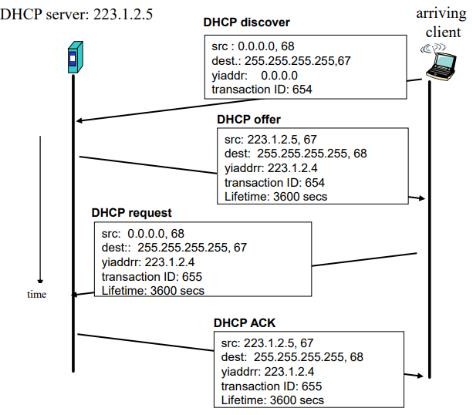
Quando uma trama é recebida por um Switch Ethernet e a tabela de encaminhamento do Switch não contém uma entrada para o endereço de destino da trama, o Switch envia a trama para todas as portas ativas exceto a porta através da qual a trama foi recebida.



Bit stuffing – 15 -> 150

Byte stuffing – FLAG -> ESC FLAG.

Bit & Byte stuffing – 0160 -> 01501 0120130

Assuma uma fila de espera M/M/1 estável, caracterizada por uma chegada de C pac/s e um serviço de P pac/s. Nesta fila, o débito de partida de clientes da fila (pac/s) é igual a:

a) C/P. b) P.

**c) C.** d) P/C.

Masks - nr hosts (2nr bytes host -2)

/27 – 30 /28 - 14

/29 – 6 /30 - 2

log\_k (x)

log\_b (x) =

log\_k (b)

log\_b (x)

log\_2 (x) =

log\_b (2)

Extra:

|  |  |
| --- | --- |
| **M/M/1 (Cadeias de Markov)**  Chegadas -> Poisson; Attend. -> Exponencial Prob. De estar em estado n: 𝑃(𝑛) = 𝜌𝑛(1 − 𝜌)  𝜆  Tamanho médio de uma fila: 𝑁 =  𝜇−𝜆  Número médio clientes em espera: 𝑁𝑤 = 𝑁 − 𝜌  𝐸[𝑋] = 1 ; 𝐸[𝑋2] = 2 𝑇 = 1  𝜇 𝜇2 𝜇−𝜆  𝑇 = 𝑇 − 𝑇 = 1 − 1 = 𝜆 = 𝜌  𝑤 𝑠 𝜇−𝜆 𝜇 𝜇2(1−𝜌) 𝜇(1−𝜌) | **M/G/1**  Chegadas->Poisson; Attend -> Arbitrário  𝜆𝐸[𝑋2]  Tempo de espera médio: 𝑇𝑤 = 2(1−𝜌)  1  𝑁 = 𝜆𝑇 = 𝜆 ൬𝑇𝑤 + 𝜇൰ = 𝑁𝑤 + 𝜌 |
| **Notação de Kendal** (A/S/s/K)  A->Processo estatístico da chegada de clientes S->Processo estatístico do serviço  s->número de servidores  K->capacidade do sistema em buffers |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Physical Layer: |  | | | | | | |
| **M-PAM**(amplitude)  𝑠(𝑡) = 𝐴𝑖 COS(2𝜋𝑓𝑐𝑡)  𝑓𝑎𝑠𝑒(𝜃) = 0 (𝑧𝑒𝑟𝑜) | |  | **M-PSK**(fase)  𝑠(𝑡) = 𝐴𝑐𝑜𝑠(𝜃𝑖 + 2𝜋𝑓𝑐𝑡)  𝐴 = 𝑐𝑜𝑛𝑠𝑡𝑎𝑛𝑡𝑒 | | **M-QAM**(ampli. e fase)  𝑠(𝑡) = 𝐴𝑖cos (𝜃𝑖 + 2𝜋𝑓𝑐𝑡) | | **M níveis**  𝐶 = 2𝐵𝑙𝑜𝑔2(𝑀) (𝐶 = 2𝐵 [2 𝑛í𝑣𝑒𝑖𝑠])  𝑠𝑦𝑚𝑏𝑜𝑙𝑠  𝐵𝑎𝑢𝑑𝑟𝑎𝑡𝑒 = 2𝐵 ൬ 𝑜𝑢 𝑏𝑎𝑢𝑑൰  𝑠  𝐵𝑖𝑡𝑟𝑎𝑡𝑒 = 2𝐵𝑙𝑜𝑔2(𝑀) = 𝐶 |
| **Lei de Shannon** (capacidade max)  𝐶 = 𝐵𝑐 log2(1 + 𝑆𝑁𝑅)  𝑃𝑟  𝑆𝑁𝑅 → 𝑆𝑖𝑔𝑛𝑎𝑙 𝑁𝑜𝑖𝑠𝑒 𝑅𝑎𝑡𝑖𝑜 =  𝑁0𝐵𝑐  𝐵𝑐 → 𝐹𝑟𝑒𝑞𝑢ê𝑛𝑐𝑖𝑎 𝑑𝑜 𝑐𝑎𝑛𝑎𝑙 (𝐻𝑧)  𝑃𝑟 → 𝑃𝑜𝑡ê𝑛𝑐𝑖𝑎 𝑅𝑒𝑐𝑒𝑏𝑖𝑑𝑎 (𝑊)  𝑊  𝑁0 → 𝑅𝑢𝑖𝑑𝑜 𝐵𝑟𝑎𝑛𝑐𝑜 ൬10−9 ൰  𝐻𝑧  𝑁0𝐵𝑐 → 𝑃𝑜𝑡ê𝑛𝑐𝑖𝑎 𝑑𝑜 𝑟𝑢í𝑑𝑜  𝑟𝑒𝑐𝑒𝑏𝑖𝑑𝑜 𝑛𝑎 𝑓𝑟𝑒𝑞𝑢ê𝑛𝑐𝑖𝑎 𝐵 (𝑊) | | | | **Cabos**  𝑃𝑟 = 𝑃𝑡 ∙ 𝐺𝑎𝑛ℎ𝑜 (𝑒𝑚 𝑊)  𝑃𝑟 = 𝑃𝑡 + 𝐺𝑎𝑛ℎ𝑜 (𝑒𝑚 𝑑𝐵)  1  𝐺𝑎𝑛ℎ𝑜 = (𝑒𝑚 𝑊)  𝐴𝑡𝑒𝑛𝑢𝑎çã𝑜  𝐺𝑎𝑛ℎ𝑜 = −𝐴𝑡𝑒𝑛𝑢𝑎çã𝑜(𝑒𝑚 𝑑𝐵) | |  |
| **Free Space Loss**  𝑃𝑡 = (4𝜋𝑑)2 = (4𝜋𝑓𝑑)2 𝜆𝑓 = 𝑐  𝑃𝑟 𝜆2 𝑐2  𝑃𝑡 → 𝑠𝑖𝑔𝑛𝑎𝑙 𝑝𝑜𝑤𝑒𝑟 𝑎𝑡 𝑡𝑟𝑎𝑛𝑠𝑚𝑖𝑡𝑡𝑖𝑛𝑔 𝑎𝑛𝑡𝑒𝑛𝑛𝑎  𝑃𝑟 → 𝑠𝑖𝑔𝑛𝑎𝑙 𝑝𝑜𝑤𝑒𝑟 𝑎𝑡 𝑟𝑒𝑐𝑒𝑖𝑣𝑖𝑛𝑔 𝑎𝑛𝑡𝑒𝑛𝑛𝑎  𝜆 → 𝑐𝑎𝑟𝑟𝑖𝑒𝑟 𝑤𝑎𝑣𝑒𝑙𝑒𝑛𝑔𝑡ℎ  𝑑 → 𝑝𝑟𝑜𝑝𝑎𝑔𝑎𝑡𝑖𝑜𝑛 𝑑𝑖𝑠𝑡𝑎𝑛𝑐𝑒 𝑏𝑒𝑡𝑤𝑒𝑒𝑛 𝑎𝑛𝑡𝑒𝑛𝑎𝑠  𝑚  𝑐 → 𝑠𝑝𝑒𝑒𝑑 𝑜𝑓 𝑙𝑖𝑔ℎ𝑡(3 ∗ 108 )  𝑠 | |
| 𝑃𝑑𝐵𝑊 = 10 log10 𝑃  𝑃  𝑃𝑑𝐵𝑚 = 10 log10 ൬ ൰ 1𝑚𝑊 | |

