

# Scheduling Performance Evaluation and Testing Functions of a Connection-Oriented Packet Switching Processor

Ju-Young Kim · Ki-Seok Choi<sup>†</sup>

Department of Industrial and Management Engineering, Hankuk University of Foreign Studies

## 연결지향형 패킷교환 처리기의 스케줄링 성능 평가 및 시험 방안 연구

김주영 · 최기석<sup>†</sup>

한국외국어대학교 산업경영공학과

In a connection-oriented packet switching network, the data communication starts after a virtual circuit is established between source and destination. The virtual circuit establishment time includes the queue waiting times in the direction from source to destination and the other way around. We use this two-way queueing delay to evaluate scheduling policies of a packet switching processor through simulation studies. In this letter, we also suggest user testing functions for the packet switching processor to manage virtual circuits. By detecting error causes, the user testing helps the packet switching processor provide reliable connection-oriented services.

**Keywords:** Scheduling Policy, Queueing Delay, Connection-Oriented Service

## 1. Introduction

Packet is the unit of data communication in packet switching networks like the Internet. Packet switching processor (PSP) like a router receives arriving packets from input links, stores them in buffers, processes their routing and scheduling, and transmits them over output links. In connection-oriented packet switching, a virtual circuit (VC) is established between end-to-end users before actual data transfer starts. For example, TCP is the transport layer protocol for connection-oriented applications in the Internet. X.25, Frame Relay, and ATM are other examples of connection-oriented packet switching protocols. The PSP considered in this letter mainly consists of two components as depicted in <Figure 1>; Frame

Scheduler (FS) and Packet Handler (PH). Packets from users are first stored in the user-side queues in FS. Then, FS serves the user-side queues to forward the waiting packet to PH for further processing such as virtual circuit management. In the opposite direction, packets from the network are first handled by PH before sent to FS. In FS, packets from PH are waiting in the network-side queues to be transmitted to the users. In order to schedule and forward packets, FS uses only link layer information. That is the reason, instead of packet, we pick up the word "frame" for naming FS, which is commonly used to refer data unit in link layer protocols. Network layer functions are supported by PH. Packets from the network-side links are first handled by PH. Then, they are buffered in separate queues, called network-side queues, in FS to be forwarded to appropriate user-side output links. Eventually, FS

---

This work was supported by Hankuk University of Foreign Studies Research Fund.

<sup>†</sup> Corresponding author : Professor Ki-Seok Choi, Dept. of Industrial and Management Engineering, Hankuk University of Foreign Studies, Yongin, 449-791, Korea, Tel : +82-31-330-4253, Fax : +82-31-330-4093, E-mail : kchoi@hufs.ac.kr

Received November 18, 2013; Revision Received November 27, 2013; Accepted November 27, 2013.

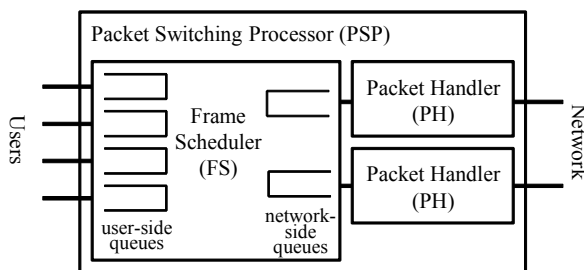
switches over multiple queues to serve user- and network-side packets. And PH processes all the network level functions like virtual circuit establishment and routing.

In this letter, we study how to improve performance of FS and PH when they conduct their main tasks; packet scheduling and VC management, respectively. In Section 2, we suggest packet scheduling policies for FS and evaluate them in a queueing delay measure. In the next section, we introduce user testing functions for PH to find out errors in VC establishment and data transfer. Finally, we conclude this letter with a few remarks.

## 2. Scheduling Performance

Scheduling policy of FS has significant impact on packet queueing delay. For performance evaluation, it is assumed that switch-over time between queues is small enough to be ignored. Another assumption is that the processing time of a packet is determined by its length whether it is from the user- or network-side. These assumptions make the total mean waiting time  $W$  constant regardless of scheduling policy as long as the scheduling policy satisfies the following conditions; FS is idle only when all the queues are empty and packet service order is independent of packet length (Cooper, 1972).

