

Redes de Computadores

Cap. 3 — Redes de Comutação de Pacotes

Departamento de Informática da
FCT/UNL

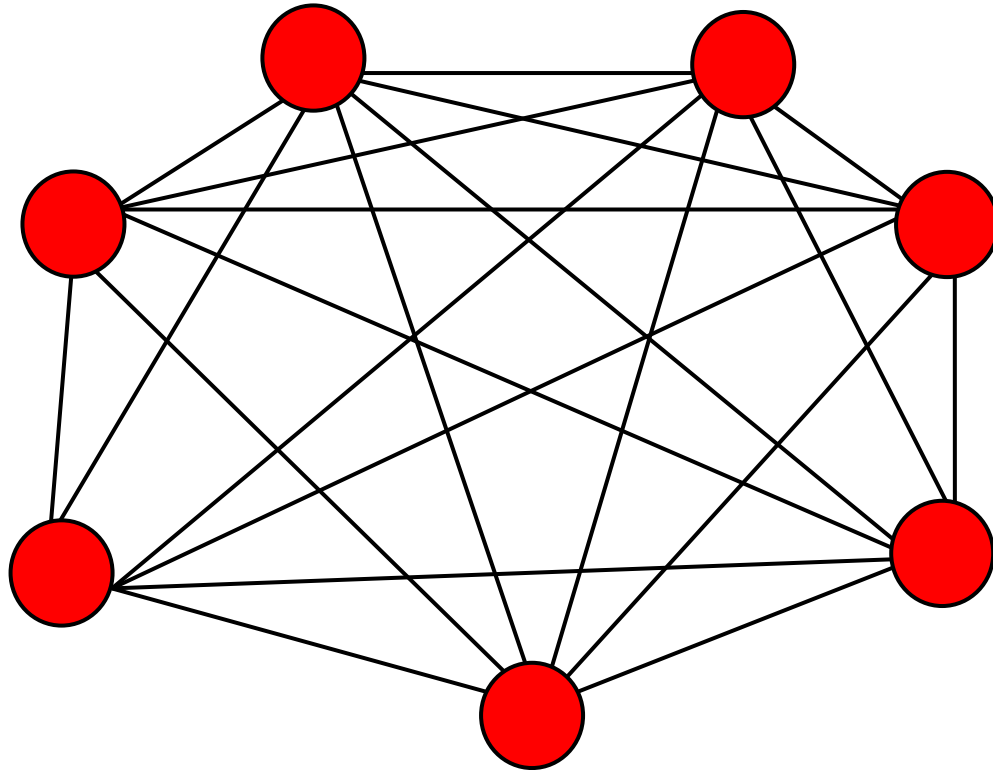
Objectivos da lição

- Qual a forma mais eficaz e mais económica de permitir a comunicação entre biliões de sistemas finais?
- Como acomodar a diversidade dos requisitos dos utilizadores e das aplicações?
- A resposta é: usar comutação de pacotes e uma rede simples, flexível e escalável

"For every complex problem there is an answer that is clear, simple, and wrong."

- Autor: *M.L. Mencken*

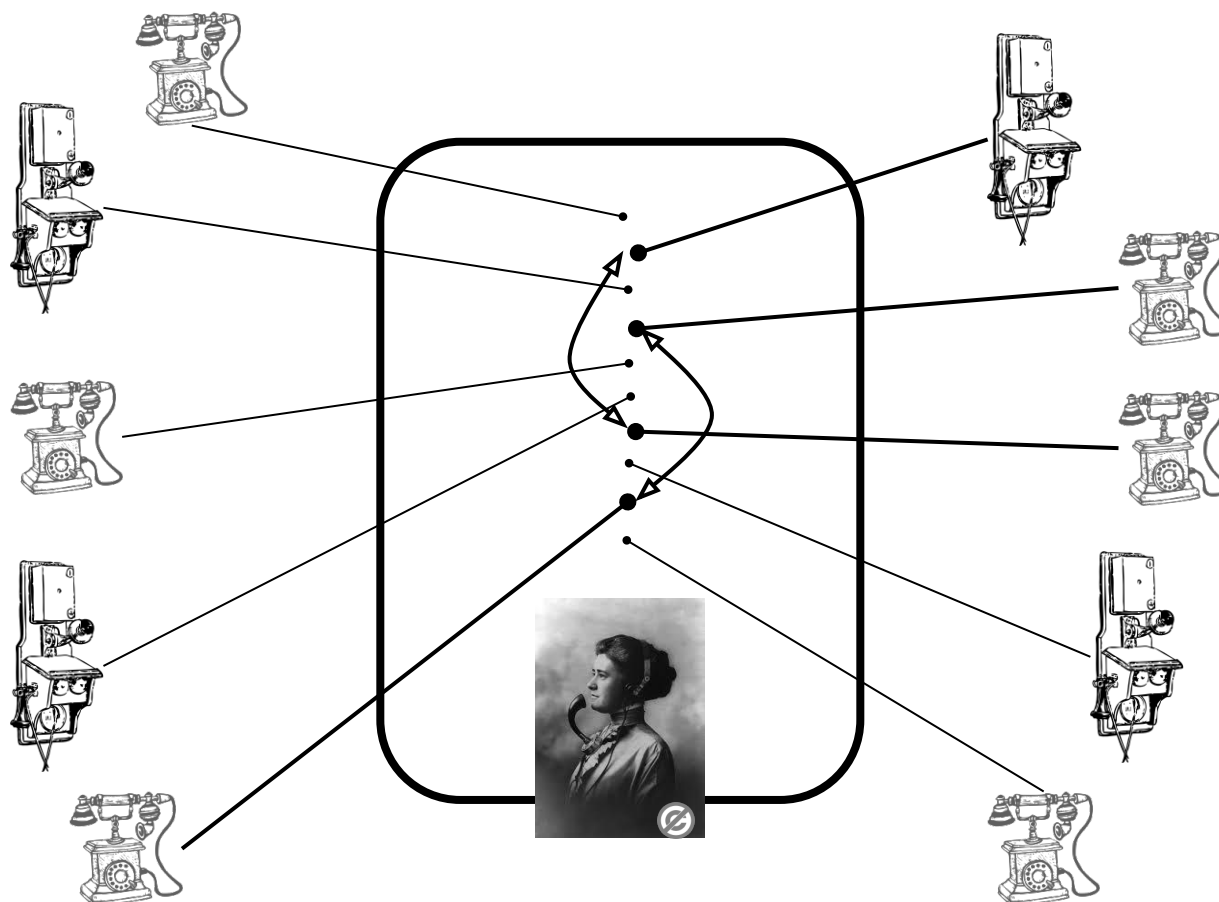
Como podem N sistemas comunicar ?



