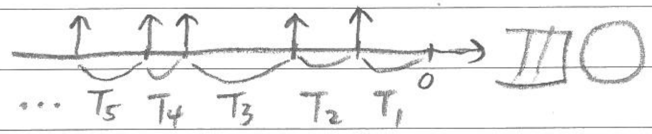
A proof that the interarrival times of the Poisson process are independent and exponentially distributed with rate λ:



$P\{T_1 > z\} = P\{A(z) = 0\}$ (by property ① in the definition of Poisson process)

$= P\{A(0+z) - A(0) = 0\}$

$= e^{-\lambda z} \cdot \frac{(\lambda z)^0}{0!}$ (by property ③ 〃)

$= e^{-\lambda z}$

Now, $P\{T_n > z \mid \sum_{k=1}^{n-1} T_k = s\}$ for some $s > 0$

$= P\{A(s+z) - A(s) = 0 \mid \sum_{k=1}^{n-1} T_k = s\}$

$= P\{A(s+z) - A(s) = 0\}$ (by property ② 〃)

$= e^{-\lambda z} \cdot \frac{(\lambda z)^0}{0!}$ (by property ③ 〃)

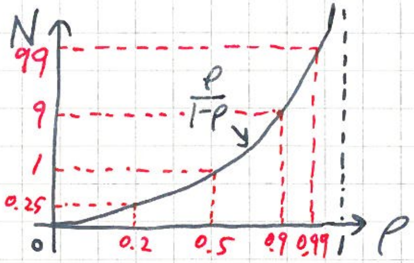
$= e^{-\lambda z}$

$\Rightarrow P\{T_n \leq z\} = 1 - e^{-\lambda z}$

$\Rightarrow f(T_n = z) = \lambda e^{-\lambda z}$, which is the p.d.f. of exponential distribution ※

In M/M/1, interpretation of $N = \frac{\rho}{1-\rho}$:

think of ρ as CPU utilization (CPU%)



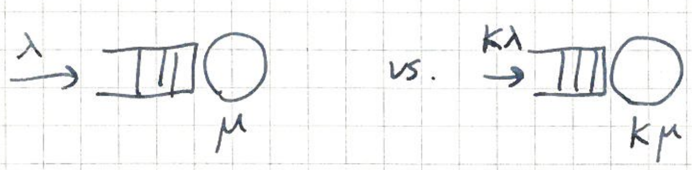
since in M/M/1, $\rho = \frac{\lambda}{\mu}$

→ $\rho > 1$: server cannot catch up with arrivals

→ $\rho < 1$: arrivals cannot feed the server fast enough

⇒ # of items in a queue (i.e., N) grows much faster than the growth of CPU utilization (i.e., ρ)!

Example 3.8 :



$\rho = \frac{\lambda}{\mu}$

$N = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu - \lambda}$

$T = N/\lambda$

$\rho = \frac{K\lambda}{K\mu} = \frac{\lambda}{\mu}$

$N = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu - \lambda}$

$T = N/K\lambda$

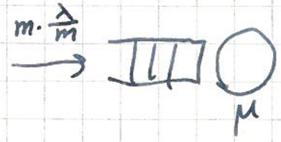
⇒ in both cases, a new arrival will see the same # of packets ahead of it statistically.

But ~~Though~~ the packets move K times faster ~~for~~ in the configuration on the right! ※

Example 3.9 : Statistical multiplexing vs. TDM/FDM

Consider m statistically identical and independent Poisson packet streams each with arrival rate $\frac{\lambda}{m}$. The packet length for all streams are independent and exponentially distributed. The average transmission time is $\frac{1}{\mu}$.

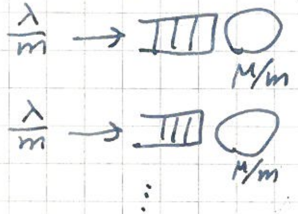
If merging all streams into one, it is like M/M/1:



and thus the average delay per packet is

$$T = \frac{1}{\mu - \lambda}.$$

If using TDM/FDM instead, it is like m M/M/1:



Then the average delay would be m times larger

$$T' = \frac{1}{\mu/m - \lambda/m} = \frac{m}{\mu - \lambda} = m \cdot T$$

⇒ why would anyone ever want to use TDM/FDM ?

