



# **Revisões sobre Probabilidades, Variáveis Aleatórias e Processos Estocásticos**

Desempenho e Dimensionamento de Redes

Prof. Amaro de Sousa ([asou@ua.pt](mailto:asou@ua.pt))

DETI-UA, 2020/2021

# Experiência aleatória

- Numa experiência aleatória, o espaço de resultados,  $S$ , é o conjunto de todos os resultados possíveis da experiência
- Qualquer subconjunto  $E$  do espaço de resultados  $S$  designa-se por evento ou acontecimento
- Dados dois acontecimentos  $E$  e  $F$ , podem-se definir outros acontecimentos:
  - A união dos acontecimentos,  $E \cup F$
  - A intersecção dos acontecimentos,  $EF$
- Quando  $EF = \emptyset$  ( $\emptyset$  é o conjunto vazio) os acontecimentos dizem-se mutuamente exclusivos
- O complemento de  $E$ ,  $E^c$ , é o conjunto de elementos de  $S$  que não pertencem a  $E$

# Probabilidades definidas sobre acontecimentos

- Para cada acontecimento  $E$  de  $S$ , admite-se a existência de um número  $P(E)$  designado por probabilidade de  $E$ , se satisfaz as seguintes condições:
  - (1)  $0 \leq P(E) \leq 1$
  - (2)  $P(S) = 1$
  - (3) Para qualquer conjunto de acontecimentos mutuamente exclusivos  $E_1, E_2, E_3, \dots$

$$P\left(\bigcup_i E_i\right) = \sum_i P(E_i)$$

- Corolários:

$$P(E) + P(E^c) = 1$$

$$P(E \cup F) = P(E) + P(F) - P(EF)$$

# Probabilidades condicionadas

- Dados dois acontecimentos  $E$  e  $F$ , a probabilidade condicionada de  $E$  ocorrer dado que  $F$  ocorreu designa-se por  $P(E|F)$  e é definida por

$$P(E|F) = P(EF)/P(F)$$

- Dois acontecimentos  $E$  e  $F$  dizem-se acontecimentos independentes se

$$P(EF) = P(E)P(F)$$

- Se  $E$  e  $F$  são independentes, então:

$$P(E|F) = P(E) \quad \text{e} \quad P(F|E) = P(F)$$

ou seja, se o conhecimento que um acontecimento ocorreu não afetar a probabilidade do outro ocorrer.

- Sejam  $F_1, F_2, \dots, F_n$  acontecimentos mutuamente exclusivos tais que a sua união forma o espaço de resultados  $S$ . Então,

$$P(E) = \sum_{i=1}^n P(EF_i) = \sum_{i=1}^n P(E|F_i)P(F_i)$$

# Regra de Bayes

Sejam  $F_1, F_2, \dots, F_n$  acontecimentos mutuamente exclusivos tais que a sua união forma o espaço de resultados  $S$ .

Tendo ocorrido o acontecimento  $E$ , a probabilidade de  $F_j$  ( $j = 1, 2, \dots, n$ ) ter ocorrido é dada por:

$$P(F_j | E) = \frac{P(EF_j)}{P(E)} = \frac{P(E|F_j)P(F_j)}{P(E)} = \frac{P(E|F_j)P(F_j)}{\sum_{i=1}^n P(E|F_i)P(F_i)}$$

## Probabilidades condicionadas – Exemplo 1

Num teste de escolha múltipla, um estudante sabe a resposta certa com probabilidade  $p$  e adivinha a resposta com probabilidade  $1 - p$ . Ao adivinhar a resposta, o estudante acerta com probabilidade  $1/m$ , sendo  $m$  o número de alternativas de escolha múltipla.

Determine a probabilidade de um estudante (i) responder corretamente à pergunta e (ii) saber a resposta dado que a respondeu corretamente.

Acontecimentos:  $E$  – o aluno responde corretamente

$F_1$  – o aluno sabe a resposta

$F_2$  – o aluno não sabe a resposta

$$\begin{aligned}(i) P(E) &= P(E|F_1)P(F_1) + P(E|F_2)P(F_2) \\&= 1 \times p + 1/m \times (1 - p) = \\&= p + (1 - p)/m\end{aligned}$$

$$\begin{aligned}(ii) P(F_1|E) &= P(E|F_1)P(F_1) / P(E) \\&= 1 \times p / [p + (1 - p)/m] = \\&= p m / [1 + (m - 1) p]\end{aligned}$$

Se  $p = 50\%$  e  $m = 4$ , então (i)  $P(E) = 62.5\%$  e (ii)  $P(F_1|E) = 80\%$

## Probabilidades condicionadas – Exemplo 2

Numa ligação sem fios (wireless) entre dois equipamentos, a probabilidade dos pacotes de dados serem recebidos com erros é de 0.1% em condições normais ou de 10% quando há interferências. A probabilidade de haver interferência é de 2%. Os equipamentos têm a capacidade de verificar na receção se os pacotes de dados foram recebidos com erros ou não.

Determine: (i) a probabilidade de um pacote ser recebido com erros e (ii) se um pacote for recebido com erros, qual a probabilidade da ligação estar com interferência.

Acontecimentos:  $E$  – o pacote é recebido com erros

$F_1$  – a ligação está em condições normais

$F_2$  – a ligação está com interferência

$$\begin{aligned}(i) P(E) &= P(E|F_1)P(F_1) + P(E|F_2)P(F_2) \\&= 0.001 \times (1 - 0.02) + 0.1 \times 0.02 \\&= 0.00298 = 0.298\%\end{aligned}$$

$$\begin{aligned}(ii) P(F_2|E) &= P(E|F_2)P(F_2) / P(E) \\&= 0.1 \times 0.02 / 0.00298 \\&= 0.671 = 67.1\%\end{aligned}$$

# Variáveis aleatórias

- Uma variável aleatória  $X$  é uma função que atribui um número real a cada ponto do espaço de resultados  $S$  de uma experiência aleatória.
- A função distribuição (ou função de distribuição cumulativa) da v.a.  $X$  é:

$$F(x) = P(X \leq x) , -\infty < x < +\infty$$

- Propriedades da função distribuição:
  - (1)  $0 \leq F(x) \leq 1$  para todo o  $x$
  - (2) se  $x_1 \leq x_2$  então  $F(x_1) \leq F(x_2)$  (função não decrescente)
  - (3)  $\lim_{x \rightarrow -\infty} F(x) = 0$  e  $\lim_{x \rightarrow +\infty} F(x) = 1$
  - (4)  $P(a < X \leq b) = F(b) - F(a)$ , para  $a < b$

# Variáveis aleatórias discretas

- Uma variável aleatória  $X$  diz-se discreta se puder tomar, quando muito, um número contável de valores  $x_1, x_2, \dots, x_i, \dots$
- Define-se função probabilidade (ou função massa de probabilidade) da v.a discreta  $X$  por

$$f(x_i) = P(X = x_i) \quad \text{para todos os valores de } i = 1, 2, 3, \dots$$

- Obrigatoriamente, tem de acontecer que:  $\sum_{i=1}^{\infty} f(x_i) = 1$
- A função distribuição da v.a discreta  $X$  é:

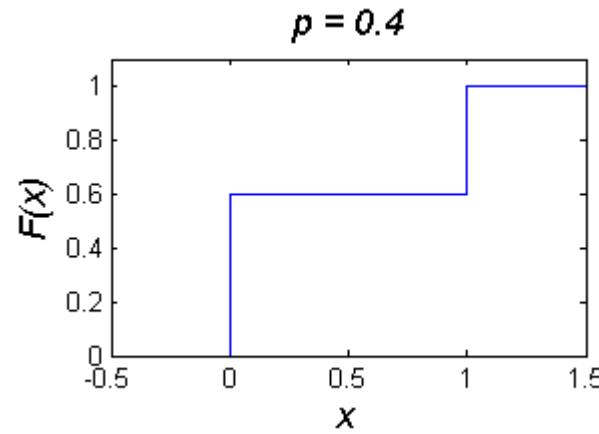
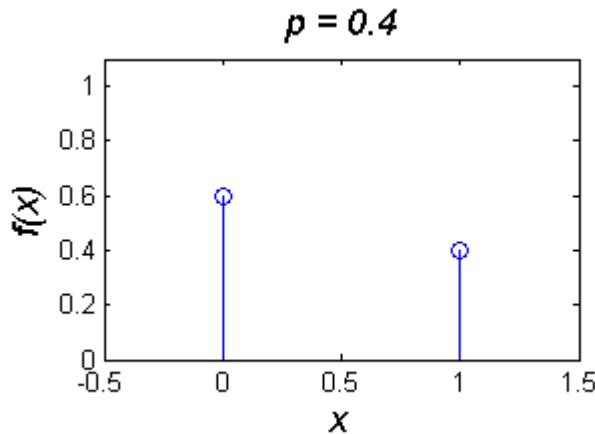
$$F(x) = \sum_{x_i \leq x} f(x_i) \quad , -\infty < x < +\infty$$

# Variáveis aleatórias discretas

Variável aleatória de Bernoulli: experiência que pode resultar em sucesso com probabilidade  $p$  ou insucesso com probabilidade  $1 - p$ .

Se  $X = 1$  representar um sucesso e  $X = 0$  um insucesso, a função probabilidade é:

$$f(i) = p^i(1-p)^{1-i}, i = 0, 1$$



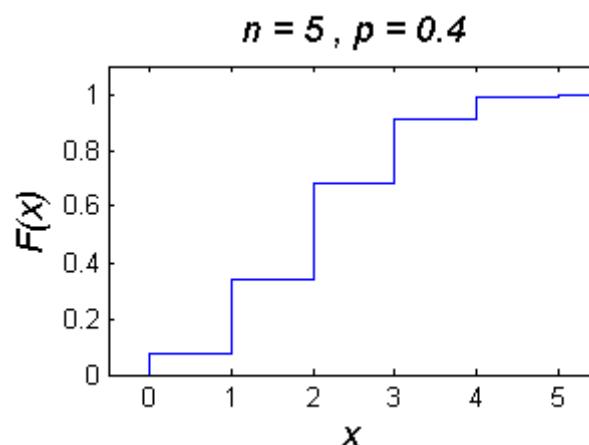
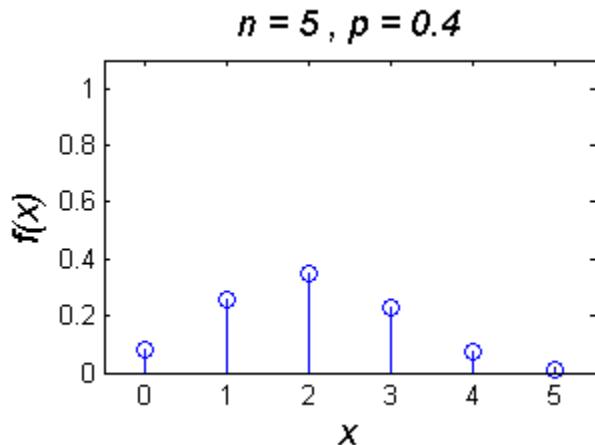
# Variáveis aleatórias discretas

Variável aleatória binomial: conjunto de  $n$  experiências de Bernoulli independentes, cada uma das quais resulta num sucesso com probabilidade  $p$  ou num insucesso com probabilidade  $1 - p$ .

Se  $X$  representar o número de sucessos em  $n$  experiências, a função probabilidade é:

$$f(i) = \binom{n}{i} p^i (1-p)^{n-i}, i = 0, 1, 2, \dots, n$$

onde  $\binom{n}{i} = \frac{n!}{i!(n-i)!}$

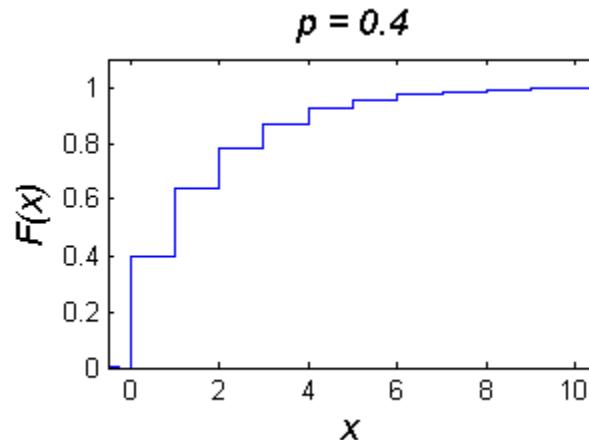
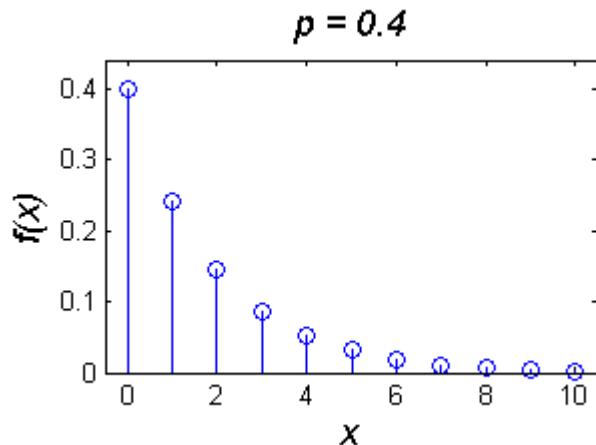


# Variáveis aleatórias discretas

Variável aleatória geométrica: são realizadas experiências de Bernoulli independentes com parâmetro  $p$  (probabilidade de sucesso) até que ocorra um sucesso.

Se  $X$  representar o número de insucessos antes do sucesso, a função probabilidade é

$$f(i) = (1-p)^i p , i = 0, 1, 2, \dots$$



Se  $X$  representar o número de experiências até ao sucesso, a função probabilidade é

$$f(i) = (1-p)^{i-1} p , i = 1, 2, \dots$$

## Variáveis aleatórias discretas – Exemplo 3

Numa dada ligação de dados, a probabilidade de erro de bit (*BER – Bit Error Rate*) é  $10^{-5}$  e os erros em diferentes bits são estatisticamente independentes.

Determine: (i) a probabilidade de um pacote de dados de 100 Bytes ser recebido sem erros e (ii) a probabilidade de um pacote de dados de 1000 Bytes ser recebido com 2 ou mais erros.

O número de erros num pacote é uma variável aleatória binomial em que a probabilidade de sucesso é o BER e o número de experiências de Bernoulli é o número de bits do pacote:

$$(i) \quad f(0) = \binom{n}{0} p^0 (1-p)^{n-0} = \binom{100 \times 8}{0} \times (1 - 10^{-5})^{100 \times 8} = 0.992 = 99.2\%$$

$$\begin{aligned} (ii) \quad 1 - f(0) - f(1) &= 1 - \binom{n}{0} p^0 (1-p)^{n-0} - \binom{n}{1} p^1 (1-p)^{n-1} \\ &= 1 - (1 - 10^{-5})^{8000} - 8000 \times 10^{-5} (1 - 10^{-5})^{7999} = 3.034E - 3 = 0.3\% \end{aligned}$$

# Variáveis aleatórias contínuas

- Uma variável aleatória  $X$  diz-se contínua se existir uma função não negativa  $f(x)$  tal que para qualquer conjunto de números reais  $B$ :

$$P(X \in B) = \int_B f(x)dx \quad \int_{-\infty}^{+\infty} f(x)dx = 1$$

$f(x)$  é a função densidade de probabilidade da v.a contínua  $X$

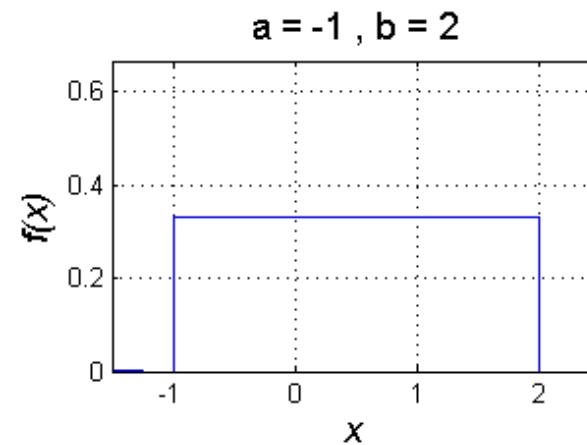
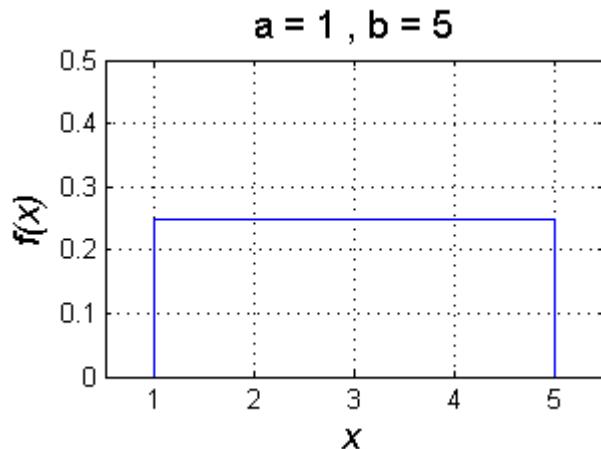
- Resulta então que:  $P\{a \leq X \leq b\} = \int_a^b f(x)dx$
- A função distribuição da v.a contínua  $X$  é:

$$F(x) = P(X \in [-\infty, x]) = \int_{-\infty}^x f(y)dy$$

# Exemplos de variáveis aleatórias contínuas

Variável aleatória com Distribuição Uniforme: uma v.a. diz-se uniformemente distribuída no intervalo  $[a,b]$  se a função densidade de probabilidade for dada por

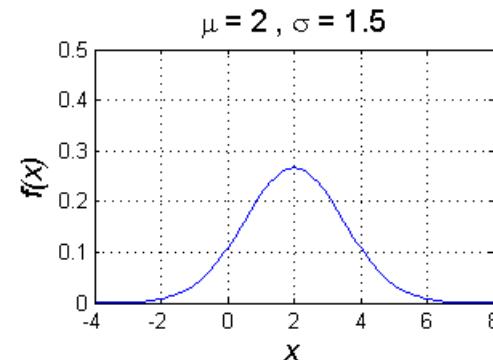
$$f(x) = \begin{cases} \frac{1}{b-a} & , a < x < b \\ 0 & , \text{cc} \end{cases}$$



# Exemplos de variáveis aleatórias contínuas

Variável aleatória com Distribuição Gaussiana (ou Normal): Uma v.a.  $X$  tem uma distribuição Gaussiana com média  $\mu$  e desvio padrão  $\sigma$  se a função densidade é dada por:

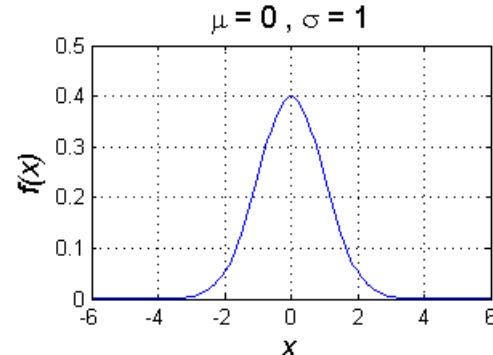
$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2}$$



Designa-se por distribuição Gaussiana (ou Normal) padrão à distribuição Gaussiana com média 0 e desvio padrão 1.

Neste caso:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$



# Média de uma variável aleatória

- Média ou valor esperado de uma v.a.  $X$ :

$$E[X] = \begin{cases} \sum_{j=1}^{\infty} x_j f_X(x_j) & \text{se } X \text{ discreta} \\ \int_{-\infty}^{+\infty} x f_X(x) dx & \text{se } X \text{ continua} \end{cases}$$

- Propriedades importantes:

$$E[cX] = cE[X]$$

$$E\left[\sum_{i=1}^n c_i X_i\right] = \sum_{i=1}^n c_i E[X_i]$$

- Média da v.a.  $Y = g(X)$ :

$$E[g(X)] = \begin{cases} \sum_{j=1}^{\infty} g(x_j) f_X(x_j) & \text{se } X \text{ discreta} \\ \int_{-\infty}^{+\infty} g(x) f_X(x) dx & \text{se } X \text{ continua} \end{cases}$$

# Variância de uma variável aleatória

- Variância de uma v.a.  $X$ :

$$\text{Var}[X] = E[(X - E[X])^2] = E[X^2] - E[X]^2$$

- Propriedades importantes:

$$\text{Var}[X] \geq 0$$

$$\text{Var}[cX] = c^2 \text{Var}[X]$$

$$\text{Var}\left[\sum_{i=1}^n X_i\right] = \sum_{i=1}^n \text{Var}[X_i] \quad \text{se } X_i \text{ forem independentes}$$

## Exemplo 4

Uma ligação de dados de 10 Mbps suporta um fluxo de pacotes cujo tamanho é 100 Bytes com probabilidade 10%, 500 Bytes com probabilidade 50% e 1500 Bytes com probabilidade 40%. Considere a variável aleatória  $X$  representativa do tempo de transmissão dos pacotes.

Determine: (i) o tempo médio (i.e.,  $E[X]$ ) de transmissão dos pacotes (ii) o segundo momento (i.e.,  $E[X^2]$ ) do tempo de transmissão dos pacotes e (iii) a variância (i.e.,  $Var[X]$ ) do tempo de transmissão dos pacotes.

$$(i) \quad E[X] = \sum_{j=1}^{\infty} x_j f_X(x_j) = \frac{100 \times 8}{10^7} \times 0.1 + \frac{500 \times 8}{10^7} \times 0.5 + \frac{1500 \times 8}{10^7} \times 0.4 \\ = 0.688 \times 10^{-3} \text{ seg} = 0.688 \text{ mseg}$$

$$(ii) \quad E[X^2] = \sum_{j=1}^{\infty} (x_j)^2 f_X(x_j) = \left( \frac{100 \times 8}{10^7} \right)^2 \times 0.1 + \left( \frac{500 \times 8}{10^7} \right)^2 \times 0.5 + \left( \frac{1500 \times 8}{10^7} \right)^2 \times 0.4 \\ = 6.5664 \times 10^{-7} \text{ seg}^2$$

## Exemplo 4 - continuação

Uma ligação de dados de 10 Mbps suporta um fluxo de pacotes cujo tamanho é 100 Bytes com probabilidade 10%, 500 Bytes com probabilidade 50% e 1500 Bytes com probabilidade 40%. Considere a variável aleatória  $X$  representativa do tempo de transmissão dos pacotes.

Determine: (i) o tempo médio (i.e.,  $E[X]$ ) de transmissão dos pacotes (ii) o segundo momento (i.e.,  $E[X^2]$ ) do tempo de transmissão dos pacotes e (iii) a variância (i.e.,  $Var[X]$ ) do tempo de transmissão dos pacotes.

$$(iii) \text{ 1ª alternativa: } Var[X] = E[(X - E[X])^2]$$

$$Var[X] = \left( \frac{100 \times 8}{10^7} - E[X] \right)^2 \times 0.1 + \left( \frac{500 \times 8}{10^7} - E[X] \right)^2 \times 0.5 + \left( \frac{1500 \times 8}{10^7} - E[X] \right)^2 \times 0.4 \\ = 1.833 \times 10^{-7} \text{ seg}^2$$

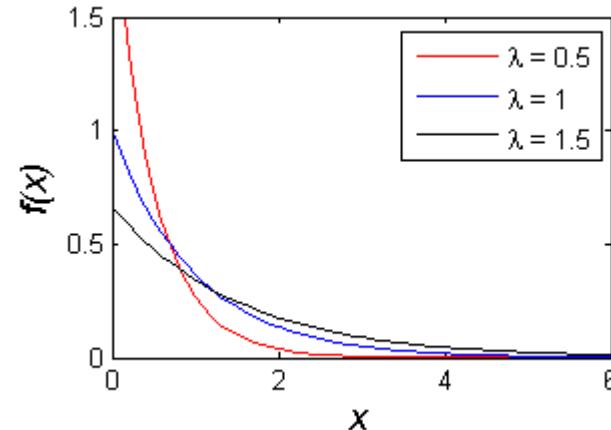
$$2^\text{a alternativa: } Var[X] = E[X^2] - E[X]^2$$

$$Var[X] = 6.5664 \times 10^{-7} - (0.688 \times 10^{-3})^2 = 1.833 \times 10^{-7} \text{ seg}^2$$

# Distribuição exponencial

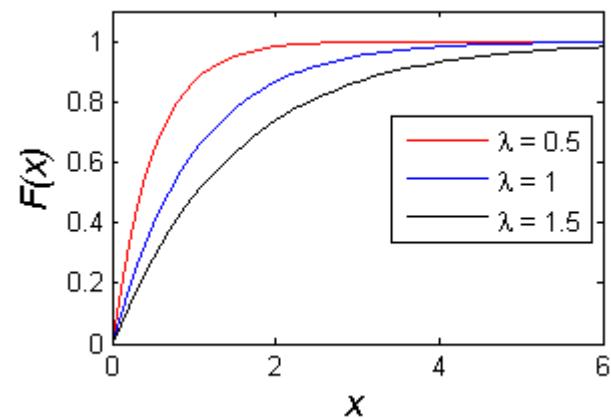
- Uma v. a.  $X$  tem uma distribuição exponencial com parâmetro  $\lambda$ ,  $\lambda > 0$ , se a sua função densidade de probabilidade for:

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$



- A função de distribuição é dada por:

$$F(x) = \begin{cases} 1 - e^{-\lambda x}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$



# Distribuição exponencial

- A média e a variância de uma distribuição exponencial são:

$$E[X] = \frac{1}{\lambda} \quad \text{Var}[X] = \left(\frac{1}{\lambda}\right)^2$$

- A distribuição exponencial não tem memória, isto é,

$$P\{X > s + t \mid X > t\} = P\{X > s\}$$

- Se  $X_1$  e  $X_2$  são v. a. independentes e exponencialmente distribuídas com médias  $1/\lambda_1$  e  $1/\lambda_2$  respectivamente, então

$$P\{X_1 < X_2\} = \frac{\lambda_1}{\lambda_1 + \lambda_2}$$

## Distribuição exponencial – Exemplo 5

Uma ligação de dados de 10 Mbps suporta um fluxo de pacotes cujo tamanho é exponencialmente distribuído com média de 1000 Bytes.

Considere a variável aleatória  $X$  representativa do tempo de transmissão dos pacotes.

Determine: (i) o tempo médio (i.e.,  $E[X]$ ) de transmissão dos pacotes (ii) a variância (i.e.,  $Var[X]$ ) do tempo de transmissão dos pacotes e (iii) o segundo momento (i.e.,  $E[X^2]$ ) do tempo de transmissão dos pacotes.

$$(i) \quad E[X] = \frac{1000 \times 8}{10^7} = 8 \times 10^{-4} = 0.8 \text{ mseg}$$

Capacidade da ligação

$$E[X] = \frac{1}{\mu} \Leftrightarrow \mu = \frac{1}{E[X]} = \frac{1}{8 \times 10^{-4}} = 1250 \text{ pacotes/s}$$

$$(ii) \quad Var[X] = \left( \frac{1}{\mu} \right)^2 = (8 \times 10^{-4})^2 = 6.4 \times 10^{-7} \text{ seg}^2$$

$$(iii) \quad Var[X] = E[X^2] - E[X]^2 \Leftrightarrow E[X^2] = Var[X] + E[X]^2$$

$$E[X^2] = 6.4 \times 10^{-7} + (8 \times 10^{-4})^2 = 1.28 \times 10^{-6} \text{ seg}^2$$

## Distribuição exponencial – Exemplo 6

Uma pessoa entra num banco e encontra os 2 empregados do banco ocupados a servir clientes. Não existem outros clientes no banco, pelo que a pessoa começará a ser atendida logo que um dos clientes que se encontra a ser atendido deixe o banco.

O empregado A serve os clientes segundo uma distribuição exponencial à taxa  $\lambda_A = 12$  clientes por hora e o empregado B serve os clientes segundo uma distribuição exponencial à taxa  $\lambda_B = 8$  clientes por hora. Determine a probabilidade da pessoa que entrou ser o segundo cliente a ser servido.

Evento A – o empregado A termina de servir o seu cliente antes do empregado B

Evento B – o empregado B termina de servir o seu cliente antes do empregado A

Evento C – a pessoa que entrou é o segundo cliente a ser servido

$$\begin{aligned} P(C) &= P(C | A) \times P(A) + P(C | B) \times P(B) = \\ &= \frac{\lambda_A}{\lambda_A + \lambda_B} \times \frac{\lambda_A}{\lambda_A + \lambda_B} + \frac{\lambda_B}{\lambda_A + \lambda_B} \times \frac{\lambda_B}{\lambda_A + \lambda_B} = \\ &= \frac{12}{12+8} \times \frac{12}{12+8} + \frac{8}{12+8} \times \frac{8}{12+8} = 0.52 = 52\% \end{aligned}$$

# Processos estocásticos

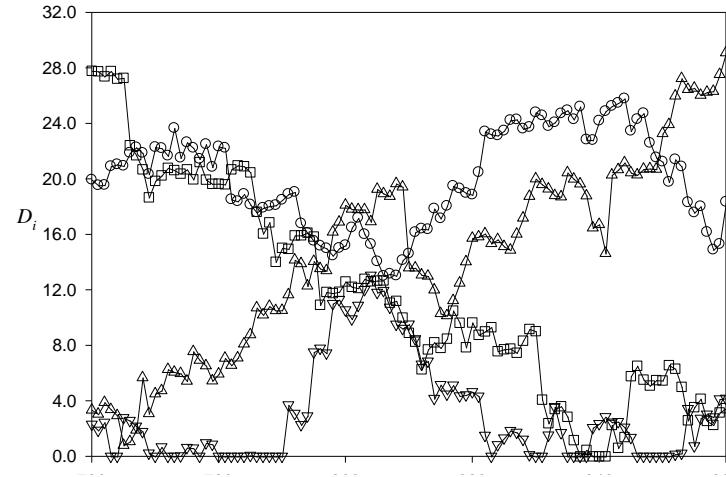
- Um processo estocástico  $\{X(t), t \in T\}$  é um conjunto de variáveis aleatórias: para cada  $t \in T$ ,  $X(t)$  é uma variável aleatória.
- O índice  $t$  é frequentemente interpretado como tempo. Nesta interpretação,  $X(t)$  é o estado do processo no instante  $t$ .
- O conjunto  $T$  é o conjunto de índices do processo.
  - (1) se  $T$  é um conjunto contável, designa-se o processo estocástico como sendo em tempo discreto
  - (2) se  $T$  é um intervalo da reta real, designa-se o processo estocástico como sendo em tempo contínuo
- O espaço de estados é o conjunto de todos os valores que as variáveis aleatórias  $X(t)$  podem tomar.

# Exemplos de processos estocásticos

Considere um sistema com uma fila de espera e um servidor. A este sistema chegam clientes para serem servidos.

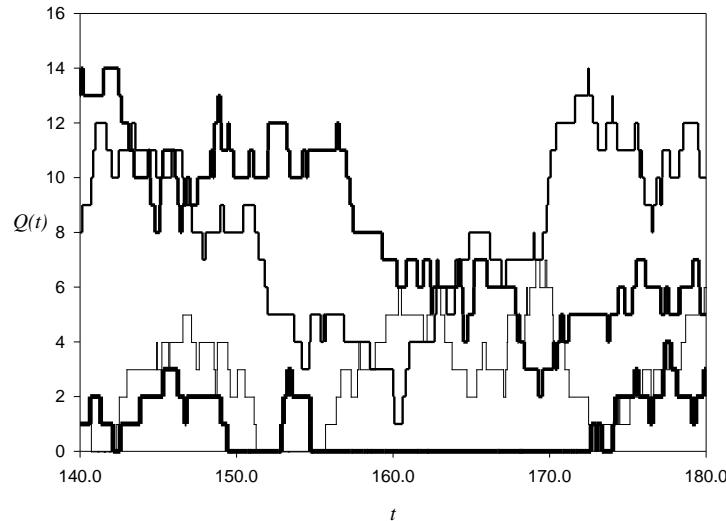
Atrasos sofridos por cada cliente na fila de espera

- (1) é um processo estocástico em tempo discreto ( $1^{\text{o}}$  cliente,  $2^{\text{o}}$  cliente, etc.)
- (2) o estado é uma variável contínua (o tempo de espera é um valor real)



O número de clientes em espera

- (1) é um processo estocástico em tempo contínuo
- (2) o estado é uma variável discreta (0 clientes, 1 cliente, 2 clientes, etc.)





# **Processos de Poisson, Cadeias de Markov em Tempo Contínuo e Sistemas de Filas de Espera**

Desempenho e Dimensionamento de Redes

Prof. Amaro de Sousa ([asou@ua.pt](mailto:asou@ua.pt))

DETI-UA, 2020/2021

# Processo de contagem

- Um processo estocástico  $\{N(t), t \geq 0\}$  diz-se um processo de contagem se  $N(t)$  representar o número total de eventos que ocorreram até ao instante  $t$ .
- Um processo de contagem satisfaz as seguintes condições:
  - (1)  $N(t) \geq 0$ .
  - (2)  $N(t)$  toma valores inteiros apenas.
  - (3) Se  $s < t$ , então  $N(s) \leq N(t)$ .
  - (4) Se  $s < t$ , então  $N(t) - N(s)$  é igual ao número de eventos ocorridos no intervalo de tempo  $[s, t]$ .
- Um processo de contagem tem incrementos independentes se o número de eventos em intervalos de tempo disjuntos for independente.
- Um processo de contagem tem incrementos estacionários se a distribuição do número de eventos que ocorre em qualquer intervalo de tempo depender apenas do comprimento do intervalo de tempo.

# Processo de Poisson

- Um processo de contagem diz-se um processo de Poisson com taxa  $\lambda$ ,  $\lambda > 0$ , se:
  - (1)  $N(0) = 0$ ;
  - (2) o processo tem incrementos independentes;
  - (3) o número de eventos num intervalo de duração  $t$  tem uma distribuição de Poisson com média  $\lambda t$ . Isto é, para todo  $s$ ,  $t \geq 0$

$$P\{N(s+t) - N(s) = n\} = e^{-\lambda t} \frac{(\lambda t)^n}{n!}$$

- Um processo de Poisson tem incrementos estacionários e média

$$E[N(t)] = \lambda t$$

razão pela qual  $\lambda$  é designada a taxa (i.e., o número médio de eventos por unidade de tempo) do processo de Poisson.

# Propriedades de um processo de Poisson

- **Propriedade 1:** Num processo de Poisson com taxa  $\lambda$  considere-se:
  - $T_1$  o instante do primeiro evento
  - $T_n$ ,  $n > 1$ , o intervalo de tempo entre o  $(n-1)$ -ésimo evento e o  $n$ -ésimo evento
- Então,  $T_n$ ,  $n = 1, 2, \dots$ , são variáveis aleatórias independentes e identicamente distribuídas com distribuição exponencial de média  $1/\lambda$ .
- **Propriedade 2:** Num processo de Poisson  $\{N(t), t \geq 0\}$  com taxa  $\lambda$  considere-se que cada evento é classificado de forma independente em:
  - evento do tipo 1 com probabilidade  $p$
  - evento do tipo 2 com probabilidade  $1 - p$
- Assim,  $\{N_1(t), t \geq 0\}$  e  $\{N_2(t), t \geq 0\}$  são o número de eventos de cada tipo que ocorreram no intervalo  $[0, t]$ .
- Então,  $N_1(t)$  e  $N_2(t)$  são ambos processos de Poisson independentes com taxas  $\lambda p$  e  $\lambda(1-p)$ .

# Propriedades de um processo de Poisson

- **Propriedade 3:** Sejam  $\{N_1(t), t \geq 0\}$  e  $\{N_2(t), t \geq 0\}$  processos de Poisson independentes com taxas  $\lambda_1$  e  $\lambda_2$ ,
- Então, o processo  $N(t) = N_1(t) + N_2(t)$  é também um processo de Poisson com taxa  $\lambda = \lambda_1 + \lambda_2$ .
- **Propriedade 4:** Sabendo-se que num processo de Poisson ocorreram exatamente  $n$  eventos até ao instante  $t$ ,
- Então, os instantes de ocorrência dos eventos são distribuídos independentemente e uniformemente no intervalo  $[0, t]$ . Por esta razão diz-se que num processo de Poisson as chegadas são aleatórias.

# Cadeias de Markov em tempo contínuo

- Considere-se um processo estocástico em tempo contínuo  $\{X(t), t \geq 0\}$  com o espaço de estados definido pelo conjunto dos números inteiros não negativos.
- $X(t)$  é uma cadeia de Markov se para todo o  $s$ ,  $t \geq 0$  e inteiros não-negativos  $i, j$ ,  $x(u)$ ,  $0 \leq u < s$  :

$$P\{X(s+t) = j | X(s) = i, X(u) = x(u), 0 \leq u < s\} = \\ P\{X(s+t) = j | X(s) = i\}$$

- Significa que a distribuição futura  $X(s+t)$  condicionada ao presente  $X(s)$  e ao passado  $X(u)$ ,  $0 \leq u < s$ , depende apenas do presente e é independente do passado (propriedade Markoviana).
- Se  $P\{X(s+t) = j | X(s) = i\}$  for independente de  $s$  então diz-se que a cadeia de Markov em tempo contínuo tem probabilidades de transição estacionárias ou homogéneas:

$$P\{X(s+t) = j | X(s) = i\} = P\{X(t) = j | X(0) = i\}$$

# Cadeias de Markov em tempo contínuo

- Uma cadeia de Markov em tempo contínuo tem como propriedades:
  - (1) Quando o processo entra no estado  $i$ , o tempo de permanência nesse estado, antes de efetuar uma transição para um estado diferente, é exponencialmente distribuído (designamos a média por  $1/q_i$ );
  - (2) Quando o processo deixa o estado  $i$ , entra de seguida no estado  $j$  com uma probabilidade  $P_{ij}$  que satisfaz as seguintes condições

$$P_{ii} = 0 \quad 0 \leq P_{ij} \leq 1 \quad , j \neq i \quad \sum_j P_{ij} = 1$$

NOTA: A propriedade (1) é equivalente a dizer que quando o processo está no estado  $i$ , ele transita para outro estado qualquer a uma taxa  $q_i$ .

- Numa cadeia de Markov em tempo contínuo, o tempo de permanência num estado e o próximo estado visitado são variáveis aleatórias independentes.

# Taxas de transição instantâneas

- Para qualquer par de estados  $i$  e  $j$  seja:

$$q_{ij} = q_i P_{ij}$$

$q_i$  - a taxa à qual o processo faz uma transição quando está no estado  $i$

$P_{ij}$  - a probabilidade que a transição seja para o estado  $j$  quando está no estado  $i$

$q_{ij}$  - a taxa à qual o processo faz uma transição para o estado  $j$  quando está no estado  $i$

- As  $q_{ij}$  designam-se por *taxis de transição instantâneas*. Estas são as grandezas habitualmente representadas nos diagramas de transição de estados.

- Como 
$$q_i = \sum_j q_i P_{ij} = \sum_j q_{ij}$$
 
$$P_{ij} = \frac{q_{ij}}{q_i} = \frac{q_{ij}}{\sum_j q_{ij}}$$

resulta que a especificação das taxas de transição instantâneas determina a cadeia de Markov em tempo contínuo.

## Exemplo 1

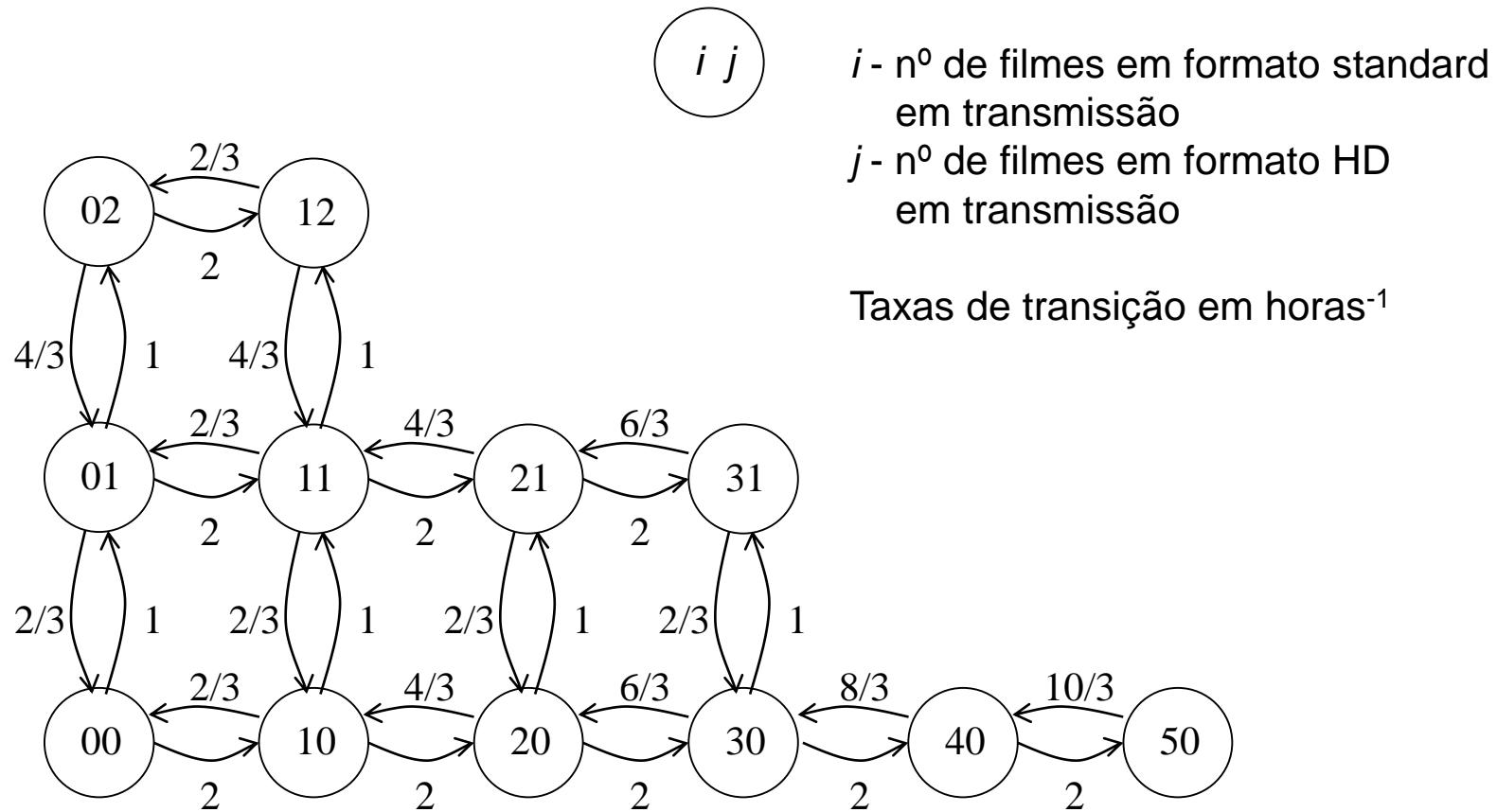
Considere um servidor de *video-streaming* que providencia filmes em formato standard (2.0 Mbps) ou HD (4.0 Mbps) e com uma interface de rede de 10 Mbps.

No período de maior tráfego, a taxa de pedidos de filmes é de 2 filmes/hora em formato standard e 1 filme/hora em formato HD.

Ambos os tipos de filmes têm uma duração exponencialmente distribuída de média 90 minutos. Quando um filme é pedido, ele começa a ser transmitido desde que haja capacidade disponível na interface de rede ou é recusado caso contrário.

Considere o estado do sistema dado pelo número de filmes de cada tipo em transmissão. Qual o diagrama de transição de estados do sistema?

# Diagrama de transição de estados do Exemplo 1



# Probabilidades limite

- Seja  $P_{ij}(t) = P\{X(s+t) = j \mid X(s) = i\}$  a probabilidade de um processo presentemente no estado  $i$  estar no estado  $j$  após um intervalo de tempo  $t$ .
- A probabilidade de uma cadeia de Markov em tempo contínuo estar no estado  $j$  no instante  $t$  converge para um valor limite independente do estado inicial:
$$\pi_j \equiv \lim_{t \rightarrow \infty} P_{ij}(t)$$
- Condição suficiente para a existência de probabilidades limite:
  - (1) a cadeia é irredutível, isto é, começando no estado  $i$  existe uma probabilidade positiva de alguma vez se estar no estado  $j$ , para todo o par de estados  $i, j$
  - (2) a cadeia de Markov é recorrente positiva, isto é, começando em qualquer estado o tempo médio para voltar a esse estado é finito

# Cálculo das probabilidades limite

- As probabilidades limite podem calcular-se resolvendo as equações:

$$q_j \pi_j = \sum_{k \neq j} q_{kj} \pi_k , \quad \text{para todos os estados } j$$

$$\sum_j \pi_j = 1$$

- Estas equações são designadas por equações de balanço:

taxa à qual o sistema transita do estado  $j$

=

taxa à qual o sistema transita para o estado  $j$

- A probabilidade  $\pi_j$  pode ser interpretada como a proporção de tempo em que o processo está no estado  $j$ .
- As probabilidades  $\pi_j$  são designadas por probabilidades estacionárias: se o estado inicial for dado pela distribuição  $\{\pi_j\}$ , então a probabilidade de se estar no estado  $j$  no instante  $t$  é  $\pi_j$ , para todo o  $t$ .

## Exemplo 1

Equações de balanço:

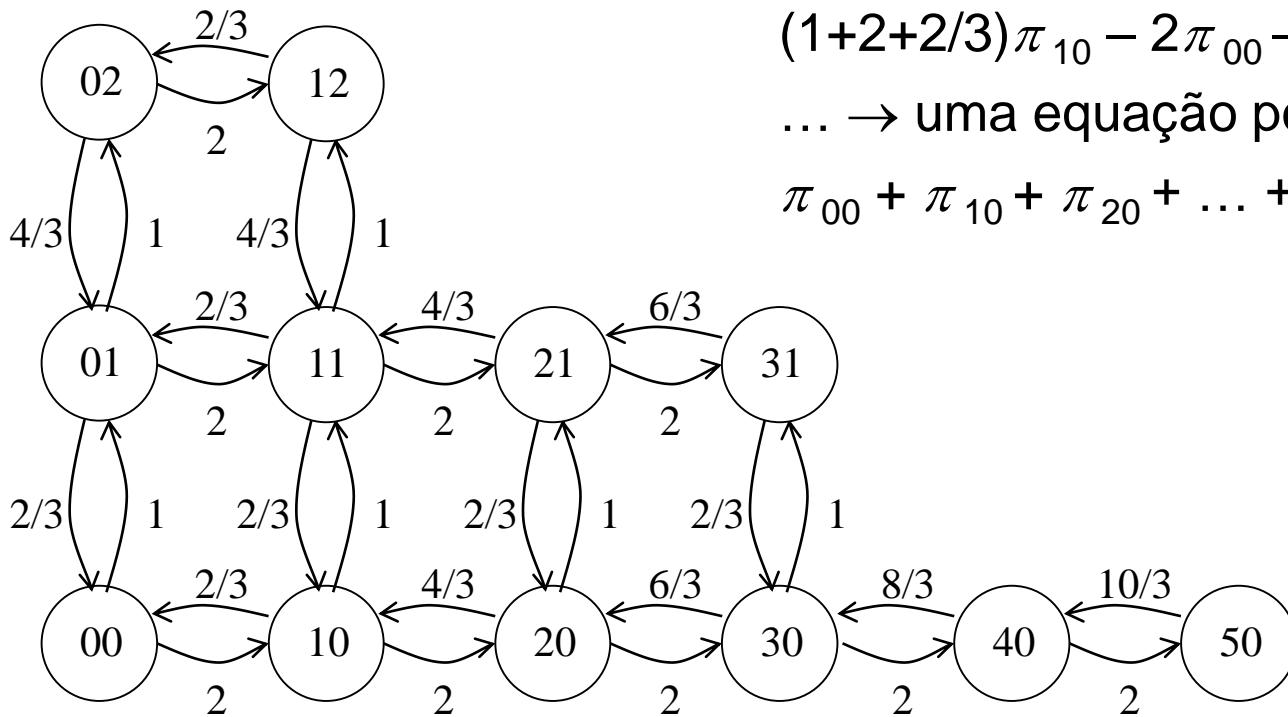
$$q_j \pi_j = \sum_{k \neq j} q_{kj} \pi_k \quad \sum_j \pi_j = 1$$

$$(1+2)\pi_{00} - 2/3\pi_{10} - 2/3\pi_{01} = 0$$

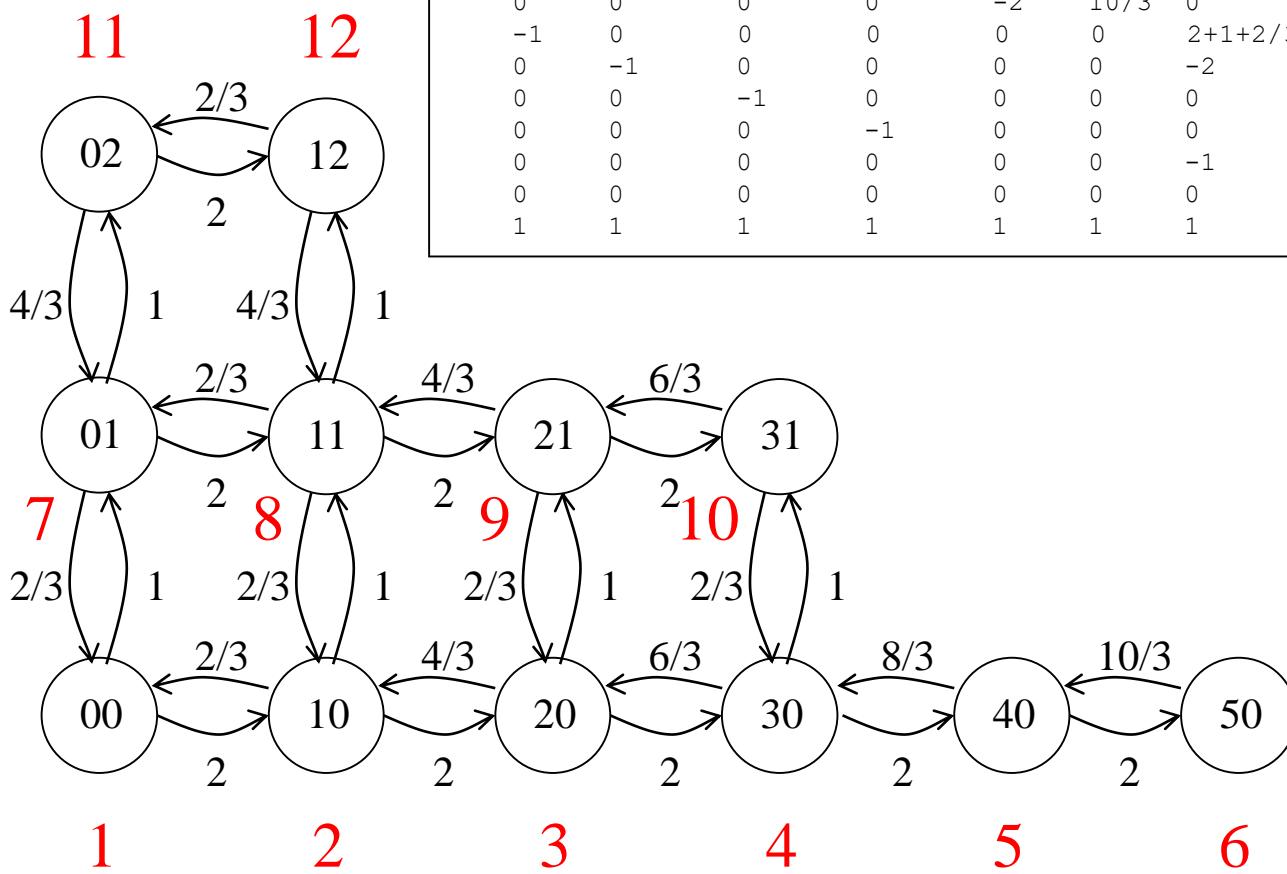
$$(1+2+2/3)\pi_{10} - 2\pi_{00} - 4/3\pi_{20} - 2/3\pi_{11} = 0$$

... → uma equação por cada estado

$$\pi_{00} + \pi_{10} + \pi_{20} + \dots + \pi_{02} + \pi_{12} = 1$$



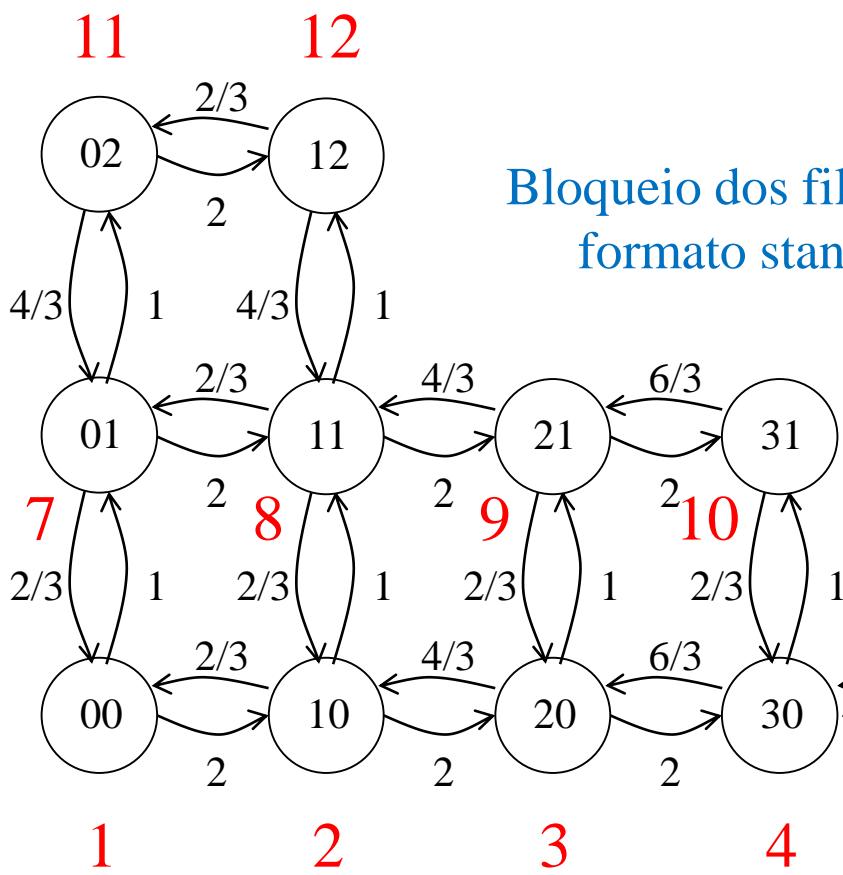
# Exemplo 1 – resolução no MATLAB



$$A = \begin{bmatrix} 2+1 & -2/3 & 0 & 0 & 0 & 0 & -2/3 & 0 & 0 & 0 & 0 & 0 \\ -2 & 2+1+2/3 & -4/3 & 0 & 0 & 0 & 0 & -2/3 & 0 & 0 & 0 & 0 \\ 0 & -2 & 2+1+4/3 & -6/3 & 0 & 0 & 0 & 0 & -2/3 & 0 & 0 & 0 \\ 0 & 0 & -2 & 2+1+6/3 & -8/3 & 0 & 0 & 0 & 0 & -2/3 & 0 & 0 \\ 0 & 0 & 0 & -2 & 2+8/3 & -10/3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2 & 10/3 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 2+1+2/3 & -2/3 & 0 & 0 & -4/3 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & -2 & 2+1+4/3 & -4/3 & 0 & 0 & -4/3 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & -2 & 2+6/3 & -6/3 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -2 & 8/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 2+4/3 & -2/3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -2 & 6/3 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

# Exemplo 1 – resolução no MATLAB



Bloqueio dos filmes em formato standard

Bloqueio dos filmes em formato HD

```
>> x=A\B
```

```
x =
0.0236
0.0708
0.1061
0.1061
0.0796
0.0478
0.0354
0.1061
0.1592
0.1592
0.0265
0.0796
```

```
>> x(6)+x(10)+x(12)
```

```
ans =
0.2866
```

```
>> x(5)+x(6)+x(9)+x(10)+x(11)+x(12)
```

```
ans =
0.5519
```

## Definições do teorema de Little

- Admita-se que se observa um sistema desde o instante  $t = 0$ . Seja:  
 $L(t)$  - o número de clientes no sistema no instante  $t$ ,  
 $N(t)$  - o número de clientes que chegaram no intervalo  $[0, t]$ ,  
 $W_i$  - o tempo despendido no sistema pelo  $i$ -ésimo cliente.
- Média temporal do número de clientes observados até ao instante  $t$ :  
$$L_t = \frac{1}{t} \int_0^t L(\tau) d\tau \quad L = \lim_{t \rightarrow \infty} L_t$$
- Média temporal da taxa de chegada no intervalo  $[0, t]$ :

$$\lambda_t = N(t)/t \quad \lambda = \lim_{t \rightarrow \infty} \lambda_t$$

- Média temporal do atraso dos clientes até ao instante  $t$ :

$$W_t = \frac{\sum_{i=0}^{N(t)} W_i}{N(t)} \quad W = \lim_{t \rightarrow \infty} W_t$$

## Teorema de Little

- O teorema de Little enuncia que

$$L = \lambda W$$

- O teorema de Little traduz a ideia intuitiva de que, para a mesma taxa de chegada de clientes, sistemas mais congestionados ( $L$  elevado) impõem maiores atrasos ( $W$  elevado).
- Num dia de chuva, o mesmo tráfego (mesmo  $\lambda$ ) é mais lento do que normalmente ( $W$  maior) e as ruas estão mais congestionadas ( $L$  maior).
- Um restaurante de refeições rápidas ( $W$  menor) precisa de uma sala menor ( $L$  menor) que um restaurante normal, para a mesma taxa de chegada de clientes (mesmo  $\lambda$ ).