Usar $N \times (N - 1)$ canais não parece ser uma solução muito barata ou sempre adequada, sobretudo quando os sistemas estão distantes

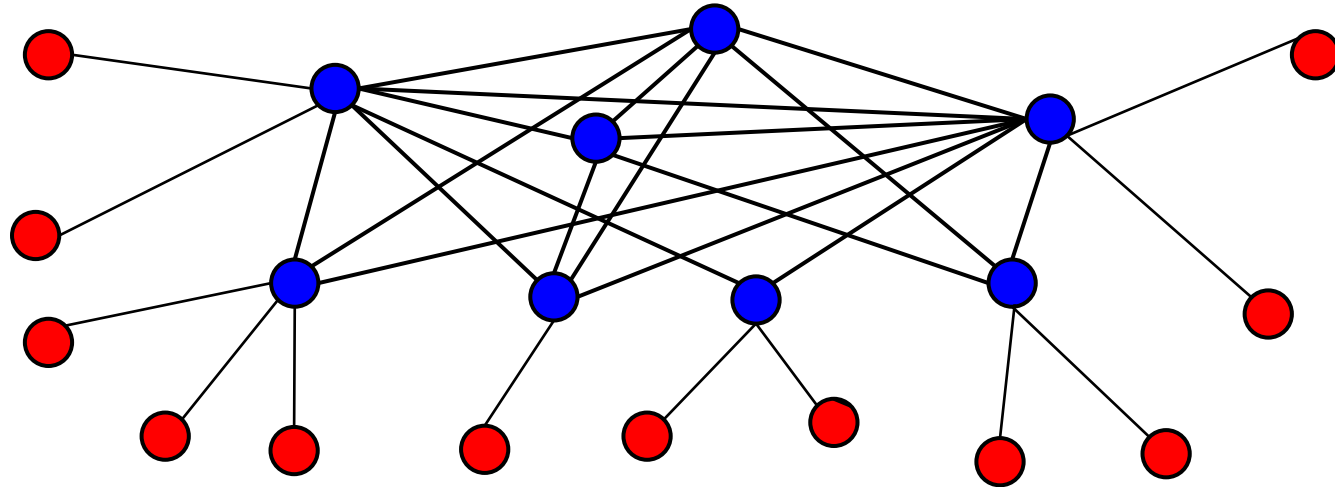
O problema não é novo

Comutação manual de circuitos telefónicos





Solução: Comutar (Aguilhar) e Agregar



- Rede comutada (*Switching*)

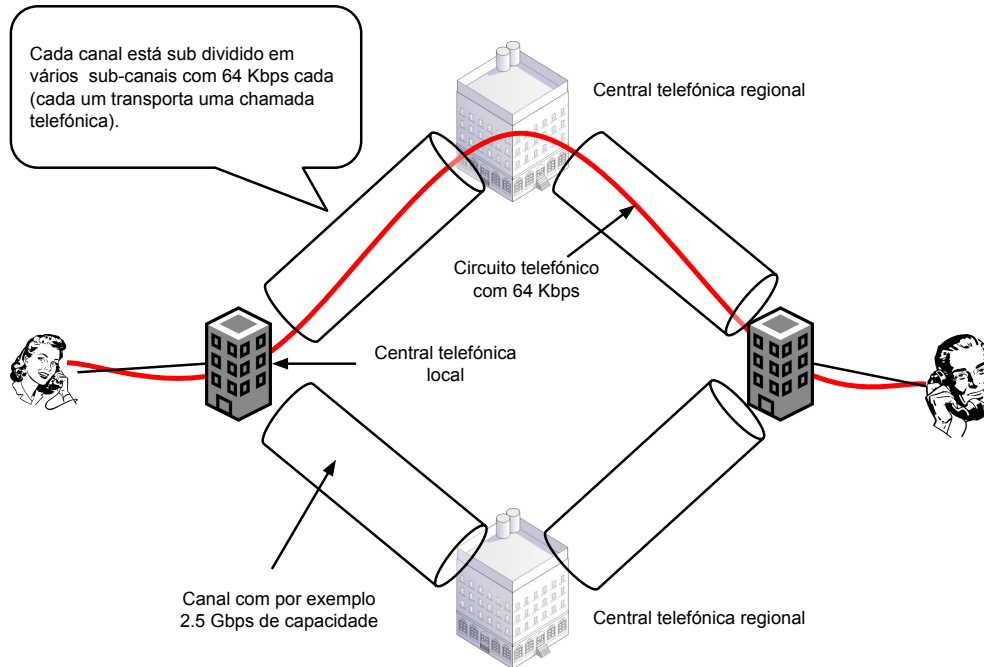
- Sistemas terminais ligados a nós de comutação
- Nós comutam (i.e. agulham) as comunicações
- Menos canais que pares de comunicação possíveis



- Agregação de canais (*Multiplexing*)

