt-VBR) service also specifies the required bandwidth considering the burstiness of the source traffic and specifies the connection bandwidth requirement in terms of the same three parameters: the PCR, the SCR, and the MBS. Like the rt-VBR, the nrt-VBR service also allows statistical multiplexing of connections.

Unlike the rt-VBR service, however, the nrt-VBR is intended for non-real-time applications. Therefore, the connections for the nrt-VBR do not need to be bound by the constraints of delay and jitter. All things being equal, therefore, the nrt-VBR service would need less bandwidth in the network at the expense of more buffer space. Airline reservations and banking transactions are some of the examples of the nrt-VBR service. With the nrt-VBR service, delay and jitter are not important and packet loss is the primary performance measure.

The Unspecified Bit Rate (UBR) service is intended for non-real-time applications with no specific delay, jitter and packet loss requirements. Traditional computer communications applications such as file transfer, email and TCP-based applications are the applications appropriate for the nrt-VBR service. The UBR service does not promise any QoS guarantees in terms of delay, jitter and packet loss. The UBR service is analogous to the "Best Effort" service of the IP network discussed in Chapter 4.

Like the nrt-VBR and UBR services, the Available Bit Rate (ABR) service is also intended for non-real time applications. One key difference is that the ABR service requires feedback from the network to the end system and cooperation by the end system during the connection period. In the ABR service, after the initial connection establishment, the bandwidth made available to the source may be modified based on the fluctuating resource conditions within the ATM network. The ABR service provides feedback to the source using the flow control mechanism and the Resource Management (RM) cells.

The ABR service is ideal for applications that are designed to respond to the network feedback for flow control. For example, the TCP/IP has a built-in mechanism for responding to the network congestion, whereby, in response to a dropped packet, the source TCP host slows down the packet emission rate. The ABR service does not require bounding delay or jitter.

During the connection establishment phase, an ABR connection is specified in terms of a maximum and a minimum required bandwidth designated as the PCR and the Minimum Cell Rate (MCR), respectively.

The MCR may be specified as zero. The ABR service can change the bandwidth made available to the source during the period of the connection but it cannot reduce the bandwidth below the MCR.

5.2 Traffic descriptors

An ATM source traffic is characterized by the following traffic descriptors:

- Peak Cell Rate (PCR)
- Sustained Cell Rate (SCR)
- Maximum Burst Size (MBS)

The Peak Cell Rate (PCR) is the maximum cell rate of the source. The Sustained Cell Rate (SCR) is a long term average cell rate and, therefore, is less than the PCR. The Maximum Burst Size (MBS) specifies the maximum number of cells that can be transmitted by the source at PCR while complying with the negotiated SCR. The MBS represents the burstiness factor of the connection. The CBR traffic is characterized by the PCR. The VBR traffic is characterized by the PCR, the SCR and the MBS. For the UBR traffic, no traffic characterization is needed. Table 6-1 shows the traffic and QoS parameters specified for the ATM service categories.

5.3 AAL types

The ATM services are supported by the AAL layer as follows:

- AAL Type 1. CBR source, e.g., voice over ATM
- AAL Type 2. Multiple VBR applications involving PDUs shorter than one cell, e.g., voice over ATM with silence suppression
- AAL Types 3 and 4. Connectionless or connection-oriented

Table 6-1. Traffic descriptors and QoS parameters for the ATM service categories.

Attribute	CBR	Rt-VBR	Nrt-VBR	UBR	ABR
PCR, CDVT	specified	specified	specified	specified	specified
SCR, MBS, CDVT	n/a	specified	specified	n/a	n/a
MCR	n/a	n/a	n/a	n/a	specified
Peak-to-peak CDV	specified	specified	unspecified	unspecified	unspecified
maxCTD	specified	specified	unspecified	unspecified	unspecified
CLR	specified	specified	specified	unspecified	network specific
Feedback	unspecified	unspecified	unspecified	unspecified	specified

Source: ATM forum Technical Committee document af-tm-0056.000.[25]

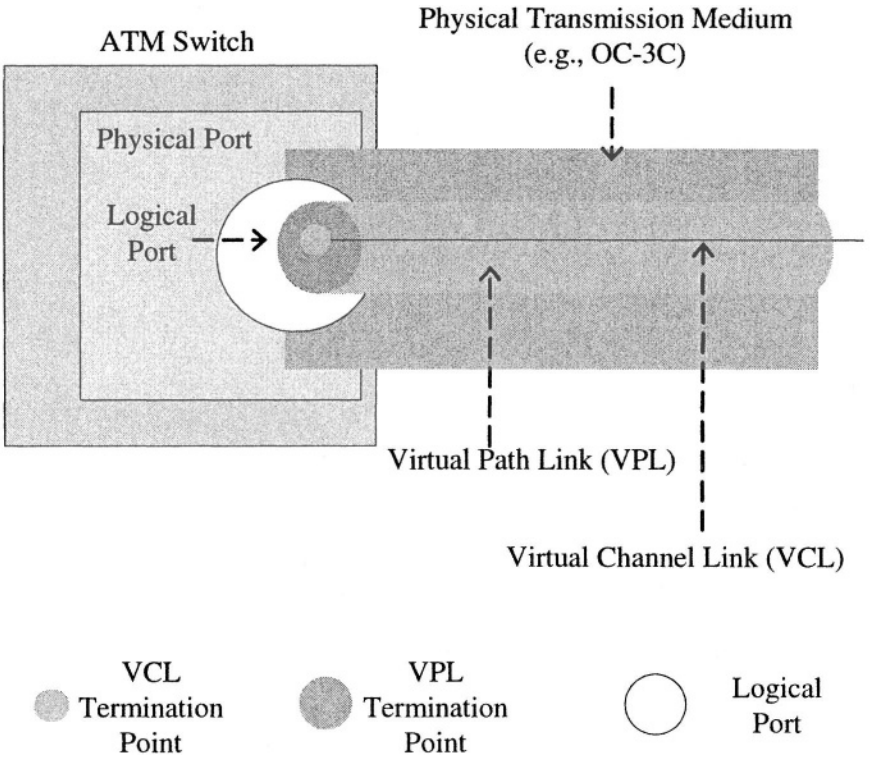


Figure 6-16. Model of an ATM switch.

- AAL Type 5. Connection-oriented higher layer protocols; VBR applications involving PDUs longer than one cell, e.g., IP/ATM, FR/ATM

6. ATM CONNECTION ADMISSION CONTROL

6.1 An ATM switch model

Figure 6-16 shows a possible generic hierarchical model of an ATM switch. Each of the input and output ports is connected to a physical transmission medium, e.g., OC-3, and DS-3. The physical port is the point at the output (or input) port where the physical transmission medium is terminated. Typically, a physical port is identified by a "shelf" and a circuit pack "slot" in the ATM switch hardware. For each physical port, logical ports are configured. A logical port terminates the VPL's and the VCL's.

6.2 Logical port bandwidth allocation

Referring to the ATM switch model in Figure 6-16, the physical port terminates a physical transmission medium with a certain amount of bandwidth, W_p. A logical port is configured to provide the intended ATM service classes, e.g., CBR, rt-VBR, nrt-VBR and UBR. The total physical port bandwidth W_p is divided among the logical ports.

The bandwidth allocated to a logical port denoted by W is divided among the ATM service categories provided at that logical port as follows:

$$w_i = W \times g_i \tag{6-22}$$

where

W = total bandwidth of a logical port
w_i = bandwidth allocated to service class i
γ_i = percentage bandwidth allocation factor for the i^{th} service

Example 2

A physical port of an ATM switch terminates an OC-3 line, which has a bandwidth of $W_p = 155$ *Mb/s*. A logical port is configured within the physical port with a bandwidth $W = 40$ *Mb/s*. The logical port is configured to provide the four ATM services with the respective bandwidths as shown in Table 6-2.

To define the ATM Connection Admission Control (CAC), consider the input and output ports shown in Figure 6-17. Suppose that the ATM source coming into the ATM switch at Input Port i requests an ATM service from Output Port j.

The CAC algorithm first determines how much bandwidth the service request from Input Port i would need at Output Port j based on the ATM service category requested and the traffic descriptors, e.g., the Peak Cell Rate (PCR), the Sustained Cell Rate (SCR) and the Maximum Burst Size (MBS), of the incoming traffic requesting the service.

If Output Port j has the required bandwidth available, the request is granted; otherwise, it is rejected. Figure 6-18 illustrates the CAC operation.

Table 6-2. Logical port configuration example.

ATM Service Categories	%Bandwidth Allocation Factor	Allocated Bandwidth (ABW) (Mb/s)
CBR	20%	8
rt-VBR	30%	12
nrt-VBR	40%	16
UBR	10%	4

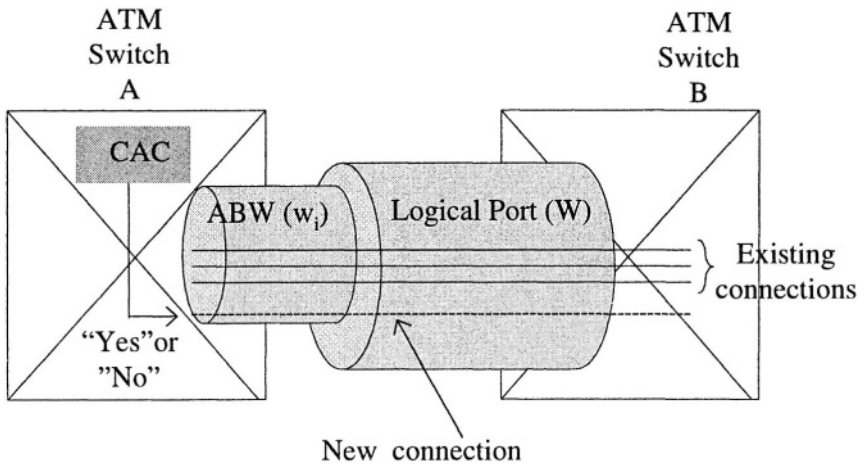


Figure 6-18. ATM CAC operation.

