Data Communication Networks

Lectures 6 - 7

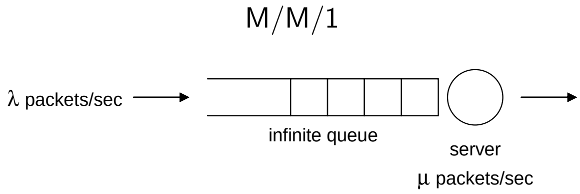
Saad Mneimneh
Computer Science
Hunter College of CUNY
New York

A simple queuing system M/M/1 . . . 2
Modeling M/M/1 . . . 3
Steady state . . . 4
A little theory . . . 5
Back to M/M/1 . . . 6
Back to M/M/1 . . . 7
Expected number of customers . . . 8
Expected delay per customer . . . 9
Expected waiting time (in queue) . . . 10
Throughput . . . 11
Delay . . . 12
Power . . . 13
Numerical example . . . 14
Packet switching vs. circuit switching . . . 15
M/M/1 with finite queue . . . 16
Computing p_n . . . 17
Computing N . . . 18
Throughput . . . 19
Delay . . . 20
Throughput, Delay, Power . . . 21
Congestion control . . . 22
A network of queues . . . 23
Kleinrock independence approximation . . . 24
M/G/1 . . . 25
Computing W . . . 26
Computing R . . . 27
Pollaczek-Khinchin (P-K) formula . . . 28
Example . . . 29
Example: sliding window . . . 30
M/G/1 with vacations . . . 31
P-K formula with vacations . . . 32
Example: slotted FDM . . . 33
Fairness . . . 34

Example . . . 35
Max-Min Fairness . . . 36
An algorithm . . . 37
Bottleneck . . . 38
Max-Min fair and bottlenecks . . . 39
Another characterization of Max-Min fairness . . . 40
Fairness index . . . 41

A simple queuing system M/M/1

- The first letter indicates the nature of the arrival process
  - M for Memoryless, i.e. Poisson process (exponentially distributed interarrival times)
  - G for general distribution of interarrival times
  - D for deterministic interarrival times
- The second letter indicates the nature of the probability distribution of the service times (e.g. M, G, or D as above)
- The last number indicates the number of servers



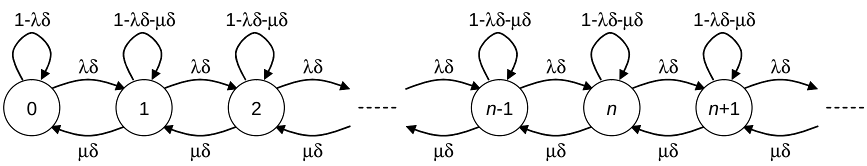
Modeling M/M/1

- Using the third form of a Poisson process, we can model the M/M/1 queuing system as a Markov chain, where state k indicates that the number of customers in the system is k
- Let us focus our attention at times $0, \delta, 2\delta, \ldots, k\delta, \ldots$

$$Pr(0\ arrivals,\ 0\ departures) \approx (1-\lambda\delta)(1-\mu\delta) \approx 1-\lambda\delta-\mu\delta$$

$$Pr(1\ arrival,\ 0\ departures) \approx \lambda\delta(1-\mu\delta) \approx \lambda\delta$$

$$Pr(0\ arrivals,\ 1\ departure) \approx (1-\lambda\delta)\mu\delta \approx \mu\delta$$



Steady state

- Let N_k be the number of customers at time $t = k\delta$
- We would like to find N, the expected number of customers in the system at steady state

$$N = \sum_{n=0}^{\infty} p_n n$$

where p_n is the steady state probability of being in state n

$$p_n = \lim_{k \to \infty} Pr(N_k = n)$$

- But, is there a steady state?

