Bounding Delays in Packet-Routing Networks

Mor Harchol-Balter* David Wolfe †

Computer Science Division
University of California at Berkeley
Berkeley, California 94720

Abstract

We consider the problem of computing the average packet delay in a general dynamic packet-routing network with Poisson input stream, during steady-state.

Any packet-routing network can be formulated as a queueing network, where each server has a constant service time and the packets are served in a first-come-first served (FCFS) order. If each server had exponentially-distributed service time, queueing theory techniques could be used to determine the expected packet delay. However, it is not known how to compute the average packet delay for all but the simplest networks with constant time servers.

It has been conjectured that to get an upper bound on expected packet delay in the constant service network, one can simply replace each constant time server with an exponential server of equal mean service time.

We prove that for a large class of networks, this conjecture is true, but that there exists a network for which it is false. This large class of networks is the Markovian queueing networks. Markovian queueing networks are important because they include many packet-routing networks where the packets are routed to random destinations.

1 Introduction

Many parallel and distributed applications require packets to be routed in a network. As packets move along their routes, they are delayed by other packets. In computing performance bounds for a given network and routing scheme, it is useful to be able to determine the time by which the average packet is delayed.

There are two general classifications of packet-routing networks: static and dynamic. Static packet-routing refers to the case where the packets to be routed are all present in the network when the routing commences. In dynamic packet-routing, packets arrive at the network at random times and the routing proceeds in a continuous fashion. In this paper we will be interested in the dynamic case, in steady state, with Poisson input stream.

Most theoretical research has concentrated on analyzing delays in the static case. The dynamic case appears more difficult to deal with using conventional techniques. The most commonly used technique for bounding the delay in packet-routing networks is to use Chernoff bounds to bound the maximum number of packets which could possibly need to traverse a given edge during a window of time (w.h.p.). Examples of research on static packet-routing networks are [Lei90], [Lei92], [VB81], [Val82], [Ale82], [Upf84], [GL85], [ALMN90], [CS86]. All of these are specific to a particular network and a particular routing scheme. They mostly concentrate on the problem of permutation routing, and use the Chernoff bound approach. Some research on static packet-routing networks applies to general networks (see [LMR88] [PU87]). This research concentrates on worst-case bounds. There are very few theoretical results for dynamic packet-routing networks. A few are [Lei90],[KL95], and [CS86]. [Lei90] and [KL95] assume a discrete Poisson arrival steam (a new packet is born at each node of the network at every second with probability p). [CS86] assume a new permutation arrives every T seconds. *Both these results are network and routing scheme specific*, and although their bounds are very strong, the analysis is very involved. Lastly, since in most of the above routing schemes packets are first sent to intermediate random destinations, there's been a lot of research which concentrates on computing delays for the case where the final destinations are random (see for example [Lei90], [Val82], [Lei92], [KL95]). Again,

*Supported by National Physical Science Consortium (NPSC) Fellowship. Also supported by NSF grant number CCR-9201092. Email: harchol@cs.berkeley.edu

†Email: wolfe@cs.berkeley.edu

except for [Lei90] and [KL95] all these results are for the static case.

For computing delays in dynamic packet-routing networks, queueing theory provides a huge body of useful results which apply to *any network configuration and routing scheme*. Unfortunately, these results rely on a few unrealistic assumptions (the most unrealistic being independent exponentially distributed service times), and therefore people are reluctant to make use of them. In this paper we discuss when the assumption of independent exponential service times can be replaced by (the much more realistic) constant service times. More specifically (see Section 1.3.3):

- Packet-routing networks may be described as queueing networks with constant-time, FCFS servers, where each server in the queueing network corresponds to an edge (or any other bottleneck) in the packet-routing network.

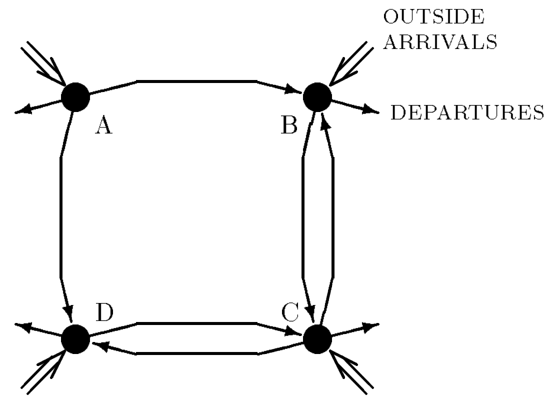
- We prove that if the queueing network is Markovian, replacing constant time servers with exponentially-distributed time servers of equal mean service time increases average packet delay. Since queueing networks with FCFS, exponential time servers are easily solvable for average delay, we therefore have an upper bound on the average delay in any packet-routing network.

- We show there exists a (Non-Markovian) queueing network for which replacing constant time servers with exponential time servers (of equal mean service time) decreases average packet delay.

1.1 Definition of Packet-Routing Network

A packet-routing network consists of nodes with wires connecting the nodes, as shown in Figure 1. Packets arrive continuously from outside the network at the nodes of the network. Each packet is born with a path. For example, in the routing scheme of Figure 1, packets with path $A \longrightarrow B \longrightarrow C \longrightarrow$ are born at a rate of one every 10 seconds, and packets with path $B \longrightarrow C \longrightarrow B \longrightarrow C \longrightarrow$ are born at a rate of one every 20 seconds, etc. Most literature considers the edges (wires) of the packet-routing network to be the bottlenecks. Specifically, it takes some constant time to traverse an edge (this constant may be different for each edge), and only one packet may traverse the edge at a time. The packets traverse the edge in a FCFS (first-come-first-served) order. This causes a packet to be delayed when it arrives at an edge that is currently being used. The nodes of the network serve only to route the packets from one edge to the next. In our analysis, it is equally easy to assume the nodes of the network also form bottlenecks (in the same way as the edges).