- Comunicações entre nós distintos partilham o mesmo canal

Comutar e Agregar Comunicações

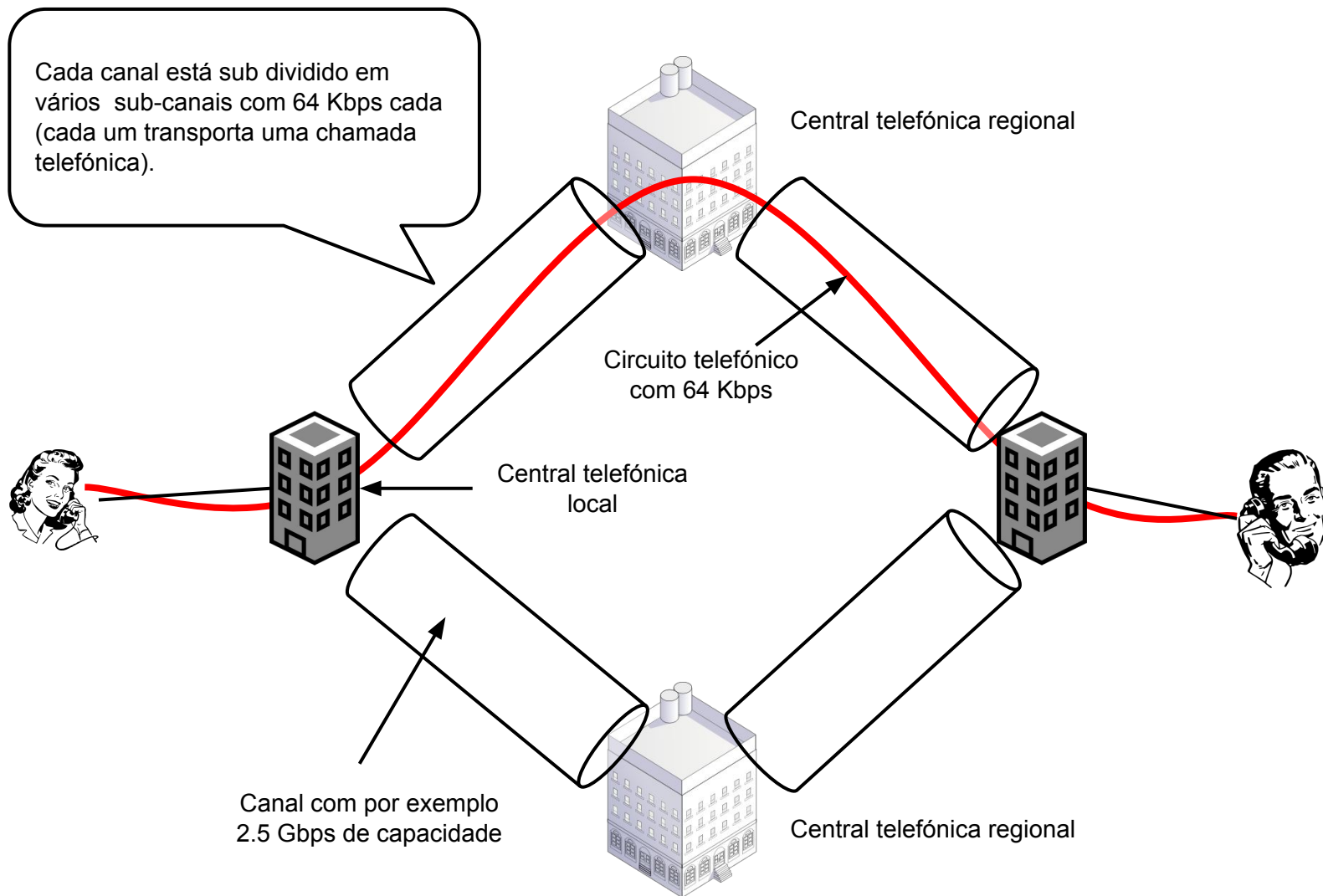


- Rede comutada (*Switching*)
 - Sistemas terminais ligados a nós de comutação
 - Nós comutam (i.e. agulham) as comunicações
 - Menos canais que pares de comunicação possíveis
- Agregação de canais (*Multiplexing*)
 - Comunicações entre nós distintos partilham o mesmo canal

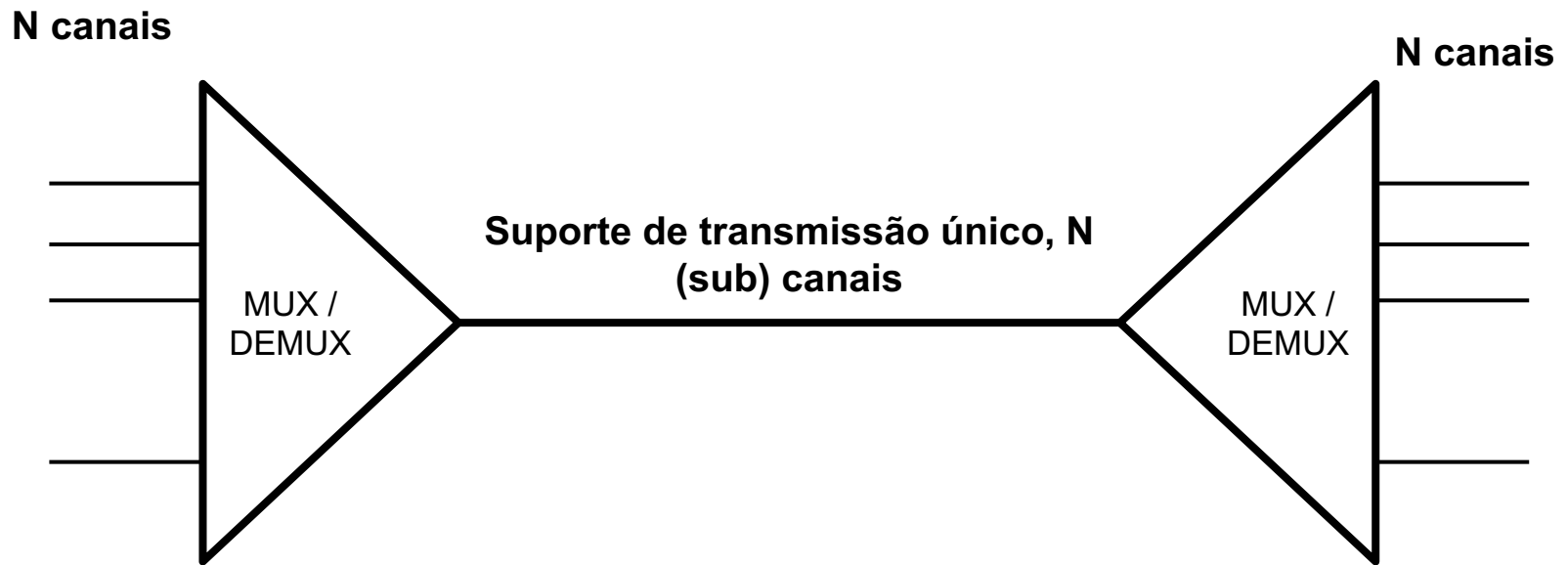
Redes de Circuitos (e.g. Telefónicos)

- Primeiro estabelece-se um circuito entre os dois nós
 - Afetando **sub-canaís** em cada canal ao circuito
 - No fundo, arranjam-se vários sub-canaís entre dois equipamentos de comutação
- A comunicação faz-se usando o circuito assim estabelecido
- Quando já não se necessita do circuito, desfaz-se o mesmo libertando os sub-canaís a ele afetados

Comutação de Circuitos Digitais Telefônicos

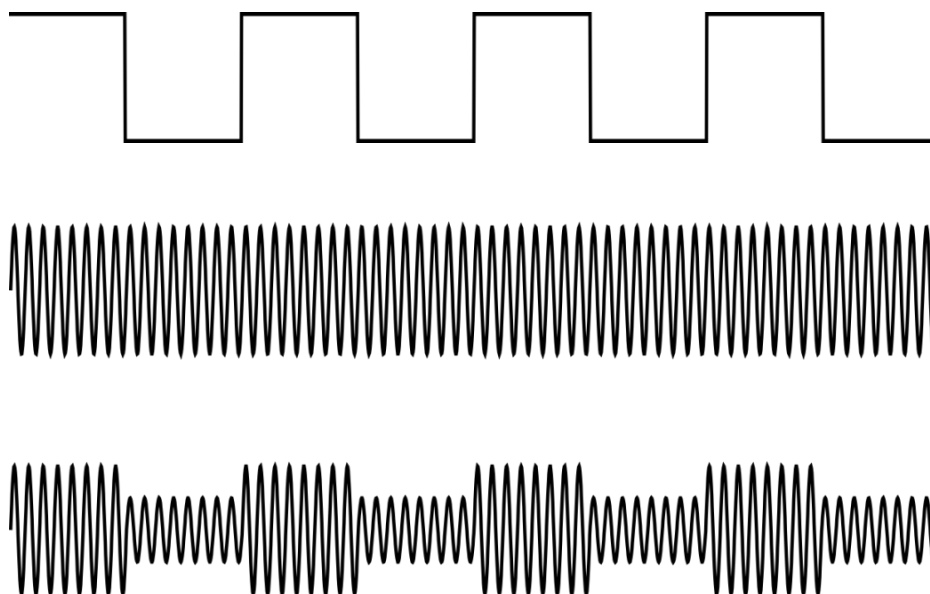


Agregação (Multiplexagem) de Canais

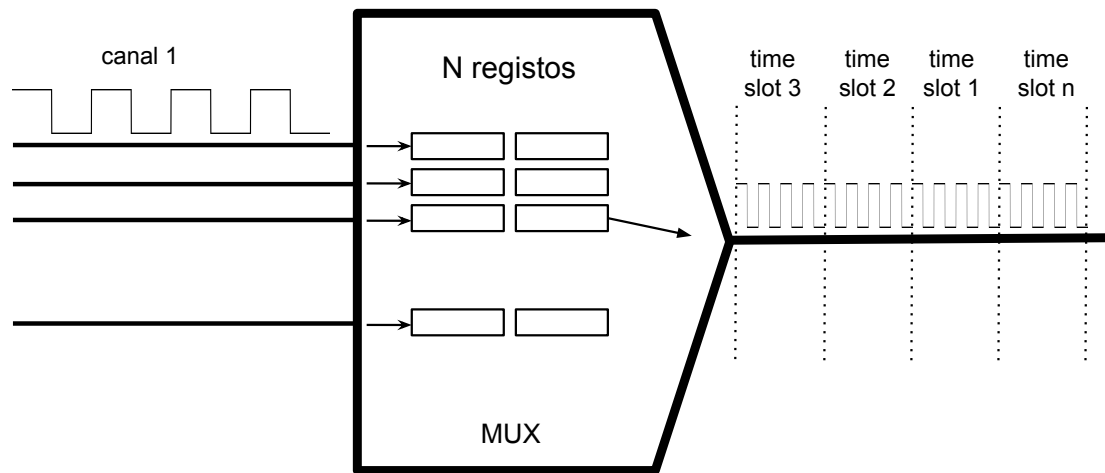


Multiplexagem em Frequência (FDM)

