AEÜ International Journal of Electronics and Communications

Performance Analysis of an ATM Buffered Switch Transmitting Two-Class Traffic over Unreliable Channels

Hamed Nassar and Mohamed Ali Ahmed

Abstract: In this article we analyze the performance of a space division output buffered switch operating in an ATM multimedia environment as follows. Fixed size packets arrive onto the switch inputs in each time slot. These packets are of two classes. Class-1 packets, representing real time communications, are sensitive to delay but insensitive to loss. Class-2 packets, representing nonreal time communications, are insensitive to delay but sensitive to loss. The switch transmits these two-class packets over communications channels which are unreliable. That is, the packets could be lost before reaching the other end.

To respond to the class-1 delay sensitivity, the switch gives class-1 packets higher service priority over class-2 packets. And to respond to the class-2 loss sensitivity, the switch requires an acknowledgment for each class-2 packet it transmits. It is this latter response that is the major contribution of the article. In particular, it gives rise to two service times, rather than one as has usually been considered in the published literature.

For the purpose of the analysis, the switch is modelled as a priority, discrete time, batch arrival, single server queueing system, with infinite buffer and two service times: one deterministic for class-1 and one geometric for class-2. Three performance measures are analyzed: occupancy, unfinished work, and waiting time.

Keywords: Data Communications, Performance Analysis, Output Buffered Switches, Unreliable Channels, Discrete Priority Queues

1. Introduction

Space division output buffered switches of the structure shown in Figure 1 are used widely in communications networks to route traffic between two sets of nodes. The performance of these switches routing one class of traffic has been analyzed in the literature extensively. For the purpose of the analysis, the trend has been to model the switch as a queueing system, with [1] seemingly one of the initiators of this trend. However, in modern networks, e.g. the B-ISDN [2], the traffic is known to be conveniently dividable into two classes [3].

Class-1 traffic is made up of packets of real time communications, e.g. video conferences, radio and TV broadcasting, or telephone conversations. Clearly, these packets are delay sensitive but loss insensitive. That is, they should be *served* by the switch so rapidly as to arrive at their destination in time. There is no problem if this haste results in losing few packets. First, given the nature of these communications, the lost packets would hardly be noticed by the end user. Second, trying to retransmit a lost packet would be wasteful as the packet would have lost its (timely) value anyway.

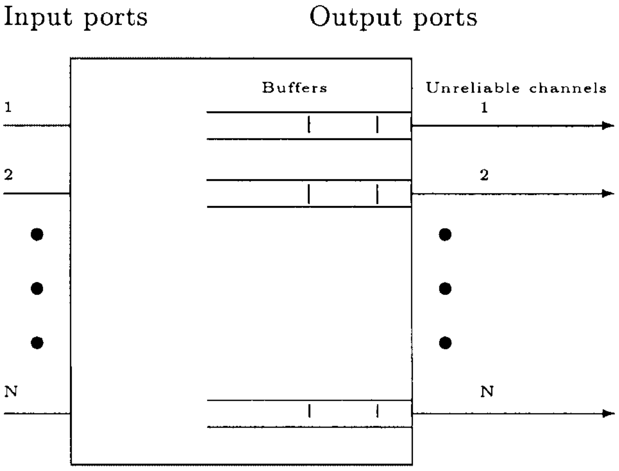

Fig. 1. An $N \times N$ output buffered space division switch whose outputs are connected to unreliable communications channels.

Class-2 traffic, on the other hand, is made up of packets of nonreal time communications, e.g. file transfer, e-mail, or routine network messages. Clearly, these packets are loss sensitive but delay insensitive. That is, they should be served by the switch so robustly as to arrive at the other end of the communications channel intact. After all, losing one byte, let alone one packet, of, say, an executable file being transferred may very well obstruct the execution of that file. Thus, the switch should make sure that each class-2 packet it transmits over a certain channel successfully reaches the other end of that channel.

The delay sensitivity of class-1 packets can be attended to by implementing a priority scheme [2] in the switch. In such a scheme class-1 packets are assigned high service priority, and class-2 packets low service priority. This entails that the system would serve class-1 packets first, until there are no more, then turn to serve class-2 packets. If such a priority scheme is implemented, one has to choose between two disciplines, concerning what happens to a class-2 packet currently in service upon the arrival of a class-1 packet. In the *preemptive* discipline, the arriving packet enters service immediately in the next slot,

Received May 30, 2002. Revised September 22, 2002.

Math. Dept., College of Science, Suez Canal University, Ismailia, Egypt. E-mail: nassar66@hotmail.com

Correspondence to H. Nassar.

1434-8411/03/57/03-190 $15.00/0

ejecting the class-2 packet back to the *buffer*. Later, when there are no more class-1 packets to serve, the ejected class-2 packet enters service again. In the *nonpreemptive* discipline, on the other hand, the arriving packet waits until the class-2 packet finishes service and then takes its place. If the preemptive discipline is chosen, one has to use on of two options, concerning how the ejected packet is treated when it comes back to service. In the *resume* option, the packet is served from the point it was ejected. In the *repeat* option, the packet is served from the start.

