⇒ Probability that output port ~~is~~ will transmit in a time slot $= 1-(1-P/N)^N$

- Utilization of port $i = 1-(1-P/N)^N$
- Utilization of switch $= N(1-(1-P/N)^N)$

⇒ Average link utilization $= 1-(1-P/N)^N$

ex: $N=1$, $P=1$, avg switch utilization $=1$

$N=2$, $P=1$, avg switch utilization $=0.75$

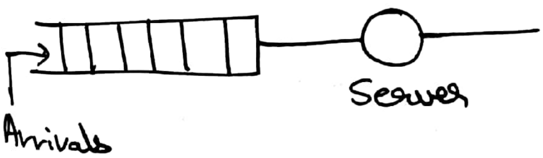
$N=4$, $P=1$, avg switch utilization $= 1-(0.75)^4$
$= 0.68$

$N \to \infty$, $\lim_{N\to\infty} 1-(1-P/N)^N = 1-e^{-1} = 0.63$

(This is better than input port buffering, so input port buffering is rarely used)

---

* Queueing Theory:



- Components of a queue
  - → Arrival process (Random, bursts etc)
  - → Service time distribution
  - → # of servers
  - → Waiting position
  - → Service discipline (FIFO or random etc)

* Kendall Notation:

Arrival Process | Service Time | # Servers | Capacity of system

⇒ Most common queue ⇒ M/M/1/∞ → M/M/1/∞/∞/FCFS

(M = Markov process; the last ∞ = Number of jobs server can handle)

(M/M/1 is a short form)

- Markov process → Follows a poisson distribution (Memoryless in nature)

* Markov Process:

- Memoryless in nature
- * Follows a poisson distribution (Arrivals) if inter arr... is exp...
- If arrival times are $t_1, t_2, \dots, t_n$
- Inter arrival times : $\underbrace{T_i = t_i - t_{i-1}}_{\text{These are independent and iid variables}}$

(Exponential iid inter arrival distribution $\Rightarrow$ Poisson arrival distribution)

---

* Notation:

1) M = Memoryless (Markovian) | Poisson } Independent of past.
   ex: M|M|1

2) E_k = Erlang
3) H_k = Hyper-exponential
4) G = General
   ex: M|G|1|∞

Inter arrival distribution:

Poisson

- X = random. variable denoting # of arrivals in T

* Poisson process with mean = λ

$$P(X=i) = \frac{(\lambda T)^i e^{-\lambda T}}{i!}, \quad i = 1, 2, \dots, \infty$$

Exponential

- $Exp(\mu)$: $f(x) = \mu e^{-\mu x}$, $x > 0$

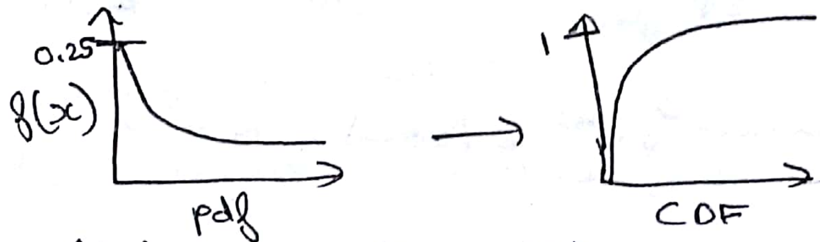
* c.d.f : $F(x) = P(X \le x) = 1 - e^{-\mu x}$

Q) $\lambda = 4, T = 1, P(X=0) = \frac{4^0 \cdot e^{-4}}{1} = e^{-4} = 0.018$

Q) $\lambda = 4, T = 1, P(X=1) = \frac{4 \cdot e^{-4}}{1} = 0.073$

Scanned by CamScanner

Q) [redacted]. For exponential functions,



$$f(x) = \mu e^{-\mu x} \Rightarrow f(0) = \mu = 0.25$$

- $P(X \leq 1) = 1 - e^{-\mu x} = 0.22$

Service time distribution:

i) Length of packet → Can be a distribution Memoryless,
ii) Bit rate Random, Deterministic etc.

---

* Note: i) Splitting a poisson process to multiple [redacted] processes, they are all poisson. [redacted].
ii) {Sum of poisson is poisson}

8

Service Disciplines:

- FCFS — First come, first serve (Most common)
- Infinite Server (IS) — Fixed delay
- LCFS — Last come, first serve

⇒ Groups of packets can arrive, $M^{[x]}$ where x is the group size (ex: $M^{[x]}$, $G^{[x]}$)

⇒ Bulk service can also be denoted by $M^{[x]}$, $G^{[x]}$ etc.