## Propriedade PASTA

- Considere um sistema em que os clientes chegam um de cada vez e são servidos um de cada vez.
- Seja  $L(t)$  o número de clientes no sistema no instante  $t$  e defina-se  $P_n$ ,  $n \geq 0$ , como

$$P_n = \lim_{t \rightarrow \infty} P\{L(t) = n\}$$

$P_n$  é a probabilidade em estado estacionário de existirem exatamente  $n$  clientes no sistema (ou a proporção de tempo em que o sistema contém exatamente  $n$  clientes).

- Considere  $a_n$  a proporção de clientes que ao chegar encontram  $n$  clientes no sistema.
- Considere  $d_n$  a proporção de clientes que ao partir deixam  $n$  clientes no sistema.
- Em qualquer sistema em que os clientes chegam um de cada vez e são servidos um de cada vez verifica-se que

$$a_n = d_n$$

# Propriedade PASTA

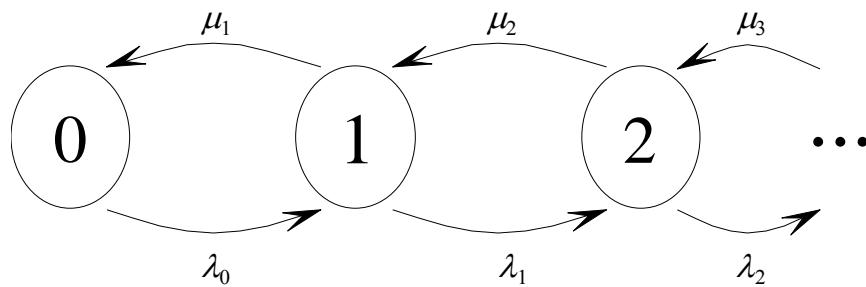
- Propriedade PASTA (*Poisson Arrivals always See Time Averages*):  
As chegadas de Poisson em que o tempo de serviço é estatisticamente independente dos instantes de chegada, vêm sempre médias temporais:

$$a_n = P_n$$

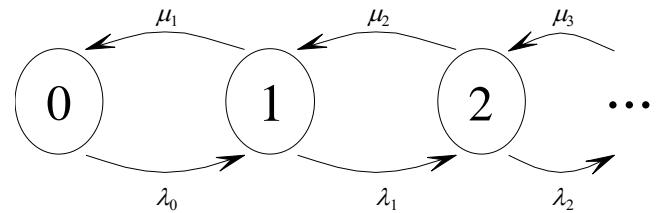
- Contra exemplos:
- Considere que o intervalo entre chegadas é uniformemente distribuído entre 2 e 6 segundos e os tempos de serviço dos clientes são uniformemente distribuídos entre 1 e 2 segundos.
  - As chegadas não são de Poisson
- Considere um sistema em que o processo de chegada de clientes é um processo de Poisson e que o tempo de serviço do  $n$ -ésimo cliente é igual a metade do intervalo entre a chegada do  $n$ -ésimo e do  $(n+1)$ -ésimo cliente.
  - O tempo de serviço não é independente das chegadas

# Processos de nascimento e morte

- Considere um sistema cujo estado representa o número de clientes no sistema.
- Sempre que o sistema tem  $n$  clientes:
  - (1) chegam novos clientes ao sistema a uma taxa exponencial  $\lambda_n$
  - (2) partem clientes do sistema a uma taxa exponencial  $\mu_n$
- Este sistema é designado por processo de nascimento e morte.
- Os parâmetros  $\lambda_n$  ( $n = 0, 1, \dots$ ) e  $\mu_n$  ( $n = 1, 2, \dots$ ) são designados por taxas de chegada (ou de nascimento) e taxas de partida (ou de morte), respetivamente.



# Equações de balanço de processos de nascimento e morte



*Estado*      *taxa de saída = taxa de entrada*

$$0 \quad \lambda_0\pi_0 = \mu_1\pi_1$$

$$1 \quad (\lambda_1 + \mu_1)\pi_1 = \mu_2\pi_2 + \lambda_0\pi_0$$

$$2 \quad (\lambda_2 + \mu_2)\pi_2 = \mu_3\pi_3 + \lambda_1\pi_1$$

$$n, n \geq 1 \quad (\lambda_n + \mu_n)\pi_n = \mu_{n+1}\pi_{n+1} + \lambda_{n-1}\pi_{n-1}$$

Ou, de forma equivalente (por manipulação das equações anteriores):

$$\lambda_n\pi_n = \mu_{n+1}\pi_{n+1}, \quad n \geq 0$$

# Probabilidades limite de processos de nascimento e morte

$$\pi_0 = \frac{1}{1 + \sum_{i=1}^{\infty} \frac{\lambda_0 \lambda_1 \cdots \lambda_{i-1}}{\mu_1 \mu_2 \cdots \mu_i}}$$

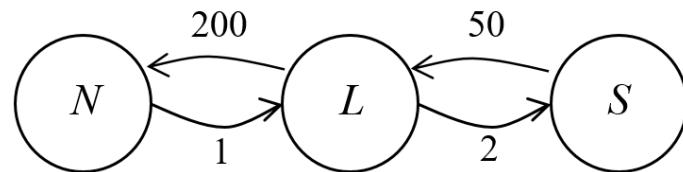
$$\pi_n = \frac{\lambda_0 \lambda_1 \cdots \lambda_{n-1}}{\mu_1 \mu_2 \cdots \mu_n \left( 1 + \sum_{i=1}^{\infty} \frac{\lambda_0 \lambda_1 \cdots \lambda_{i-1}}{\mu_1 \mu_2 \cdots \mu_i} \right)} = \frac{\lambda_0 \lambda_1 \cdots \lambda_{n-1}}{\mu_1 \mu_2 \cdots \mu_n} \cdot \pi_0, \quad n \geq 1$$

Condição necessária para a existência de probabilidades limite:

$$\sum_{i=1}^{\infty} \frac{\lambda_0 \lambda_1 \cdots \lambda_{i-1}}{\mu_1 \mu_2 \cdots \mu_i} < \infty$$

## Exemplo 2

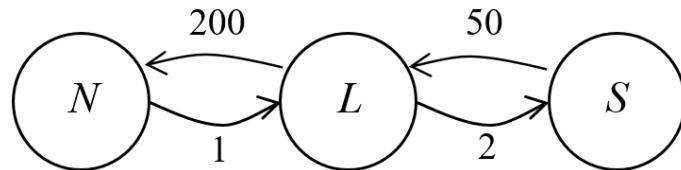
Considere que uma ligação sem fios para comunicação de pacotes pode estar num de 3 estados possíveis – Normal ( $N$ ), Interferência Ligeira ( $L$ ) ou Interferência Severa ( $S$ ) – de acordo com a cadeia de Markov seguinte (taxas em número de transições por hora):



- Determine a probabilidade de cada um dos estados.
- Determine o tempo de permanência médio de cada estado (em minutos).
- Sabendo que a probabilidade de cada pacote ser recebido com erros é de 0.01% no estado  $N$ , 0.1% no estado  $L$  e 1% no estado  $S$ , qual a probabilidade da ligação estar no estado  $N$  quando um pacote é recebido com erros?

## Exemplo 2 – Resolução (a)

Considere que uma ligação sem fios para comunicação de pacotes pode estar num de 3 estados possíveis – Normal ( $N$ ), Interferência Ligeira ( $L$ ) ou Interferência Severa ( $S$ ) – de acordo com a cadeia de Markov seguinte (taxas em número de transições por hora):



(a) Determine a probabilidade de cada um dos estados.

$$P_N = \frac{1}{1 + \frac{1}{200} + \frac{1}{200} \times \frac{2}{50}} = 0.99483 = 99.483\%$$

$$\pi_0 = \frac{1}{1 + \sum_{i=1}^{\infty} \frac{\lambda_0 \lambda_1 \dots \lambda_{i-1}}{\mu_1 \mu_2 \dots \mu_i}}$$

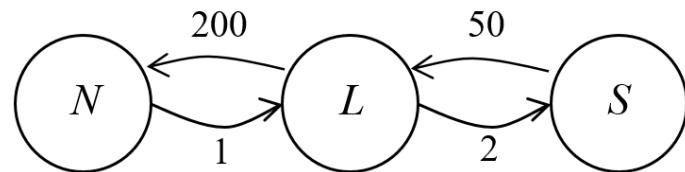
$$P_L = \frac{\frac{1}{200}}{1 + \frac{1}{200} + \frac{1}{200} \times \frac{2}{50}} = 0.00497 = 0.497\%$$

$$\pi_n = \frac{\lambda_0 \lambda_1 \dots \lambda_{n-1}}{\mu_1 \mu_2 \dots \mu_n} \cdot \pi_0$$

$$P_S = \frac{\frac{1}{200} \times \frac{2}{50}}{1 + \frac{1}{200} + \frac{1}{200} \times \frac{2}{50}} = 0.0002 = 0.02\%$$

## Exemplo 2 – Resolução (b)

Considere que uma ligação sem fios para comunicação de pacotes pode estar num de 3 estados possíveis – Normal ( $N$ ), Interferência Ligeira ( $L$ ) ou Interferência Severa ( $S$ ) – de acordo com a cadeia de Markov seguinte (taxas em número de transições por hora):



(b) Determine o tempo de permanência médio de cada estado (em minutos).

$$T_N = \frac{1}{1} = 1 \text{ hora} = 60 \text{ minutos}$$

Tempo médio de permanência  $T = 1/q_i$

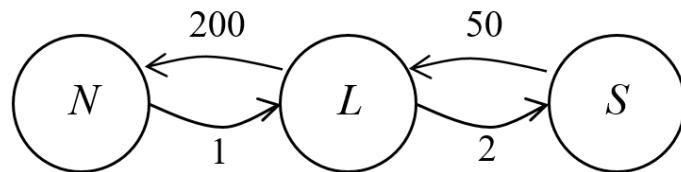
$$T_L = \frac{1}{2 + 200} = 0.00495 \text{ horas} = 0.3 \text{ minutos}$$

$$q_i = \sum_j q_i P_{ij} = \sum_j q_{ij}$$

$$T_S = \frac{1}{50} = 0.02 \text{ horas} = 1.2 \text{ minutos}$$

## Exemplo 2 – Resolução (c)

Considere que uma ligação sem fios para comunicação de pacotes pode estar num de 3 estados possíveis – Normal ( $N$ ), Interferência Ligeira ( $L$ ) ou Interferência Severa ( $S$ ) – de acordo com a cadeia de Markov seguinte (taxas em número de transições por hora):

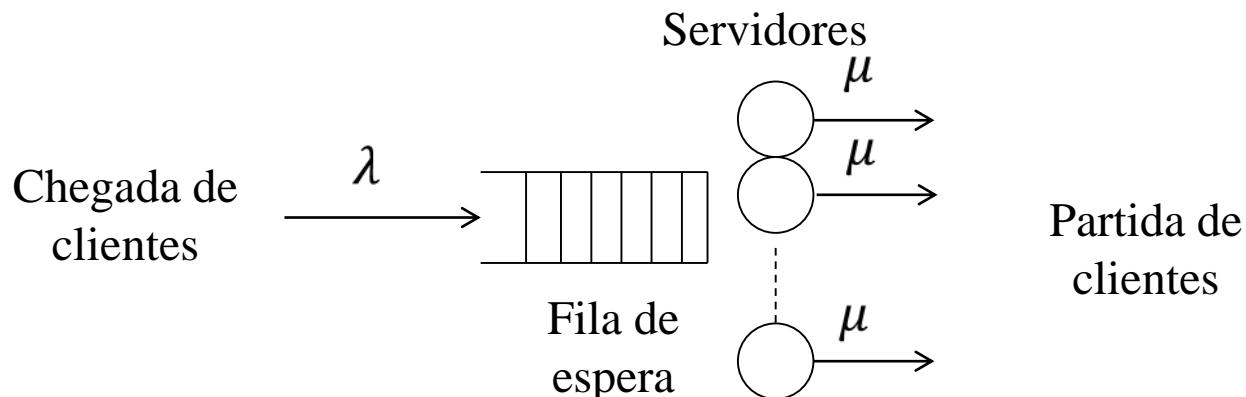


(c) Sabendo que a probabilidade de cada pacote ser recebido com erros é de 0.01% no estado  $N$ , 0.1% no estado  $L$  e 1% no estado  $S$ , qual a probabilidade da ligação estar no estado  $N$  quando um pacote é recebido com erros?

$$\begin{aligned} P(N|E) &= \frac{P(E|N) \times P(N)}{P(E|N) \times P(N) + P(E|L) \times P(L) + P(E|S) \times P(S)} \\ &= \frac{0.01\% \times 0.99483}{0.01\% \times 0.99483 + 0.1\% \times 0.00497 + 1\% \times 0.00020} \\ &= 0.9346 = 93.46\% \end{aligned}$$

# Sistema de fila de espera

- Um sistema de fila de espera é caracterizado por:
  - um conjunto de  $c$  servidores, cada um com capacidade para servir clientes a uma taxa  $\mu$
  - uma fila de espera com uma determinada capacidade (em nº de clientes)
- A este sistema chegam clientes a uma taxa  $\lambda$
- Quando um cliente chega:
  - ele começa a ser servido por um servidor se houver algum disponível
  - ele é colocado da fila de espera se os servidores estiverem todos ocupados (ou é perdido se a fila de espera estiver cheia)
- Os clientes na fila de espera são atendidos segundo uma disciplina *First-In-First-Out*



# Sistema de fila de espera

- Um sistema de fila de espera é representado por:

$$A/B/c/d$$

*A* – o processo de chegada de clientes:

*M* – Markoviano, *D* – Determinístico, *G* – Genérico

*B* – o processo de atendimento de clientes:

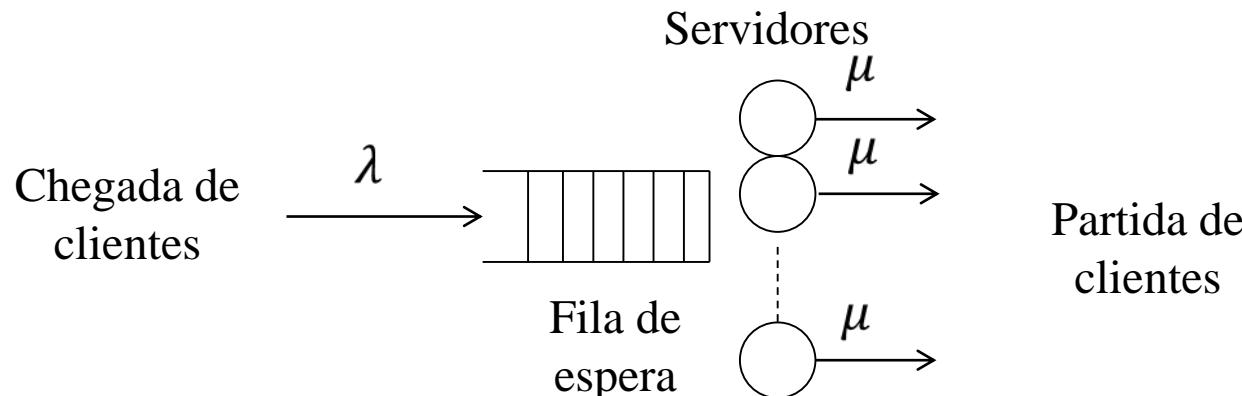
*M* – Markoviano, *D* – Determinístico, *G* – Genérico

*c* – o número de servidores

*d* – capacidade do sistema em nº de clientes:

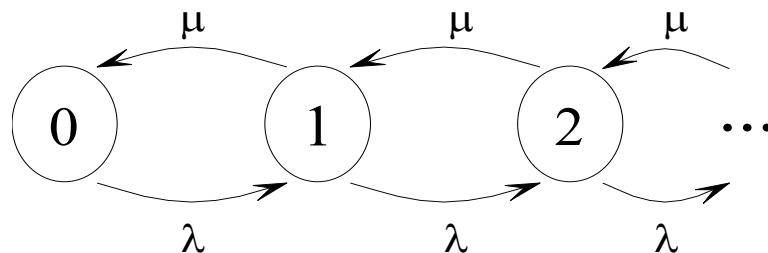
número de servidores + capacidade da fila de espera

- Quando *d* é omissio, a fila de espera tem tamanho infinito.



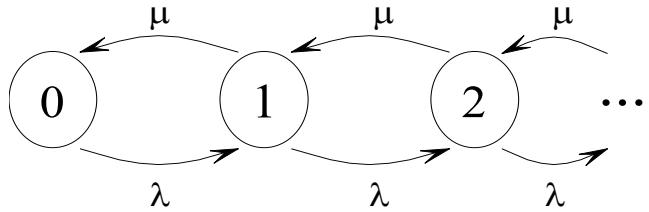
## Sistema $M/M/1$

- Processo de nascimento e morte em que:
  - (1) a chegada de clientes é um processo de Poisson com taxa  $\lambda$
  - (2) o tempo de atendimento de um servidor é exponencialmente distribuído com média  $1/\mu$
  - (3) o sistema tem 1 servidor
  - (4) o sistema acomoda um número infinito de clientes



- Uma ligação ponto-a-ponto com capacidade  $\mu$  pacotes/s e uma fila de espera muito grande onde chegam pacotes a uma taxa de Poisson  $\lambda$  pacotes/s com comprimento exponencialmente distribuído de média  $1/\mu$  é um sistema  $M/M/1$

# Sistema $M/M/1$



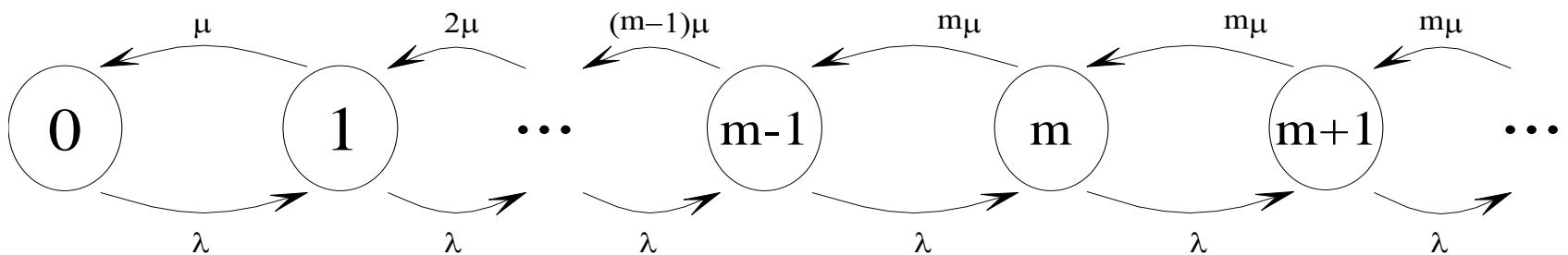
$$P_0 = \frac{1}{1 + \sum_{i=1}^{\infty} \frac{\lambda_0 \lambda_1 \cdots \lambda_{i-1}}{\mu_1 \mu_2 \cdots \mu_i}} = \frac{1}{1 + \sum_{i=1}^{\infty} \left(\frac{\lambda}{\mu}\right)^i}$$

$$P_n = \frac{\lambda_0 \lambda_1 \cdots \lambda_{n-1}}{\mu_1 \mu_2 \cdots \mu_n} \cdot P_0 = \left(\frac{\lambda}{\mu}\right)^n \cdot P_0 = \frac{\left(\frac{\lambda}{\mu}\right)^n}{1 + \sum_{i=1}^{\infty} \left(\frac{\lambda}{\mu}\right)^i}$$

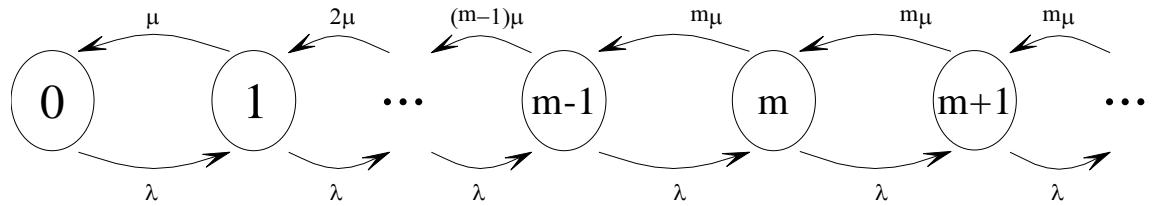
- Número médio de clientes no sistema:  $L = \sum_{n=0}^{\infty} nP_n = \frac{\lambda}{\mu - \lambda}$
- Atraso médio no sistema:  $W = \frac{L}{\lambda} = \frac{1}{\mu - \lambda}$
- Atraso médio na fila de espera:  $W_Q = W - \frac{1}{\mu} = \frac{\lambda}{\mu(\mu - \lambda)}$
- Número médio de clientes na fila de espera:  $L_Q = \lambda W_Q = \frac{\lambda^2}{\mu(\mu - \lambda)}$

## Sistema $M/M/m$

- Processo de nascimento e morte em que:
  - (1) a chegada de clientes é um processo de Poisson com taxa  $\lambda$
  - (2) o tempo de atendimento de um servidor é exponencialmente distribuído com média  $1/\mu$
  - (3) o sistema tem  $m$  servidores
  - (4) o sistema acomoda um número infinito de clientes



## Sistema $M/M/m$



Equações de balanço:

$$\lambda P_{n-1} = n\mu P_n, \quad n \leq m$$

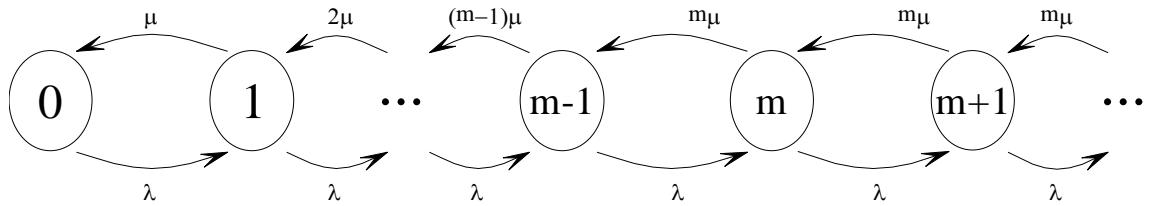
$$\lambda P_{n-1} = m\mu P_n, \quad n > m$$

Probabilidade de  $n$  clientes no sistema em estado estacionário ( $\rho = \lambda/m\mu < 1$ ):

$$P_0 = \frac{1}{\sum_{n=0}^{m-1} \frac{(m\rho)^n}{n!} + \frac{(m\rho)^m}{m!(1-\rho)}}$$

$$P_n = \begin{cases} P_0 \frac{(m\rho)^n}{n!}, & n \leq m \\ P_0 \frac{m^m \rho^n}{m!}, & n > m \end{cases}, n \geq 1$$

## Sistema $M/M/m$



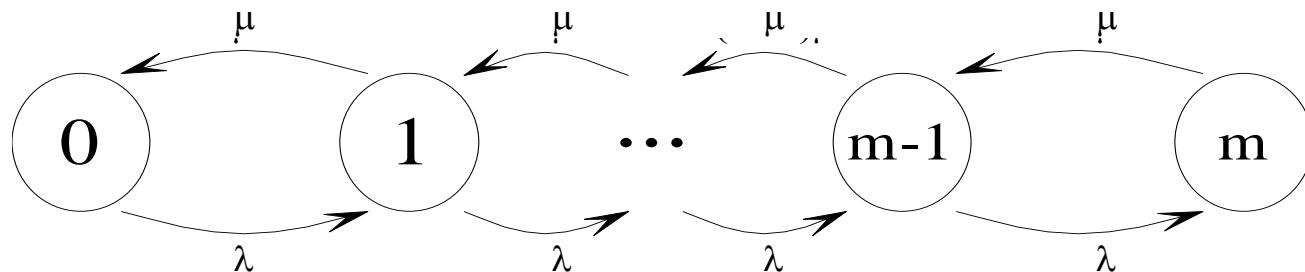
- Probabilidade de uma chegada encontrar todos os servidores ocupados (fórmula de Erlang C):

$$P_Q = \sum_{n=m}^{\infty} p_n = \frac{P_0(m\rho)^m}{m!(1-\rho)}$$

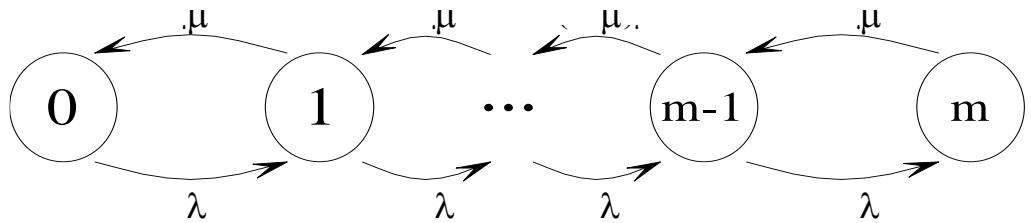
- Número médio de clientes na fila de espera:  $L_Q = \sum_{n=0}^{\infty} n P_0 \frac{m^m \rho^{m+n}}{m!} = P_Q \frac{\rho}{1-\rho}$
- Atraso médio na fila de espera:  $W_Q = \frac{L_Q}{\lambda} = P_Q \frac{\rho}{\lambda(1-\rho)}$
- Atraso médio no sistema:  $W = \frac{1}{\mu} + W_Q = \frac{1}{\mu} + \frac{P_Q}{m\mu - \lambda}$
- Número médio de clientes no sistema:  $L = \lambda W = m\rho + \frac{\rho P_Q}{1-\rho}$

## Sistema $M/M/1/m$

- Processo de nascimento e morte em que:
  - (1) a chegada de clientes é um processo de Poisson com taxa  $\lambda$
  - (2) o tempo de atendimento de um servidor é exponencialmente distribuído com média  $1/\mu$
  - (3) o sistema tem 1 servidor
  - (4) o sistema acomoda no máximo  $m$  clientes (*i.e.*, a fila de espera tem capacidade para  $m - 1$  clientes)



## Sistema $M/M/1/m$



- Equações de balanço:

$$\lambda P_{n-1} = \mu P_n, \quad n = 1, 2, \dots, m$$

- Probabilidade de  $n$  clientes no sistema em estado estacionário:

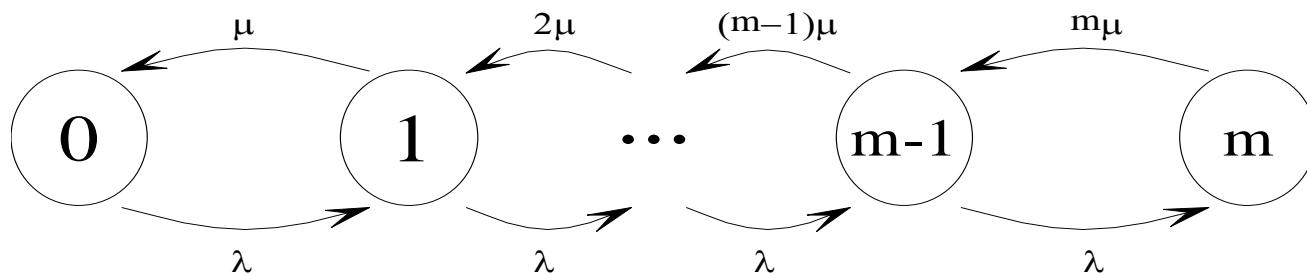
$$P_n = \frac{(\lambda/\mu)^n}{\sum_{i=0}^m (\lambda/\mu)^i} \quad n = 0, 1, \dots, m$$

- Pela propriedade PASTA, a probabilidade de uma chegada encontrar o sistema cheio (*i.e.*, o servidor ocupado e a fila de espera cheia):

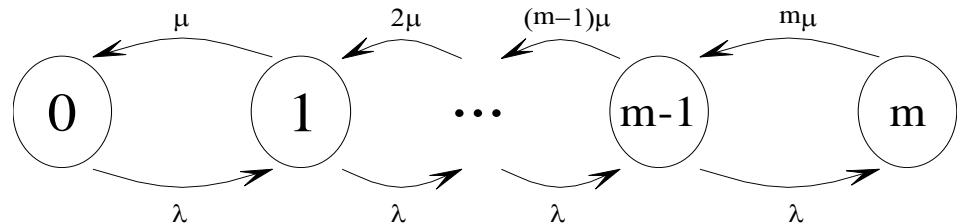
$$P_m = \frac{(\lambda/\mu)^m}{\sum_{i=0}^m (\lambda/\mu)^i}$$

## Sistema $M/M/m/m$

- Processo de nascimento e morte em que:
  - (1) a chegada de clientes é um processo de Poisson com taxa  $\lambda$
  - (2) o tempo de atendimento de um servidor é exponencialmente distribuído com média  $1/\mu$
  - (3) o sistema tem  $m$  servidores
  - (4) o sistema acomoda no máximo  $m$  clientes (*i.e.*, não tem fila de espera)



## Sistema $M/M/m/m$



- Equações de balanço:

$$\lambda P_{n-1} = n\mu P_n, \quad n = 1, 2, \dots, m$$

- Probabilidade de  $n$  clientes no sistema em estado estacionário:

$$P_n = \frac{(\lambda/\mu)^n / n!}{\sum_{i=0}^m (\lambda/\mu)^i / i!} \quad n = 0, 1, \dots, m$$

- Pela propriedade PASTA, a probabilidade de uma chegada encontrar o sistema cheio é (fórmula de Erlang B):

$$P_m = \frac{(\lambda/\mu)^m / m!}{\sum_{i=0}^m (\lambda/\mu)^i / i!}$$

## Sistema $M/G/1$

- Processo de nascimento e morte em que:
  - (1) a chegada de clientes é um processo de Poisson com taxa  $\lambda$
  - (2) o tempo de atendimento  $S$  do servidor tem uma distribuição genérica e independente das chegadas dos clientes
  - (3) o sistema tem 1 servidor
  - (4) o sistema acomoda um número infinito de clientes
- Sendo conhecidos  $E[S]$  e  $E[S^2]$  do tempo de atendimento  $S$ , a fórmula de Pollaczek - Khintchine enuncia que o atraso médio na fila de espera é dado por:

$$W_Q = \frac{\lambda E[S^2]}{2(1 - \lambda E[S])}$$

- O atraso médio no sistema é dado por:

$$W = \frac{\lambda E[S^2]}{2(1 - \lambda E[S])} + E[S]$$

## Sistema $M/G/1$

$$W_Q = \frac{\lambda E[S^2]}{2(1 - \lambda E[S])}$$

- Quando o tempo de serviço é exponencialmente distribuído, o sistema resulta num  $M/M/1$ :

$$E[S] = 1/\mu$$

$$E[S^2] = Var[S] + (E[S])^2 = 1/\mu^2 + 1/\mu^2 = 2/\mu^2$$

$$W_Q = \frac{\lambda}{\mu(\mu - \lambda)}$$

- Quando os tempos de serviço são iguais para todos os clientes com valor  $1/\mu$ , o sistema resulta num  $M/D/1$ :

$$E[S] = 1/\mu$$

$$E[S^2] = 1/\mu^2$$

$$W_Q = \frac{\lambda}{2\mu(\mu - \lambda)}$$

- Uma ligação ponto-a-ponto com capacidade  $\mu$  pacotes/s e uma fila de espera muito grande onde chegam pacotes a uma taxa de Poisson  $\lambda$  pacotes/s é um sistema  $M/G/1$ .

- Se o comprimento dos pacotes for exponencialmente distribuído, resulta num sistema  $M/M/1$ .
- Se o comprimento dos pacotes for fixo, resulta num sistema  $M/D/1$ .

## Exemplo 3

Considere um sistema de transmissão de pacotes com uma fila de espera seguida de uma linha de transmissão de 64 Kbps. O tráfego oferecido é de Poisson com taxa de 15 pacotes/segundo. O comprimento dos pacotes é exponencialmente distribuído com média de 400 bytes. A fila de espera tem capacidade para 3 pacotes. Um pacote que chegue ao sistema é encaminhado pela linha de transmissão, se estiver livre, ou posto em fila de espera. Se a fila de espera se encontrar cheia, o pacote é perdido. Calcule:

- (a) A percentagem de pacotes perdidos.
- (b) A percentagem de pacotes que não sofre atraso na fila de espera.
- (c) A percentagem de utilização da linha de transmissão.

## Exemplo 3 – Resolução (a)

Considere um sistema de transmissão de pacotes com uma fila de espera seguida de uma linha de transmissão de 64 Kbps. O tráfego oferecido é de Poisson com taxa de 15 pacotes/segundo. O comprimento dos pacotes é exponencialmente distribuído com média de 400 bytes. A fila de espera tem capacidade para 3 pacotes. Um pacote que chegue ao sistema é encaminhado pela linha de transmissão, se estiver livre, ou posto em fila de espera. Se a fila de espera se encontrar cheia, o pacote é perdido. Calcule:

(a) A percentagem de pacotes perdidos.

$$\mu = \frac{64000 \text{ bps}}{400 \times 8 \text{ bpp}} = 20 \text{ pps}$$

$$P_n = \frac{(\lambda/\mu)^n}{\sum_{i=0}^m (\lambda/\mu)^i} \quad n = 0, 1, \dots, m$$

$$P_4 = \frac{\left(\frac{\lambda}{\mu}\right)^4}{\left(\frac{\lambda}{\mu}\right)^0 + \left(\frac{\lambda}{\mu}\right)^1 + \left(\frac{\lambda}{\mu}\right)^2 + \left(\frac{\lambda}{\mu}\right)^3 + \left(\frac{\lambda}{\mu}\right)^4}$$

$$P_4 = \frac{\left(\frac{15}{20}\right)^4}{1 + \frac{15}{20} + \left(\frac{15}{20}\right)^2 + \left(\frac{15}{20}\right)^3 + \left(\frac{15}{20}\right)^4} = 0.104 = 10.4\%$$

## Exemplo 3 – Resolução (b)

Considere um sistema de transmissão de pacotes com uma fila de espera seguida de uma linha de transmissão de 64 Kbps. O tráfego oferecido é de Poisson com taxa de 15 pacotes/segundo. O comprimento dos pacotes é exponencialmente distribuído com média de 400 bytes. A fila de espera tem capacidade para 3 pacotes. Um pacote que chegue ao sistema é encaminhado pela linha de transmissão, se estiver livre, ou posto em fila de espera. Se a fila de espera se encontrar cheia, o pacote é perdido. Calcule:

(b) A percentagem de pacotes que não sofre atraso na fila de espera.

$$\mu = \frac{64000 \text{ bps}}{400 \times 8 \text{ bpp}} = 20 \text{ pps}$$

$$P_n = \frac{(\lambda/\mu)^n}{\sum_{i=0}^m (\lambda/\mu)^i} \quad n = 0, 1, \dots, m$$

$$P_0 = \frac{\left(\frac{\lambda}{\mu}\right)^0}{\left(\frac{\lambda}{\mu}\right)^0 + \left(\frac{\lambda}{\mu}\right)^1 + \left(\frac{\lambda}{\mu}\right)^2 + \left(\frac{\lambda}{\mu}\right)^3 + \left(\frac{\lambda}{\mu}\right)^4}$$

$$P_0 = \frac{1}{1 + \frac{15}{20} + \left(\frac{15}{20}\right)^2 + \left(\frac{15}{20}\right)^3 + \left(\frac{15}{20}\right)^4} = 0.328 = 32.8\%$$

## Exemplo 3 – Resolução (c)

Considere um sistema de transmissão de pacotes com uma fila de espera seguida de uma linha de transmissão de 64 Kbps. O tráfego oferecido é de Poisson com taxa de 15 pacotes/segundo. O comprimento dos pacotes é exponencialmente distribuído com média de 400 bytes. A fila de espera tem capacidade para 3 pacotes. Um pacote que chegue ao sistema é encaminhado pela linha de transmissão, se estiver livre, ou posto em fila de espera. Se a fila de espera se encontrar cheia, o pacote é perdido. Calcule:

(c) A percentagem de utilização da linha de transmissão.

$$U = 0 \times P_0 + 1 \times P_1 + 1 \times P_2 + 1 \times P_3 + 1 \times P_4$$

$$U = P_1 + P_2 + P_3 + P_4 = 1 - P_0$$

$$U = 1 - 0.328 = 0.672 = 67.2\%$$

## Exemplo 4

Considere um sistema de transmissão ponto-a-ponto de 128 kbps que suporta dois fluxos de pacotes: no fluxo 1, os pacotes têm um tamanho constante de 128 Bytes e a chegada de pacotes é um processo de Poisson com taxa de 30 pacotes/segundo; no fluxo 2, os pacotes têm um tamanho constante de 512 Bytes e a chegada de pacotes é um processo de Poisson com taxa de 10 pacotes/segundo. Os fluxos são multiplexados estatisticamente numa fila de espera muito grande.

- (a) Indique justificando que tipo de fila de espera modela o desempenho deste sistema de transmissão.
- (b) Calcule o atraso médio no sistema dos pacotes de cada fluxo.

## Exercício 4 – Resolução (a)

Este sistema é modelado por uma fila de espera  $M/G/1$ :

- a soma de 2 processos de Poisson é um processo de Poisson e, assim, o processo de chegada de pacotes é um processo de Poisson (' $M$ ' em  $M/G/1$ ) com taxa  $30 + 10 = 40$  pps;
- o tempo de transmissão dos pacotes é genérico (' $G$ ' em  $M/G/1$ ) porque a distribuição dos tamanhos não segue uma distribuição comum: o tamanho é 128 Bytes com probabilidade  $30/(30+10) = 0.75 = 75\%$  ou 512 Bytes com probabilidade  $10/(30+10) = 0.25 = 25\%$ ;
- o número de servidores é um ('1' em  $M/G/1$ ) pois o canal de transmissão é usado para transmitir um pacote de cada vez;
- como o sistema tem uma fila de espera muito grande, a notação relativa à capacidade do sistema é omissa (considera-se que o sistema tem capacidade infinita).

## Exercício 4 – Resolução (b)

$$S_{128} = (128 \times 8) / 128000 = 8 \times 10^{-3} \text{ seg} \quad S_{512} = (512 \times 8) / 128000 = 32 \times 10^{-3} \text{ seg}$$

$$\begin{aligned} E[S] &= 0.75 \times S_{128} + 0.25 \times S_{512} = \\ &= 0.75 \times 8 \times 10^{-3} + 0.25 \times 32 \times 10^{-3} = 14e-3 \text{ seg} \end{aligned}$$

$$\begin{aligned} E[S^2] &= 0.75 \times (S_{128})^2 + 0.25 \times (S_{512})^2 = \\ &= 0.75 \times (8 \times 10^{-3})^2 + 0.25 \times (32 \times 10^{-3})^2 = 3.04e-4 \text{ seg}^2 \end{aligned}$$

$$W_Q = \frac{\lambda E[S^2]}{2(1 - \lambda E[S])} = \frac{40 \times 3.04e-4}{2(1 - 40 \times 14e-3)} = 0.0143 = 14.3 \text{ mseg}$$

$$W_{128} = W_Q + S_{128} = 14.3 + 8 = 22.3 \text{ mseg}$$

$$W_{512} = W_Q + S_{512} = 14.3 + 32 = 46.3 \text{ mseg}$$



# **Introduction to Discrete Event Simulation**

Desempenho e Dimensionamento de Redes

Prof. Amaro de Sousa ([asou@ua.pt](mailto:asou@ua.pt))

DETI-UA, 2020/2021

# Discrete event simulation

A discrete event simulation models the operation of a system whose state changes with events that happen in discrete time instants:

- each event might force a change of the system state;
- between consecutive events, the system remains in the same state;
- thus, the simulation can directly jump in time from one event to the next event.

## Elements of a discrete event simulator:

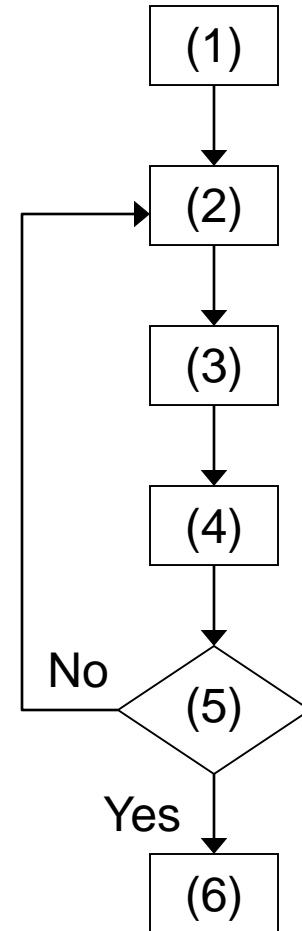
- (1) State variables: describe the state of the system at any time instant
- (2) Statistical counters: variables that store the appropriate statistical data related with the performance of the system
- (3) Simulation clock: variable indicating the current simulated time instant  
(simulated time  $\neq$  computation time)
- (4) Events: types of occurrences that change either the system state and/or the statistical counters
- (5) Event list: list of future events, their time instants and associated parameters

Besides these elements, additional supporting variables might be required. 2

# Basic structure of a discrete event simulator

A discrete event simulator is mainly composed by the following steps:

- (1) Initialization of the state variables, the statistical counters and the event list with the first event(s);
- (2) Determination of the next event from the event list;
- (3) Update of the simulation clock to the time instant of the event and removal of the event from the event list;
- (4) Execution of all actions associated to the event (generation of new events, update of state variables and/or statistical counters);
- (5) Check if the simulation must end; if not, return to Step (2);
- (6) Update of the statistical counters and determination of the performance parameters.



# **Example: performance of a video-streaming service with a single server**

*Input parameters of simulation:*

- (1) Time between movie requests: exponential distributed variable with an average of 60 minutes
- (2) Movie duration: exponential distributed variable with an average of 60 minutes
- (3) Server capacity = 2 movies

*Stopping criteria of simulation:*

Time instant of the 4<sup>th</sup> movie request (this request counts for the statistical counters)

*Performance parameters to be estimated by the simulation:*

- (1) Blocking probability: percentage of blocked movie requests (i.e., not provided due to server capacity)
- (2) Average server utilization (in number of movies)

# **Example: performance of a video-streaming service with a single server**

## Events:

ARRIVAL: the request of a movie

DEPARTURE: the end of a movie transmission

## State variables:

STATE: number of movies in transmission

## Statistical counters:

OCUPATION: integral of the server occupation (in number of movies) from the beginning of the simulation until the current time instant

BLOCKED: number of blocked movie requests from the beginning of the simulation until the current time instant

REQUESTS: number of movie requests from the beginning of the simulation until the current time instant

**Instant  $t = 0,0$**   
**Beginning of simulation**

Simulation Clock:

0

---

State Variable:

STATE  
0

---

Statistical  
Counters:

OCUPATION  
0

BLOCKED  
0

REQUESTS  
0

EVENT LIST  
ARRIVAL - 63 min

## Instant $t = 63$ (First ARRIVAL)

Simulation Clock:

63

EVENT LIST  
ARRIVAL - 63 min  
ARRIVAL - 117 min  
DEPARTURE - 153 min

---

State Variable:

STATE  
1

$\leftarrow 0 + 1$

---

Statistical  
Counters:

OCUPATION  
0

$\leftarrow 0 + 0 \times (63 - 0,0)$

BLOCKED  
0

$\leftarrow 0 + 0$

REQUESTS  
1

## Instant $t = 117$ (Second ARRIVAL)

Simulation Clock:

117

State Variable:

STATE  
2

Statistical  
Counters:

OCUPATION  
54

BLOCKED  
0

REQUESTS  
2

### EVENT LIST

ARRIVAL - 63 min

ARRIVAL - 117 min

ARRIVAL - 150 min

DEPARTURE - 153 min

DEPARTURE - 207 min

$$\leftarrow 1 + 1$$

$$\leftarrow 0 + 1 \times (117 - 63)$$

$$\leftarrow 0 + 0$$

## Instant $t = 150$ (Third ARRIVAL)

Simulation Clock:

150

State Variable:

STATE  
2

Statistical  
Counters:

OCUPATION  
120

BLOCKED  
1

REQUESTS  
3

### EVENT LIST

ARRIVAL - 63 min

ARRIVAL - 117 min

ARRIVAL - 150 min

DEPARTURE - 153 min

ARRIVAL - 204 min

DEPARTURE - 207 min

$$\leftarrow 2 + 0$$

$$\leftarrow 54 + 2 \times (150 - 117)$$

$$\leftarrow 0 + 1$$

## Instant $t = 153$ (First DEPARTURE)

Simulation Clock:

153

State Variable:

STATE  
1

Statistical  
Counters:

OCUPATION  
126

BLOCKED  
1

REQUESTS  
3

### EVENT LIST

ARRIVAL - 63 min

ARRIVAL - 117 min

ARRIVAL - 150 min

DEPARTURE - 153 min

ARRIVAL - 204 min

DEPARTURE - 207 min

$\leftarrow 2 - 1$

$\leftarrow 120 + 2 \times (153 - 150)$

**Instant  $t = 204$  (Fourth ARRIVAL)**

End of Simulation

Simulation Clock:

204

State Variable:

STATE  
2

Statistical  
Counters:

OCUPATION  
177

BLOCKED  
1

REQUESTS  
4

### EVENT LIST

ARRIVAL - 63 min

ARRIVAL - 117 min

ARRIVAL - 150 min

DEPARTURE - 153 min

ARRIVAL - 204 min

DEPARTURE - 207 min

$$\leftarrow 1 + 1$$

$$\leftarrow 126 + 1 \times (204 - 153)$$

$$\leftarrow 1 + 0$$

Blocking probability:

BLOCKED / REQUESTS = 1/4 = 25%

Average server utilization:

OCUPATION /  $t = 177/204 = 0,87$  movies

# MATLAB simulation of a video-streaming service with a single server

```
function [b o]= simulator1(lambda,invmiu,C,M,R)
invlambda=60/lambda;
ARRIVAL= 0;
DEPARTURE= 1;
STATE= 0;
OCUPATION= 0;
REQUESTS= 0;
BLOCKED= 0;
Clock= 0;
EventList= [ARRIVAL exprnd(invlambda)];
while REQUESTS < R
    event= EventList(1,1);
    Previous_Clock= Clock;
    Clock= EventList(1,2);
    EventList(1,:)= [];
    OCUPATION= OCUPATION + STATE * (Clock - Previous_Clock);
    if event == ARRIVAL
        EventList= [EventList; ARRIVAL, Clock + exprnd(invlambda)];
        REQUESTS= REQUESTS + 1;
        if STATE + M <= C
            STATE= STATE + M;
            EventList= [EventList; DEPARTURE, Clock + exprnd(invmiu)];
        else
            BLOCKED= BLOCKED + 1;
        end
    else
        STATE= STATE - M;
    end
    EventList= sortrows(EventList,2);
end
b= BLOCKED/REQUESTS;
o= OCUPATION/Clock;
end
```

## INPUT PARAMETERS:

lambda: movie request rate (requests/hour)  
invmiu: average movie duration (minutes)  
C: server capacity (in Mbps)  
M: throughput of each movie (in Mbps)  
R: no. of requests to stop the simulation

# **Generation of random numbers with a uniform distribution between 0 and 1**

A Linear Congruential Generator (LCG) is an algorithm that yields a sequence of randomized numbers calculated with a linear equation.

The method represents one of the oldest and best-known pseudorandom number generator algorithms.

Generation method:

(1) Generate integer values  $Z_1, Z_2, \dots$  with the following recursive expression:

$$Z_i = (aZ_{i-1} + c) \pmod{m}$$

where  $m, a, c$  and  $Z_0$  are non-negative integer parameters;

(2) Compute  $U_i = Z_i / m$ .

The values  $U_i$  seem to be real values uniformly distributed on the interval  $[0,1]$

## Example

Example:  $Z_i = (5Z_{i-1} + 3)(\text{mod } 16)$

$$Z_0 = 7$$

$i$	$Z_i$	$U_i$	$i$	$Z_i$	$U_i$
0	7	----	10	9	0.563
1	6	0.375	11	0	0.000
2	1	0.063	12	3	0.188
3	8	0.500	13	2	0.125
4	11	0.688	14	13	0.813
5	10	0.625	15	4	0.250
6	5	0.313	16	7	0.438
7	12	0.750	17	6	0.375
8	15	0.938	18	1	0.063
9	14	0.875	19	8	0.500

- In this example,  $m = 16$  and the algorithm repeats the generated numbers after 16 iterations (we say the generator has a period of 16).
- The random generator of MATLAB has a period of  $2^{31}-1$ .

# Generation of random numbers with other distributions

## Discrete variables:

Consider a random variable that can have the values  $X_1, X_2, \dots, X_n$ .

Consider the probability of value  $X_i$  as  $P(X = X_i) = f_i$ .

Method:

- Split the interval  $[0,1]$  in  $n$  intervals proportional to  $f_i$ ,  $i = 1 \dots n$
- Generate a uniformly distributed random value  $U$  in  $[0,1]$
- Return  $X_i$  if  $U$  falls into the  $i$ -th interval

For example, the Bernoulli variable  $X$  with  $p(0) = 1/4$  and  $p(1) = 3/4$  can be randomly generated as:

(1) Generate  $U \sim U(0,1)$

(2) If  $U \leq 1/4$ , return  $X = 0$ ; otherwise, return  $X = 1$

# Generation of random numbers with other distributions

Discrete variables (MATLAB example)

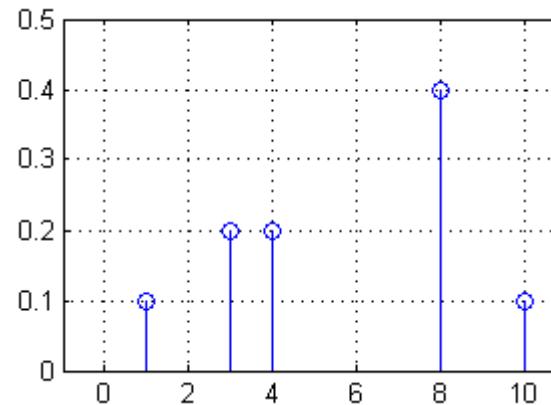
```
x= [1 3 4 8 10];  
f= [0.1 0.2 0.2 0.4 0.1];
```

```
figure(1)  
stem(x,f)  
axis([-1 11 0 0.5])  
grid on
```

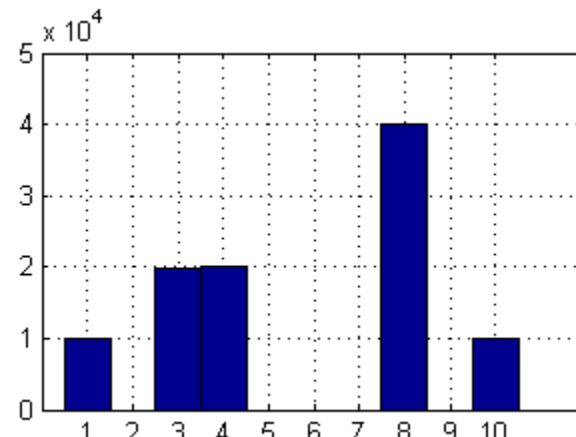
```
f_cum= [0 cumsum(f)]
```

```
a= zeros(1,100000);  
for it= 1:100000  
    a(it)= x(sum(rand()>f_cum));  
end
```

```
figure(2)  
hist(a,1:10)  
grid on
```



```
f_cum =  
0.0 0.1 0.3 0.5 0.9 1.0
```



# Generation of random numbers with other distributions

Continuous variables:

The most popular methods are based on the inverse of the cumulative distribution function (cdf).

Consider  $F(X)$  as the cdf of a continuous random variable and  $F^{-1}(U)$  as its inverse function.

Method:

- (1) Generate  $U \sim U(0,1)$
- (2) Return  $X = F^{-1}(U)$

For example, an exponential distributed random variable with average  $1/\lambda$ :

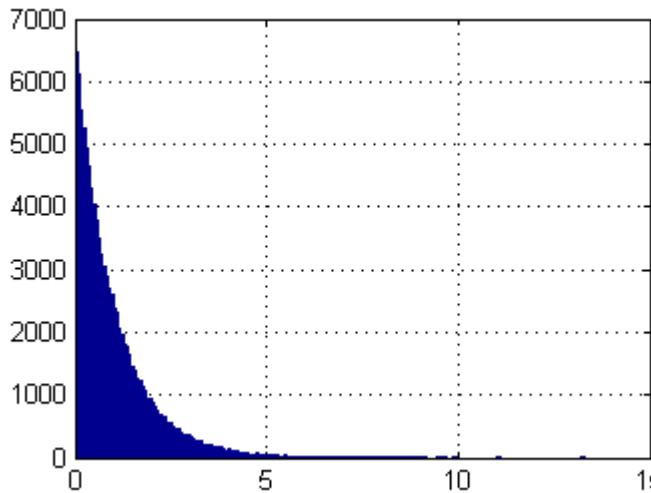
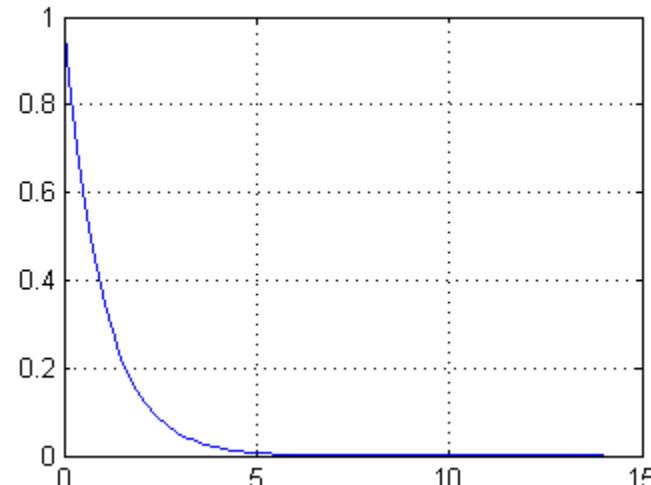
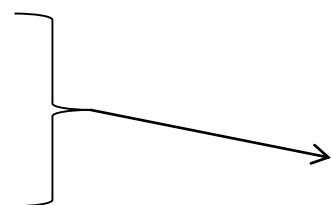
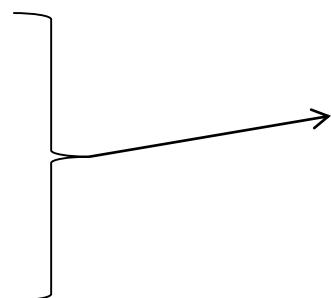
$$F(x) = \begin{cases} 1 - e^{-\lambda x}, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad F^{-1}(U) = -\frac{1}{\lambda} \ln(U)$$

# Generation of random numbers with other distributions

Exponential variable (MATLAB example)

```
x= 0:0.1:14;  
f=exppdf(x,1)  
figure(1)  
plot(x,f)  
grid on
```

```
a=exprnd(1,1,100000);  
figure(2)  
hist(a,200)  
grid on
```



# Analysis of the results of a simulation

Consider  $X_1, X_2, \dots, X_n$  as the observations of independent and identically distributed (IID) random variables with average  $\mu$  and finite variance  $\sigma^2$  (for example, the results of different simulations of a given system).