Example 3

Referring to Figure 6-19, *12 Mb/s* is allocated to nrt-VBR. The total bandwidth used by the existing nrt-VBR connections is *10 Mb/s.* A new nrt-VBR connection request has just arrived. The CAC algorithm determines that *4 Mb/s* is required to support the new nrt-VBR connection.

The CAC algorithm updates the total required bandwidth for nrt-VBR

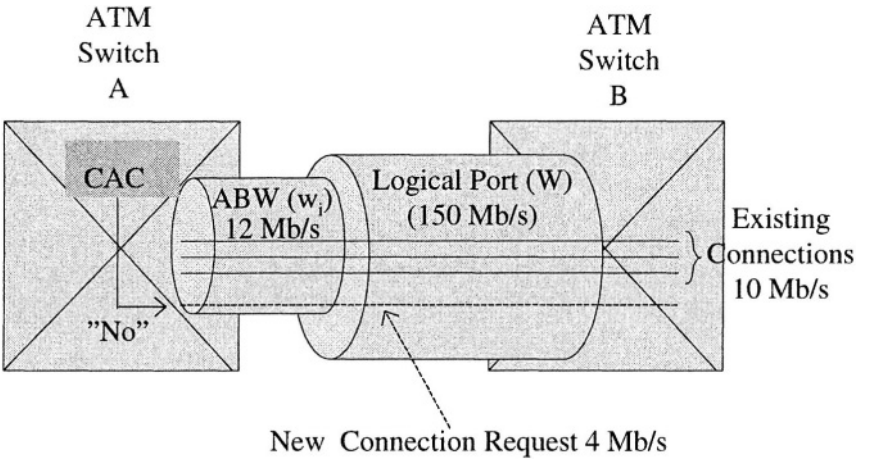


Figure 6-19. A CAC example.

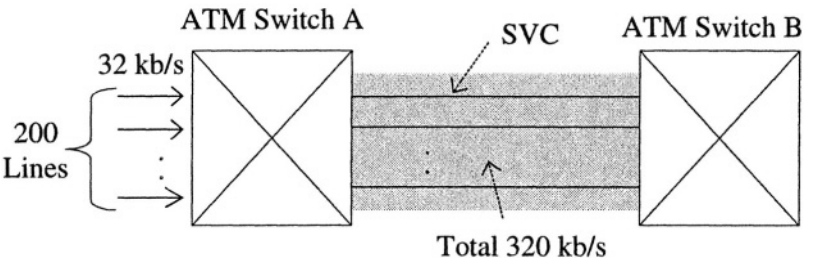


Figure 6-20. CBR CAC example.

including the new request to *14 Mb/s.* The CAC algorithm rejects the new nrt-VBR request because the updated bandwidth exceeds the total bandwidth for nrt-VBR connections, i.e., *14 Mb/s > 12 Mb/s.*

6.3 CAC for CBR traffic

The CAC algorithms for the CBR service are relatively straightforward. A CBR traffic source emits ATM cells periodically at every 1/PCR units. The CAC for CBR takes the PCR as the bandwidth required for the CBR connection. Let the total bandwidth allocated to CBR connections be w_{CBR}. The total bandwidth of the existing CBR connections is given by:

$$w_{CBR}^{N} = \sum_{i}^{N} PCR_i \qquad (6\text{-}23)$$

The $(N+1)^{th}$ request for a virtual connection with PCR_{N+1} comes to the ATM switch.

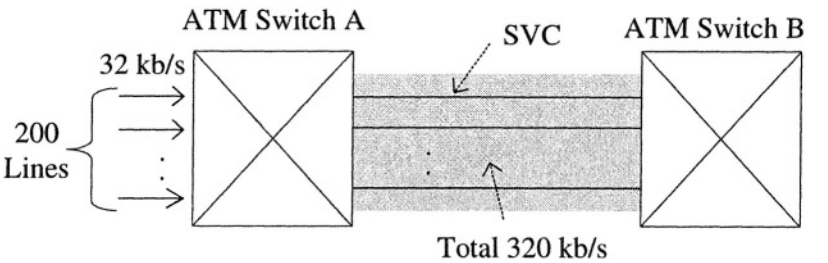


Figure 6-21. CBR CAC example.

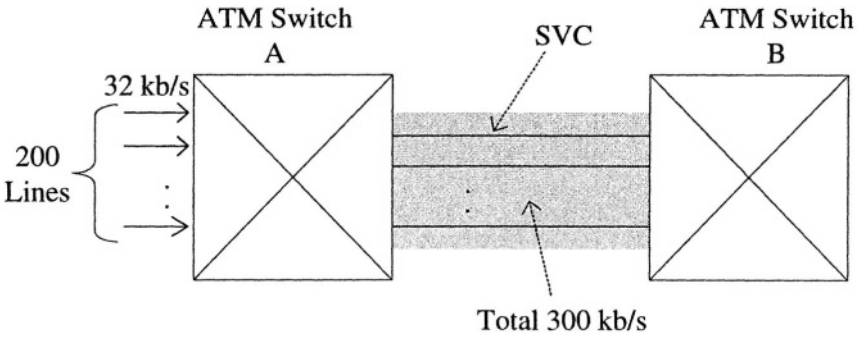


Figure 6-22. VBR CAC example.

$$w_{CBR}^{N+1} = w_{CBR}^{N} + PCR_{N+1} \tag{6-24}$$

Accept the request if

$$w_{CBR}^{N+1} \prec w_{CBR} \tag{6-25}$$

Reject the request if

$$w_{CBR}^{N+1} \geq w_{CBR} \tag{6-26}$$

Example 4 - CBR CAC

Consider *200* customer lines with PCR of *32 kb/s* each as shown in Figure 6-21. Each line generates *1.8 ccs* during the busy hour. The total trunk capacity between ATM switches A and B is *320 kb/s.* Using this amount of bandwidth, SVCs are created, each with bandwidth equal to PCR, i.e., *32 kb/s.* Determine the blocking probability during the busy hour.

Solution

First, determine the number of trunked channels as follows:

$$N = \frac{320\ kbps}{32\ kbps} = 10$$

Next, determine the total offered load at ATM switch A as follows:

$$L = 200 \times 1.8 = 360\,ccs = 10\,erlangs$$

From the Erlang B table, with linear interpolation, $P_B = 21.4\ \%$.

6.4 CAC for VBR Traffic

$$Burstiness = \frac{SCR}{PCR} \tag{6-27}$$

If the burstiness << 1, the PCR-based CAC would be very inefficient. For the VBR services, a CAC based on effective bandwidths, α_i, is used:

$$\sum \alpha_i \leq Link\ Capacity \tag{6-28}$$

Example 5 - VBR CAC

Consider *200* customer lines with PCR of *32 kb/s* and SCR of *16 kb/s* each as shown in Figure 6-22. Each line generates *1.8 ccs* during the busy hour. The total trunk capacity between ATM switches A and B is *300 kb/s*. Using this amount of bandwidth, SVCs are created, each with bandwidth equal to *30 kb/s*. Determine the blocking probability during the busy hour.

Solution

First, determine the number of trunked channels as follows:

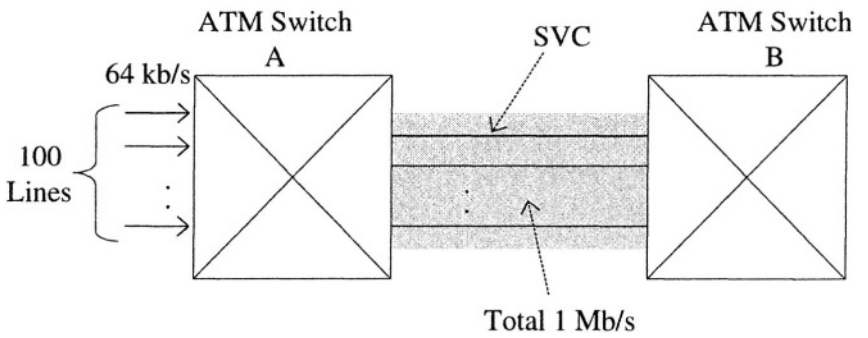


Figure 6-23. Exercise 1

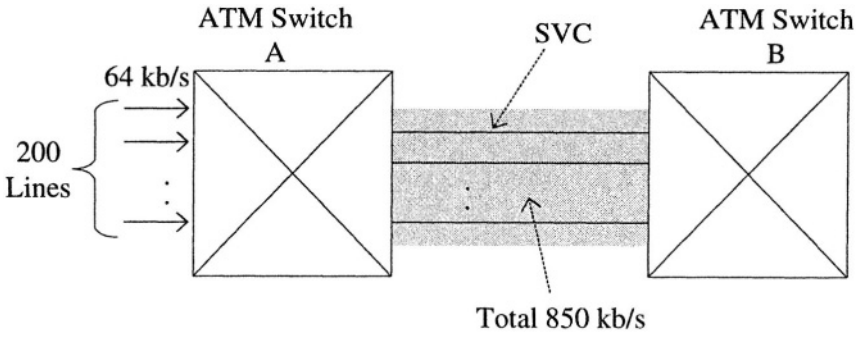


Figure 6-24. Exercise 2

$$N = \frac{300\ kbps}{30\ kbps} = 10$$

Next, determine the total offered load at ATM switch A as follows:

$$L = 200 \times 1.8 = 360\ ccs = 10\ erlangs$$

From the Erlang B table, with linear interpolation, $P_B = 21.4\ \%$.