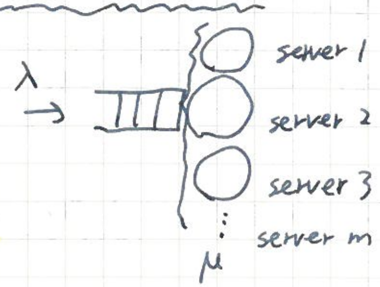
→ TDM/FDM guarantees a specific service rate for each stream. Plus, statistical multiplexing may introduce a higher variability in latency, which is undesirable for applications such as voice/video streaming.

The PASTA theorem: Poisson Arrivals See Time Average

(read Sections 3.3.2–3.3.3)

→ "When arrivals are Poisson, [...] both an arriving and a departing customer in steady-state see a system that is statistically identical to the one seen by an observer looking at the system at an arbitrary time." We may use that to analyze the probability that an arrival will need to wait in queue.

M/M/m

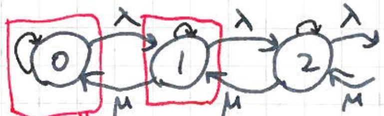


let μ be the service rate,

$\rho = \frac{\lambda}{m\mu}$ the utilization.

To analyze M/M/m, it is helpful to take a closer look at how we got $p_n\lambda = p_{n+1}\mu$ in M/M/1:

recall the Markov chain in M/M/1



$\sum$(probability of entering a state) = $\sum$(probability of leaving the state)

$p_0\lambda = p_1\mu$

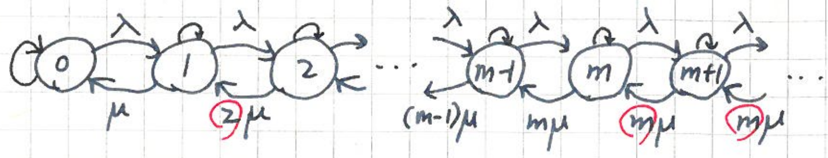
$p_1\lambda + p_1\mu = p_0\lambda + p_2\mu$

...

$p_n\lambda + p_n\mu = p_{n-1}\lambda + p_{n+1}\mu$

$\Rightarrow p_n\lambda = p_{n+1}\mu$

On M/M/m, we have the following Markov chain



for $n \le m-1$,

$p_0\lambda = p_1\mu$

$p_1\lambda + \cancel{p_0\mu} = \cancel{p_0\lambda} + p_2\cdot 2\mu$

$p_2\lambda + \cancel{p_1 2\mu} = \cancel{p_1\lambda} + p_3\cdot 3\mu$

...

$\Rightarrow p_n\cdot\lambda = p_{n+1}\cdot(n+1)\mu$

$\Rightarrow p_m\cdot m\mu = p_{m-1}\cdot\lambda$

$p_m = \frac{\lambda}{m\mu}p_{m-1}$

$\Rightarrow p_n = \frac{\lambda^n}{n!\,\mu^n}p_0$

from $\boxed{\rho = \frac{\lambda}{m\mu}}$

we see that $\left(\frac{\lambda}{\mu}\right)^n = (m\rho)^n$

therefore $p_n = \frac{(m\rho)^n}{n!}p_0$

for $n \ge m$,

$p_m\cdot\lambda + \cancel{p_m(m\mu)} = \cancel{p_{m-1}\lambda} + p_{m+1}(m\mu)$

$p_{m+1}\lambda + \cancel{p_{m+1}(m\mu)} = \cancel{p_m\lambda} + p_{m+2}(m\mu)$

...

$\Rightarrow p_n\cdot\lambda = p_{n+1}(m\mu)$.

similarly, for $n \ge m$ we have

$p_n = \underbrace{\frac{\lambda}{m\mu}}_{\rho} p_{n-1} = \rho^{n-m}\frac{(m\rho)^m}{m!}p_0$

$= \frac{(m\rho)^n}{m!\,m^{n-m}}p_0 = \frac{m^m\rho^n}{m!}p_0$

$= \frac{\rho^n\cdot m^n}{m!\cdot m^{n-m}}p_0$

$\Rightarrow p_n = \frac{m^m\rho^n}{m!}p_0$

Then p_0 can be obtained by $\sum_{n=0}^{\infty} p_n = 1$.