- Cada sub-canal usa uma frequência distinta mas todos partilham o mesmo meio de transmissão



Multiplexagem Temporal (TDM)



- Cada sub-canal é servido quando chega a sua vez de uma forma bem definida e constante.
- Também se pode fazer o mesmo com multiplexagem em frequência.

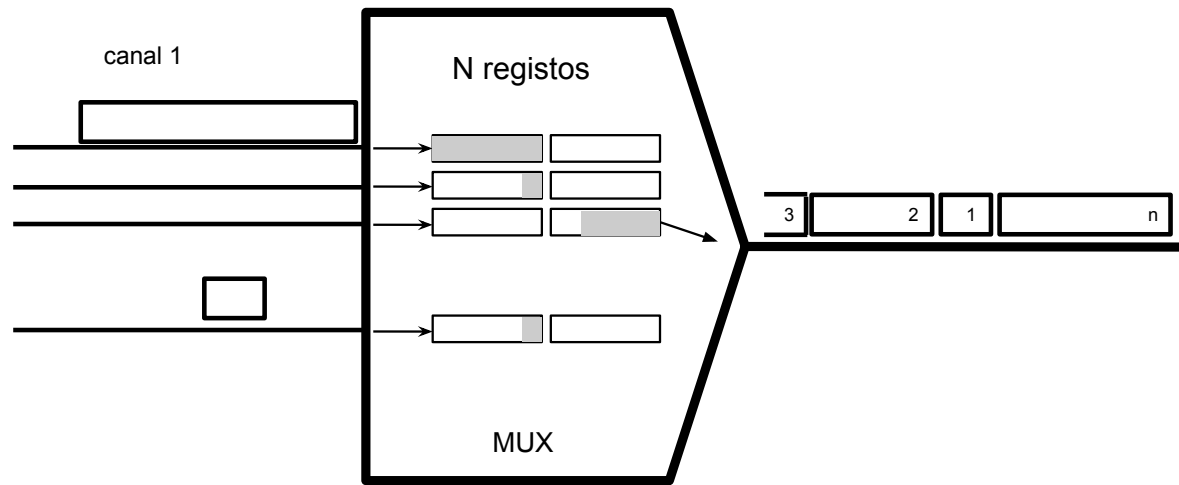
Vantagens dos Circuitos

- Não há a necessidade de ter cabeçalhos complexos
 - Os comutadores comutam os bits com base em frequências ou intervalos de tempo
 - Não há cabeçalhos IP, UDP, TCP,
 - No limite eram construídos apenas com eletrónica, sem necessidade de computadores
- Capacidade garantida
 - A capacidade afetada de extremo a extremo é garantida
 - Os dispositivos que comunicam podem ser mais simples
 - Não têm que lidar com problemas como a perda de pacotes, tempos de transito variáveis, pacotes fora de ordem

Desvantagens dos Circuitos

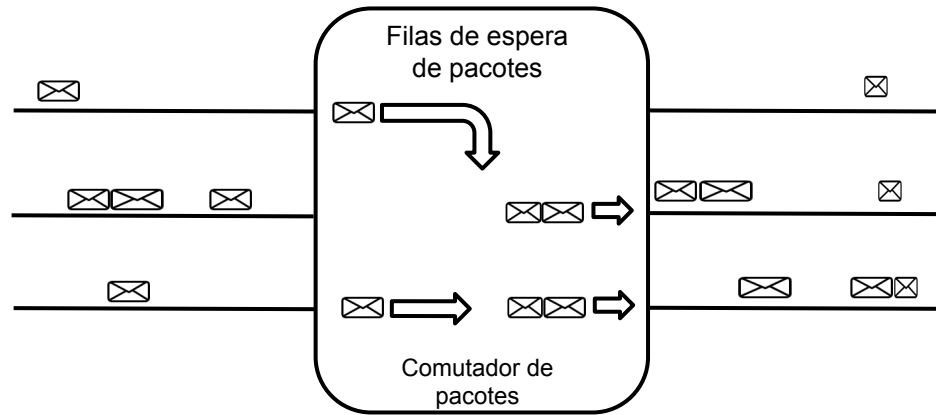
- Desperdício da capacidade
 - Quando não há tráfego a capacidade dos circuitos é desperdiçada visto que não é usada
- Bloqueio de comunicações
 - Quando não há sub-canais disponíveis bloqueia as comunicações pois não se conseguem instalar circuitos
- Antes de comunicar é necessário estabelecer o circuito
 - No caso do DNS por exemplo seria um grande desperdício
- Complexidade nos nós
 - Se há uma avaria num nó, todos os circuitos que o atravessam se perdem, por isso os nós de comutação têm de ser de alta fiabilidade

Multiplexagem Estatística



A afectação de um canal a um sub-fluxo depende das características do tráfego dos diferente sub-fluxos