7. EXERCISES

7.1 Problems

1. Referring to Figure 6-23, consider *100* customer lines with PCR of *64 kb/s* each. Each line generates *3.6 ccs* during the busy hour. The total trunk capacity between ATM switches A and B is *1 Mb/s*. How much more bandwidth needs to be added between the two ATM switches to meet the blocking probability of 2% during a busy hour?

2. Referring to Figure 6-24, consider *200* customer lines, each with PCR of *64 kb/s* and SCR of *40 kb/s*. Each line generates *1.8 ccs* during a busy hour. The total trunk capacity between ATM switches A and B is *850 kb/s*. Using this amount of bandwidth, SVCs are created, each with

bandwidth equal to α *kb/s*, where $SCR < \alpha < PCR$. What should α be to meet the blocking probability of 2%?

7.2 Solutions

1. First, determine the total offered load at ATM switch A as follows:

$$L = 100 \times 3.6 = 360\ ccs = 10\ erlangs$$

From the Erlang B table, determine the number of channels needed to meet $P_B = 2\%$: $N = 17$.

The total bandwidth needed is:

$$BW = 64 \times 17 = 1088\ kb/s$$

Hence, *88 kb/s* of additional bandwidth is needed.

2. First, determine the total offered load at ATM switch A as follows:

$$L = 200 \times 1.8 = 360\ ccs = 10\ erlangs$$

From the Erlang B table, the number of channels required to meet $P_B = 2\%$ is $N = 17$.

Solve the following for α:

$$\frac{850\ kbps}{\alpha} = 17$$

$$\alpha = 50\ kb/s$$

Chapter 7

MPLS

In this chapter, we will discuss the following topics:

- MPLS architecture
- MPLS implementation
- MPLS operation
- MPLS support of DiffServ

1. BACKGROUND

1.1 Why use MPLS?

The amount of traffic over the IP network is growing so explosively that it is doubling just about every few months. And yet, today's IP network is not scaling rapidly enough to meet this demand. In connection-oriented networks such as the circuit switched Time Division Multiplex (TDM) networks, where circuits are organized in trunk groups and switching can be used to select routing in real time, traffic engineering can be used to balance the traffic across the network.

In most commercial IP networks, routing is static, and all IP traffic automatically takes the "shortest" path. For this reason, the bandwidth in the IP network is not optimally distributed. While one part of the IP network is congested, other parts of the network may be lightly loaded. Unlike the circuit switched network, the IP network does not easily lend itself to traffic engineering.

The Multi-Protocol Label Switching (MPLS) is a solution to the problem faced by the IP network and, to some extent, the ATM network. MPLS

provides a mechanism for traffic engineering for packet networks such as IP and ATM networks.

In particular, MPLS can now support the IP DiffServ, and can be used in conjunction with the IP DiffServ.

The ATM network has a connection-oriented network infrastructure based on virtual connections and is ideally suited for MPLS. The word "multiprotocol" in MPLS means that MPLS is applicable to any network layer protocol, that is, it is independent of the higher layer protocols above the MPLS layer.

1.2 Conventional IP packet forwarding

To appreciate the benefits of MPLS, we first discuss how the conventional IP router forwards packets. Figure 7-1 shows the conventional IP routing. At each router in the IP network, a packet is forwarded to the next router based on a routing table. A routing table is programmed into the router and remains fixed until it is changed manually by the network operator.

At each router, there are a finite number of possible output ports to which a packet can be routed. Incoming packets are first mapped to a small number of groups of packets referred to as Forwarding Equivalence Classes (FEC's). All the packets in an FEC are routed to the same output port. From the point of view of routing, therefore, all packets grouped into the same

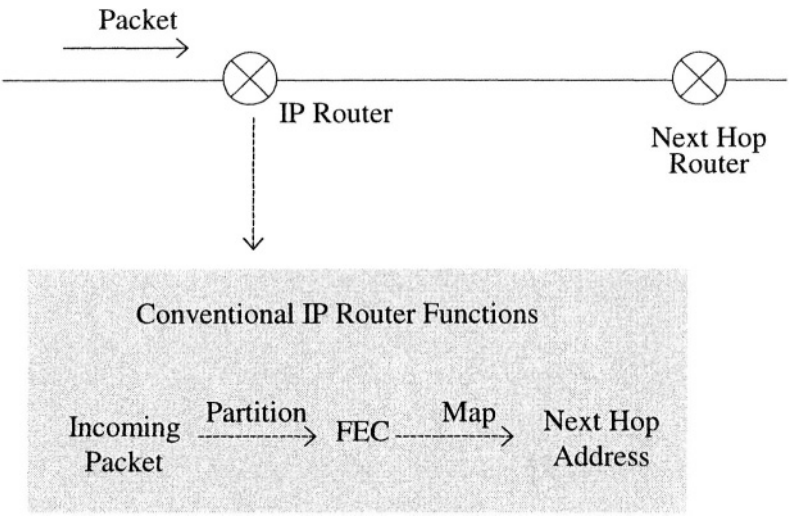


Figure 7-1. Conventional IP routing functions.

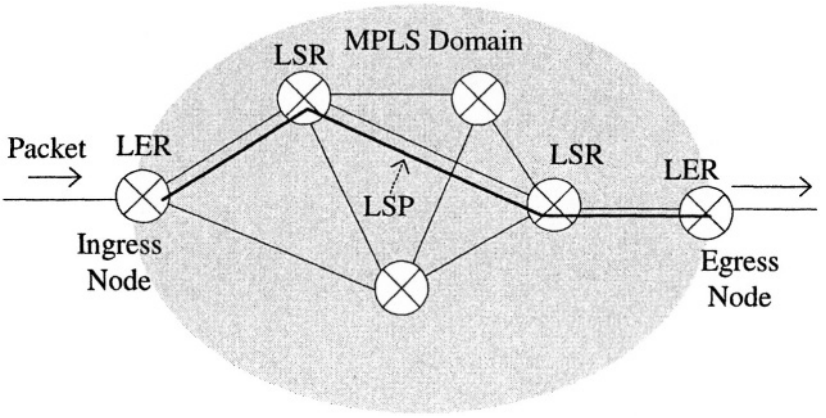


Figure 7-2. MPLS architecture.

FEC are indistinguishable and are forwarded the same way.

As shown in Figure 7-1, when a packet arrives at a router, the router first examines the packet header and, based on the information contained in the header, maps the packet to an FEC. Once the packet is mapped to an FEC, the routing table shows which output port the packet of that FEC should be forwarded to.

The conventional IP routers support only a very limited number of FEC's, e.g., the destination IP address. Multi-dimensional FEC classification is difficult because of network packet processing and scalability issues. At each router, this process is repeated: the packet header is reexamined, and the mapping to an FEC and mapping to the output port are repeated. The process of examining the packet header is resource-intensive and time-consuming.

1.3 MPLS advantages

MPLS does not avoid packet forwarding. What then is the advantage of using MPLS? By using labels rather than the information contained in the IP header, processing involved in packet forwarding is made simple and fast. This benefit is very similar to the benefit of the "zip code" system used by the U.S Postal service. Using the zip codes instead of the actual addresses makes letter delivery fast.

There are other benefits of MPLS. MPLS provides a means of packet network traffic engineering. By using labels on top of the native packet headers, MPLS can direct the packet traffic to the pre-engineered paths, which may be different from the paths the native packets would

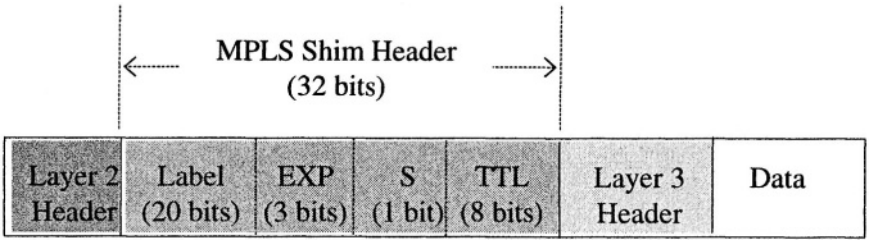


Figure 7-3. MPLS Shim header.

automatically take. These traffic paths can be easily re-configured simply by using different labels without changing the contents of the packet headers.

Since, in MPLS, packet forwarding is based on processing simple labels and not based on processing native packet headers, MPLS forwarding can be implemented at devices that do not have the capability of forwarding native packets. In fact, the word "Multi-Protocol" in MPLS means that the MPLS capability is independent of the native packet protocols.

In MPLS, labels can be defined for packets based on the factors other than the native packet header contents, e.g., the incoming port identification, the ingress router identification, and the ingress router identification. Grouping of traffic classes for labeling is flexible and can be done without depending on the native packet routing structures.

1.4 MPLS architecture

The MPLS architecture is defined in RFC 3031.[28] Referring to Figure 7-2, an MPLS node is a node which runs MPLS and is capable of forwarding packets based on labels. An MPLS domain is a contiguous set of MPLS nodes which are also in one routing or administrative domain, e.g., an Internet Service Provider (ISP).