A little theory

- A Markov chain is *irreducible* iff the directed graph formed is connected, i.e. given any two states $i \neq j$, there is a path from i to j
- A state i in a Markov chain is *periodic* iff there is a path from i to i and the length of every such path is a multiple of some integer $d > 1$ (d is said to be the period)
- A Markov chain is *aperiodic* iff none of its states is periodic periodic

Given an irreducible and aperiodic Markov chain, let p_i be the probability of being in state i at steady state, then

- $p_i = 0$ for all i, in which case the chain has no steady state distribution
- $p_i > 0$ for all i, in which case this is the unique stationary distribution of the chain, and

$$p_i = \sum_j p_j P_{ij} \quad (why?)$$

where P_{ij} is the transition probability from state i to state j

Back to M/M/1

- The Markov chain for M/M/1 is irreducible and aperiodic
- Steady state equation for p_0

$$p_0 = p_0(1 - \lambda\delta) + p_1\mu\delta$$

- Steady state equation for p_n, $n > 0$

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

- Taking $\lim_{\delta\to 0}$ we get

$$\frac{p_n}{p_{n-1}} = \frac{\lambda}{\mu} = \rho \ \ \forall n > 0$$

- Since $\sum_n p_n = 1$ and $p_n = \rho^n p_0$ we have

$$p_0 \sum_{n=0}^{\infty} \rho^n = 1$$

Back to M/M/1

$$p_0 \sum_{n=0}^{\infty} \rho^n = 1$$

- Note that the above sum converges only for $\rho < 1$, i.e. $\lambda < \mu$
  - if $\rho = 1$, all states are equally likely and hence $p_n = 0 \ \ \forall n$ (no steady state distribution)
  - if $\rho > 1$, further states are more likely (no steady state distribution since the queue is infinite)
- when $\rho < 1$, $\sum_{n=0}^{\infty} \rho^n = \frac{1}{1-\rho}$
- $p_0 = 1 - \rho$, $p_n = \rho^n(1 - \rho)$
- Interpretation of ρ
  - since p_0 is the probability that the system is empty, ρ is the probability that the server is busy
  - ρ can be thought of as utilization, efficeincy, or throughput, i.e. expected number of customers getting service (note $\rho \leq 1$ always)
  - The last interpretation is consistent with Little's theorem: $\rho = \lambda.\frac{1}{\mu}$

Expected number of customers

$$
\begin{aligned}
N &= \textstyle\sum_{n=0}^{\infty} p_n n \\
&= \rho(1-\rho)\textstyle\sum_{n=0}^{\infty} n\rho^{n-1} \\
&= \rho(1-\rho)\tfrac{\partial}{\partial\rho}\textstyle\sum_{n=0}^{\infty} \rho^{n} \\
&= \rho(1-\rho)\tfrac{\partial}{\partial\rho}\tfrac{1}{1-\rho} \\
&= \tfrac{\rho}{1-\rho} = \tfrac{\lambda}{\mu-\lambda}
\end{aligned}
$$

Expected delay per customer

$$N = \lambda T \quad (Little's\ theorem)$$

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

Expected waiting time (in queue)

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

Note that from Little's theorem we get the expected number of customers in queue

$$N_Q = \lambda W = \frac{\rho^2}{1 - \rho}$$

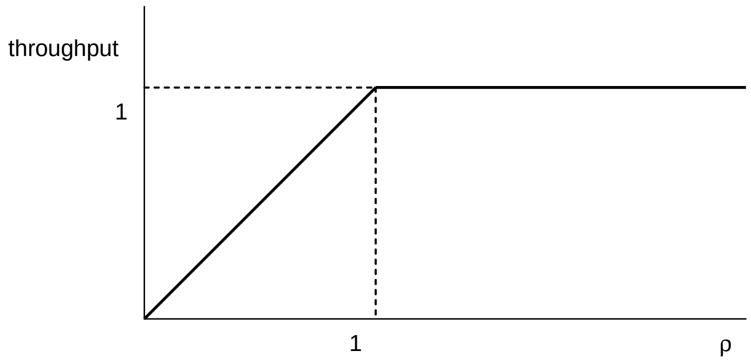
Verify:

$$N_Q + \rho = N$$

Throughput

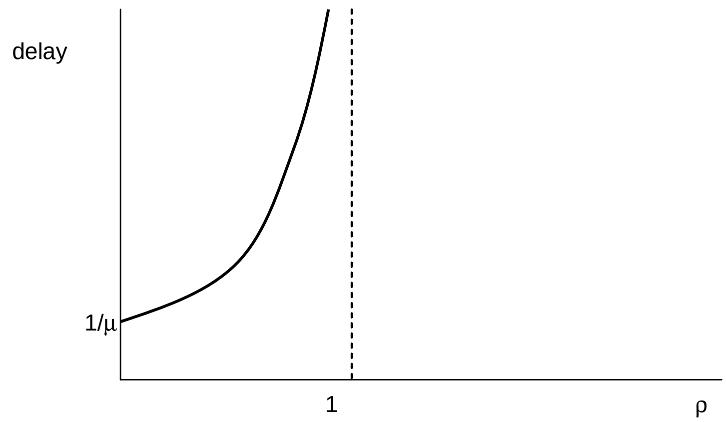
- We interpreted ρ as being the throughput of the system
- If $\rho \geq 1$, no steady state solution, but server is always busy
- Throughput is

$$\min(1, \rho)$$



Delay

$$T = \frac{1/\mu}{1-\rho}$$

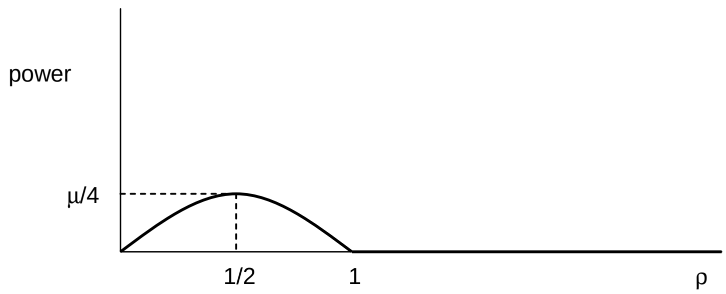


Power

- The best operating point of the system is when throughput is high and delay is low
- Define the power as

$$power = \frac{throughput}{delay} = \lambda(1 - \lambda/\mu)$$

- Power is maximized when $\lambda = \mu/2$ $(\rho = 1/2)$



Numerical example

Conisder a fast food restaurant with Poisson arrivals at a rate of 100 customers per hour. The service time is exponentially distributed with an average of 30 seconds.

- $\lambda = 100$
- $\mu = 1/0.5 = 2$ customers/min = 120 customers/hour
- A customers spends on average $T = \frac{1}{\mu - \lambda} = \frac{1}{20}$ hours = 3 min until completely served
- A customer waits in line on average $W = T - 1/\mu = 3 - 0.5 = 2.5$ min
- The average number of customers in the restaurant at any time is $\lambda T = 100\frac{1}{20} = 5$
- The throughput (servant utilization) is $\rho = \lambda/\mu = 5/6$

Packet switching vs. circuit switching

Consider m sessions each with a Poisson arrival rate of λ/m. Packet sizes are exponentially distributed with an average of L bits. The line has a bandwidth of μ bps.

- The transit time is $\frac{packet\ size}{\mu}$, thus transit times are exponentially distributed with an average of L/μ
- Packet switching (Poisson processes of m sessions are merged)

$$T = \frac{1}{\mu/L - \lambda}$$

$$N = \lambda T = \frac{\lambda}{\mu/L - \lambda}$$

- Circuit switching (each session is given $1/m$ of link bandwidth)

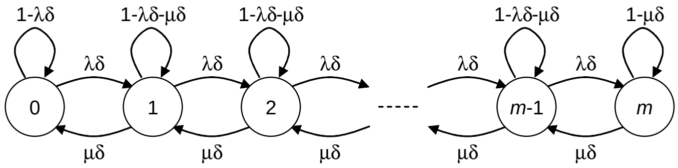
$$T = \frac{1}{\mu/mL - \lambda/m} = \frac{m}{\mu/L - \lambda}$$

$$N = (\lambda/m)T = \frac{\lambda}{\mu/L - \lambda} \quad (per\ session)$$

- Delay and number of packets are both multiplied by m in circuit switching

M/M/1 with finite queue

- The queue has length m
- If a packet arrives while the queue is full, it is dropped
- We have a finite Markov chain as follows



- We need to find p_n at steady state for $n = 0 \dots m$, then we find $N = \sum_{n=0}^{m} p_n n$
- Steady state equations:

$$p_0 = p_0(1 - \lambda\delta) + p_1 \mu\delta$$

$$p_m = p_{m-1}\mu\delta + p_m(1 - \mu\delta)$$

$$p_n = p_{n-1}\lambda\delta + p_n(1 - \lambda\delta - \mu\delta) + p_{n+1}\mu\delta \quad \forall n \neq 0, m$$