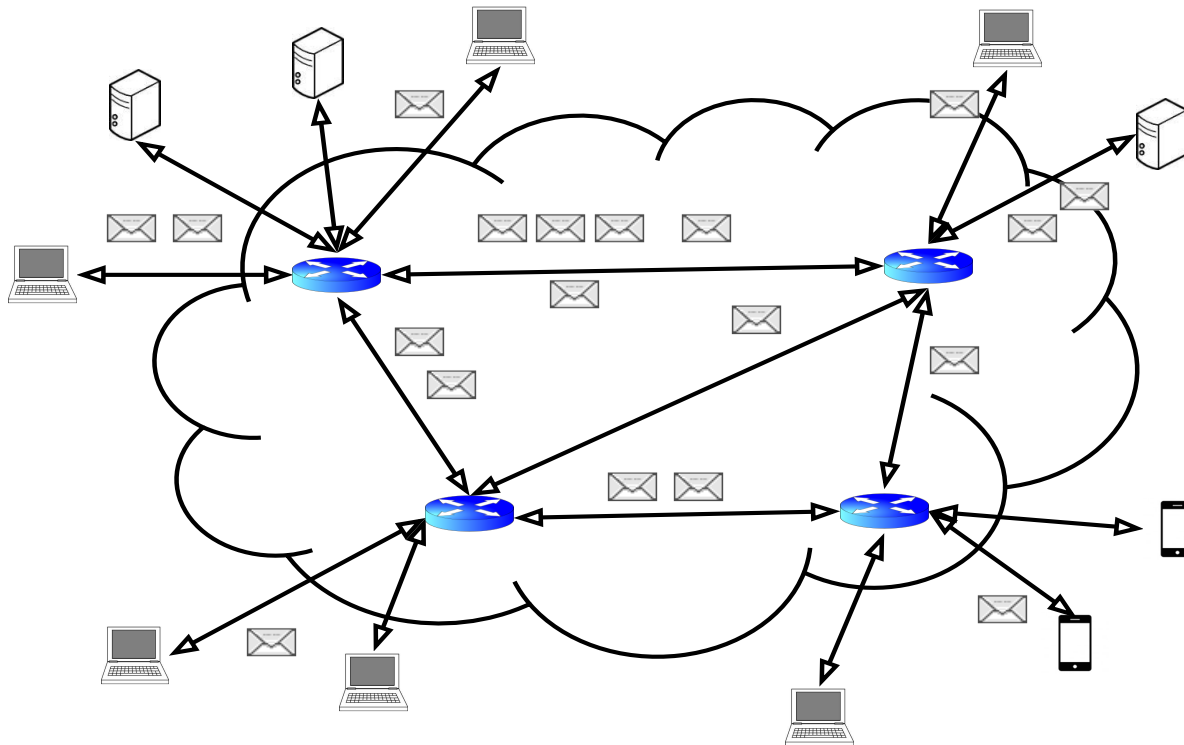
Comutação de Pacotes



- O tráfego é dividido em pequenos pacotes (de bits)
 - Cada pacote tem um cabeçalho com o endereço do destino
- Os pacotes atravessam a rede de forma flexível
 - O encaminhamento é baseado no endereço de destino
 - Os comutadores memorizam os pacotes momentaneamente usando uma forma de trabalho dita **store & forward**
 - Geralmente, associada a cada saída (e entrada) existe uma fila de espera de pacotes

Comutação de Pacotes (*Packet Switching*)

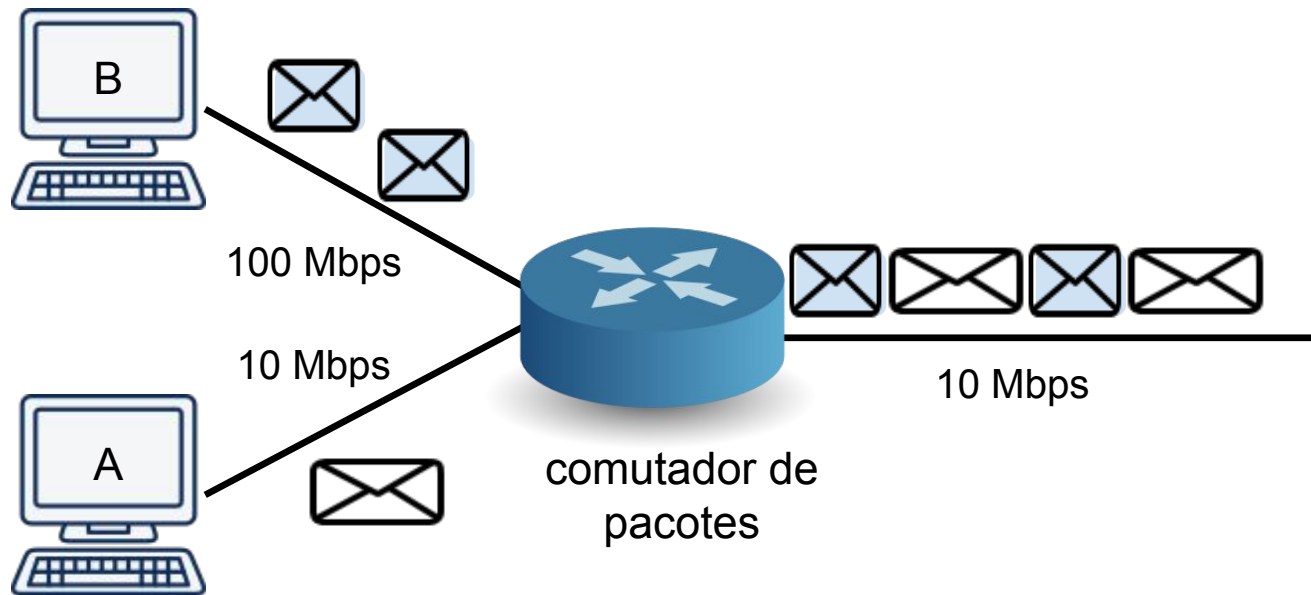
- O tráfego é dividido em pequenos pacotes (de bits)
 - Cada pacote tem um cabeçalho com o endereço do destino
- Os pacotes atravessam a rede de forma flexível
 - O encaminhamento é baseado no endereço de destino
 - Os comutadores podem memorizar os pacotes momentaneamente usando uma forma de trabalho dita *store & forward*
- O computador no destino final reconstrói a mensagem original



Flexibilidade é a Vantagem Principal

- Como o tráfego de dados é muito irregular é possível acomodar muitos mais fluxos de comunicação simultaneamente
- Como o tráfego de dados é predominantemente elástico, pode adaptar-se e admitir ainda mais comunicações simultâneas
- Cada nó emissor tem necessariamente uma fila de espera de pacotes à espera de serem transmitidos o que permite tratar de forma diferente fluxos de pacotes com necessidades distintas

Filas de Espera



- A fila de espera implementa a flexibilidade
 - Não há reserva de capacidade e aproveita-se ao máximo a que está disponível
 - Se necessário, podem dar-se prioridades diferentes a diferentes fluxos de dados

Redes de Comutação de Pacotes

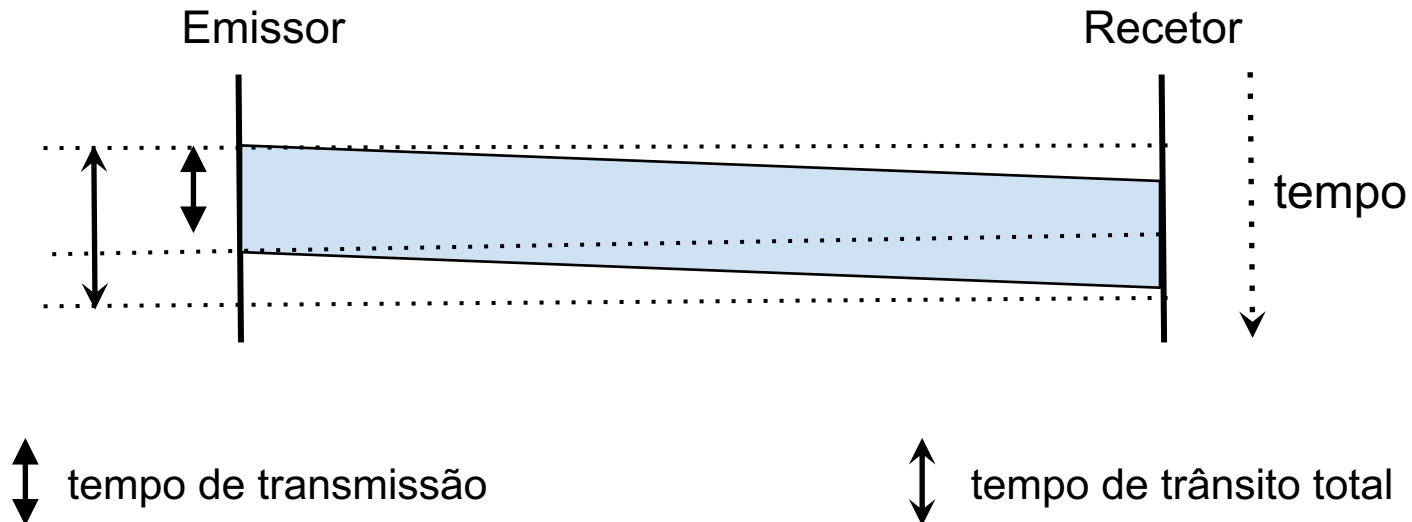
- Para que N sistemas comuniquem entre si usam-se
 - Nós de comutação
 - Agregam-se diversas comunicações para partilha dos canais (*multiplexing*)
- A comunicação de dados na Internet baseia-se em
 - transmissão e comutação de pacotes
 - *statistical multiplexing*
 - nós de comutação de pacotes que funcionam segundo o modo *store & forward*