MPLS ingress and egress nodes are MPLS edge nodes in the role of handling traffic as it enters an MPLS domain and as it leaves an MPLS domain. A Label Switching Router (LSR) is an MPLS node that is capable of forwarding native Layer 3 packets. A Label Edge Router (LER) is an LSR at the ingress or egress node.

A Label Switched Hop is a hop between two MPLS nodes, on which forwarding is done using labels. A Label Switched Path (LSP) is a routing path through one or more LSR's at one level of hierarchy followed by packets in a particular FEC.

In MPLS, a packet is assigned a label. The packets belonging to a same FEC are assigned a same label. A label has only local significance, and may

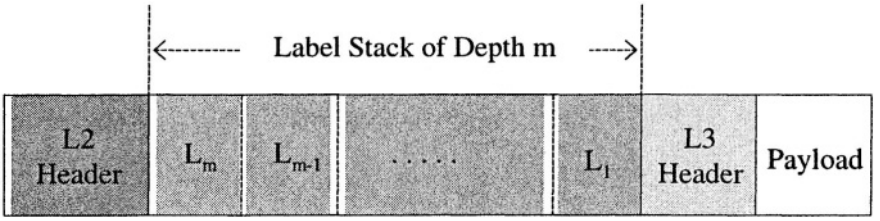


Figure 7-4. Label stack of depth m.

be mapped to other labels as a packet goes through the network. An unlabeled packet is a packet that has not yet been assigned a label; a labeled packet, a packet to which a label has been assigned.

2. LABEL ENCODING

There are two types of MPLS label encoding techniques:

- Use of a new label for MPLS, e.g., the Shim header.
- Use of the information that is in an existing data link or network layer header, e.g., the VPI and VCI in the ATM header.

2.1 MPLS shim header

The MPLS shim header is defined in RFC 3032.[29] Figure 7-3 shows the MPLS shim header. The shim header is inserted between the Layer 3 header and the Layer 2 header. It is 32 bits long. Twenty bits are allocated for defining labels. The three-bit EXP field is reserved for "experimental" purposes. The EXP field will be discussed later for MPLS's support of DiffServ referred to as the E-LSP. The one-bit long Stack (S) field is used for creating a label stack, and indicates the presence of a label stack. The eight bit long Time to Live (TTL) field is similar to the TTL field of other protocols, e.g., IP header, and is decremented at each LSR hop.

Labels can be organized in a hierarchical manner referred to as a "stack." Figure 7-4 shows a total of *m* MPLS shim headers placed one after another above the Layer 3 header as a stack. Each shim header defines a separate label. The label at the bottom of the stack or the label closest to the Layer 3 header is the Level 1 label, and the top most label of the stack is the Level m label. A label stack of depth zero refers to an empty stack and is associated with an unlabeled packet. A label stack is used to create a hierarchy of MPLS tunnels, i.e., a tunnel inside a tunnel and so on.

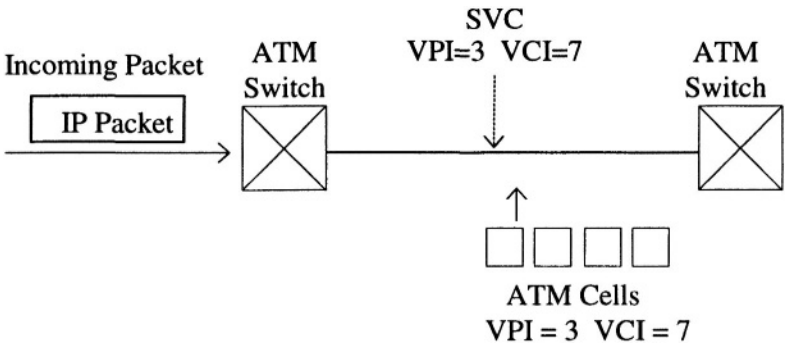


Figure 7-5. MPLS LSP using the ATM SVC.

2.2 Label encoding over ATM

2.2.1 ATM SVC encoding

Since ATM is a connection-oriented packet network, it provides an infrastructure that is easily adaptable to MPLS implementation. The VPI and VCI fields of ATM provide a convenient means for MPLS labeling. Figure 7-5 shows an LSP created by using an SVC. The MPLS label in this

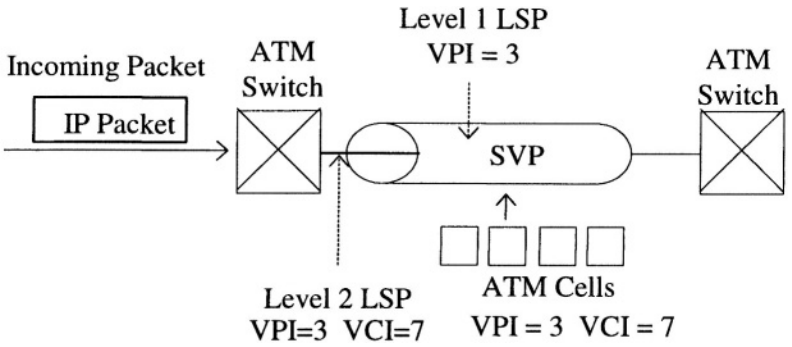


Figure 7-6. MPLS LSP by ATM SVP

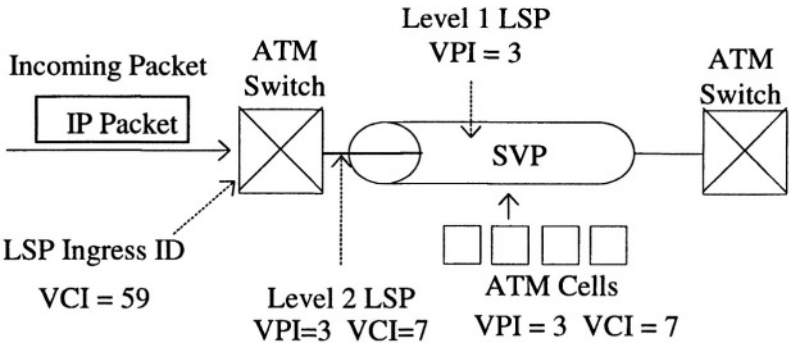


Figure 7-7. MPLS LSP by ATM SVP multipoint coding

case is simply the VPI/VCI combination.

2.2.2 ATM SVP encoding

Figure 7-6 shows LSP's with a label stack of depth two created by using the VPI and VCI fields. In this example, the VPI field is used as the Level 2 label and the VCI, the Level 1 label. The VPI defines an LSP tunnel and the VCI defines a tunnel inside the VPI tunnel.

2.2.3 ATM SVP multipoint encoding

Figure 7-7 shows a mixture of the two examples discussed above. In this case, the VPI field is used as the Level 2 label as in the example of Figure 7-6. In the current example, however, only a part of the VCI field is used as the Level 1 label. The remainder of the VCI field is used to identify the LSP

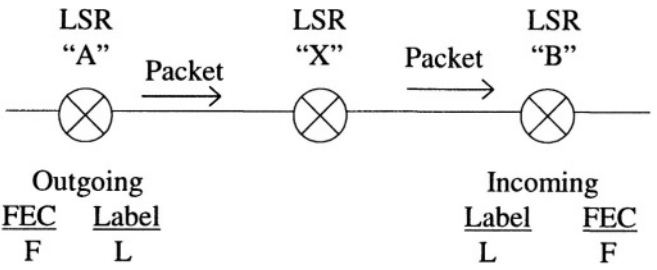


Figure 7-8. Label binding.

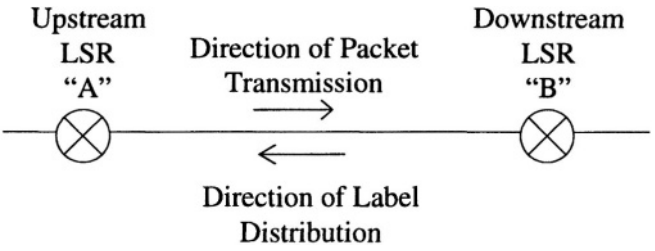


Figure 7-9. Label distribution.

ingress. In this example, ATM cells from different packets can carry different VCI values.

3. MPLS IMPLEMENTATION

To implement MPLS, the traffic classes or FEC's must first be defined. The labels to be used for the FEC's must be defined and assigned to the FEC's. The labels have only local significance and are used between adjacent LSR's. As shown in Figure 7-8, the two LSR's, LSR A and LSR B, must agree on what labels are used for the FEC's. This process is referred to as label binding.

The labels must be distributed to the MPLS nodes using a label distribution protocol. As shown in Figure 7-9, the direction of label distribution is from the downstream LSR to the upstream LSR and is in the opposite direction of packet transmission.

There are two classes of label distribution protocols. There are existing

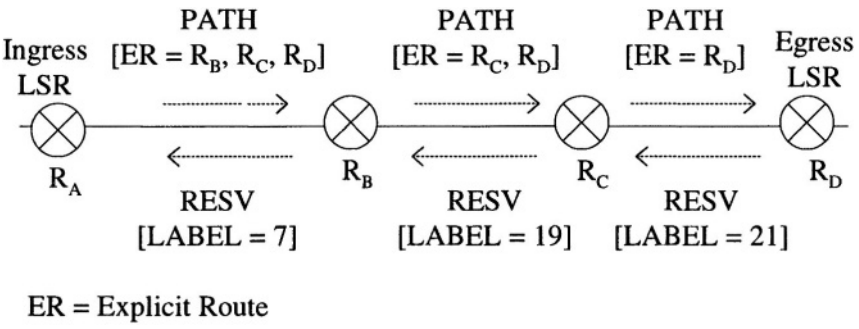


Figure 7-10. Label distribution by RSVP-TE.