The sample mean defined by  
is an estimator for average  $\mu$ .

$$\bar{X}(n) = \frac{\sum_{i=1}^n X_i}{n}$$

The sample variance defined by  
is an estimator for variance  $\sigma^2$ .

$$S^2(n) = \frac{\sum_{i=1}^n (X_i - \bar{X}(n))^2}{n-1}$$

The analysis of the results of a simulation is, usually, based on the Central Limit Theorem.

# Analysis of the results of a simulation

Consider  $Z_n$  as a random variable given by:  $Z_n = \frac{\bar{X}(n) - \mu}{\sqrt{\sigma^2/n}}$

Consider  $F_n(z)$  the cumulative distribution function of  $Z_n$  for a sample of size  $n$ .

The Central Limit Theorem states that

$$\lim_{n \rightarrow +\infty} F_n(z) = \Phi(z)$$

where  $\Phi(z)$  is the cumulative distribution function of a standard Gaussian random variable (i.e., a Gaussian distribution with mean 0 and variance 1).

Given that  $\lim_{n \rightarrow +\infty} S^2(n) = \sigma^2$  than, the random variable  $\frac{\bar{X}(n) - \mu}{\sqrt{S^2(n)/n}}$

has approximately a standard Gaussian distribution.

# Analysis of the results of a simulation

For a sufficiently high value of  $n$ ,

$$P\left(-z_{1-\alpha/2} \leq \frac{\bar{X}(n) - \mu}{\sqrt{S^2(n)/n}} \leq z_{1-\alpha/2}\right) = \\ P\left(\bar{X}(n) - z_{1-\alpha/2} \sqrt{S^2(n)/n} \leq \mu \leq \bar{X}(n) + z_{1-\alpha/2} \sqrt{S^2(n)/n}\right) \approx 1 - \alpha$$

where  $z_{1-\alpha/2}$  is the critical value of the standard Gaussian distribution ( $z_{1-\alpha/2}$  is the value  $z$  such that  $P(x \leq z) = 1 - \alpha/2$  where  $x$  is a random variable with a standard Gaussian distribution).

Therefore, the approximate confidence interval of  $100(1-\alpha)\%$  for the average  $\mu$  is given as

$$\bar{X}(n) \pm z_{1-\alpha/2} \sqrt{S^2(n)/n}$$

# Analysis of the results of a simulation

The approximate confidence interval of  $100(1-\alpha)\%$  for the average  $\mu$  is

$$\bar{X}(n) \pm z_{1-\alpha/2} \sqrt{S^2(n)/n}$$

In MATLAB:

```
N = 20; %number of simulations
results= zeros(1,N); %vector with N simulation results
for it= 1:N
    results(it)= simulator();
end

alfa= 0.1; %90% confidence interval%
media = mean(results);
term = norminv(1-alfa/2)*sqrt(var(results)/N);

fprintf('resultado = %.2e +- %.2e\n',media,term)
```

# Analysis of the results of a simulation

The central limit theorem requires variables  $X_1, X_2, \dots, X_n$  to be independent and identically distributed (IID).

- One way to guarantee the independence between the different values is to run different simulations guaranteeing that the random values are different on the different runs.
- This is done by using different seeds on the random generators.

In general, the stochastic processes have initial transient states (which are dependent on the initial conditions) before reaching the stationary state.

- In order to guarantee that the performance estimations are correct, the simulation must first warm-up to let the transient states vanish.
- If the simulated time is much higher than the warm-up time, statistical counters can be initialized at the beginning of the simulation.
- Otherwise, the statistical counters must be initialized only after the warm-up time (this time must be estimated though).



## **Partilha de Recursos de uma Ligação Ponto-a-Ponto**

Desempenho e Dimensionamento de Redes

Prof. Amaro de Sousa ([asou@ua.pt](mailto:asou@ua.pt))

DETI-UA, 2020/2021

# Recursos de uma ligação digital ponto-a-ponto

Uma ligação digital é um sistema de transmissão que permite a troca de informação em formato digital.

Funciona como um *túnel de bits* através do qual pode ser transmitido/recebido um dado valor em bits/s. Este valor é designado por capacidade da ligação.

Exemplos:

- Cada sentido de uma ligação ponto-a-ponto bidirecional (um cabo Ethernet a ligar um PC a um *switch* Ethernet) é um recurso de comunicações entre os dois elementos.
- Uma interface de rede de um servidor de vídeo-streaming é um recurso de comunicações do servidor para os terminais de vídeo.
- Uma ligação de um router de uma casa ao ISP é constituído por dois recursos de comunicação, um em cada sentido, cuja capacidade é normalmente diferente em cada sentido.

# Recursos de uma ligação estruturada em circuitos

Uma ligação pode ser explicitamente estruturada em circuitos (ou canais), que correspondem a partições da capacidade total da ligação.

- Um circuito é caracterizado pela sua largura de banda (um valor também em bits/s).
- A estruturação de uma ligação em circuitos pode ser feita através de:
  - (1) multiplexagem no tempo (TDM – *Time Division Multiplexing*)
  - (2) multiplexagem na frequência (FDM – *Frequency Division Multiplexing*)
  - (3) multiplexagem de comprimentos de onda em redes óticas (WDM – *Wavelength Division Multiplexing*)

Uma ligação pode também ser implicitamente usada como se um fosse estruturada em circuitos (noção de circuitos virtuais).

- Cada comunicação usa uma partição da capacidade da ligação.

# Partilha de recursos de uma ligação estruturada em circuitos

- Uma chamada, quando é estabelecida, corresponde à atribuição temporária de um ou mais circuitos.
- Após estabelecida, a chamada dura um tempo finito após o qual a largura de banda dos circuitos atribuídos é libertada e fica disponível para pedidos futuros de chamadas.
- Assim, uma ligação pode ser partilhada por diferentes fluxos de chamadas.

Uma classe de serviço identifica um fluxo de chamadas com as mesmas características de tráfego e os mesmos requisitos de largura de banda.

# Recurso de comunicações com estabelecimento de circuitos

- As características de tráfego de uma classe de serviço são determinadas:
  - (i) pela descrição estatística do processo de chegada de chamadas,
  - (ii) pela descrição estatística da duração das chamadas, e
  - (iii) pelos recursos (em número de circuitos) requeridos por cada chamada.
- O parâmetro de qualidade de serviço mais importante é a probabilidade de bloqueio (probabilidade de um pedido de chamada não ser aceite por falta de recursos disponíveis na ligação).
- Um recurso de comunicações pode ser:
  - (1) uni-serviço – se suportar apenas fluxos de chamadas de uma única classe de serviço
  - (2) multi-serviço – se suportar fluxos de chamadas de diferentes classes de serviço
- A diferenciação entre diferentes classes de serviço faz-se através da função de controle de fluxos (também designada de controlo de admissão de chamadas)

# Recurso de comunicações uni-serviço - Distribuição de ErlangB

Um recurso de comunicações:

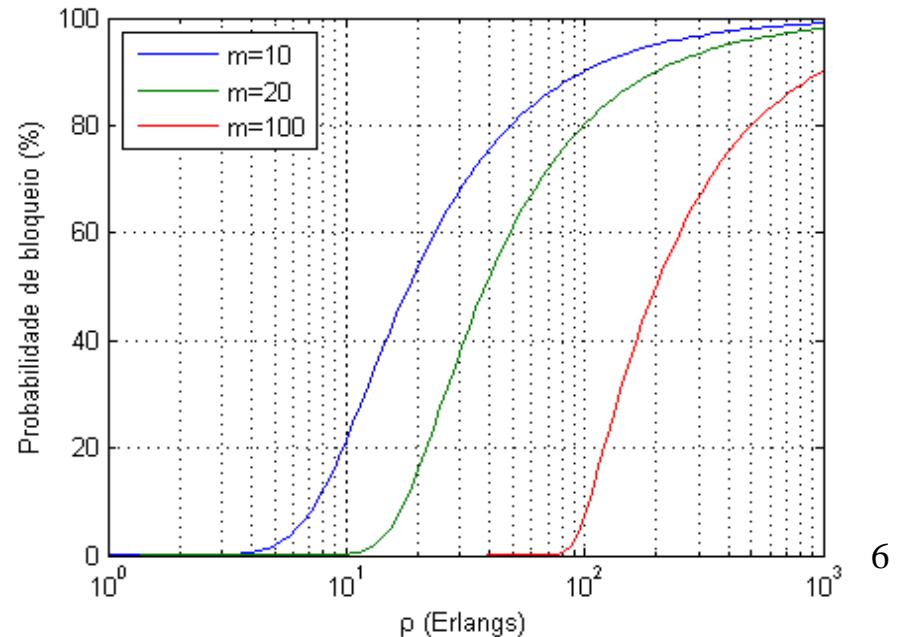
- (i) estruturado por um grupo de  $m$  circuitos
- (ii) ao qual é oferecido um fluxo de chamadas de Poisson com taxa  $\lambda$
- (iii) em que cada chamada requer um circuito
- (iv) e a duração das chamadas é exponencialmente distribuída com média  $1/\mu$

é modelado por um sistema de fila de espera  $M/M/m/m$ .

Designa-se como unidade de intensidade de tráfego,  
 $\rho = \lambda/\mu$ , o Erlang.

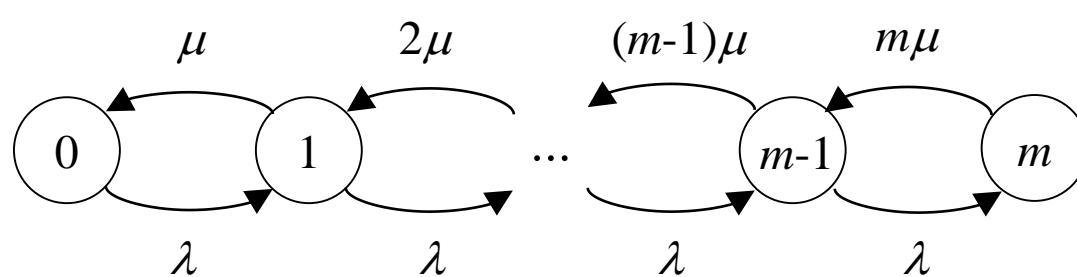
A fórmula de ErlangB, dá a probabilidade de bloqueio:

$$P_m = \frac{\rho^m / m!}{\sum_{n=0}^m \rho^n / n!} = E(\rho, m)$$



# Estabelecimento de circuitos uni-serviço

Cadeia de Markov de um sistema de fila de espera  $M/M/m/m$ :

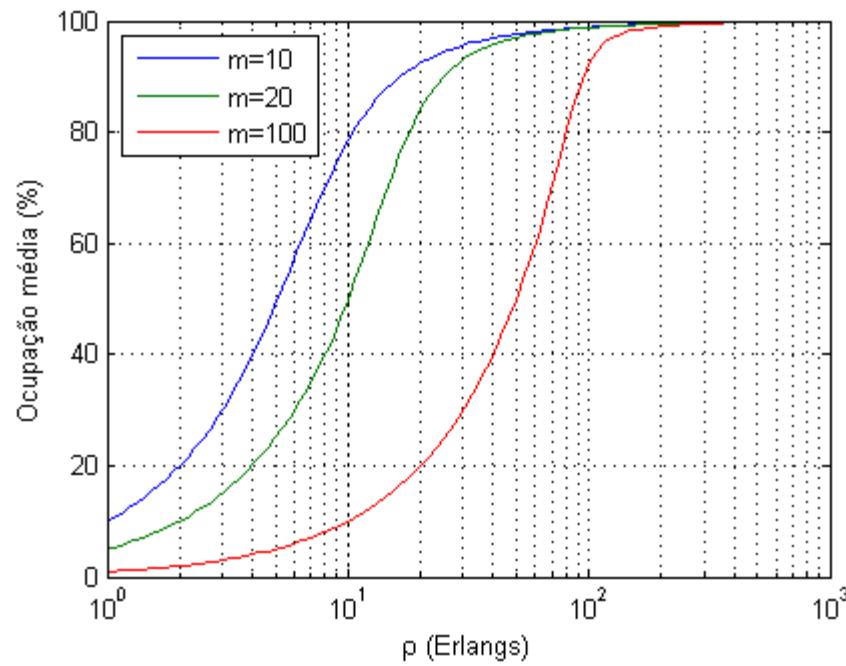


$$\pi_0 = \frac{1}{1 + \sum_{i=1}^m \frac{\lambda_0 \lambda_1 \cdots \lambda_{i-1}}{\mu_1 \mu_2 \cdots \mu_i}}$$

$$\pi_n = \frac{\lambda_0 \lambda_1 \cdots \lambda_{n-1}}{\mu_1 \mu_2 \cdots \mu_n} \cdot \pi_0, \quad n \geq 1$$

A ocupação média da ligação  
é ( $\rho = \lambda/\mu$ ):

$$\sum_{i=0}^m (i \times \pi_i) = \frac{\sum_{i=1}^m \frac{\rho^i}{(i-1)!}}{\sum_{i=0}^m \frac{\rho^i}{i!}}$$



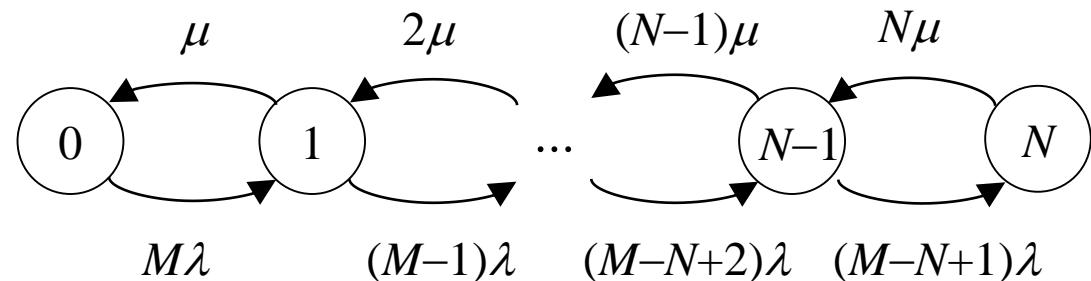
# Recurso de comunicações uni-serviço

## Distribuição de Engset

- Considere-se uma ligação estruturada em  $N$  circuitos e que serve  $M$  clientes ( $M > N$ ).
- Cada um dos  $M$  clientes está inativo durante um período de tempo exponencialmente distribuído com média  $1/\lambda$  e gera uma chamada com uma duração média  $1/\mu$ .
- A cada chamada pedida é atribuído um dos  $N$  circuitos disponíveis; se não houver qualquer circuito disponível a chamada é bloqueada.
- Este sistema é modelado por um processo de nascimento e morte com os estados  $n = 0, 1, \dots, N$  (representando o número de circuitos ocupados) e com taxas de nascimento e de morte dadas por:

$$\lambda_n = (M-n)\lambda, \quad 0 \leq n \leq N-1$$

$$\mu_n = n\mu, \quad 1 \leq n \leq N$$



# Recurso de comunicações uni-serviço

## Distribuição de Engset

A probabilidade de  $n$  chamadas no sistema é:

$$\pi_n = \frac{\binom{M}{n} \left(\frac{\lambda}{\mu}\right)^n}{\sum_{n=0}^N \binom{M}{n} \left(\frac{\lambda}{\mu}\right)^n}$$

A probabilidade de bloqueio é:

$$P_B = \frac{\binom{M-1}{N} \left(\frac{\lambda}{\mu}\right)^N}{\sum_{n=0}^N \binom{M-1}{n} \left(\frac{\lambda}{\mu}\right)^n}$$

Esta é a chamada distribuição de Engset.

- Note-se que a propriedade PASTA não se verifica ( $\pi_n \neq P_B$ ) dado que a taxa de chegada de chamadas depende do estado do sistema (i.e., não é estatisticamente independente do estado do sistema).
- Esta fórmula degenera na fórmula de ErlangB, quando  $M \rightarrow \infty$   $\lambda \rightarrow 0$ , com  $M\lambda$  constante.

# Estabelecimento de circuitos multi-serviço

Considere uma ligação com  $C$  circuitos que serve as classes de serviço  $k = 1, 2, \dots, K$  ( $K$  é o número de classes de serviço).

A cada classe  $k$  está associada uma taxa de chegada,  $\lambda_k$ , um tempo médio de serviço,  $1/\mu_k$  e uma largura de banda  $b_k$  (em número de circuitos):

- As chamadas das  $K$  classes chegam de acordo com processos independentes de Poisson à taxa  $\lambda_k$ .
- Uma chamada da classe  $k$  que tenha sido admitida pelo sistema, ocupa  $b_k$  circuitos durante o tempo de serviço da chamada, o qual é exponencialmente distribuído com média  $1/\mu_k$ .

Seja  $\rho_k$  a intensidade de tráfego de cada classe, isto é,  $\rho_k = \lambda_k / \mu_k$ .

Assuma-se que os tempos de serviço das chamadas são independentes entre si e independentes dos processos de chegadas.

# Estabelecimento de circuitos multi-serviço

Seja  $n_k$  o número de chamadas da classe  $k$  no sistema.

Considerem-se os vetores  $\mathbf{n} = (n_1, \dots, n_k)$  e  $\mathbf{b} = (b_1, \dots, b_k)$ .

O número total de circuitos ocupados no sistema é:

$$\mathbf{b} \cdot \mathbf{n} = \sum_{k=1}^K b_k n_k$$

Uma chamada da classe  $k$  é admitida no sistema se  $b_k \leq C - \mathbf{b} \cdot \mathbf{n}$ ; caso contrário é bloqueada e perdida.

O espaço de estados do processo de nascimento e morte multidimensional é:

$$S = \left\{ \mathbf{n} \in I^K : \mathbf{b} \cdot \mathbf{n} \leq C \right\}$$

onde  $I$  é o conjunto dos inteiros não negativos.

Seja  $S_k$  o subconjunto dos estados para os quais uma chamada da classe  $k$  é admitida quando chega à rede, isto é,

$$S_k = \left\{ \mathbf{n} \in S : \mathbf{b} \cdot \mathbf{n} \leq C - b_k \right\}$$

# Estabelecimento de circuitos multi-serviço

As probabilidades limite de cada estado são dadas por:

$$\text{onde } G = \sum_{\mathbf{n} \in S} \prod_{l=1}^K \frac{\rho_l^{n_l}}{n_l!} \quad P(\mathbf{n}) = \frac{1}{G} \prod_{l=1}^K \frac{\rho_l^{n_l}}{n_l!} \quad \mathbf{n} \in S$$

e a probabilidade de bloqueio da classe  $k$  por:

$$B_k = 1 - \frac{\sum_{\mathbf{n} \in S_k} \prod_{l=1}^K \frac{\rho_l^{n_l}}{n_l!}}{\sum_{\mathbf{n} \in S} \prod_{l=1}^K \frac{\rho_l^{n_l}}{n_l!}} \Leftrightarrow B_k = \frac{\sum_{\mathbf{n} \in S \setminus S_k} \prod_{l=1}^K \frac{\rho_l^{n_l}}{n_l!}}{\sum_{\mathbf{n} \in S} \prod_{l=1}^K \frac{\rho_l^{n_l}}{n_l!}}$$

- Primeira expressão: 1 menos a probabilidade dos estados  $S_k$  (estados para os quais uma chamada da classe  $k$  é admitida).
- Segunda expressão: probabilidade dos estados  $S \setminus S_k$  (estados para os quais uma chamada da classe  $k$  é rejeitada).

Este sistema é por vezes designado de *stochastic knapsack*.

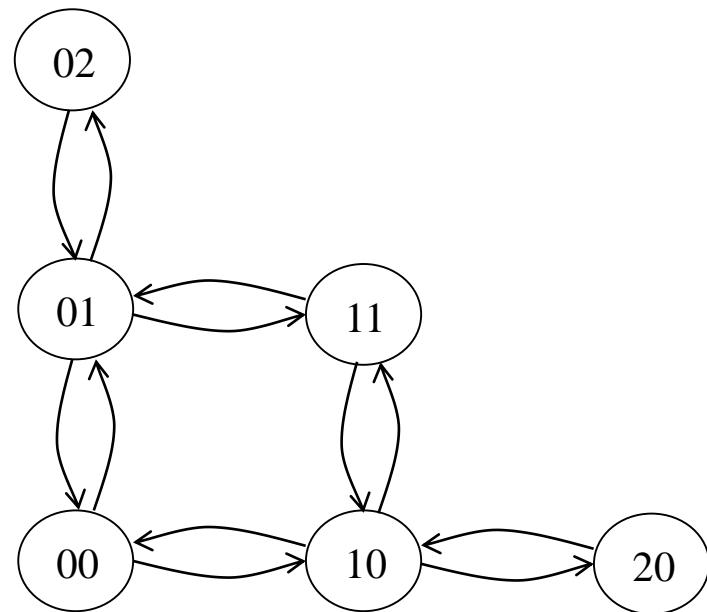
## Exemplo 1

Considere uma rede telefónica VoIP de uma empresa em que o número de chamadas VoIP para o exterior é no máximo 2. Existem dois tipos de chamadas VoIP: chamadas pessoais (tipo 1) e chamadas de apoio ao cliente (tipo 2).

O tráfego VoIP de/para a rede pública é um processo de Poisson com taxa de 5 chamadas/hora para chamadas do tipo 1 e de 2 chamadas/hora para as chamadas do tipo 2.

As chamadas do tipo 1 têm duração exponencial com média de 3 minutos e as chamadas do tipo 2 têm duração exponencial com média de 20 minutos.

Determine a probabilidade de bloqueio de cada tipo de chamada VoIP.



$$\rho_1 = \lambda_1/\mu_1 = 5/60 \times 3 = 1/4 \text{ Erlangs}$$

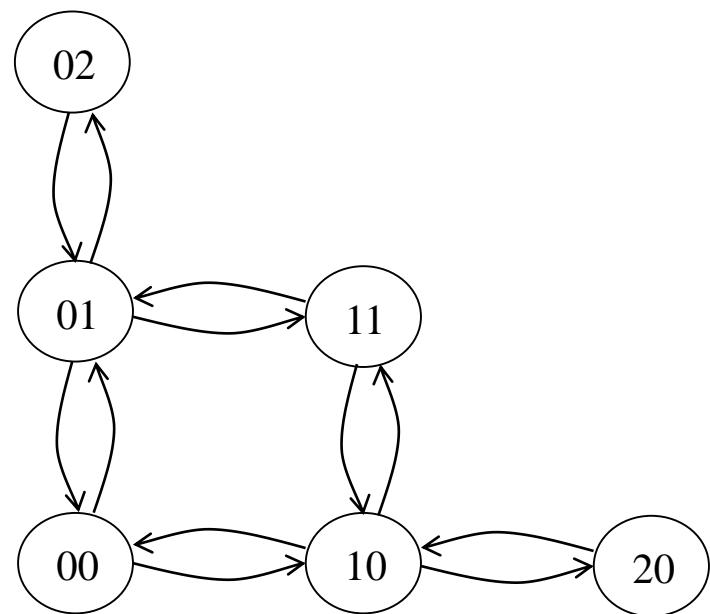
$$\rho_2 = \lambda_2/\mu_2 = 2/60 \times 20 = 2/3 \text{ Erlangs}$$

# Exemplo 1 - Resolução

$$\rho_1 = \lambda_1/\mu_1 = 5/60 \times 3 = 1/4 \text{ Erlangs}$$

$$\rho_2 = \lambda_2/\mu_2 = 2/60 \times 20 = 2/3 \text{ Erlangs}$$

$$B_k = \frac{\sum_{n \in S \setminus S_k} \prod_{l=1}^K \frac{\rho_l^{n_l}}{n_l!}}{\sum_{n \in S} \prod_{l=1}^K \frac{\rho_l^{n_l}}{n_l!}}$$



$$B_1 = \frac{\frac{(1/4)^2(2/3)^0}{2! 0!} + \frac{(1/4)^1(2/3)^1}{1! 1!} + \frac{(1/4)^0(2/3)^2}{0! 2!}}{\frac{(1/4)^0(2/3)^0}{0! 0!} + \frac{(1/4)^1(2/3)^0}{1! 0!} + \frac{(1/4)^2(2/3)^0}{2! 0!} + \frac{(1/4)^0(2/3)^1}{0! 1!} + \frac{(1/4)^1(2/3)^1}{1! 1!} + \frac{(1/4)^0(2/3)^2}{0! 2!}}$$

$$B_1 = \frac{\frac{(1/4)^2}{2} + 1/4 \times 2/3 + \frac{(2/3)^2}{2}}{1 + \frac{1}{4} + \frac{(1/4)^2}{2} + 2/3 + 1/4 \times 2/3 + \frac{(2/3)^2}{2}} = 0.1798 = 17.98\% \quad B_2 = B_1 = 17.98\%$$

# Recurso de comunicações com comutação de pacotes

Diferentes fluxos de pacotes podem ser

- (1) suportados por um circuito - multiplexagem estatística,
- (2) suportados por diferentes circuitos - multiplexagem determinística.

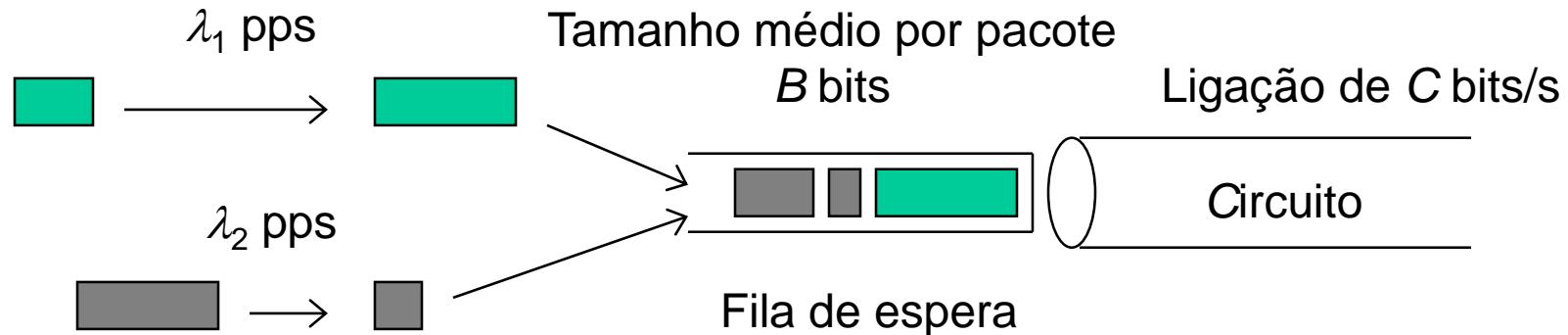
Tal como nos fluxos de chamadas, uma classe de serviço identifica um conjunto de fluxos de pacotes com as mesmas características de tráfego e os mesmos requisitos de qualidade de serviço.

As características de tráfego são determinadas por

- (i) descrição estatística do processo de chegada dos pacotes, e
- (ii) descrição estatística do tamanho dos pacotes.

Parâmetros de qualidade de serviço mais importantes são: o atraso médio por pacote e a taxa de perda de pacotes.

# Multiplexagem estatística de fluxos de pacotes



Considere-se que:

- (i) as chegadas de pacotes são processos de Poisson,
- (ii) o tamanho dos pacotes é exponencialmente distribuído,
- (iii) a fila de espera é atendida com uma disciplina *First-In-First-Out*.

Assim:

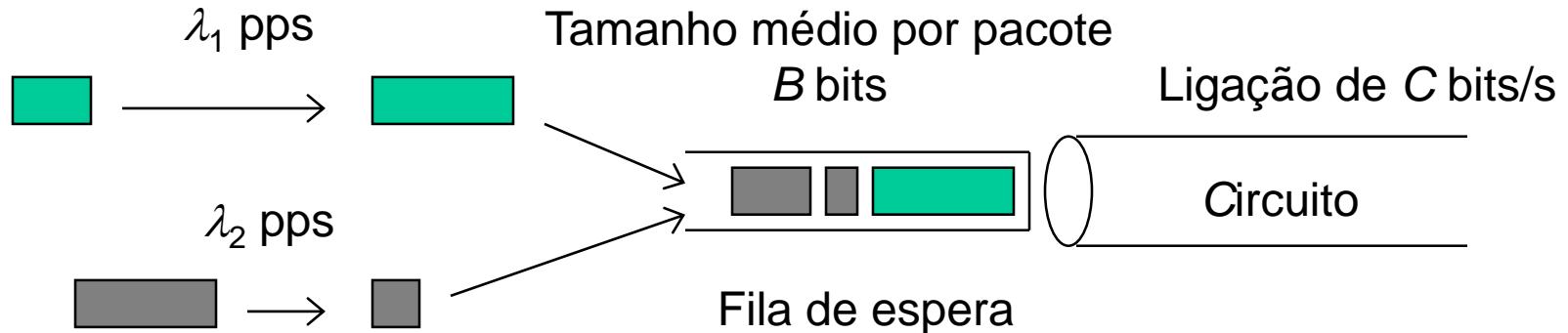
$$\begin{aligned}\lambda &= \lambda_1 + \lambda_2 \text{ pps (pacotes por segundo)} \\ \mu &= C/B \text{ pps}\end{aligned}$$

Se a fila de espera for de tamanho infinito, este sistema é modelado pelo sistema de filas de espera *M/M/1*:

Atraso médio dos pacotes:

$$W = \frac{1}{\mu - \lambda} = \frac{1}{C/B - (\lambda_1 + \lambda_2)}$$

# Multiplexagem estatística de fluxos de pacotes



Se a fila de espera for finita com capacidade para  $m-1$  pacotes, este sistema é modelado pelo sistema de filas de espera  $M/M/1/m$ :

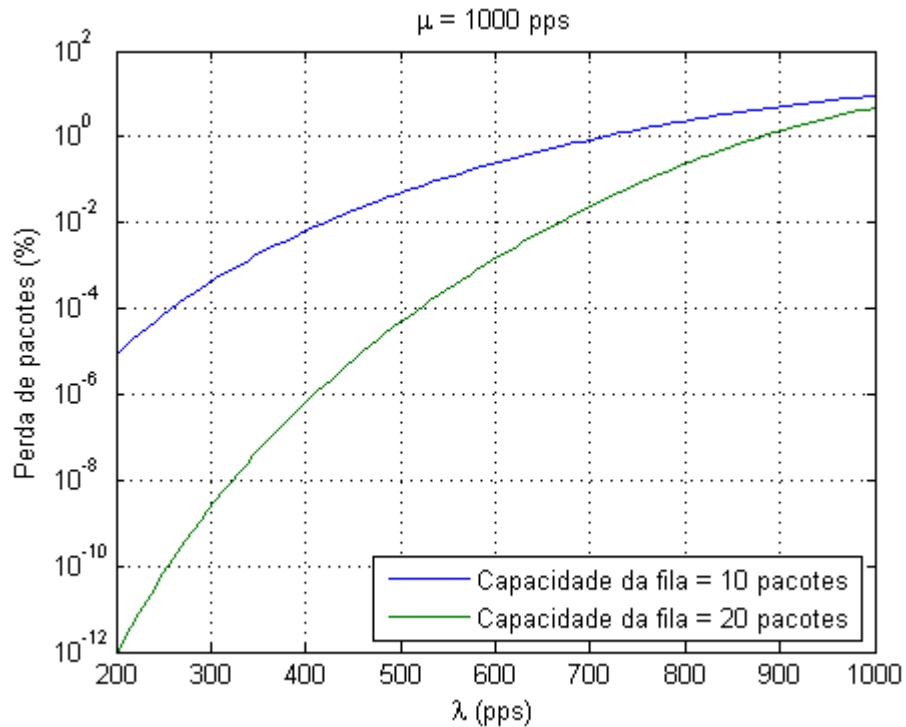
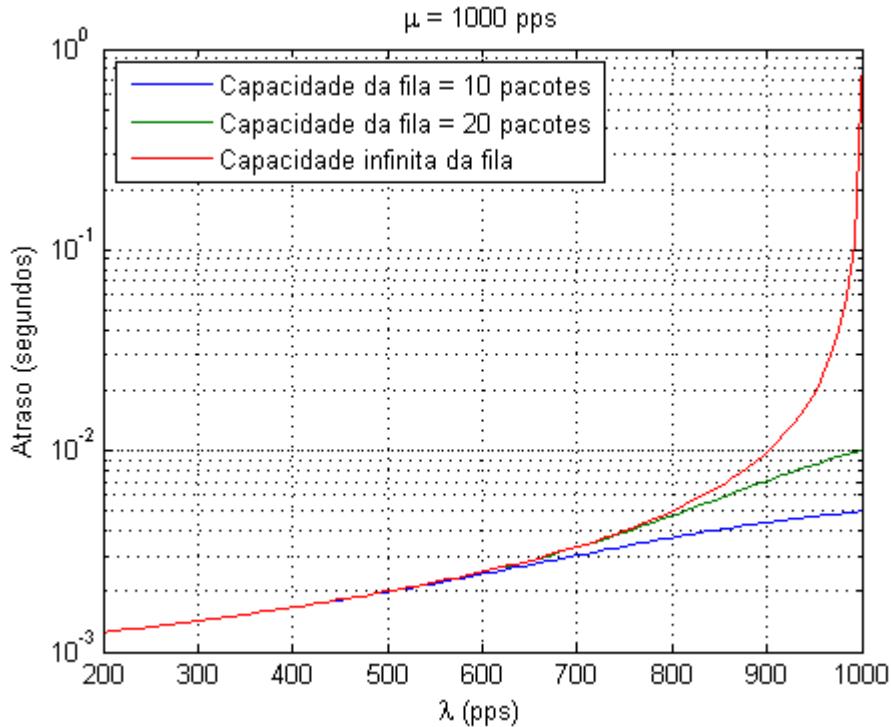
Percentagem de perda de pacotes:  $\mu_m = \frac{(\lambda/\mu)^m}{\sum_{j=0}^m (\lambda/\mu)^j}$

Número médio de pacotes no sistema:  $L = \sum_{i=0}^m i \times \pi_i = \frac{\sum_{i=0}^m i \times (\lambda/\mu)^i}{\sum_{j=0}^m (\lambda/\mu)^j}$

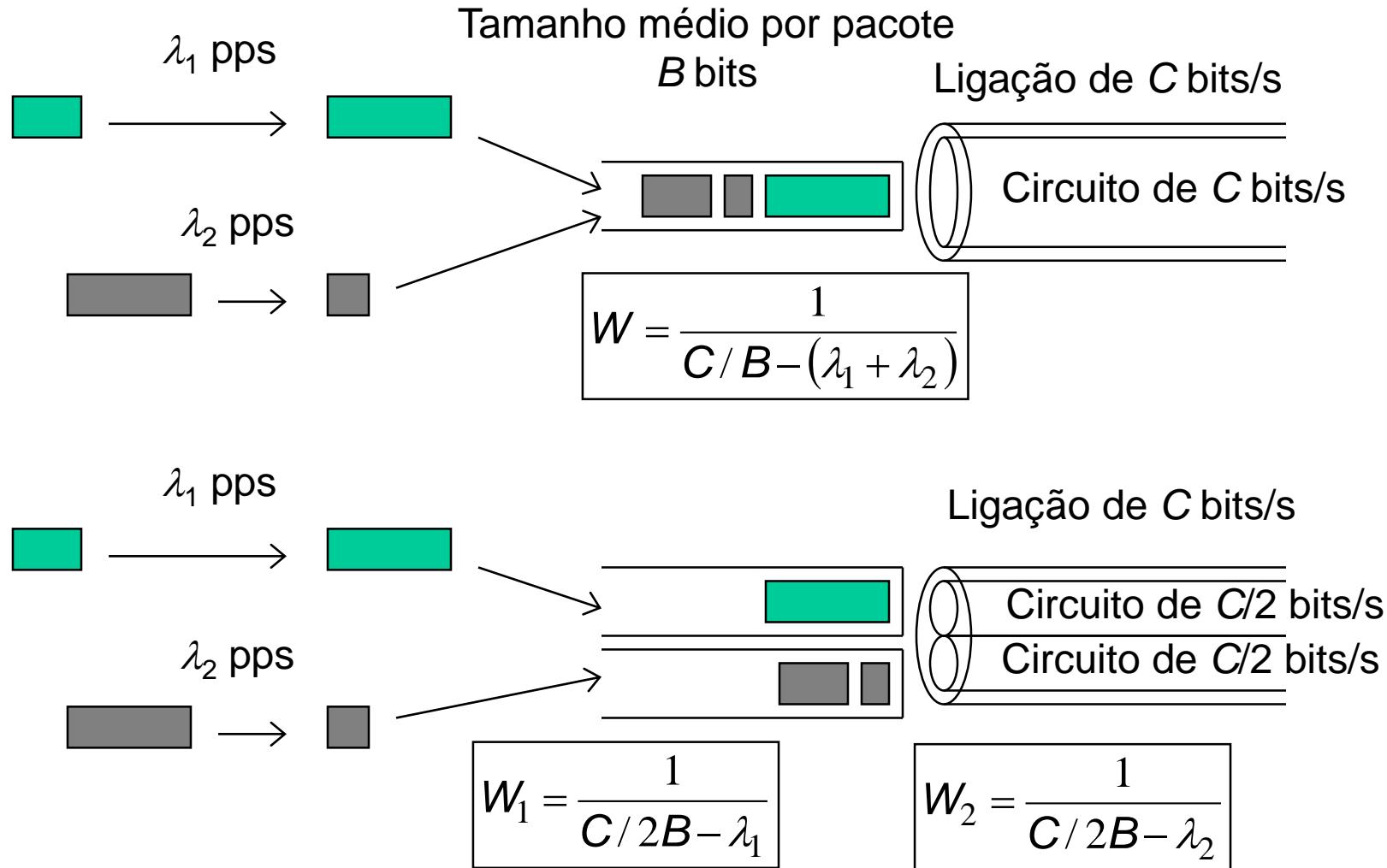
Atraso médio dos pacotes:  $W = \frac{L}{\lambda(1 - \mu_m)}$

# Multiplexagem estatística de fluxos de pacotes

- Se a fila de espera for de tamanho infinito, sistema modelado por  $M/M/1$
- Se a fila de espera tiver capacidade para  $m-1$  pacotes, sistema modelado por  $M/M/1/m$
- Exemplo:
  - ligação de 10 Mbps e tamanho médio de pacotes de 1250 Bytes
  - $\mu = 10^7 / (1250 \times 8) = 1000 \text{ pps}$



# Multiplexagem Estatística vs. Multiplexagem Determinística



## Exemplo 2

Considere uma ligação ponto-a-ponto de 10 Mbps que suporta dois fluxos de pacotes: fluxo A de 1 Mbps e fluxo B de 4 Mbps. Ambos os fluxos geram pacotes de tamanho exponencialmente distribuído com média de 1000 bytes. Calcule o atraso médio por pacote de cada fluxo quando:

- (a) os fluxos são multiplexados estatisticamente na capacidade total da ligação;
- (b) os fluxos são multiplexados deterministicamente em que é atribuída metade da capacidade da ligação a cada fluxo.

## Exemplo 2 – resolução

Considere uma ligação ponto-a-ponto de 10 Mbps que suporta dois fluxos de pacotes: fluxo A de 1 Mbps e fluxo B de 4 Mbps. Ambos os fluxos geram pacotes de tamanho exponencialmente distribuído com média de 1000 bytes. Calcule o atraso médio por pacote de cada fluxo quando:

- (a) os fluxos são multiplexados estatisticamente na capacidade total da ligação;

$$\mu = \frac{10 \times 10^6}{8 \times 1000} = 1250 \text{ pps} \quad \lambda_A = \frac{1 \times 10^6}{8 \times 1000} = 125 \text{ pps} \quad \lambda_B = \frac{4 \times 10^6}{8 \times 1000} = 500 \text{ pps}$$

$$W_A = W_B = \frac{1}{\mu - (\lambda_A + \lambda_B)} = \frac{1}{1250 - (125 + 500)} = 1.6 \times 10^{-3} = 1.6 \text{ ms}$$

- (b) os fluxos são multiplexados deterministicamente em que é atribuída metade da capacidade da ligação a cada fluxo.

$$W_A = \frac{1}{\frac{\mu}{2} - \lambda_A} = \frac{1}{\frac{1250}{2} - 125} = 2 \text{ ms} \quad W_B = \frac{1}{\frac{\mu}{2} - \lambda_B} = \frac{1}{\frac{1250}{2} - 500} = 8 \text{ ms}$$

# Multiplexagem Estatística vs. Multiplexagem Determinística

- Em geral, a multiplexagem estatística conduz a atrasos médios por pacote inferiores:

Na multiplexagem determinística, a capacidade atribuída a um fluxo e que esteja momentaneamente livre não pode ser usado pelos outros fluxos.

- No entanto, a multiplexagem determinística conduz a variâncias de atraso menores:

Na multiplexagem estatística, todos os fluxos partilham uma única fila de espera.

Assim, a diferença entre o número mínimo e o número máximo de pacotes na fila de espera é maior.

# Disciplina com prioridades

Na multiplexagem estatística, os pacotes de cada fluxo não podem ser tratados (i.e., transmitidos) de forma diferenciada.

Uma possibilidade é atribuir prioridades aos fluxos, de tal forma que os pacotes de um fluxo com maior prioridade são sempre transmitidos antes do que os pacotes de um fluxo com menor prioridade (esquema designado de *prioritização estrita*).

O sistema M/G/1 com prioridades pode ser utilizado para modelar este sistema.

Considere um sistema  $M/G/1$  em que existem  $n$  classes de serviço.

A  $k$ -ésima classe é determinada por:

(1) taxa de chegadas:  $\lambda_k$

(2) 1º e 2º momentos do tempo de serviço:  $E(S_k) = \frac{1}{\mu_k}$        $E(S_k^2)$

(3) prioridade:  $k$  (quanto menor este valor, maior a prioridade)

# Sistema M/G/1 com prioridades

O sistema transmite primeiro os pacotes das classes com maior prioridade.

Os pacotes de uma mesma classe são transmitidos por ordem de chegada (disciplina *FIFO - First In First Out*).

Considera-se que as chegadas dos pacotes de cada classe são independentes e de Poisson e independentes dos tempos de transmissão.

A transmissão de um pacote não é interrompida pela chegada de um pacote de uma classe com maior prioridade (disciplina de serviço designada por *não-preemptiva*).

O atraso médio na fila de espera correspondente aos pacotes da classe  $k$  é dado por:

$$W_{Qk} = \frac{\sum_{i=1}^n \lambda_i E(S_i^2)}{2(1 - \rho_1 - \dots - \rho_{k-1})(1 - \rho_1 - \dots - \rho_k)} \quad \begin{aligned} \rho_k &= \lambda_k / \mu_k \\ \rho_1 + \dots + \rho_n &< 1 \end{aligned}$$

## Exemplo 3

Considere uma ligação ponto-a-ponto de 10 Mbps que suporta dois fluxos de pacotes: fluxo A de 0.5 Mbps e fluxo B de 4 Mbps. Ambos os fluxos geram pacotes de tamanho exponencialmente distribuído com média de 1000 bytes. O atraso médio por pacote do fluxo A deverá ser de 2 milissegundos.

Pretende-se implementar um esquema de multiplexagem determinística que permita cumprir com a qualidade de serviço do fluxo A. Calcule:

- (a) a capacidade mínima da ligação que deve ser atribuída ao fluxo A;
- (b) o atraso médio por pacote do fluxo B no sistema resultante.

## Exemplo 3 – resolução

Considere uma ligação ponto-a-ponto de 10 Mbps que suporta dois fluxos de pacotes: fluxo A de 0.5 Mbps e fluxo B de 4 Mbps. Ambos os fluxos geram pacotes de tamanho exponencialmente distribuído com média de 1000 bytes. O atraso médio por pacote do fluxo A deverá ser de 2 milissegundos.

Pretende-se implementar um esquema de multiplexagem determinística que permita cumprir com a qualidade de serviço do fluxo A. Calcule:

(a) a capacidade mínima da ligação que deve ser atribuída ao fluxo A;

$$\mu = \frac{10 \times 10^6}{8 \times 1000} = 1250 \text{ pps} \quad \lambda_A = \frac{0.5 \times 10^6}{8 \times 1000} = 62.5 \text{ pps} \quad \lambda_B = \frac{4 \times 10^6}{8 \times 1000} = 500 \text{ pps}$$

$$W_A = \frac{1}{\mu_A - \lambda_A} = \frac{1}{\mu_A - 62.5} = 2 \times 10^{-3} \Leftrightarrow$$

$$\Leftrightarrow \mu_A = \frac{1}{2 \times 10^{-3}} + 62.5 = 562.5 \text{ pps} = 562.5 \times (8 \times 1000) = 4.5 \times 10^6 = 4.5 \text{ Mbps}$$

(b) o atraso médio por pacote do fluxo B no sistema resultante.

$$W_B = \frac{1}{\mu_B - \lambda_B} = \frac{1}{(1250 - 562.5) - 500} = 0.0053 = 5.3 \text{ ms}$$

## Exemplo 4

Considere uma ligação ponto-a-ponto de 10 Mbps que suporta dois fluxos de pacotes: fluxo A de 0.5 Mbps e fluxo B de 4 Mbps. Ambos os fluxos geram pacotes de tamanho exponencialmente distribuído com média de 1000 bytes.

O sistema atribui maior prioridade ao fluxo A com uma disciplina não preemptiva. Calcule o atraso médio por pacote de cada fluxo.

$$\lambda_A = \frac{0.5 \times 10^6}{8 \times 1000} = 62.5 \text{ pps}$$

$$\lambda_B = \frac{4 \times 10^6}{8 \times 1000} = 500 \text{ pps}$$

$$\mu_A = \mu_B = \mu = \frac{10 \times 10^6}{8 \times 1000} = 1250 \text{ pps}$$

$$E(S_A) = E(S_B) = \frac{1}{\mu} = \frac{1}{1250} \text{ seg.}$$

$$E(S_A^2) = E(S_B^2) = \frac{2}{\mu^2} = \frac{2}{1250^2} \text{ seg.}^2$$

$$W_{Qk} = \frac{\sum_{i=1}^n \lambda_i E(S_i^2)}{2(1 - \rho_1 - \dots - \rho_{k-1})(1 - \rho_1 - \dots - \rho_k)} \quad \rho_k = \lambda_k / \mu_k$$

$$W_A = \frac{\lambda_A \times E(S_A^2) + \lambda_B \times E(S_B^2)}{2 \times \left(1 - \frac{\lambda_A}{\mu_A}\right)} + E(S_A) = 1.2 \text{ ms}$$

$$W_B = \frac{\lambda_A \times E(S_A^2) + \lambda_B \times E(S_B^2)}{2 \times \left(1 - \frac{\lambda_A}{\mu_A}\right) \times \left(1 - \frac{\lambda_A}{\mu_A} - \frac{\lambda_B}{\mu_B}\right)} + E(S_B) = 1.5 \text{ ms}$$



# **Desempenho de Redes com Comutação de Circuitos**

Desempenho e Dimensionamento de Redes

Prof. Amaro de Sousa ([asou@ua.pt](mailto:asou@ua.pt))

DETI-UA, 2020/2021

# **Encaminhamento em redes com comutação de circuitos**

## **1. Encaminhamento fixo:**

- considera um único percurso de encaminhamento para cada fluxo de chamadas suportado pela rede
- bloqueia a chamada se o percurso não tiver recursos suficientes para a chamada pedida

## **2. Encaminhamento dinâmico:**

- considera uma sequência ordenada de percursos de encaminhamento para cada fluxo de chamadas
- estabelece a chamada no  $n$ -ésimo percurso se nenhum dos percursos até ao  $(n-1)$ -ésimo tiver recursos
- a sequência de percursos varia ao longo do tempo

# Encaminhamento fixo

Vamos abordar 3 métodos para o cálculo das probabilidades de bloqueio de cada fluxo de chamadas

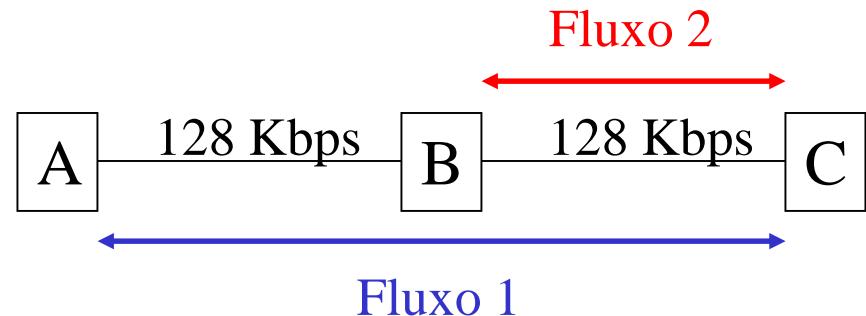
- Método exato
  - Computacionalmente pesado
  - Exige identificação de todos os estados possíveis da rede
- Teorema do limite do produto
  - Computacionalmente leve
  - É um majorante da probabilidade de bloqueio exata
- Aproximação de carga reduzida
  - Uma aproximação (normalmente boa) dos valores exatos
  - Matematicamente complexo
  - Existem algoritmos iterativos de cálculo

# Encaminhamento fixo – Método Exato

- Considere-se uma rede com  $J$  ligações que serve  $K$  fluxos de chamadas.
- A ligação  $j = 1, \dots, J$  tem capacidade  $C_j$  (em número de circuitos).
- Ao fluxo  $k = 1, \dots, K$  está associada uma taxa de chegada,  $\lambda_k$ , um tempo médio de serviço  $1/\mu_k$  (intensidade de tráfego  $\rho_k = \lambda_k / \mu_k$ ), uma largura de banda  $b_k$  (em número de circuitos) e um percurso de encaminhamento fixo  $R_k \subseteq \{1, 2, \dots, J\}$ :
  - (1) As chamadas do fluxo  $k$  chegam de acordo com um processo de Poisson à taxa  $\lambda_k$ .
  - (2) As chamadas do fluxo  $k$  admitidas pelo sistema ocupam  $b_k$  circuitos e têm uma duração exponencialmente distribuída com média  $1/\mu_k$ .
  - (3) A duração das chamadas é independente entre chamadas e independente dos instantes de chegada para todos os fluxos.
- O conjunto dos fluxos que atravessam a ligação  $j$  é dado por  $K_j$ .

# Encaminhamento fixo (Exemplo 1)

Considere a rede da figura que suporta 2 fluxos de chamadas: fluxo 1 com taxa  $\lambda_1 = 3$  chamadas/hora, duração média das chamadas  $1/\mu_1 = 2$  minutos e cada chamada ocupa  $b_1 = 64$  Kb/s; fluxo 2 com taxa  $\lambda_2 = 4$  chamadas/hora, duração média das chamadas  $1/\mu_2 = 3$  minutos e cada chamada ocupa  $b_2 = 128$  Kb/s.



$K = 2$  fluxos:

1:  $\rho_1 = \lambda_1/\mu_1 = 3/60 \times 2 = 0.1$  Erlangs,  $b_1 = 1$  circuito,  $R_1 = \{AB, BC\}$

2:  $\rho_2 = \lambda_2/\mu_2 = 4/60 \times 3 = 0.2$  Erlangs,  $b_2 = 2$  circuitos,  $R_2 = \{BC\}$

$J = 2$  ligações:

AB:  $C_{AB} = 2$  circuitos,  $K_{AB} = \{1\}$

BC:  $C_{BC} = 2$  circuitos,  $K_{BC} = \{1,2\}$

## Encaminhamento fixo – Método Exato

Seja  $n_k$  o número de chamadas do fluxo  $k$  no sistema,  $\mathbf{n} = (n_1, \dots, n_k)$

Uma chamada do fluxo  $k$  não é aceite pela rede se em pelo menos uma das ligações pertencentes ao percurso de  $k$ :

$$b_k + \sum_{l \in K_j} b_l n_l > C_j$$

O espaço de estados do processo de nascimento e morte multidimensional é

$$S = \left\{ \mathbf{n} \in I^K : \sum_{k \in K_j} b_k n_k \leq C_j, \quad j = 1, \dots, J \right\}$$

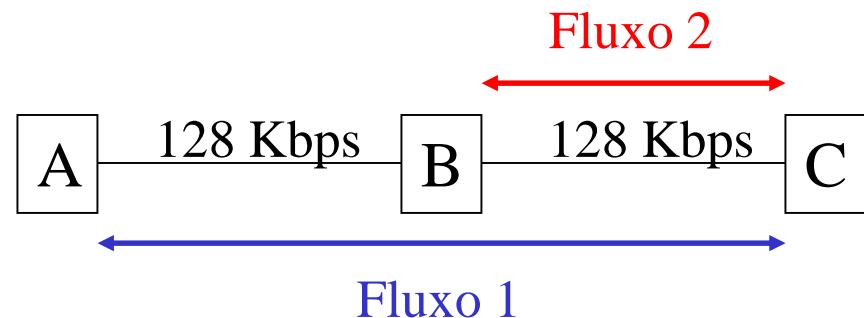
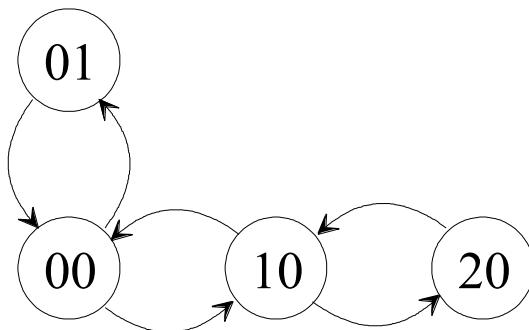
onde  $I$  é o conjunto dos inteiros não negativos.

Seja  $S_k$  o subconjunto dos estados nos quais uma chamada do fluxo  $k$  é admitida quando chega à rede, isto é,

$$S_k = \left\{ \mathbf{n} \in S : \sum_{l \in K_j} b_l n_l \leq C_j - b_k, \quad j \in R_k \right\}$$

# Encaminhamento fixo (Exemplo 1)

Considere a rede de figura que suporta 2 fluxos de chamadas: fluxo 1 com taxa  $\lambda_1 = 3$  chamadas/hora, duração média das chamadas  $1/\mu_1 = 2$  minutos e cada chamada ocupa  $b_1 = 64$  Kb/s; fluxo 2 com taxa  $\lambda_2 = 4$  chamadas/hora, duração média das chamadas  $1/\mu_2 = 3$  minutos e cada chamada ocupa  $b_2 = 128$  Kb/s.