* • T = Inter arrival time, $\lambda = 1/\mathbb{E}[T]$ = Mean arrival rate
- S = Service time per job, μ = Mean service rate per server $= 1/\mathbb{E}[S]$
- ? m =
- $n = n_q + n_s$ } Number in queue + Number in system.

$\Rightarrow$ Mean number of jobs in system = Arrival rate (λ) $\times$ Mean response time ($\mathbb{E}(T)$)

$\Rightarrow$ Mean num of jobs in queue = Arrival rate $\times$ Mean waiting time

* In steady state $\lambda < m\mu$ → Mean service rate
  - λ ↓ Arrival rate
  - m ↓ Number of servers

---

* Stochastic Processes:

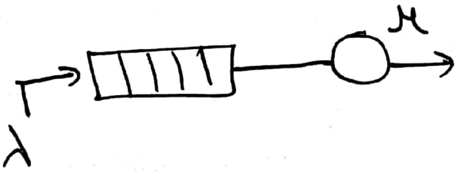
↳ Discrete or continuous state processes
- → Stochastic chain (state based processes with probability to move from one state to another)
- → Markov process (Memoryless)
  - ↳ Discrete state markov chain
    - (M|M|m queues can be modelled with markov chains & processes)
- → Birth-Death processes
  - (Discrete space markov process where state i can only move to state $i-1$ & $i+1$ only)
- → Poisson process
  - Interarrival time is IID (Independent & Identically Distributed) and exponential.
  - Splitting | Merging gives poisson distribution

---

1)

* M|M|1 Queue:

μ



λ

* ρ = Utilization of server $= \lambda/\mu$

* Average total delay $\mathbb{E}[T] = \frac{1}{\mu-\lambda} = \frac{1}{\mu(1-\rho)}$?

  ($\rho < 1$)

* $\mathbb{E}[W] = \mathbb{E}[T] - \frac{1}{\mu} = \frac{1}{\mu-\lambda} - \frac{1}{\mu}$

  $= \frac{\rho}{\mu(1-\rho)}$

  ↓ Waiting time average

Scanned by CamScanner

• Average number of packets in system (Little's Law)

$$\mathbb{E}[n] = \lambda \times \mathbb{E}[T] = \frac{\lambda}{\mu - \lambda} = \frac{\rho}{1-\rho}$$

• Avg waiting time in queue $= \mathbb{E}[w] = \mathbb{E}[T] - \frac{1}{\mu}$

Avg no of packets in queue $= \mathbb{E}[n_q] = \lambda \times \mathbb{E}[w]$

$$= \frac{\rho^2}{1-\rho}$$

---

ex: $\lambda = 10000$pps

Service = 100 Mbps

Packet length = Exp(1000bytes)

$\mathbb{E}[T] = ?$, $\mathbb{E}[w] = ?$, $\mathbb{E}[n] = ?$, $\mathbb{E}[n_q] = ?$

$\Rightarrow$ Service rate $= \mu = \cancel{\frac{100\times 8}{100\times 10}} \quad \frac{100\times 10^6}{1000\times 8}$

$$\mu = 125000 \text{ pps}$$

$$\mathbb{E}[T] = \frac{1}{\mu - \lambda} = \frac{1}{2500} = 4\times10^{-4}\,s$$

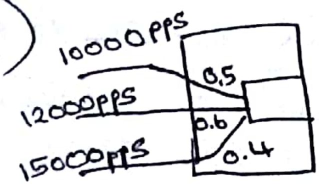
$$\rho = \lambda/\mu = 0.8$$

$$\mathbb{E}[w] = \frac{\cancel{0.8}\,4}{12500\times\cancel{0.2}} = 3.2\times10^{-4}\,s$$

$$\mathbb{E}[n] = \frac{\rho}{1-\rho} = \frac{0.8}{0.2} = 4$$

$$\mathbb{E}[n_q] = 10000 \times 3.2\times10^{-4} = 3.2$$

Q)