In this paper we'll be interested in computing the time an average packet is delayed by waiting in queues.



Possible packet path	Average rate of birth
A →B →C →	1 every 10 sec
A →B →C →B →	1 every 15 sec
B →C →B →C →	1 every 20 sec
⋮	⋮

Figure 1: *A packet-routing network*

1.2 Queueing Network Definitions and Background

A *queueing network* $\mathcal{N}$ consists of servers with edges connecting the servers, as shown in Figure 2. It's behavior is very similar to our definition of a packet-routing network, except that time is only spent at the servers, and no time is spent on the edges. (Thus packets queue up at the servers of $\mathcal{N}$). Packets arrive continuously at the servers of $\mathcal{N}$, and each packet has a path associated with it which it follows. A queueing network is defined by 3 parameters:

service-time distribution The service time associated with a server is a random variable from a distribution. (Note the distribution — or just its mean — may be different for each server).

resolution protocol The order in which packets are served in case of conflict at a server.

outside arrival process In this paper, whenever we speak of a queueing network, we will assume that outside arrivals occur at each server according to a Poisson process.

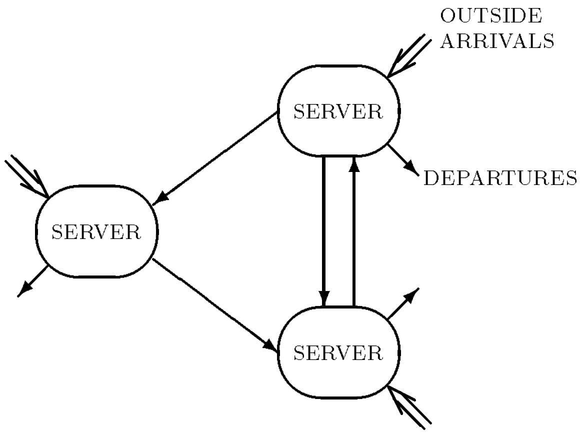


Figure 2: *A queueing network*

A *Markovian queueing network* is a special case of a queueing network. In a Markovian queueing network, when a packet finishes serving at a server i, the probability that it next moves to some server j (or leaves the network) depends only on where the packet last served and is independent of its previous history (or route). In this case the packets appear indistinguishable. Thus a Markovian queueing network can simply be described by a directed graph with probabilities on the edges.[1]

Given a queueing network $\mathcal{N}$, we define $\mathcal{N}_{C,FCFS}$ (respectively, $\mathcal{N}_{E,FCFS}$) to be queueing network $\mathcal{N}$ where each server has a constant (respectively, exponentially distributed) service time with the same mean as the corresponding server in $\mathcal{N}$, and the packets are served in a First-Come-First-Served order.

1.2.1 Recasting a packet-routing network $\mathcal{P}$ as a queueing network $\mathcal{N}_{C,FCFS}$

Observe that every packet-routing network $\mathcal{P}$ may be described as a queueing network of type $\mathcal{N}_{C,FCFS}$ as follows: Corresponding to each bottleneck in $\mathcal{P}$, we create a server in $\mathcal{N}$. For example, the edges of the packet-routing network are bottlenecks, so we create one server in $\mathcal{N}$ corresponding to each edge in $\mathcal{P}$. Each server has a FCFS order of service, since only one packet at a time may traverse an edge in $\mathcal{P}$. Each server has constant service time equal to the time required to traverse the corresponding edge in $\mathcal{P}$.

Thus from now on, we will never refer to packet-routing networks again, but rather we will only address how to compute delays in queueing networks of type $\mathcal{N}_{C,FCFS}$. Unfortunately, it is not known how to compute average packet delay for all but the simplest $\mathcal{N}_{C,FCFS}$ networks, since $\mathcal{N}_{C,FCFS}$ networks don't have product-form. However, $\mathcal{N}_{E,FCFS}$ is a product-form network (more specifically it's a classed Jackson queueing network, see [BS93]) and therefore the average packet delay is easy to determine for networks of this type (see, for example, [Wal89] [BS93]).

> *The objective of this paper is therefore to bound the average delay of $\mathcal{N}_{C,FCFS}$ (which we care about) by the average delay of $\mathcal{N}_{E,FCFS}$ (which we know how to compute).*

1.3 In This Paper We Show ...

1.3.1 Overall Goal

Our overall goal is to identify for which $\mathcal{N}$

$$\text{AvgDelay}(\mathcal{N}_{C,FCFS}) \leq \text{AvgDelay}(\mathcal{N}_{E,FCFS}) \qquad (1)$$

1.3.2 Previous Work on Goal

All previous work seems to indicate (1) holds for all queueing networks $\mathcal{N}$.

For example, the average packet delay is an increasing function of the variance in the service time distribution for each of the following single queue networks: the M/G/1 queue, the M/G/1 queue with batch arrivals, the M/G/1 queue with priorities, and the M/G/k queue, [Whi83] [Whi80] [Ros89, pp. 353–356].

With respect to networks of queues, [ST91] showed that for all *layered*[2] Markovian networks $\mathcal{N}$, $\mathcal{N}_{E,FCFS}$ has greater average packet delay than $\mathcal{N}_{C,FCFS}$. There are also simulation studies of several non-Markovian networks $\mathcal{N}$ (i.e. general classed networks) which find the average packet delay to be greater for $\mathcal{N}_{E,FCFS}$ than for $\mathcal{N}_{C,FCFS}$ (see [HBB94] [MC86] [HC86]) .