Computing p_n

From steady state equations we get:

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

$$\sum_{n=0}^{m} p_n = 1 \Rightarrow \sum_{n=0}^{m} \rho^n p_0 = 1 \Rightarrow p_0 = \frac{1}{\sum_{n=0}^{m} \rho^n} = \frac{\rho - 1}{\rho^{m+1} - 1}$$

$$(\rho \neq 1 \Rightarrow \sum_{n=0}^{m} \rho^n = \frac{\rho^{m+1} - 1}{\rho - 1})$$

$$p_m = \rho^m p_0 = \frac{\rho^{m+1} - \rho^m}{\rho^{m+1} - 1}$$

- $\rho = 1$, $\sum_{n=0}^{m} \rho^n = m + 1$: $p_n = \frac{1}{m+1} \quad \forall n$ (all states are equally likely)
- p_m is the probability of dropping the packet
- $m \to \infty$, $\rho < 1$: $p_m \to 0$ (never drop, inifite queue)
- $\rho \to \infty$: $p_m \to 1$ (always drop)

Computing N

$$N = \sum_{n=0}^{m} p_n n = \sum_{n=0}^{m} \rho^n p_0 n = \frac{\rho - 1}{\rho^{m+1} - 1} \rho \sum_{n=0}^{m} n \rho^{n-1}$$

$$N = \frac{\rho - 1}{\rho^{m+1} - 1} \rho \frac{\partial}{\partial \rho} \sum_{n=0}^{m} \rho^n$$

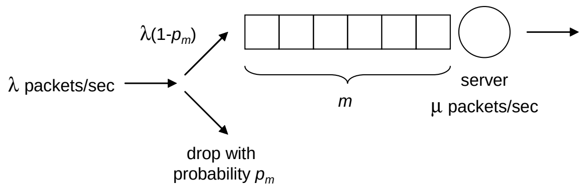
$$N = \frac{(m+1)\rho^{m+1}}{\rho^{m+1} - 1} + \frac{\rho}{1 - \rho}$$

- $\rho = 1$, $p_n = \frac{1}{m+1}$ $\forall n$:

$$N = \sum_{n=0}^{m} \frac{1}{m+1} n = \frac{m}{2}$$

- $m \to \infty$, $\rho < 1$: $N \to \frac{\rho}{1-\rho}$ (as before)
- $\rho \to \infty$: $N \to m + 1 - 1 = m$ (queue always full)

Throughput



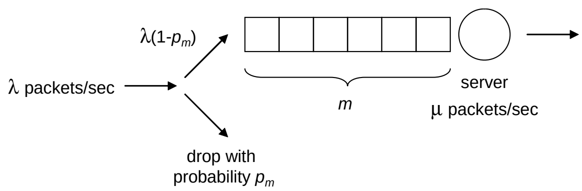
- The server sees an arrival process (Poisson by the splitting argument) with rate $\lambda(1 - p_m)$
- The throughput of the server is $\lambda(1 - p_m)/\mu = \rho(1 - p_m)$ by Little's theorem
- The throughput at the front of the queue is $\lambda(1 - p_m)/\lambda = (1 - p_m)$
- The throughput of the system is given by

$$\min(\rho(1 - p_m), 1 - p_m)$$

- Compare this to the previous case where $p_m = 0$ and the throughput is given by