Instead of  $W$ , we consider another metric to evaluate the scheduling performance of FS; average two-way queueing delay  $W_{TW}$ , which is defined as the sum of the mean queueing delay in user- and network-side queues. In connection-oriented packet switching,  $W_{TW}$  is highly related to the virtual circuit establishment time. Generally, in order to request and accept a VC establishment, the source and destination hosts exchange small-sized control packets such as request and accept packets. If a source user requests a VC establishment, for instance, a request packet flows from the user-side input link to network-side output link. Meanwhile, it passes through the user-side queue in FS before being sent to PH.



**Figure 1.** Components of Packet Switching Processor

Once the destination user approves the VC establishment, an accept packet travels the established VC in the opposite direction; from network-side input link to user-side output link. This time the accept packet waits in the network-side queue in FS to be forwarded to the source user. Thus, the total time elapsed for the VC establishment includes the queueing delays in both directions. In data transfer phase after VC establishment,  $W_{TW}$  also has impact on network service level that users experience. In a connection-oriented packet switching, the end-user protocols usually perform window control in order to detect errors and avoid congestion. The source user can transmit data packets in a row as many as window size. Before sending more packets, the user must receive acknowledgements that the previously sent data have been received correctly. The destination informs the acknowledgements either in a piggybacked form in data packets heading to the source or by a dedicated control packet called ACK. In either way, the chances for the source to hold data transmission increase if the delay gets longer in the user- and/or network-side queues. Therefore, from the viewpoint of quality of service,  $W_{TW}$  is an important measure for evaluating the scheduling performance of FS.

To enhance  $W_{TW}$ , in other words, to reduce the queueing delays in both directions, one needs to decrease the average queueing delay difference between the user- and network-side queues. Obviously, with the assumption that the switch-over time between queues is 0, FCFS (First Come First Served) is the best scheduling policy because it equalizes the average waiting times in all queues. Under FCFS scheduling policy,  $W_{TW}$  is simply two times  $W$ , one for each direction, i.e.  $W_{TW} = 2W$ . However, to implement FCFS policy and serve packets in multiple queues according to their arrival order, FS needs additional memory space and computing capacity for recoding and processing the packet arrival times. Since it is beyond the extents of our research to compare scheduling policies including these effects on additional resource requirements, we confine ourselves to the scheduling policies which switch over to the next queue after severing a certain number of packets in a queue.

Another obvious case where  $W_{TW} = 2W$  is when the traffic is symmetric. The total mean waiting time  $W$  can be considered as the weighted sum of the user- and network-side queueing delays. Let  $W_U$  and  $W_N$  denote the average queueing delay in the user- and network-side queues, respectively. Then  $W = uW_U + nW_N$ , where  $u$  and  $n (= 1-u)$  are traffic load proportions from the user- and network-side, respectively. If the traffic is symmetric ( $u = n = 0.5$ ), then the average two-way queueing delay  $W_{TW}$  is equal to  $2W$ ;  $W_{TW} = W_U + W_N = 2(0.5W_U + 0.5W_N) = 2W$ . Therefore, we do not consider symmetric traffic in the rest of this section. Especially, we assume that more traffic arrives from the network-side than the user-side. This is a general case in a connection-oriented

packet switching where the large number of clients request data transfer from a relatively small number of servers like in the cases of FTP file download and HTTP web surfing.

In the example of <Figure 1>, there are four queues to store packets from the user-side. With  $\rho$  denoting the offered load to one user-side queue, the total user-side offered load is equal to  $4\rho$ . To study asymmetric traffic cases, as mentioned in the previous paragraph, the offered load to one of the two network-side queues in <Figure 1> is assumed to be  $4\rho$ , which makes the total network-side offered load equal to  $8\rho$  and the total offered load to FS equal to  $12\rho$ . For the simulation experiments conducted to compare scheduling policies, we assume that the total traffic load to FS is less than 1, i.e.  $\rho < 1/12$  and there is no limit on the queue length. Thus, there is no packet loss due to buffer limit and all the packets are eventually served. The packet arrival is assumed to follow a Poisson process. Since packets have a variable length in most packet switching networks, we model the packet processing time in FS as a sum of a constant 0.2 msec for performing basic tasks and a random variable following an exponential distribution with mean 0.08 msec for performing packet size dependent tasks.