With respect to how tight this upper bound is, in all of the above simulations the average delay in $\mathcal{N}_{E,FCFS}$ was never greater than that of $\mathcal{N}_{C,FCFS}$ by more than a factor of 3 (this included networks loaded to 99% of capacity and having 100 servers). However, since the difference increases both with the load *and* with the number as servers (see for example Section 2.1 and also [KL95]), this ratio could be greater for large networks.

The above results have led to a general belief that greater variance in service times leads to greater average packet delay [Whi84] [Wal94] [Fer94] [Kle94]. In Section 2.1, we give some intuition for this. Counterexamples to this theory have only been found in the case where arrivals are not Poisson [Wol77] [Ros78]. For example Figure 3 indicates why counterexamples can be

[1] Note an equivalent, but more elegant, way to define a queueing network $\mathcal{N}$ is to say that each outside arrival to $\mathcal{N}$ is associated with some class. A packet of class ℓ moves from server i to server j next with probability p_{ij}^{ℓ}. The special case of a *Markovian* network $\mathcal{N}$ is defined as a network with only one class of packets.

[2] A packet-routing network is layered if its corresponding queueing network is acyclic.

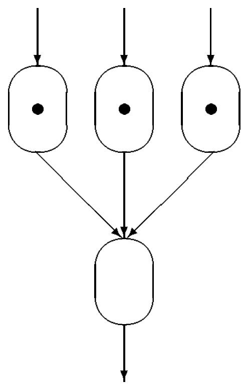


Figure 3: *Non-Poisson (in this case batch-Poisson) arrivals can favor more variance in service distributions. For example, if three packets arrive in a batch (serving in the top three servers above), they'll collide at the next server unless their service-completions are staggered.*

found which use *batch Poisson* arrivals such as those in [Wol77]. The final thing we do in this paper is to demonstrate a counterexample for the case of Poisson arrivals.

1.3.3 Our Results

- (Section 2) For all Markovian queueing networks $\mathcal{N}$,

  $$\text{AvgDelay}(\mathcal{N}_{C,FCFS}) \leq \text{AvgDelay}(\mathcal{N}_{E,FCFS})$$

  Significance of this result: Recall that computing delays in packet-routing networks when the packets have random destinations is important because most randomized routing algorithms consist of two random routing problems (see the third paragraph of the introduction). Markovian queueing networks are important because they include many packet-routing networks in which the packets have random destinations. A couple common examples are the mesh network with greedy routing (packets are first routed to the correct column and then to the correct row) and the hypercube network with canonical routing (packets cross each dimension if needed in order). When the packets have random destinations, rather than choosing the random destination when the packet is born, we can view the random destination as being decided a little at a time, by flipping a coin after each server.[3] This result tells us that we can easily compute an upper bound on the average delay for any packet-routing network which can be modeled by a Markovian queueing networks. Also, Section 1.3.2 cites evidence that this upper bound is not far from tight in practice.

- (Section 3) There exists (a non-Markovian) network $\mathcal{N}$, s.t.

  $$\text{AvgDelay}(\mathcal{N}_{C,FCFS}) > \text{AvgDelay}(\mathcal{N}_{E,FCFS})$$

  Significance of this result: The counterexample disproves the widely believed conjecture that for all networks $\mathcal{N}_{C,FCFS}$ has better average delay than $\mathcal{N}_{E,FCFS}$ (see Section 1.3.2).

[3] Observe that since the server in the queueing network represents an edge in the packet-routing network, all we need to know to determine the probabilities is the server (edge) at which the packet just finished serving.

2 Upper Bounding Average Delay in Markovian Queueing Networks

In this section we will prove the following theorem:

Theorem 1 *For all Markovian queueing networks $\mathcal{N}$,*

$$\textit{AvgDelay}(\mathcal{N}_{C,FCFS}) \leq \textit{AvgDelay}(\mathcal{N}_{E,FCFS})$$

[ST91] proved this result for layered Markovian networks. Our proof parallels their proof, except that whereas their proof uses induction on the layers of the network, we induct on time, thereby obviating the need for a layered network.

Let $\mathcal{N}$ be a Markovian queueing network. Define $\mathcal{N}_{C,PS}$ to be the queueing network $\mathcal{N}$ where each server has a constant service time with the same mean as the corresponding server in $\mathcal{N}$ and the service order is Processor Sharing. (In Processor Sharing, the server is shared equally by all the packets currently waiting at the server. So, for example if the service time at the server is 2, and there are 3 packets waiting at the server, each packet is being served at a rate of $\frac{1}{6}$). By [BCMPG75] and [Kel75], we know that the average packet delay in $\mathcal{N}_{C,PS}$ is equal to the average packet delay in $\mathcal{N}_{E,FCFS}$ for all $\mathcal{N}$.[4] It is therefore sufficient to prove[5]

$$\begin{aligned}\text{AvgDelay}(\mathcal{N}_{C,FCFS}) &\leq \text{AvgDelay}(\mathcal{N}_{C,PS})\\ &= \text{AvgDelay}(\mathcal{N}_{E,FCFS})\end{aligned}$$

We start by proving the inequality for a single server network.

[4] This powerful theorem is also described more recently in [Wal89] and [Kle76].

[5] Our proof of the inequality is valid for *any* sequence of outside arrivals, not just a Poisson arrival stream.