• Effective arrival rate = 18200pps

• Mean packet length = 1000byte

Avg total delay per packet = 400 μs

• Average service rate $= \mu = ?$

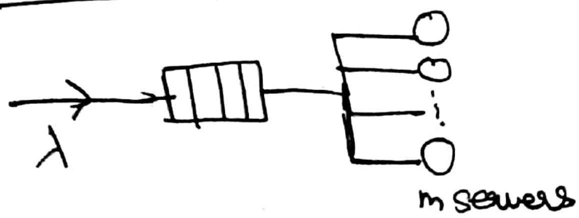
$$\mathbb{E}[T] = \frac{1}{\mu - \lambda} = 400\times10^{-6}$$

• Bitrate $= \mu \times 1000 \times 8 = 165.6$Mbps

$\therefore \mu = 20{,}700$pps

Note:

M/M/m Queue:



M/M/m/B Queue:

- m server queue with a buffer size of B

9

2) M/D/1 Queue:

D → Deterministic

$$E[W]_{M/D/1} = \frac{1}{2} E[W]_{M/M/1}$$

$$= \frac{1}{2} \frac{1}{\mu} \frac{\rho}{1-\rho}$$

$$\Rightarrow E[T] = E[W] + \frac{1}{\mu} = \frac{\rho}{2\mu(1-\rho)} + \frac{1}{\mu}$$

$$E[n] = \lambda \cdot E[T] = \frac{\rho^2}{2(1-\rho)} + \rho$$

$$E[n_q] = \lambda \cdot E[W] = \frac{1}{2} \cdot \frac{\rho^2}{1-\rho}$$

3) M/G/1 Queue:

G → General

- λ – arrival rate (poisson)

$E[s]$ = Average service time (or mean packet length)

σ = Standard deviation of service time

C_s = Coefficient of variation $= \sigma / E[s]$

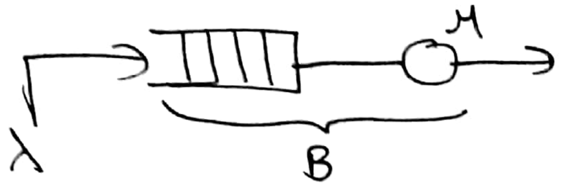
$$\Rightarrow \rho = \lambda E[s]$$

$$E[W] = \frac{\rho E[s] (1 + C_s^2)}{2(1-\rho)}$$

⇒ For M/M/1 Queue, $c_s = 1$, $E[W] = \frac{\rho}{\mu(1-\rho)}$

⇒ For M/D/1 Queue, $E[S] = \frac{1}{\mu}$, $C_s = 0$,
$$E[W] = \frac{\rho}{2\mu(1-\rho)}$$

4) M/M/1/B System:

* 

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

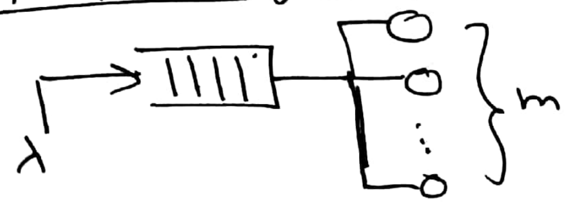
- P_B = Probability that packets get dropped

⇒ Effective arrival rate $= \lambda - \lambda P_B$
$= \lambda(1-P_B)$

- Blocking / Dropping prob (P_B) $= \frac{(1-\rho)\rho^B}{1-\rho^{B+1}}$

$$E[n] = \frac{\rho}{1-\rho} - \frac{(B+1)\rho^{B+1}}{1-\rho^{B+1}}$$

* 5) M/M/m System:

* 

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

P_0 = Probability that there are no packets in the system

$$= \left[\sum_{i=0}^{m-1} \frac{(m\rho)^i}{i!} + \frac{(m\rho)^m}{m!(1-\rho)}\right]^{-1}$$

- Probability of queueing (Q) $= \frac{(m\rho)^m}{m!(1-\rho)} \cdot P_0$

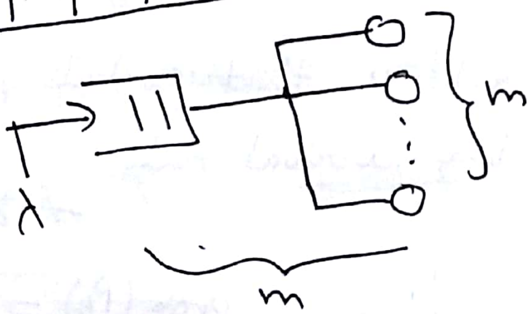
$$E[n_q] = \frac{\rho G}{1-\rho}$$

$$\Rightarrow E[w] = \frac{\rho G}{\lambda(1-\rho)}$$

$$E[T] = E[w] + 1/\mu$$

$$E[n] = \lambda E[T]$$

6) M/M/m/m System:



- $E[T] = 1/\mu$

  $E[w] = \phi$ → Zero

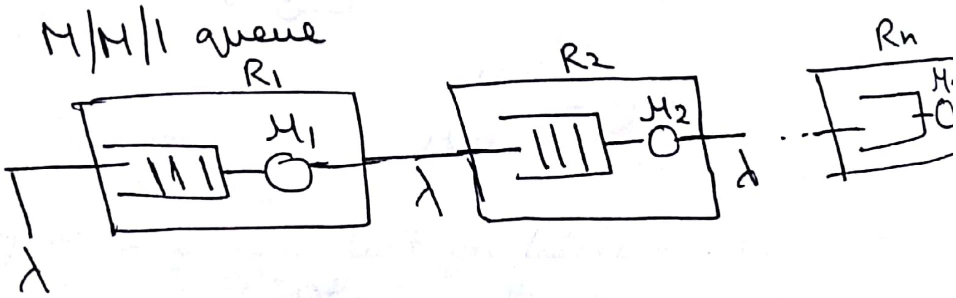
- Effective arrival rate $= \lambda(1-P_B)$

$$\Rightarrow E[n] = \frac{\lambda(1-P_B)}{\mu}$$

$$E[n_q] = \phi$$

* Path:

* • For a system of routers, each has M/M/1 queue



Stable if $\lambda < \mu_i \ \forall i \in 1 \ldots n$

- Avg end to end delay $= \frac{1}{\mu_1-\lambda} + \frac{1}{\mu_2-\lambda} + \ldots + \frac{1}{\mu_n-\lambda}$

$$= \sum_{i=1}^{n} \frac{1}{\mu_i-\lambda}$$

Q) For M|M|1|B queue,

$\lambda = 10,000$ pps

$\mu = 12,500$ pps

$B = 10$, $P_B = ?$, $E[n] = ?$

Blocking probability, $P_B = \dfrac{(1-\rho)\cdot \rho^B}{1-\rho^{B+1}} = 0.0233$

$\rho = \lambda/\mu = 0.8$

$$\rho = \frac{\lambda(1-P_B)}{\mu}$$

$$E[n] = \frac{\rho}{1-\rho} - \frac{(B+1)\rho^{B+1}}{1-\rho^{B+1}}$$

Q) For the above system,

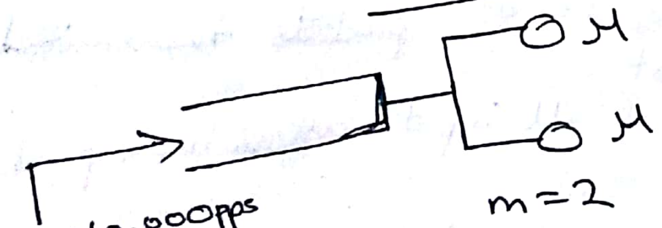
if $P_B \leq 0.0001$

$B = ?$

$$P_B = \frac{(1-\rho)\cdot \rho^B}{1-\rho^{B+1}}$$

- A system that has less average response time performs better than any other system.

Q)



$m = 2$

- $\mu = 25,000$ pps

$$\rho = \frac{\lambda}{m\mu} = \frac{10,000}{2 \times 25000} = 0.2$$

$$P_0 = \left[\sum_{i=0}^{m-1} \frac{(m\rho)^i}{i!} + \frac{(m\rho)^m}{m!(1-\rho)}\right]^{-1}$$

$$P = \left[1 + m\rho + \frac{(m\rho)^2}{2\times(1-\rho)}\right]^{-1}$$

$$= \left[1 + 2\times0.2 + \frac{(2\times0.2)^2}{2\times0.8}\right]^{-1}$$

$$= 0.67$$

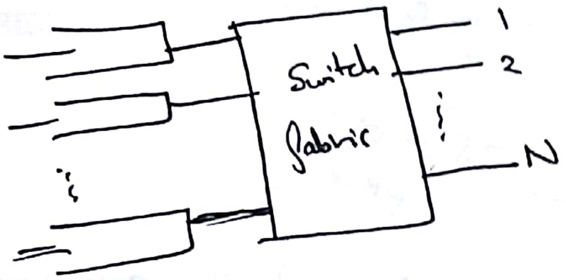
$$G = \frac{(m\rho)^m}{m!\,(1-\rho)} \cdot P_0 = 0.067$$