We first consider an exhaustive scheduling policy in which FS switches to the next queue after serving all the packets in one queue. This is one of the popular scheduling policies for serving multiple queues cyclically (Cooper, 1970). In the example of <Figure 1>, we assume that FS forwards all the waiting packets in one network-side queue and switch to the other network-side queue. After serving the four user-side queues one by one with the same manner, FS starts to serve the network-side queues again. <Table 1> shows the simulation results with this Exhaustive scheduling policy.

Next we consider a non-exhaustive scheduling policy in which the number of packets served in a queue is limited (Chang and Down, 2000; Kühn, 1979; van Vuuren and Winands, 2006). We use the offered load to each queue to limit the maximum number of packets FS can serve each time. Since, in the example of <Figure 1>, the ratio of offered load to a network- and a user-side queue is  $4\rho : \rho = 4 : 1$ , FS serves at most four packets in each network-side queue while it serves at most one packet from a user-side queue. We call this scheduling policy Proportional to Individual Queue traffic load (PIQ) scheduling policy. <Table 2> shows the simulation results with PIQ scheduling policy. Compared to Exhaustive scheduling policy case in <Table 1>, the network-side queueing delay  $W_N$  decreases and the user-side queueing delay  $W_U$  increases in most cases while the total average queueing delay  $W$  remains at the same level. In terms of  $W_{TW}$ , PIQ scheduling policy shows worse performance than Exhaustive scheduling policy. The enlarged gap between  $W_U$  with  $W_N$  results from the fact that PIQ scheduling policy serves packets as if the network-side queues have more priority than the user-side queues. Probably, it is because the time between visiting the user-side queues gets too long with PIQ scheduling policy.

**Table 1.** Queueing Delays with Exhaustive Scheduling Policy

$\rho$	$W_U$	$W_N$	$W$	$W_{TW}$
0.01	0.021	0.020	0.020	0.041
0.02	0.048	0.045	0.046	0.092
0.03	0.088	0.079	0.082	0.167
0.04	0.150	0.130	0.137	0.280
0.05	0.245	0.205	0.219	0.450
0.06	0.424	0.340	0.368	0.763
0.07	0.899	0.681	0.754	1.580
0.08	4.261	3.142	3.515	7.403

**Table 2.** Queueing Delays with PIQ Scheduling Policy

$\rho$	$W_U$	$W_N$	$W$	$W_{TW}$
0.01	0.021	0.020	0.020	0.041
0.02	0.049	0.044	0.046	0.093
0.03	0.092	0.077	0.082	0.169
0.04	0.161	0.125	0.137	0.285
0.05	0.265	0.195	0.218	0.460
0.06	0.477	0.313	0.368	0.790
0.07	1.056	0.603	0.754	1.659
0.08	5.504	2.521	3.515	8.025

In order to improve  $W_{TW}$  performance measure, we devise another scheduling policy in which FS visits the user-side queues more often. Instead of the individual queue traffic, we use the overall one-way traffic to limit the number of packets served in a row. The new scheduling policy tries to visit the user-side as often as possible while serving each side in proportion to the total one-way offered loads. In other words, with the new scheduling policy, named Proportional to One-Way traffic load (POW) scheduling policy, FS serves at most one user-side packet in one cycle of service. Under the assumption that the user side has less total traffic than the network-side, one can easily calculate the maximum network-side packets to serve before switching to the user-side. For example, in <Figure 1>, the ratio of the total offered load to the user- and network-side queues is  $4\rho : 8\rho = 1 : 2$ . FS serves at most one user-side packet then two packets from the network side (one from each network-side queue). It has an impact on the waiting time in the individual queue in which order to visit user-side queues. However, since the visiting order changes neither  $W_U$  nor  $W_{TW}$ , we simply assume that FS visits the user-side queues cyclically. The observed

queuing delay with POW scheduling policy is summarized in <Table 3>.