Claim 1 *If the sequence of arrivals to a (single server) FCFS queue is no later than the arrivals to a PS queue, then the i^{th} departure from the FCFS queue occurs no later than the i^{th} departure from the PS queue.*

Proof: In both queues, each packet must wait for all packets with earlier arrivals to depart, but only in the PS queue must a packet also wait while later arrivals get service. ■

To generalize the statement from the single server to the network, we'll use a coupling argument. Consider the behavior of the two networks when coupled to run on the same *sample point* consisting of:

1. the sequence of *outside* inter-arrival times at each server, and
2. the choices for where the j^{th} packet served at each server proceeds next.

Note the above quantities are all independent for a Markovian network. Also, the j^{th} packet to complete at a particular server in the two networks may not be the same packet.

Claim 2 *For a given sample point, the j^{th} service completion at any server of the FCFS network occurs no later than the j^{th} service completion at the corresponding server of the PS network.*

Proof: Assume the claim is true at time t. We show it's true at time $t' > t$, where t' is the time of the next service completion. We distinguish between *outside arrivals* to a server (packets arriving from outside the network) and *inside arrivals* to the server (service completions), and make the following sequence of observations:

- During $[0, t')$, Claim 2 is true.
- During $[0, t')$, every arrival at any PS server, q, must have already occurred at the corresponding FCFS server, q (see Figure 4). (This is true for inside arrivals because any inside arrival at PS server q is, say, the j^{th} service completion at some PS server q', and by the previous observation, the j^{th} service completion at FCFS server q' is at least as early. By definition of the sample point, the observation is also true for outside arrivals.)
- Therefore, during $[0, t')$, the i^{th} packet to arrive at any server of the FCFS network arrives no later than the i^{th} arrival at the corresponding server of the PS network.
- Hence, by Claim 1, we see Claim 2 holds during $[0, t']$. This includes the current service completion.

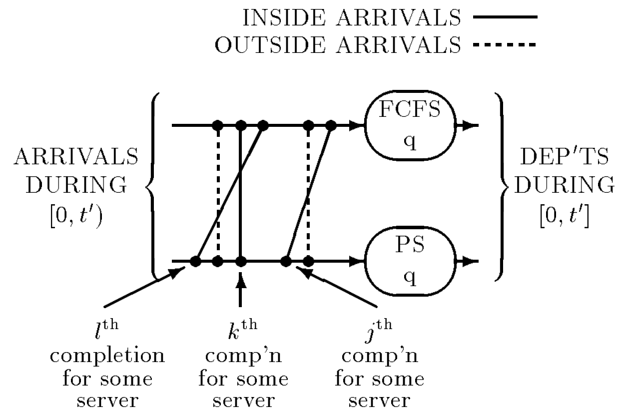


Figure 4: *Illustration of proof of Claim 2. We consider the same server q in the FCFS network and the PS network. The arrival stream into PS server q is delayed relative to the arrival stream into FCFS server q. Of course, the order of arrivals may be different for the two queues, but for a Markovian network, packets are indistinguishable anyway.*

■

By Claim 2, it follows that for any sample point, the i^{th} departure from $\mathcal{N}_{C,FCFS}$ occurs no later than the i^{th} departure from $\mathcal{N}_{C,PS}$. This implies that

$$\begin{aligned} &\mathbf{E}\{\text{Number of packets in } \mathcal{N}_{C,FCFS} \text{ at time t}\} \\ &\quad \leq \quad \mathbf{E}\{\text{Number of packets in } \mathcal{N}_{C,PS} \text{ at time t}\} \end{aligned}$$

So by Little's Law [Wol89, p. 236] we have shown that

$$\text{AvgDelay}(\mathcal{N}_{C,FCFS}) \leq \text{AvgDelay}(\mathcal{N}_{C,PS})$$

and therefore proved Theorem 1 above.

2.1 How much worse is PS than FCFS?

In section 1.3.2 we stated that simulations indicate that the average delay in $\mathcal{N}_{C,PS}$ is always within a factor of 3 of the average delay in $\mathcal{N}_{C,FCFS}$. However, in this section we will show that when the number of servers, n, in a network is very large, this difference might be much greater. Consider a queueing network $\mathcal{N}$ consisting of only a single line of n servers, each with service time 1. Packets arrive only at the first server, and leave the network after serving at the nth server. $\mathcal{N}_{C,FCFS}$ (respectively, $\mathcal{N}_{C,PS}$) is the network $\mathcal{N}$ where the service resolution protocol is FCFS (respectively, Processor-Sharing). To determine the average delay in each network, consider the delay experienced by a newly-arriving packet p. In both $\mathcal{N}_{C,FCFS}$ and $\mathcal{N}_{C,PS}$, p is delayed by the packets it finds queued up at the first server (in $\mathcal{N}_{C,PS}$

later arrivals also cause p to be delayed, but we ignore them). The difference is that in $\mathcal{N}_{C,FCFS}$, these packets only each delay p by 1 (after that initial delay the packets are spread out and move in lockstep), whereas in $\mathcal{N}_{C,PS}$, these packets each delay p by n (since the packets all move in a "clump" down the network).