Tempo de Trânsito de um Pacote

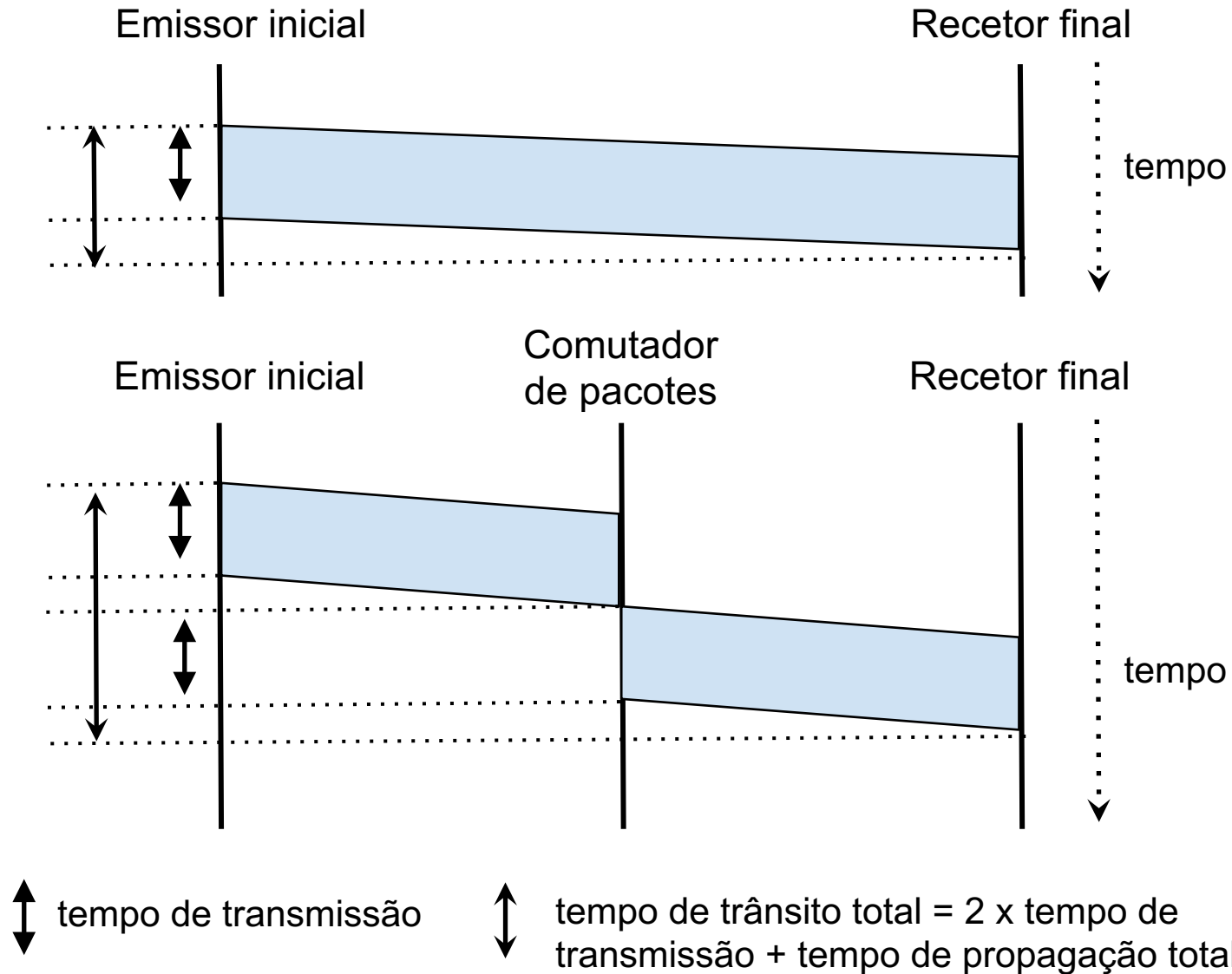
Tempo de trânsito de um pacote =
Tempo de transmissão + Tempo de propagação

Tempo de transmissão = Dimensão do pacote / Débito

Tempo de propagação = Dimensão do canal / Velocidade de propagação



Store-&-forward Introduz Atrasos Extra



Tempo de Trânsito Extremo a Extremo

$$T = \sum (N/D(i) + C(i)/V_p) = \sum (T_t(i) + T_p(i))$$

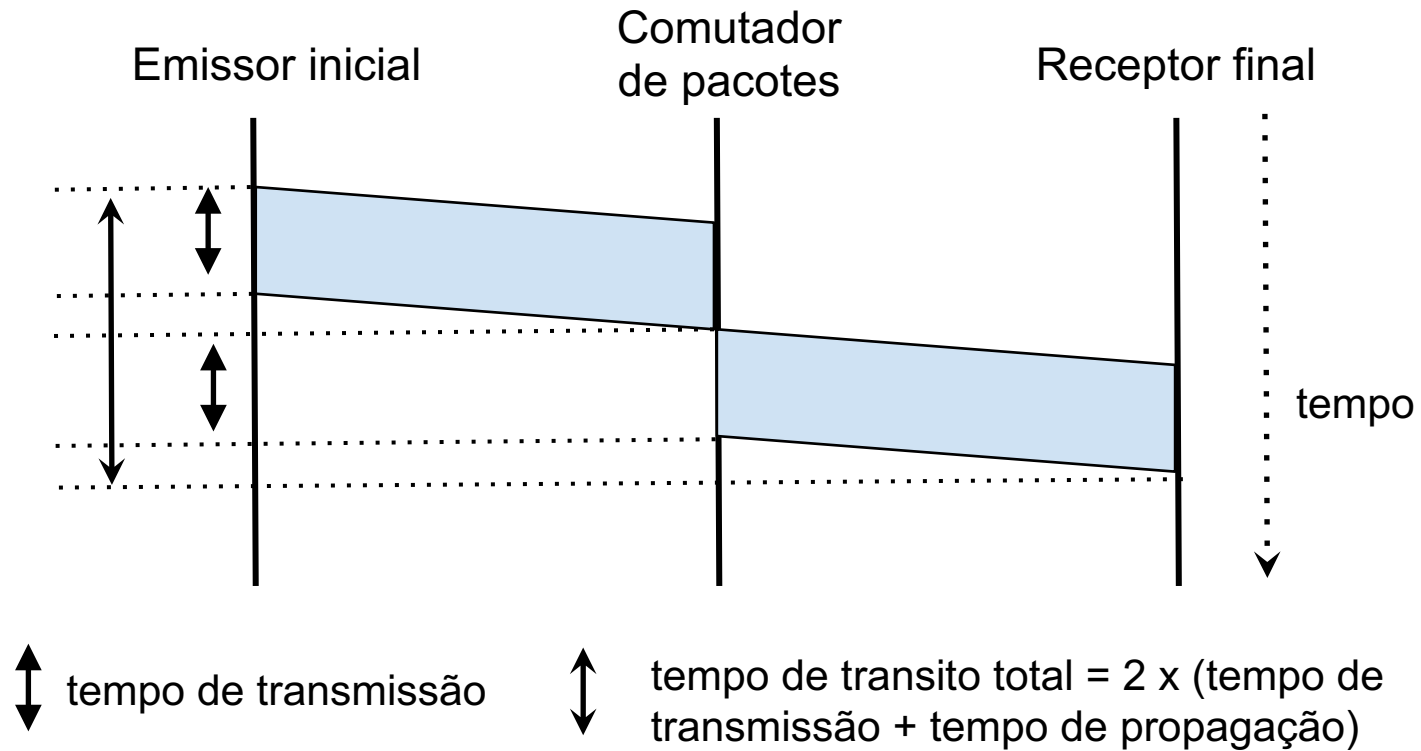
- O tempo de transito extremo a extremo de um pacote com N bits é igual ao somatório em i (totalidade dos canais que este atravessa) de