Espaço de estados:  $S = \{(0,0), (1,0), (2,0), (0,1)\}$

Estados nos quais uma chamada do fluxo 1 é admitida:

$$S_1 = \{(0,0), (1,0)\}$$

Estados nos quais uma chamada do fluxo 2 é admitida:

$$S_2 = \{(0,0)\}$$

# Encaminhamento fixo – Método Exato

A probabilidade limite de cada estado é dada por:

$$P(\mathbf{n}) = \frac{1}{G} \prod_{k=1}^K \frac{\rho_k^{n_k}}{n_k!} \quad \mathbf{n} \in S$$

onde  $G = \sum_{\mathbf{n} \in S} \prod_{k=1}^K \frac{\rho_k^{n_k}}{n_k!}$

e a probabilidade de bloqueio da classe  $k$  é dada por:

$$B_k = 1 - \frac{\sum_{\mathbf{n} \in S_k} \prod_{l=1}^K \frac{\rho_l^{n_l}}{n_l!}}{\sum_{\mathbf{n} \in S} \prod_{l=1}^K \frac{\rho_l^{n_l}}{n_l!}} \Leftrightarrow B_k = \frac{\sum_{\mathbf{n} \in S \setminus S_k} \prod_{l=1}^K \frac{\rho_l^{n_l}}{n_l!}}{\sum_{\mathbf{n} \in S} \prod_{l=1}^K \frac{\rho_l^{n_l}}{n_l!}}$$

# Encaminhamento fixo (Exemplo 1)

Considere a rede de figura que suporta 2 fluxos de chamadas: fluxo 1 com taxa de chegada  $\lambda_1 = 3$  chamadas/hora, duração média das chamadas  $1/\mu_1 = 2$  minutos e cada chamada ocupa  $b_1 = 64$  Kb/s; fluxo 2 com taxa de chegada  $\lambda_2 = 4$  chamadas/hora, duração média das chamadas  $1/\mu_2 = 3$  minutos e cada chamada ocupa  $b_2 = 128$  Kb/s.

$$1: \rho_1 = 0.1 \text{ Erlangs}$$

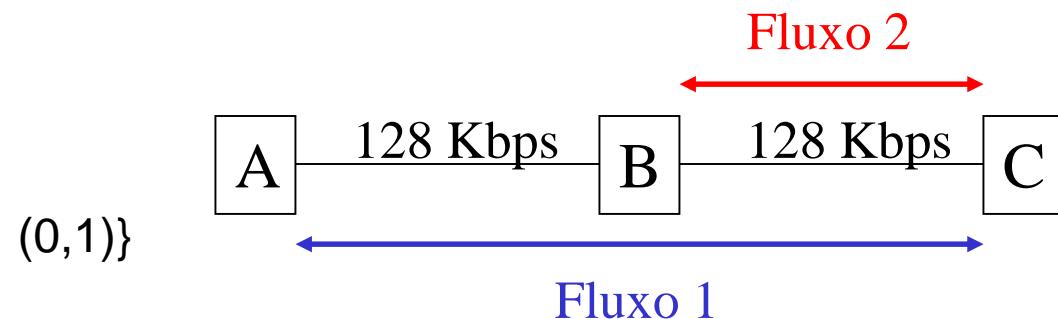
$$2: \rho_2 = 0.2 \text{ Erlangs}$$

$$S = \{(0,0), (1,0), (2,0), (0,1)\}$$

$$S_1 = \{(0,0), (1,0)\}$$

$$S_2 = \{(0,0)\}$$

$$B_k = 1 - \frac{\sum_{n \in S_k} \prod_{l=1}^K \frac{\rho_l^{n_l}}{n_l!}}{\sum_{n \in S} \prod_{l=1}^K \frac{\rho_l^{n_l}}{n_l!}}$$



$$\begin{aligned}
 B_1 &= 1 - \frac{\frac{0.1^0 0.2^0}{0! 0!} + \frac{0.1^1 0.2^0}{1! 0!}}{\frac{0.1^0 0.2^0}{0! 0!} + \frac{0.1^1 0.2^0}{1! 0!} + \frac{0.1^2 0.2^0}{2! 0!} + \frac{0.1^0 0.2^1}{0! 1!}} \\
 &= 1 - \frac{1 + 0.1}{1 + 0.1 + \frac{0.01}{2} + 0.2} = 0.157 = 15.7\%
 \end{aligned}$$

# Encaminhamento fixo (Exemplo 1)

Considere a rede de figura que suporta 2 fluxos de chamadas: fluxo 1 com taxa de chegada  $\lambda_1 = 3$  chamadas/hora, duração média das chamadas  $1/\mu_1 = 2$  minutos e cada chamada ocupa  $b_1 = 64$  Kb/s; fluxo 2 com taxa de chegada  $\lambda_2 = 4$  chamadas/hora, duração média das chamadas  $1/\mu_2 = 3$  minutos e cada chamada ocupa  $b_2 = 128$  Kb/s.

$$1: \rho_1 = 0.1 \text{ Erlangs}$$

$$2: \rho_2 = 0.2 \text{ Erlangs}$$

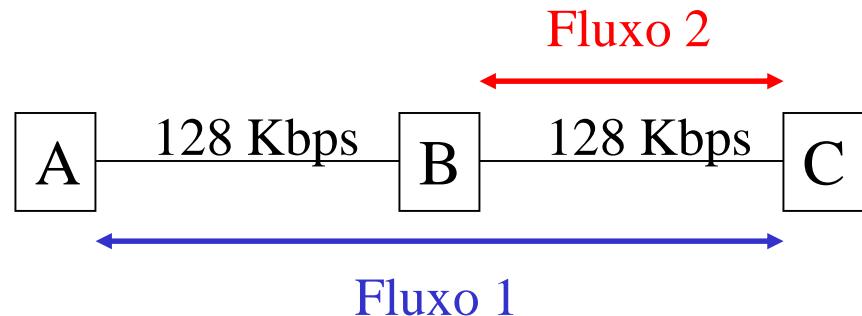
$$S = \{(0,0), (1,0), (2,0), (0,1)\}$$

$$S_1 = \{(0,0), (1,0)\}$$

$$S_2 = \{(0,0)\}$$

$$B_K = 1 - \frac{\sum_{\mathbf{n} \in S_k} \prod_{l=1}^K \frac{\rho_l^{n_l}}{n_l!}}{\sum_{\mathbf{n} \in S} \prod_{l=1}^K \frac{\rho_l^{n_l}}{n_l!}}$$

$$\begin{aligned} B_2 &= 1 - \frac{\frac{0.1^0 0.2^0}{0! 0!}}{\frac{0.1^0 0.2^0}{0! 0!} + \frac{0.1^1 0.2^0}{1! 0!} + \frac{0.1^2 0.2^0}{2! 0!} + \frac{0.1^0 0.2^1}{0! 1!}} \\ &= 1 - \frac{1}{1 + 0.1 + \frac{0.01}{2} + 0.2} = 0.234 = 23.4\% \end{aligned}$$



# Teorema do Limite do Produto

- Aplica-se apenas quando as chamadas de todos os fluxos  $k$  requerem a mesma largura de banda  $b_k$  (em número de circuitos).
- Seja a intensidade de tráfego suportada pela ligação  $j$  dada por:

$$\bar{\rho}_j = \sum_{k \in K_j} \rho_k$$

- O teorema do limite do produto declara que

$$B_k \leq 1 - \prod_{j \in R_k} \left( 1 - ER[\bar{\rho}_j, C_j] \right) \quad C_j - \text{capacidade da ligação } j \text{ (em número de chamadas)}$$

em que  $ER[\rho, C]$  representa a fórmula de ErlangB.

- Prova-se matematicamente que este valor é um majorante das probabilidades de bloqueio exatas. É uma boa aproximação quando:
  - (1) os fluxos atravessam poucas ligações
  - (2) as probabilidades de bloqueio são pequenas (menores que 1%)

# Aproximação de carga reduzida

Uma possibilidade para melhorar a aproximação associada ao teorema do limite do produto é reduzir o tráfego oferecido à ligação  $j$ , tomando em linha de conta o bloqueio nas restantes ligações do percurso de cada fluxo.

- O teorema do limite do produto implica que a probabilidade de uma ligação  $j$  estar totalmente ocupada é majorada por:

$$ER \left[ \sum_{k \in K_j} \rho_k, C_j \right]$$

- Substituindo  $\rho_k$  por  $\rho_k t_k(j)$ , em que  $t_k(j)$  corresponde à probabilidade de existir pelo menos uma unidade de capacidade disponível em cada ligação pertencente a  $R_k - \{j\}$ , a probabilidade de bloqueio (aproximada) da ligação  $j$  é dada por:

$$L_j = ER \left[ \sum_{k \in K_j} \rho_k t_k(j), C_j \right]$$

# Aproximação de carga reduzida

- Tomando como aproximação adicional que o bloqueio é independente entre ligações, resulta:

$$t_k(j) = \prod_{i \in R_k - \{j\}} (1 - L_i)$$

e, combinando as equações anteriores, obtêm-se as seguintes equações de ponto fixo (uma por cada ligação da rede):

$$L_j = ER \left[ \sum_{k \in K_j} \rho_k \prod_{i \in R_k - \{j\}} (1 - L_i), C_j \right], \quad j = 1, 2, \dots, J$$

- Admitindo novamente que o bloqueio é independente entre ligações, a probabilidade de bloqueio das chamadas do fluxo  $k$  é:

$$B_k \approx 1 - \prod_{j \in R_k} (1 - L_j) \quad k = 1, 2, \dots, K$$

# Algoritmo iterativo de cálculo da aproximação de carga reduzida

Seja  $\mathbf{L} = (L_1, L_2, \dots, L_J)$  e o operador  $\mathbf{T}(\mathbf{L}) = (T_1(\mathbf{L}), T_2(\mathbf{L}), \dots, T_J(\mathbf{L}))$  onde

$$T_j(\mathbf{L}) = ER \left[ \sum_{k \in K_j} \rho_k \prod_{i \in R_k - \{j\}} (1 - L_i), C_j \right]$$

As equações de ponto fixo podem ser expressas na forma  $\mathbf{L} = \mathbf{T}(\mathbf{L})$ .

Método iterativo – partindo de um vetor inicial  $\mathbf{L} \in [0,1]^J$  aplica-se sucessivamente o operador  $\mathbf{T}$ :

$$\mathbf{L}^0 = \mathbf{L}$$

$$\mathbf{L}^m = \mathbf{T}(\mathbf{L}^{m-1}), \quad m = 1, \dots, n$$

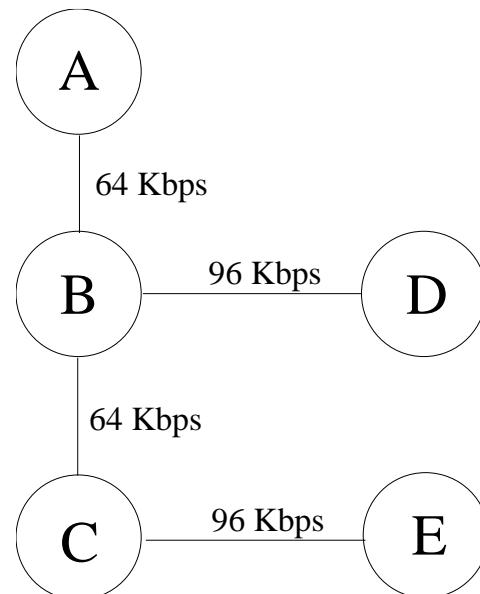
Partindo de  $\mathbf{L}^0 = (1, 1, \dots, 1)$ , o método dá origem a  $\mathbf{L}^1 = (0, 0, \dots, 0)$ ,  $\mathbf{L}^{2n}$  converge para um  $\mathbf{L}^+$  e  $\mathbf{L}^{2n+1}$  converge para um  $\mathbf{L}^-$  tal que  $\mathbf{L}^- \leq \mathbf{L}^* \leq \mathbf{L}^+$ .

As sucessivas iterações  $m$  determinam majorantes da solução  $\mathbf{L}^*$  quando  $m$  é par e minorantes da solução  $\mathbf{L}^*$  quando  $m$  é ímpar. Termina-se o algoritmo quando os dois limites estão suficientemente próximos.

## Exemplo 2

Considere a rede da figura que suporta: fluxo 1 entre A e D, fluxo 2 entre C e D e fluxo 3 entre E e B. As chamadas chegam de acordo com processos de Poisson com taxa  $\lambda_1 = 10$  chamadas/hora,  $\lambda_2 = 30$  chamadas/hora e  $\lambda_3 = 20$  chamadas/hora. Em todos os fluxos, a duração das chamadas é exponencialmente distribuída com média  $1/\mu = 3$  minutos e cada chamada requer uma largura de banda de 32 Kbps.

Calcule a probabilidade de bloqueio de cada fluxo segundo o teorema do limite do produto.



## Exemplo 2

- Fluxo 1 entre A e D, fluxo 2 entre C e D e fluxo 3 entre E e B
- $\lambda_1 = 10 \text{ cham/h}$ ,  $\lambda_2 = 30 \text{ cham/h}$  e  $\lambda_3 = 20 \text{ cham/h}$
- $1/\mu = 3 \text{ minutos}$ , cada chamada requer 32 Kbps

$$\bar{\rho}_j = \sum_{k \in K_j} \rho_k$$

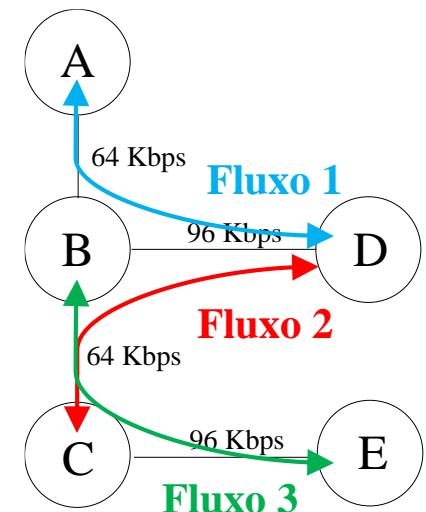
$$B_k \leq 1 - \prod_{j \in R_k} \left( 1 - ER[\bar{\rho}_j, C_j] \right)$$

$$E[\rho, C] = \frac{\rho^c / C!}{\sum_{n=0}^c \rho^n / n!}$$

$$\rho_1 = \frac{\lambda_1}{\mu_1} = \frac{10 \times 3}{60} = 0.5 \text{ Erl.} \quad R_1 = \{AB, BD\}$$

$$\rho_2 = \frac{\lambda_2}{\mu_2} = \frac{30 \times 3}{60} = 1.5 \text{ Erl.} \quad R_2 = \{BC, BD\}$$

$$\rho_3 = \frac{\lambda_3}{\mu_3} = \frac{20 \times 3}{60} = 1 \text{ Erl.} \quad R_3 = \{BC, CE\}$$



$$\bar{\rho}_{AB} = \rho_1 = 0.5 \text{ Erl.}$$

$$\bar{\rho}_{BD} = \rho_1 + \rho_2 = 2 \text{ Erl.}$$

$$\bar{\rho}_{BC} = \rho_2 + \rho_3 = 2.5 \text{ Erl.}$$

$$\bar{\rho}_{CE} = \rho_3 = 1 \text{ Erl.}$$

$$C_{AB} = 64/32 = 2 \text{ cham.}$$

$$C_{BD} = 96/32 = 3 \text{ cham.}$$

$$C_{BC} = 64/32 = 2 \text{ cham.}$$

$$C_{CE} = 96/32 = 3 \text{ cham.}$$

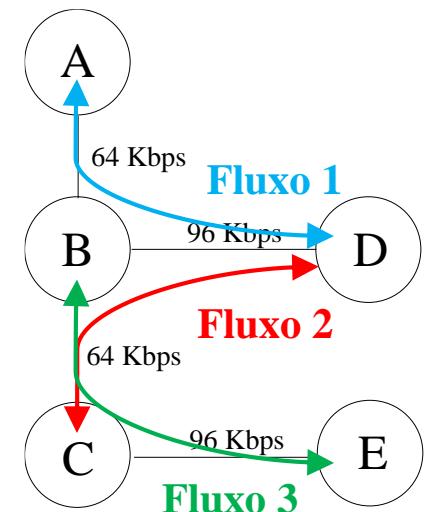
$$B_1 \leq 1 - \left( 1 - ER[\bar{\rho}_{AB}, C_{AB}] \right) \times \left( 1 - ER[\bar{\rho}_{BD}, C_{BD}] \right)$$

$$B_1 \leq 1 - \left( 1 - \frac{\frac{0.5^2}{2!}}{\frac{0.5^0}{0!} + \frac{0.5^1}{1!} + \frac{0.5^2}{2!}} \right) \times \left( 1 - \frac{\frac{2^3}{3!}}{\frac{2^0}{0!} + \frac{2^1}{1!} + \frac{2^2}{2!} + \frac{2^3}{3!}} \right)$$

$$B_1 \leq 0.2713 = \underline{27.13\%}$$

## Exemplo 2

- Fluxo 1 entre A e D, fluxo 2 entre C e D e fluxo 3 entre E e B
- $\lambda_1 = 10 \text{ cham/h}$ ,  $\lambda_2 = 30 \text{ cham/h}$  e  $\lambda_3 = 20 \text{ cham/h}$
- $1/\mu = 3 \text{ minutos}$ , cada chamada requer 32 Kbps



$$\bar{\rho}_j = \sum_{k \in K_j} \rho_k$$

$$B_k \leq 1 - \prod_{j \in R_k} \left( 1 - ER[\bar{\rho}_j, C_j] \right)$$

$$E[\rho, C] = \frac{\rho^c / C!}{\sum_{n=0}^c \rho^n / n!}$$

$$\rho_1 = \frac{\lambda_1}{\mu_1} = \frac{10 \times 3}{60} = 0.5 \text{ Erl.} \quad R_1 = \{AB, BD\}$$

$$\rho_2 = \frac{\lambda_2}{\mu_2} = \frac{30 \times 3}{60} = 1.5 \text{ Erl.} \quad R_2 = \{BC, BD\}$$

$$\rho_3 = \frac{\lambda_3}{\mu_3} = \frac{20 \times 3}{60} = 1 \text{ Erl.} \quad R_3 = \{BC, CE\}$$

$$\bar{\rho}_{AB} = \rho_1 = 0.5 \text{ Erl.}$$

$$C_{AB} = 64/32 = 2 \text{ cham.}$$

$$\bar{\rho}_{BD} = \rho_1 + \rho_2 = 2 \text{ Erl.}$$

$$C_{BD} = 96/32 = 3 \text{ cham.}$$

$$\bar{\rho}_{BC} = \rho_2 + \rho_3 = 2.5 \text{ Erl.}$$

$$C_{BC} = 64/32 = 2 \text{ cham.}$$

$$\bar{\rho}_{CE} = \rho_3 = 1 \text{ Erl.}$$

$$C_{CE} = 96/32 = 3 \text{ cham.}$$

$$B_2 \leq 1 - \left( 1 - ER[\bar{\rho}_{BC}, C_{BC}] \right) \times \left( 1 - ER[\bar{\rho}_{BD}, C_{BD}] \right)$$

$$B_2 \leq 1 - \left( 1 - \frac{\frac{2.5^2}{2!}}{\frac{2.5^0}{0!} + \frac{2.5^1}{1!} + \frac{2.5^2}{2!}} \right) \times \left( 1 - \frac{\frac{2^3}{3!}}{\frac{2^0}{0!} + \frac{2^1}{1!} + \frac{2^2}{2!} + \frac{2^3}{3!}} \right)$$

$$B_2 \leq 0.5829 = \underline{58.29\%}$$

## Exemplo 2

- Fluxo 1 entre A e D, fluxo 2 entre C e D e fluxo 3 entre E e B
- $\lambda_1 = 10 \text{ cham/h}$ ,  $\lambda_2 = 30 \text{ cham/h}$  e  $\lambda_3 = 20 \text{ cham/h}$
- $1/\mu = 3 \text{ minutos}$ , cada chamada requer 32 Kbps

$$\bar{\rho}_j = \sum_{k \in K_j} \rho_k$$

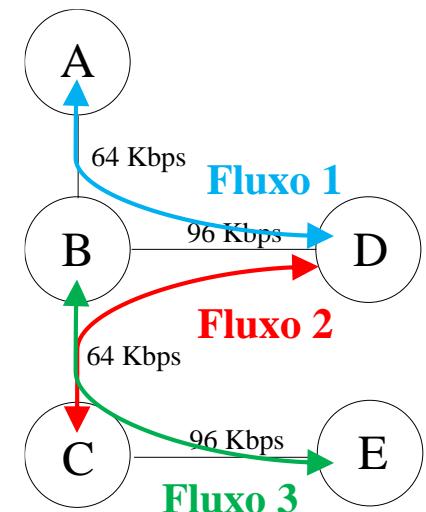
$$B_k \leq 1 - \prod_{j \in R_k} \left( 1 - ER[\bar{\rho}_j, C_j] \right)$$

$$E[\rho, C] = \frac{\rho^c / C!}{\sum_{n=0}^c \rho^n / n!}$$

$$\rho_1 = \frac{\lambda_1}{\mu_1} = \frac{10 \times 3}{60} = 0.5 \text{ Erl.} \quad R_1 = \{AB, BD\}$$

$$\rho_2 = \frac{\lambda_2}{\mu_2} = \frac{30 \times 3}{60} = 1.5 \text{ Erl.} \quad R_2 = \{BC, BD\}$$

$$\rho_3 = \frac{\lambda_3}{\mu_3} = \frac{20 \times 3}{60} = 1 \text{ Erl.} \quad R_3 = \{BC, CE\}$$



$$\bar{\rho}_{AB} = \rho_1 = 0.5 \text{ Erl.}$$

$$\bar{\rho}_{BD} = \rho_1 + \rho_2 = 2 \text{ Erl.}$$

$$\bar{\rho}_{BC} = \rho_2 + \rho_3 = 2.5 \text{ Erl.}$$

$$\bar{\rho}_{CE} = \rho_3 = 1 \text{ Erl.}$$

$$C_{AB} = 64/32 = 2 \text{ cham.}$$

$$C_{BD} = 96/32 = 3 \text{ cham.}$$

$$C_{BC} = 64/32 = 2 \text{ cham.}$$

$$C_{CE} = 96/32 = 3 \text{ cham.}$$

$$B_3 \leq 1 - \left( 1 - ER[\bar{\rho}_{BC}, C_{BC}] \right) \times \left( 1 - ER[\bar{\rho}_{CE}, C_{CE}] \right)$$

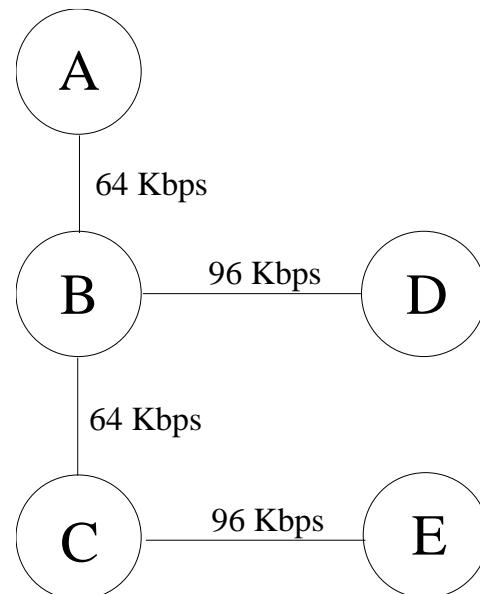
$$B_3 \leq 1 - \left( 1 - \frac{\frac{2.5^2}{2!}}{\frac{2.5^0}{0!} + \frac{2.5^1}{1!} + \frac{2.5^2}{2!}} \right) \times \left( 1 - \frac{\frac{1^3}{3!}}{\frac{1^0}{0!} + \frac{1^1}{1!} + \frac{1^2}{2!} + \frac{1^3}{3!}} \right)$$

$$B_3 \leq 0.5057 = \underline{50.47\%}$$

## Exemplo 3

Considere a rede da figura que suporta: fluxo 1 entre A e D, fluxo 2 entre C e D e fluxo 3 entre E e B. As chamadas chegam de acordo com processos de Poisson com taxa  $\lambda_1 = 10$  chamadas/hora,  $\lambda_2 = 30$  chamadas/hora e  $\lambda_3 = 20$  chamadas/hora. Em todos os fluxos, a duração das chamadas é exponencialmente distribuída com média  $1/\mu = 3$  minutos e cada chamada requer uma largura de banda de 32 Kbps.

Escreva as equações que permitem calcular a probabilidade de bloqueio de cada fluxo através da aproximação de carga reduzida.



## Exemplo 3

- Fluxo 1 entre A e D, fluxo 2 entre C e D e fluxo 3 entre E e B
- $\lambda_1 = 10 \text{ cham/h}$ ,  $\lambda_2 = 30 \text{ cham/h}$  e  $\lambda_3 = 20 \text{ cham/h}$
- $1/\mu = 3 \text{ minutos}$ , cada chamada requer 32 Kbps

$$L_j = ER \left[ \sum_{k \in K_j} \rho_k \prod_{i \in R_k - \{j\}} (1 - L_i), C_j \right], j = 1, 2, \dots, J$$

$$B_k \approx 1 - \prod_{j \in R_k} (1 - L_j) \quad k = 1, 2, \dots, K$$

$$\rho_1 = \frac{\lambda_1}{\mu_1} = \frac{10 \times 3}{60} = 0.5 \text{ Erl.}$$

$$\rho_2 = \frac{\lambda_2}{\mu_2} = \frac{30 \times 3}{60} = 1.5 \text{ Erl.}$$

$$\rho_3 = \frac{\lambda_3}{\mu_3} = \frac{20 \times 3}{60} = 1 \text{ Erl.}$$

~~$\rho_{AB} = \rho_1 = 0.5 \text{ Erl.}$~~

~~$\rho_{BD} = \rho_1 + \rho_2 = 2 \text{ Erl.}$~~

~~$\rho_{BC} = \rho_2 + \rho_3 = 2.5 \text{ Erl.}$~~

~~$\rho_{CE} = \rho_3 = 1 \text{ Erl.}$~~

$$R_1 = \{AB, BD\}$$

$$R_2 = \{BC, BD\}$$

$$R_3 = \{BC, CE\}$$

$$C_{AB} = 64 / 32 = 2 \text{ cham.}$$

$$C_{BD} = 96 / 32 = 3 \text{ cham.}$$

$$C_{BC} = 64 / 32 = 2 \text{ cham.}$$

$$C_{CE} = 96 / 32 = 3 \text{ cham.}$$

$$\begin{aligned} L_{AB} &= ER[\rho_1 \times (1 - L_{BD}), C_{AB}] \\ &= ER[0.5 \times (1 - L_{BD}), 2] \end{aligned}$$

$$\begin{aligned} L_{BD} &= ER[\rho_1 \times (1 - L_{AB}) + \rho_2 \times (1 - L_{BC}), C_{BD}] \\ &= ER[0.5 \times (1 - L_{AB}) + 1.5 \times (1 - L_{BC}), 3] \end{aligned}$$

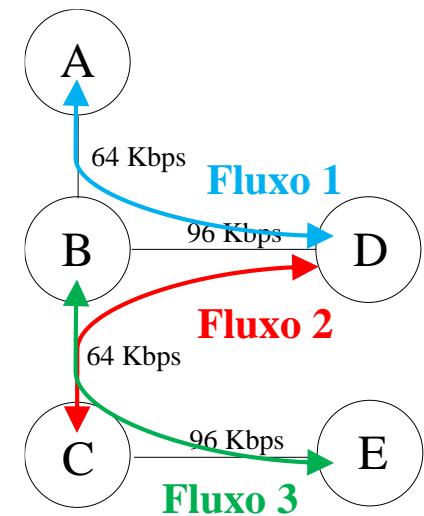
$$\begin{aligned} L_{BC} &= ER[\rho_2 \times (1 - L_{BD}) + \rho_3 \times (1 - L_{CE}), C_{BC}] \\ &= ER[1.5 \times (1 - L_{BD}) + (1 - L_{CE}), 2] \end{aligned}$$

$$\begin{aligned} L_{CE} &= ER[\rho_3 \times (1 - L_{BC}), C_{CE}] \\ &= ER[(1 - L_{BC}), 3] \end{aligned}$$

$$B_1 = 1 - (1 - L_{AB}) \times (1 - L_{BD})$$

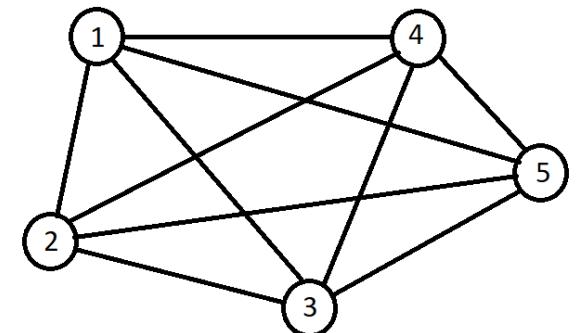
$$B_2 = 1 - (1 - L_{BC}) \times (1 - L_{BD})$$

$$B_3 = 1 - (1 - L_{BC}) \times (1 - L_{CE})$$



# Encaminhamento dinâmico da rede telefónica

- Os métodos de encaminhamento dinâmico eram usados nas redes de transporte dos operadores telefónicos.
- Estas redes tinham conectividade total, ou seja, incluíam uma ligação direta entre todos os pares de nós da rede.
- Assim, numa rede com  $N$  nós:
  - existem  $N \times (N - 1) / 2$  ligações
  - o número de percursos com apenas duas ligações entre quaisquer duas centrais é  $N - 2$ .



- Numa rede com conectividade total, o percurso direto é o percurso pela ligação direta do nó origem para o nó destino.
- É sempre preferível encaminhar uma chamada pelo percurso direto (os percursos alternativos consomem mais recursos).
- Trunk reservation: reserva de um conjunto de circuitos em cada ligação para chamadas no percurso direto (limita o excesso de encaminhamento alternativo).

# **Encaminhamento dinâmico da rede telefónica**

Os métodos de encaminhamento dinâmico têm as seguintes características em comum:

1. quando é pedido o estabelecimento de uma chamada entre duas centrais, a chamada é encaminhada no percurso direto, se houver pelo menos um circuito disponível;
2. quando o percurso direto está indisponível, a chamada pode ser encaminhada num dos percursos alternativos permitidos;
3. os percursos alternativos têm sempre apenas duas ligações, ou seja, as chamadas nunca são estabelecidas em percursos com três ou mais ligações.

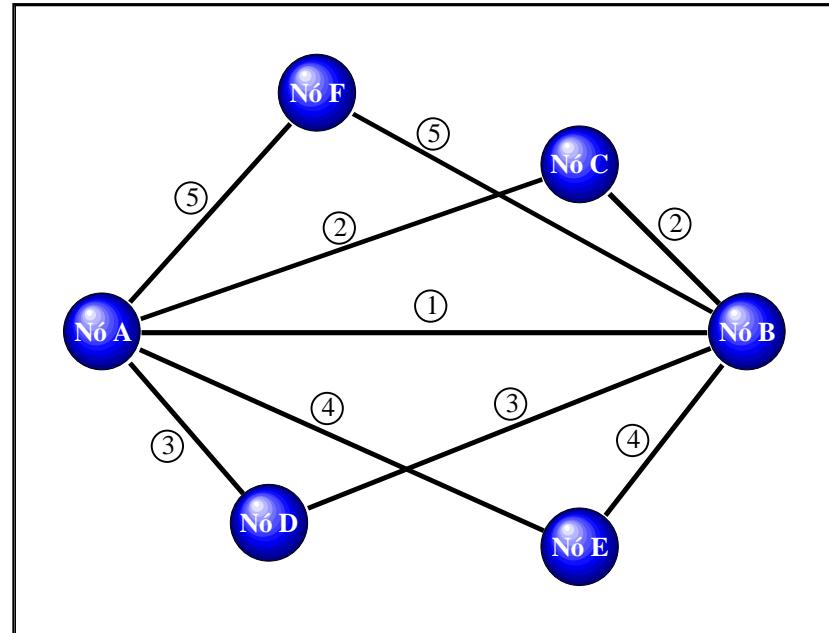
Os diferentes métodos de encaminhamento diferem na forma como é definido, em cada instante, o conjunto de percursos alternativos.

# Encaminhamento sequencial

- É a base do protocolo introduzido pela AT&T nos anos 80 designado por DNHR - Dynamic Non-Hierarchical Routing.
- A cada par de centrais origem-destino associa-se uma lista ordenada de percursos alternativos. Se um pedido de chamada encontra o percurso direto indisponível:
  - 1) a chamada é estabelecida no primeiro percurso alternativo disponível da lista ordenada;
  - 2) se todos os percursos alternativos estiverem indisponíveis a chamada é bloqueada.
- Os diferentes parâmetros da rede (a lista de percursos alternativos, a ordenação dos percursos na lista, a percentagem de circuitos reservados em cada ligação, etc...) variam no tempo, sendo considerados até 10 períodos distintos em cada dia.
- Quando a segunda ligação de um percurso alternativo está indisponível, a central intermédia tem de sinalizar a central origem dessa ocorrência para que esta possa tentar outro percurso alternativo (função designada por crankback).

# DNHR

Nó A → Nó B



Lista #N	Percursos recomendados presentes na lista	Período de tempo
Listá #1	1→3→2→4	10:00– 12:00
Listá #2	1→4→2	12:00 – 17:00
Listá #3	1→3→5→2	17:00– 19:00
Listá #4	1→5→4	19:00– 23:00
Listá #5	1→2→3	23:00 – 10:00

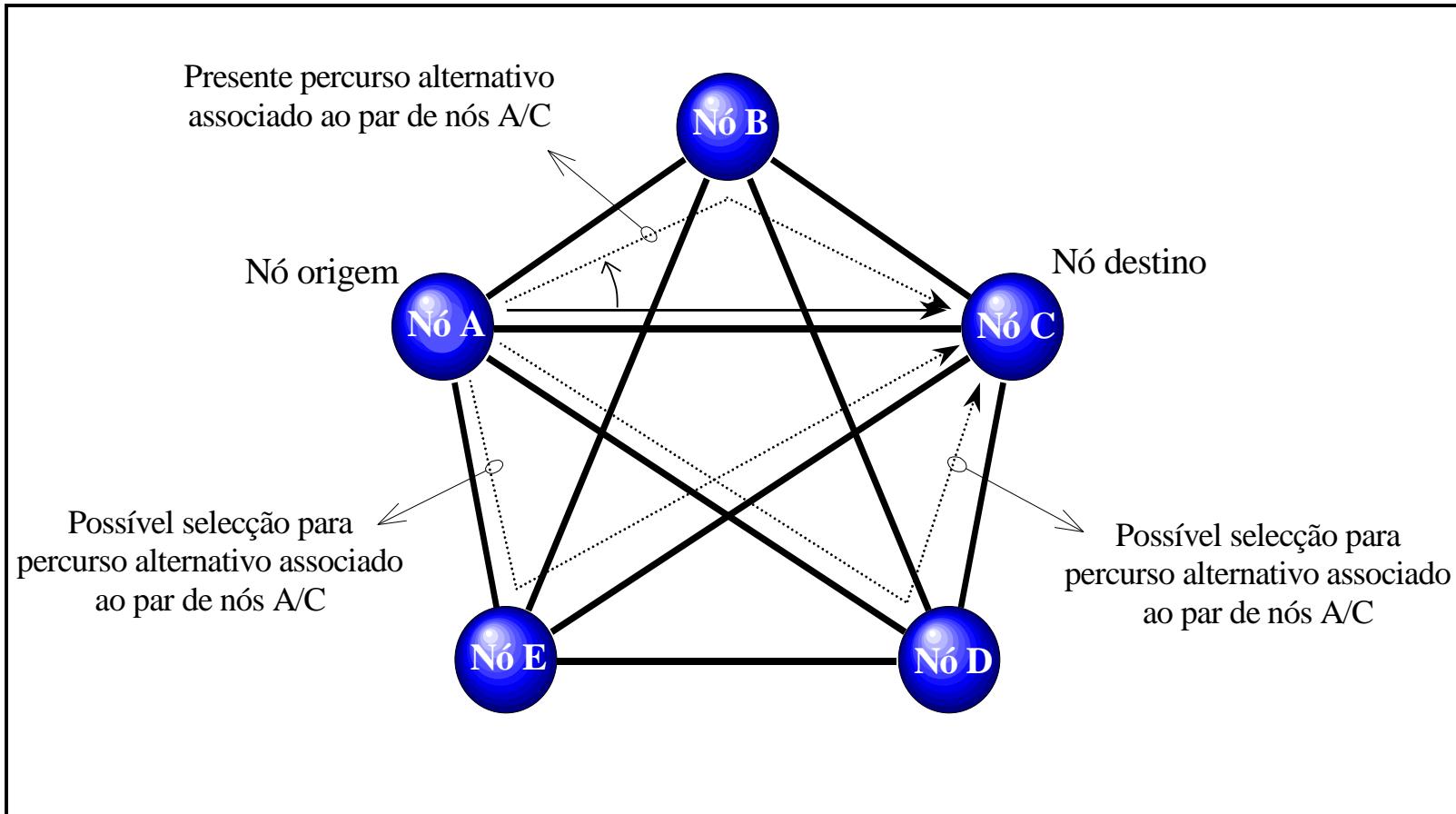
# Encaminhamento aleatório retardado

- Este método é a base do protocolo DAR - Dynamic Alternative Routing introduzido pelos British Telecom nos anos 90.
- Associa-se a cada par de centrais origem-destino um percurso alternativo (e apenas um):
  - 1) se quando uma chamada chega o percurso direto está indisponível e o percurso alternativo está disponível a chamada é estabelecida no percurso alternativo;
  - 2) caso contrário, a chamada é bloqueada e é escolhido um novo percurso alternativo para as chamadas subsequentes;
  - 3) o novo percurso alternativo é escolhido aleatoriamente de entre os  $N - 3$  percursos alternativos restantes.

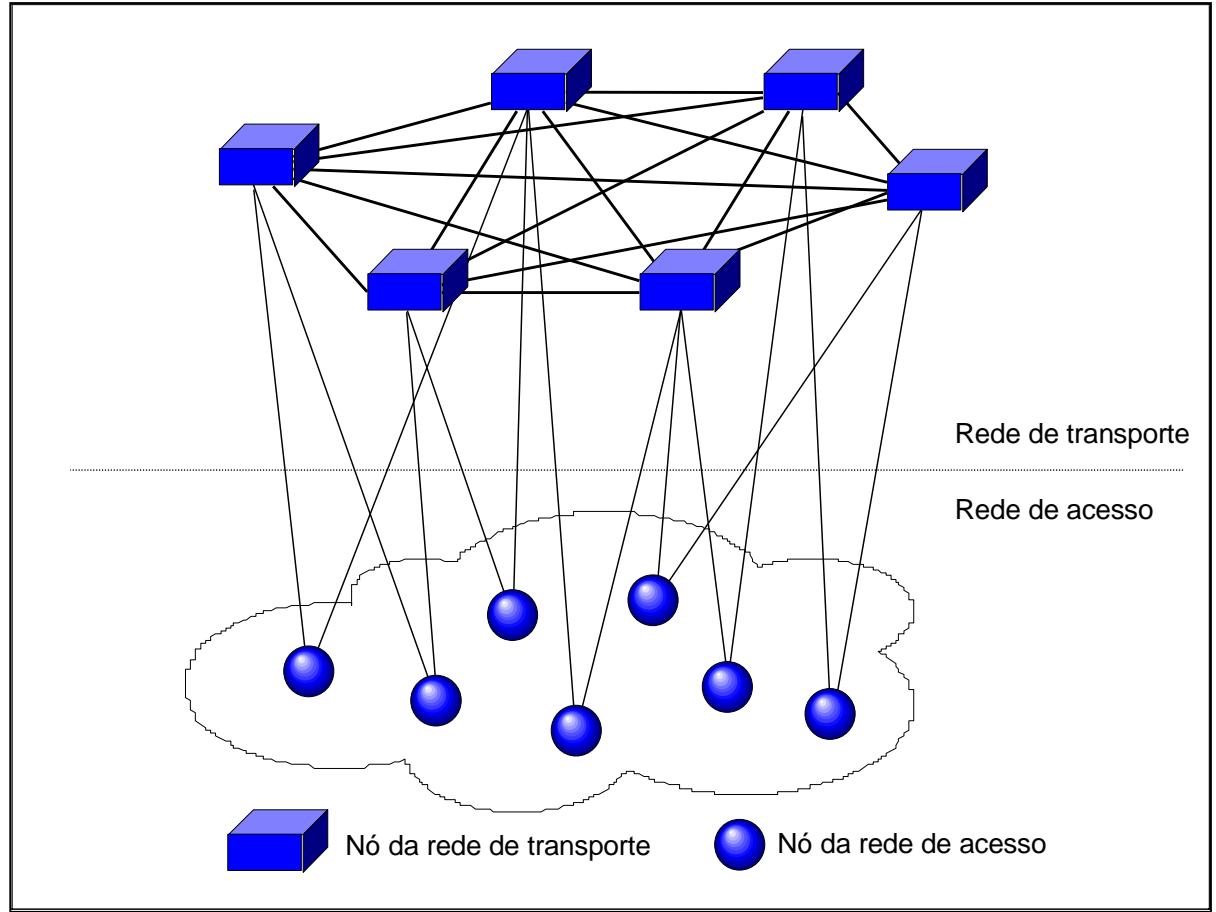
Relativamente ao DHNR:

- não é necessário implementar a função de *crankback* dado que apenas é tentado um percurso alternativo (os mecanismos de sinalização são mais simples)
- adapta-se dinamicamente ao tráfego sem necessidade de gestão centralizada dos percursos alternativos

# Operação do protocolo DAR

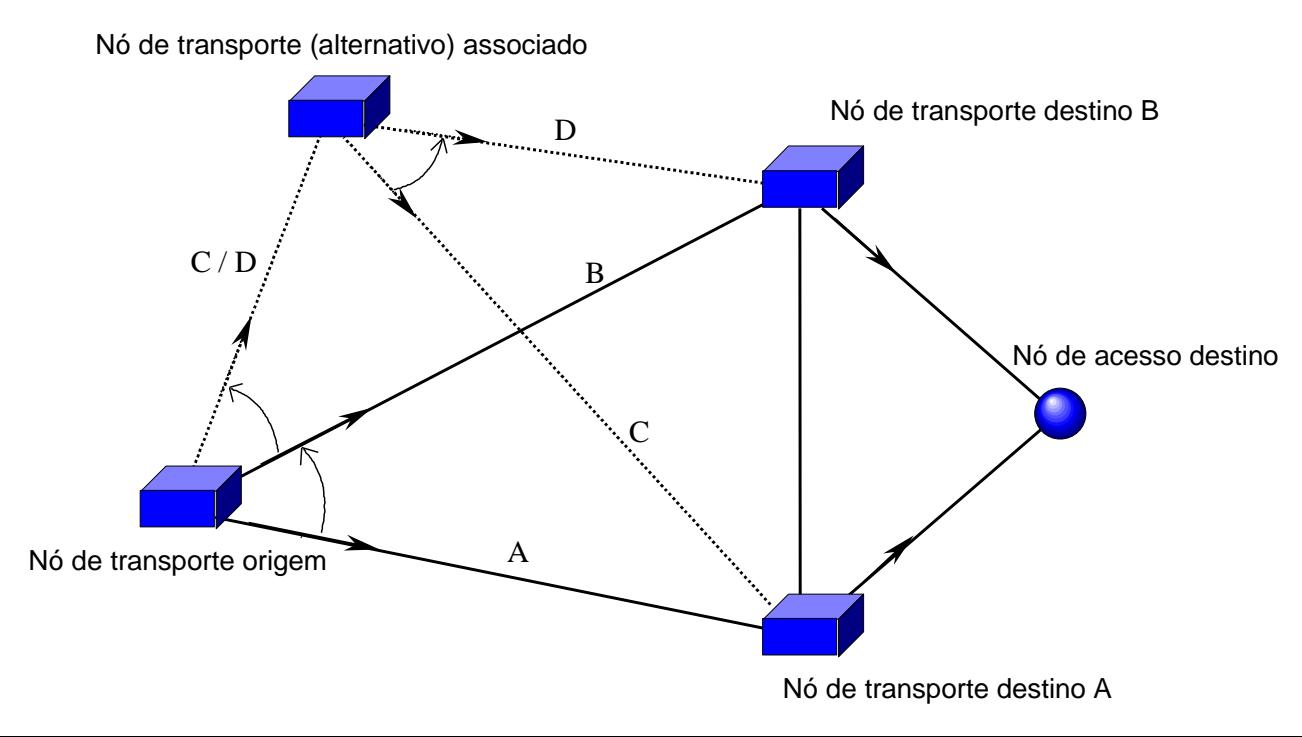


# DAR na rede da British Telecom



- Rede organizada em rede de transporte e rede de acesso.
- Rede de transporte com conectividade física total.
- Cada nó de acesso com ligação física a dois nós de transporte (para proteção de falha individual de ligação).

# DAR na rede da British Telecom



- No nó de transporte onde chega o pedido de chamada existem 2 percursos diretos e um nó de transporte alternativo.
- Primeiro são tentados os dois percursos diretos.
- Depois são tentados os dois percursos via nó alternativo.
- Se a chamada bloquear, outro nó alternativo é escolhido para o próximo pedido de chamada para o mesmo destino.

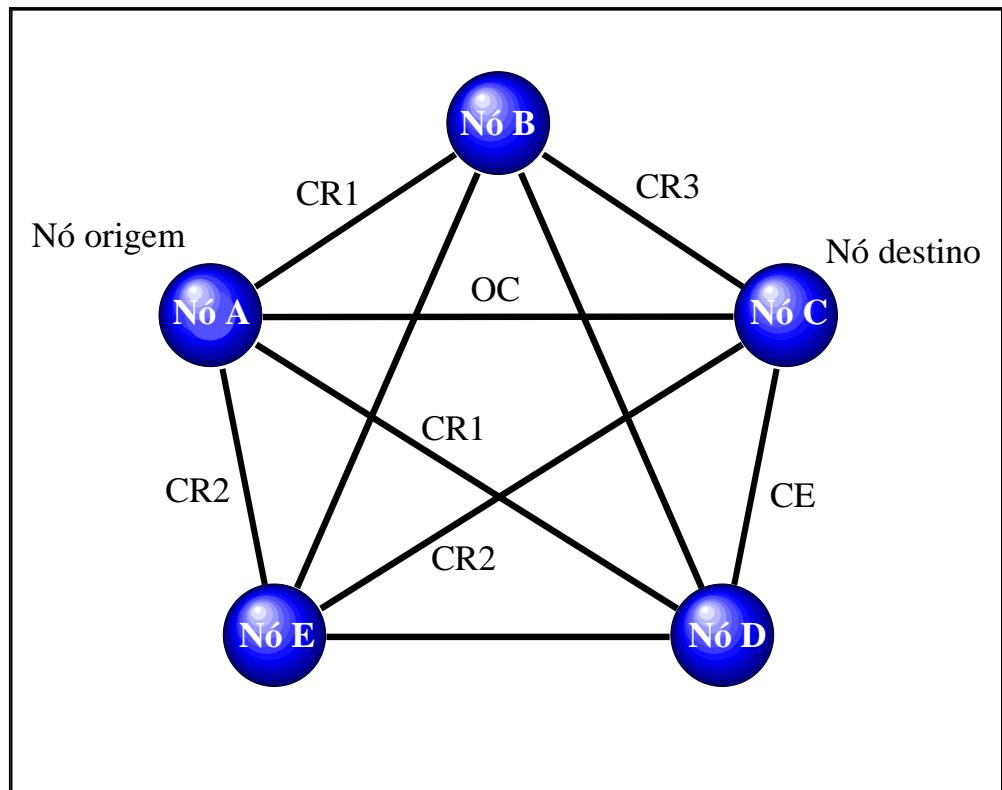
## Encaminhamento de menor carga

- Este método mantém um registo da capacidade não utilizada em cada percurso alternativo (corresponde ao número de circuitos não ocupados, para lá dos circuitos reservados).
- Quando o percurso direto está indisponível, escolhe o percurso alternativo de menor carga máxima; se todos os percursos alternativos estiverem indisponíveis a chamada é bloqueada.

Este método está na base do protocolo RTNR - Real-Time Network Routing introduzido pela AT&T no início dos anos 90:

- 1) quando o percurso direto está indisponível, a central origem pergunta à central destino qual a capacidade não utilizada de todas as suas ligações;
  - 2) após receber essa informação, a central origem determina qual o percurso alternativo de menor carga máxima, com base na informação de que dispõe sobre a capacidade não utilizada das suas próprias ligações.
- Este método é mais eficiente que o DAR mas exibe uma maior complexidade nos mecanismos de sinalização.

# RTNR



## Níveis de carga considerados:

CR1 - Carga Reduzida de nível 1

CR2 - Carga Reduzida de nível 2

CR3 - Carga Reduzida de nível 3

CE - Carga Elevada

RS – Reservado

OC - Ocupado

No cenário da figura, chega um pedido de chamada:

**Nó A → Nó C**

- O percurso direto está Ocupado.
- A chamada é estabelecida por E pois neste caso a carga máxima das duas ligações ( $A \rightarrow E$  e  $E \rightarrow C$ ) é CR2.

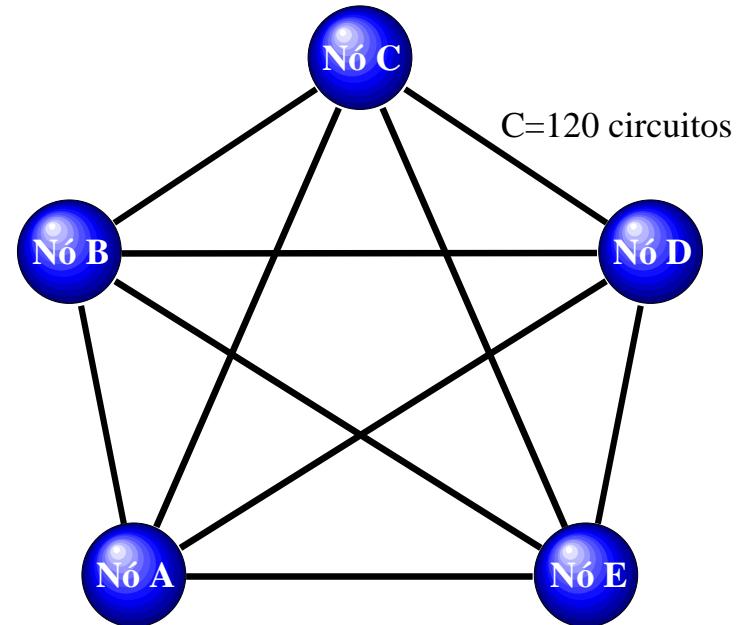
# Metaestabilidade e reserva de recursos

Reserva de recursos  $r$  para percursos diretos numa ligação de capacidade  $C$ : número de circuitos que só pode ser ocupado por percursos diretos (i.e., percursos alternativos podem ocupar no máximo  $C - r$ ).

Considere-se o exemplo da figura:

- rede com conectividade total;
- encaminhamento aleatório global;
- tráfego oferecido igual para todos os pares origem-destino;
- mesma reserva de recursos  $r$  em todas as ligações.

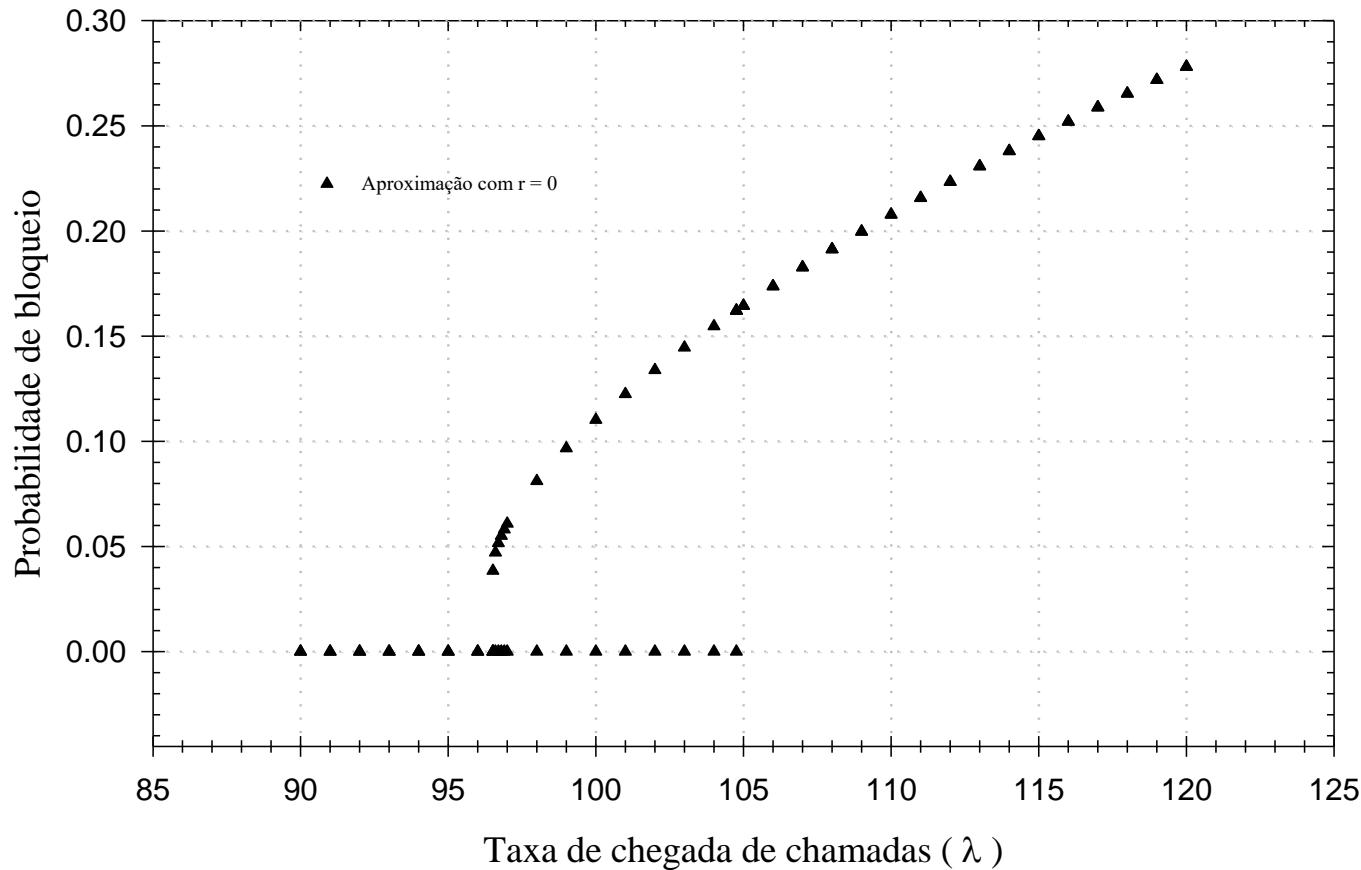
O desempenho desta rede pode ser calculado de forma aproximada por processos analíticos.



# Desempenho sem reserva de recursos ( $r=0$ )

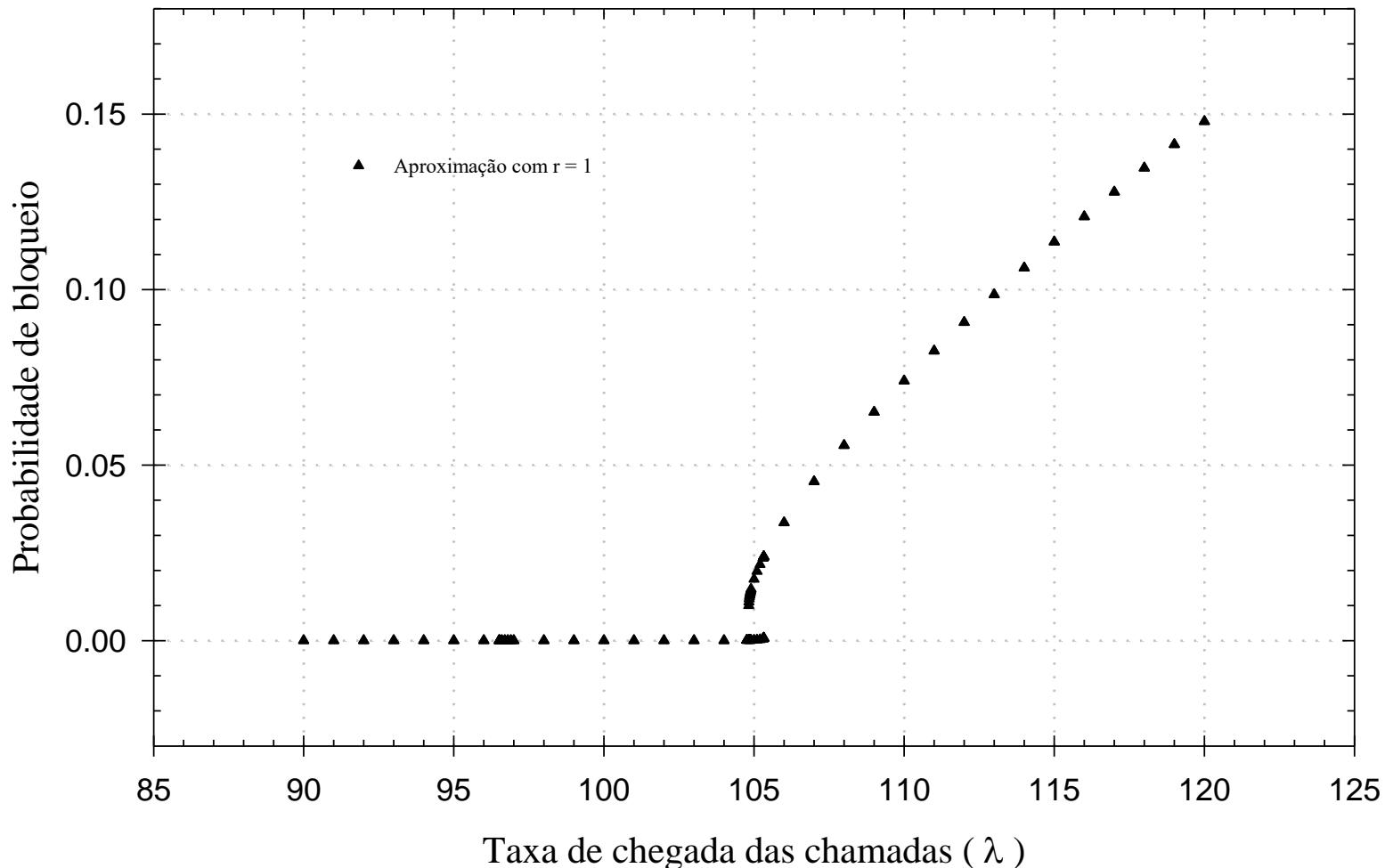
Instabilidade para valores de  $\lambda$  entre 96 e 105:

- Existem intervalos de tempo em que a maior parte do tráfego é encaminhada pelo percurso direto (probabilidade de bloqueio desprezável).
- Existem intervalos de tempo em que a maior parte do tráfego é encaminhada pelos percursos alternativos (probabilidade de bloqueio elevada).