3 A Non-Markovian Counterexample

In this section, we demonstrate an $\mathcal{N}$ for which

$$\mathrm{AvgDelay}(\mathcal{N}_{C,FCFS}) > \mathrm{AvgDelay}(\mathcal{N}_{E,FCFS})$$

More specifically, defining $\mathcal{N}_{C,PS}$ as in Section 2, we will demonstrate a network $\mathcal{N}$ for which

$$\begin{aligned} \mathrm{AvgDelay}(\mathcal{N}_{C,FCFS}) &> \mathrm{AvgDelay}(\mathcal{N}_{C,PS}) \\ &= \mathrm{AvgDelay}(\mathcal{N}_{E,FCFS}) \end{aligned}$$

For some insight into why it is counterintuitive that such a network $\mathcal{N}$ exists, see Section 2.1.

3.1 Network Description

Let $\mathcal{N}$ be the queueing network shown in Figure 5. The servers in $\mathcal{N}$ either have service time 1 or ϵ, as shown. The only outside arrivals are into the top server. Packets arrive from outside $\mathcal{N}$ according to a Poisson Process with rate $\lambda = \frac{1}{n^3}$, where n is the number of servers of mean service time 1 in $\mathcal{N}$. Half the arriving packets are of type *solid* and half are of type *dashed* (by "type" we mean class). Packets of type *solid* are routed straight down, only passing through the time 1 servers. Packets of type *dashed* are routed through the dashed edges, i.e. through all the ϵ servers and through every other 1-server.

3.2 Intuition

We will compare the average delay in $\mathcal{N}_{C,FCFS}$ with the average delay in $\mathcal{N}_{C,PS}$, as shown in Figure 6, by comparing the average delay experienced by an arriving packet p at $\mathcal{N}_{C,FCFS}$ and $\mathcal{N}_{C,PS}$. Throughout our argument, we implicitly use PASTA (Poisson Arrivals See Time Averages).

The intuition behind the analysis is as follows: Since λ is so low, usually for either network, p will see no other packets during its traversal of the network. In this case $\mathcal{N}_{C,FCFS}$ behaves identically to $\mathcal{N}_{C,PS}$. With some probability, however, one other packet will be present in the network during p's traversal of the network. The expected delay on p in this case is greater for the $\mathcal{N}_{C,FCFS}$ network than for the $\mathcal{N}_{C,PS}$ network. Figure 7 shows us

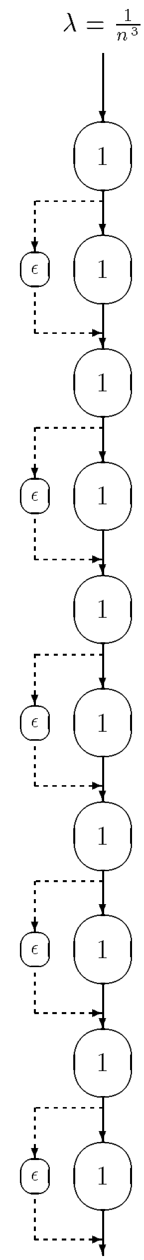


Figure 5: *Counterexample network $\mathcal{N}$ with n servers of mean service time 1 and $n/2$ with mean service time ϵ. Packets arrive at the top; half follow the dashed route, while half follow the solid route.*

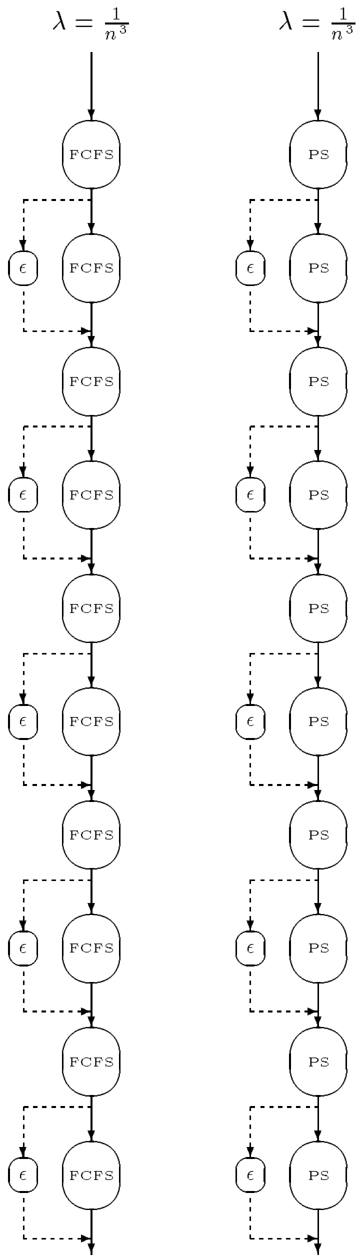


Figure 6: $\mathcal{N}_{C,FCFS}$ *and* $\mathcal{N}_{C,PS}$ *networks*

why: Consider first $\mathcal{N}_{C,FCFS}$. Suppose q is of type *solid* and some packet p of type *dashed* enters $\mathcal{N}_{C,FCFS}$ within $\frac{n}{2}$ seconds after q. Then p will eventually catch up to q, and from this point on, q will delay p by one second at every other server throughout the rest of the $\mathcal{N}_{C,FCFS}$. That is, p will be delayed by $\Theta(n)$ seconds. Now observe that the same scenario would only cause a delay of at most 2 seconds in $\mathcal{N}_{C,PS}$, because when p catches up to q, it will only interfere with q for two servers and then p will pass q forever. A worse situation for $\mathcal{N}_{C,PS}$ is the case where p meets up with another packet of the same type as p during its traversal (since in that case p is clearly delayed by $\Theta(n)$). Observe, however, that this scenario can only happen if the two packets both arrived at $\mathcal{N}_{C,PS}$ within a second of each other. This occurs with such low probability for our choice of small λ that the scenario's affect on average delay is negligible.