# Switch - Frame forwarding/flooding

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MAC: |  | | | | |
| **Random Access** (pouco eficiente em canais | | **CSMA**  Se canal livre transmite, senão espera. Há col.  𝑇𝑝𝑟𝑜𝑝  𝑎 = 𝑃𝑐𝑜𝑙 = 𝑇 ≪ 1  𝑓𝑟𝑎𝑚𝑒  -Persistent: ocup-espera que fique livre  -Non-persistent:ocup-espera tempo aleatório  -p-persistent: espera até ter um slot livre para enviar  𝑣𝑢𝑙𝑛. 𝑡𝑖𝑚𝑒 = 2𝑇𝑝𝑟𝑜𝑝 | **CSMA/CD**  Parecido a p- persistent Ouve enquanto envia, colisão-  >aborta->binary exponencial back- off->retransmite só funciona se  𝑇𝑓 > 2𝑇𝑝𝑟𝑜𝑝, senão não deteta colisão | **CSMA/CA**  Monitoriza canal até estar livre durante um período maior que DIFS (Distributed Inter-Frame Space) e transmite. Se estiver ocupado, define tempo de back off aleatório que vai diminuindo.  Necessita de ACK | **Channel Partitioning** (pouco eficiente em canais pouco carregados)  -Time Div. Multi.; -Freq. Div. Multi. |
| **Aloha**  Transmite, espera por round-trip propagation delay, recebe ack e envia ou, atrasa e envia   * Pure Aloha * Slotted Aloha | | **Taking Turns** – Cada estação com o seu turno.  Estações com mais info -> turnos maiores.  -**Polling**-estação mestre.  Prob: Overhead; latência; ponto de falha  -**Passagem de tokens**- passsam tokens entre si para saber quem transmite. Prob: same as above |

When Switch receives a frame: 1. record link associated with sending host 2. index forwarding table using MAC destination address 3.

if (entry found in table) {

if (destination is on segment from which frame arrived) drop the frame

else

forward the frame on interface indicated

O Algoritmo Spanning Tree:

* Permite obter um caminho único entre nós Ethernet.
* Permite que uma única árvore seja calculada na rede, com raiz no nó com menor identificador.

Services provided by network layer

» Datagram network -> connectionless service (IP is datagram)

» Virtual Circuit network -> connection oriented service

} else

**Stop & Wait ARQ**

1

𝑁𝑟 = 1 − 𝑃

𝑒

1 − 𝑃𝑒

𝑆 =

1 + 2𝑎

𝑇𝑒𝑠𝑡 → 𝑇𝑒𝑚𝑝𝑜 𝑑𝑒 𝑒𝑠𝑡𝑎𝑏𝑒𝑙𝑒𝑟 𝑙𝑖𝑔𝑎çã𝑜

𝑇𝑝𝑟𝑜𝑝 → 𝑇𝑒𝑚𝑝𝑜 𝑑𝑒 𝑝𝑟𝑜𝑝𝑎𝑔𝑎çã𝑜 ~0

𝐿

* Link-by-Link ARQ - Repairs losses link by link. Store packets in case they have to be retransmitted (memory required)
* End-to-End ARQ - Switches/routers become simpler. Packets may follow different end-to-end paths. (Not acceptable when Packet Loss Ratio is high)

𝑇𝑚𝑠𝑔 → 𝑇𝑒𝑚𝑝𝑜 𝑑𝑒 𝑒𝑛𝑣𝑖𝑎𝑟 𝑑𝑎𝑑𝑜𝑠 ൬ ൰

𝑅

𝑇𝑖 → 𝑇𝑒𝑚𝑝𝑜 𝑑𝑒 𝑒𝑛𝑣𝑖𝑎𝑟 𝑝𝑎𝑐𝑜𝑡𝑒 𝑛𝑢𝑚𝑎 𝑙𝑖𝑔𝑎çã𝑜 𝑖 ൬ ൰

𝐿

𝑅

𝑇𝑝𝑖 → 𝑇𝑒𝑚𝑝𝑜 𝑑𝑒 𝑝𝑟𝑜𝑝𝑎𝑔𝑎çã𝑜 𝑛𝑢𝑚𝑎 𝑙𝑖𝑔𝑎çã𝑜 𝑖

Packet Switching

𝑇𝑝𝑎𝑐 = 𝑆𝑢𝑚(𝑇𝑖)

𝑇𝑖 = 𝑇𝑝𝑖 + 𝑇𝑚𝑠𝑔𝑖

Circuit Switching

𝑇𝑡𝑜𝑡 = 𝑇𝑒𝑠𝑡 + 𝑇𝑝𝑟𝑜𝑝 + 𝑇𝑚𝑠𝑔

Intro:

Nota: No caso de 𝑃𝑜 = 0, as fórmulas de S são as de “Stop & Wait” e “Sliding Window”

Stop & Wait: quando a << 1 (eficiente apenas em distâncias curtas)

Selective Reject: quando a >> 1 (grandes distâncias, evitar retransmissões em caso de erro)

= 2𝑘−1

𝑀

2

𝑊𝑚𝑎𝑥 =

𝑊 ∙ (1 − 𝑃𝑒)

1 + 2𝑎

𝑆 =

𝑊 = 2𝑘−1

𝑆 = 1 − 𝑃𝑒

Se 𝑊 < 1 + 2𝑎:

**Selective Reject/Repeat ARQ** (janelas grandes)

Se 𝑊 ≥ 1 + 2𝑎: tamanho máximo:

𝐺(𝑥)

𝑅(𝑥) → 𝑟𝑒𝑠𝑡𝑜 (𝑀(𝑥) ∙ )

𝑀(𝑥) → 𝑑𝑎𝑑𝑜𝑠

𝑇(𝑥) → 𝑖𝑛𝑓𝑜𝑟𝑚𝑎çã𝑜 𝑓𝑖𝑛𝑎𝑙 = 𝑀(𝑥) ∙ 𝑥𝑟 + 𝑅(𝑥)

𝐺(𝑥) → 𝑥𝑟 + ⋯ + 1

𝑥𝑟

**CRC**

𝑑

𝑇𝑝 = 𝑑 ∗ 𝜏𝑎 = 𝑉

𝑒 𝑒

𝑊𝑚𝑎𝑥 = 𝑀 − 1 = 2𝑘 − 1

K is number of bits used to code sequence numbers

𝑊(1 − 𝑃𝑒)

𝑆 = (1 + 2𝑎) ∙ (1 − 𝑃 + 𝑊 ∙ 𝑃 )

𝑊 = 2𝑘 − 1

Se W < 1+2a :

1+2𝑎∗𝑃𝑒

𝑆 = 1−𝑃𝑒

**Go-Back-N ARQ** (janelas pequenas)

𝑆𝑒 𝑊 ≥ 1 + 2𝑎: tamanho máximo:

𝐿

𝑇𝑓 = 𝑅

𝑇𝑓

𝑇𝑝

𝑎 =

𝐹𝐸𝑅 = 1 − (1 − 𝐵𝐸𝑅)𝐿

𝑖

𝑃 = (𝑛) 𝑝𝑖(1 − 𝑝)𝑛−𝑖

No Error Probability:

𝑃 = (1 − 𝑝)𝑛

Error Probability:

𝑃 = 1 − (1 − 𝑝)𝑛

𝑖 Error Probability:

Débito máximo: 𝑅𝑀𝐴𝑋 = 𝑆 ∗ (𝑅 [𝑜𝑢 𝐶])

1 + 2𝑎

𝑈 =

**Sliding Window**

𝑆𝑒 𝑊 ≥ 1 + 2𝑎:

𝑈 = 1

𝑆𝑒 𝑊 < 1 + 2𝑎:

𝑊

𝑇𝑓 → 𝑡𝑒𝑚𝑝𝑜 𝑑𝑒 𝑡𝑟𝑎𝑛𝑠𝑚𝑖𝑠𝑠ã𝑜(𝑚𝑠)

𝐿 → 𝑡𝑎𝑚𝑎𝑛ℎ𝑜 𝑑𝑎 𝑡𝑟𝑎𝑚𝑎(𝑏𝑖𝑡𝑠)

𝑇𝑃 → 𝑡𝑒𝑚𝑝𝑜 𝑑𝑒 𝑝𝑟𝑜𝑝𝑎𝑔𝑎çã𝑜(𝑚𝑠)

𝑑 → 𝑑𝑖𝑠𝑡â𝑛𝑐𝑖𝑎(𝑘𝑚)

𝜏𝑎 → 𝑎𝑡𝑟𝑎𝑠𝑜 𝑑𝑒 𝑝𝑟𝑜𝑝𝑎𝑔𝑎çã𝑜(𝜇𝑠/𝑘𝑚)

𝑆 → 𝑒𝑓𝑖𝑐𝑖ê𝑛𝑐𝑖𝑎

𝑁𝑟 → 𝑛º 𝑚é𝑑𝑖𝑜 𝑡𝑒𝑛𝑡. 𝑝. 𝑡𝑟𝑎𝑛𝑠𝑚𝑖𝑡𝑖𝑟 𝑡𝑟𝑎𝑚𝑎 𝑐. 𝑠𝑢𝑐𝑒𝑠𝑠𝑜

𝑃𝑒 (𝐹𝐸𝑅) → 𝑝𝑟𝑜𝑏. 𝑑𝑒 𝑡𝑟𝑎𝑛𝑠𝑚𝑖𝑠𝑠ã𝑜 𝑑𝑒 𝑡𝑟𝑎𝑚𝑎 𝑐𝑜𝑚 𝑒𝑟𝑟𝑜𝑠

𝑊 → 𝑡𝑎𝑚𝑎𝑛ℎ𝑜 𝑑𝑎 𝑗𝑎𝑛𝑒𝑙𝑎

𝑀 → 𝑅𝑒𝑝𝑟𝑒𝑠𝑒𝑛𝑡𝑎çã𝑜 𝑒𝑚 𝑚ó𝑑. 𝑑𝑒 𝑛º 𝑑𝑒 𝑠𝑒𝑞.