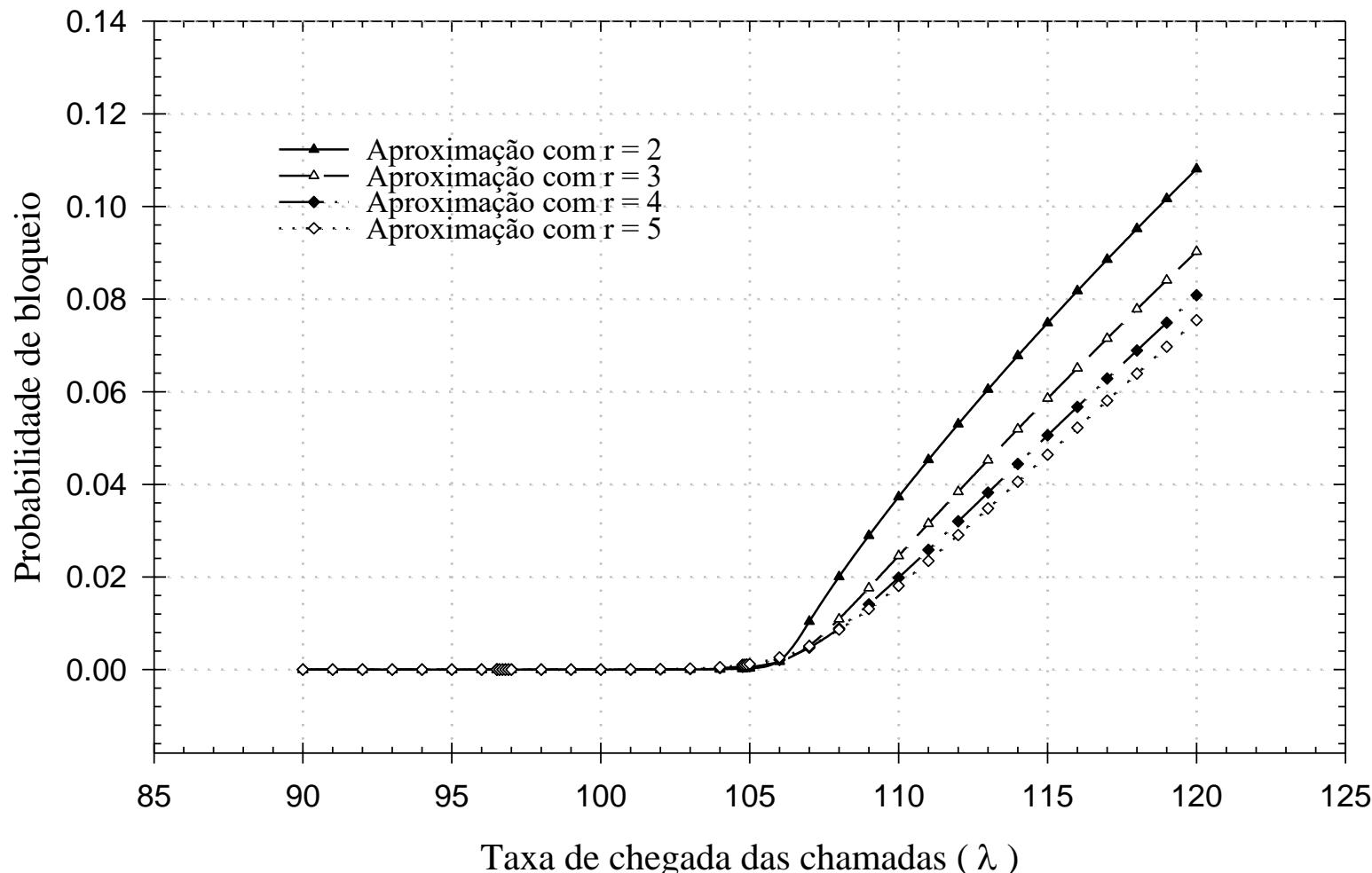
# Desempenho com reserva de um circuito ( $r=1$ )

A instabilidade é reduzida com uma reserva de 1 em 120 circuitos



# Desempenho com reserva de 2 ou mais circuitos

Valores crescentes de reserva anulam a instabilidade e aumentam o desempenho da rede (diminuem a percentagem de tráfego que usa percursos alternativos)





# **Optimization based on Integer Linear Programming**

Desempenho e Dimensionamento de Redes

Prof. Amaro de Sousa ([asou@ua.pt](mailto:asou@ua.pt))

DETI-UA, 2020/2021

# Mathematical programming model

- In an *optimization problem*, the aim is to maximize (or minimize) a given quantity designated by the *objective* that depends on a finite number of variables.
- The variables might be independent or might be related between them through one or more *constraints*.
- A *mathematical programming problem* is an optimization problem such that the objective and the constraints are defined by mathematical functions and functional relations.
- A *mathematical programming model* describes a mathematical programming problem.

# Mathematical programming model

For a given set of  $n$  variables  $X = \{x_1, x_2, \dots, x_n\}$ , the standard way of defining a Mathematical Programming Model is:

Minimize (or Maximize)

$$f(X)$$

Subject to:

$$g_i(X) \leq k_i \quad , i = 1, 2, \dots, m$$

(=)

( $\geq$ )

where:

- $m$  is the number of constraints
- $f(X)$  and all  $g_i(X)$  are functions of the variables
- $k_i$  are real constants

# (Mixed Integer) Linear Programming model

- A *Linear Programming (LP) model* is a mathematical programming model where all variables  $X = \{x_1, x_2, \dots, x_n\}$  are non-negative reals and  $f(X)$  and  $g_i(X)$  are linear functions:
  - functions in the form  $a_1x_1 + a_2x_2 + \dots + a_nx_n$  where all  $a_i$  are real parameters.
- An *Integer Linear Programming (ILP) model* is an LP model where all variables  $X = \{x_1, x_2, \dots, x_n\}$  are non-negative integers.
- A *Mixed Integer Linear Programming (MILP) model* is an LP model where some variables  $X = \{x_1, x_2, \dots, x_n\}$  are non-negative integers and others are non-negative reals.

## Illustrative example

Consider a transportation company that has been requested to deliver the following items to a particular destination:

Item $i$ :	1	2	3	4	5	6
Revenue ( $r_i$ ):	2.3	4.5	1.5	5.4	2.9	3.2
Size ( $s_i$ ):	30	70	20	80	35	40

The company has 2 vans for item delivery:

- the first van has a capacity of 100
- the second van has a capacity of 60.

Since it is not possible to deliver all items with the 2 vans, the aim is to choose the items to be carried on each van to maximize the revenue.

Solving steps:

1<sup>st</sup> - define the ILP model of the optimization problem

2<sup>nd</sup> – solve the ILP model (using an available solver)

# Illustrative example

Item $i$ :	1	2	3	4	5	6
Revenue ( $r_i$ ):	2.3	4.5	1.5	5.4	2.9	3.2
Size ( $s_i$ ):	30	70	20	80	35	40

## VARIABLES DEFINING THE PROBLEM:

- $x_1$  – Binary variable that, if is 1 in the solution, indicates that item 1 is delivered
- $x_2$  – Binary variable that, if is 1 in the solution, indicates that item 2 is delivered
- ...
- $x_6$  – Binary variable that, if is 1 in the solution, indicates that item 6 is delivered

- $y_{1\_1}$  – Binary variable that, if is 1 in the solution, indicates that item 1 is carried by first van
- $y_{1\_2}$  – Binary variable that, if is 1 in the solution, indicates that item 1 is carried by second van
- ...
- $y_{6\_1}$  – Binary variable that, if is 1 in the solution, indicates that item 6 is carried by first van
- $y_{6\_2}$  – Binary variable that, if is 1 in the solution, indicates that item 6 is carried by second van

# Illustrative example

Item $i$ :	1	2	3	4	5	6
Revenue ( $r_i$ ):	2.3	4.5	1.5	5.4	2.9	3.2
Size ( $s_i$ ):	30	70	20	80	35	40

INTEGER LINEAR PROGRAMMING (ILP) MODEL (in LP format):

The objective function is the total revenue  
of the delivered items

Maximize

$$+ 2.3 x_1 + 4.5 x_2 + 1.5 x_3 + 5.4 x_4 + 2.9 x_5 + 3.2 x_6$$

Subject To

$$+ 30 y_{1\_1} + 70 y_{2\_1} + 20 y_{3\_1} + 80 y_{4\_1} + 35 y_{5\_1} + 40 y_{6\_1} \leq 100$$

$$+ 30 y_{1\_2} + 70 y_{2\_2} + 20 y_{3\_2} + 80 y_{4\_2} + 35 y_{5\_2} + 40 y_{6\_2} \leq 60$$

$$+ y_{1\_1} + y_{1\_2} - x_1 = 0$$

$$+ y_{2\_1} + y_{2\_2} - x_2 = 0$$

$$+ y_{3\_1} + y_{3\_2} - x_3 = 0$$

$$+ y_{4\_1} + y_{4\_2} - x_4 = 0$$

$$+ y_{5\_1} + y_{5\_2} - x_5 = 0$$

$$+ y_{6\_1} + y_{6\_2} - x_6 = 0$$

The total size of the items carried on each van must be within the van capacity

If an item is carried in one van, then, the item is delivered

Binary

$$x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6$$

$$y_{1\_1} \ y_{1\_2} \ y_{2\_1} \ y_{2\_2} \ y_{3\_1} \ y_{3\_2} \ y_{4\_1} \ y_{4\_2} \ y_{5\_1} \ y_{5\_2} \ y_{6\_1} \ y_{6\_2}$$

End

List of binary variables

# **Illustrative example – using CPLEX (1)**

## **Starting CPLEX:**

```
Welcome to IBM(R) ILOG(R) CPLEX(R) Interactive Optimizer 12.6.1.0
with Simplex, Mixed Integer & Barrier Optimizers
5725-A06 5725-A29 5724-Y48 5724-Y49 5724-Y54 5724-Y55 5655-Y21
Copyright IBM Corp. 1988, 2014. All Rights Reserved.
```

```
Type 'help' for a list of available commands.
Type 'help' followed by a command name for more
information on commands.
```

```
CPLEX>
```

## **Reading file ‘exemplo.lp’ on CPLEX:**

```
CPLEX> read exemplo.lp
Problem 'exemplo.lp' read.
Read time = 0.01 sec. (0.00 ticks)
CPLEX>
```

## Illustrative example – using CPLEX (2)

Solving the problem on CPLEX:

```
CPLEX> optimize
```

	Nodes				Cuts /		
Node	Left	Objective	IInf	Best Integer	Best Bound	ItCnt	Gap
*	0+	0		1.5000	19.8000		---
*	0+	0		11.2000	19.8000		76.79%
	0	0	11.8286	1	11.2000	11.8286	4 5.61%
*	0+	0		11.5000	11.8286		2.86%
	0	0	cutoff	11.5000		4	0.00%

Elapsed time = 0.14 sec. (1.12 ticks, tree = 0.00 MB, solutions = 3)

Root node processing (before b&c):

Real time = 0.14 sec. (1.12 ticks)

Parallel b&c, 4 threads:

Real time = 0.00 sec. (0.00 ticks)

Sync time (average) = 0.00 sec.

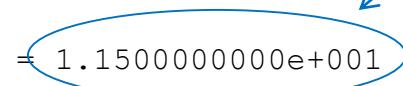
Wait time (average) = 0.00 sec.

-----  
Total (root+branch&cut) = 0.14 sec. (1.12 ticks)

Solution pool: 4 solutions saved.

MIP - Integer optimal solution: Objective = 1.1500000000e+001  
Solution time = 0.16 sec. Iterations = 4 Nodes = 0  
Deterministic time = 1.12 ticks (7.17 ticks/sec)

Optimal solution value



## Illustrative example – using CPLEX (3)

Displaying the values of the optimal solution:

Items 1, 2, 3 and 6  
are selected to be  
delivered

Items 1 and 2 are  
carried by first van

Items 3 and 6 are  
carried by second van

```
CPLEX> display solution variables -  
Incumbent solution  
Variable Name           Solution Value  
x1                      1.000000  
x2                      1.000000  
x3                      1.000000  
x6                      1.000000  
y1_1                     1.000000  
y2_1                     1.000000  
y3_2                     1.000000  
y6_2                     1.000000  
All other variables in the range 1-18 are 0.  
CPLEX>
```

Item $i$ :	1	2	3	4	5	6
Revenue ( $r_i$ ):	2.3	4.5	1.5	5.4	2.9	3.2
Size ( $s_i$ ):	30	70	20	80	35	40

## Illustrative example – mathematical notation

Parameters:

$n$  – number of items

$r_i$  – revenue of delivering item  $i$ , with  $i = 1, \dots, n$

$v$  – number of vans

$s_i$  – size of item  $i$ , with  $i = 1, \dots, n$

$c_j$  – capacity of van  $j$ , with  $j = 1, \dots, v$

Variables:

$x_i$  – binary variable that is 1 if item  $i$  is delivered,  $i = 1, \dots, n$

$y_{ij}$  – binary variable that is 1 if item  $i$  is carried on van  $j$ ,  $i = 1, \dots, n$  and  $j = 1, \dots, v$

ILP model:      Maximize  $\sum_{i=1}^n r_i x_i$

Subject to:

$$\sum_{i=1}^n s_i y_{ij} \leq c_j \quad , j = 1 \dots v$$

$$\sum_{j=1}^v y_{ij} = x_i \quad , i = 1 \dots n$$

$$x_i \in \{0,1\} \quad , i = 1 \dots n$$

$$y_{ij} \in \{0,1\} \quad , i = 1 \dots n \quad , j = 1, \dots, v$$

# Illustrative example – generating LP file with MATLAB

$$\text{Maximize } \sum_{i=1}^n r_i x_i$$

$$\sum_{i=1}^n s_i y_{ij} \leq c_j , j = 1 \dots v$$

$$\sum_{j=1}^v y_{ij} = x_i , i = 1 \dots n$$

$$x_i \in \{0,1\}, i = 1 \dots n$$

$$y_{ij} \in \{0,1\}, i = 1 \dots n , j = 1, \dots v$$

```

r= [2.3 4.5 1.5 5.4 2.9 3.2];
s= [30 70 20 80 35 40];
c= [100 60];
n= length(r);
v= length(c);
fid = fopen('exemplo.lp', 'wt');
fprintf(fid, 'Maximize\n');
for i=1:n
    fprintf(fid, ' + %f x%d', r(i), i);
end
fprintf(fid, '\nSubject To\n');
for j=1:v
    for i=1:n
        fprintf(fid, ' + %f y%d_%d', s(i), i, j);
    end
    fprintf(fid, ' <= %f\n', c(j));
end
for i=1:n
    for j=1:v
        fprintf(fid, ' + y%d_%d', i, j);
    end
    fprintf(fid, ' - x%d = 0\n', i);
end
fprintf(fid, 'Binary\n');
for i=1:n
    fprintf(fid, ' x%d\n', i);
    for j=1:v
        fprintf(fid, ' y%d_%d\n', i, j);
    end
end
fprintf(fid, 'End\n');
fclose(fid);

```

# Illustrative example - using Gurobi on Internet (1)

- Prepare an ASCII file with the problem defined in LP format and compress it with Zip:  
for example: exemplo.zip
- Go to <https://neos-server.org/neos/solvers/index.html>
- Select Mixed Integer Linear Programming tools
- Select Gurobi [[LP Input](#)]

## Mixed Integer Linear Programming

- Cbc [[AMPL Input](#)][[GAMS Input](#)][[MPS Input](#)]
- CPLEX [[AMPL Input](#)][[GAMS Input](#)][[LP Input](#)][[MPS Input](#)]
- feaslp [[AMPL Input](#)][[CPLEX Input](#)][[MPS Input](#)]
- FICO-Xpress [[AMPL Input](#)][[GAMS Input](#)][[MOSEL Input](#)][[MPS Input](#)]
- Gurobi [[AMPL Input](#)][[GAMS Input](#)][[LP Input](#)][[MPS Input](#)]
- MINTO [[AMPL Input](#)]
- MOSEK [[AMPL Input](#)][[GAMS Input](#)][[LP Input](#)][[MPS Input](#)]
- proxy [[CPLEX Input](#)][[MPS Input](#)]
- qsopt\_ex [[AMPL Input](#)][[LP Input](#)][[MPS Input](#)]
- scip [[AMPL Input](#)][[CPLEX Input](#)][[GAMS Input](#)][[MPS Input](#)][[OSIL Input](#)][[ZIMPL Input](#)]
- SYMPHONY [[MPS Input](#)]

## Illustrative example - using Gurobi on Internet (2)

1 . Upload exemplo.zip

2. Check the box

3. Insert a valid  
email address

4. Submit your  
ILP problem

The screenshot shows a web-based form for submitting an Integer Linear Programming (ILP) problem. The form includes fields for an LP file, a parameter file, and a checkbox for returning a .sol file. It also features a large text area for comments and sections for additional settings and priority options. At the bottom, there is a note about not clicking the submit button more than once, and buttons for 'Submit to NEOS' and 'Clear this Form'.

LP file  
Enter the path to the LP file  
 No file chosen

Parameter file  
Enter the path to the parameter file  
 No file chosen

Return .sol file  
Check the box to include the solution file as part of the results

Comments

Additional Settings

Dry run: generate job XML instead of submitting it to NEOS

Short Priority: submit to higher priority queue with maximum CPU time of 5 minutes

E-Mail address:

Please do not click the 'Submit to NEOS' button more than once.

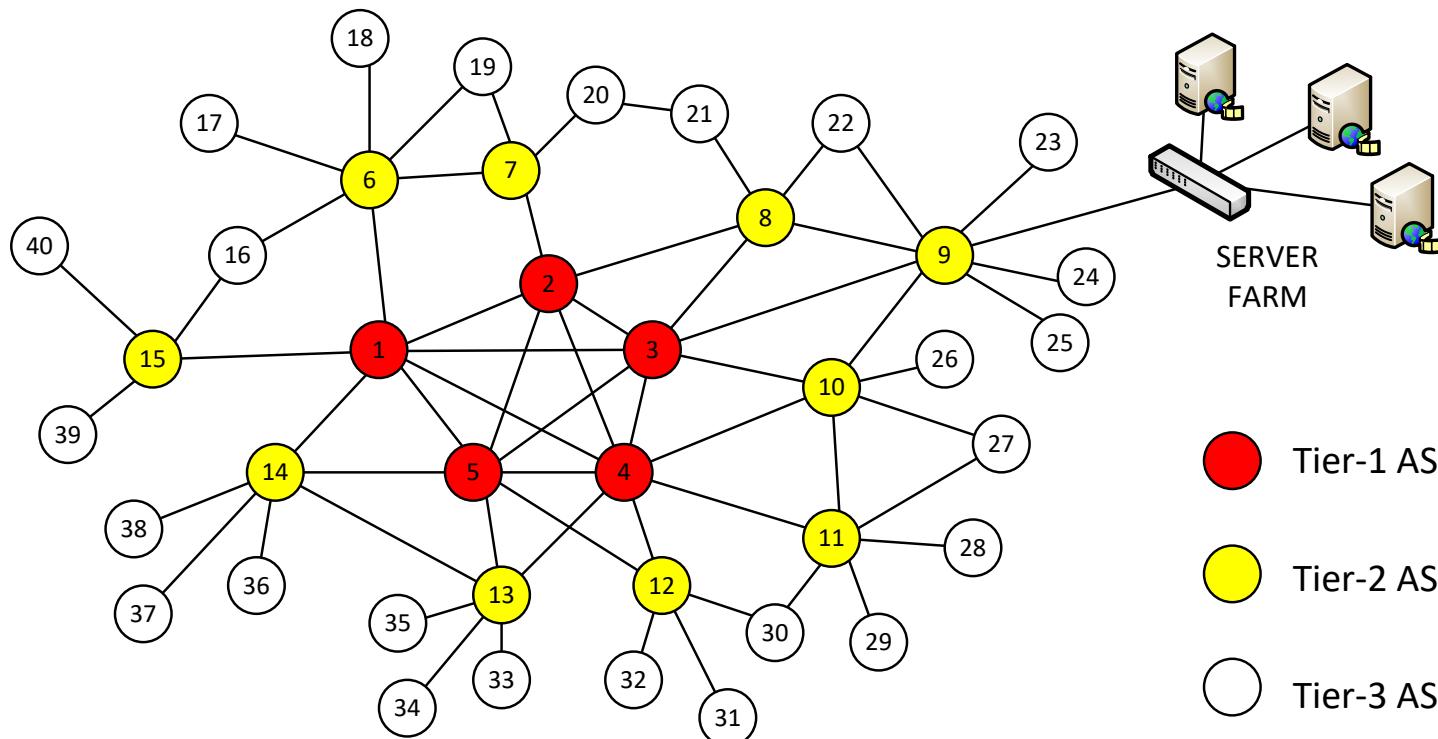
## Illustrative example - using Gurobi on Internet (3)

- After the problem is solved, the solution is displayed (and also sent to the email address):

```
Optimal solution found (tolerance 1.00e-04)
Best objective 1.15000000000e+01, best bound 1.15000000000e+01, gap 0.0000%
Optimal objective: 11.5
***** Begin .sol file *****
# Objective value = 11.5
x1 1
x2 1
x3 1
x4 0
x5 0
x6 1
y1_1 1
y2_1 1
y3_1 0
y4_1 0
y5_1 0
y6_1 0
y1_2 0
y2_2 0
y3_2 1
y4_2 0
y5_2 0
y6_2 1
***** End .sol file *****
```

# Solving the server farm location problem with ILP

- We have a set of Autonomous Systems (ASs) and we aim to select a subset of ASs to connect one server farm on each selected AS.
- Only Tier-2 of Tier-3 ASs provide the Internet access service.
- The solution must guarantee that there is a path from each Tier-2 and Tier-3 ASs to at least one server farm with no more than one intermediate AS.



# **Server farm location problem: Notation and Variables**

## **PARAMETERS:**

$n_1, n_2, \dots$  – IDs of Tier-2 and Tier-3 ASs (i.e., ASs where server farms can be connected to)

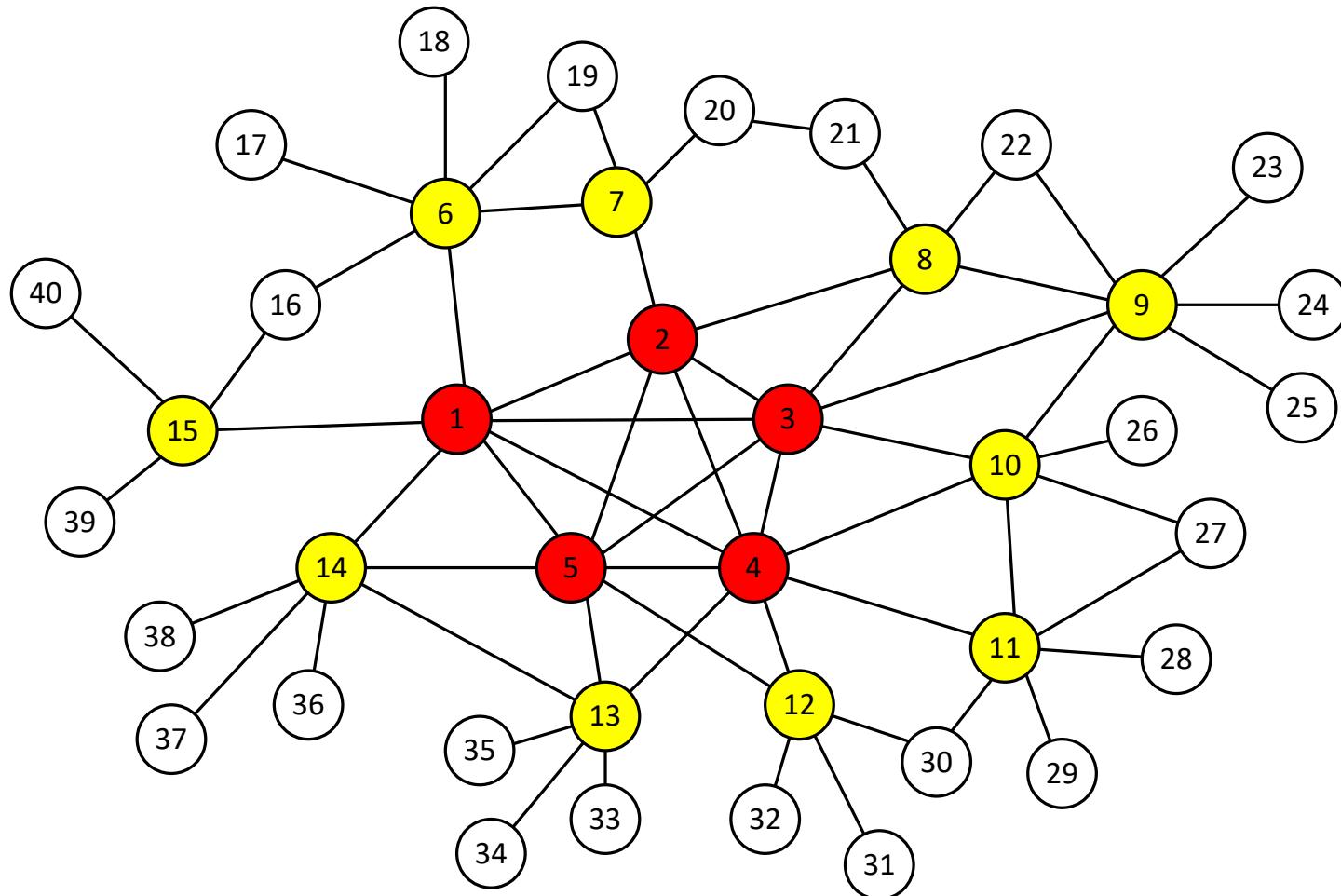
$c_i$  – cost of the Internet connection to AS  $i$ , with  $i = n_1, n_2, \dots$

$I(j)$  – set of Tier-2 and Tier-3 AS IDs such that there is a shortest path from AS  $j$  to each AS  $i \in I(j)$  with at most one intermediate AS

## **VARIABLES:**

$x_i$  – binary variable, with  $i = n_1, n_2, \dots$ , that when is equal to 1 means that AS  $i$  must have a connected server farm

## Server farm location problem: examples of sets $I(j)$



Set  $I(j)$  for  $j = 6$  is: {6,7,14,15,16,17,18,19,20}

for  $j = 16$  is: {6,7,15,16,17,18,19,39,40}

# Server farm location problem: ILP Model

$$\text{Minimize } \sum_{i=6}^{40} c_i x_i \quad (1)$$

Subject to:

$$\sum_{i \in I(j)} x_i \geq 1 \quad , j = 6, \dots, 40 \quad (2)$$

$$x_i \in \{0,1\} \quad , i = 6, \dots, 40 \quad (3)$$

- The objective (1) is the minimization of the total Internet connection costs on ASs with connected server farms.
- Constraints (2) guarantee that each AS  $j$  has at least one server farm connected to one AS  $i \in I(j)$ , i.e., one server farm whose shortest path has at most one intermediate AS.
- Constraints (3) define all variables as binary variables.



# **Encaminhamento em Redes com Comutação de Pacotes**

Desempenho e Dimensionamento de Redes

Prof. Amaro de Sousa ([asou@ua.pt](mailto:asou@ua.pt))

DETI-UA, 2020/2021

# Sumário do Módulo

- Primeira parte:
  - Redes de circuitos virtuais e redes de datagramas
  - Encaminhamento e atribuição de recursos
- Segunda parte:
  - Desempenho de redes com comutação de pacotes (aproximação de Kleinrock)
- Terceira parte:
  - Encaminhamento ótimo quando os fluxos de pacotes podem ser bifurcados por múltiplos percursos de encaminhamento



# **Encaminhamento em Redes com Comutação de Pacotes**

**Primeira parte:**

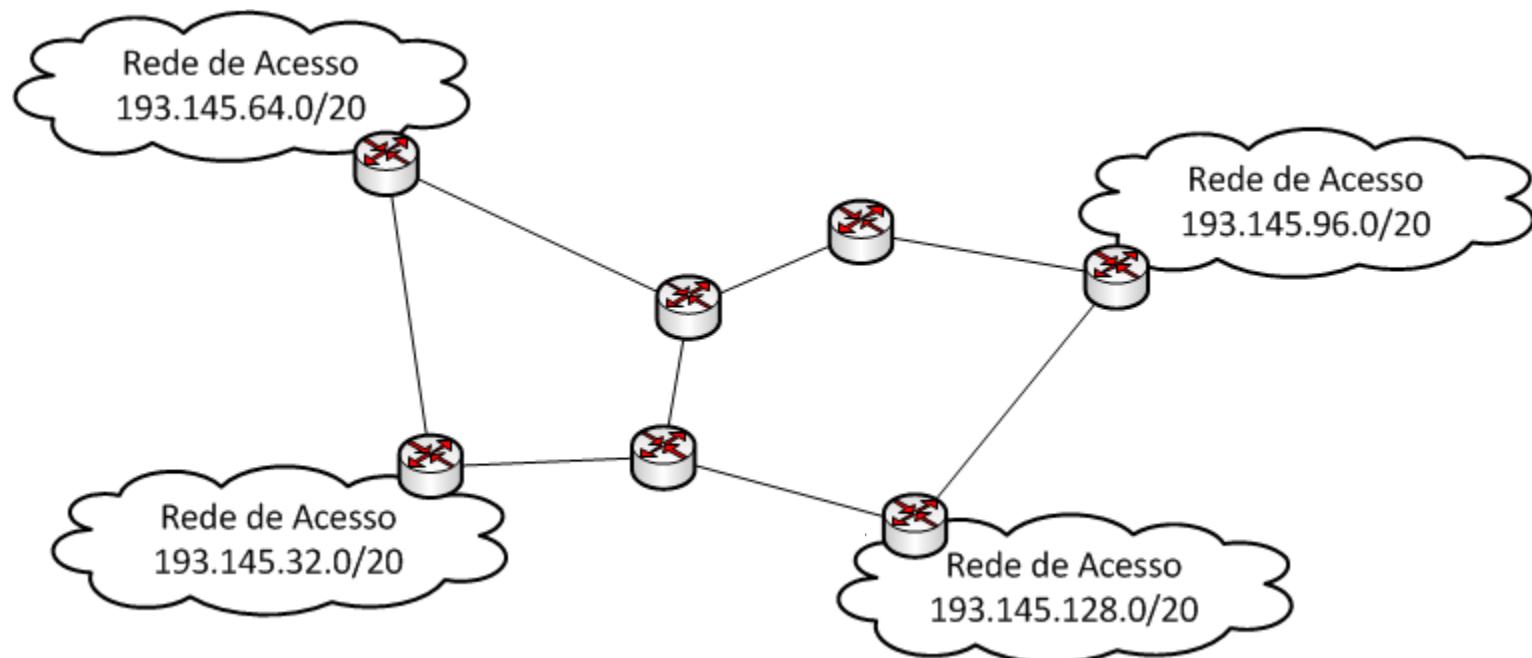
- **Redes de circuitos virtuais e redes de datagramas**
- **Encaminhamento e atribuição de recursos**

# Encaminhamento em redes com comutação de pacotes

Existem 2 tipos de redes com comutação de pacotes:

- redes de circuitos virtuais
- redes de datagramas

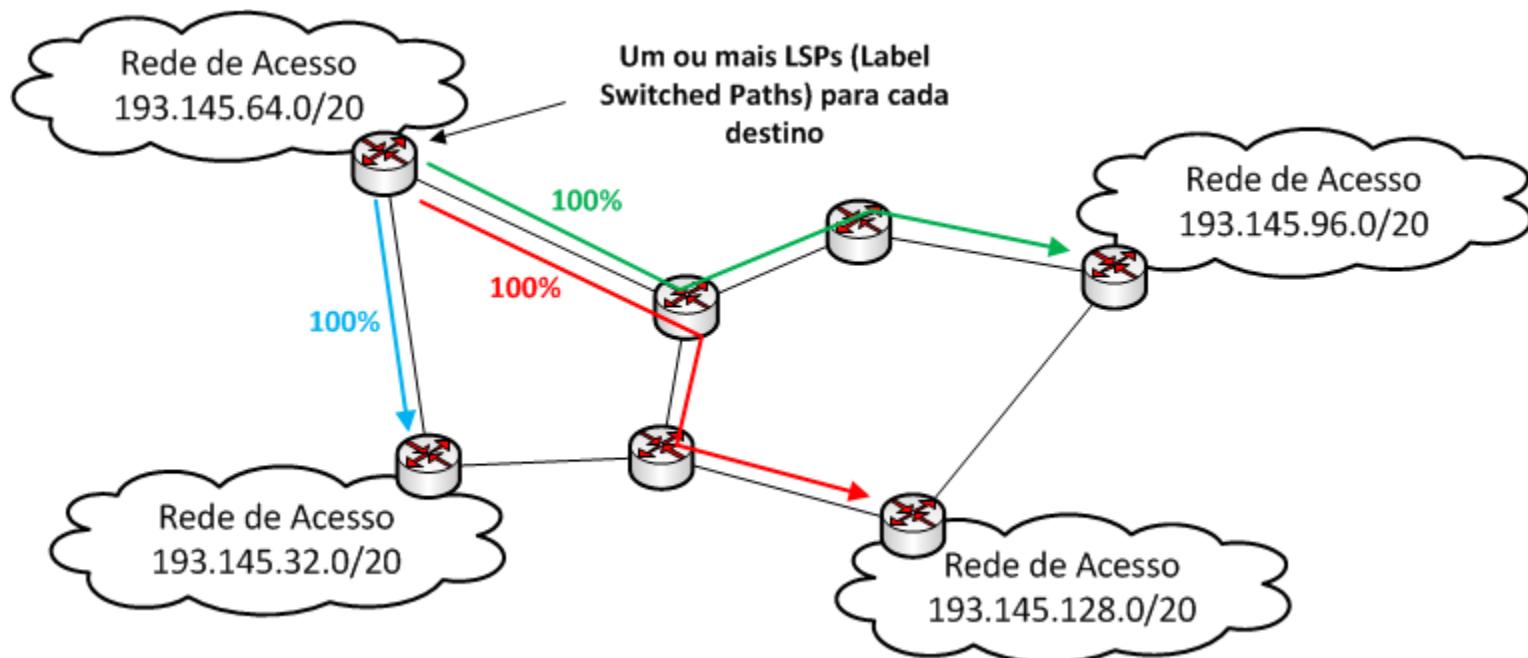
Considere-se o seguinte exemplo de uma rede de um ISP (*Internet Service Provider*) que liga 4 redes de acesso:



# Encaminhamento em redes com comutação de pacotes – redes de circuitos virtuais

- A cada fluxo de pacotes, é atribuído pelo menos um circuito virtual.
- Os percursos dos circuitos virtuais são inicialmente estabelecidos.
- Após o estabelecimento dos circuitos virtuais, os pacotes de cada fluxo são encaminhados pelos circuitos virtuais atribuídos.

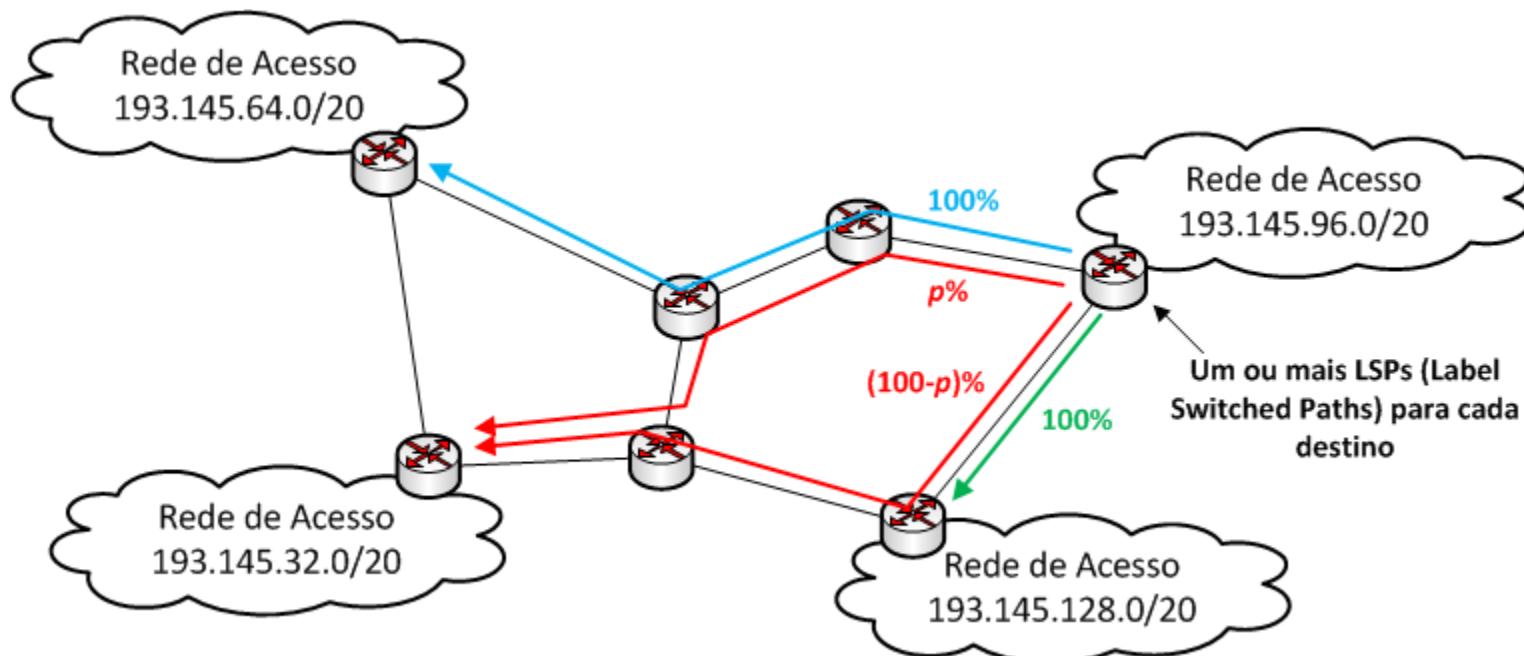
Exemplo: redes IP/MPLS em que os circuitos virtuais se designam por LSPs (*Label Switched Paths*).



# Encaminhamento em redes com comutação de pacotes – redes de circuitos virtuais

- A cada fluxo de pacotes, é atribuído pelo menos um circuito virtual.
- Os percursos dos circuitos virtuais são inicialmente estabelecidos.
- Após o estabelecimento dos circuitos virtuais, os pacotes de cada fluxo são encaminhados pelos circuitos virtuais atribuídos.

Exemplo: redes IP/MPLS em que os circuitos virtuais se designam por LSPs (*Label Switched Paths*).



# **Encaminhamento em redes com comutação de pacotes – redes de datagramas**

- As decisões de encaminhamento são efetuadas pacote a pacote.
- Assim, dois pacotes do mesmo par origem-destino podem seguir percursos distintos na rede.

Exemplo: redes IP com o protocolo de encaminhamento RIP ou OSPF.

Nas redes IP, o encaminhamento é baseado em percursos de custo mínimo de cada nó (router) para cada rede destino

- No OSPF, é atribuído a cada ligação um número positivo designado por custo da ligação.
- No RIP, o custo é 1 para cada ligação.
- Cada percurso de um router para um destino tem um custo igual à soma dos custos das ligações que o compõem.
- Em cada router, cada pacote IP é encaminhado por um dos percursos de custo mínimo para a rede destino do pacote.

# Encaminhamento em redes com comutação de pacotes – redes de datagramas

Cada pacote IP é encaminhado por um dos percursos de custo mínimo para o destino do pacote:

Método estático: o custo das ligações é fixo (o caso do RIP e do OSPF).

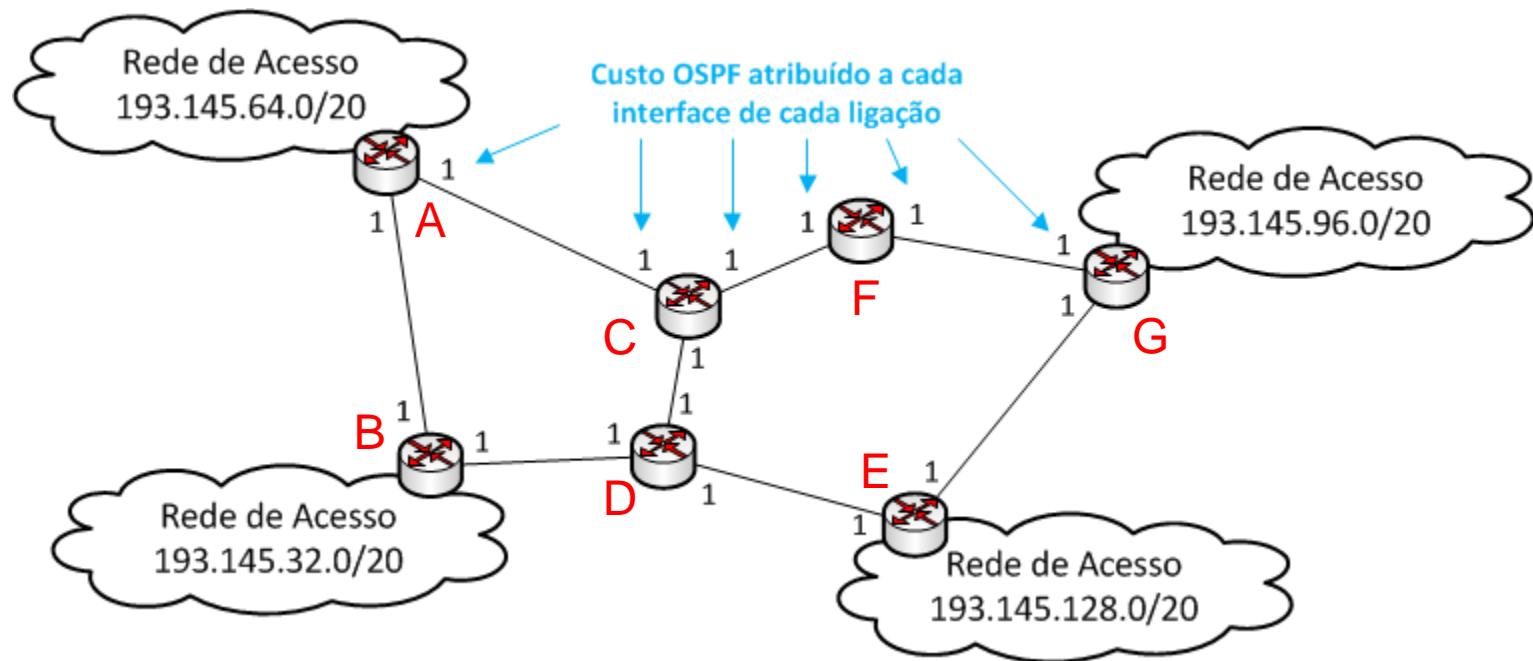
Método dinâmico: o custo das ligações varia ao longo do tempo em função do seu nível de utilização (exemplo: protocolos IGRP e EIGRP)

- o percurso de custo mínimo adapta-se a situações de sobrecarga obrigando os pacotes a evitarem as ligações mais utilizadas
- introduz um efeito de realimentação que pode levar a oscilações indesejáveis.

Quando existem múltiplos percursos de custo mínimo de um nó para um destino, é usada a técnica ECMP (*Equal Cost Multi-Path*):

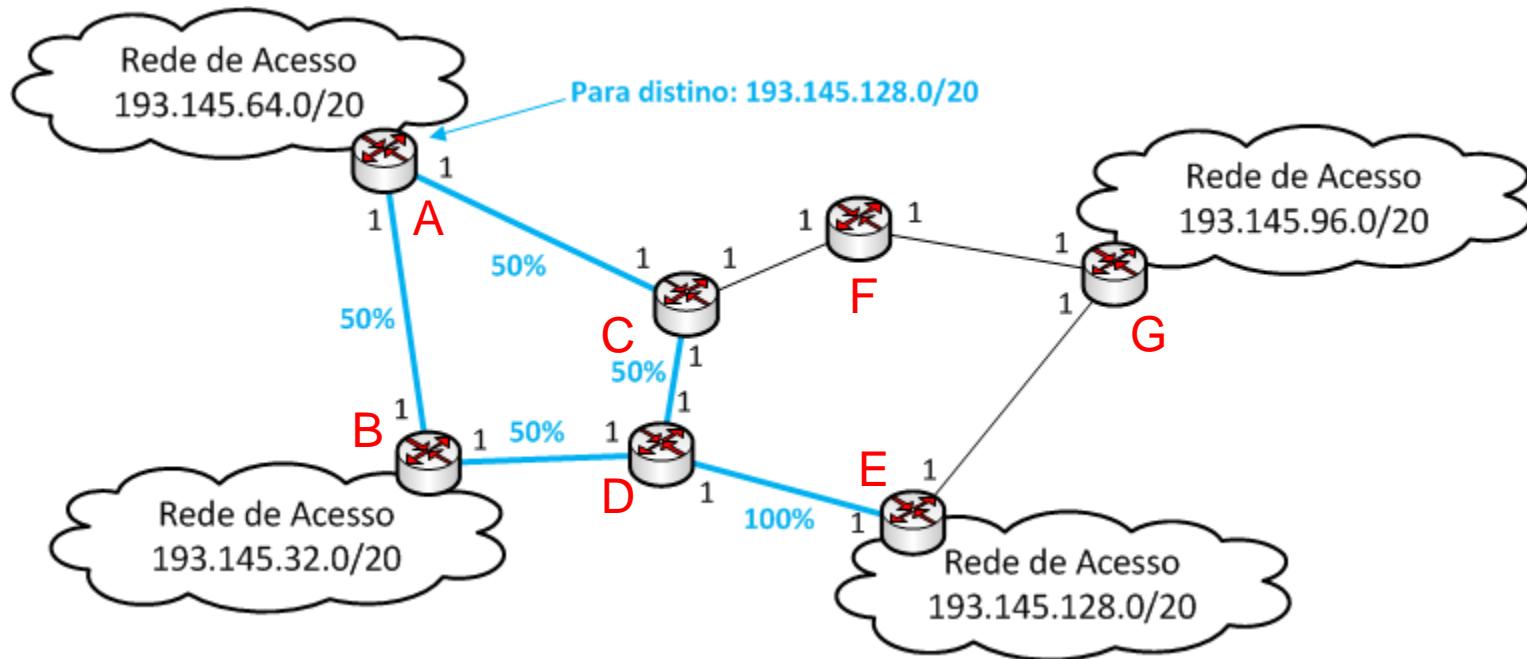
- em cada nó, o tráfego é bifurcado em igual percentagem por todas as ligações de saída que proporcionam percursos de custo mínimo

# Encaminhamento em redes IP com encaminhamento OSPF (I)



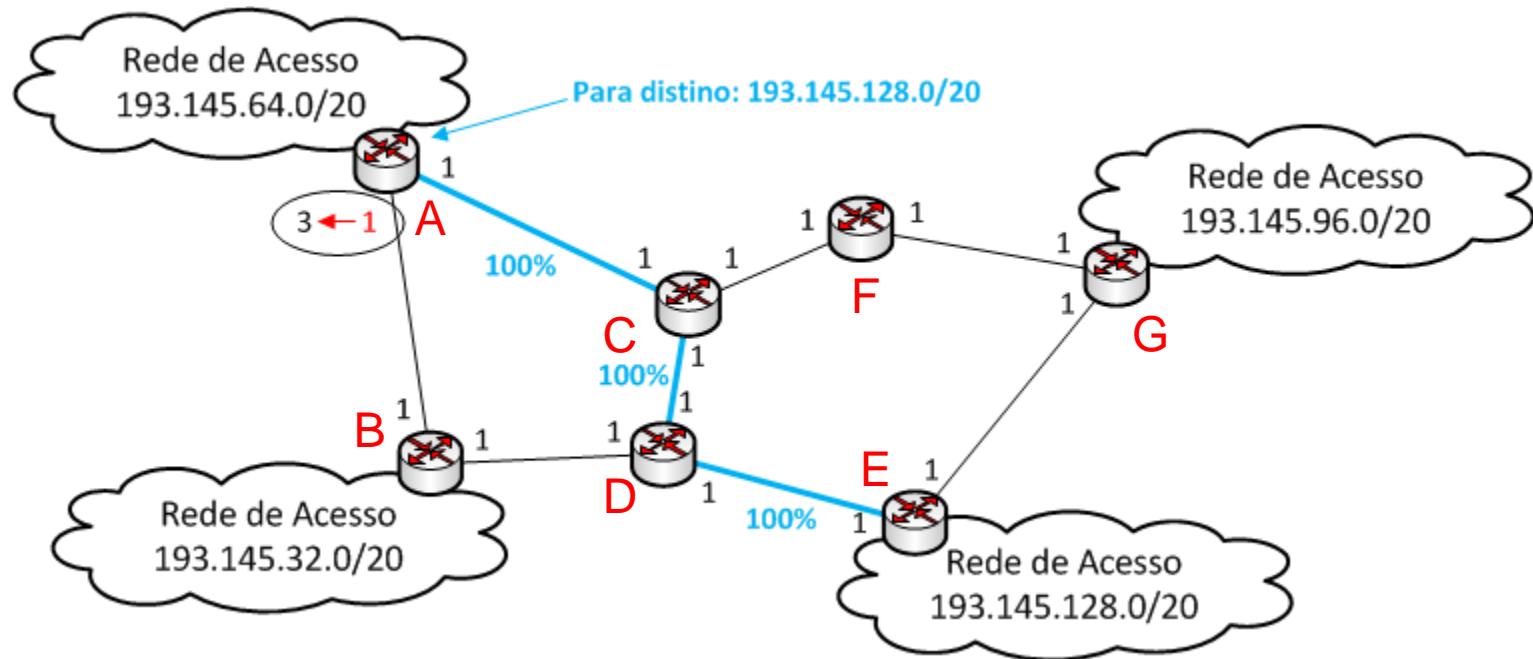
Neste exemplo, todos os custos OSPF estão configurados a 1 (equivalente ao RIP).

# Encaminhamento em redes IP com encaminhamento OSPF (II)



Pelo ECMP, o router A encaminha os pacotes IP com destino para um endereço IP da rede 193.145.128.0/20 em igual percentagem pelos percursos que passam por B e por C.

# Encaminhamento em redes IP com encaminhamento OSPF (III)



Mudando o custo da ligação de A para B de 1 para 3, o router A encaminha os pacotes IP com destino para um endereço IP da rede 193.145.128.0/20 pelo único percurso de custo mínimo.



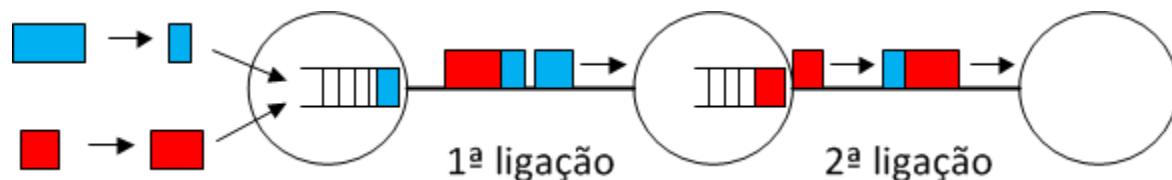
## **Encaminhamento em Redes com Comutação de Pacotes**

**Segunda parte:**

- **Desempenho de redes com comutação de pacotes  
(aproximação de Kleinrock)**

# Redes de ligações ponto-a-ponto

Numa rede de ligações ponto-a-ponto os intervalos entre chegadas de pacotes estão correlacionados com o comprimento dos pacotes, após a passagem pela primeira ligação. Este facto dificulta a análise.

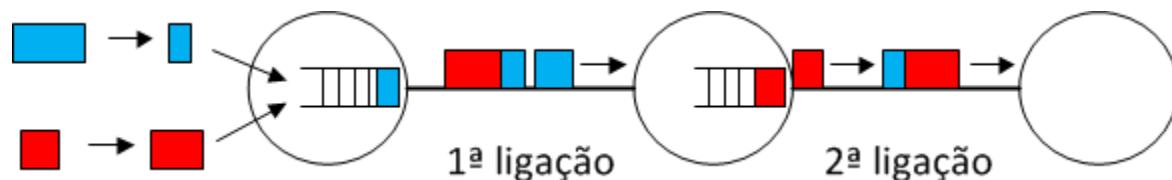


Exemplo:

- Considerem-se duas ligações ponto-a-ponto em cascata.
- Considere-se um fluxo de pacotes com origem no nó à esquerda e destino no nó à direita.
- Considere-se que os pacotes deste fluxo chegam segundo um processo de Poisson e o comprimento dos pacotes é exponencialmente distribuído.

# Redes de ligações ponto-a-ponto

Numa rede de ligações ponto-a-ponto os intervalos entre chegadas de pacotes estão correlacionados com o comprimento dos pacotes, após a passagem pela primeira ligação. Este facto dificulta a análise.



- a 1<sup>a</sup> fila de espera é do tipo  $M/M/1$
- no entanto, a 2<sup>a</sup> fila de espera não é do tipo  $M/M/1$ :
  - o intervalo entre a chegada de dois pacotes consecutivos à 2<sup>a</sup> fila de espera é sempre superior ou igual ao tempo de transmissão do segundo pacote na 1<sup>a</sup> fila de espera;
  - assim, tipicamente pacotes maiores esperam menos tempo na 2<sup>a</sup> fila de espera que pacotes mais pequenos.

# Aproximação de Kleinrock

A aproximação de Kleinrock consiste em assumir que as chegadas de pacotes são processos de Poisson em todos as ligações

- i.e. ignora a correlação entre comprimento dos pacotes e intervalos entre chegadas de pacotes

Considerando adicionalmente que:

- as filas de espera são muito grandes em todas as ligações;
- o tamanho dos pacotes é exponencialmente distribuído com a mesma média em todos os fluxos;

então, cada ligação é modelada por um sistema  $M/M/1$ .

De notar que:

- os fluxos de pacotes são unidirecionais e as ligações das redes de comutação de pacotes são bidireccionais
- assim, uma ligação de rede entre os nós de comutação  $i$  e  $j$  é representada pelos pares ordenados  $(i,j)$  e  $(j,i)$  que representam cada um dos sentidos da ligação

# Aproximação de Kleinrock

Considere-se uma rede de ligações ponto-a-ponto, onde existem diversos fluxos de pacotes  $s = 1 \dots S$ .