Lastly, we have to consider the case that two or more packets are present in the network during p's traversal of the network. The expected delay on p in this case is greater for the $\mathcal{N}_{C,PS}$ network than for the $\mathcal{N}_{C,FCFS}$ network, but this case occurs with such low probability that its effect on p's delay is also negligible.

3.3 The details

Define

$$P_i^{kn} \quad = \quad \mathbf{Pr}\left\{i \text{ arrivals in the last } kn \text{ seconds}\right\}.$$

Recall that arrivals are Poisson with rate $\lambda = 1/n^3$, where n is the number of servers of mean service time 1. So,

$$\begin{aligned} P_i^{kn} &\approx \frac{k^i}{n^{2i}} \\ P_i^{kn} &= \Theta\left(\frac{1}{n^{2i}}\right), \text{ for fixed } i,k \end{aligned}$$

By PASTA, the expected delay a newly arriving packet experiences is equal to the average packet delay for the network. We will compute an upper bound on the delay an arrival experiences in $\mathcal{N}_{C,PS}$ and a lower bound on the delay an arrival experiences in $\mathcal{N}_{C,FCFS}$. We will show

$$\begin{aligned} &\text{lowerbound}\left(\mathbf{E}\left\{\begin{array}{c}\text{Delay on arrival}\\ \text{in } \mathcal{N}_{C,FCFS}\end{array}\right\}\right) \\ &\quad > \text{upperbound}\left(\mathbf{E}\left\{\begin{array}{c}\text{Delay on arrival}\\ \text{in } \mathcal{N}_{C,PS}\end{array}\right\}\right). \end{aligned}$$

3.3.1 Upperbound $\mathbf{E}\left\{\textbf{Delay on arrival in } \mathcal{N}_{C,PS}\right\}$

Let p represent an arriving packet in $\mathcal{N}_{C,PS}$.

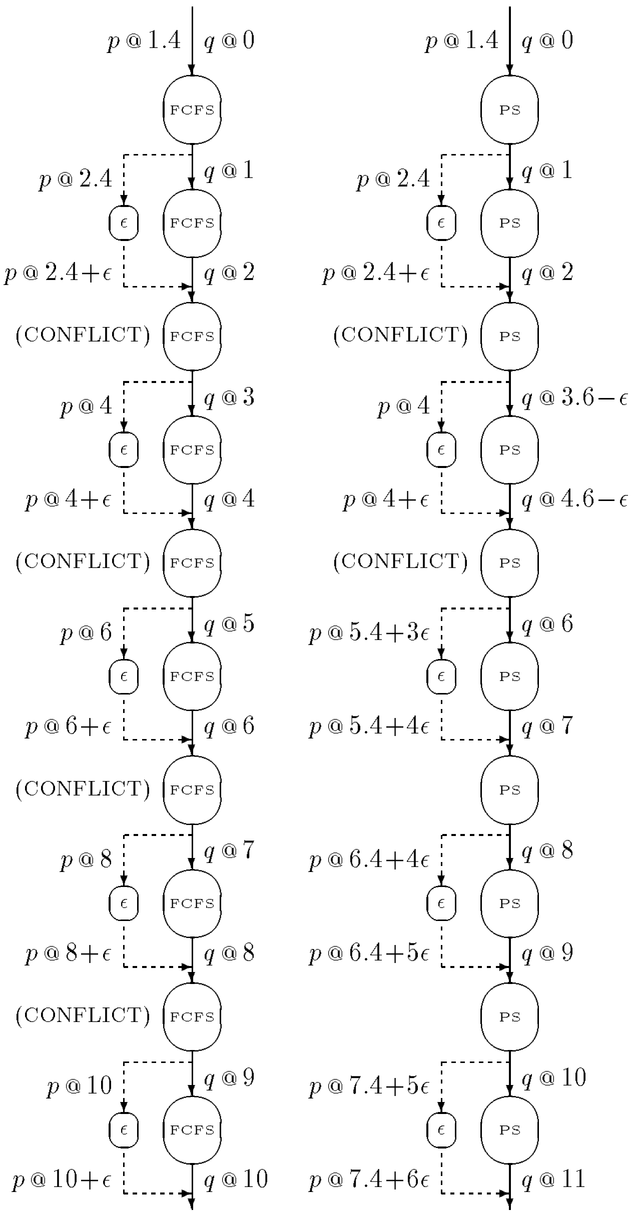


Figure 7: *Example illustrating how a packet, p, of type dashed and a packet, q, of type solid clash repeatedly in $\mathcal{N}_{C,FCFS}$, but only twice in $\mathcal{N}_{C,PS}$.*

Clearly, p may only be delayed by packets which are in $\mathcal{N}_{C,PS}$ during the time p is in $\mathcal{N}_{C,PS}$. Note that if k packets are in $\mathcal{N}_{C,PS}$, they may take up to time kn to clear the system. So, if we call p's arrival time 0, packet p *may* be delayed if one of the following occur:

- 1 other packet arrives during $(-n, n)$.
- 2 other packets arrives during $(-2n, 2n)$.
- 3 other packets arrives during $(-3n, 3n)$.
- etc.