$$\min(\rho, 1)$$

Delay



By Little's theorem

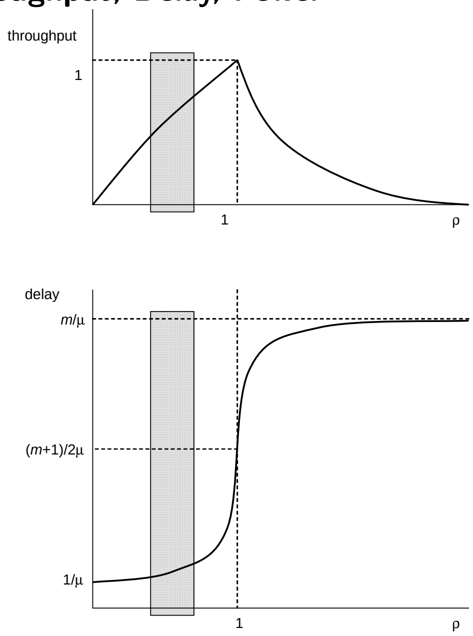
$$T = \frac{N}{\lambda(1-p_m)}$$

We obtain

$$T = \frac{1}{\mu}\left[\frac{(m+1)(1-\rho)\rho^m + \rho^{m+1} - 1}{(\rho^m - 1)(1-\rho)}\right]$$

- $\rho = 1$, $p_m = \frac{1}{m+1}$, $N = \frac{m}{2}$: $T = \frac{m+1}{2\mu}$
- $m \to \infty$, $\rho < 1$: $T = \frac{1}{\mu-\lambda}$ (as before)
- $\rho \to \infty$: $T = \frac{m}{\mu}$

Throughput, Delay, Power



Dashed region shows desired region for power=throughput/delay

Congestion control

To summarize what we have seen so far:

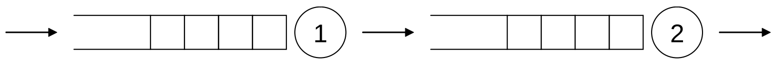
- Throughput increases with ρ until it reaches a maximum and then starts decreasing rapidly (this is when we have considerably high droping probability)
- Delay start increasing considerably when throughput approaches 1
- Best operating point is to keep throughput around 50%

Therefore, a possible congestion control algorithm would be the following:

- Upon detection of a loss (a sign that we have approached a throughput close to 1), decrease the rate (reduce window size) by a factor of 2
- Increase rate slowly until detecting another loss

A network of queues

- Consider two queues in tandem



- Packets arrive according to a Poisson process
- Packet lengths are exponentially distributed
- The first queue can be modelled as M/M/1
- The second queue, however, cannot!
  - interarrival times at the second queue are strongly correlated with the packet lengths
  - in particular, the interarrival time of two packets at the second queue is greater than or equal to the transmission time of the second packet at the first queue
  - long packets will typically wait less time at the second queue

Kleinrock independence approximation

- If the second transmission line in the preceeding tandem queue case were to receive a substantial amount of additional external Poisson traffic, the dependence of interarrival times and service times would be weakened
- It is often appropriate to adopt M/M/1 model for each link when
  - Poisson arrivals at entry points
  - packet lengths nearly exponentially distributed
  - densely connected network
  - moderate to heavy traffic loads
- Therefore, given that link (i, j) has a total rate λ_{ij}

$$N_{ij} = \frac{\lambda_{ij}}{\mu_{ij} - \lambda_{ij}}$$

The average delay per packet (Little's theorem) is $T = \frac{\sum_{i,j} N_{ij}}{\sum_{i,j} \lambda_{ij}}$ (ignoring propagation delay)

M/G/1

- Service times have a general (G) distribution, not necessarily exponential as in M/M/1
- Customers (packets) are served in the order they arrive, i.e. FIFO
- X_i = service time of customer i (assume X_i are identically distributed and mutually independent)

$$\overline{X} = E[X] = \frac{1}{\mu} = average\ service\ time$$

$$\overline{X^2} = E[X^2]$$

- W_i = waiting time in queue for customer i
- N_i = number of customers seen in queue by customer i upon arrival
- R_i = residual service time seen by customer i, i.e. the time needed for current customer in service to finish service

Then (FIFO),

$$W_i = R_i + \sum_{j=i-N_i}^{i-1} X_j$$

Computing W

$$W_i = R_i + \sum_{j=i-N_i}^{i-1} X_j$$

$$E[W_i] = E[R_i] + E[N_i]\overline{X} \quad \text{(assuming } X \text{ independent of } N\text{)}$$