tempo de transmissão do pacote pelo canal i ($N / \text{Débito}_i$) +

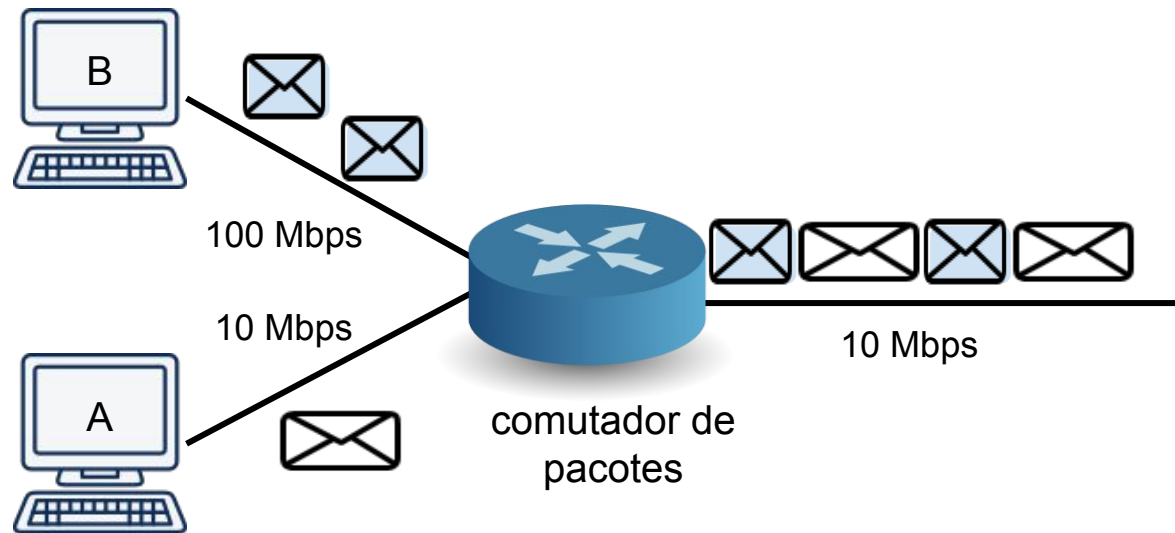
tempo de propagação do canal i ($\text{Comprimento}_i / V_p$)

$$\sum (T_t(i) + T_p(i))$$

Exemplo com Dois Canais

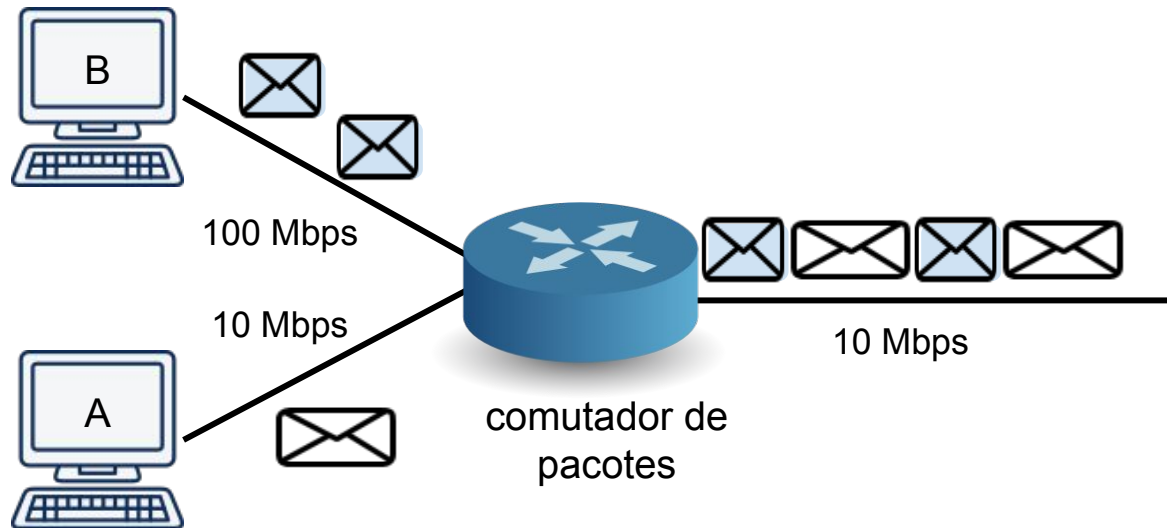


What If ?