$$E[T] = \frac{\rho G}{\lambda(1-\rho)} + \frac{1}{\mu} = 41.67\,ms$$

$$E[n] = \lambda \cdot E[T] = 0.4167$$

---

10 **Input Queueing:**



- Assumptions
  - time is slotted
  - 1 time slot = 1 packet transmission
  - packets are of fixed length.

→ n packets generated at every time slot.

→ F = ~~[illegible]~~ no of packets transmitted per slot
(Max if all input buffers have packets)

→ B^i = No of packets destined for output port i in a slot, that are not selected for transmission due to output contention

$$F = N - \sum_{i=1}^{N} B^i \text{, in a given slot.}$$

$$\Rightarrow E[F] = N - \sum_{i=1}^{N} E[B^i]$$

$\Rightarrow B^i \Rightarrow$ iid random variable

$$\therefore E[F] = N - N E[B^i] \rightarrow (1)$$

$$\gamma_0 = \text{Switch throughput} \quad (0 \le \gamma_0 \le 1)$$

$$E[F] = N \cdot \gamma_0 \rightarrow (2)$$

$\Rightarrow$ From (1), (2)

$$1 - E[B^i] = \gamma_0$$

$$E[B^i] = 1 - \gamma_0 \quad \} \forall i$$

$$\cancel{\therefore \gamma_0 = F/N}$$

$\Rightarrow$ This system represents an M/D/1 Queue

$$\therefore E[B^i] = \frac{\gamma_0^2}{2(1-\gamma_0)} = 1 - \gamma_0 \quad \} \text{From above}$$

E[nq] of M/D/1 Queue

$$\therefore \gamma_0^2 = 2(1-\gamma_0)^2$$

$$\gamma_0 = \sqrt{2}(1-\gamma_0)$$

$$\gamma_0 + \sqrt{2}\gamma_0 = \sqrt{2}$$

$$\gamma_0 = \frac{\sqrt{2}}{1+\sqrt{2}} \approx 58\%$$

* Scheduling:

• Fairness & performance guarantees

⇒ Scheduler → Scheduling discipline → Order to serve request at HoQ & Manage service queue of requests

⇒ Traffic type:

- → Best effort (BE) → Fairness important } File transfer, internet etc
- → Guaranteed Service application (GS) } Need to reserve resource
  - Performance guarantees important
  - → Delay
  - → Bandwidth
  - → Loss bounds (packet)
  - → Jitter bounds

⇒ Requirements (Scheduling Algos)

- → Simple to implement (BE & GS)
- → Fairness & protection (BE)
- → Performance bounds (GS)
- → Ease & efficiency of admission control (GS)

---

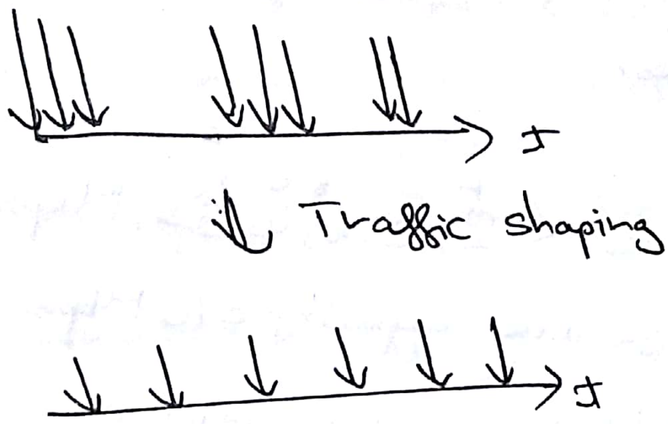
ex: FCFS → Does not provide protection if some ~~[illegible]~~ guy keeps sending large amounts of data which causes delay to others.