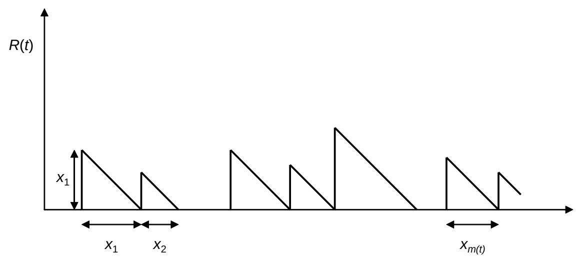
By PASTA property (and assuming steady state exists)

$$W = R + N_Q \frac{1}{\mu}$$

But by Little's theorem, $N_Q = \lambda W$, so

$$W = R + \rho W \Rightarrow W = \frac{R}{1-\rho}$$

Computing R



$$R_t = \frac{1}{t}\sum_{i=1}^{m(t)} \frac{1}{2}X_i^2 = \frac{m(t)}{t}\frac{\sum_{i=1}^{m(t)} \frac{1}{2}X_i^2}{m(t)}$$

Taking the limit as $t \to \infty$

$$R = \frac{1}{2}\lambda\overline{X^2}$$

Pollaczek-Khinchin (P-K) formula

$$W = \frac{\lambda \overline{X^2}}{2(1-\rho)}$$

$$T = \overline{X} + W$$

- M/M/1: $\overline{X^2} = 2/\mu^2$, $W = \frac{\rho}{\mu(1-\rho)}$
- M/D/1: (deterministic), $\overline{X^2} = 1\mu^2$, $W = \frac{\rho}{2\mu(1-\rho)}$

Althoug we assumed FIFO, the P-K formula is valid for any order of sevice as long as the order is independent of the service times of individual customers (e.g. serving smaller jobs first or larger jobs first would make the order dependent on service times)

Example

Consider an M/G/1 system:

- What is the probability that the system is empty?
  - By Little's theorem, the average number of customers in service is $\lambda\overline{X}$
  - $P[empty] = 1 - \lambda\overline{X}$
- What is the average time I (for idle) between busy periods?
  - consider the end of a busy period, since arrivals are Poisson (memoryless), the next arrival is exponentially distributed (which signals the beginning of next busy period)
  - $I = 1/\lambda$
- What is the average time B of a busy period?
  - $\frac{B}{B+I} = \lambda\overline{X}$
  - $B = \frac{\overline{X}}{1-\lambda\overline{X}}$
- What is the average number of customers served in a busy period
  - from above, $\frac{1}{1-\lambda\overline{X}}$

Example: sliding window

- Assume Go back n with one sided error
  - probability of error is p
  - ACKs always arrive
  - timeout $= n$ (for retransmissions)
- When packet i is successfully transmitted, packet $i+1$ is successfully transmitted $1+kn$ time units later with probability $(1-p)p^k$
- Transmitter's queue behaves like M/G/1
  - $Pr[X = 1 + kn] = (1-p)p^k$
  - $\overline{X} = 1 + \frac{np}{1-p}$, $\overline{X^2} = 1 + \frac{2np}{1-p} + \frac{n^2(p+p^2)}{(1-p)^2}$ (after some calculation)
  - A packet waits on average $W = \frac{\lambda \overline{X^2}}{2(1-\lambda \overline{X})}$
- What if general sliding window, i.e. receiver has a window size m (ACKs are not FIFO now)
  - same average result

M/G/1 with vacations

- Suppose that at the end of each busy period, the server goes on vacation for some random interval of time
- If the system is still idle at the completion of a vacation, a new vacation start immediately
- Let V_1, V_2, ..., $V_{l(t)}$ be the durations of vacations at up to time t
- Each customer sees at most one vacation
- The following formula is still valid

$$W = \frac{R}{(1-\rho)}$$

  where R is now the residual time for completion of the service or vacation in process when the customer arrives
- Using a similar graphical argument

$$R_t = \frac{1}{t}\sum_{i=1}^{m(t)} \frac{1}{2}X_i^2 + \frac{1}{t}\sum_{i=1}^{l(t)} \frac{1}{2}V_i^2$$

P-K formula with vacations

$$R_t = \frac{m(t)}{t}\frac{\sum_{i=1}^{m(t)} \frac{1}{2}X_i^2}{m(t)} + \frac{l(t)}{t}\frac{\sum_{i=1}^{l(t)} \frac{1}{2}V_i^2}{l(t)}$$

$$R = \frac{1}{2}\lambda\overline{X^2} + \frac{1}{2}\overline{V^2}\lim_{t\to\infty}\frac{l(t)}{t}$$