- Considere-se que cada fluxo  $s$  é suportado por um percurso único na rede, formado por uma sequência de ligações  $(i,j)$  definida pelo conjunto  $R_s$ .
- Seja  $\lambda_s$  a taxa de chegada de pacotes do fluxo  $s$ , em pacotes/segundo.

Então a taxa de chegada de pacotes à ligação  $(i,j)$  é:

$$\lambda_{ij} = \sum_{s:(i,j) \in R_s} \lambda_s$$

# Aproximação de Kleinrock

Considere-se agora o caso em que pode haver múltiplos percursos associados a cada fluxo de pacotes  $s$ :

- Seja  $f_{ij}(s)$  a fração de pacotes do fluxo  $s$  que atravessa a ligação  $(i,j)$
- Considere-se que nenhum pacote atravessa duas vezes a mesma ligação (i.e., não há ciclos de encaminhamento).
- Neste caso, o conjunto  $R_s$  inclui todas as ligações  $(i,j)$  tais que  $f_{ij}(s) > 0$ .

Então a taxa de chegada de pacotes à ligação  $(i,j)$  é:  $\lambda_{ij} = \sum_{s:(i,j) \in R_s} f_{ij}(s) \lambda_s$

Considerando  $\mu_{ij}$  a capacidade da ligação  $(i,j)$  em número médio de pacotes/segundo, o número médio de pacotes em todas as ligações é:

$$L = \sum_{(i,j)} \frac{\lambda_{ij}}{\mu_{ij} - \lambda_{ij}}$$

# Aproximação de Kleinrock

Usando o teorema de Little, o atraso médio por pacote da rede é:

$$W = \frac{1}{\gamma} \sum_{(i,j)} \frac{\lambda_{ij}}{\mu_{ij} - \lambda_{ij}} \quad \gamma = \sum_s \lambda_s$$

Nos casos em que os atrasos de processamento dos pacotes nos nós de comutação e os atrasos de propagação nas ligações não são desprezáveis, o atraso médio por pacote da rede passa a ser

$$W = \frac{1}{\gamma} \sum_{(i,j)} \left( \frac{\lambda_{ij}}{\mu_{ij} - \lambda_{ij}} + \lambda_{ij} d_{ij} \right) \quad \gamma = \sum_s \lambda_s$$

em que  $d_{ij}$  é o atraso médio de processamento e propagação associado à ligação  $(i, j)$ .

# Aproximação de Kleinrock

No caso em que a cada fluxo  $s$  está associado um percurso único na rede, o atraso médio por pacote do fluxo de tráfego  $s$  é:

$$W_s = \sum_{(i,j) \in R_s} \left( \frac{1}{\mu_{ij} - \lambda_{ij}} + d_{ij} \right)$$

No caso em que há diferentes percursos associados a cada fluxo de pacotes  $s$ , o atraso médio por pacote do fluxo  $s$  é:

- a média pesada do atraso de cada percurso (fórmula acima)
- o peso de cada percurso é a percentagem da taxa de chegada do fluxo  $s$ ,  $\lambda_s$ , que é encaminhado pelo percurso.

- 
- Nas redes com um percurso por fluxo, a maior fonte de erro associada à aproximação de Kleinrock deve-se à correlação entre os comprimentos dos pacotes e os intervalos entre chegadas.
  - Nas redes com múltiplos percursos por fluxo, pode existir um fator adicional de erro, dependendo da forma como os fluxos são bifurcados nos nós.

## Exemplo 1

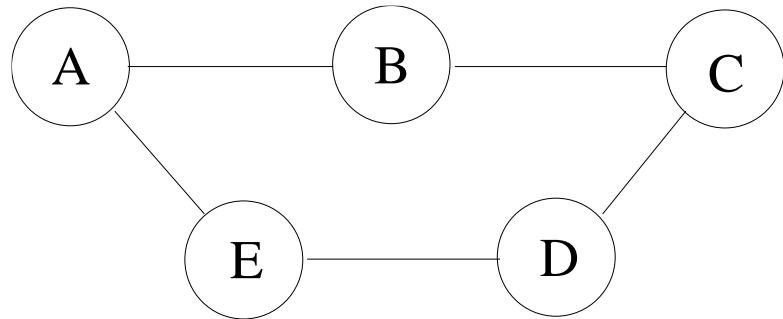
Considere a rede IP da figura com todas as ligações bidirecionais de 10 Mbps. A rede suporta 4 fluxos de pacotes:

- de A para C com uma taxa de Poisson de 1000 pps,
- de A para D com uma taxa de Poisson de 250 pps,
- de B para D com uma taxa de Poisson de 1000 pps e
- de B para E com uma taxa de Poisson de 750 pps.

O tamanho dos pacotes é exponencialmente distribuído com média 500 bytes em todos os fluxos. O tempo de propagação da ligação B-C é de 10 ms em cada sentido e desprezável nas outras ligações.

O protocolo de encaminhamento nos routers é o RIP. Utilizando a aproximação de Kleinrock, calcule:

- (a) o atraso médio por pacote de cada fluxo;
- (b) o atraso médio por pacote de todos os fluxos;
- (c) a utilização (em percentagem) de cada ligação em cada sentido.

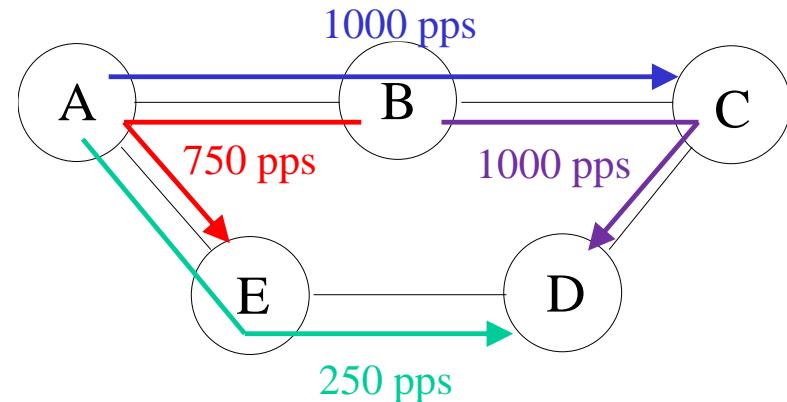


# Exemplo 1

- Ligações bidirecionais de 10 Mbps
- Pacotes de 500 bytes, em média
- Tempo de propagação da ligação B-C de 10 ms em cada sentido
- Encaminhamento RIP

(a) O atraso médio por pacote de cada fluxo.

$$\mu_{AB} = \mu_{BA} = \mu_{BC} = \dots = \mu = \frac{10 \times 10^6 \text{ bps}}{500 \times 8 \text{ bpp}} = 2500 \text{ pps}$$



$$W_s = \sum_{(i,j) \in R_s} \left( \frac{1}{\mu_{ij} - \lambda_{ij}} + d_{ij} \right)$$

$$W_{A \rightarrow C} = \frac{1}{\mu_{AB} - \lambda_{AB}} + d_{AB} + \frac{1}{\mu_{BC} - \lambda_{BC}} + d_{BC} = \frac{1}{2500 - 1000} + 0 + \frac{1}{2500 - (1000 + 1000)} + 0.01 = 0.0127 \text{ seg.}$$

$$W_{A \rightarrow D} = \frac{1}{\mu_{AE} - \lambda_{AE}} + d_{AE} + \frac{1}{\mu_{ED} - \lambda_{ED}} + d_{ED} = \frac{1}{2500 - (750 + 250)} + 0 + \frac{1}{2500 - 250} + 0 = 0.0011 \text{ seg.}$$

$$W_{B \rightarrow D} = \frac{1}{\mu_{BC} - \lambda_{BC}} + d_{BC} + \frac{1}{\mu_{CD} - \lambda_{CD}} + d_{CD} = \frac{1}{2500 - (1000 + 1000)} + 0.01 + \frac{1}{2500 - 1000} + 0 = 0.0127 \text{ seg.}$$

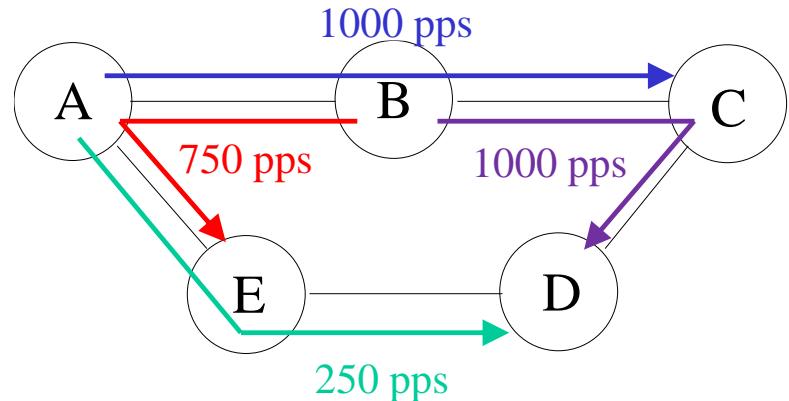
$$W_{B \rightarrow E} = \frac{1}{\mu_{BA} - \lambda_{BA}} + d_{BA} + \frac{1}{\mu_{AE} - \lambda_{AE}} + d_{AE} = \frac{1}{2500 - 750} + 0 + \frac{1}{2500 - (750 + 250)} + 0 = 0.0012 \text{ seg.}$$

# Exemplo 1

- Ligações bidirecionais de 10 Mbps
- Pacotes de 500 bytes, em média
- Tempo de propagação da ligação B-C de 10 ms em cada sentido
- Encaminhamento RIP

(b) O atraso médio por pacote de todos os fluxos.

$$\mu_{AB} = \mu_{BA} = \mu_{BC} = \dots = \mu = \frac{10 \times 10^6 \text{ bps}}{500 \times 8 \text{ bpp}} = 2500 \text{ pps}$$



$$W = \frac{1}{\gamma} \sum_{(i,j)} \left( \frac{\lambda_{ij}}{\mu_{ij} - \lambda_{ij}} + \lambda_{ij} d_{ij} \right)$$

$$\gamma = \sum_s \lambda_s$$

$$\gamma = \lambda_{A \rightarrow C} + \lambda_{A \rightarrow D} + \lambda_{B \rightarrow D} + \lambda_{B \rightarrow E} = 1000 + 250 + 1000 + 750 = 3000 \text{ pps}$$

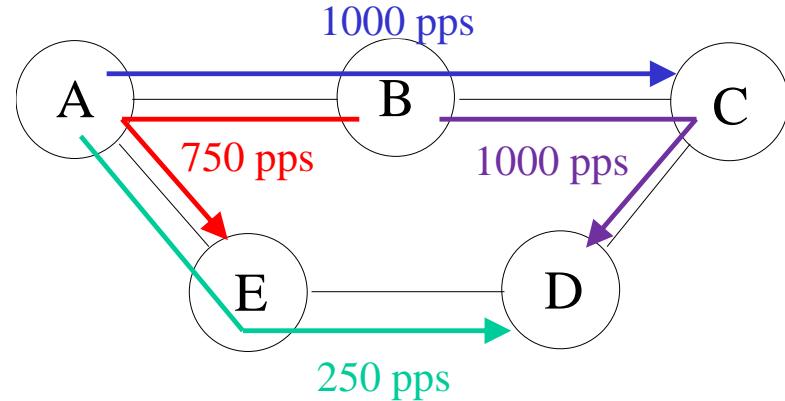
$$W = \frac{1}{\gamma} \times \left( \frac{\lambda_{AB}}{\mu_{AB} - \lambda_{AB}} + \frac{\lambda_{BA}}{\mu_{BA} - \lambda_{BA}} + \frac{\lambda_{BC}}{\mu_{BC} - \lambda_{BC}} + \lambda_{BC} d_{BC} + \frac{\lambda_{CD}}{\mu_{CD} - \lambda_{CD}} + \frac{\lambda_{AE}}{\mu_{AE} - \lambda_{AE}} + \frac{\lambda_{ED}}{\mu_{ED} - \lambda_{ED}} \right)$$

$$W = \frac{1}{3000} \times \left( \frac{1000}{2500 - 1000} + \frac{750}{2500 - 750} + \frac{2000}{2500 - 2000} + 2000 \times 0.01 + \frac{1000}{2500 - 1000} + \frac{1000}{2500 - 1000} + \frac{250}{2500 - 250} \right)$$

$$W = 0.00865 \text{ seg.}$$

# Exemplo 1

- Ligações bidirecionais de 10 Mbps
- Pacotes de 500 bytes, em média
- Tempo de propagação da ligação B-C de 10 ms em cada sentido
- Encaminhamento RIP



(c) A utilização (em percentagem) de cada ligação em cada sentido.

$$\mu_{AB} = \mu_{BA} = \mu_{BC} = \dots = \mu = \frac{10 \times 10^6 \text{ bps}}{500 \times 8 \text{ bpp}} = 2500 \text{ pps}$$

$$U_{AB} = \frac{\lambda_{AB}}{\mu_{AB}} = \frac{1000}{2500} = 0.4 = 40\%$$

$$U_{BA} = \frac{\lambda_{BA}}{\mu_{BA}} = \frac{750}{2500} = 0.3 = 30\%$$

$$U_{BC} = \frac{\lambda_{BC}}{\mu_{BC}} = \frac{2000}{2500} = 0.8 = 80\%$$

$$U_{CD} = \frac{\lambda_{CD}}{\mu_{CD}} = \frac{1000}{2500} = 0.4 = 40\%$$

$$U_{AE} = \frac{\lambda_{AE}}{\mu_{AE}} = \frac{1000}{2500} = 0.4 = 40\%$$

$$U_{ED} = \frac{\lambda_{ED}}{\mu_{ED}} = \frac{250}{2500} = 0.1 = 10\%$$

## Exemplo 2

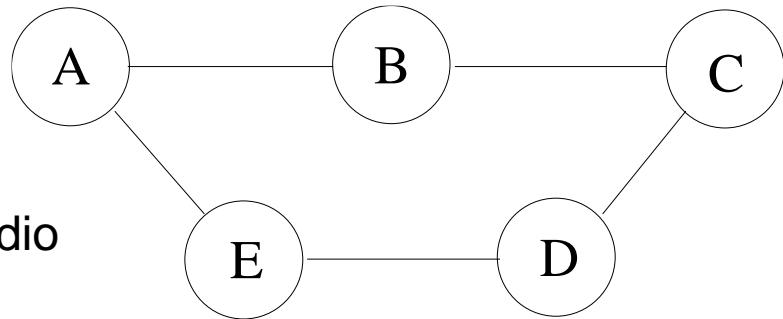
Considere a rede IP da figura com todas as ligações bidireccionais de 10 Mbps. A rede suporta 4 fluxos de pacotes:

- de A para C com uma taxa de Poisson de 1000 pps,
- de A para D com uma taxa de Poisson de 250 pps,
- de B para D com uma taxa de Poisson de 1000 pps e
- de B para E com uma taxa de Poisson de 750 pps.

O tamanho dos pacotes é exponencialmente distribuído com média 500 bytes em todos os fluxos. O tempo de propagação da ligação B-C é de 10 ms em cada sentido e desprezável nas outras ligações.

O protocolo de encaminhamento nos routers é o OSPF.

- Determine os custos OSPF que permitem minimizar a utilização da ligação mais carregada.
- Utilizando a aproximação de Kleinrock, determine o atraso médio por pacote de todos os fluxos na solução anterior.



## Exemplo 2

- Ligações bidirecionais de 10 Mbps
- Pacotes de 500 bytes, em média
- Propagação da ligação B-C de 10 ms em cada sentido
- Encaminhamento OSPF

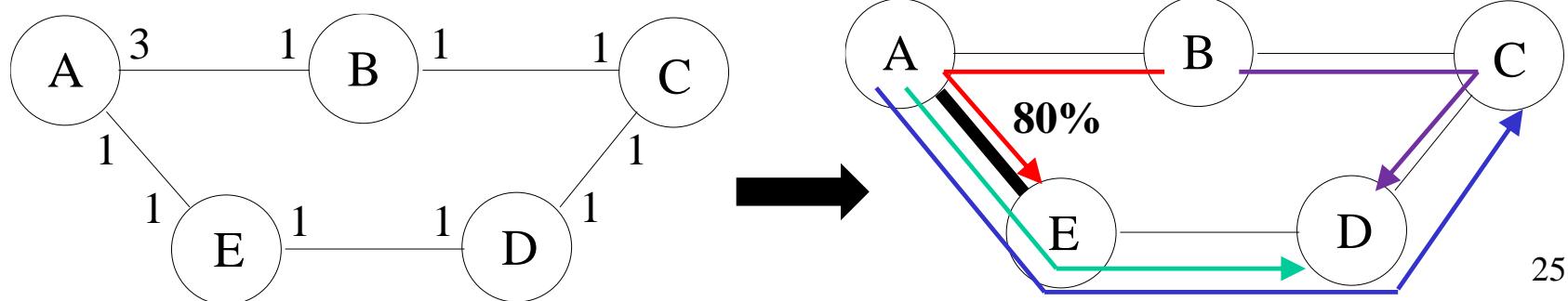
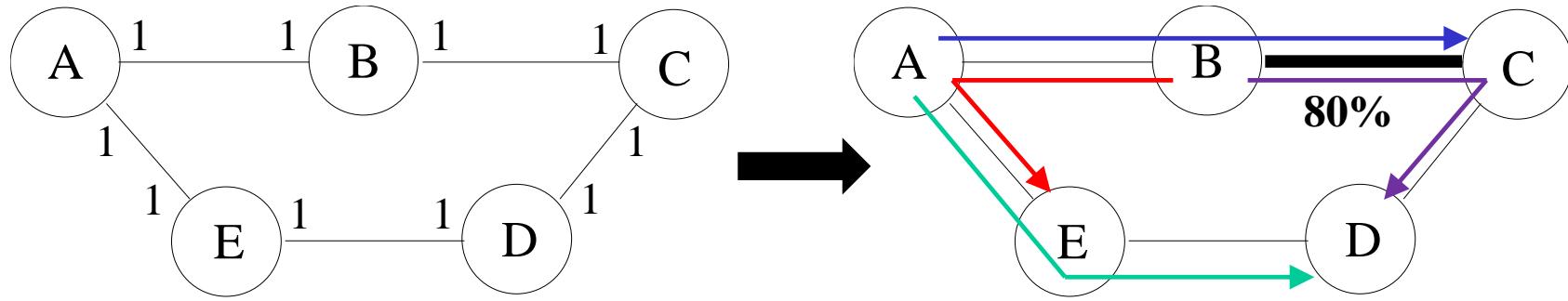
(a) Determine os custos OSPF que permitem minimizar a utilização da ligação mais carregada.

$$\lambda_{A \rightarrow C} = 1000 \text{ pps}$$

$$\lambda_{A \rightarrow D} = 250 \text{ pps}$$

$$\lambda_{B \rightarrow D} = 1000 \text{ pps}$$

$$\lambda_{B \rightarrow E} = 750 \text{ pps}$$



## Exemplo 2

- Ligações bidirecionais de 10 Mbps
- Pacotes de 500 bytes, em média
- Propagação da ligação B-C de 10 ms em cada sentido
- Encaminhamento OSPF

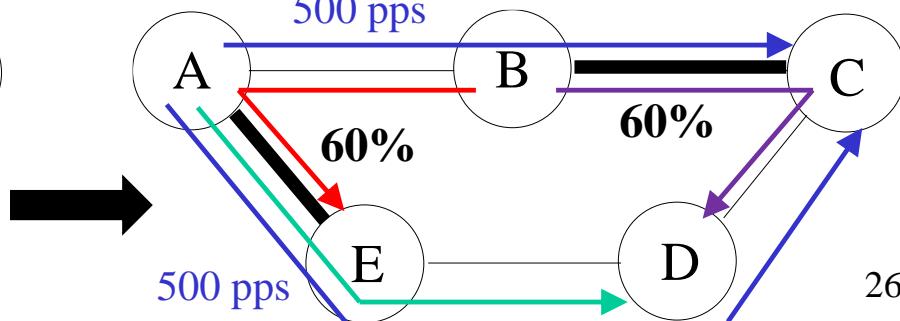
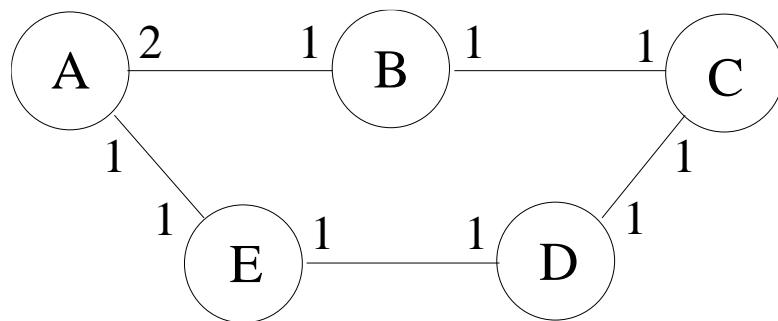
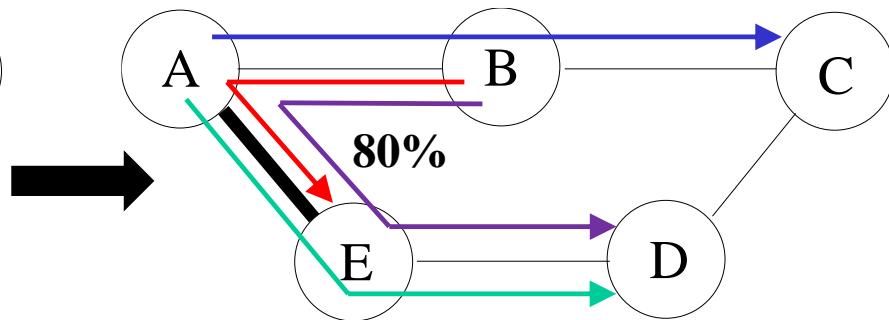
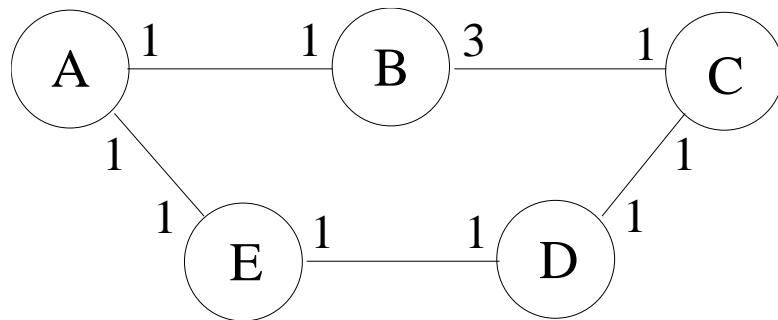
(a) Determine os custos OSPF que permitem minimizar a utilização da ligação mais carregada.

$$\lambda_{A \rightarrow C} = 1000 \text{ pps}$$

$$\lambda_{A \rightarrow D} = 250 \text{ pps}$$

$$\lambda_{B \rightarrow D} = 1000 \text{ pps}$$

$$\lambda_{B \rightarrow E} = 750 \text{ pps}$$



## Exemplo 2

- Ligações bidirecionais de 10 Mbps
- Pacotes de 500 bytes, em média
- Propagação da ligação B-C de 10 ms em cada sentido
- Encaminhamento OSPF

(b) Utilizando a aproximação de Kleinrock, determine o atraso médio por pacote de todos os fluxos na solução anterior.

$$\mu_{AB} = \mu_{BA} = \mu_{BC} = \dots = \mu = \frac{10 \times 10^6 \text{ bps}}{500 \times 8 \text{ bpp}} = 2500 \text{ pps}$$

$$\gamma = \lambda_{A \rightarrow C} + \lambda_{A \rightarrow D} + \lambda_{B \rightarrow D} + \lambda_{B \rightarrow E} = 1000 + 250 + 1000 + 750 = 3000 \text{ pps}$$

$$W = \frac{1}{\gamma} \times \left( \frac{\lambda_{AB}}{\mu_{AB} - \lambda_{AB}} + \frac{\lambda_{BA}}{\mu_{BA} - \lambda_{BA}} + \frac{\lambda_{BC}}{\mu_{BC} - \lambda_{BC}} + \lambda_{BC} d_{BC} + \frac{\lambda_{CD}}{\mu_{CD} - \lambda_{CD}} + \frac{\lambda_{DC}}{\mu_{DC} - \lambda_{DC}} + \frac{\lambda_{AE}}{\mu_{AE} - \lambda_{AE}} + \frac{\lambda_{ED}}{\mu_{ED} - \lambda_{ED}} \right)$$

$$W = \frac{1}{3000} \times \left( \frac{500}{2500 - 500} + \frac{750}{2500 - 750} + \frac{1500}{2500 - 1500} + 1500 \times 0.01 + \frac{1000}{2500 - 1000} + \frac{500}{2500 - 500} + \frac{1500}{2500 - 1500} + \frac{750}{2500 - 750} \right)$$

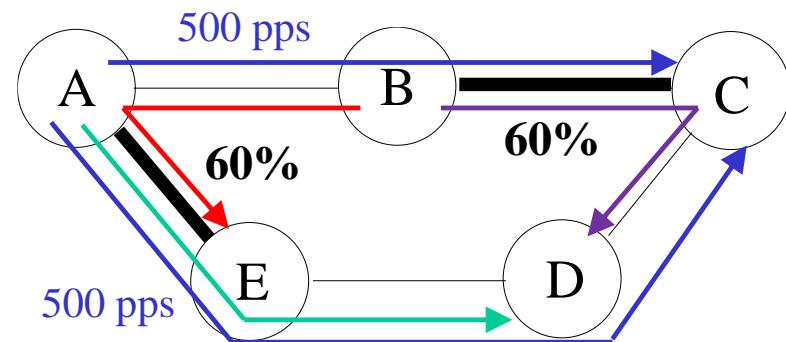
$$W = 0.00667 \text{ seg.} \quad (\text{Exemplo 1: } W = 0.00865 \text{ seg.})$$

$$\lambda_{A \rightarrow C} = 1000 \text{ pps}$$

$$\lambda_{A \rightarrow D} = 250 \text{ pps}$$

$$\lambda_{B \rightarrow D} = 1000 \text{ pps}$$

$$\lambda_{B \rightarrow E} = 750 \text{ pps}$$





## **Encaminhamento em Redes com Comutação de Pacotes**

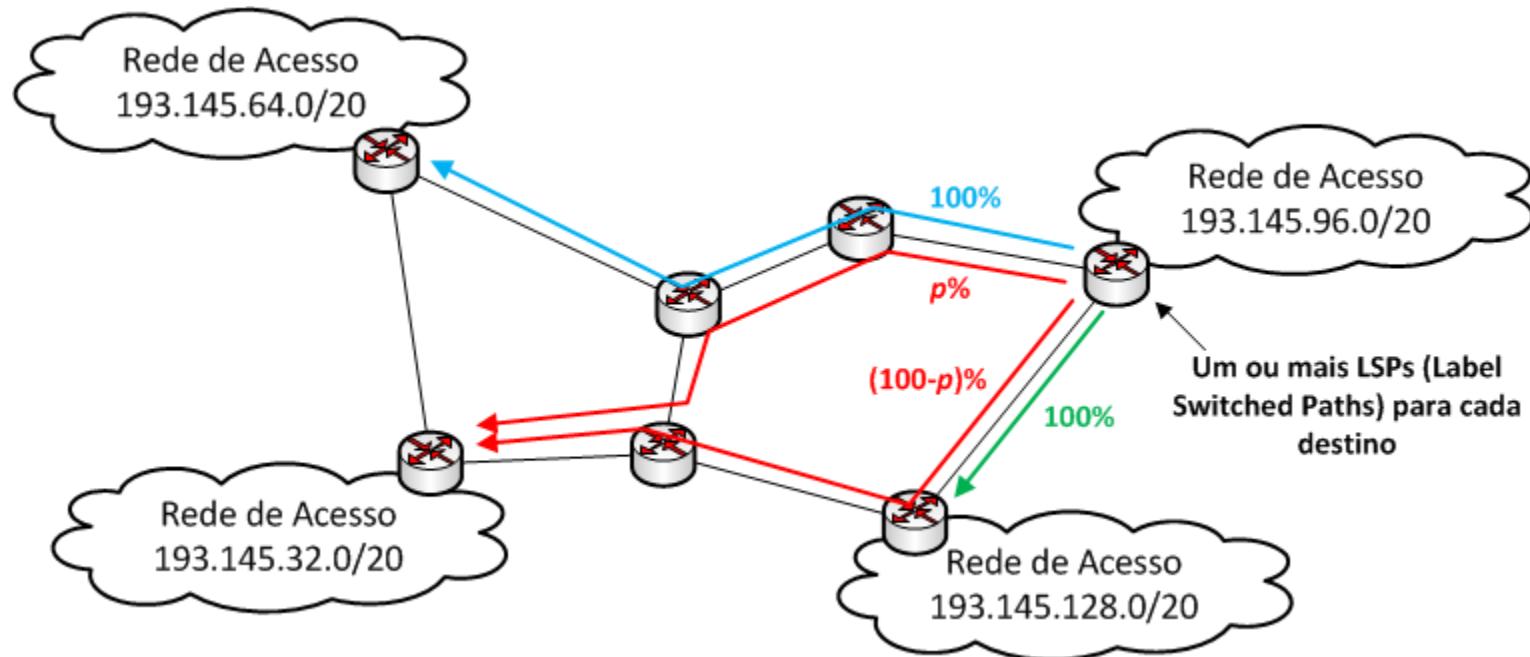
**Terceira parte:**

- **Encaminhamento ótimo quando os fluxos de pacotes podem ser bifurcados por múltiplos percursos de encaminhamento**

# Encaminhamento ótimo com bifurcação de fluxos

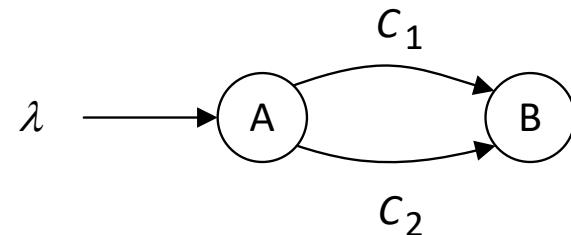
Numa rede MPLS, é possível:

- ter múltiplos LSPs atribuídos a um fluxo de pacotes
- escolher que porção do fluxo de pacotes é encaminhada por cada LSP



Questão: Como escolher as percentagens de bifurcação que optimizam o atraso médio por pacote da rede?

## Encaminhamento ótimo com bifurcação de fluxos (exemplo)



Na figura, o fluxo de pacotes  $\lambda$  (pacotes/seg) é bifurcado por duas ligações com capacidades  $C_1$  e  $C_2$  (ambas em pacotes/seg).

- Designemos os fluxos em cada ligação por  $x_1$  e  $x_2$ , respetivamente ( $\lambda = x_1 + x_2$ ). O número médio de pacotes nesta rede é, pela aproximação de Kleinrock, dado por:

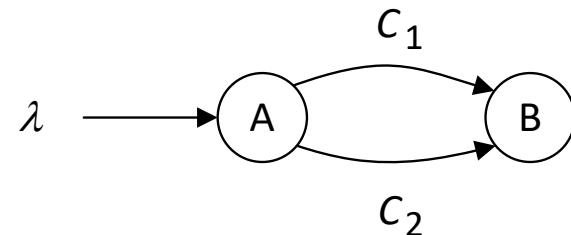
$$L = \frac{x_1}{C_1 - x_1} + \frac{x_2}{C_2 - x_2}$$

- Os valores de  $x_1$  e  $x_2$  que minimizam o atraso médio por pacote  $W$  são os que minimizam o número médio de pacotes na rede  $L$  (teorema de Little:  $L = \lambda W$ ).
- Assim, atendendo à restrição  $\lambda = x_1 + x_2$  temos:

$$L = \frac{x_1}{C_1 - x_1} + \frac{\lambda - x_1}{C_2 - (\lambda - x_1)} \quad \frac{\partial L}{\partial x_1} = \frac{C_1}{(C_1 - x_1)^2} - \frac{C_2}{(C_2 - (\lambda - x_1))^2}$$

Relembrar regra das derivadas:  $\left(\frac{u}{v}\right)' = \frac{u'v - uv'}{v^2}$

## Encaminhamento ótimo com bifurcação de fluxos (exemplo)



Fazendo

$$\frac{\partial L}{\partial x_1} = \frac{C_1}{(C_1 - x_1)^2} - \frac{C_2}{(C_2 - (\lambda - x_1))^2} = 0$$

temos

$$x_1^* = \frac{\sqrt{C_1} \left[ \lambda - (C_2 - \sqrt{C_1 C_2}) \right]}{\sqrt{C_1} + \sqrt{C_2}}$$

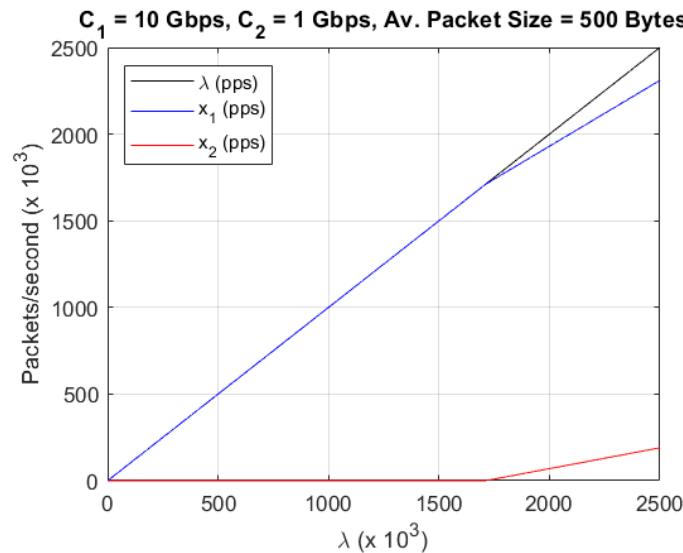
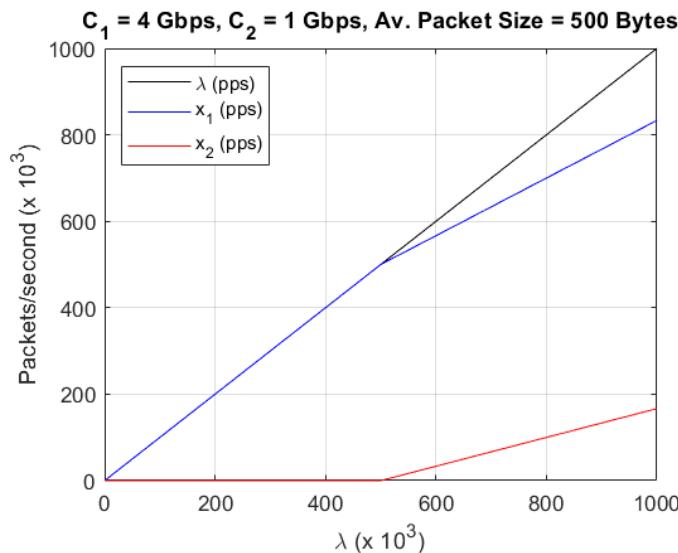
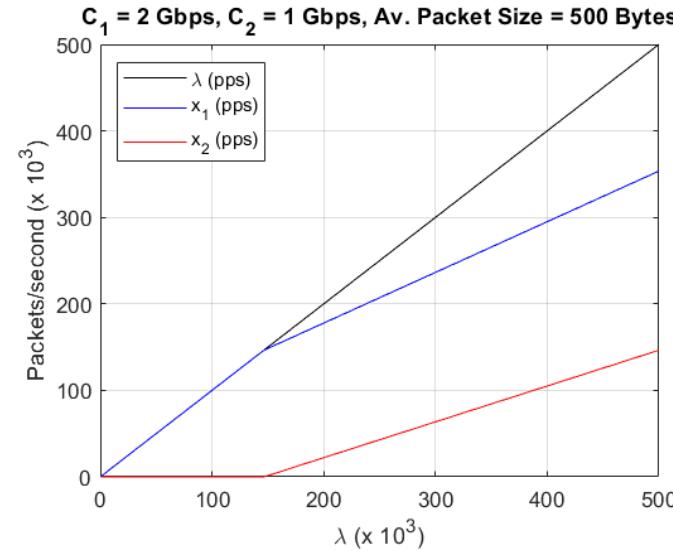
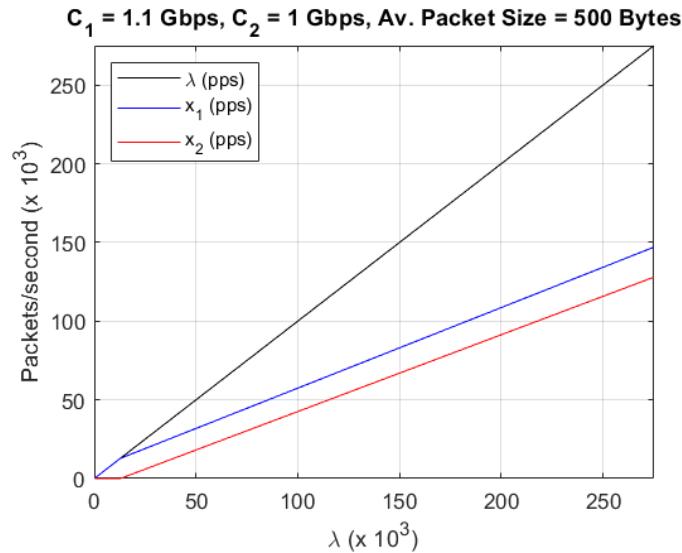
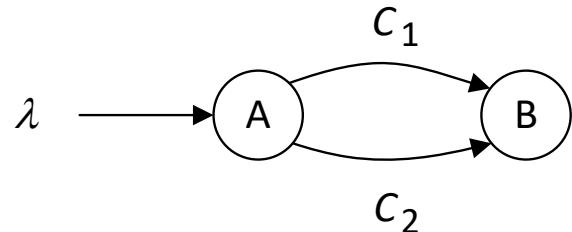
$$x_2^* = \frac{\sqrt{C_2} \left[ \lambda - (C_1 - \sqrt{C_1 C_2}) \right]}{\sqrt{C_1} + \sqrt{C_2}}$$

Assumindo que  $C_1 \geq C_2$  temos dois casos possíveis:

$\lambda > C_1 - \sqrt{C_1 C_2}$  →  $0 < x_1^* < \lambda$  ,  $0 < x_2^* < \lambda$  solução ótima: fluxo bifurcado por  $C_1$  e  $C_2$

$\lambda < C_1 - \sqrt{C_1 C_2}$  →  $x_1^* = \lambda$  ,  $x_2^* = 0$  solução ótima: fluxo encaminhado apenas por  $C_1$

# Encaminhamento ótimo com bifurcação de fluxos (exemplo)



## Encaminhamento ótimo - caso geral

- No encaminhamento ótimo, os fluxos em cada percurso são definidos por forma a otimizar uma função de custo que representa o desempenho da rede:

$$\sum_{(i,j)} D_{ij}(F_{ij})$$

onde  $F_{ij}$  representa o fluxo total na ligação  $(i, j)$ , em pacotes por segundo, e a função  $D_{ij}$  é monótona crescente.

- Uma função  $D_{ij}$  usada com frequência é a obtida com base na aproximação de Kleinrock:

$$D_{ij}(F_{ij}) = \frac{F_{ij}}{C_{ij} - F_{ij}} + d_{ij}F_{ij}$$

onde  $C_{ij}$  é a capacidade da ligação  $(i,j)$ , em pacotes por segundo, e  $d_{ij}$  é o atraso de propagação e processamento na ligação  $(i,j)$ , em segundos.

# Encaminhamento ótimo - caso geral

- $W$  - conjunto dos pares OD (origem - destino) de todos os fluxos  
 $\lambda_w$  - fluxo total do par OD  $w$   
 $P_w$  - conjunto de todos os percursos do nó origem para o nó destino do par OD  $w$   
 $x_p$  - fluxo encaminhado no percurso  $p$

O encaminhamento ótimo é dado pelo seguinte problema de otimização:

$$\text{Minimizar: } D(x) = \sum_{(i,j)} D_{ij} \left( \sum_{\substack{\text{todos os percursos } p \\ \text{contendo } (i,j)}} x_p \right)$$

Sujeito a:

$$\sum_{p \in P_w} x_p = \lambda_w , \forall w \in W$$

$$x_p \geq 0 , \forall p \in P_w, \forall w \in W$$

# Solução para o encaminhamento ótimo

Define-se o comprimento da primeira derivada do percurso  $p \in P_w$  dado por:

$$\frac{\partial D(x)}{\partial x_p} = \sum_{\substack{\text{todas as ligações } (i,j) \\ \text{no percurso } p}} D'_{ij}$$

Prova-se que um vetor de fluxos  $x^* = \{x_p^*, \forall p \in P_w\}$  para o par OD  $w$  é ótimo se e só se:

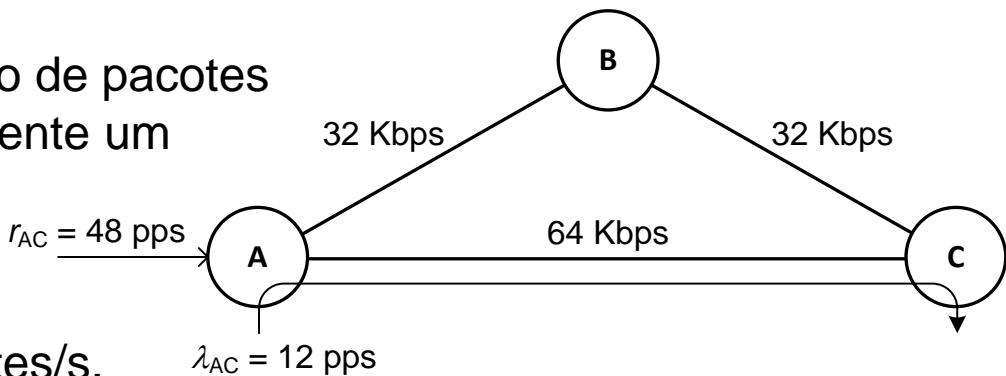
$$x_p^* > 0 \Rightarrow \frac{\partial D(x^*)}{\partial x_{p'}} \geq \frac{\partial D(x^*)}{\partial x_p}, \forall p' \in P_w$$

O fluxo ótimo é positivo apenas nos percursos  $p \in P_w$  com um comprimento de primeira derivada mínimo.

Assim, os percursos usados no encaminhamento ótimo têm comprimento de primeira derivada igual.

## Exemplo 3

Considere a rede com comutação de pacotes da figura. A rede suporta inicialmente um único fluxo de 12 pacotes/s no percurso direto AC. Admita que é oferecido um novo fluxo do nó A para o nó C, de 48 pacotes/s.

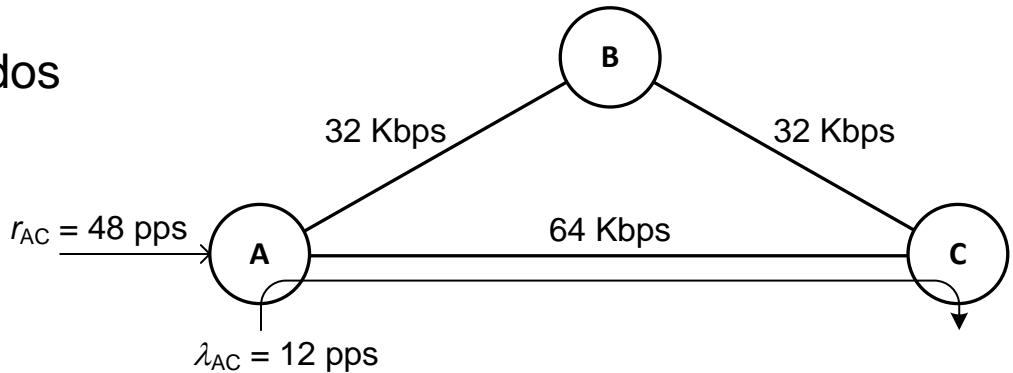


Assuma que ambos os fluxos são caracterizados por intervalos entre chegadas e comprimentos de pacotes independentes e exponencialmente distribuídos, e que o comprimento médio dos pacotes é 125 bytes.

- Calcule o atraso médio total dos pacotes (isto é, o atraso médio calculado sobre todos fluxos), quando o novo fluxo é encaminhado em igual percentagem pelos dois percursos possíveis.
- Admitindo que o novo fluxo (e apenas o novo) pode ser bifurcado pelos dois percursos possíveis, calcule os fluxos ótimos que minimizam o atraso médio total dos pacotes e determine o atraso médio resultante.

## Exemplo 3

(a) Calcule o atraso médio total dos pacotes (isto é, o atraso médio calculado sobre todos fluxos), quando o novo fluxo é encaminhado em igual percentagem pelos dois percursos possíveis.



$$\mu_{AB} = \mu_{BC} = \frac{32000}{125 \times 8} = 32 \text{ pps}$$

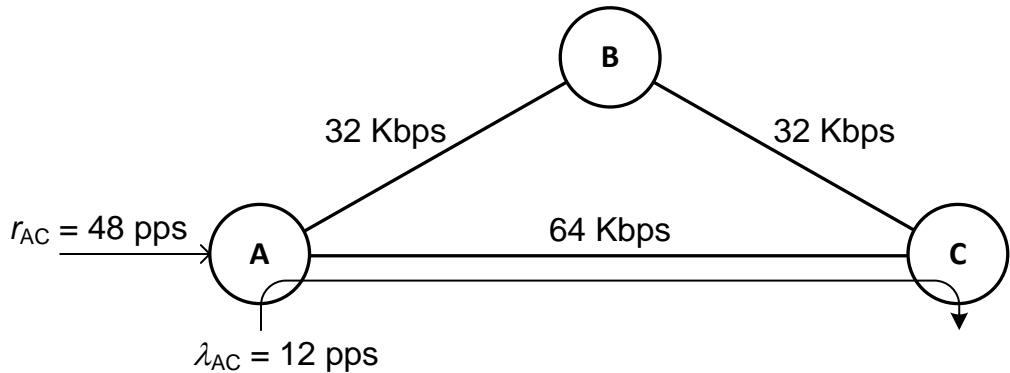
$$\mu_{AC} = \frac{64000}{125 \times 8} = 64 \text{ pps}$$

$$W = \frac{L}{\gamma} = \frac{L_{AB} + L_{BC} + L_{AC}}{\gamma} = \frac{\frac{\lambda_{AB}}{\mu_{AB} - \lambda_{AB}} + \frac{\lambda_{BC}}{\mu_{BC} - \lambda_{BC}} + \frac{\lambda_{AC}}{\mu_{AC} - \lambda_{AC}}}{\gamma}$$

$$= \frac{\frac{24}{32 - 24} + \frac{24}{32 - 24} + \frac{24 + 12}{64 - (24 + 12)}}{48 + 12} = 0.121 \text{ seg.}$$

## Exemplo 3

(b) Admitindo que o novo fluxo (e apenas o novo) pode ser bifurcado pelos dois percursos possíveis, calcule os fluxos ótimos que minimizam o atraso médio total dos pacotes e determine o atraso médio resultante.



$$L = \frac{x_1}{32 - x_1} + \frac{x_1}{32 - x_1} + \frac{x_2 + 12}{64 - (x_2 + 12)}$$

$$\frac{\partial L}{\partial x_1} = \frac{32}{(32 - x_1)^2} + \frac{32}{(32 - x_1)^2} + 0 = \frac{64}{(32 - x_1)^2}$$

$$\frac{\partial L}{\partial x_2} = 0 + 0 + \frac{64}{(52 - x_2)^2} = \frac{64}{(52 - x_2)^2}$$

$$\begin{cases} \frac{\partial L}{\partial x_1} = \frac{\partial L}{\partial x_2} \\ x_1 + x_2 = 48 \end{cases} = \begin{cases} \frac{64}{(32 - x_1)^2} = \frac{64}{(52 - x_2)^2} \\ x_1 + x_2 = 48 \end{cases} = \begin{cases} \frac{8}{32 - x_1} = \frac{8}{52 - x_2} \\ x_2 = 48 - x_1 \end{cases} = \begin{cases} x_1 = 14 \text{ pps} \\ x_2 = 34 \text{ pps} \end{cases}$$

$$W = \frac{L}{\gamma} = \frac{\frac{14}{32 - 14} + \frac{14}{32 - 14} + \frac{34 + 12}{64 - (34 + 12)}}{48 + 12} = 0.069 \text{ seg.}$$



# **Controlo de Fluxos em Redes com Comutação de Pacotes**

Desempenho e Dimensionamento de Redes

Prof. Amaro de Sousa ([asou@ua.pt](mailto:asou@ua.pt))

DETI-UA, 2020/2021

# Sumário do Módulo

Primeira Parte:

- Noções básicas de controlo de fluxos em redes com comutação de pacotes
- Controlo de fluxos de pacotes baseado em janelas extremo-a-extremo

Segunda Parte:

- Mecanismos de controlo de taxas de transmissão de fluxos de pacotes
- Atribuição de taxas de transmissão a fluxos de pacotes segundo o princípio de equidade do tipo max-min



# **Controlo de Fluxos em Redes com Comutação de Pacotes**

**Primeira parte:**

- **Noções básicas de controlo de fluxos em redes com comutação de pacotes**
- **Controlo de fluxos de pacotes baseado em janelas extremo-a-extremo**

# Controlo de fluxo - introdução

O tráfego efetivo reflete a quantidade de serviço suportada por uma rede com comutação de pacotes.

O atraso médio reflete a qualidade de serviço proporcionada por uma rede com comutação de pacotes.

---

Controlo de fluxo: mecanismo de realimentação que estabelece um compromisso entre o tráfego efetivo e o atraso médio por forma a manter o atraso médio dentro de limites aceitáveis:

- Quando o tráfego oferecido é reduzido, é aceite na sua totalidade pelo algoritmo de controlo de fluxo e, neste caso,

$$\text{tráfego efetivo} = \text{tráfego oferecido}$$

- Quando o tráfego oferecido é excessivo, o algoritmo de controlo de fluxo rejeita parte dele e, neste caso,

$$\text{tráfego efetivo} = \text{tráfego oferecido} - \text{tráfego rejeitado}$$

- À medida que o algoritmo de encaminhamento aumenta o atraso médio, o controlo de fluxo reduz o tráfego efetivo.

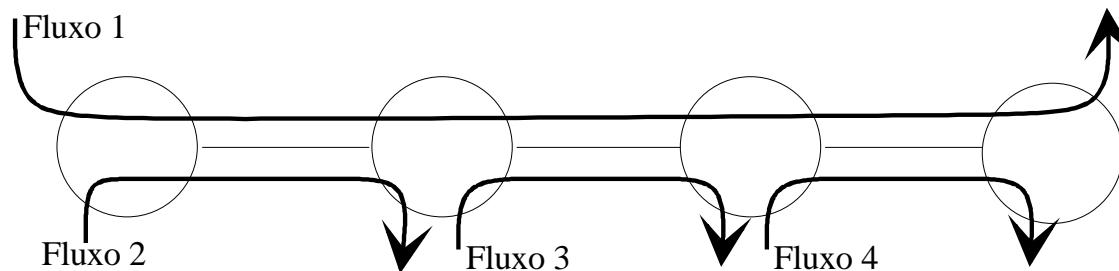
# Controlo de fluxo - introdução

Os algoritmos de controlo de fluxo devem idealmente observar os seguintes requisitos:

- Estabelecer um bom compromisso entre:
  - a quantidade de serviço (o tráfego efetivo, sujeito eventualmente à garantia de uma taxa de transmissão mínima) e
  - a qualidade de serviço (medida, por exemplo, a partir do atraso médio e da taxa de pacotes perdidos)
- Garantir um tratamento equitativo dos diferentes fluxos de pacotes, ao fornecer a qualidade de serviço requerida.

# Gestão de recursos: tráfego efetivo vs. equidade

Considere-se o exemplo da figura assumindo que a capacidade de cada ligação é 100.



Tráfego efetivo máximo:

$$\text{Fluxo 1} = 0, \text{ Fluxos 2,3,4} = 100$$

$$\text{Tráfego efetivo total} = 0 + 100 + 100 + 100 = 300$$

Partilha equitativa dos recursos:

$$\text{Fluxo 1} = 25, \text{ Fluxos 2,3,4} = 75$$

$$\text{Tráfego efetivo total} = 25 + 75 + 75 + 75 = 250$$

Máxima equidade (i.e., mesma taxa de transmissão a todos os fluxos):

$$\text{Fluxos 1,2,3,4} = 50$$

$$\text{Tráfego efetivo total} = 50 + 50 + 50 + 50 = 200$$

# Controlo de fluxo através de janelas

- Considere um fluxo de pacotes de um emissor A para um recetor B.
- Por cada pacote recebido, o recetor B notifica o emissor A através do envio para A de uma permissão:
  - Uma permissão pode ser transmitida num pacote de controlo dedicado ou pode ser encavalitada (*piggybacked*) num pacote de dados enviado no sentido contrário.
- Quando recebe uma permissão, o emissor A fica autorizado a enviar mais um pacote para o recetor B.
- Um esquema de controlo de fluxos pode ser combinado com um protocolo ARQ (*Automatic Repeat Request*) de controlo de erros
  - neste caso, os pacotes são numerados (*sequence numbers*) e as permissões indicam o número de pacotes recebidos (*acknowledgment numbers*) sem erros

# Controlo de fluxo através de janelas

- Um fluxo de pacotes entre o emissor A e o recetor B diz-se controlada através de janelas se existir um limite máximo para o número de pacotes que, tendo sido transmitidos por A, não foram ainda notificadas como tendo sido recebidos por B.
- O limite máximo é designado por tamanho da janela, ou simplesmente, *janela*.
- O emissor e o recetor podem ser dois nós da rede, um terminal e o nó de entrada da rede ou os dois terminais que estão nos extremos do fluxo.

De seguida, considera-se a estratégia de ***janelas extremo-a-extremo*** (*end-to-end*):

- para cada fluxo de pacotes, o controlo de fluxos é implementado entre o seu emissor e o seu recetor
- estratégia usada pelo TCP nas redes TCP/IP

## Janelas extremo-a-extremo

- No controlo de fluxos através de janelas, a taxa de transmissão do emissor é reduzida à medida que as permissões demoram mais tempo a regressar.
- Assim, se o percurso de encaminhamento do fluxo estiver congestionado, a diferença de tempo entre o envio de cada pacote e a receção da sua permissão aumenta o que obriga o emissor a reduzir a sua taxa de transmissão (aliviando o congestionamento do percurso).
- Além disso, o receptor pode atrasar intencionalmente o envio de permissões para reduzir a taxa de transmissão do fluxo com o objetivo de, por exemplo, evitar a sobrecarga do seu *buffer* de receção.