The loss sensitivity of class-2 packets, on the other hand, can be attended to by implementing a class-2 acknowledgment scheme in the switch. In such a scheme, the switch will *not* dispose of a class-2 packet it has transmitted unless its receipt is acknowledged by the other end of the channel. Until the acknowledgment arrives, the switch will keep automatically on retransmitting the packet, every slot, no matter how many of these retransmissions are made. As for class-1, the switch will dispose of the packet immediately after it has been transmitted for the first time. Calling the transmission time of one packet a *slot*, it can be easily seen that this acknowledgment scheme makes the service time different for each class. Specifically, the service time for a class-1 packet is the time it takes to transmit the packet, namely 1 slot. On the other hand, the service time for a class-2 packet is the time it takes to transmit the packet *successfully*, namely, a random variable (RV) that is geometrically distributed, with the channel loss probability being the distribution parameter.

Analyses of a system attending to the delay sensitivity, i.e. implementing priority schemes, abound. In [4], [5] and [6], analyses of such systems are carried out with the assumption that the service times of both classes are deterministically 1 slot. These assumptions are used also in [7] with the arrivals taken as batches of general size. In [8], the additional assumption of multiple servers is made. Extensions of the deterministic service time have appeared in many analyses, such as [9] where the arrivals are assumed to come from three-state sources, [10] where the service time for both classes is assumed geometric, and both [11] and [12] where the service time for both classes is assumed general and the priority scheme assumed preemptive resume.

However, the published literature seems to have no analysis of systems attending to both the delay and loss sensitivities, i.e. implementing both priority and class-2 acknowledgment schemes, and it is the aim of our article to carry out such an analysis. We obtain results for three performance measures: occupancy, unfinished work, and waiting time. It is worth mentioning that our analysis of the waiting time features a new approach.

The article is organized as follows. We start by formally introducing the model assumptions in Section 2. In Sections 3 and 4, we derive the Probability Generating Functions (PGFs) of the output port occupancy and unfinished work, respectively. In Section 5 we derive the PGF of the class-2 waiting time. In Section 6, we present numerical results, and in the last Section we draw some conclusions.

2. Model assumptions

First of all, it is assumed that the switch operates in a discrete time manner. That is, the time axis is divided into slots, each equal to the transmission time of one packet. Nonnegative integers $k = 0, 1, \ldots\,,$ are assigned to the individual slot boundaries. Time interval $[k, k+1)$ is referred to as slot $k+1$. Furthermore, most of the quantities considered in the article are RVs, all of them nonnegative and integral valued.

The switch has the following assumptions, largely reflected by Figure 1. There are N input ports and N buffered output ports. The arrivals at the input ports are Bernoulli processes. That is, monitoring an arbitrary input port, every slot a packet will arrive with probability r and will not arrive with probability $\overline{r} = 1 - r$. This implies that the arrival rate at any port is r packets per slot. Also, it implies that the packet interarrival time is geometrically distributed with parameter r.

Given that a packet has arrived at an input port, it is either of class-1 with probability λ or of class-2 with probability $\overline{\lambda} = 1 - \lambda$. This implies that at each input port, the class-1 arrival rate is $r_1 = \lambda r$ and the class-2 arrival rate is $r_2 = \overline{\lambda} r$. This also implies that the interarrival times of class-1 and class-2 packets are each geometrically distributed with parameters r_1 and r_2, respectively. It is clear that the packet arrival rate r, regardless of class, is related to r_1 and r_2 through the relation

$$r = r_1 + r_2\,. \qquad (1)$$

A packet that has arrived at an input port is routed in the *same* slot to its requested output. The probability that the packet requests a particular output port i is $1/N$, for all $i = 1, 2, \ldots\,, N$. The packet request is independent of the input port it arrives into.

It can be seen from the above assumptions that the traffic into the switch, out of the switch, and inside the switch is uniform. As a consequence, modelling the switch reduces to modelling an arbitrarily ‘tagged’ output port. Unless otherwise indicated, the word ‘the port’ in the sequel will refer to this tagged output port. Buffered, the port can be conveniently modelled as a queueing system. In every slot, a batch of packets arrives at the port from the input ports. These packets wait in the port until they are served out of the port, hence out of the switch.

The port can be looked upon as made up of two parts: the buffer and the server. The buffer is of infinite capacity and is used to host packets arriving from the input ports. The time the packet spends in the buffer is called *queueing* time. The server is used to host the departing packet. Physically, it could be a register. The time the packet spends in the server is called *service* time. The sum of queueing time and service time is called *waiting* time. If a packet arrives into the port, it enters either service, if there is no packet in the server, or queue, if there is a packet in the server. In either case, the entry takes place exactly at the beginning of the slot following the arrival slot. This implies that a packet is *not* considered to be in the port in its arrival slot. If a packet is being served dur-