Let P_Q be the probability that an arrival will need to wait in queue (because all m servers are busy), and we have

$$P_Q = \sum_{n=m}^{\infty} p_n = \sum_{n=m}^{\infty} p_0\frac{m^m\rho^n}{m!} = p_0\frac{m^m\rho^m}{m!}\sum_{n=m}^{\infty}\rho^{n-m}$$

$$\Rightarrow \boxed{P_Q = \frac{(m\rho)^m p_0}{m!\,(1-\rho)}}$$

← The Erlang C formula, named after the pioneer of Queueing Theory, A. K. Erlang.

Now let N_Q be the expected # of customers waiting in queue. We may obtain it by $\sum_{n=0}^{\infty} n \cdot P_{m+n}$.
Alternatively, we may consider M/M/m's relation to M/M/1, and get $N_Q = P_Q \cdot \underbrace{\frac{\rho}{1-\rho}}_{\text{from M/M/1}}$

Finally, $N = N_Q + N_S$ where N_S is # of customer in service

from the linearity of expected #.

$N_S = \sum_{n=1}^{m} n \cdot P_n = \ldots\ldots$

we can use Little's Theorem here, and $N_S = \lambda \cdot \frac{1}{\mu} = m\rho$

$$\Rightarrow \boxed{N = m\rho + \frac{\rho P_Q}{1-\rho}}$$

$$\Rightarrow \boxed{T = N/\lambda = \frac{1}{\mu} + \frac{P_Q}{m\mu-\lambda}} \leftarrow \text{using Little's Theorem}$$

Detail derivation: let X be a random variable for the number of customers in the queue.

$$N_Q = P_Q \cdot E[X \mid \text{queueing}] + P\{\text{no queueing}\} \cdot E[X \mid \text{no queueing}]$$
$$= P_Q \cdot E[X \mid \text{queueing}]$$

To get $E[X \mid \text{queueing}]$, we note that

$E[X \mid \text{queueing}]$ in M/M/m with service rate λ

is equal to $E[X \mid \text{queueing}]$ in M/M/1 with service rate $m\lambda$

let N_{Q_1} be the expected # of customers waiting in this M/M/1 system, then $N_{Q_1} = \frac{\rho}{1-\rho} - \rho = \frac{\rho^2}{1-\rho}$

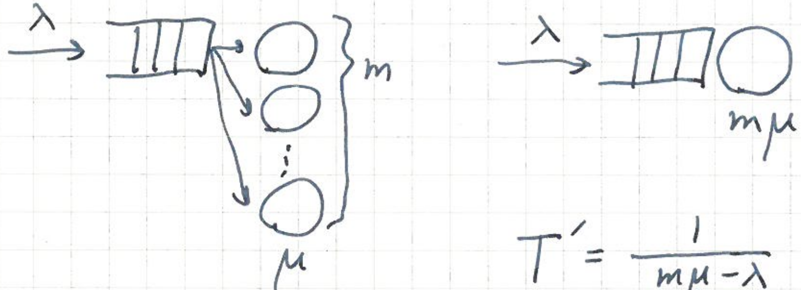
From $N_{Q_1} = P_{Q_1} \cdot E[X \mid \text{queueing}]$, we have

$E[X \mid \text{queueing}] = N_{Q_1}/P_{Q_1} = \frac{\rho^2}{1-\rho}/\rho = \frac{\rho}{1-\rho}$

and therefore $N_Q = P_Q \cdot E[X \mid \text{queueing}] = P_Q \cdot \frac{\rho}{1-\rho}$ ※

Example 3.10 (compared with examples 3.8 and 3.9)

M/M/m v.s. M/M/1



$$T' = \frac{1}{m\mu - \lambda}$$

$$\rho = \frac{\lambda}{m\mu}$$

$$T = \frac{1}{\mu} + \frac{P_Q}{m\mu-\lambda}$$

$\Rightarrow$ when $\rho \ll 1$,

$P_Q \approx 0$ and $m\mu \gg \lambda$, which imply

$$\frac{T}{T'} \approx m$$

$\Rightarrow$ when $\rho \to 1$,

$P_Q \approx 1$ and $\frac{1}{\mu} \ll \frac{1}{m\mu-\lambda}$, which imply

$$\frac{T}{T'} \approx 1$$

The above result suggests that using one faster channel (i.e., server) for statistical multiplexing will lead to a better performance in latency, compared with one using multiple slower channels (i.e., servers)