We will compute the expected delay on p due to each of the above events, and then we'll sum these. This will be an overcount, but that's o.k. because we're just upperbounding.

$$
\begin{aligned}
& \mathbf{E}\left\{ \begin{array}{c} \text{Delay on } p \text{ caused by 1 other} \\ \text{packets arriving in } (-n, n) \end{array} \right\} \\
= \;& \mathbf{E}\left\{ \begin{array}{c} \text{Delay on } p \text{ caused by 1 packet} \\ \text{of same type arriving in } (-n, n) \end{array} \right\} \\
& + \mathbf{E}\left\{ \begin{array}{c} \text{Delay on } p \text{ caused by 1 packet} \\ \text{of opposite type arriving in } (-n, n) \end{array} \right\} \\
= \;& \mathbf{Pr}\left\{ \begin{array}{c} \text{same type} \\ \text{arrival} \end{array} \right\} \cdot \mathbf{E}\left\{ \text{Delay} \;\middle|\; \begin{array}{c} \text{same type} \\ \text{arrival} \end{array} \right\} \\
& + \mathbf{Pr}\left\{ \begin{array}{c} \text{opp. type} \\ \text{arrival} \end{array} \right\} \cdot \mathbf{E}\left\{ \text{Delay} \;\middle|\; \begin{array}{c} \text{opp. type} \\ \text{arrival} \end{array} \right\} \\
= \;& \Theta\left(\frac{1}{n^2}\right) \cdot \Theta\left(\frac{1}{n} \cdot n\right) \quad \begin{array}{c} \text{(delayed by } n \text{ only if packet} \\ \text{arrived in } (-1, 1)) \end{array} \\
& + \Theta\left(\frac{1}{n^2}\right) \cdot \Theta(1) \quad \begin{array}{c} \text{(opposite type packet causes} \\ \text{at most delay of } \Theta(1)) \end{array} \\
= \;& \Theta\left(\frac{1}{n^2}\right)
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
& \mathbf{E}\left\{ \begin{array}{c} \text{Delay on } p \text{ caused by 2 other} \\ \text{packets arriving in } (-2n, 2n) \end{array} \right\} \\
\leq \;& \mathbf{Pr}\left\{ \begin{array}{c} \text{2 packets} \\ \text{arrive in } (-2n, 2n) \end{array} \right\} \\
& \times \mathbf{Max}\left\{ \text{Delay} \;\middle|\; \begin{array}{c} \text{2 arrivals} \\ \text{in } (-2n, 2n) \end{array} \right\} \\
= \;& \Theta\left(\frac{1}{n^4}\right) \cdot O(2n) = \Theta\left(\frac{1}{n^3}\right)
\end{aligned}
$$

And,

$$
\begin{aligned}
& \mathbf{E}\left\{ \begin{array}{c} \text{Delay on } p \text{ caused by 3 other} \\ \text{packet arriving in } (-3n, 3n) \end{array} \right\} \\
= \;& \Theta\left(\frac{1}{n^6}\right) \cdot O(3n) = \Theta\left(\frac{1}{n^5}\right)
\end{aligned}
$$

To compute an upper bound on $\mathbf{E}\{\text{Delay on } p\}$, we sum the above terms. From the above computations its clear that the delay on p from the remaining summands not shown above is negligible. We find that

$$\mathbf{E}\{\text{Delay on } p\} = O\left(\frac{1}{n^2}\right)$$

3.3.2 Lowerbounding $\mathbf{E}\{\text{Delay on arrival in } \mathcal{N}_{C,FCFS}\}$

Let p represent an arriving packet in $\mathcal{N}_{C,FCFS}$. Assume p arrives at $\mathcal{N}_{C,FCFS}$ at time 0. To lowerbound the $\mathbf{E}\{\text{Delay on } p \text{ in } \mathcal{N}_{C,FCFS}\}$, we consider only the delay on p caused by 1 packet arriving during $(-n, n)$. Observe that if 1 packet (other than p) arrived during $(-n, n)$, and if the packet was of a different type than p, then p and the packet would meet, and the delay caused to p (if p is dashed) is $\Theta(n)$.

$$\begin{aligned}
&\mathbf{E}\left\{\begin{array}{c}\text{Delay on } p \text{ caused by 1 other} \\ \text{packet arriving in } (-n,n)\end{array}\right\} \\
&= \mathbf{Pr}\{1 \text{ other arrives during } (-n,n)\} \\
&\quad \times \mathbf{E}\{\text{Delay} \mid 1 \text{ arrival in } (-n,n)\} \\
&= \Theta\left(\frac{1}{n^2}\right) \cdot \Theta(n) \qquad \text{(see Intuition Section)} \\
&= \Theta\left(\frac{1}{n}\right)
\end{aligned}$$

Thus,

$$\mathbf{E}\{\text{Delay on } p\} = \Omega\left(\frac{1}{n}\right)$$

4 Conclusion and Future Work

We started this paper by formulating any dynamic packet routing network as a queueing network of type $\mathcal{N}_{C,FCFS}$. Since queueing theory only provides us with results on $\mathcal{N}_{E,FCFS}$, our goal became to bound the average delay of $\mathcal{N}_{C,FCFS}$ by the average delay of $\mathcal{N}_{E,FCFS}$:

$$\text{AvgDelay}(\mathcal{N}_{C,FCFS}) \leq \text{AvgDelay}(\mathcal{N}_{E,FCFS}) \qquad (2)$$

We first proved that (2) holds for all Markovian queueing networks. This result was significant because many packet-routing networks where the packets have random destinations can be formulated as Markovian queueing networks. We then gave a counterexample showing that (2) does not always hold, contrary to popular belief.

There are three natural open questions raised by these results. Let $\mathcal{S}$ be those networks for which (2) holds. The first is *"How large is the set $\mathcal{S}$?"* We know $\mathcal{S}$ contains more than just Markovian networks. For instance it's easy to prove that $\mathcal{S}$ contains the network $\mathcal{N}$ which consists of just a single server, where each incoming packet serves once, goes back to the end of the queue, and then serves a second time. Also, simulations suggest $\mathcal{S}$ contains many other non-Markovian networks (see Section 1.3.2). In fact, the difficulty in constructing a network not in $\mathcal{S}$ leads us to speculate that almost all networks are in $\mathcal{S}$.