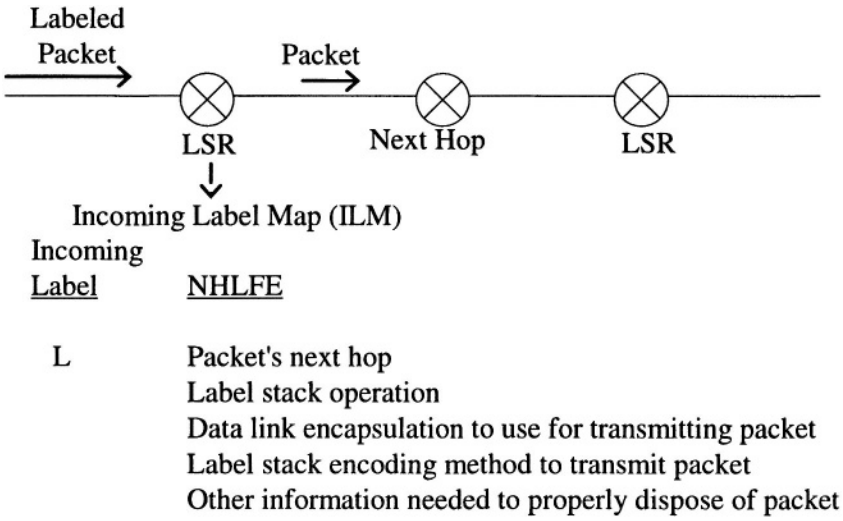


Figure 7-11. Incoming Label Map (ILM).

protocols extended to piggyback label distribution, e.g., the BGP extension and the RSVP with Tunneling Extensions (RSVP-TE) specified by RFC 3209.[30] There are also new protocols specifically designed for the MPLS label distribution, e.g., the LDP and the CR-LDP.

The RSVP with Tunneling Extensions (RSVP-TE) extends RSVP to provide the capability to establish MPLS LSP's, i.e., tunnels. With the original RSVP, a path is reserved for a particular destination and transport-layer protocol. However, the original RSVP does not change the routing and packet forwarding and packets follow the same IP packet forwarding determined by the routing tables in the IP routers. The original RSVP is a resource reservation protocol and is not a routing protocol.

With the RSVP-TE, an LSP can be established between an ingress LSR and an egress LSR. The ingress LSR can determine which packets can be sent over the LSP. The RSVP-TE extends the original RSVP by adding the following five new objects to the original object list:

- LABEL REQUEST
- LABEL
- EXPLICIT ROUTE
- RECORD ROUTE
- SESSION ATTRIBUTE

The LABEL REQUEST, EXPLICIT ROUTE and SESSION ATTRIBUTE objects are used only in the PATH message. The LABEL object is used only in the RESV message. Finally, the RECORD ROUTE

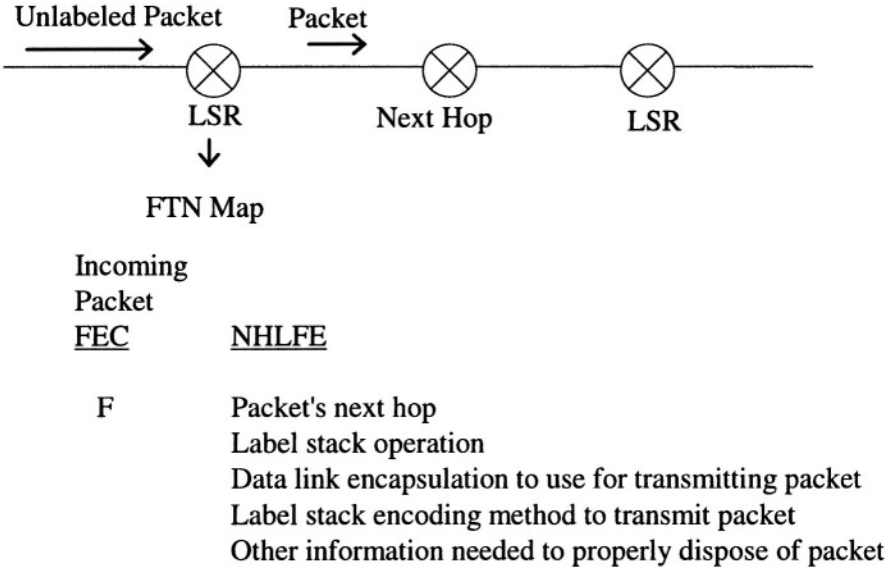


Figure 7-12. The FTN map.

object is used in both the PATH and the RESV messages. Figure 7-10 shows some of the above new objects used in the PATH and the RESV messages to set up an LSP.

4. MPLS OPERATION

4.1 Label mapping

4.1.1 Incoming Label Map (ILM)

The Incoming Label Map is a label switching table similar to the IP routing table in the IP routers. It is used by MPLS for forwarding labeled packets. Figure 7-11 shows the ILM. The ILM maps the label of an incoming packet to the Next Hop Label Forwarding Entry (NHLFE). Among the entries in the NHLFE are the next hop and a label stack operation to be performed.

4.1.2 FEC-to-NHLFE (FTN) map

The FEC to NHLFE (FTN) map is similar to the ILM. The main difference is that the FTN map is used for forwarding unlabeled packets that

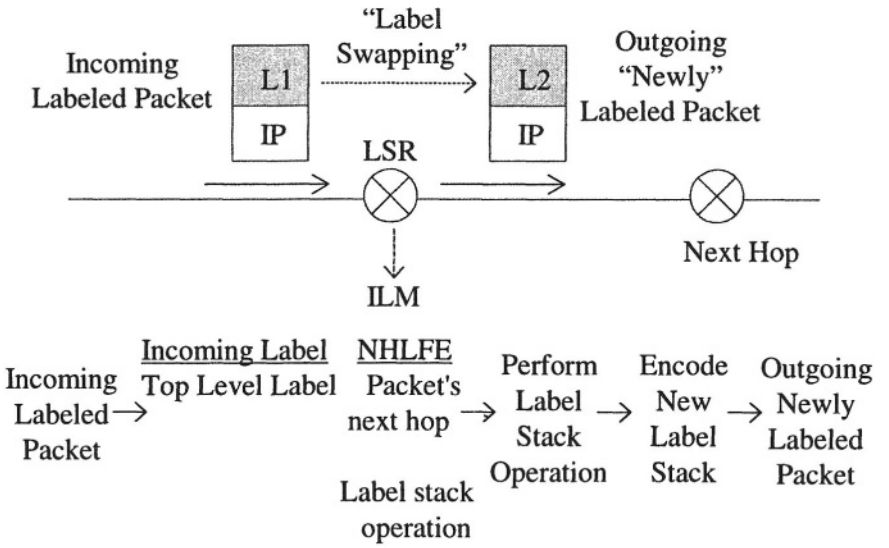


Figure 7-13. Label swapping.

need to be labeled before being forwarded. Figure 7-12 shows the FTN map.

4.1.3 Label swapping

Figure 7-13 shows the MPLS label swapping process for a labeled packet using the ILM. An incoming labeled packet is processed by mapping its label to the corresponding FEC and then to the NHLFE. The NHLFE entry indicates the next hop for the packet. It also shows the label stack operation to perform. For example, the incoming label (Label 1) is "popped" and a new label (Label 2) is "pushed."

Based on the FEC and the next hop information obtained from the ILM, and based on the label binding between the current LSR and the next hop LSR, a new label is "pushed" to the packet. The newly labeled packet is then forwarded to the next hop LSR. Figure 7-14 shows forwarding of unlabeled packets.

Label swapping is performed for packets that arrive already labeled. To forward an unlabeled packet, the difference is in determining the FEC. When an unlabeled packet arrives at an LSR, the LSR must first determine its FEC not from the label because there is no label but from the Layer 3 header of the packet. Once the FEC for the unlabeled packet is determined, the rest of the process is similar to the label swapping process discussed earlier except that the FTN map is used instead of the ILM map. In this

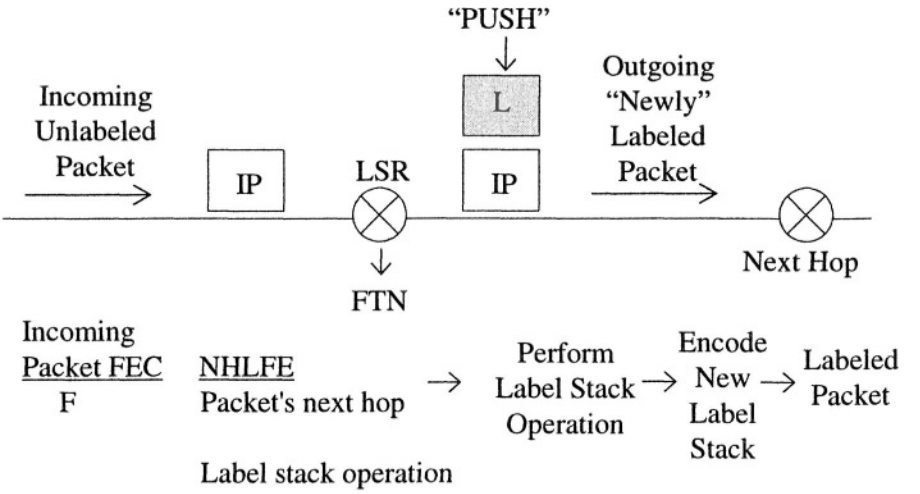


Figure 7-14. Label pushing.

case, there is no label to pop. A new label is pushed on the packet and the labeled packet is forwarded to the next hop LSR.