Round Robin → Provides protection

---

* Traffic Shaping:

* • Reorganize bursty traffic into evenly spaced input.



i). Leaky bucket:

ii). Token bucket algorithm:

---

* Conservation Law:

- 'Work conserving' system
- If the scheduler is work conserving then,

$$\sum_{i=1}^{N} \rho_i q_i = \text{Constant}$$

where N = # connections at scheduler

ρ_i = Mean utilization of a link due to connection i

q_i = Mean waiting time at scheduler.

i) Max - Min fairness:

*

- Resources are allocated in order of increasing demand.
- No source gets resource larger than their demand

- Unsatisfied demands get equal share of resources.

---

ex: 4 sources $\langle \overset{A}{2}, \overset{B}{2.6}, \overset{C}{4}, \overset{D}{5} \rangle$ Mbps

***

Service capacity = 10 Mbps

→ Run <u>max-min</u> fairness

- Sorted order ⇒ 2, 2.6, 4, 5

⇒ Ideal split = 10/4 = 2.5 Mbps

↳ Going in sorted order, A has 0.5 extra, so split b/w B, C, D.

∴ A → 2
B → 2.5 + 0.5/3 = 2.66
C → 2.5 + 0.5/3 = 2.66
D → 2.5 + 0.5/3 = 2.66

↳ Next move to B, has 0.06 extra, share equally b/w C & D

B → 2.6
C → 2.66 + 0.03 = 2.7
D → 2.66 + 0.03 = 2.7

↳ C does not have any extra, so stop

∴ Allocation: $\langle \overset{A}{2}, \overset{B}{2.6}, \underbrace{\overset{C}{2.7}, \overset{D}{2.7}}_{\text{Backlog users}} \rangle$

Users who are not satisfied. ← Backlog users (Backlogs)

---

ii) Weighted Max-min fairness:

- Each link has a weight w_i associated with it. (Rest are similar to max-min, except initial allocation is done wrt weights)

ex: Capacity = 16 Mbps

$$\text{Demand} = \langle \overset{A}{4}, \overset{B}{2}, \overset{C}{10}, \overset{D}{4} \rangle$$

$$\text{Associated weights} = \langle 2.5, 4, 0.5, 1 \rangle$$

$\Rightarrow$ Normalized initial capacity $= \langle 5, 8, 1, 2 \rangle$

~~Now start with A: Excess = 5 − 4 = 1~~

	A	B	C	D
Init Alloc	5	8	1	2
Required	4	2	10	4
Excess	1	6	—	—

$\underbrace{1 + 6}_{7}$ has to be divided amongst C & D. based on their weight.

	C	D
Alloc	$1 + 7 \times \frac{1}{3} = 3.33$	$2 + 7 \times \frac{2}{3} = 6.66$
Required	10	4
Excess	—	2.66

$\underbrace{2.66}$ Is given to C

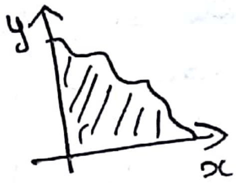
	C
Alloc	6
Required	10
Excess	—

Done, C is a backlog user.

Final alloc: $\langle 4, 2, 6, 4 \rangle$

* Admission Control:

- Schedulable region:
  ↓
  Large region means our Scheduling is good.



} Performance bound 1 vs Performance bound 2?
↓
For 2 connections

(II) Best effort Scheduling:

* 1) Generalized Processor Sharing: (GPS)
(Ideal) — Maximum fairness

- Assume 1 byte of each available queues is served at a single round. So max number of rounds = max (packet size)
- Any connections having packets is called a backlog connection.

Connection weight
↑
⇒ Given a backlog connection 'i' , $\phi(i)$
Non backlog connection 'j' , $\phi(j)$

$S(i, \tau, t)$ = No. of services transmitted from connection i in ~~time~~ τ to t.

$$\frac{S(i,\tau,t)}{S(j,\tau,t)} \geq \frac{\phi(i)}{\phi(j)}$$

⇒ $g(i,k)$ = Service rate allocated to connection 'i' at k^{th} switch.