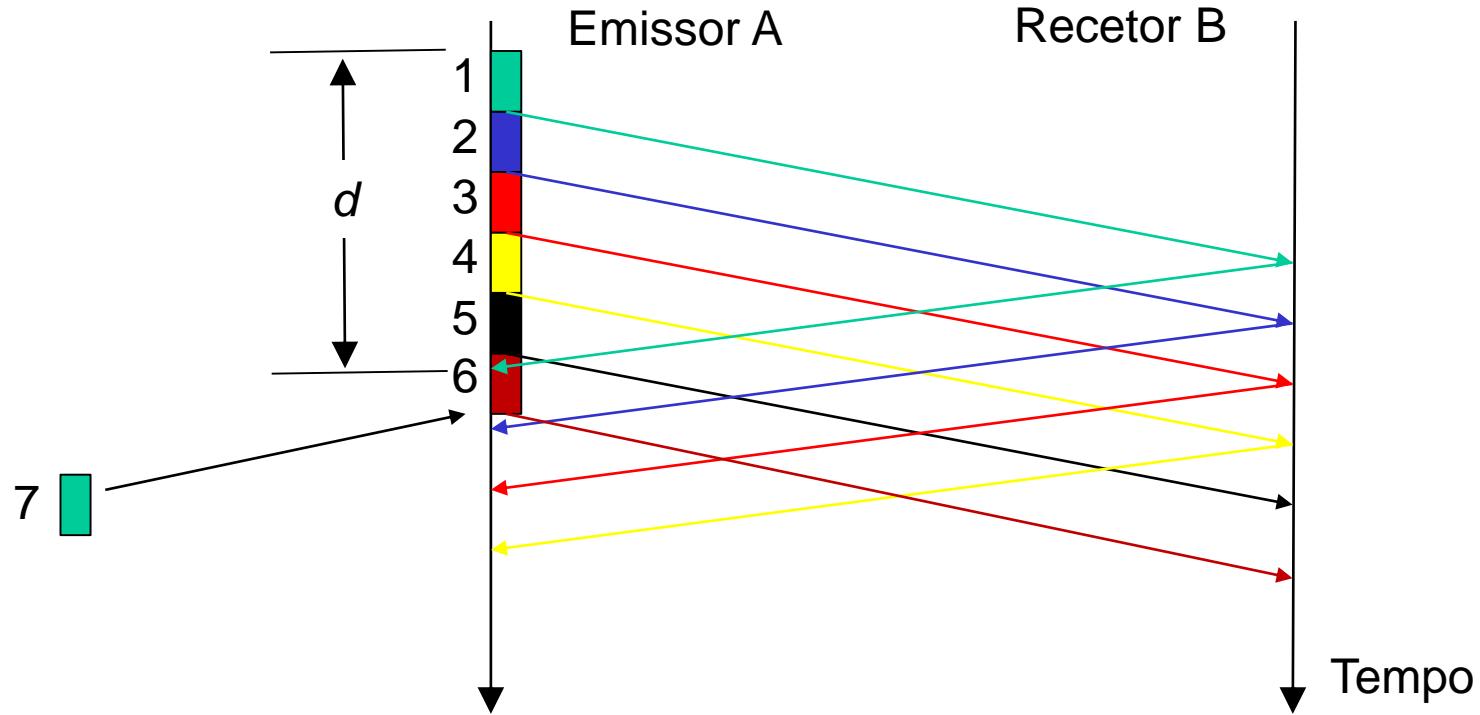
## Janelas extremo-a-extremo

- Considere-se o tamanho da janela dado por  $W$ , em número de pacotes (pode ser outras unidades como por exemplo Bytes no TCP).
  - Cada vez que um pacote é recebido no nó destino, é enviada uma permissão autorizando o envio de um novo pacote.
- Considere-se o atraso de ida-e-volta dado por  $d$  e o tempo de transmissão médio de cada pacote dado por  $X$  (i.e., o tráfego efetivo máximo disponível na rede é  $1/X$ , em pacotes por segundo):
  - ✓ Se  $d \leq WX$ , a transmissão de  $W$  pacotes demora mais que o atraso de ida-e-volta; assim, o emissor pode transmitir à velocidade máxima de  $1/X$  pacotes por segundo.
  - ✓ Se  $d > WX$ , o controlo de fluxos está ativo pois o atraso de ida-e-volta é tão elevado que  $W$  pacotes são transmitidos antes da receção da permissão relativa ao primeiro dos pacotes.

Então, o ritmo de transmissão é dado por:  $r = \min\left\{\frac{1}{X}, \frac{W}{d}\right\}$

## Ilustração das janelas extremo-a-extremo

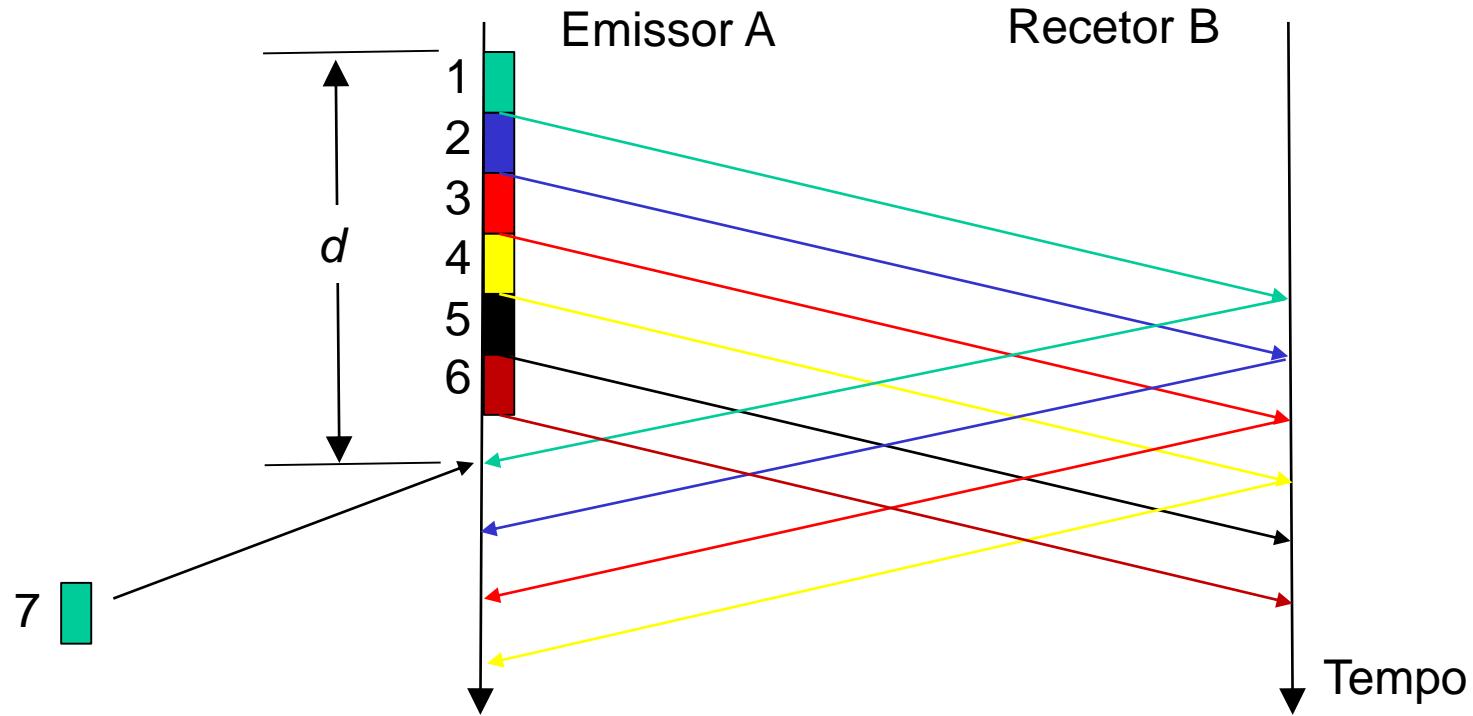
Considere-se  $W = 6$  pacotes do emissor A para o recetor B.



$d \leq WX$  (a transmissão de 6 pacotes demora mais tempo que o atraso de ida-e-volta  $d$ )  $\rightarrow$  o 7º pacote pode ser transmitido logo após o 6º pacote

## Ilustração das janelas extremo-a-extremo

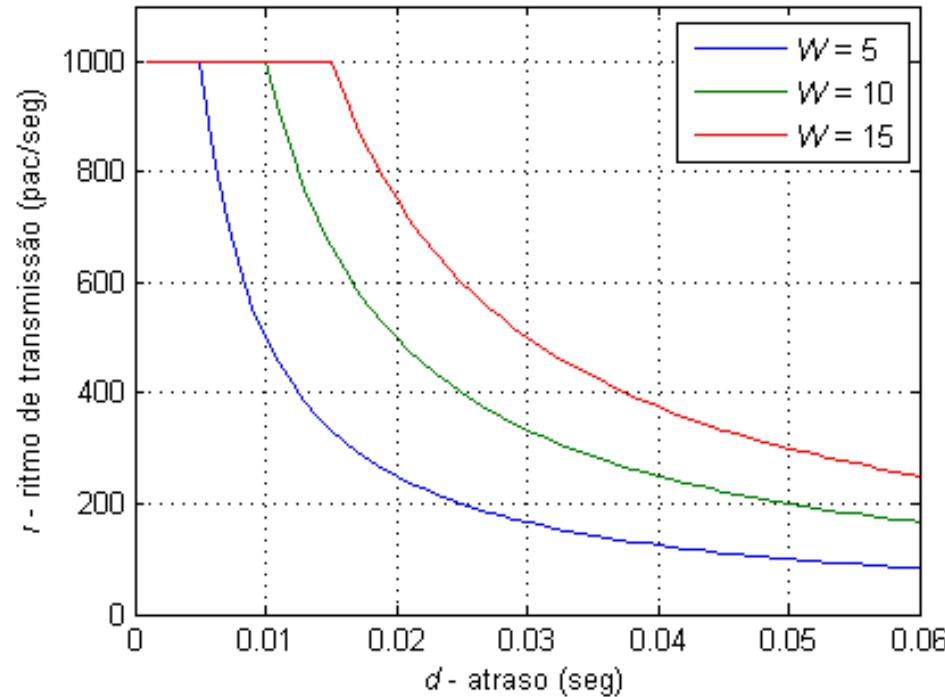
Considere-se  $W = 6$  pacotes do emissor A para o recetor B.



$d > WX$  (a transmissão de 6 pacotes demora menos tempo que o atraso de ida-e-volta  $d$ )  $\rightarrow$  o 7º pacote só pode ser transmitido quando o emissor A recebe a permissão do 1º pacote

## Janelas extremo-a-extremo

Exemplo:  $X = 1$  mseg. e janela  $W = 5, 10$  e  $15$  pacotes.



$$r = \min\left\{\frac{1}{X}, \frac{W}{d}\right\}$$

- ✓ Para valores  $d \leq WX$ , o emissor transmite ao ritmo máximo  $r = 1/10^{-3} = 1000$  (em pacotes/segundo)
- ✓ Para valores  $d > WX$ , o controlo de fluxos está ativo e o emissor transmite ao ritmo  $r = W/d$  (em pacotes/segundo)

# Dimensionamento do tamanho da janela

Existe um compromisso entre tráfego efetivo e atraso:

- por um lado, a janela deve ser pequena para limitar o número de pacotes na rede, evitando assim grandes atrasos e congestão;
- por outro, a janela deve ser grande para permitir a transmissão ao ritmo máximo (i.e., tráfego efetivo máximo) a todos os fluxos em condições de tráfego moderado na rede.

De qualquer modo, é sempre desejável que cada fluxo possa transmitir ao ritmo máximo quando não existe nenhum outro fluxo ativo na rede.

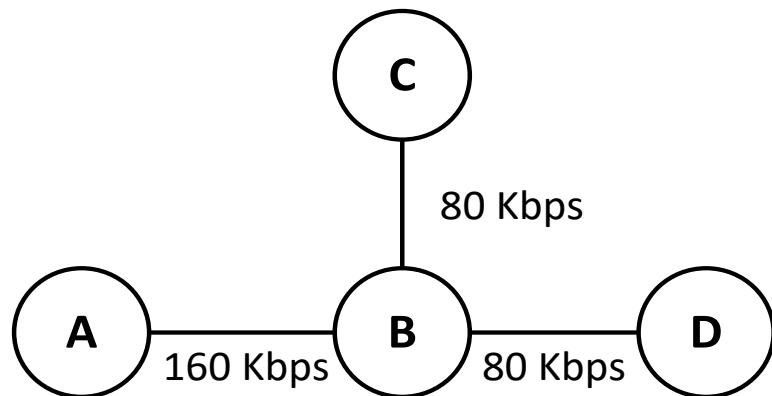
Esta condição impõe um limite inferior ao tamanho da janela. Se  $d \leq WX$  então o fluxo pode transmitir à velocidade máxima pelo que o tamanho da janela (em número de pacotes) deverá ser dado por

$$W = \left\lceil \frac{d}{X} \right\rceil$$

onde  $\lceil z \rceil$  representa o menor inteiro não inferior a  $z$  e  $d$  deverá ser o menor atraso de ida-e-volta proporcionado pela rede.

## Exemplo 1

Considere a rede com comutação de pacotes da figura em que o atraso de propagação de cada ligação é 10 mseg em cada sentido. A rede suporta dois fluxos: A→D com pacotes de tamanho médio 1000 bytes e C→D com pacotes de tamanho médio 500 bytes. A ambos os fluxos é aplicado um mecanismo de controle de fluxos baseado no método das janelas extremo-a-extremo e em ambos os casos, as permissões têm um tamanho fixo de 100 Bytes. Determine o tamanho mínimo (em número de pacotes) das janelas de emissão garantindo que cada fluxo pode emitir ao ritmo máximo quando o outro não está a emitir pacotes.

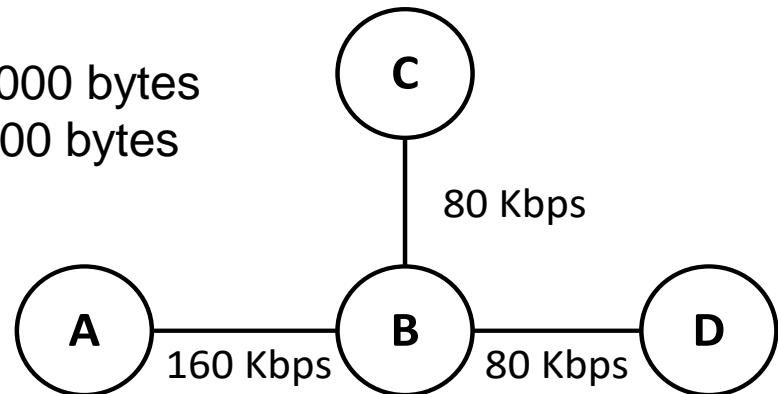


## Exemplo 1 - resolução

A → D com pacotes de tamanho médio 1000 bytes

C → D com pacotes de tamanho médio 500 bytes

$$W \geq \left\lceil \frac{d}{X} \right\rceil$$



$$W_{AD} \geq \left\lceil \frac{\frac{8 \times 1000}{160000} + 0.01 + \frac{8 \times 1000}{80000} + 0.01 + \frac{8 \times 100}{80000} + 0.01 + \frac{8 \times 100}{160000} + 0.01}{\frac{8 \times 1000}{80000}} \right\rceil = \left\lceil \frac{0.205}{0.1} \right\rceil = 3 \text{ pacotes}$$

$$W_{CD} \geq \left\lceil \frac{\frac{8 \times 500}{80000} + 0.01 + \frac{8 \times 500}{80000} + 0.01 + \frac{8 \times 100}{80000} + 0.01 + \frac{8 \times 100}{80000} + 0.01}{\frac{8 \times 500}{80000}} \right\rceil = \left\lceil \frac{0.16}{0.05} \right\rceil = 4 \text{ pacotes}$$

# Limitações do controlo de fluxo baseado em janelas extremo-a-extremo

1. Não permite assegurar uma taxa mínima de transmissão. Quantos mais fluxos forem submetidos na rede, menor é o tráfego efetivo que cada fluxo obtém.
2. Não fornece um controlo adequado do atraso. Considerem-se  $n$  fluxos com controlo de fluxos ativo através de janelas com tamanho fixo  $W_1, \dots, W_n$ . O número total de pacotes e permissões é  $\sum_{i=1}^n W_i$  e o número de pacotes é  $\sum_{i=1}^n \beta_i W_i$  onde  $\beta_i$  é um valor entre 0 e 1.

Pelo teorema de Little, o atraso médio por pacote é

$$T = \frac{\sum_{i=1}^n \beta_i W_i}{\lambda}$$

onde  $\lambda$  é o tráfego efetivo de todos os fluxos. À medida que o número de fluxos aumenta, o tráfego efetivo tende para um valor constante (limitado pela capacidade das ligações). Assim, o atraso médio por pacote aumenta aproximadamente de forma proporcional ao número de fluxos.



# **Controlo de Fluxos em Redes com Comutação de Pacotes**

**Segunda parte:**

- **Mecanismos de controlo de taxas de transmissão de fluxos de pacotes**
- **Atribuição de taxas de transmissão a fluxos de pacotes segundo o princípio de equidade do tipo max-min**

# Controlo de taxas de transmissão

- A função de controlo de fluxos pode atribuir a cada fluxo uma taxa de transmissão máxima compatível com as suas necessidades.
- Essa taxa pode, por exemplo, ser definida na fase de estabelecimento de um circuito virtual (redes IP com RSVP, redes MPLS).
- De seguida, consideram-se dois métodos para controlar a taxa de transmissão:
  - por janelas
  - através de *leaky bucket* (usado pela arquitetura *Integrated Services* (IntServ) nas redes IP)

## Controlo de taxas de transmissão por janelas (I)

- Considere-se que foi atribuída uma taxa de transmissão de  $r$  pacotes por segundo a um determinado fluxo (de um emissor para um receptor).
- Uma possibilidade para garantir esta taxa poderia ser aceitar no emissor, quando muito, um pacote em cada  $1/r$  segundos.
- No entanto, este esquema tende a introduzir grandes atrasos quando a fonte que gera os pacotes no emissor é em rajada.
- Neste caso, é preferível aceitar no emissor  $W$  pacotes em cada  $W/r$  segundos (permite rajadas de  $W$  pacotes).

## Controlo de taxas de transmissão por janelas (II)

Se foi atribuído a um determinado fluxo: (i) uma taxa de transmissão de  $r$  pacotes/segundo e (ii) uma janela de  $W$  pacotes, então:

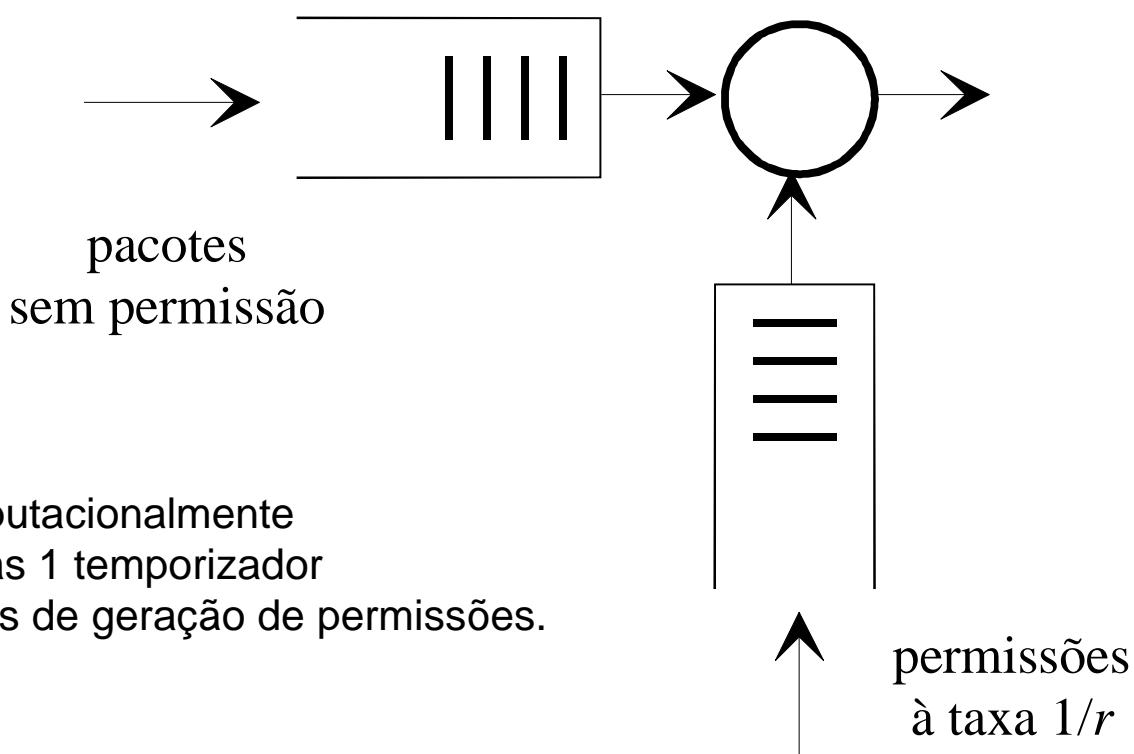
- 1.O emissor mantém um contador  $x$  que indica, em cada instante, o número de pacotes dessa janela que ainda pode ser transmitido ( $x$  é inicializado a  $W$ ).
- 2.Sempre que um pacote é transmitido, o contador  $x$  é decrementado e passados  $W/r$  segundos é novamente incrementado (exige um temporizador por cada pacote transmitido).
- 3.Os pacotes só são enviados para a rede se  $x > 0$  (o número máximo de temporizadores é  $W$ ).

**Nota:** O método do controlo de fluxo por janelas extremo-a-extremo é semelhante a este com a diferença apenas de que o contador é incrementado por cada permissão recebida.

**Desvantagem:** este método é computacionalmente pesado pois exige  $W$  temporizadores simultâneos por cada fluxo.

# Controlo de taxas de transmissão por *leaky bucket*

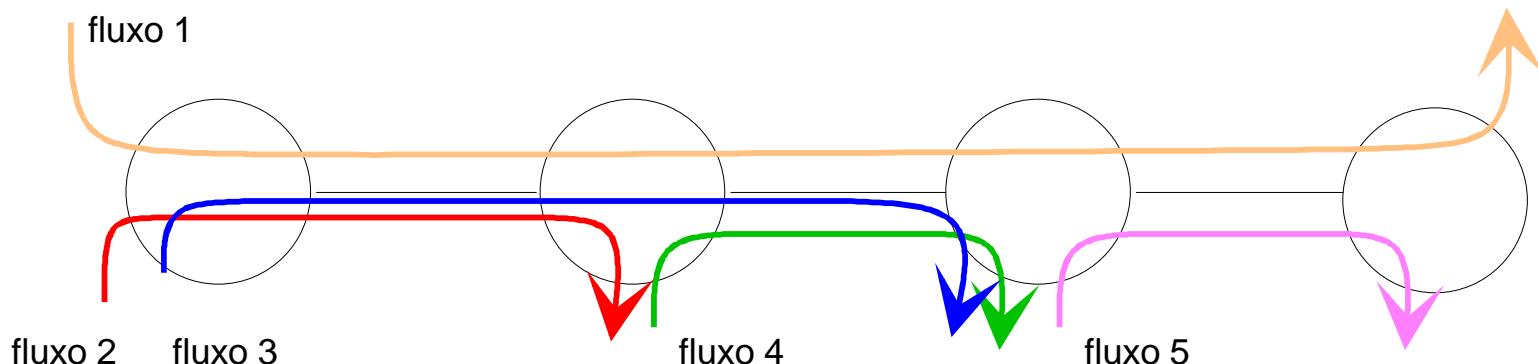
- Neste método, o contador é incrementado periodicamente em cada  $1/r$  segundos, até um máximo de  $W$  pacotes.
- O método pode ser visto da seguinte forma (modelo *leaky bucket*):
  - existe uma fila de espera de pacotes e uma fila de espera de permissões, com capacidade para  $W$  permissões;
  - é gerada uma nova permissão em cada  $1/r$  segundos;
  - os pacotes só são transmitidos quando existe uma permissão disponível na fila de espera respetiva.



**Vantagem:** este método é computacionalmente menos pesado pois exige apenas 1 temporizador por fluxo para definir os instantes de geração de permissões.

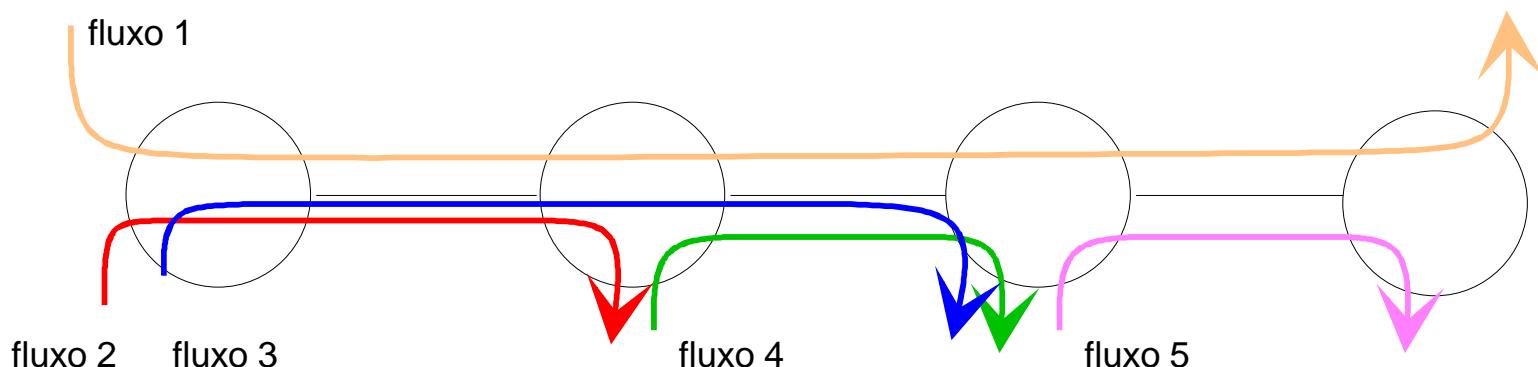
# Atribuição de taxas de transmissão

- Considere a rede da figura em que as ligações têm todas capacidade para 120 pacotes/s.
- Uma solução equilibrada (*fair*) seria atribuir a todos os fluxos uma taxa de  $1/3 \times 120 = 40$  pacotes/s.
- No entanto, não faz sentido restringir a taxa do fluxo 5 a 40 pacotes/s, pois este fluxo pode usar 80 pacotes/s sem prejudicar os fluxos 1, 2, 3 e 4.



# Equidade do tipo *max-min*

- Surge assim o conceito de equidade do tipo max-min (*max-min fairness*).
- Segundo este princípio, maximizam-se os recursos atribuídos aos fluxos que podem usar menos recursos.
- Uma forma alternativa de formular este princípio:
  - Maximizam-se as taxas atribuídas a cada fluxo, respeitando a restrição segundo a qual um incremento na atribuição ao fluxo  $i$  não conduz a uma diminuição da taxa atribuída a qualquer outro fluxo cuja taxa seja menor ou igual que a de  $i$ .



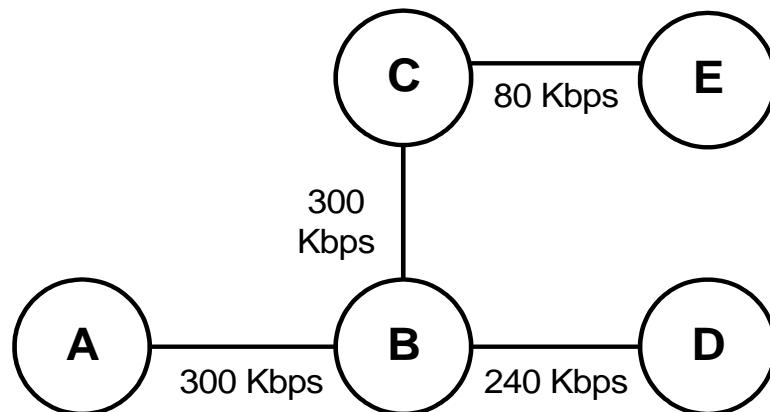
## Exemplo 2

Considere a rede com comutação de pacotes da figura.

A rede suporta 5 fluxos de pacotes: de A para B, de A para C, de A para D, de B para D e de B para E.

A rede permite controlar a taxa de transmissão máxima de cada fluxo através de um qualquer mecanismo adequado.

Calcular que taxas de transmissão máxima se devem atribuir a cada fluxo segundo o princípio de equidade do tipo *max-min*.



## Exemplo 2 - resolução

5 fluxos de pacotes:

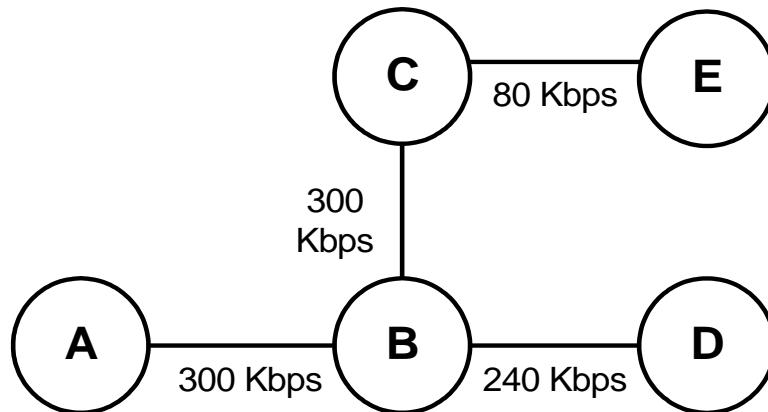
de A para B

de A para C

de A para D

de B para D

de B para E



1<sup>a</sup> iteração:

- a ligação AB atribui  $300/3 = 100$  Kbps por fluxo
- a ligação BC atribui  $300/2 = 150$  Kbps por fluxo
- a ligação BD atribui  $240/2 = 120$  Kbps por fluxo
- a ligação CE atribui  $80/1 = 80$  Kbps por fluxo

O menor valor é o da ligação CE: é atribuído 80 Kbps ao fluxo B→E.

2<sup>a</sup> iteração:

- a ligação AB atribui  $300/3 = 100$  Kbps por fluxo
- a ligação BC atribui  $(300-80)/1 = 220$  Kbps por fluxo
- a ligação BD atribui  $240/2 = 120$  Kbps por fluxo

O menor valor é o da ligação AB: é atribuído 100 Kbps aos fluxos A→B, A→C e A→D.

3<sup>a</sup> iteração:

- a ligação BD atribui  $(240-100)/1 = 140$  Kbps por fluxo

É atribuído 140 Kbps ao fluxo B→D.



# **Mecanismos de Escalonamento e Descarte de Pacotes em Redes com Comutação de Pacotes**

Desempenho e Dimensionamento de Redes

Prof. Amaro de Sousa ([asou@ua.pt](mailto:asou@ua.pt))

DETI-UA, 2020/2021

# Mecanismos de escalonamento e descarte de pacotes

Considere-se uma rede com comutação de pacotes. Em cada interface de saída de cada ligação de um nó, existe uma fila de espera para condicionar temporariamente os pacotes a transmitir pela interface. Em cada interface de saída de cada ligação:

Disciplina de escalonamento: decide a ordem pela qual são transmitidos pela ligação os pacotes de diferentes fluxos que estão na fila de espera

- impõe assim diferentes atrasos médios (*average delays*) a diferentes fluxos ao definir a ordem de transmissão dos pacotes.

Método de descarte de pacotes: decide a forma como os pacotes dos diferentes fluxos são aceitos na fila de espera quando a ligação está ocupada com a transmissão de outro pacote

- impõe assim diferentes taxas de perda de pacotes (*packet loss rates*) a diferentes fluxos ao definir que pacotes são descartados.

# Sumário do Módulo

Primeira Parte:

- Caracterização das disciplinas de escalonamento de pacotes

Segunda Parte:

- Disciplinas de escalonamento de pacotes: FIFO, com prioridades e que funcionam de forma rotativa

Terceira Parte

- Disciplinas de escalonamento de pacotes que funcionam por aproximação ao sistema GPS

Quarta Parte

- Métodos de descarte de pacotes
- Ilustração da combinação de disciplinas de escalonamento com métodos de descarte de pacote na arquitectura *DiffServ* do IETF.



# **Mecanismos de Escalonamento e Descarte de Pacotes em Redes com Comutação de Pacotes**

**Primeira parte:**

- **Caracterização das disciplinas de escalonamento de pacotes**

# Equidade das disciplinas de escalonamento

Quando uma ligação está congestionada (i.e., a sua fila de espera não está vazia), o problema mais básico que se coloca à função de escalonamento é:

divisão de um recurso escasso por fluxos com iguais direitos mas com diferentes necessidades de utilização desse recurso.

Idealmente, a atribuição deve ser feita de acordo com o princípio de equidade *max-min*:

- Os recursos são atribuídos aos fluxos por ordem crescente de necessidade.
- A nenhum fluxo é atribuída uma quantidade de recursos maior do que a sua necessidade.
- A fluxos cuja necessidade não tenha sido satisfeita é atribuída uma igual quantidade de recursos.

# Equidade max-min com direitos iguais

Considere-se:

- um conjunto de fluxos  $1, 2, \dots, n$  com necessidades  $x_1, x_2, \dots, x_n$  e ordenados pelas suas necessidades ( $x_1 \leq x_2 \dots \leq x_n$ );
- uma ligação com capacidade  $C$ .

A atribuição dos recursos da ligação é efetuada do seguinte modo:

- Inicialmente todos os fluxos têm direito a  $d = C/n$
- $d$  é menor que  $x_1$ ?
  - se sim, atribui-se  $d$  a todos os fluxos, i.e., aos fluxos  $1, 2, \dots, n$ ;
  - se não, atribui-se  $x_1$  ao fluxo 1 e os fluxos  $2, 3, \dots, n$  têm direito a  $d = d + (d - x_1)/(n - 1)$
- $d$  é menor que  $x_2$ ?
  - se sim, atribui-se  $d$  aos fluxos  $2, 3, \dots, n$ ;
  - se não, atribui-se  $x_2$  ao fluxo 2 e os fluxos  $3, 4, \dots, n$  têm direito a  $d = d + (d - x_2)/(n - 2)$
- E assim sucessivamente...

## Exemplo 1

Considere-se uma ligação com capacidade de 128 Mbps e 4 fluxos de tráfego de 8, 36, 48 e 128 Mbps. Determine que recursos são atribuídos a cada fluxo pelo princípio de equidade max-min quando todos os fluxos têm direitos iguais.

i) O fluxo 1 tem direito a  $d = 128/4 = 32$  Mbps.

Como o fluxo 1 gera menos que 32 Mbps, o fluxo 1 fica com 8 Mbps. Sobram  $32 - 8 = 24$  Mbps.

ii) O fluxo 2 tem direito a  $d = 32 + 24/3 = 40$  Mbps.

Como o fluxo 2 gera menos que 40 Mbps, o fluxo 2 fica com 36 Mbps. Sobram  $40 - 36 = 4$  Mbps.

ii) O fluxo 3 tem direito a  $d = 40 + 4/2 = 42$  Mb/s.

Como o fluxo 3 (e o fluxo 4) geram mais de 42 Mbps, os fluxos 3 e 4 ficam com 42 Mbps.

## **Equidade max-min com direitos diferentes**

São atribuídos pesos aos fluxos proporcionais aos seus direitos.

A atribuição de recursos é feita de acordo com o princípio *weighted max-min fair*.

Neste caso:

- Os recursos são atribuídos aos fluxos por ordem crescente de necessidade, estando esta normalizada em relação ao peso.
- A nenhum fluxo é atribuído uma quantidade de recursos maior do que a sua necessidade.
- A fluxos cuja necessidade não tenha sido satisfeita é atribuída uma quantidade de recursos proporcional ao seu peso.

## Exemplo 2

Considere uma ligação com capacidade de 128 Mbps e 4 fluxos de tráfego de 8, 36, 48 e 128 Mbps. Determine que recursos são atribuídos a cada fluxo quando os fluxos têm pesos 1, 1, 3 e 3, respetivamente.

i) Fluxo 1 :  $1/(1+1+3+3) \times 128 = 16 \text{ Mbps}$

Fluxo 2 :  $1/(1+1+3+3) \times 128 = 16 \text{ Mbps}$

Fluxo 3 :  $3/(1+1+3+3) \times 128 = 48 \text{ Mbps}$

Fluxo 4 :  $3/(1+1+3+3) \times 128 = 48 \text{ Mbps}$

Atribui-se 8 Mbps ao fluxo 1 (<16 Mbps) e 48 Mbps ao fluxo 3.

Sobram  $(16 - 8) + (48 - 48) = 8 \text{ Mbps}$ .

ii) Fluxo 2 :  $16 + 1/(1+3) \times 8 = 18 \text{ Mbps}$

Fluxo 4 :  $48 + 3/(1+3) \times 8 = 54 \text{ Mbps}$

Atribui-se 18 Mbps ao fluxo 2 (<36 Mbps) e 54 Mbps ao fluxo 4 (<128 Mbps).

## Comparação dos Exemplos 1 e 2

Capacidade da ligação: 128 Mbps

Fluxos:	1	2	3	4
Débito de transmissão (Mbps):	8	36	48	128

---

Pesos:	1	1	1	1
Atribuição (Mbps):	8	36	42	42

---

Pesos:	1	1	3	3
Atribuição (Mbps):	8	18	48	54

- Quando os pesos são todos iguais, os fluxos 1 e 2 conseguem todo o seu débito porque são os fluxos com menor débito de transmissão
- Quando os fluxos 3 e 4 têm 3 vezes maior peso que os fluxos 1 e 2, conseguem maior débito enquanto que o fluxo 2 já não tem todo o seu débito de transmissão.

## Proteção nas disciplinas de escalonamento

Idealmente, a função de escalonamento deve procurar proteger os fluxos bem comportados dos fluxos mal comportados.

Um fluxo mal comportado é um fluxo que envia tráfego a uma taxa superior à taxa a que tem direito (de acordo com o princípio de atribuição de recursos em vigor).

Como veremos à frente:

- as disciplinas de escalonamento do tipo FIFO ou com prioridades não protegem os fluxos bem comportados dos fluxos mal comportados;
- por exemplo, as disciplinas de escalonamento do tipo *round-robin* conseguem.

# Disciplinas de escalonamento

As disciplinas de escalonamento podem classificar-se em disciplinas com e sem conservação de trabalho (*work conserving*):

- numa disciplina com conservação de trabalho, a ligação só está inativa (i.e., não está a ser usada para transmitir pacotes) se não houver qualquer pacote à espera de ser transmitido;
- numa disciplina sem conservação de trabalho, a ligação pode estar inativa mesmo que haja pacotes na fila de espera.

Todas as disciplinas de escalonamento que iremos abordar são disciplinas com conservação de trabalho e são as seguintes:

- (1) por ordem de chegada (FIFO),
- (2) com base em prioridade estrita,
- (3) de uma forma rotativa (RR, WRR, DRR),
- (4) por aproximação ao sistema GPS (WFQ, SCFQ).

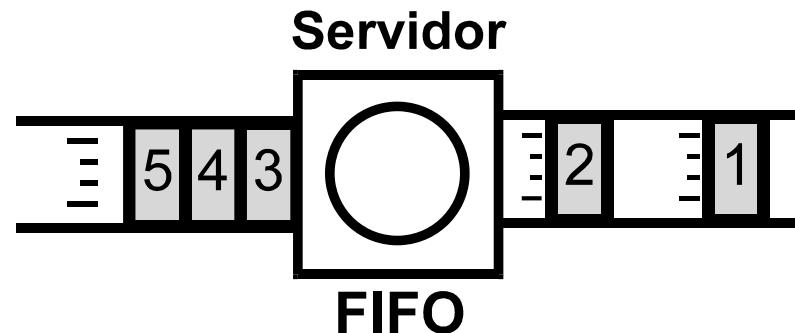


## **Mecanismos de Escalonamento e Descarte de Pacotes em Redes com Comutação de Pacotes**

**Segunda parte:**

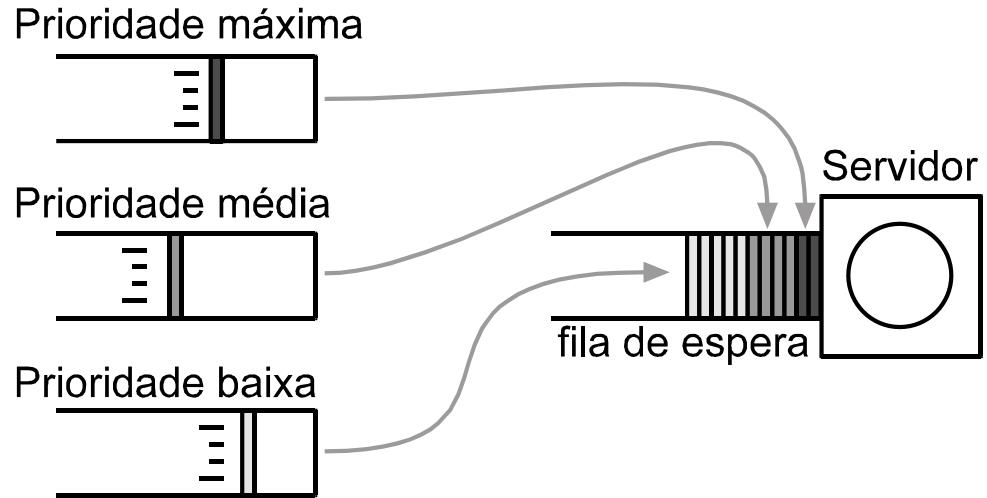
- **Disciplinas de escalonamento de pacotes: FIFO, com prioridades e que funcionam de forma rotativa**

# First-In-First-Out (FIFO)



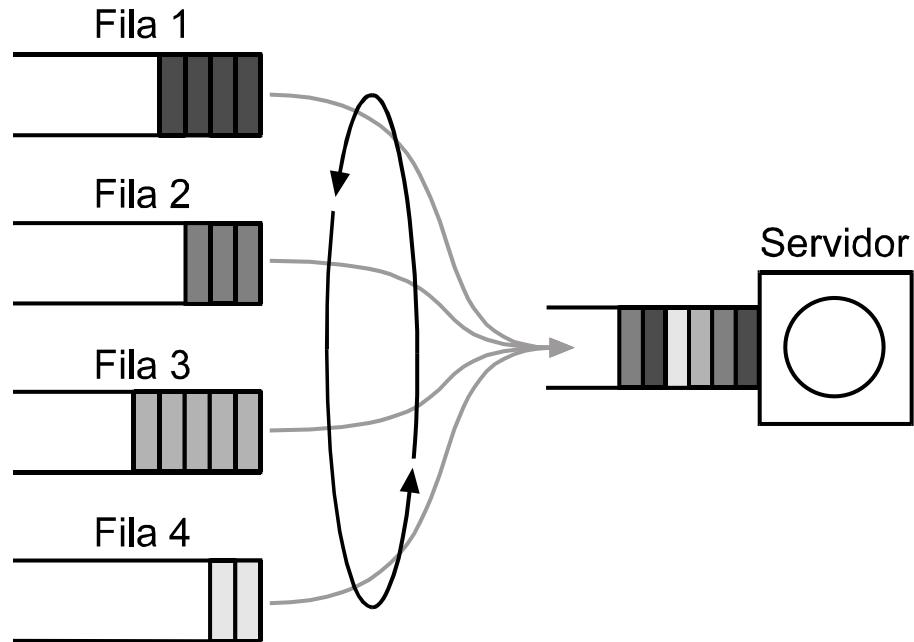
- Os pacotes de todos os fluxos são transmitidos pela sua ordem de chegada.
- Não envolve processamento de ordenação nem de classificação de pacotes.
- Não permite diferenciação de qualidade de serviço (o atraso médio na fila de espera é igual para os pacotes de todos os fluxos).
- Quando a fila de espera não está vazia, fluxos com  $n$  vezes mais tráfego recebem  $n$  vezes mais taxa de serviço pelo que os fluxos bem comportados não são protegidos.

## Prioridade Estrita



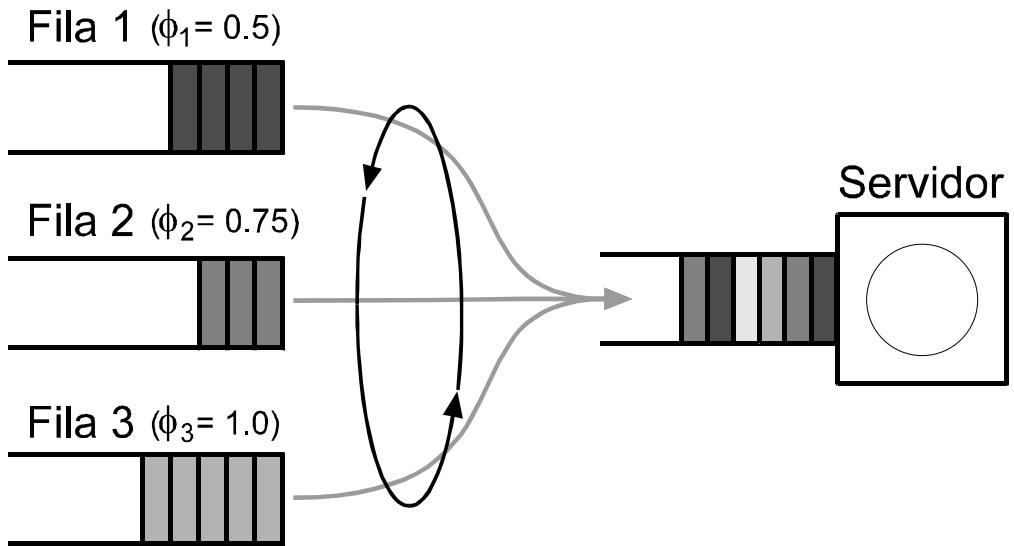
- Os pacotes classificados como de maior prioridade são sempre transmitidos antes dos pacotes de menor prioridade (os pacotes com a mesma prioridade são transmitidos com a disciplina FIFO).
- Não envolve processamento de ordenação.
- Envolve classificação dos pacotes de acordo com a prioridade.
- Permite diferenciação da qualidade de serviço (o atraso médio na fila de espera é menor para os pacotes de maior prioridade).
- Fluxos de pacotes de maior prioridade podem impedir que os fluxos de menor prioridade recebam qualquer serviço.

## Round Robin (RR)



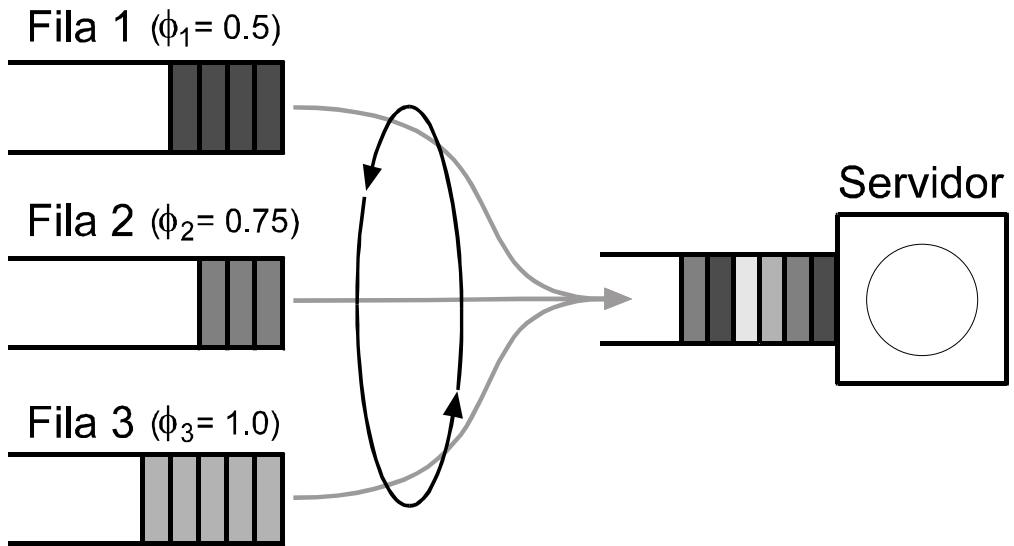
- Existe uma fila por fluxo de pacotes e o algoritmo seleciona um pacote de cada fila não vazia de forma rotativa.
- Não permite diferenciação de qualidade de serviço.
- Ao contrário do FIFO, o RR serve o mesmo número de pacotes de todos os fluxos ativos (i.e. fluxos com pacotes na fila).
- Fluxos de pacotes maiores têm maior taxa de serviço.
- Protege os fluxos bem comportados (os fluxos mal comportados apenas penalizam o seu próprio atraso na fila de espera).

## Weighted Round Robin (WRR)



- É atribuído um peso  $\phi_i$  a cada fila de espera proporcional à taxa de serviço a proporcionar a cada fluxo em situação de congestão.
- Em cada ciclo, o WRR serve um número de pacotes de cada fila de espera tal que a soma dos seus tamanhos (em bytes) é proporcional ao peso da fila.
- É necessário conhecer a priori o comprimento médio dos pacotes.
- A ligação pode ficar demasiado tempo a servir cada fluxo de pacotes o que tem um impacto negativo no *jitter* introduzido pela ligação.

## Weighted Round Robin (WRR)



No exemplo da figura, se o comprimento médio (em Bytes) dos pacotes de cada fluxo for:

$$L_1 = 50, L_2 = 500, L_3 = 1500$$

Os pesos normalizados são:

$$\varphi_1 = 0.5/50 = 1/100 = 60/6000$$

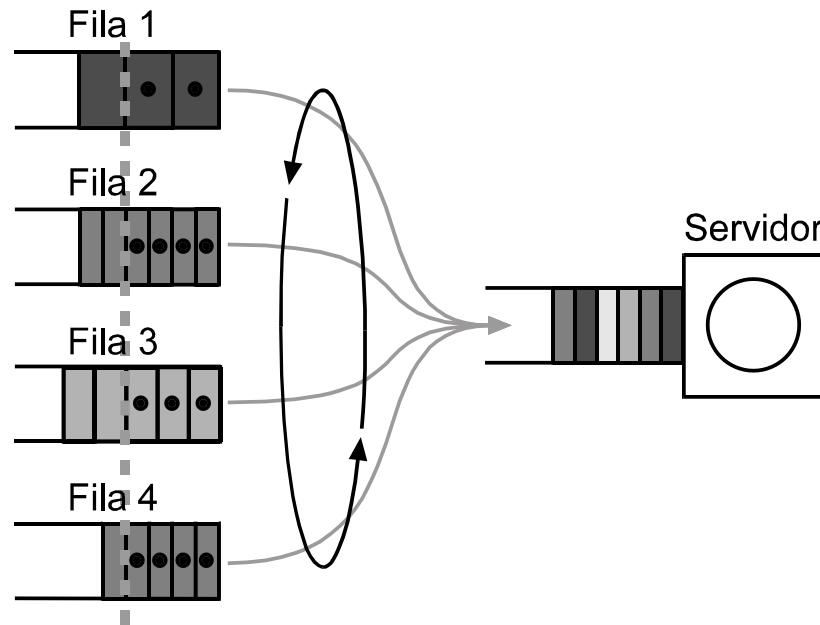
$$\varphi_2 = 0.75/500 = 3/2000 = 9/6000$$

$$\varphi_3 = 1/1500 = 4/6000$$

Número de pacotes por ciclo:

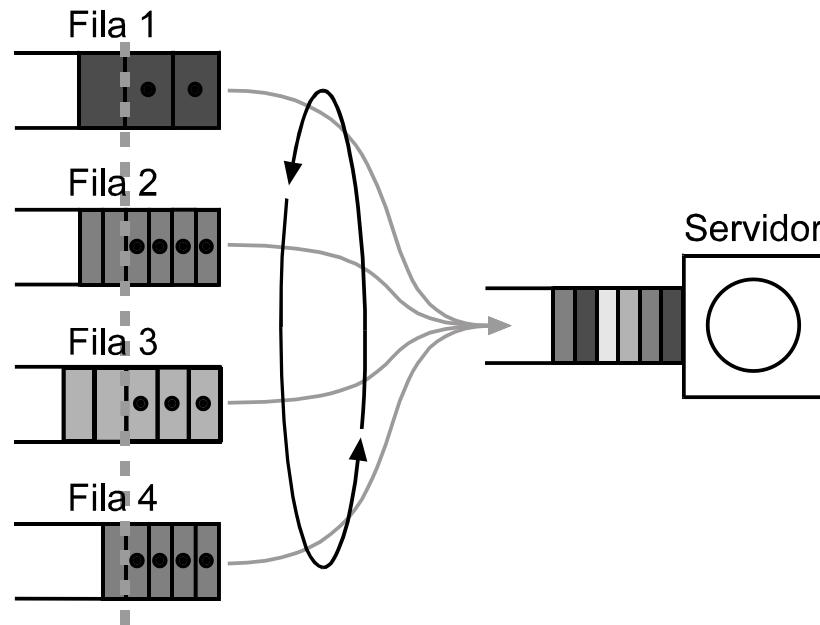
$$\Phi_1 = 60, \Phi_2 = 9, \Phi_3 = 4$$

## Deficit Round Robin (DRR)

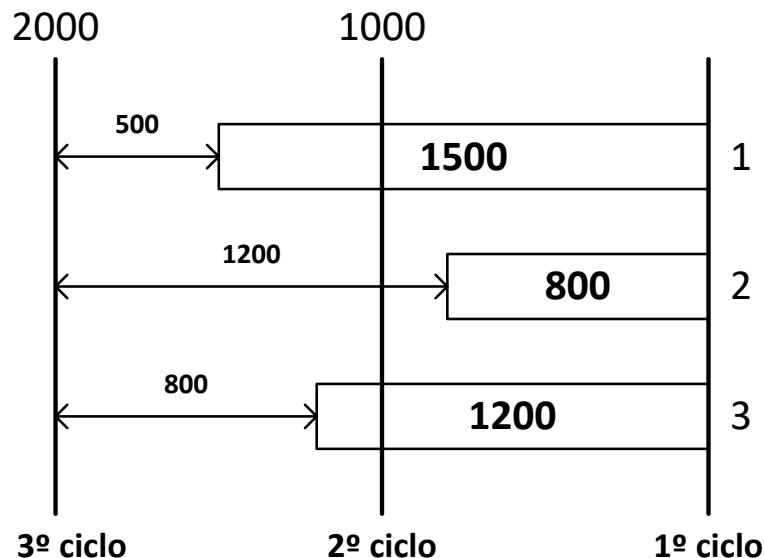


- Em cada ciclo, o DRR serve uma quantidade de bytes até um valor máximo designado por limiar.
- A diferença entre a quantidade servida e o limiar é contabilizada em forma de crédito para o ciclo seguinte.
- Quando uma fila está vazia, o crédito respetivo é colocado a zero.
- Se se considerarem limiares diferentes para as diferentes filas, a taxa de serviço de cada fluxo é proporcional ao limiar da sua fila de espera.
- Ao contrário do WRR, não é necessário saber o comprimento médio dos pacotes.

# Deficit Round Robin (DRR)



limiar = 1000 bytes (para todos os fluxos)



1º Ciclo:

- a) fila 1 não é servida, obtém crédito de 1000
- b) fila 2 é servida, obtém crédito de 200
- c) fila 3 não é servida, obtém crédito de 1000

2º Ciclo:

- a) fila 1 é servida, obtém crédito de 500
- b) fila 2 está vazia, fica com crédito a 0
- c) fila 3 é servida, obtém crédito de 800



# **Mecanismos de Escalonamento e Descarte de Pacotes em Redes com Comutação de Pacotes**

**Terceira parte:**

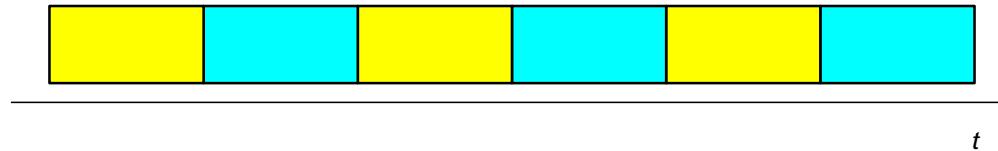
- **Disciplinas de escalonamento de pacotes que funcionam por aproximação ao sistema GPS**

# Generalized Processor Sharing (GPS)

modelo de fluídos



modelo de pacotes



- Algoritmo ideal que proporciona equidade perfeita, baseado num modelo de fluídos, em que o tráfego é considerado infinitamente divisível.
  - Exemplo: num dado instante, 50% da capacidade de uma ligação é utilizada por um fluxo e 50% por outro fluxo.
- Existe uma fila de espera por fluxo e é atribuído um peso  $\phi_i$  a cada fluxo.
- Quando um pacote chega a uma fila, se nenhum outro pacote da mesma fila estiver a ser transmitido, este começa imediatamente a ser transmitido, em paralelo com os pacotes das outras filas, a uma taxa de serviço proporcional ao seu peso.
- É um algoritmo impossível de realizar na prática, mas constitui uma boa base teórica para o desenvolvimento de outros algoritmos.

## Exemplo 3

Considere-se uma ligação de 64 Kbps com 2 filas de espera de pesos  $\phi_1 = 3$  e  $\phi_2 = 1$ , em que os 2 fluxos de pacotes são servidos pela disciplina de escalonamento ideal GPS. Chegam a esta ligação os seguintes pacotes:

- pacote 1 à fila 1 com 62 Bytes em  $t = 0$ ,
- pacote 1 à fila 2 com 32 Bytes em  $t = 4$  ms e
- pacote 2 à fila 1 com 18 Bytes em  $t = 6$  ms.