This leads us to the second question of *"How tight an upper bound is $\mathcal{N}_{E,FCFS}$ on $\mathcal{N}_{C,FCFS}$ with respect to average delay?"*, both in practice and theoretically.

Lastly, *"For the networks not in $\mathcal{S}$, how far off is the AvgDelay($\mathcal{N}_{C,FCFS}$) from the AvgDelay($\mathcal{N}_{E,FCFS}$)?"*

References

[Ale82] R. Aleliunas. Randomized parallel communication. In *ACM-SIGOPS Symposium on Principles of Distributed Systems*, pages 60–72, 1982.

[ALMN90] Bill Aiello, Tom Leighton, Bruce Maggs, and Mark Newman. Fast algorithms for bit-serial routing on a hypercube. In *2nd Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 55–64, July 1990.

[BCMPG75] F. Baskett, K.M. Chandy, R.R. Muntz, and F. Palacios-Gomez. Open, closed, and mixed networks of queues with different classes of customers. *Journal of the Association for Computing Machinery*, 22:248–260, 1975.

[BS93] John A. Buzacott and J. George Shanthikumar. *Stochastic Models of Manufacturing Systems.* Prentice Hall, 1993.

[CS86] Yukon Chang and Janos Simon. Continuous routing and batch routing on the hypercube. In *Proceedings of the 5th Annual ACM Symposium on Principles of Distributed Computing*, pages 272–278, 1986.

[Fer94] Dominico Ferrari, 1994. Personal Communication.

[GL85] Ronald I. Greenberg and Charles E. Leiserson. Randomized routing on fat-trees. In *26th Annual Symposium on Foundations of Computer Science*, pages 241–9, October 1985.

[HBB94] Mor Harchol-Balter and Paul E. Black. Queueing analysis of oblivious packet-routing algorithms. In *Proceedings of the*

Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 583–592, January 1994.

[HC86] Bruce Hajek and Rene L. Cruz. Delay and routing in interconnection networks. In *Flow Control of Congested Networks*, pages 235–242, 1986.

[Kel75] F. P. Kelly. Networks of queues with customers of different types. *Journal of Applied Probability*, 12:542–554, 1975.

[KL95] Nabil Kahale and Tom Leighton. Greedy dynamic routing on arrays. In *Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 558–566, January 1995.

[Kle76] Leonard Kleinrock. *Queueing Systems Volume II: Computer Applications*. John Wiley and Sons, New York, 1976.

[Kle94] Len Kleinrock, 1994. Personal Communication.

[Lei90] Tom Leighton. Average case analysis of greedy routing algorithms on arrays. In *2nd Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 2–10, July 1990.

[Lei92] Tom Leighton. Methods for message routing in parallel machines. In *24th Annual ACM Symposium on Theory of Computing*, pages 77–96, May 1992.

[LMR88] Tom Leighton, Bruce Maggs, and Satish Rao. Universal packet routing algorithms. In *29th Annual Simposium on Foundations of Computer Science*, pages 256–69, October 1988.

[MC86] Debasis Mitra and R. Cieslak. Randomized parallel communications. In *International Conference on Parallel Processing*, pages 224–230, 1986.

[PU87] David Peleg and Eli Upfal. The generalized packet routing problem. *Theoretical Computer Science*, 53(2–3):281–93, 1987.

[Ros78] Sheldon M. Ross. Average delay in queues with non-stationary Poisson arrivals. *Journal of Applied Probability*, 15:602–609, 1978.

[Ros89] Sheldon M. Ross. *Introduction to Probability Models*. Academic Press, Inc., Boston, 1989.

[ST91] George D. Stamoulis and John N. Tsitsiklis. The efficiency of greedy routing in hypercubes and butterflies. In *3rd Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 248–59, July 1991.

[Upf84] Eli Upfal. Efficient schemes for parallel communication. *Journal of the ACM*, 31:507–517, 1984.

[Val82] L. G. Valiant. A scheme for fast parallel communication. *SIAM Journal on Computing*, 11(2):350–61, May 1982.

[VB81] L.G. Valiant and G.J. Brebner. Universal schemes for parallel communication. In *13th Annual ACM Symposium on Theory of Computing*, pages 263–277, May 1981.

[Wal89] Jean Walrand. *Introduction to Queueing Networks*. Prentice Hall, New Jersey, 1989.

[Wal94] Jean Walrand, 1994. Personal Communication.

[Whi80] Ward Whitt. The effect of variability in the $GI/G/s$ queue. *Journal of Applied Probability*, 17(4):1062–1071, 1980.

[Whi83] Ward Whitt. Comparison conjectures about the $M/G/s$ queue. *Operations Research Letters*, 2(5):203–209, December 1983.

[Whi84] Ward Whitt. Minimizing delays in the $GI/G/1$ queue. *Operations Research*, 32(1):41–51, 1984.

[Wol77] R. W. Wolff. The effect of service time regularity on system performance. In K. M. Chandy and M. Reiser, editors, *International Symposium on Computer Performance Modeling, Measurement, and Evaluation*, pages 297–304, Amsterdam, 1977.

[Wol89] R. W. Wolff. *Stochastic Modeling and the Theory of Queues*. Prentice Hall, Englewood Cliffs, NJ, 1989.