But

$$\lim_{t\to\infty}\frac{t(1-\rho)}{l(t)} = \overline{V}$$

$$R = \frac{1}{2}\lambda\overline{X^2} + \frac{1}{2}\frac{(1-\rho)\overline{V^2}}{\overline{V}}$$

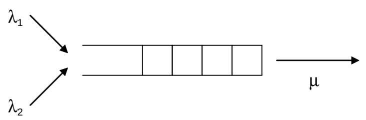
$$W = \frac{R}{1-\rho} = \frac{\lambda\overline{X^2}}{2(1-\rho)} + \frac{\overline{V^2}}{2\overline{V}}$$

Example: slotted FDM

- Consider m sessions, each a Poisson process with rate λ/m, frequency division multiplexed on a channel
- Transmission time per packet is m time units on a subchannel
- M/D/1 $\Rightarrow W = \frac{\lambda m}{2(1-\lambda)}$ $(\overline{X} = 1/\mu = m)$
- If the system is slotted, i.e. packets can only leave at times m, $2m$, $3m$, etc..., then we can view this as M/D/1 with vacation
- If no packet waiting, server takes a vacation of m units, $\overline{V} = m$, $\overline{V^2} = m^2$
- M/D/1 with vacations $\Rightarrow W = \frac{\lambda m}{2(1-\lambda)} + \frac{m}{2} = \frac{m}{2(1-\lambda)}$, $T = \frac{m}{2(1-\lambda)} + m$

Fairness

- Throughput or efficiency is a very important property of a network (delay is another of course)
- If multiple flows are sharing the network, one could achieve high throughput by making one flow send enough and data preventing others from sending
- Thereofre, we need another property, called Fairness, to ensure that all users receive an equal share of the network resources

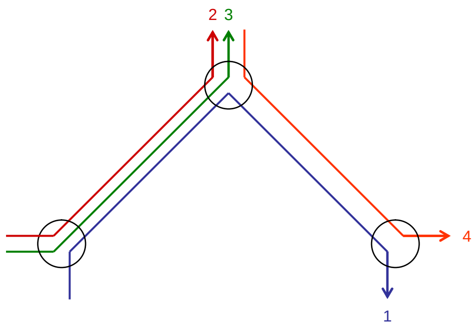


  - network is efficient $\Rightarrow \lambda_1 + \lambda_2 = \mu$ (or close)
  - network is fair $\Rightarrow \lambda_1 = \lambda_2$

- But a network is not just one link, the notion of "equal share" is not necessarily what we think of equal, in particular, how do we handle flows that use different paths?

Example

- Assume there is no explicit demand or reservation of bandwidth
- Assume all links have capacity 1 (unit bandwidth)



- It makes sense to limit the rates of flows 1, 2, and 3 to 1/3 each
- But it is pointless to do the same for flow 4
- Flow 4 may have a rate of 2/3
  - if less, no one else benefits
  - if more, flow 1 must decrease its rate to <1/3 (unfair)
- So fair does not necessarily mean "equal", but what does it mean?

Max-Min Fairness

- Let f denote a flow, l denote a link, and

$$F_l = \sum_{f\ crosses\ l} r_f$$

  then we have the following constraints:

$$r_f \geq 0 \quad \forall \ f$$

$$F_l \leq c_l \quad \forall \ l$$

  where r_f is the rate of flow f, and c_l is the capacity of link l

Max-Min Fairness

- Increase the rates of all flows simultaneously by the same amount until one or more link saturate ($F_l = c_l$)
- Freeze all flow passing through the saturated links
- Repeat with the remaining set of flows

An algorithm

How to simultaneously increase all rates until a link saturates?