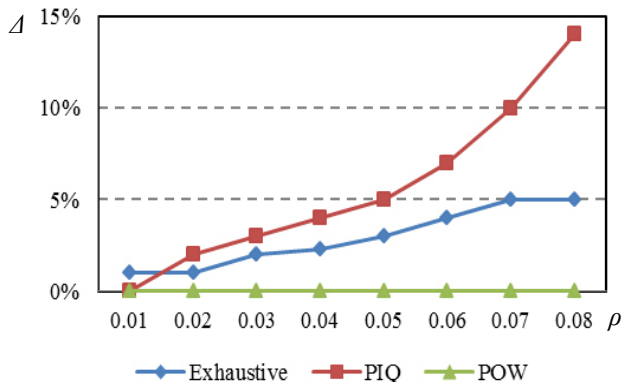
The difference between  $W_U$  with  $W_N$  reduces close to 0 except for the high offered load case ( $\rho = 0.08$ ). Even in this case,  $W_{TW}$  (7.059) decreases close to its lower limit  $2W$  ( $2 \times 3.515 = 7.030$ ). <Figure 2> plots the ratio  $\Delta := (W_{TW} - 2W) / 2W$  which indicates the performance of scheduling policies; as the ratio  $\Delta$  approaches 0, the scheduling policy shows better performance in terms of  $W_{TW}$ . It is clear from <Figure 2> that POW scheduling policy which considers the overall traffic load in each direction achieves the ultimate performance level in the sense that it schedules the user- and network-side packets fairly enough to experience almost the same queuing delay.

**Table 3.** Queuing Delays with POW Scheduling Policy

$\rho$	$W_U$	$W_N$	$W$	$W_{TW}$
0.01	0.021	0.021	0.020	0.041
0.02	0.046	0.046	0.046	0.091
0.03	0.081	0.083	0.082	0.164
0.04	0.135	0.138	0.137	0.273
0.05	0.214	0.220	0.218	0.434
0.06	0.365	0.369	0.368	0.734
0.07	0.757	0.753	0.754	1.509
0.08	3.572	3.487	3.515	7.059

### 3. Packet User Testing

In this section we suggest a user testing functions of PH in order to assure that VC establishment and other VC management procedures are carried out correctly between a certain user and PH. First, PH sets up a link layer connection to the user through FS. After sending a VC request packet to the user, PH waits for the user to reply. If the user answers with



**Figure 2.** Scheduling Policy Performance Comparison

an accept packet, a VC for testing is established and enters data transfer phase (Kim and Lee, 1996). The main role of PH in the data transfer is to forward data packets between the source and destination users. For data packet transfer, PH generates arbitrary data and transmits them towards the user. In addition to transmission testing, PH provides data receiving testing. After receiving data packets from the user, PH replies with ACK control packets. To test the acknowledgement piggybacked in data packets, PH can echo back the data packets from the user.

There are three cases for VC disconnection. First, PH finishes the test and sends a disconnect request packet to the user. Second, the user initiates VC disconnection by sending a disconnect request packet during the test. In this case, PH follows normal procedures for VC disconnection. Last, VC disconnection results from an error. If an error occurs in either VC establishment or data transfer phase, PH conducts the error handling process. The main purpose of the user testing is to recognize possible errors in VC maintenance procedures and find out what has caused the errors. Even though they are both connection-oriented, VC packet switching has more complicated connection management procedures than circuit switching (Baran, 2002). It is usually not easy for a user to find out error cause during packet exchanges. In most cases, fast and correct identification of errors requires information from PSP, especially VC information from PH. By testing a few routine VC management procedures, PH can easily obtain necessary information to pinpoint the error cause. Following are the typical errors whose cause can be identified by the user testing in PH.

- VC Establishment Error:

If an error occurs in VC establishment phase, PH indicates the proper error cause. VC establishment errors include connection request rejected by the user and internal error in PH. The PSP can find out what is the case from the error.

- Flow Control Parameter Configuration Error:

Flow control parameters like window size must be agreed between source and destination. These flow control parameters have a significant impact on the performance such as delay and throughput (Baset *et al.*, 2010; Gürbüz, 1989). Once an error occurs in data transfer phase, PH can generate and transmit packets containing flow control parameters for the user testing. The user, then, compares the parameter values with his configurations and finds out if the error has occurred due to parameter mismatching. For instance, if the receiving window size on the user is configured to be larger than the sending window size stored in PH, the user may not send ACK packets and, in return, PH stops transmitting further data. In this case, PH alarms the user and the PSP operator by sending an error message predefined for the testing. Similarly, the error caused by a larger receiving window size on PH

than the sending window size on the user can be detected when the user transmits data packets. If he is not able to transmit more data packets after sending a few, the user can attribute the error to window size mismatching.