𝑘 → 𝑛º𝑏𝑖𝑡𝑠 𝑛𝑒𝑐𝑒𝑠𝑠á𝑟𝑖𝑜𝑠 𝑝𝑎𝑟𝑎 𝑐𝑜𝑑𝑖𝑓𝑖𝑐𝑎𝑟 𝑊 𝑡𝑟𝑎𝑚𝑎𝑠

𝑑 → 𝑛º min 𝑑𝑒 𝑒𝑟𝑟𝑜𝑠 𝑛𝑒𝑐. 𝑝𝑎𝑟𝑎 𝑒𝑟𝑟𝑜 ñ 𝑠𝑒𝑟 𝑑𝑒𝑡𝑒𝑡𝑎𝑑𝑜

𝑅 → 𝑑𝑎𝑡𝑎 𝑟𝑎𝑡𝑒 (𝑏𝑖𝑡𝑠 / 𝑠)

Parity Check -> d=2

Bi-dimensional Parity -> d=4 Internet Checksum -> d=2

Cyclic Redundacy Check (CRC) -> d>3

Protocolos de Ligação de Dados:

𝜌

𝑁 = 𝑁𝑤 + 𝑁𝑠 = 𝜆 ∙ 𝑇𝑎 = 1 − 𝜌

**Time. Div. Multi.**

𝐶 = 𝐶 𝑇 = 𝐿 ∙ 𝑚

𝑐 𝑝

𝑚

𝐶

𝐶 → 𝑐𝑎𝑝𝑎𝑐𝑖𝑑𝑎𝑑𝑒 𝑑𝑜 𝑐𝑎𝑛𝑎𝑙 (𝑘𝑏𝑖𝑡𝑠/𝑠)

𝐿 → 𝑡𝑎𝑚𝑎𝑛ℎ𝑜 𝑑𝑜 𝑝𝑎𝑐𝑜𝑡𝑒 (𝑏𝑖𝑡𝑠)

𝑅 → 𝑡𝑟á𝑓𝑒𝑔𝑜 𝑚é𝑑𝑖𝑜 (𝑘𝑏𝑖𝑡/𝑠𝑒𝑔)

𝜆 → 𝑡𝑎𝑥𝑎 𝑑𝑒 𝑐ℎ𝑒𝑔𝑎𝑑𝑎𝑠 (𝑝𝑎𝑐𝑜𝑡𝑒𝑠/𝑠𝑒𝑔)

𝜇 → 𝑡𝑎𝑥𝑎 𝑑𝑒 𝑒𝑛𝑣𝑖𝑜𝑠 (𝑝𝑎𝑐𝑜𝑡𝑒𝑠/𝑠𝑒𝑔)

𝜌 → 𝑖𝑛𝑡𝑒𝑛𝑠𝑖𝑑𝑎𝑑𝑒 𝑚é𝑑𝑖𝑎 𝑑𝑒 𝑡𝑟á𝑓𝑒𝑔𝑜(𝑡𝑎𝑥𝑎 𝑑𝑒 𝑢𝑡𝑖𝑙𝑖𝑧𝑎çã𝑜)

𝑇𝑎 → 𝑡𝑒𝑚𝑝𝑜 𝑚é𝑑𝑖𝑜 𝑑𝑒 𝑎𝑡𝑟𝑎𝑠𝑜 𝑑𝑜𝑠 𝑝𝑎𝑐𝑜𝑡𝑒𝑠 (𝑚𝑠)