- Find the smallest ϵ such that when $r_f \leftarrow r_f + \epsilon \quad \forall \ f$, a link will saturate
- This ϵ is given by:

$$\min_l \frac{c_l - F_l}{n_l}$$

  where n_l is the number of flows crossing link l

$k = 1 \ \ F_l^0 = 0 \ \ r_f^0 = 0 \ \ F^1 = all\ flows \ \ L^1 = all\ links$

repeat

  $n_l^k \leftarrow$ # of flows $f \in F^k$ crossing link l

  $\epsilon^k \leftarrow \min_{l \in L^k}(c_l - F_l^{k-1})/n_l^k$

  if $f \in F^k$

   then $r_f^k \leftarrow r_f^{k-1} + \epsilon^k$

   else $r_f^k \leftarrow r_f^{k-1}$

  $F_l^k \leftarrow \sum_{f\ crossing\ l} r_f^k$

  $L^{k+1} \leftarrow \{l | F_l^k < c_l\}$

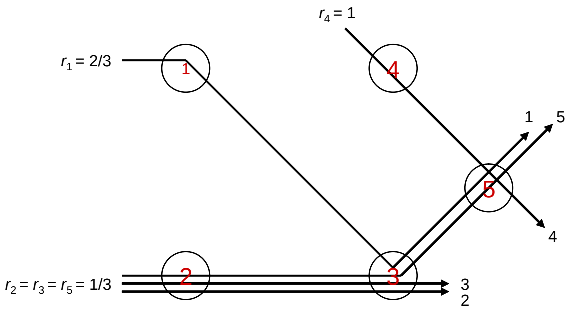
  $F^{k+1} \leftarrow F^k - \{f | f\ crosses\ a\ link\ \notin L^{k+1}\}$

  $k \leftarrow k + 1$

until $F^k = \emptyset$

Bottleneck

- Define a Link l to be a bottleneck for flow f iff:
  - f crosses l
  - $c_l = F_l$
  - all flows f' crossing l satisfy $r'_f \leq r_f$
- From the definition of a bottleneck, if f and f' have a common bottleneck, then $r_f = r'_f$
- Example (assume all links have unit capacity):



flow	bottleneck
1	(3,5)
2	(2,3)
3	(2,3)
4	(4,5)
5	(2,3)

Max-Min fair and bottlenecks

- The set of links that saturate in the k^{th} iteration is $L^k - L^{k+1}$:

$$l \in L^k - L^{k+1} \Rightarrow F_l^k = c_l$$

- The set of flows frozen in the k^{th} iteration is $F^k - F^{k+1}$; moreover,

$$f, f' \in F^k - F^{k+1} \Rightarrow r_f = r_{f'}$$

- Therefore, if f crosses l, $l \in L^k - L^{k+1}$, and $f \in F^k - F^{k+1}$, then l is a bottleneck for f
- Max-Min fair $\Rightarrow$ every flow has a bottleneck
- Every flow has a bottleneck $\Rightarrow$ Max-Min fair ? Yes, given a the set of bottlenecks, the Max-Min fair algorithm must produce exactly the same set

Another characterization of Max-Min fairness

(1) For each flow f, r_f cannot be increased without decreasing $r_{f'}$ for some flow f' where $r_{f'} \leq r_f$ (f' could be the same as f).

(2) Every flow has a bottleneck.

- $(1) \Rightarrow (2)$: Assume we increase r_f for some f. Since f has a bottleneck l, $F_l = c_l$ (l is saturated) and all flows f' going through l satisfy $r_{f'} \leq r_f$. Therefore, some flow f' with rate $r'_f \leq r_f$ must decrease.
- $(2) \Rightarrow (1)$: Assume some flow f does not have a bottleneck. Therefore, for every link l that f crosses, either $F_l < c_l$ or there exists a flow f' crossing l with $r_{f'} > r_f$ and $F_l = c_l$. As a result, we can increase r_f by only decreasing rates for flows f' such that $r_{f'} > r_f$, a contradiction.

Fairness index

- We expect n flows sharing a common bottleneck to receive the same rates
- But what if they don't? How do we measure fairness?
- Fainess index

$$F(r) = \frac{(\sum_i r_i)^2}{n \sum_i r_i^2}$$

- This index has nice properties
  - $0 < F(r) \leq 1$:
    - totally fair: all r_i's are equal: $F(r) = 1$
    - totally unfair: only one user is given the resource: $F(r) = 1/n$ (which goes to zero when $n \rightarrow \infty$)
  - independent of scale: the unit of measurment is irrelevant, i.e. multiplying all rates by the same constant keeps the index unchanged
  - continuous function: any slight change in allocation shows up
  - if only k users share the resource equally, $F(r) = k/n$