- Data Transfer Error :

One of the user testing function in PH is to echo back the data packets from the user side. The user can transmit arbitrary data packets and compare the returned packets with the original ones. This test enables the user to check if any problem happens while sending and receiving data packets. A VC can be disconnected by an unexpected error during data transfer. PH reports every VC disconnection whether it occurs following a proper procedure or not. Abnormal VC disconnections include control errors which PH can detect. PH provides diagnosis codes to indicate that the error results from link layer disconnection, abnormal packet receptions, and so on.

## 4. Conclusions

Generally connection-oriented services are more sensitive to quality of service. In this letter, a PSP is suggested in the perspective of connection-oriented packet switching service users in order to guarantee that it delivers the required performance. Latency is one of the key performance measures in data communication and scheduling policy in PSP has a huge impact on it, especially in high traffic load environment. We consider a two-way queueing delay metric and suggest a packet scheduling policy which achieves a desirable performance level. The new scheduling policy helps to reduce both VC establishment and data transfer times. Also, when implementing the protocols for a connection-oriented packet switching, a PSP should follow the recommended procedures for VC management to avoid errors and serve packet switching users properly. When developing and implementing protocols, performance is an important issue and the developers would face a lot of problems, especially in the testing period (Choi, 2012; Paik, 2011; Rohanimanesh *et al.*, 1995). Most of the problems result from the different interpretation and implementation of protocol recommendations. If a PSP does not

provide the function of tracing the whole VC processing procedures with detailed information such as flow control parameter configurations, it is almost impossible to find out what caused errors and how to fix them. In this letter, we introduce the concept of packet user testing which helps to resolve this problem and finally assures quality of service to the users.

## References

- Baran, P. (2002), The Beginnings of Packet Switching : Some Underlying Concepts, *Communications Magazine*, **40**(7), 42-48.
- Baset, S. A., Misra, V., Rubenstein, D., and Schulzrinne, H. (2010), The Delay-Friendliness of TCP for Real-time Traffic, *IEEE/ACM Transactions on Networking*, **18**(5), 1478-1491.
- Chang, W. and Down, D. (2000), Asymptotics for Polling Models with Limited Service Policies, *Proceedings of the 39th IEEE Conference Decision and Control*, Sydney, Australia.
- Choi, W. (2012), Tighter Throughput Lower Bounds of Connectivity-Based Reliable Multicast MAC Protocol for IEEE 802.11 Wireless LANs, *Journal of the Korean Institute of Industrial Engineers*, **38**(4), 235-289.
- Cooper, R. B. (1970), Queues Served in Cyclic Order : Waiting Times, *The Bell System Technical Journal*, **49**(3), 399-413.
- Cooper, R. B. (1972), *Introduction to Queueing Theory*, Macmillan Company, New York.
- Gürbüz, Ö. (2008), Design and Performance Analysis of an ARQ Scheme for Broadband Wireless Access, *IEICE Transactions on Communications*, **E91-B**(6), 1989-1998.
- Kim, S. H. and Lee, H. H. (1996), On-Line ISDN Packet Protocol Test Using Reference Protocol, *Proceedings of the 18th AIC Conference*, Beijing, China.
- Kühn, P. J. (1979), Multi-Queue Systems with Non-Exhaustive Cyclic Service, *Bell System Technical Journal*, **58**(3), 671-698.
- Paik, C. (2011), Queueing Traffic Model of Giving a Priority to Handoff Calls in OFDMA Wireless Communication Systems, *Journal of the Korean Operations Research and Management Science Society*, **36**(3), 45-59.
- Rohanimanesh, K., Eydgahi, A. M., Ghavamzadeh, M., and Poorali, B. (1995), Design of an ISDN Terminal, *Proceedings of the 1995 IEEE IECON 21st International Conference*, Orlando, Florida.
- van Vuuren, M. and Winands, E. M. M. (2006), Iterative Approximation of  $k$ -Limited Polling Systems, *Queueing Systems*, **55**(3), 161-178.