𝑇𝑊 → 𝑡𝑒𝑚𝑝𝑜 𝑚é𝑑𝑖𝑜 𝑑𝑒 𝑒𝑠𝑝𝑒𝑟𝑎 𝑛𝑎 𝑓𝑖𝑙𝑎

𝑇𝑠 → 𝑡𝑒𝑚𝑝𝑜 𝑚é𝑑𝑖𝑜 𝑑𝑒 𝑠𝑒𝑟𝑣𝑖ç𝑜

𝑁 → 𝑛º 𝑑𝑒 𝑐𝑙𝑖𝑒𝑛𝑡𝑒𝑠 𝑛𝑜 𝑠𝑖𝑠𝑡𝑒𝑚𝑎

𝑁𝑠 → 𝑛º 𝑑𝑒 𝑐𝑙𝑖𝑒𝑛𝑡𝑒𝑠 𝑎 𝑠𝑒𝑟𝑒𝑚 𝑠𝑒𝑟𝑣𝑖𝑑𝑜𝑠

𝑁𝑤 → 𝑜𝑐𝑢𝑝𝑎çã𝑜 𝑚é𝑑𝑖𝑎 𝑑𝑎 𝑓𝑖𝑙𝑎 𝑑𝑒 𝑒𝑠𝑝𝑒𝑟𝑎

𝑉𝑠 → 𝑝𝑎𝑐𝑜𝑡𝑒𝑠 𝑒𝑚 𝑝𝑟𝑜𝑐𝑒𝑠𝑠𝑎𝑚𝑒𝑛𝑡𝑜

𝑉𝑆 → 𝑝𝑎𝑐𝑜𝑡𝑒𝑠 𝑒𝑚 𝑒𝑠𝑝𝑒𝑟𝑎

𝑀 → 𝑛º 𝑑𝑒 𝑏𝑢𝑓𝑓𝑒𝑟𝑠

𝑃𝑛 → 𝑛º 𝑑𝑒 𝑐ℎ𝑒𝑔𝑎𝑑𝑎𝑠 𝑛𝑜 𝑖𝑛𝑡𝑒𝑟𝑣𝑎𝑙𝑜 𝑇

𝑃𝑏 → 𝑛º𝑝𝑟𝑜𝑏 𝑑𝑒 𝑏𝑙𝑜𝑞𝑢𝑒𝑖𝑜(𝑝𝑒𝑟𝑑𝑎 𝑑𝑒 𝑝𝑎𝑐𝑜𝑡𝑒𝑠) Tp → Tempo nec. para trasm. um pacote

Cc → Capacidade canal

m → uma divisão da ligação

𝐶

𝑚

𝑐 𝑝

𝐶 = 𝐶 𝑇 = 𝐿 ∙ 𝑚

**Freq. Div. Multi.**

**Statistical Multiplexing**

𝐿

𝑇𝑝 = 𝐶

(𝜆𝑇)𝑛 𝑒−𝜆𝑇

𝑃𝑛(𝑇) = 𝑛!

1

𝑃𝑏 = 𝑀 + 1 , 𝑠𝑒 𝜌 = 1

(1 − 𝜌)𝜌𝑀

𝑃𝑏 = 1 − 𝜌𝑀+1 , 𝑠𝑒 𝜌 ≠ 1

𝑁𝑤 = 𝜆𝑇𝑤 = 𝜌𝑁

**Teorema Little**

𝑁𝑠 = 𝜆𝑇𝑠 = 𝜌

1 − 𝜌

log(𝜌) , 𝑠𝑒 𝜌 ≠ 1

𝑀 =

log ( 𝑃𝑏 )

𝐶

𝜇 =

𝐿

𝜇 𝐶

𝜆 𝑅

𝜌 = =

1

𝑇𝑠 = 𝜇

𝑁

𝑇𝑤 = 𝜇

1 1 𝑁

𝑇𝑎 = 𝑇𝑤 + 𝑇𝑠 = 𝜇 − 𝜆 = 𝜇(1 − 𝜌) = 𝜆

𝑅

𝜆 =

𝐿

FILAS DE ESPERA 1:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| FILAS DE ESPERA 2: | **M/M/1/B (B buffers)**  (1−𝜌)𝜌𝐵  Probabilidade de perder dados: 𝑃(𝐵) =  1−𝜌𝐵+1  1  𝑠𝑒 𝜌 = 1 → 𝑃(𝐵) =  𝐵 + 1  𝜆 − 𝜇  𝑠𝑒 𝜌 ≫ 1 → 𝑃(𝐵) =  𝜆 |  | **M/D/1**  1 1  𝐸[𝑋] = ; 𝐸[𝑋2] = 2  𝜇 𝜇  𝜆 𝜌  𝑇𝑤 = 2𝜇2(1 − 𝜌) = 2𝜇(1 − 𝜌) |  | **D/D/1** Chegadas e atendimentos  seguem distribuição determinista |  |

flood <- (forward on all but the interface on which the frame arrived)

# Link-State Routing

* + Each router keeps track of its incident links -> link up, link down ; cost on the link
  + Each router broadcasts link state every router gets a complete view of the graph
  + Each router runs Dijkstra’s algorithm, to -> compute the shortest paths ; construct the forwarding table

# Distance Vector Algorithm

* + Iterative, asynchronous each local iteration caused by:
    - local link cost change
    - distance vector update message from neighbor
  + Distributed

» node notifies neighbors only when its DV changes

* + Neighbors then notify their neighbors, if necessary

Considere a fila de espera (de saída) da interface de rede eth0 de um computador que se encontra ligado a um *switch* por uma ligação de capacidade C bit/s. Considerando que a fila de espera do *device driver* é estável, poderemos afirmar que o tempo médio que um pacote espera nessa fila até ser transmitido depende:

**Additive Increase/Multiplicative Decrease**

* Algorithm

» increases CongestionWindow by 1 segment

* + for each RTT (Round Trip Time) -> additive increase

» divide CongestionWindow by 2

* + when there is a packet loss -> multiplicative decrease
* In practice,

» Increases by ACK received

» Increment= MSS \* (MSS / CongestionWindow)

» CongestionWindow += Increment

» MSS -> Maximum Segment Size

Transport:

𝑀𝑎𝑥𝑊𝑖𝑛 = 𝑀𝐼𝑁(𝐶𝑜𝑛𝑔𝑒𝑠𝑡𝑖𝑜𝑛𝑊𝑖𝑛𝑑𝑜𝑤, 𝐴𝑑𝑣𝑒𝑟𝑡𝑖𝑠𝑒𝑑𝑊𝑖𝑛𝑑𝑜𝑤𝑠)

𝐸𝑓𝑓𝑊𝑖𝑛 = 𝑀𝑎𝑥𝑊𝑖𝑛 − (𝐿𝑎𝑠𝑡𝐵𝑦𝑡𝑒𝑆𝑒𝑛𝑡 − 𝐿𝑎𝑠𝑡𝐵𝑦𝑡𝑒𝐴𝑐𝑘𝑒𝑑)

𝐵𝑖𝑡𝑟𝑎𝑡𝑒(𝑏𝑦𝑡𝑒/𝑠) = 𝐶𝑜𝑛𝑔𝑒𝑠𝑡𝑖𝑜𝑛𝑊𝑖𝑛𝑑𝑜𝑤𝑠/𝑅𝑇𝑇

network congestion decreases -> CongestionWindow Increases network congestion increases -> CongestionWindow decreases

## Apenas do débito a que as camadas superiores enviam pacotes para a fila de espera (pacote/s).

Uma rede composta por um conjunto de routers IP interligados entre si que transporta apenas tráfego TCP constitui

**TCP – Transmission Control Protocol**

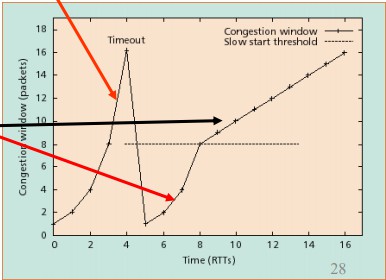
* Connection oriented
* Full-duplex
* Byte stream
* Flow control -> Reliability; ARQ mechanism; Avoids receiver’s congestion
* Congestion control -> Avoids network’s congestion

## Uma rede de comutação de pacotes e oferece um serviço não orientado às ligações.

No processo de transferência de um ficheiro usando o protocolo FTP em modo passivo, a abertura de ligação de dados é feita

## Pelo cliente para uma porta indicada pelo servidor.

Num sistema de transmissão assíncrona o recetor sincroniza-se com o emissor



\*1

\*3

\*2

* Slow Start

» Sender starts with CongestionWindow=1sgm

» Doubles CongestionWindow by RTT (\*1 on graph)

* When a segment loss is detected, by timeout

» threshold = ½ congestionWindow(\*)

» CongestionWindow=1 sgm (router gets time to empty queues)

» Lost packet is retransmitted

» Slow start while

congWindow < threshold (\*2 on graph)

» Then -> *Congestion Avoidance* phase (\*3 on graph)

## Através da pré-configuração da frequência de amostragem do sinal e utilizando “*start*” e “*stop*” bits.

O protocolo *Address Resolution Protocol*

(ARP) usa serviços oferecidos pelo protocolo

1. **IP.**

O mecanismo de controlo de congestionamento do TCP estudado nas aulas passa da fase de *Congestion Avoidance* para a fase de *Slow Start* quando:

## Deteta a perda de um segmento por timeout.

Protocolos de ligação de dados do tipo ARQ (por exemplo, *Go-Back-N*) oferecem:

## Um serviço fiável, com conexão.

**Congestion Avoidance**

* Congestion Avoidance (additive increase)

» increments congestionWindow by 1 sgm, per RTT

* Detection of segment loss, by reception of 3 duplicated ACKs

» Assumes packet is lost, – Not by severe congestion, because following segments have arrived

» Retransmits lost packet

» CongestionWindow=CongestionWindow/2

Considere uma rede *Ethernet / IEEE 802.3* partilhada. Admita que, após ocorrer uma colisão entre duas estações, estas tentam resolver o conflito (com base no protocolo **CSMA/CD**) e que nenhuma outra tenta aceder ao meio