⇒ $r(k)$ = Service rate of the k^{th} switch

$$g(i,k) = \frac{\phi(i,k)}{\sum_j \phi(j,k)} \times r(k) \quad , \quad g(i) = \sum_{k=1}^{N} \min g(j,k)$$

⇒ Relative fairness bound (RFB): (Performance metric)

$$* \quad RFB = \left| \frac{S(i,T,t)}{g(i)} - \frac{S(j,T,t)}{g(j)} \right|$$

⇒ Absolute fairness bound (AFB):

$$AFB = \left| \frac{S(i,T,t)}{g(i)} - \frac{G(i,T,t)}{g(j)} \right|$$

→ GPS Algo

*2) Weighted Round Robin: (WRR)

- Simple emulation of GPS
- Serves a packet from each non-empty queue

ex:

	A	B	C	
				} Assume same packet size
weights	0.5	0.75	1	
Normalized weights	2	3	4	} To get integers

∴ In round 1, we serve 2 packets of A, then 3 packets of B & then 4 packets of C and so on

ex:

	A	B	C
weights	0.5	0.75	1
Mean pckt len	50	500	1500 bytes
wt / Packet len	0.01	0.0015	0.00067
	60	9	4

∴ In round 1 we serve 60 packets of A, 9 packets of B & 4 packets of C & move on.

- In real life scenarios we don't know mean packet length so we need other methods.

*3) Deficit Round Robin: (DRR)

(short time it is not fair)

* - Modified WRR
- Each ~~counter~~ connection is set to a deficit counter, $DC_i = 0$
- Let Q_i be the quantum size for connection i.

  $Q_i = Q \ \forall i$ if same weight.
- For a weighted system,

  $Q_i = Q \times$ connection's weight $= Q \times \phi(i)$

$\Rightarrow$ if ($DC_i + Q_i \geq$ packet size at head of queue)

```
{
    Transport packet from connection i
    DC_i = DC_i + Q_i - packet size
}
else
{
    DC_i += Q_i
}
```

$\rightarrow$ If no packets at head of connection i, $DC_i = 0$.

---

ex:

	A	B	C
packet size	1500	800	1200

(Assume same weight, $Q = 1000$)

- We perform rounds until all the packets are sent.

Round 1 : $DC_A = 1000$ (packet not sent)

$DC_B = 0 + 1000 - 800 = 200$ (packet sent)

$DC_C = 1000$ (packet not sent)

Round 2 :

$DC_A = 2000$ ~~(packet not sent)~~ $-1500 = 500$ (packet sent)

$DC_C = 2000 - 1200 = 800$ (packet sent)

---

ex: (Weighted DRR example with multiple packets of different sizes in each queue in Harsha's not

* • If capacity = 3690 & weights are 0.5, 0.75 & 1,

Max min fair allocation (ideal)

$A's\ share = \frac{0.5}{2.35} \times 3690 = 820$

$B's\ share = \frac{0.75}{2.35} \times 3690 = 1230$

$C's\ share = \frac{1}{2.35} \times 3690 = 1640$

⇒ From DRR, A's share = 750 → after 2 rounds

B's share = 1040

C's share = 1900

∴ If we perform DRR for loads of iterations then it will be close to max-min, in a small no. of rounds it's not close.

Improvements to DRR:

* Note: • One straightforward improvement to DRR is to add Q_i to DC_i in each round & keep sending packets when $DC_i \geq$ packet size & DC_i −= pck size

• After K rounds ($K > 1$), set $DC_i = 0$

• Set an upper limit to DC_i.

Scanned by CamScanner

(12) Weighted Fair Queueing (WFQ)

* 4) • Finish number (FN) ~~When the packet completely served~~ is

**

• Round number (RN)

<u>FN:</u>

⇒ A packet's finish time under GPS is called as finish number, it is a tag that indicates the relative order in which a packet must be served. It has nothing to do with the actual time at which the packet is served.

⇒ <u>RN:</u>

- The number of rounds of service a bit by bit round robin scheduler has completed at a given time.

* • A service is active when finish number is <u>greater than</u> ~~equal to~~ round number

⇒ $R'(t)$ = Slope of round number $\propto \dfrac{1}{\#\text{ of active connections at } t}$

k^{th} packet

$$F(i,k,t) = \max\left(R(t), F(i,k-1,t)\right) + P(i,k,t)$$

i^{th} connection; time t; → we can serve k^{th} packet only after $k-1^{th}$ is served; → Packet size

* • Buffer drop policy

↳ Sort finish numbers of current packets & if incoming packet has no slots in queue, we delete based on largest F.N.

Scanned by CamScanner

ex: ~~packet = 10 bytes~~

*

*

	A	B	C
t=0	1	2	2
t=4	2	—	—

• Consider GPS scenario,

A	B	C	B	C	A	A

t=0 1 2 3 4 5 6 7

• 1 byte per second sent.