4.2 An example of hierarchical MPLS tunnels

RFC 3031[28] gives an example of a two level tunnel. Figure 7-15 illustrates the example pictorially in a step-by-step manner. In the figure, key events are numbered. An unlabeled IP packet "P" arrives at R_1. This packet is to take a level 1 LSP, LSP 1, from R_1 to R_4. Step 1 in the figure is to push a level 1 label, L_1-1, onto the packet P.

The labeled packet with L_1-1 is forwarded to R_2. At R_2, Step 2 is to perform label swapping on L_1-1 and push a new level 1 label, L_1-2, on P. R2 also recognizes that the packet P must take a level 2 LSP tunnel, LSP 2, from R_2 to R_3. Step 3 is to push a level 2 label, L_2-2, on top of L_1-2. This packet with label of depth 2 is then forwarded to R_{21}.

From R_{21} to R_{23}, level 2 label swapping is performed. Since R_{23} is the penultimate LSR of LSP 2, Step 4 at R_{23} is to pop the level 2 label, L_2-22, and forward the packet with level 1 label, L_1-2, to R_3. At R_3, which is the penultimate LSR of LSP 1, Step 5 is to pop the level 1 label, L_1-1, and forward the unlabeled packet P to R_4. At Step 6, the unlabeled native Layer 3 packet P arrives at the terminating point of LSP 1.

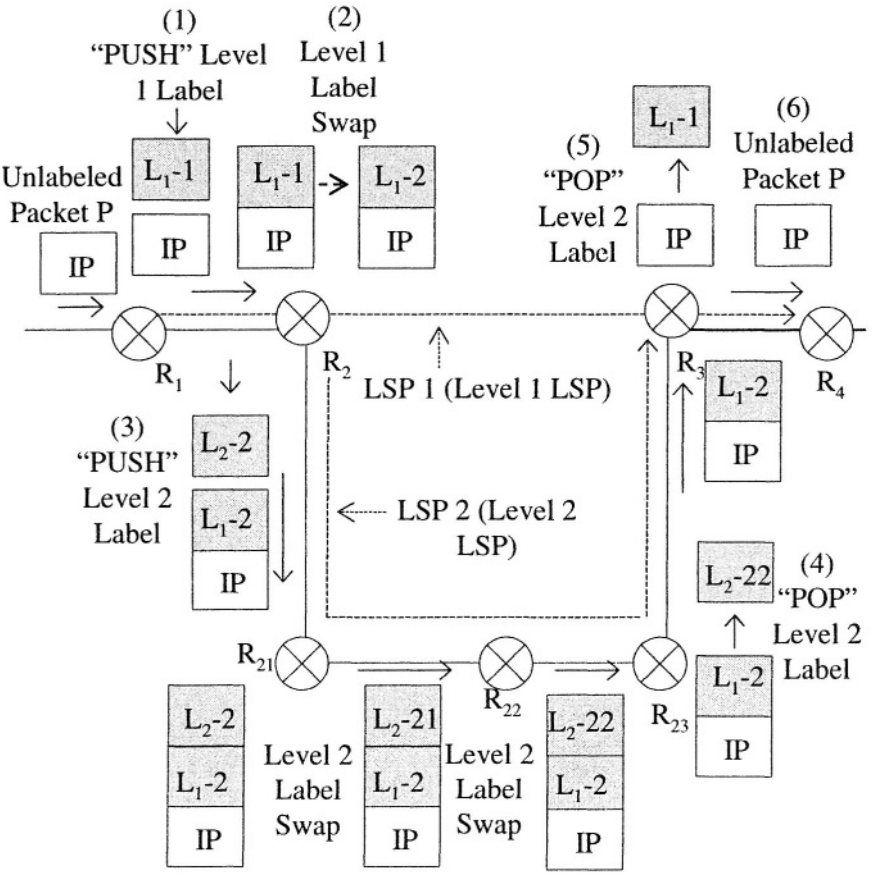


Figure 7-15. An example of hierarchical LSP.

5. LABEL MERGING

5.1 General description

Figure 7-16 illustrates the concept of label merging. There are two original LSP's shown in the figure: LSP 1 from $\mathbf{R_A}$ to $\mathbf{R_D}$ via $\mathbf{R_C}$ and LSP 2 from $\mathbf{R_B}$ to $\mathbf{R_D}$ via $\mathbf{R_C}$. Both LSP 1 and LSP 2 have the same destination $\mathbf{R_D}$ and share a common intermediate point $\mathbf{R_C}$. LSP 1 uses **Label-1**$_{\mathbf{AC}}$ from $\mathbf{R_A}$ to $\mathbf{R_C}$ and **Label-1**$_{\mathbf{CD}}$ from $\mathbf{R_C}$ to $\mathbf{R_D}$. LSP 2 uses **Label-2**$_{\mathbf{BC}}$ from $\mathbf{R_B}$ to $\mathbf{R_C}$ and **Label-2**$_{\mathbf{CD}}$ from $\mathbf{R_C}$ to $\mathbf{R_D}$.

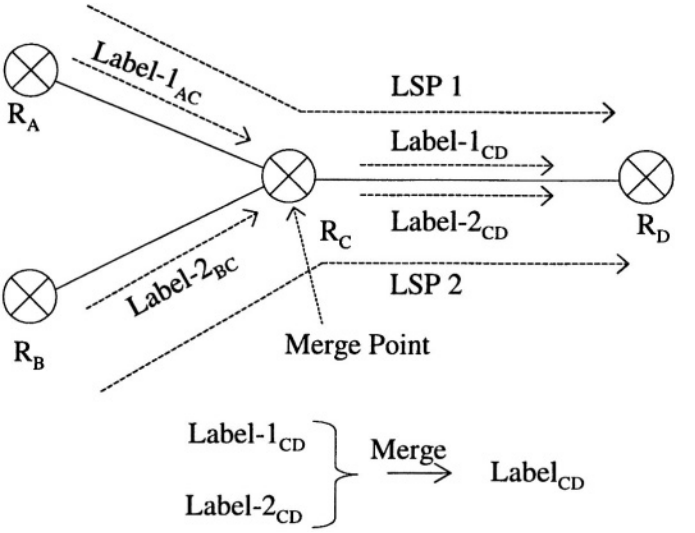


Figure 7-16. LSP merging.

Since the portion of the LSP's from R_C to R_D is common to both LSP's, the two LSP's can use a same label, say $Label_{CD}$, instead of the two separate original labels. This process of combining labels is referred to as label merging. The common intermediate point, R_C, is referred to as the merge point.

5.2 Label merging over ATM

ATM provides a natural means of label merging. There are two types of label merging over ATM: VP merging and VC merging.

5.2.1 VP merging

VP merging is to merge the LSP's based on Virtual Paths (VP's). Figure 7-17 shows VP merging. Before the merging, there are two LSP's based on two separate VP's, one labeled with VPI = 2 and one labeled with VPI = 4. These two LSP's are merged into a single LSP using the label VPI = 9. VCI's can be used to distinguish the sources sharing this same VPI.

5.2.2 VC merging

VC merging is to merge the LSP's based on Virtual Channels (VC's). Figure 7-18 shows VC merging. Before the merging, there are two LSP's

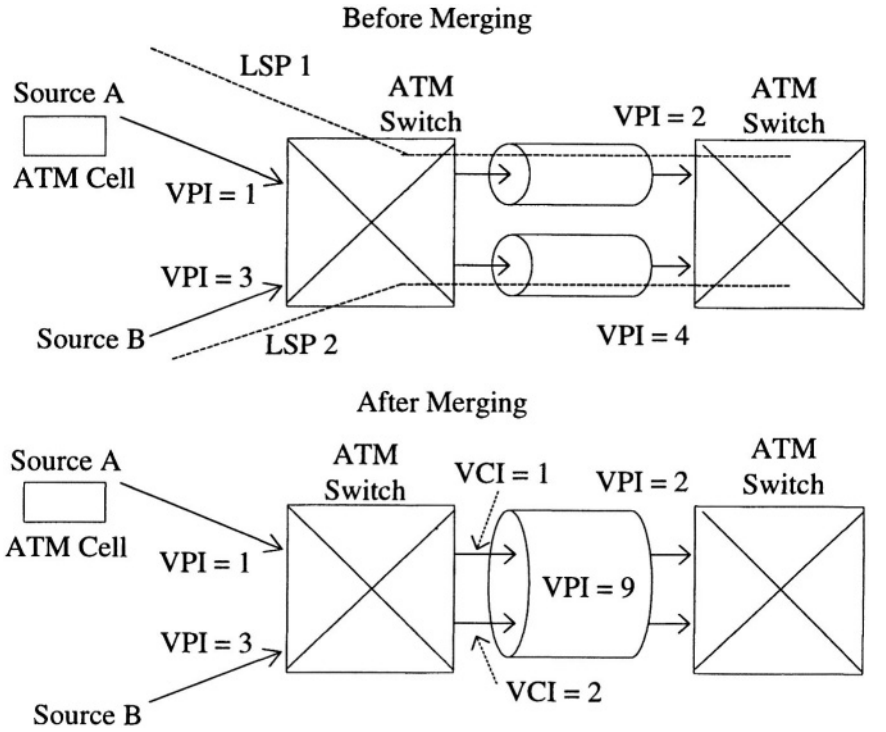


Figure 7-17. VP merging.

based on two separate VC's, one labeled with VPI/VCI = 2/2 and one labeled with VPI/VCI = 4/4. These two LSP's are merged into a single LSP using the label VPI/VCI = 9/7.