**UDP - User Datagram Protocol (UDP)**

* Datagram oriented

» Unreliable -> no error control mechanism

» Connectionless

* Allows applications to interface directly to IP with minimal additional protocol overhead
* UDP header

» Port numbers identify sending and receiving processes

» UDP length = length of packet in bytes

» Checksum covers header and data; optional

## À medida que aumenta o número de colisões, a probabilidade de uma nova colisão diminui.

Redes de comutação de pacotes podem operar nos modos de *Datagrams* (DG) ou Circuitos Virtuais (CV):

## Em CV o percurso de cada pacote está predefinido; em DG é determinado nó a nó.

A camada de transporte recebe da camada de rede IP um serviço que (ou: A camada de rede IP fornece ao protocolo UDP um serviço que)

1. **Não garante a entrega de todos os pacotes nem a sua sequência.**

Os protocolos da camada de transporte usam vários mecanismos de controlo, incluindo o mecanismo de Controlo de Fluxo (CF) e o mecanismo de Controlo de Congestionamento (CC). Na Internet, o protocolo *User Datagram Protocol* (UDP) usa:

## Não usa CF nem CC.

No protocolo TCP o emissor controla uma janela de congestionamento; no início da sessão TCP ou após time-out entra-se numa fase de *slow start*, que é seguida, após se atingir um limiar, por uma fase de *congestion avoidance*:

* 1. **A janela do emissor aumenta mais rapidamente durante *slow start* do que durante *congestion avoidance*.**

Uma rede composta por um conjunto de comutadores Ethernet interligados entre si constitui:

1. **Uma rede de comutação de pacotes e oferece um serviço não orientado às ligações.**

Um ligação TCP fica univocamente identificada pelo seguinte vector

1. **< PORTorig, IPorig, PORTdest, IPdest >**

Num canal sem fios e para uma dada potência de transmissão, a potência recebida é tanto maior

1. **Quanto menor for distância emissor-receptor e maior for o comprimento de onda da portadora.**

No protocolo de acesso ao meio CSMA/CD, quando uma estação emissora deteta uma colisão, esta estação:

## Aborta a transmissão da trama e retransmite a trama após espera de um número aleatório de *timeslots*.

Quando uma trama é recebida por um Switch Ethernet e a tabela de encaminhamento do Switch não contém uma entrada para o endereço de destino da trama, o Switch

1. **Envia a trama para todas as portas exceto a porta através da qual a trama foi recebida.**

O programa ping usado nas aulas laboratoriais gera pacotes de informação do:

O protocolo *Internet Control Message Protocol* (ICMP) usa serviços oferecidos pelo protocolo

**c) IP.**

## protocolo ICMP, que por usa vez são encapsulados em pacotes IP, que por sua vez são encapsulados em tramas Ethernet.

Considere uma rede de circuitos virtuais e, nesta rede, um dado circuito virtual. Nesta situação:

1. **Todos os pacotes do circuito transportam o mesmo identificador mas este identificador pode mudar de ligação para ligação.**

Numa transferência de dados FTP feita em modo passivo:

1. **As ligações de controlo e de dados são estabelecidas pelo cliente.**

Que protocolo de transporte (UDP ou TCP) usaria para as seguintes aplicações: (A1) obtenção de informação do servidor de nomes DNS;

A2) envio de um email;

A3) transferência de voz em pacotes.)

## A1=UDP; A2=TCP; A3=UDP.

Assuma que 8 estações competem para aceder a um meio partilhado, que cada estação gera em média 1 pacote/s e que o meio é capaz de transportar 10 pacote/s. Neste cenário, sob o ponto de vista do atraso:

## Um mecanismo TDMA é preferível a um mecanismo de acesso aleatório.

As técnicas de routing orientadas ao estado das ligações (link state routing) permitem obter

1. **O caminho único mais curto entre 2 nós IP.**

Quando uma trama é recebida num HUB Ethernet e se destina a um computador que não está ligado ao HUB, o HUB:.

1. **Envia a trama para todas as portas exceto a porta através da qual a trama foi recebida.**

No protocolo FTP, em resposta ao pedido de transferência de dados em modo passivo, o servidor envia ao cliente:

## O endereço da porta do servidor para a ligação de dados.

A camada de rede IP fornece ao protocolo UDP um serviço que:

## Não garante a entrega de todos os pacotes nem a sua sequência.

O protocolo de Transporte UDP (*User Datagram Protocol*) oferece às Aplicações que o usam:

1. **Um serviço não fiável, não orientado às ligações.**