

1. >12 pacotes (0.5)
2. >10 pacotes (0.25)
3. >4 pacotes (0.25)

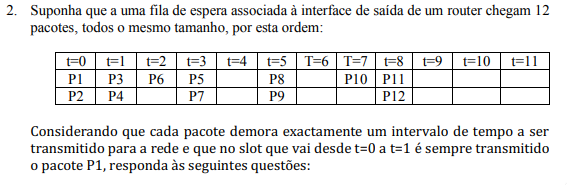
Agrupando de 4 em 4, vem que:

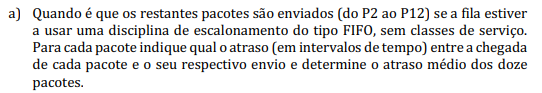
4\*0.5=2;

4\*0.25=1;

(Agrupou-se de 4 em 4 uma vez que é o menor número pelo qual tem que se multiplicar cada prioridade para dar apenas inteiros).

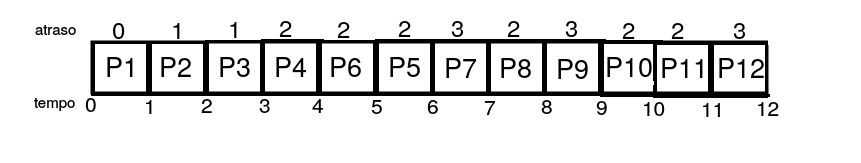
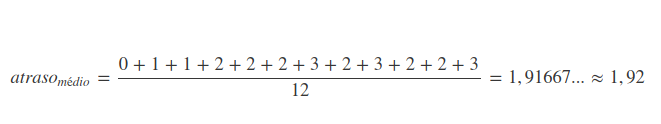
AABC AABC AABC AABC AAB AAB BBBB

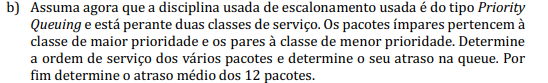




No FIFO, (First in First out), os pacotes são enviados conforme chegam, (o que

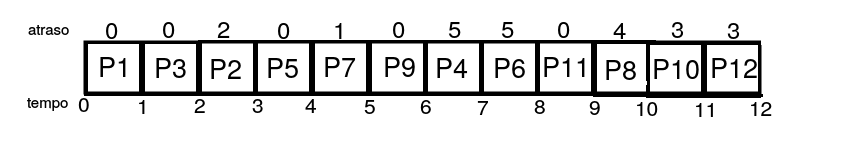
chegue primeiro sai primeiro)

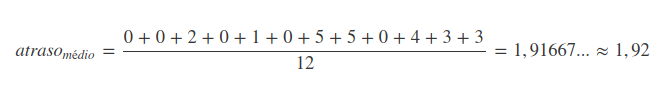
 



Maior prioridade->P1,P3,P5,P7,P9,P11

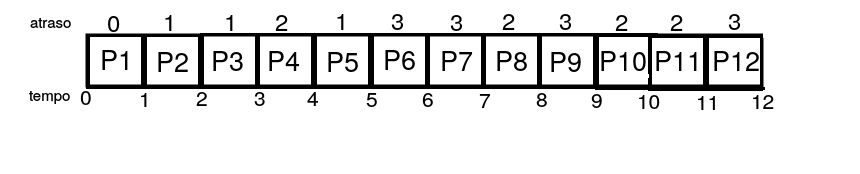
Menor prioridade->P2,P4,P6,P8,P10,P12

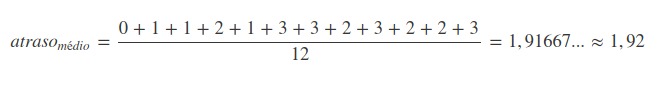






Neste tipo de escalonamento, os pacotes são enviados em alternância de prioridade (se acabou-se de enviar um pacote com menor prioridade, então de seguida, envia-se outro de maior prioridade).

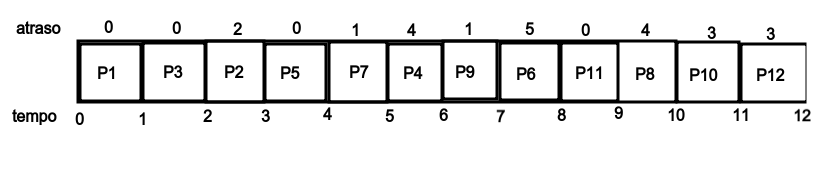


WFQ->Se já se mandou 2 do mais prioritário, tem que se mandar 1 do menos prioritário.

Classe ímpar->peso 2;

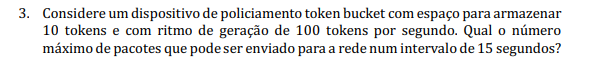
Classe par->peso 1;







Conclui-se que a política de escalonamento não influencia o atraso médio de envio dos pacotes, apenas influencia o atraso individual de cada pacote.



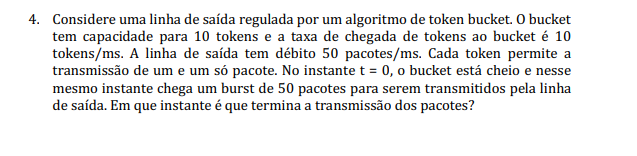
O policiamento token bucket limita a entrada tendo em conta o tamanho do burst(nº máximo de pacotes enviados consecutivamente) e a taxa média especificada.

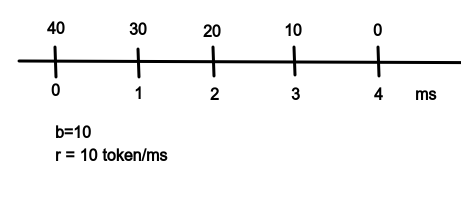
Espaço para 10 tokens.->b

Geração de 100 tokens/s-> r

Nº máx. de pacotes em 15 s?-> t

(rt+b)=100\*15+10=1510 pacotes em 15 segundos





Os primeiros 10 pacotes são logo enviados, logo ficam a faltar 40 pacotes a 10 tokens/ms, daí sai que 40/10= **4ms**(tempo que demora a transmissão).