Unlike VP merging, VC merging uses the VCI to identify the VC's involved and the VCI is no longer available to distinguish the sources. Therefore, the ATM switches must buffer the ATM cells from the same Layer 3 packet until the entire packet is received before forwarding it on to the VC.

6. MPLS SUPPORT OF DIFFERENTIATED SERVICES

The basic idea of MPLS is to use a label and avoid processing the IP header. Hence, MPLS is unaware of what is carried in the payload, which is IP packets including the IP header. How can MPLS be used to support DiffServ? The basic difficulty is that DiffServ uses DSCP and DSCP is in

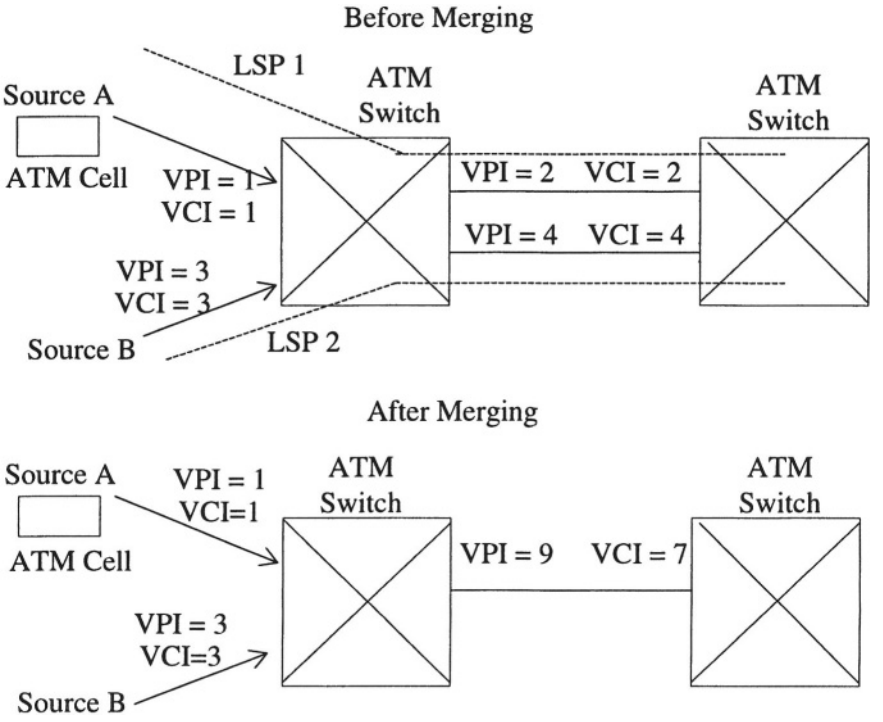


Figure 7-18. VC merging.

the IP header. How can DSCP be made visible to the MPLS layer? RFC 3270[31] provides a solution for supporting DiffServ over MPLS networks.

Table 6-1. DiffServ PHB to EXP mapping.

PHB class	PHB subclass	DSCP	EXP
EF	AF41	101110	111
AF4	AF41	100010	110
	AF42	100100	
	AF43	100110	
AF3	AF31	011010	101
	AF32	011100	
	AF33	011110	
AF2	AF21	010010	100
	AF22	010100	
	AF23	010110	
AF1	AF11	001010	011
	AF12	001100	
	AF13	001110	
BE		000000	010

Source: IETF RFC 2597.[23]

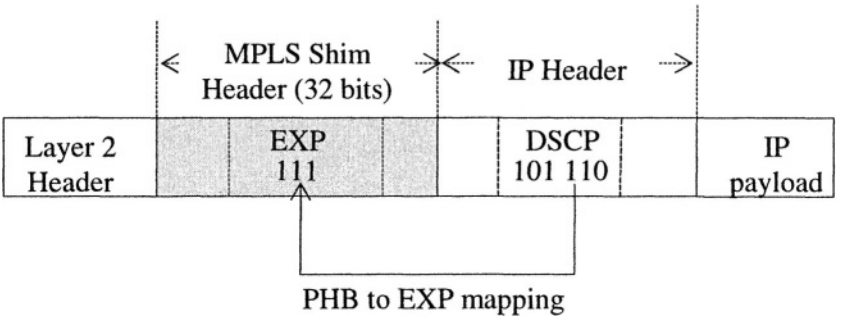


Figure 7-19. Mapping of DiffServ PHB to MPLS EXP bits.

There are basically two ways of handling this problem. One is to use a field in the MPLS shim header to map the DiffServ PHB's corresponding to the DSCP's in the IP header; the other is to create separate LSP's per PHB's represented by the DSCP's. The first type of LSP is referred to as the E-LSP, and the latter type, L-LSP.

6.1 E-LSP

The MPLS shim header contains three bits reserved for experimental use. This three-bit field is referred to as the EXP field. This EXP field can be used to support the DiffServ PHBs by MPLS. An LSP capable of supporting the DiffServ PHB's created by using the EXP field is referred to as the E-LSP.

The E-LSP stands for the EXP inferred PHB scheduling class LSPs. In the E-LSP, up to eight DiffServ PHB's can be distinguished on a single physical LSP using the 2^3 permutations of the three bits of the EXP field. Figure 6-19 shows the mapping of the DiffServ PHB's to the MPLS Shim EXP field. Note that multiple DSCP's can be grouped into a single PHB.

Table 6-1 shows an example of mapping the EF, four AF classes and the best effort class to EXP. Figure 7-20 shows an E-LSP with seven different PHB's. At each LSR, the seven PHB's can be scheduled by seven separate queues.

6.2 L-LSP

Another method of supporting the DiffServ PHB's is the L-LSP. The L-LSP stands for the Label only inferred PHB scheduling class LSPs. In the L-LSP method, multiple LSPs are established between an ingress LSR and an egress LSR. Each LSP carries traffic belonging to a single class requiring

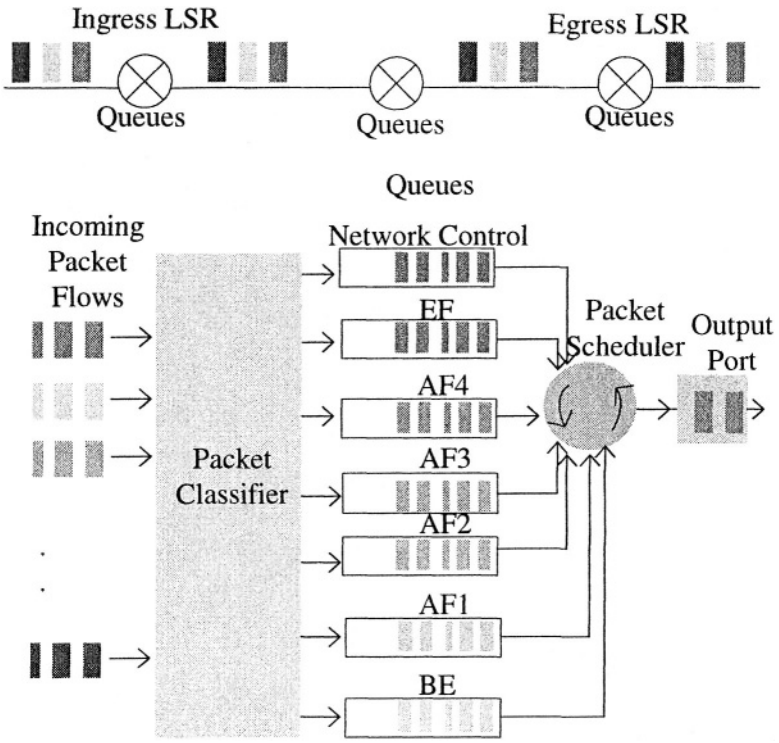


Figure 7-20. E-LSP.

a particular PHB. The LSP's are pre-configured in such a way that the label indicates a particular PHB class.

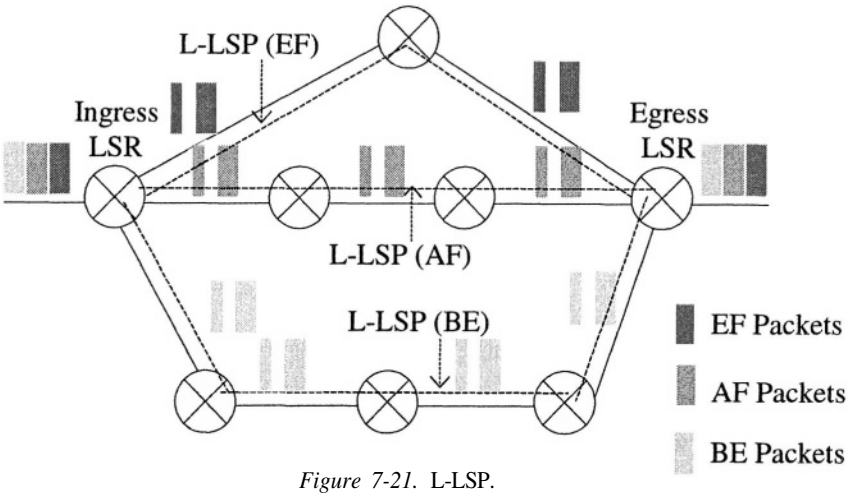


Figure 7-21. L-LSP.

Figure 6-21 illustrates the L-LSP.

