Re: Probability question in an M/M/2/4 queue

Source: http://sci.tech−archive.net/Archive/sci.math/2007−03/msg00054.html

* *From*: Marcaias <dnilbretniw@xxxxxxxxx>
* *Date*: Thu, 01 Mar 2007 06:02:48 GMT

On 28 Feb 2007 19:17:48 −0800, matt271829−news@xxxxxxxxxxx wrote:

> On Feb 28, 10:04 pm, matt271829−n...@xxxxxxxxxxx wrote:
>
> > On Feb 28, 7:06 pm, "C...@xxxxxxx" <C...@xxxxxxx> wrote:
> >
> > > > On Feb 27, 10:18 pm, Marcaias <dnilbret...@xxxxxxxxx>
> > > > wrote:
> > > >
> > > > > Consider the following scenario:
> > > > >
> > > > > A queue with two servers in which service
> > > > > times are exponentially
> > > > > distributed with mean 1/m. They're both
> > > > > currently servicing packets,
> > > > > and in addition there are two packets waiting
> > > > > in the queue (P_1 will
> > > > > be the first to be served, P_2 second.)
> > > > >
> > > > > What's the probability that P_2 will finish its
> > > > > service before P_1? I
> > > > > have an answer, but is my reasoning okay?
> > > > >
> > > > > There are two ways this can happen: (1)
> > > > > Server 1 can finish its job

and get P_2 followed by Server 2 finishing
its job and receiving P_2,
and then Server 2 can finish servicing P_2
before Server 1 finishes
with P_1. Or, (2) the same deal with Server
1 and 2 replaced.

Since both servers have exponentially
distributed service times with
the same mean, it's easy enough to show that
there's a 50/50 chance
that Server 1 will finish its original job
before Server 2 does, and
by the memoryless property of the
exponential distribution it seems to
me that the final answer should be (1/2)^3 +
(1/2)^3 = 1/4, but this
seems too easy.

Thanks,
Mark

At the moment P2 enters service there is a 50% chance that
P1 is still
in service (because there is a 50% chance the other customer
will
leave before P1 is finished). At this point there is a 50%
chance that
P2 leaves before P1, so the overall probability is (1/2)*(1/2)
= 1/4.

R.G. Vickson

I don't quite follow this, and in fact I suspect that a symmetry
argument that takes no note of the actual distributions does not work.
I worked through the problem again assuming a uniform distribution of
service times (any uniform distribution)

Clarification: that's any uniform distribution over [0,b], not over
[a,b].

> > , rather than exponential, and
> > I get the required probability equal to 13/45 which is patently not 1/4. I suspect that it may be just a rather remarkable coincidence that the probability comes out to exactly 1/4 for an exponential distribution.
> >
> > Another possibility (other than that I made an error) is that we are interpreting the question differently.
>
> Some more musings...
>
> This seems an interesting problem. Originally I did it like this:
>
> 1. Assume that the servers are continuously active processing incoming packets. (Each packet is processed by the first available server.)
>
> 2. Observe the situation at a random time, and find (of course) that both are active.
>
> 3. Calculate the probability that the second pending queued packet is finished processing before the first.
>
> I have now written a program to simulate this, and in random trials the calculated probabilities of 1/4 for exponentially distributed service times and 13/45 for uniformly distributed service times are observed to a reasonably convincing degree of accuracy.
>
> However, if I change the setup so that the servers are not continuously active, and packets arrive regularly on average more slowly than they are processed, and then I pick random observation times until I get a time when both servers are active, then I get a quite different answer, depending on the exact pattern of packet arrival. This is very obvious with the uniform distribution, but difficult to detect with the exponential.
>
> Anyway, based on this experiment, it's possible that information about distribution of packet arrival times is required to accurately answer the question.
>
> Or perhaps I am just making this vastly more complicated than it needs to be!

To clarify, both servers are continuously active. As soon as a server finishes processing a packet it receives another (assuming there's one waiting in the queue) and begins to process it.

The incoming flow was assumed to be Poisson−distributed (i.e. with exponentially distributed service times as well) but for the sake of this question that didn't matter.

The question was indeed supposed to be very simple, but thanks for the additional information! Gave me something to think about.

Thanks,
Mark
.