Determinar os instantes em que os pacotes são servidos (i.e., os instantes de tempo em que termina a transmissão de cada pacote).

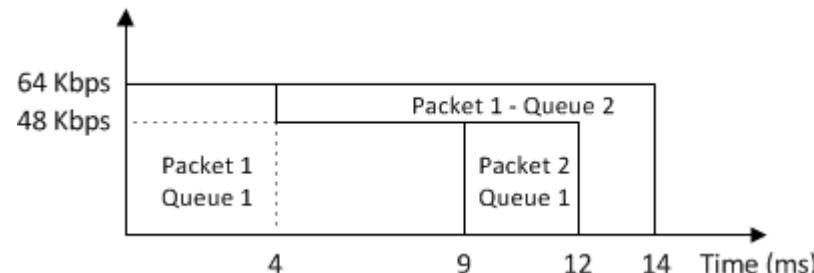
Ligaçāo: 64 Kbps

2 filas de espera:  $\phi_1 = 3$  e  $\phi_2 = 1$

Chegam:

- pacote 1 à fila 1 com 62 Bytes ( $t = 0$ ),
- pacote 1 à fila 2 com 32 Bytes ( $t = 4$  ms) e
- pacote 2 à fila 1 com 18 Bytes ( $t = 6$  ms).

## Resolução do Exemplo 3



- O pacote 1 da fila 1 é servido inicialmente a 64 Kb/s. Em  $t = 4$  ms, foram servidos  $(64\text{Kb/s}) \times (4\text{ms}) = 256$  bits = 32 Bytes do pacote 1 da fila 1. A partir daqui, a fila 1 é servida a  $(3/4) \times 64$  Kb/s = 48 Kb/s e a fila 2 a  $(1/4) \times 64$  Kb/s = 16 Kb/s.
- Com estas taxas, o pacote 1 da fila 1 demora  $((62-32) \times 8) / (48\text{Kb/s}) = 5$  ms a finalizar a sua transmissão e o pacote 1 da fila 2 demora  $(32 \times 8) / (16\text{Kb/s}) = 16$  ms. Assim, o pacote 1 da fila 1 termina a sua transmissão em  $t = 4 + 5 = 9$  ms. Neste instante, inicia-se a transmissão do pacote 2 da fila 1 porque chegou no instante  $t = 6$  ms.
- O pacote 2 da fila 1 demora  $(18 \times 8) / (48\text{Kb/s}) = 3$  ms a ser transmitido. Assim, o pacote 2 da fila 1 termina a sua transmissão em  $t = 9 + 3 = 12$  ms.
- A partir de  $t = 12$  ms, o pacote 1 da fila 2 é transmitido a 64 Kb/s. Como até este instante foram transmitidos  $(16\text{Kb/s}) \times (8\text{ms}) = 128$  bits = 16 Bytes, os restantes 16 Bytes demoram  $(16 \times 8) / (64\text{Kb/s}) = 2$  ms. Assim, o pacote 1 da fila 2 termina a transmissão em  $t = 12 + 2 = 14$  ms.

# Weighted Fair Queuing (WFQ)

É uma aproximação ao sistema GPS: o WFQ tenta servir os pacotes pela ordem em que terminariam de ser transmitidos no sistema GPS.

Sempre que chega um pacote a uma fila, é atribuído ao pacote um ***Finish Number*** (*FN*) que indica a ordem pela qual ele será enviado relativamente aos outros pacotes.

***Round Number*** (*RN*) é uma variável real que cresce no tempo a uma taxa inversamente proporcional aos pesos dos fluxos ativos.

Num intervalo de tempo  $[\tau_i, \tau_{i+1})$  em que o número de fluxos ativos se mantenha constante:

$$RN(\tau_i + t) = RN(\tau_i) + \frac{1}{\sum_{j \text{ ativos}} \phi_j} t \quad t \in [\tau_i, \tau_{i+1})$$

O *RN* é processado sempre que o número de fluxos ativos se altera:

- quando um pacote chega de um fluxo que não tem pacotes no sistema;
- quando um pacote de um fluxo termina de ser transmitido e o fluxo não tem nenhum outro pacote na fila de espera.

Quando o pacote  $k$  com comprimento  $L_k$  pertencente à fila  $i$  chega, é-lhe atribuído o *finish number*  $FN_{i,k}$  dado por:

$$FN_{i,k} = \max(FN_{i,k-1}, RN) + \frac{L_k / C}{\phi_i}$$

## Self Clock Fair Queuing (SCFQ)

A principal desvantagem do WFQ é o peso computacional do cálculo do *RN*.

Por forma a evitar o cálculo do *RN* do WFQ, o SCFQ substitui este parâmetro pelo valor do *FN* do pacote que está a ser transmitido,  $FN_s$ , qualquer que seja o fluxo a que pertence.

Assim, quando o pacote  $k$  com comprimento  $L_k$  pertencente à fila  $i$  chega, é-lhe atribuído o *finish number*  $FN_{i,k}$  dado por:

$$FN_{i,k} = \max(FN_{i,k-1}, FN_s) + \frac{L_k}{\phi_i}$$

Não se utiliza o valor da capacidade da ligação ( $C$ ), uma vez que não é necessário saber o tempo que o pacote demoraria a ser servido no sistema GPS.

Apesar do SCFQ ser de muito menor complexidade que o WFQ, pode não ser tão justo para pequenos intervalos de tempo (i.e., não se aproxima tão bem ao GPS como o WFQ).

## Exemplo 4

Considere-se uma ligação de 64 Kbps com 2 filas de espera de pesos  $\phi_1 = 3$  e  $\phi_2 = 1$ . Chegam a esta ligação os seguintes pacotes:

- pacote 1 à fila 1 com 62 Bytes em  $t = 0$ ,
- pacote 1 à fila 2 com 32 Bytes em  $t = 4$  ms e
- pacote 2 à fila 1 com 18 Bytes em  $t = 6$  ms.

Determinar os instantes em que os pacotes são servidos (i.e., os instantes de tempo em que termina a transmissão de cada pacote) considerando que os 2 fluxos de pacotes são servidos por uma:

- (a) uma disciplina de escalonamento WFQ
- (b) uma disciplina de escalonamento SCFQ

## Exemplo 4 – resolução de (a)

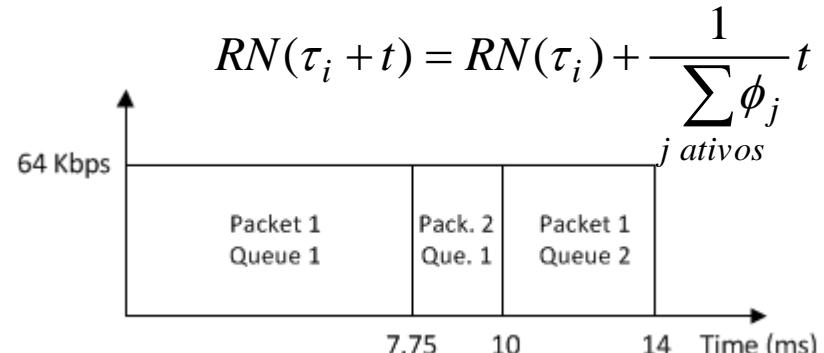
Ligação: 64 Kbps

2 filas de espera:  $\phi_1 = 3$  e  $\phi_2 = 1$

Chegam:

- pacote 1 à fila 1 com 62 Bytes ( $t = 0$ ),
- pacote 1 à fila 2 com 32 Bytes ( $t = 4$  ms) e
- pacote 2 à fila 1 com 18 Bytes ( $t = 6$  ms).

$$FN_{i,k} = \max(FN_{i,k-1}, RN) + \frac{L_k/C}{\phi_i}$$



- Em  $t = 0$  ms,  $RN = 0$  e  $FN_{1,1} = 0 + (62 \times 8) / 64000 / 3 = 2.58 \times 10^{-3}$ . O pacote 1 da fila 1 é transmitido em  $(62 \times 8) / (64 \text{Kb/s}) = 7.75$  ms. Assim, o pacote 1 da fila 1 termina a sua transmissão em  $t = 0 + 7.75 = 7.75$  ms.
- Em  $t = 4$  ms :  $RN = 0 + (4 \times 10^{-3}) / 3 = 1.33 \times 10^{-3}$   
 $FN_{2,1} = 1.33 \times 10^{-3} + (32 \times 8) / 64000 / 1 = 5.33 \times 10^{-3}$
- Em  $t = 6$  ms :  $RN = 1.33 \times 10^{-3} + (6 \times 10^{-3} - 4 \times 10^{-3}) / 4 = 3.33 \times 10^{-3}$   
 $FN_{1,2} = \max(2.58 \times 10^{-3}, 3.33 \times 10^{-3}) + (18 \times 8) / 64000 / 3 = 4.08 \times 10^{-3}$
- Em  $t = 7.75$  ms, como  $FN_{1,2} < FN_{2,1}$ , o pacote 2 da fila 1 começa a ser transmitido. O pacote 2 da fila 1 é transmitido em  $(18 \times 8) / (64 \text{Kb/s}) = 2.25$  ms. Assim, o pacote 2 da fila 1 termina a sua transmissão em  $t = 7.75 + 2.25 = 10$  ms.
- Em  $t = 10$  ms, o pacote 1 da fila 2 começa a ser transmitido. O pacote 1 da fila 2 é transmitido em  $(32 \times 8) / (64 \text{Kb/s}) = 4$  ms. Assim, o pacote 1 da fila 2 termina a sua transmissão em  $t = 10 + 4 = 14$  ms.

## Exemplo 4 – resolução de (b)

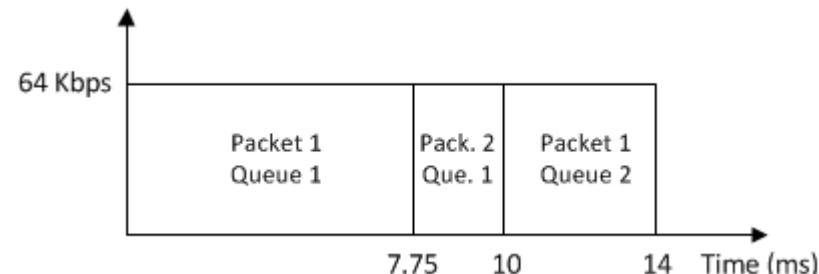
$$FN_{i,k} = \max(FN_{i,k-1}, FN_s) + \frac{L_k}{\phi_i}$$

Ligaçāo: 64 Kbps

2 filas de espera:  $\phi_1 = 3$  e  $\phi_2 = 1$

Chegam:

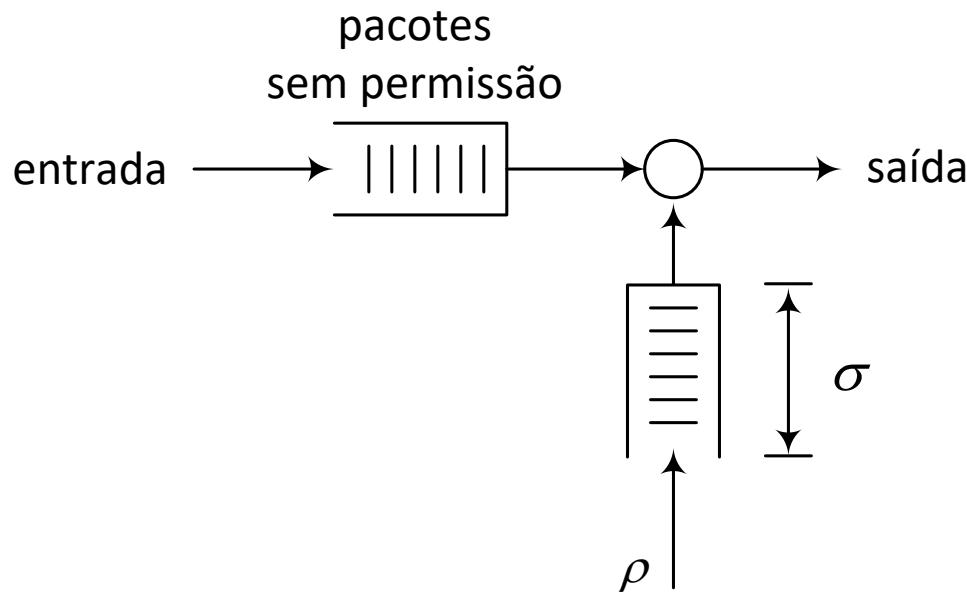
- pacote 1 à fila 1 com 62 Bytes ( $t = 0$ ),
- pacote 1 à fila 2 com 32 Bytes ( $t = 4$  ms) e
- pacote 2 à fila 1 com 18 Bytes ( $t = 6$  ms).



- Em  $t = 0$  ms,  $FN_{1,1} = 0 + (62 \times 8)/3 = 165.3$ . O pacote 1 da fila 1 é transmitido em  $(62 \times 8)/(64 \text{ Kb/s}) = 7.75$  ms. Assim, o pacote 1 da fila 1 termina a sua transmissão em  $t = 0 + 7.75 = 7.75$  ms.
- Em  $t = 4$  ms :  $FN_{2,1} = 165.3 + (32 \times 8)/1 = 421.3$
- Em  $t = 6$  ms :  $FN_{1,2} = \max(165.3, 165.3) + (18 \times 8)/3 = 213.3$
- Em  $t = 7.75$  ms, como  $FN_{1,2} < FN_{2,1}$ , o pacote 2 da fila 1 começa a ser transmitido. O pacote 2 da fila 1 é transmitido em  $(18 \times 8)/(64 \text{ Kb/s}) = 2.25$  ms. Assim, o pacote 2 da fila 1 termina a sua transmissão em  $t = 7.75 + 2.25 = 10$  ms.
- Em  $t = 10$  ms, o pacote 1 da fila 2 começa a ser transmitido. O pacote 1 da fila 2 é transmitido em  $(32 \times 8)/(64 \text{ Kb/s}) = 4$  ms. Assim, o pacote 1 da fila 2 termina a sua transmissão em  $t = 10 + 4 = 14$  ms.

# Desempenho do GPS com controlo de taxa de transmissão por *leaky bucket*

O *Leaky Bucket* é um mecanismo de controlo de taxas de transmissão que permite impor um majorante ao tráfego gerado por um dado fluxo.



Se  $A_i(\tau, t)$  representar a quantidade de tráfego (em Bytes) do fluxo  $i$  que é submetido à rede no intervalo de tempo  $[\tau, t]$ , então:

$$A_i(\tau, t) \leq \sigma_i + \rho_i(t - \tau)$$

# Desempenho do GPS com controlo de taxa de transmissão por *leaky bucket*

Numa disciplina GPS, se designarmos por  $S_i(\tau, t)$  o tráfego (em Bytes) de um fluxo  $i$  que é servido num intervalo de tempo  $[\tau, t]$ , então:

$$S_i(\tau, t) \geq r_i(t - \tau) \quad \text{em que} \quad r_i = \frac{\phi_i}{\sum_j \phi_j} C$$

A quantidade máxima de tráfego em espera  $Q_{i,\max}(t)$  do fluxo  $i$ , desde um instante em que o fluxo não tinha tráfego no sistema ( $\tau = 0$ ) até um qualquer instante  $t$  é:

$$\begin{aligned} Q_{i,\max}(t) &= A_{i,\max}(0, t) - S_{i,\min}(0, t) \\ &= \sigma_i + \rho_i t - r_i t \\ &\leq \sigma_i \quad \Leftarrow r_i \geq \rho_i \end{aligned}$$

O atraso máximo  $D_i$  é o tempo necessário para transmitir todo o tráfego em espera, que na pior das hipóteses é servido à taxa mínima de serviço  $r_i$ . Assim, se  $r_i \geq \rho_i$ , o atraso máximo de qualquer pacote do fluxo  $i$  é:

$$D_i = \frac{\sigma_i}{r_i}$$

# Desempenho do WFQ com controlo de taxa de transmissão por *leaky bucket*

Numa disciplina WFQ, o atraso máximo é maior que no GPS porque a informação é transmitida em pacotes.

Considere um fluxo  $i$  formatado por um *leaky bucket* com parâmetros  $\sigma_i$  e  $\rho_i$  que atravessa  $n$  ligações:

$C_j$  - capacidade da ligação  $j$

$r_i$  - largura de banda reservada para o fluxo  $i$  em todas as ligações ( $r_i \geq \rho_i$ )

$L_i$  - tamanho máximo dos pacotes do fluxo  $i$

$L_{\max}$  - tamanho máximo dos pacotes de todos os fluxos

Prova-se que o atraso máximo ( $D_i$ ) que os pacotes do fluxo  $i$  sofrem é:

$$D_i = \frac{\sigma_i + (n-1)L_i}{r_i} + \sum_{j=1}^n \frac{L_{\max}}{C_j} + \Gamma$$

em que  $\Gamma$  é o atraso total de propagação de todas as ligações.

## Exemplo 5

---

Considere um fluxo de pacotes de comprimento máximo de 200 Bytes formatado por um *leaky bucket* com parâmetros  $\sigma = 1000$  bytes e  $\rho = 150$  Kbps. O fluxo atravessa 8 ligações todas com capacidade 100 Mbps servidas por uma disciplina WFQ. O comprimento máximo dos pacotes de todos os fluxos é de 1500 bytes. O atraso de propagação total é 2 mseg. Qual a taxa (em Mbps) que é necessário reservar em todas as ligações para este fluxo, por forma a garantir um atraso máximo extremo-a-extremo de 20 mseg?

---

$$D_i = \frac{\sigma_i + (n-1)L_i}{r_i} + \sum_{j=1}^n \frac{L_{\max}}{C_j} + \Gamma$$

$$0.02 = \frac{1000 \times 8 + 7 \times 200 \times 8}{r} + 8 \times \frac{1500 \times 8}{100 \times 10^6} + 0.002$$

$$r = \frac{1000 \times 8 + 7 \times 200 \times 8}{0.018 - 8 \times \frac{1500 \times 8}{100 \times 10^6}} = 1127 \text{ Kbps} = 1.127 \text{ Mbps}$$



# **Mecanismos de Escalonamento e Descarte de Pacotes em Redes com Comutação de Pacotes**

**Quarta parte:**

- **Métodos de descarte de pacotes**
- **Ilustração da combinação de disciplinas de escalonamento com métodos de descarte de pacote na arquitectura DiffServ do IETF**

## Métodos de Descarte de Pacotes

Os métodos de descarte de pacotes podem ser classificados quanto a:

- Posição de descarte
- Prioridade de descarte
- Grau de agregação
- Descarte antecipado

# Métodos de Descarte de Pacotes

## *Posição de descarte*

- Cauda da fila – Normalmente usado por omissão; mais simples de implementar (o pacote não chega a entrar na fila).
  - Em muitos casos, a fila tem muitos pacotes pertencentes a poucos fluxos. Se o pacote que chega não pertence a nenhum desses fluxos, a estratégia não é justa.
- Posição aleatória – Escolhe-se aleatoriamente um pacote (entre todos os da fila + o novo) para ser eliminado (computacionalmente pesado).
  - Os fluxos com mais pacotes na fila são mais penalizados: estratégia mais justa.
- Cabeça da fila – Retira-se o pacote mais antigo da fila e aceita-se o que chegou (computacionalmente leve).
  - Tão bom como a posição aleatória em termos de justiça.
  - Útil quando o controle de fluxo é baseado em perdas de pacotes (porquê? relembrar controlo de congestão do TCP!)

# Métodos de Descarte de Pacotes

## Prioridades de descarte

- O emissor ou a rede (o policiador de um domínio DiffServ) podem marcar alguns pacotes com maior prioridade de descarte. Estes, em situação de congestionamento serão os primeiros a ser descartados.
- Quando um pacote é fragmentado e um dos fragmentos é descartado, os restantes fragmentos podem (e devem) também ser descartados pois deixam de ter qualquer utilidade.
  - Podia ser usado no protocolo IP? Relembrar utilização da flag '*more fragments*' e do campo *Fragment Offset*.
- Um método de descarte possível consiste em dar maior prioridade de descarte aos pacotes que passaram por menos ligações (*i.e.*, usaram menos recursos).
  - Este método não pode ser implementado no protocolo IP (porquê? relembrar utilização do campo TTL no IPv4)

# Métodos de Descarte de Pacotes

## **Grau de agregação**

### Agregação de fluxos

- O método de descarte pode considerar os fluxos individualmente ou de forma agregada.
  - Na forma agregada, o método é aplicado a cada pacote do agregado, sem tomar em consideração o fluxo a que pertence.
  - Quanto mais fluxos forem agregados, menor a proteção entre os fluxos pertencentes ao mesmo agregado.

### Agregação da memória dedicada às filas de espera

- Se existe uma fila de espera por fluxo de pacotes e a memória é partilhada por todas as filas, consegue-se uma atribuição de memória *max-min fair* quando se descarta o último pacote da fila mais longa (*i.e.*, da fila com um maior número de pacotes).
  - Com o WFQ, isto corresponde a descartar o pacote com maior *Finish Number* de entre todos os fluxos.

# Métodos de Descarte de Pacotes

## ***Descarte antecipado***

Descarte quando a fila de espera está cheia:

- Quando a fila enche por um largo período (a ligação está muito com-gestionada), múltiplos pacotes são descartados provocando a reação simultânea de todas as ligações TCP afetadas; o tráfego tende a variar ciclicamente entre períodos de baixo débito e períodos de congestão.

Descarte antecipado (RED - Random Early Discard):

- Quando cada pacote chega à fila, ele é descartado com uma probabilidade proporcional à ocupação da fila; evita-se o sincronismo do controle de congestão das ligações TCP.
- Não proporciona diferenciação de qualidade de serviço.

Descarte antecipado pesado (WRED – Weighted RED):

- Atribuem-se diferentes probabilidades de descarte a pacotes pertencentes a diferentes fluxos (ou agregados de fluxos).
- Quanto menor a probabilidade de descarte, menor é a taxa de perda de pacotes que o fluxo (ou o agregado) sofre.

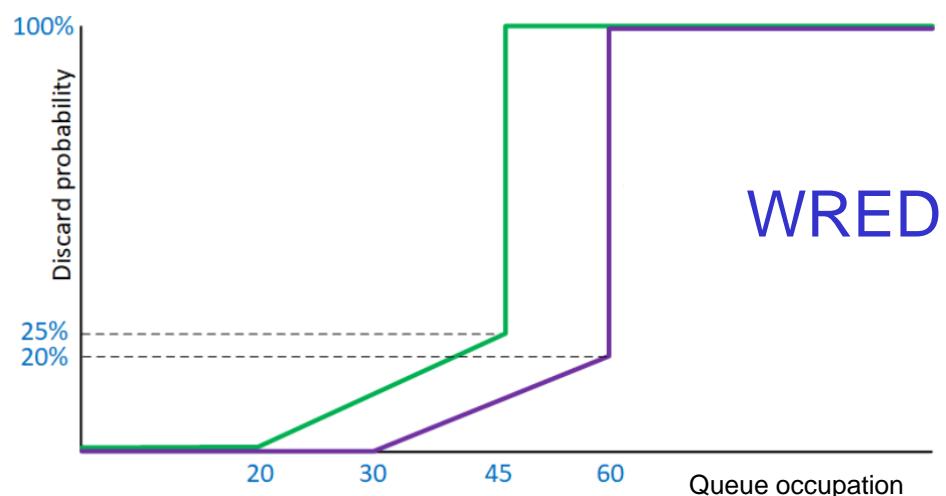
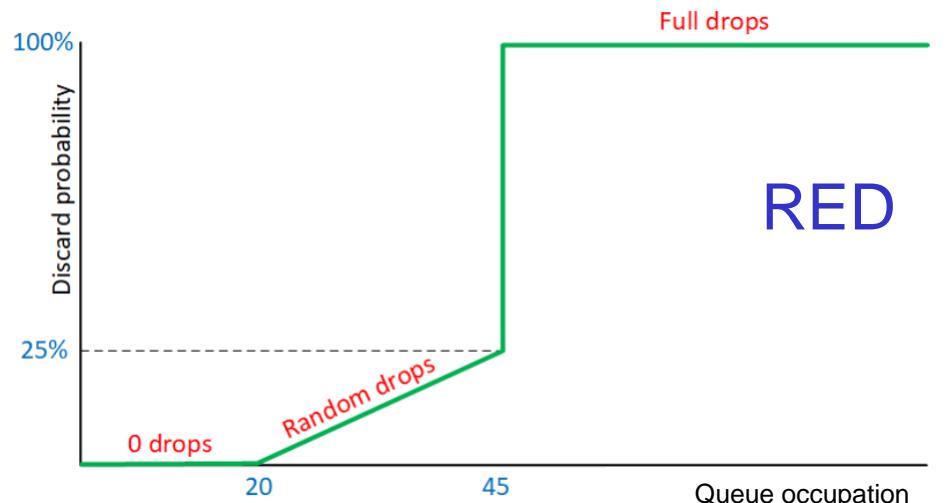
# RED e WRED

No RED:

- Límite Mínimo ( $m$ ): quando um pacote chega e a ocupação da fila  $f$  é menor que o limite mínimo ( $f < m$ ), o pacote é sempre aceite na fila.
- Límite Máximo ( $M$ ): quando um pacote chega e a ocupação da fila  $f$  é maior que o limite máximo ( $f > M$ ), o pacote é sempre descartado.
- Mark Probability Denominator (MPD): quando um pacote chega e a ocupação  $f$  está entre os limites mínimo e máximo ( $m \leq f \leq M$ ), o pacote é descartado com probabilidade  $(f-m)/(M-m) \times MPD$

No WRED:

- São atribuídos diferentes valores de  $m$ ,  $M$  e  $MPD$  a diferentes fluxos (ou agregados de fluxos)



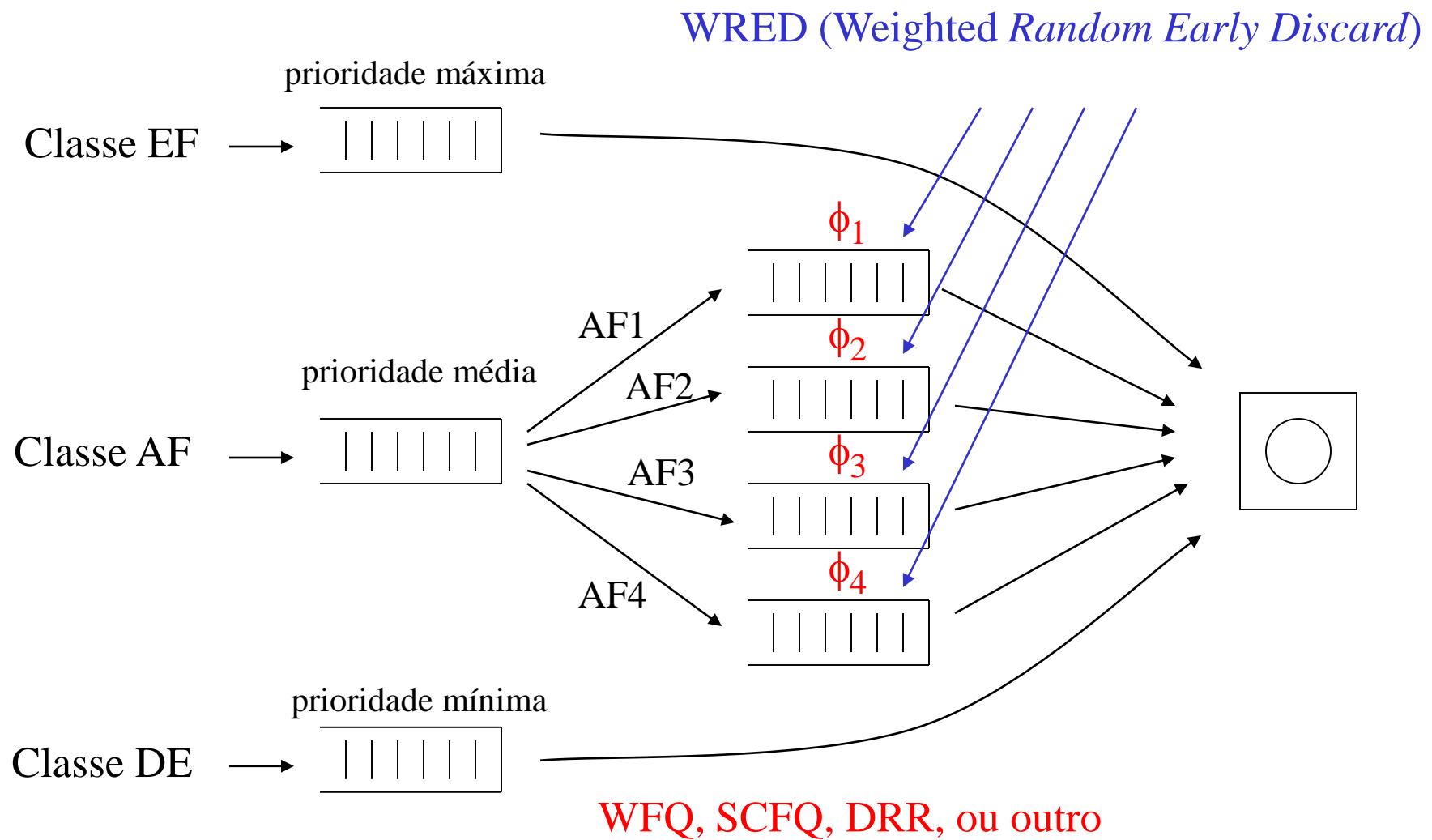
# Exemplo – Arquitectura *DiffServ*

## Classes de Serviço

- *Default (DE)* → DSCP = 000000
  - serviço *best-effort* com uma única fila de espera do tipo FIFO
- *Expedited Forwarding (EF)* → DSCP = 101110
  - serviço tipo “linha alugada virtual”
  - disponibiliza controle de perdas, do atraso e da variância do atraso dentro de uma determinada largura de banda máxima
- *Assured Forwarding (AF)*
  - fornece uma Qualidade de Serviço relativa entre até 4 classes AF
  - em cada classe AF, pode haver até 3 níveis de precedência para descarte de pacotes (em caso de congestionamento)

<i>AF Codepoints</i>	AF1	AF2	AF3	AF4
<i>Low drop precedence</i>	<b>001010</b>	<b>010010</b>	<b>011010</b>	<b>100010</b>
<i>Medium drop precedence</i>	<b>001100</b>	<b>010100</b>	<b>011100</b>	<b>100100</b>
<i>High drop precedence</i>	<b>001110</b>	<b>010110</b>	<b>011110</b>	<b>100110</b>

# Possível Esquema de Escalonamento do DiffServ





## **Resolução de Exercícios de Revisão**

Desempenho e Dimensionamento de Redes

Prof. Amaro de Sousa ([asou@ua.pt](mailto:asou@ua.pt))

DETI-UA, 2020/2021

## Exercício 1

Numa ligação sem fios (wireless) entre dois equipamentos, a probabilidade dos pacotes de dados serem recebidos com erros é de 0.1% em condições normais ou de 10% quando há interferências. A probabilidade de haver interferência é de 2%. Os equipamentos têm a capacidade de verificar na receção se os pacotes de dados foram recebidos com erros ou não.

- (a) Se um pacote de dados for recebido com erros, qual a probabilidade da ligação estar com interferência.
- (b) Os terminais decidem que a ligação está com interferência quando recebem 2 pacotes seguidos com erros. Qual a probabilidade desta decisão estar correta?

## Exercício 1 – resolução da (a)

- (a) Se um pacote de dados for recebida com erros, qual a probabilidade da ligação estar com interferência.
- 

$$P(F_j|E) = \frac{P(E|F_j)P(F_j)}{\sum_{i=1}^n P(E|F_i)P(F_i)} \quad \leftarrow \text{Regra de Bayes}$$

$F_1$  – estado normal

$E$  – pacote recebido com erros

$F_2$  – estado em interferência

$$P(F_1) = 1 - P(F_2) = 98\% = 0.98 \quad P(F_2) = 2\% = 0.02$$

$$P(E|F_1) = 0.1\% = 0.001 \quad P(E|F_2) = 10\% = 0.1$$

$$\begin{aligned} P(F_2|E) &= \frac{P(E|F_2)P(F_2)}{P(E|F_1)P(F_1) + P(E|F_2)P(F_2)} \\ &= \frac{0.1 \times 0.02}{0.001 \times 0.98 + 0.1 \times 0.02} = 0.671 = 67.1\% \end{aligned}$$

## Exercício 1 – resolução da (b)

(b) Os terminais decidem que a ligação está com interferência quando recebem 2 pacotes seguidos com erros. Qual a probabilidade desta decisão estar correta?

---

$$P(F_j|E) = \frac{P(E|F_j)P(F_j)}{\sum_{i=1}^n P(E|F_i)P(F_i)}$$

$F_1$  – estado normal

$E$  – 2 pacotes seguidos com erros

$F_2$  – estado em interferência

$$P(F_1) = 1 - P(F_2) = 98\% = 0.98 \quad P(F_2) = 2\% = 0.02$$

$$P(E|F_1) = 0.001 \times 0.001 = 10^{-6} \quad P(E|F_2) = 0.1 \times 0.1 = 0.01$$

$$\begin{aligned} P(F_2|E) &= \frac{P(E|F_2)P(F_2)}{P(E|F_1)P(F_1) + P(E|F_2)P(F_2)} \\ &= \frac{0.01 \times 0.02}{10^{-6} \times 0.98 + 0.01 \times 0.02} = 0.995 = 99.5\% \end{aligned}$$

## **Exercício 2**

Considere um sistema de transmissão de pacotes ponto-a-ponto de 64 Kbps. Os pacotes têm um tamanho exponencialmente distribuído com média de 500 Bytes e a chegada de pacotes é um processo de Poisson com taxa de 4 pacotes/segundo. Calcule a capacidade mínima da fila de espera (em número de pacotes) para que a taxa de pacotes perdidos seja menor que 2%.

## Exercício 2 - resolução

Este sistema é um M/M/1/X em que X é a capacidade do sistema (a capacidade da fila de espera é  $X - 1$ ). Pela propriedade PASTA, a taxa de pacotes perdidos é igual à probabilidade do sistema estar cheio:

$$\lambda = 4 \text{ pac/s} \quad \mu = 64000/(500 \times 8) = 16 \text{ pac/s}$$

$$\lambda / \mu = 4/16 = 0.25$$

$$P_X = \frac{\frac{\lambda_0 \lambda_1 \dots \lambda_{X-1}}{\mu_1 \mu_2 \dots \mu_X}}{1 + \sum_{n=1}^X \left( \frac{\lambda_0 \lambda_1 \dots \lambda_{n-1}}{\mu_1 \mu_2 \dots \mu_n} \right)} = \frac{\frac{\lambda_0 \lambda_1 \dots \lambda_{X-1}}{\mu_1 \mu_2 \dots \mu_X}}{1 + \frac{\lambda_0}{\mu_1} + \frac{\lambda_0 \lambda_1}{\mu_1 \mu_2} + \dots + \frac{\lambda_0 \lambda_1 \dots \lambda_{X-1}}{\mu_1 \mu_2 \dots \mu_X}} = \frac{0.25^X}{1 + 0.25 + 0.25^2 + \dots + 0.25^X}$$

$$X = 1 \Rightarrow P_1 = 0.25/(1+0.25) = 0.2 = 20\% > 2\%$$

$$X = 2 \Rightarrow P_2 = 0.25^2/(1+0.25+0.25^2) = 0.0476 = 4.76\% > 2\%$$

$$X = 3 \Rightarrow P_3 = 0.25^3/(1+0.25+0.25^2+0.25^3) = 0.0118 = 1.18\% < 2\%$$

Assim, é necessário que a fila de espera tenha capacidade para  $X - 1 = 2$  pacotes.

## Exercício 3

Considere uma ligação ponto-a-ponto de 64 kbps partilhada por 3 fluxos de pacotes:

- o fluxo A gera pacotes de tamanho constante de 200 bytes a uma taxa de Poisson de 10 pacotes/s.
- o fluxo B gera pacotes de tamanho exponencialmente distribuído com média de 500 bytes a uma taxa de Poisson de 2 pacotes/s.
- o fluxo C gera pacotes de tamanho exponencialmente distribuído com média de 500 bytes a uma taxa de Poisson de 6 pacotes/s.

A fila de espera tem tamanho infinito e atribui maior prioridade ao fluxo A e menor prioridade aos fluxos B e C (segundo um mecanismo de prioritização estrita não-preemptiva). Determine o atraso médio dos pacotes para cada um dos 3 fluxos.

## Exercício 3 - resolução

Os fluxos B e C têm a mesma prioridade e o mesmo tamanho médio de pacotes pelo que são considerados um único fluxo BC com taxa de chegada de pacotes igual à soma das taxas de chegada dos dois fluxos ( $2 + 6 = 8$  pacotes/s).

$$W_{Q_k} = \frac{\sum_{i=1}^n \lambda_i E(S_i^2)}{2(1-\rho_1 - \dots - \rho_{k-1})(1-\rho_1 - \dots - \rho_k)} \quad \text{onde} \quad \rho_k = \lambda_k / \mu_k$$

$$\lambda_A = 10 \text{ pac/s} \quad \mu_A = 64000/(200 \times 8) = 40 \text{ pac/s} \quad \rho_A = \lambda_A / \mu_A = 0.25$$

$$\lambda_{BC} = 2+6 = 8 \text{ pac/s} \quad \mu_{BC} = 64000/(500 \times 8) = 16 \text{ pac/s} \quad \rho_{BC} = \lambda_{BC} / \mu_{BC} = 0.5$$

$$E(S_A^2) = VAR(S_A) + E(S_A)^2 = 0^2 + (1/\mu_A)^2 = 625 \times 10^{-6}$$

$$E(S_{BC}^2) = VAR(S_{BC}) + E(S_{BC})^2 = (1/\mu_{BC})^2 + (1/\mu_{BC})^2 = 2 \times (1/\mu_{BC})^2 = 7812.5 \times 10^{-6}$$

$$W_A = \frac{\lambda_A E(S_A^2) + \lambda_{BC} E(S_{BC}^2)}{2(1-\rho_A)} + \frac{1}{\mu_A} = 70.8 \text{ ms}$$

$$W_B = W_C = W_{BC} = \frac{\lambda_A E(S_A^2) + \lambda_{BC} E(S_{BC}^2)}{2(1-\rho_A)(1-\rho_A - \rho_{BC})} + \frac{1}{\mu_{BC}} = 245.8 \text{ ms}$$

## Exercício 4

Considere uma ligação ponto-a-ponto de 128 Kbps. Chegam a esta ligação dois fluxos de chamadas de Poisson ambos com taxas de 2 chamadas/hora. Cada chamada ocupa uma capacidade de 64 Kbps no fluxo 1 e 128 Kbps no fluxo 2. As chamadas têm duração exponencialmente distribuída com média de 3 minutos no fluxo 1 e 6 minutos no fluxo 2.

Determine:

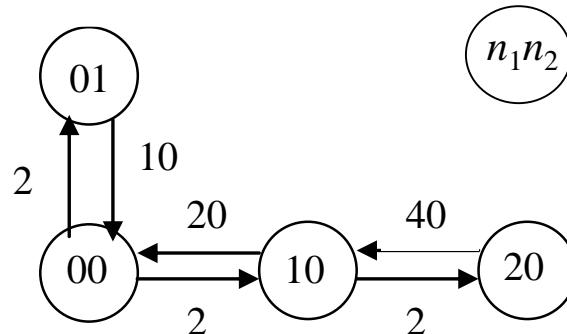
- (a) a cadeia de Markov que representa o estado da ligação,
- (b) a probabilidade de bloqueio de cada fluxo,
- (c) a taxa de ocupação da ligação.

## Exercício 4 - resolução

(a)  $\lambda_1 = \lambda_2 = 2$  chamadas/hora

$$\mu_1 = 60/3 = 20 \text{ chamadas/hora}$$

$$\mu_2 = 60/6 = 10 \text{ chamadas/hora}$$



(b)  $\rho_1 = \lambda_1 / \mu_1 = 2/20 = 0.1$  Erl.

$$\rho_2 = \lambda_2 / \mu_2 = 2/10 = 0.2$$
 Erl.

$$B_K = 1 - \frac{\sum_{\mathbf{n} \in S_k} \prod_{l=1}^K \frac{\rho_l^{n_l}}{n_l!}}{\sum_{\mathbf{n} \in S} \prod_{l=1}^K \frac{\rho_l^{n_l}}{n_l!}}$$

$$B_1 = 1 - \frac{\frac{0.1^0 0.2^0}{0! 0!} + \frac{0.1^1 0.2^0}{1! 0!}}{\frac{0.1^0 0.2^0}{0! 0!} + \frac{0.1^1 0.2^0}{1! 0!} + \frac{0.1^2 0.2^0}{2! 0!} + \frac{0.1^0 0.2^1}{0! 1!}} = 15.7\%$$

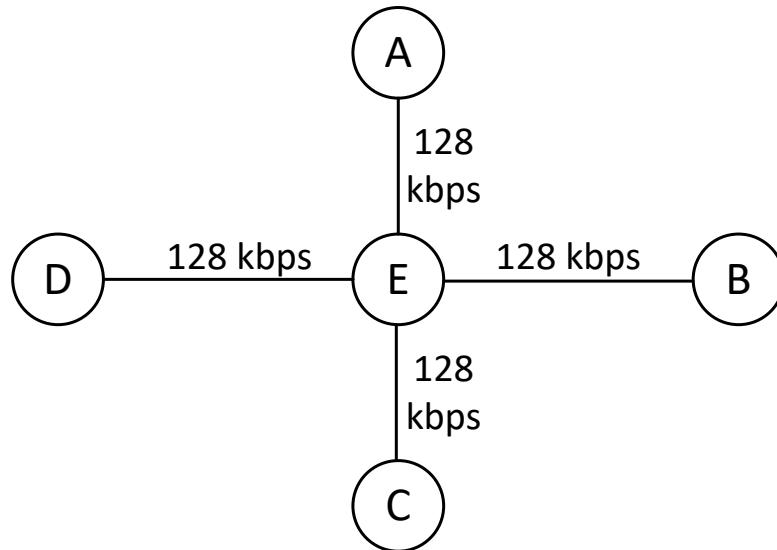
$$B_2 = 1 - \frac{\frac{0.1^0 0.2^0}{0! 0!}}{\frac{0.1^0 0.2^0}{0! 0!} + \frac{0.1^1 0.2^0}{1! 0!} + \frac{0.1^2 0.2^0}{2! 0!} + \frac{0.1^0 0.2^1}{0! 1!}} = 23.4\%$$

(c)

$$0 \times P(00) + 0.5 \times P(10) + 1 \times P(20) + 1 \times P(01) = \frac{0 \times \frac{0.1^0 0.2^0}{0! 0!} + 0.5 \times \frac{0.1^1 0.2^0}{1! 0!} + 1 \times \frac{0.1^2 0.2^0}{2! 0!} + 1 \times \frac{0.1^0 0.2^1}{0! 1!}}{\frac{0.1^0 0.2^0}{0! 0!} + \frac{0.1^1 0.2^0}{1! 0!} + \frac{0.1^2 0.2^0}{2! 0!} + \frac{0.1^0 0.2^1}{0! 1!}} = 19.5\%$$

## Exercício 5

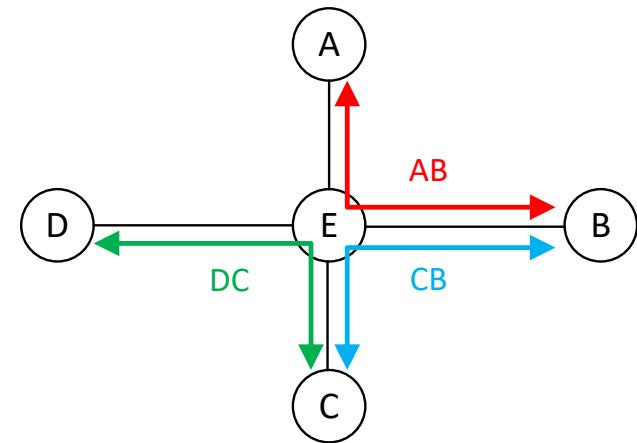
Considere a rede com comutação de circuitos da figura seguinte. A rede suporta três sessões de chamadas: AB, CB e DC. As chamadas das três sessões chegam de acordo com um processo de Poisson com taxas  $\lambda_{AB} = 2$  chamadas/hora,  $\lambda_{CB} = 1$  chamada/hora e  $\lambda_{DC} = 5$  chamadas/hora. Em todas as sessões, as chamadas requerem uma largura de banda de 64 Kbps e a duração de cada chamada é exponencialmente distribuída com média de 3 minutos. Determine os limites superiores para a probabilidade de bloqueio de cada sessão com base no teorema do limite do produto.



# Exercício 5 - resolução

Teorema do Limite do Produto:

$$\bar{\rho}_j = \sum_{k \in K_j} \rho_k \quad B_k \leq 1 - \prod_{j \in R_k} (1 - ER[\bar{\rho}_j, C_j])$$



$$\rho_{AB} = \frac{2 \times 3}{60} = 0.1 \text{ Erlangs} \quad \rho_{CB} = \frac{1 \times 3}{60} = 0.05 \text{ Erlangs} \quad \rho_{DC} = \frac{5 \times 3}{60} = 0.25 \text{ Erlangs}$$

$$B_{AB} \leq 1 - (1 - ER[\rho_{AB}, 2])(1 - ER[\rho_{AB} + \rho_{CB}, 2])$$

$$B_{AB} \leq 1 - \left( 1 - \frac{\frac{0.1^2}{2!}}{\frac{0.1^0}{0!} + \frac{0.1^1}{1!} + \frac{0.1^2}{2!}} \right) \left( 1 - \frac{\frac{0.15^2}{2!}}{\frac{0.15^0}{0!} + \frac{0.15^1}{1!} + \frac{0.15^2}{2!}} \right) = 0.0142 = 1.42\%$$

$$B_{CB} \leq 1 - (1 - ER[\rho_{CB} + \rho_{DC}, 2])(1 - ER[\rho_{AB} + \rho_{CB}, 2])$$

$$B_{CB} \leq 1 - \left( 1 - \frac{\frac{0.3^2}{2!}}{\frac{0.3^0}{0!} + \frac{0.3^1}{1!} + \frac{0.3^2}{2!}} \right) \left( 1 - \frac{\frac{0.15^2}{2!}}{\frac{0.15^0}{0!} + \frac{0.15^1}{1!} + \frac{0.15^2}{2!}} \right) = 0.0428 = 4.28\%$$

$$B_{DC} \leq 1 - (1 - ER[\rho_{DC}, 2])(1 - ER[\rho_{CB} + \rho_{DC}, 2])$$

$$B_{DC} \leq 1 - \left( 1 - \frac{\frac{0.25^2}{2!}}{\frac{0.25^0}{0!} + \frac{0.25^1}{1!} + \frac{0.25^2}{2!}} \right) \left( 1 - \frac{\frac{0.3^2}{2!}}{\frac{0.3^0}{0!} + \frac{0.3^1}{1!} + \frac{0.3^2}{2!}} \right) = 0.0570 = 5.70\% \quad 12$$

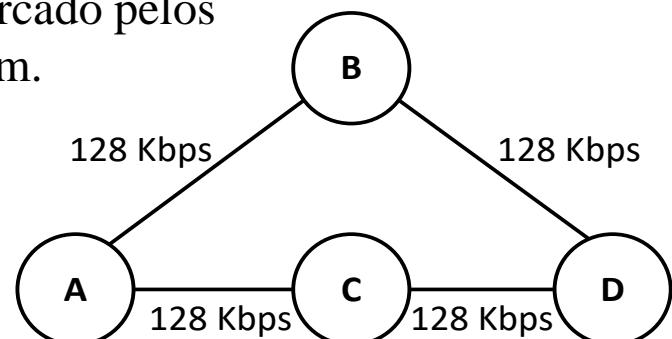
## Exercício 6

Considere a rede com comutação de pacotes da figura. Inicialmente, a rede serve um fluxo de 24 pacotes/s no percurso direto A→B e um de fluxo 3 pacotes/s no percurso direto A→C. É oferecido um novo fluxo de 10 pacotes/s de A para D. Todos os fluxos são caracterizados por intervalos entre chegadas e comprimentos de pacotes independentes e exponencialmente distribuídos. O comprimento médio dos pacotes é de 250 bytes.

(a) Admita que o novo fluxo A→D é encaminhado em igual percentagem pelos dois percursos possíveis. Determine pela aproximação de Kleinrock o atraso médio que os pacotes de cada fluxo sofrem na rede.

(b) Admita que o novo fluxo A→D pode ser bifurcado pelos dois percursos possíveis em qualquer percentagem.

Determine o sistema de equações cuja resolução permite calcular a bifurcação do novo fluxo que minimiza o atraso médio total da rede.



## Exercício 6 - resolução

(a) O atraso médio do fluxo  $s$  é dado por:

$$W_s = \sum_{(i,j) \in R_s} \left( \frac{1}{\mu_{ij} - \lambda_{ij}} + d_{ij} \right)$$

$x_1 = 5$  pps,  $x_2 = 5$  pps,  $d_{ij}$  nulos. Em todas as ligações temos  $\mu = \frac{128000}{250 \times 8} = 64$  pps

$$W_{AB} = \frac{1}{64 - (24 + 5)} = 28.6 \text{ ms}$$

$$W_{AC} = \frac{1}{64 - (3 + 5)} = 17.9 \text{ ms}$$

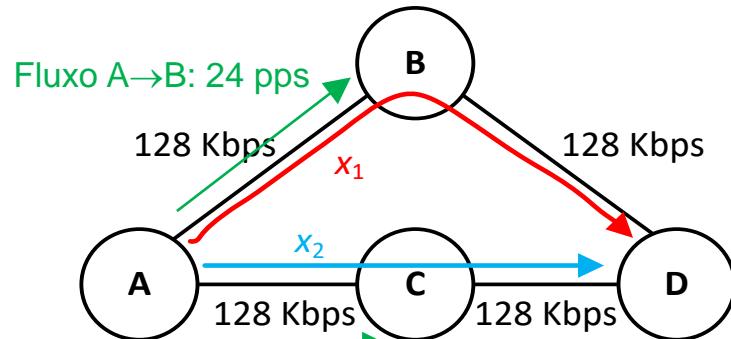
$$W_{AD} = 0.5 \times \left( \frac{1}{64 - (24 + 5)} + \frac{1}{64 - 5} \right) + 0.5 \times \left( \frac{1}{64 - (3 + 5)} + \frac{1}{64 - 5} \right) = 40.2 \text{ ms}$$

$$(b) L = L_{AB} + L_{AC} + L_{BD} + L_{CD} = \frac{x_1 + 24}{64 - (x_1 + 24)} + \frac{x_2 + 3}{64 - (x_2 + 3)} + \frac{x_1}{64 - x_1} + \frac{x_2}{64 - x_2}$$

$$\frac{\partial L}{\partial x_1} = \frac{64}{(40 - x_1)^2} + \frac{64}{(64 - x_1)^2}$$

$$\frac{\partial L}{\partial x_2} = \frac{64}{(61 - x_2)^2} + \frac{64}{(64 - x_2)^2}$$

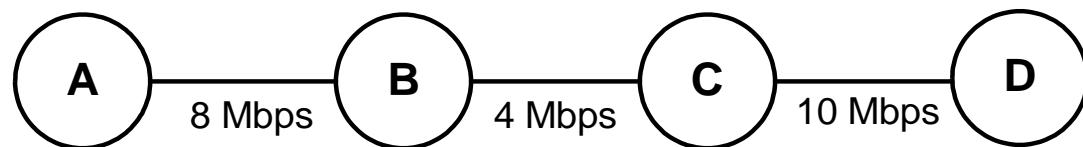
Regra das derivadas:  $\left(\frac{u}{v}\right)' = \frac{u'v - uv'}{v^2}$



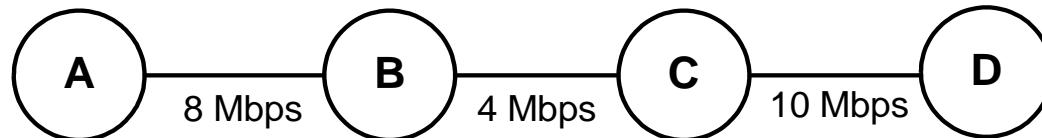
$$\begin{cases} \frac{64}{(40 - x_1)^2} + \frac{64}{(64 - x_1)^2} = \frac{64}{(61 - x_2)^2} + \frac{64}{(64 - x_2)^2} \\ x_1 + x_2 = 10 \end{cases}$$

## Exercício 7

Considere a rede com comutação de pacotes da figura. Em todas as ligações, o atraso de propagação é de 10 milissegundos em cada sentido. A rede suporta fluxos entre todos os nós com pacotes de tamanho médio 1000 bytes. O fluxo de A para C é controlado por janelas extremo-a-extremo em que as permissões têm um tamanho fixo de 40 Bytes. Determine o tamanho mínimo (em número de pacotes) da janela de emissão garantindo que este fluxo pode emitir ao ritmo máximo quando nenhum outro fluxo está ativo.



## Exercício 7 - resolução



$$W_{AC} \geq \left\lceil \frac{d}{X} \right\rceil$$

$d$  – atraso de ida (de um pacote) e volta (da permissão)

$X$  – tempo médio de transmissão de cada pacote (ao ritmo permitido pela rede)

$$d = \frac{8 \times 1000}{8000000} + 0.01 + \frac{8 \times 1000}{4000000} + 0.01 + \frac{8 \times 40}{4000000} + 0.01 + \frac{8 \times 40}{8000000} + 0.01$$

$$X = \frac{8 \times 1000}{4000000}$$

$$W_{AC} \geq \left\lceil \frac{d}{X} \right\rceil = \left\lceil \frac{0.04312}{0.002} \right\rceil = [21.56] = 22$$

O tamanho mínimo da janela de emissão é 22 pacotes.

## Exercício 8

Considere uma ligação de 2 Mbps que serve três fluxos de pacotes (A, B e C) com o algoritmo de escalonamento *Deficit Round Robin* e um limiar de 1200 bytes para cada fluxo.

- No fluxo A, chegam 3 pacotes: pacote A.1 de 800 Bytes no instante 0 ms, pacote A.2 de 700 Bytes no instante 12 ms e pacote A.3 de 200 Bytes no instante 15 ms.
- No fluxo B, chegam 2 pacotes: pacote B.1 de 500 Bytes no instante 5 ms e pacote B.2 de 600 Bytes no instante 6 ms.
- No fluxo C, chegam 2 pacotes: pacote C.1 de 1000 Bytes no instante 0 ms e pacote C.2 de 1300 Bytes no instante 4 ms.

O ciclo segue a sequência A → B → C e o algoritmo decide no início de cada ciclo os pacotes a enviar e respetiva ordem. Determine que pacotes e por que ordem são enviados em cada ciclo.

## **Exercício 8 - resolução**

- Fluxo A:
    - pacote A.1 de 800 Bytes no instante 0 ms
    - pacote A.2 de 700 Bytes no instante 12 ms
    - pacote A.3 de 200 Bytes no instante 15 ms
  - Fluxo B:
    - pacote B.1 de 500 Bytes no instante 5 ms
    - pacote B.2 de 600 Bytes no instante 6 ms
  - Fluxo C:
    - pacote C.1 de 1000 Bytes no instante 0 ms
    - pacote C.2 de 1300 Bytes no instante 4 ms

*1º ciclo:* Início: 0 ms

$$2^{\text{o}} \text{ ciclo: Início: } 0 + 8 \times (800+1000)/2000000 = 7.2 \times 10^{-3} = 7.2 \text{ ms}$$

Pacotes: B.1 → B.2 → C.2 Créditos: Fluxo A = 0 Bytes  
Fluxo B =  $1200 - (500+600) = 100$  Bytes  
Fluxo C =  $(200+1200) - 1300 = 100$  Bytes

$$3^{\text{o}} \text{ ciclo: Início: } 7.2 \times 10^{-3} + 8 \times (500+600+1300)/2000000 = 16.8 \times 10^{-3} = 16.8 \text{ ms}$$