Chapter 8

REFERENCES

1. Leonard Kleinrock: Queuing Systems.
2. Athanasios Papoulis: Probability, Random Variables, and Stochastic Processes.
3. Richard von Mises: Probability, Statistics, and Truth.
4. Pulse Code Modulation (PCM) of Voice Frequencies, ITU-T Recommendation G.711, November 1988.
5. Reduced complexity 8 kbit/s CS-ACELP speech codec, ITU-T Recommendation G.729 Annex A, November 1996.
6. Richard V. Cox, "Three New Speech Coders from the ITU Cover a Range of Applications," IEEE Communications Magazine, September 1997.
7. 40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM), ITU-T Recommendation G.726, 1990.
8. 7 kHz audio coding within 64 kbit/s Using Sub-Band Adaptive Differential Pulse Code Modulation (SB-ADPCM), ITU-T Recommendation G.722, 1988.
9. One-way transmission time, ITU-T Recommendation G.114, May 2003.
10. The Emodel, a computational model for use in transmission planning, ITU-T Recommendation G.107, December 1998.
11. Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP), ITU-T Recommendation G.729, March 1996.
12. Bur Goode, "Voice Over Internet Protocol (VoIP)," Proceedings of the IEEE, Vol. 90, No. 9, September 2002.
13. J. Heinanen and R. Guerin, "A Single Rate Three Color Marker," RFC 2697, September 1999.
14. Heinanen and R. Guerin, "A Two Rate Three Color Marker," RFC 2698, September 1999.
15. K. Ramakrishnan, "A Proposal to add Explicit Congestion Notification (ECN) to IP", RFC 2481, January 1999.
16. Information Sciences Institute, University of Southern California, "Transmission Control Protocol DARPA Internet Program Protocol Specification," RFC 793, September 1981.
17. R. Braden, Ed., L. Zhang, S. Berson, S. Herzog and S. Jamin, "Resource ReSerVation Protocol (RSVP) Version 1 Functional Specification," RFC 2205, September 1997.
18. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, „An Architecture for Differentiated Services," RFC 2475, December 1998.

19. Information Sciences Institute, University of Southern California, "Internet Protocol Darpa Internet Program Protocol Specification," RFC 791, September 1981
20. K. Nichols, S. Blake, F. Baker and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, " RFC 2474, December 1998.
21. V. Jacobson, K. Nichols and K. Poduri, "An Expedited Forwarding PHB," RFRC 2598, June 1999.
22. B. Davie, A. Charny, J.C.R. Bennett, K. Benson, J.Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu and D. Stiliadis, "An Expedited Forwarding PHB (Per-Hop Behavior)," RFC 3246, March 2002.
23. J. Heinanen, W. Weiss and J. Wroclawski, "Assured Forwarding PHB Group," RFC 2597, June 1999.
24. Vocabulary of terms for broadband aspects of ISDN, ITU-T Recommendation I.113, June 1997.
25. The ATM Forum Technical Committee, "Traffic Management Specification Version 4.0," af-tm-0056.000, April 1996.
26. B-ISDN ATM layer cell transfer performance, ITU-T Recommendation I.356, March 2000.
27. Call processing performance for switched Virtual Channel Connections (VCCs) in a B-ISDN, ITU-T Recommendation I.358, June 1998.
28. E. Rosen, A. Viswanathan and R. Callon, "Multiprotocol Label Switching Architecture," RFC 3031, January 2001.
29. E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li and A. Conta, "MPLS Label Stack Encoding," RFC 3032, January 2001.
30. D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels," RFC 3209, December 2001.
31. F. Le Faucheur, Editor, L. Wu, B. Davie, S. Davari, P. Vaananen, R. Krishnan, P. Cheval and J. Heinanen, "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services," RFC 3270, May 2002.

Acronyms

ABR	Available Bit Rate
A/D	Analog/Digital
ADPCM	Adaptive Differential Pulse Code Modulation
AQM	Active Queue management
AAL	ATM Adaptation Layer
ABW	Allocated Bandwidth
ACELP	Algebraic Code –Excited Linear Predictive
AAL1	ATM Adaptation Layer 1
AF	Assured Forwarding
ATM	Asynchronous Transfer Mode
BISDN	Broadband Integrated Services Digital Network
BA	Behavior Aggregate
BRI	Basic Rate Interface
B-ICI	Broadband ISDN Inter Carrier Interface
B-ISUP	B-ISDN User Part
CDF	Cumulative Distribution Function
CAC	Connection Admission Control
CIR	Committed Information Rate
ccs	Hundred Call Second
CS-ACELP	Conjugate-Structure Algebraic Code Excited Linear Predictive
CELP	Code –Excited Linear Predictive
CBR	Continuous Bit Rate
CMR	Cell Mis-insertion Rate
CBS	Committed Burst Size
CCITT	
CE	Congestion Experienced
CWR	Congestion Window Reduced
CBQ	Class-Based Queuing
CB WFQ	Class-Based Weighted Fair Queuing

CDV	Cell Delay Variation
CSCP	Class Selector Code Points
CS	Convergence Sublayer
CTD	Cell Transfer Delay
CR-LDP	Constraint Routing-Label Distribution Protocol
CPE	Customer Premises Equipment
D/A	Digital/Analog
DiffServ	Differentiated Services
DS-1	Digital Signal-1
DSCP	DiffServ Code Point
DS	DiffServ
EBS	Excess Burst Size
ECN	Explicit Congestion Notification
ECT	ECN-Capable Transport
EF	Expedited Forwarding
FIFO	First-In-First-Out
FQ	Fair-queuing
FCFS	First-Come, First-Served
FF	Fixed-Filter
FL	Flow Label
FEC	Forwarding Equivalence Classes
FTN	FEC to NHLFE
GFC	Generic Flow Control
HoQ	Head of Queue
HEC	Header Error Control
IntServ	Integrated Services
ISP	Internet Service Provider
ITU-T	International Telecommunications Union-Transport
ILM	Incoming Label Map
LPC	Linear Predictive Coding
LEO	Low Earth Orbit
LSR	Label Switching Router
LER	Label Edge Router
LSP	Label Switched Path
MBS	Maximum Burst Size
MCR	Minimum Cell Rate
MIB	Management Information Base
MPLS	Multi Protocol Label Switching
MMPP	Markov modulated Poisson process
MEO	Medium Earth Orbit
MOS	Mean Opinion Score
MF	Multi-Field

NAK	Negative Acknowledgement
NNI	N to-Network Interface
NE	Network Element
nrt-VBR	non-real-time Variable Bit Rate
NHLFE	Next Hop Label Forwarding Entry
OAM	Operations, Administration & Maintenance
PCM	Pulse Code Modulation
PCR	Peak Cell Rate
pdf	probability density function
PIR	Peak Information Rate
PBS	Peak Burst Size
PHB	Per Hop Behavior
PNNI	Private Network-Network Interface
PQ	Priority Queuing
PRI	Primary Rate Interface
PTI	Payload Type Identifier
PVC	Permanent Virtual Connection
PVCC	Permanent Virtual Channel Connection
PVPC	Permanent Virtual Path Connection
RED	Random Early Discard
RM	Resource Management
RV	random variable
RSVP	Resource Reservation Protocol
RSVP-TE	RSVP with Tunneling Extensions
rt-VBR	real-time Variable Bit Rate
SSS	Strict Sense Stationarity
srTCM	Single Rate Three Color Marker
SE	Shared-Explicit
SLA	Service Level Agreement
SAR	Segmentation and Re-assembly
SVC	Switched Virtual Connection
SCR	Sustainable Cell Rate
TC	Traffic Class
TCA	Traffic Conditioning Agreement
TCP	Transaction Control Protocol
TDM	Time Division Multiplexing
ToQ	Tail of Queue
ToS	Type of Service
trTCM	Two Rate Three Color Marker
TTL	Time to Live
UNI	User-to-Network Interface
UBR	Unspecified Bit Rate

VC	Virtual Connection
VoIP	Voice over IP
VP	Virtual Path
VPI	Virtual Path Identifier
VCI	Virtual Channel Identifier
VCL	Virtual Channel Link
VCC	Virtual Channel Connection
VPL	Virtual Path Link
VPC	Virtual Path Connection
WSS	Wide Sense Stationarity
WRED	Weighted Random Early Discard
WRR	Weighted Round Robin
WF	Wildcard-Filter

Index

About the author

Dr. Kun I. Park is Technical Manager at The MITRE Corporation.* Previously, he worked at Bell Labs, Bellcore, and Lucent. He held technical management positions of Supervisor at Bell Labs and District Manager and Director at Bellcore. At Bell Labs, he developed stochastic models of "crosstalk" intelligibility between telephone channels and end-to-end voice and voiceband data performance. He supervised a group responsible for digital communications performance, the pre-divestiture Bell System end-to-end voice and voiceband data performance characterization and development of a hardware/software-based ISDN protocol verification system. Through his career, he also worked on wireless, IP, ATM, optical networks, and network performance management. At Lucent, he worked on QoS and congestion control for packet networks. He holds two patents on IP and ATM congestion control methods.

Since 1986, he has been Adjunct Professor of Electrical Engineering with the rank of Full Professor at Stevens Institute of Technology and has taught, among other courses, a graduate course on probability theory and stochastic processes. He is a Senior Member of IEEE. He has published in refereed journals including *Bell Syst. Tech. Jour, IEEE Trans. on Comm.,* and *IEEE Trans. on Vehicular Tech.* He authored a book entitled, "Personal and Wireless Communications," Kluwer, 1996. He is a graduate of Seoul National University and has a Ph.D. in Electrical Engineering from the University of Pennsylvania.

* The author's affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions or viewpoints expressed by the author.