⇒ Now consider WFQ,

t	R(t)	F.N	No. of active Connections	R'(t)	Actions
0	0	F(A,1)=1 F(B,1)=2 F(C,1)=2	3	1/3	Tx (A,1)
1	$0+\frac{1}{3}\times1$ =0.33	—	3	1/3	Finish (A,1) Tx (B,1)
3	$0.33+\frac{1}{3}\times2$ =1	—	2	1/2	Finish (B,1) Tx (C,1)
4	$1+\frac{1}{2}\times1$ =1.5	F(A,2) = 1.5+2 =3.5	3	1/3	—
5	$1.5+\frac{1}{3}\times1$ =1.83	—	3	1/3	Finish (C,1) Tx (A,2)
6	$1.83+\frac{1}{3}\times1$ =2	—	1	1	B, C Inactive A is active

• $R(t) = R(t)_{\text{until now}} + \frac{T - T_{\text{until now}}}{N}$

*when considering t, select t where next event occurs like finish (x,y) & x inactive & Tx(x,y) etc. (example in harsha's

~~Iterated Deletion:~~

$\Rightarrow$ Iterated deletion is an issue we face in WFQ,

* 5 packets of size 1

$t=5$, $RN = 0 + \frac{1}{5} \times 5 = 1$ } Expected

* If one packet ends early,

$t=4$, $RN = \frac{1}{5} \times 4 + \frac{1}{4} \times 1 = 1.05$

This issue is addressed in WF2Q ← { (If 1 packet is inactive, this would make all other packets also inactive since their F(N)'s are 1)

$$\Rightarrow F(i,k,t) = \max\left(R(t) + F(i,k-1,t)\right) + \frac{P(i,k,t)}{\phi(i)}$$

weight of connection i. ← $\phi(i)$

* Absolute fairness bound,

$$\frac{S(i,T,t)}{g(i)} \geq \frac{G(i,T,t)}{g(i)} - \underbrace{\frac{P_{max}}{g(i)}}_{\text{Lower bound}}$$

Another disadvantage of WFQ. ← (Upper bound is unknown)

---

5) Worst case fair queue: (WF^2Q)

$$\frac{G(i,T,t)}{g(i)} + \frac{P_{max}}{g(i)} \geq \frac{S(i,T,t)}{g(i)}$$

P_{max} → Max packet length

(Gives upper bound)

$$\Rightarrow RFB = \frac{2P_{max}}{g(i)}, \quad AFB = \frac{P_{max}}{g(i)}$$

- Find active connections first & then compute finish numbers, rest same as WFQ.

?

*6) Self-clocked fair queue :(SCFQ)

ex:

Connections	1	2	......	50	51
				1	50
Weight	1	1	------	0	0+8≃1
Time	0	0	------	1	1
Packet Size	1	1	------		

$$FN(51,1) = 1 + \frac{1}{50} = 1.02$$

?

*7) Start-time fair queue :(STFQ)

- Start Number SN = For an inactive connection it is set to current round number. otherwise set to FN of previous.

$$\Rightarrow R(t) = SN(i,k,t)$$

$$F(i,k,t) = SN(i,k,t) + \frac{P(i,k,t)}{\phi(i)}$$

$\phi(i)$ → weight of connection

- Service in order of increasing SN.

⇒ This ensure there is no short term unfairness.

- Better worst case delay than SCFQ

* Note: If FN $\leq$ RN at some time, then rewind t such that RN = FN & proceed.

eric: *

t	A	B	C	D
0	240	120	960	—
50	160	480	640	400
100	—	—	160	—
1500	—	—	—	360

(Assume all are weighted equally $\phi(i) = 1 \; \forall i$)

t	RN	FN	N	R'(t)	Actions
0	0	F(A,1)=240 F(B,1)=120 F(C,1)=960	3	1/3	Tx(B,1)
50	$0+\frac{50}{3}$ =16.67	F(A,2)=240 +160 =400 F(B,2)=120 +480 =600 F(C,2)=1600 F(D,2)=416.67	4	1/4	
100	$16.67+\frac{50}{4}$ = 29.17	F(C,3)=1600 +160 =1760	4	1/4	
120	$29.17+\frac{20}{4}$ = 34.17	—	4	1/4	Finish(B,1) Tx(A,1)
(120+240) 360	$34.17+\frac{240}{4}$ = 94.17	—	4	1/4	Finish(A,1)