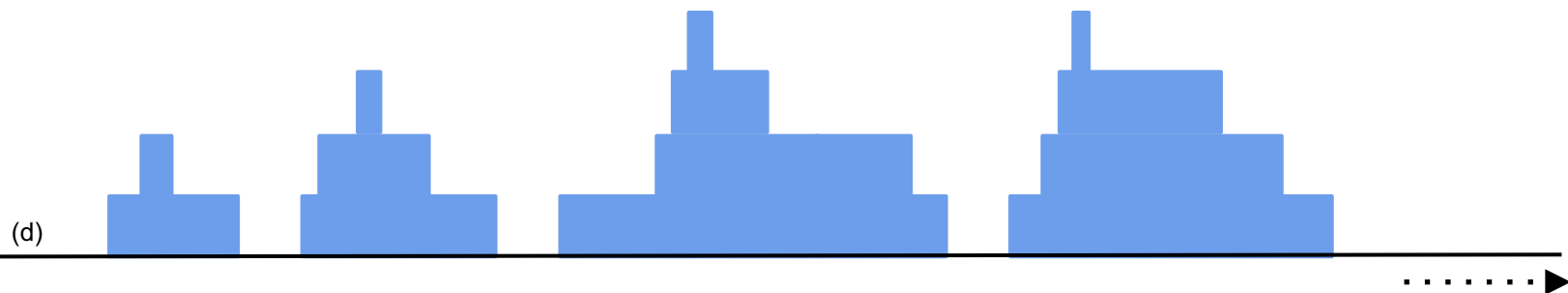
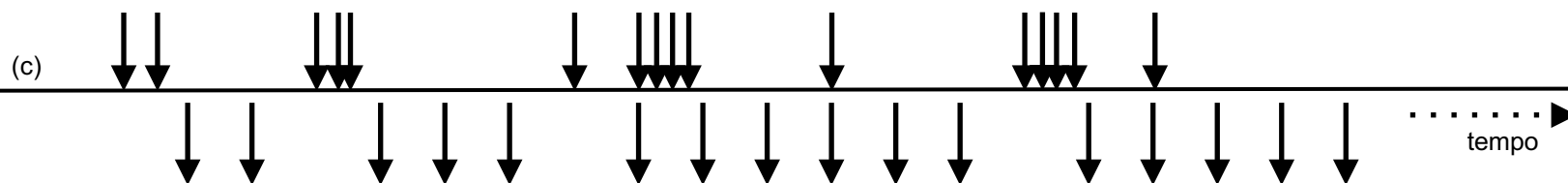
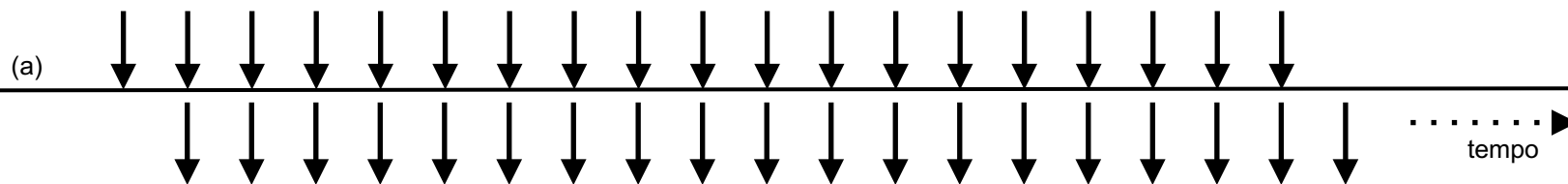
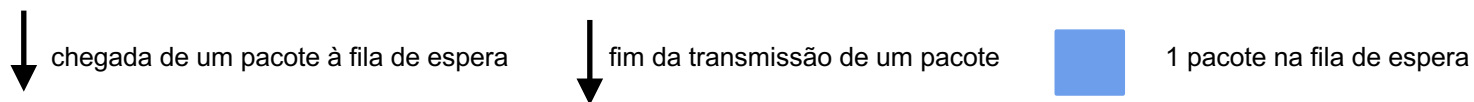
Que acontece se A e B transmitem pacotes simultaneamente e quando algum pacote chega, encontra pacotes à sua frente na fila de espera ?

Dois Cenários



- No primeiro só **A** emite pacotes, todos iguais com intervalos certos e **B** não envia pacotes
- No segundo **A** e **B** enviam ambos tráfego e de forma descoordenada

Dois Cenários



Tempo de Trânsito Extremo a Extremo

$$T = \sum (T_t(i) + T_p(i) + T_{fe}(i))$$

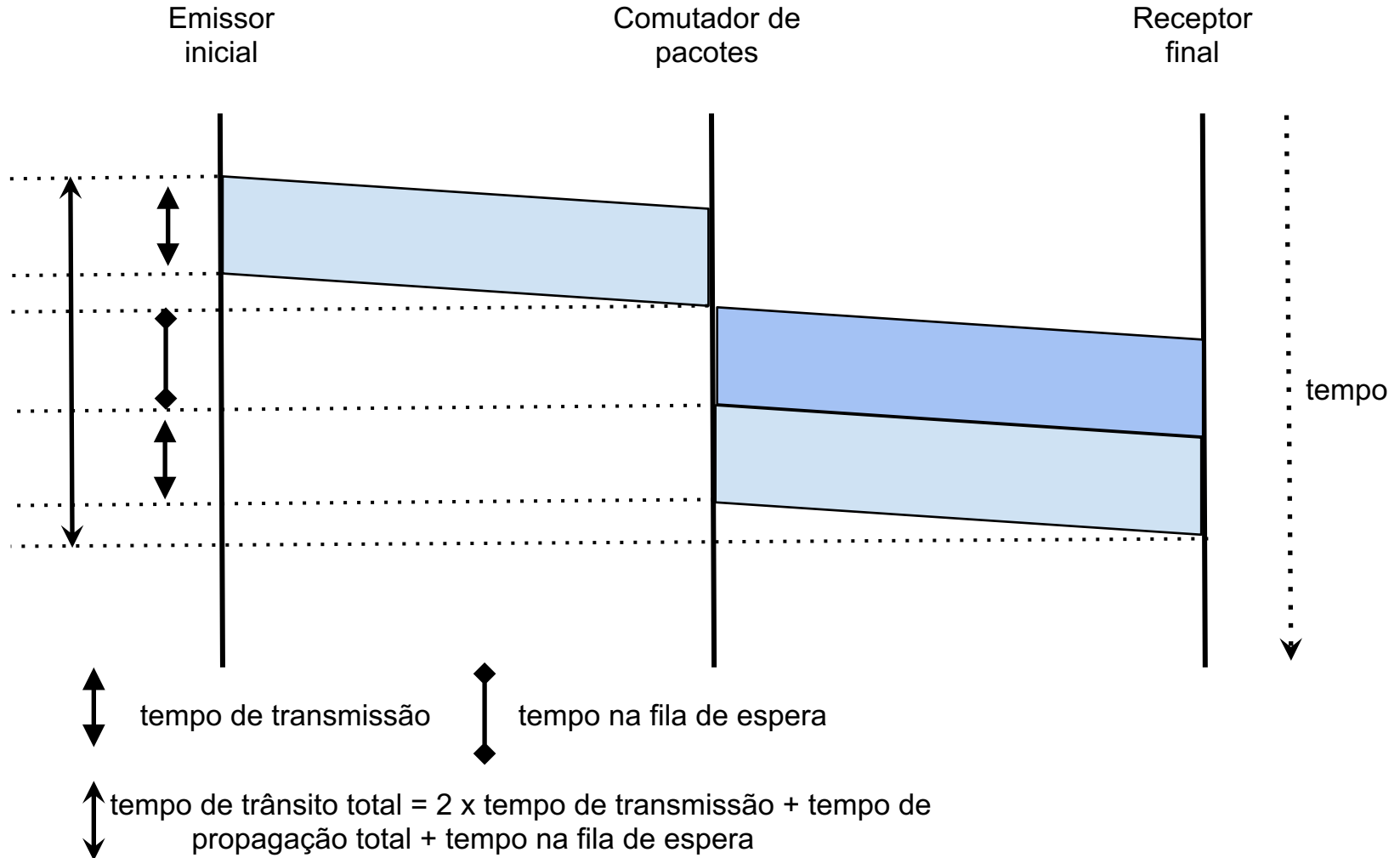
- O tempo de transito extremo a extremo de um pacote com N bits é igual ao somatório em i (totalidade dos canais que este atravessa) de:

tempo de transmissão do pacote pelo canal i +

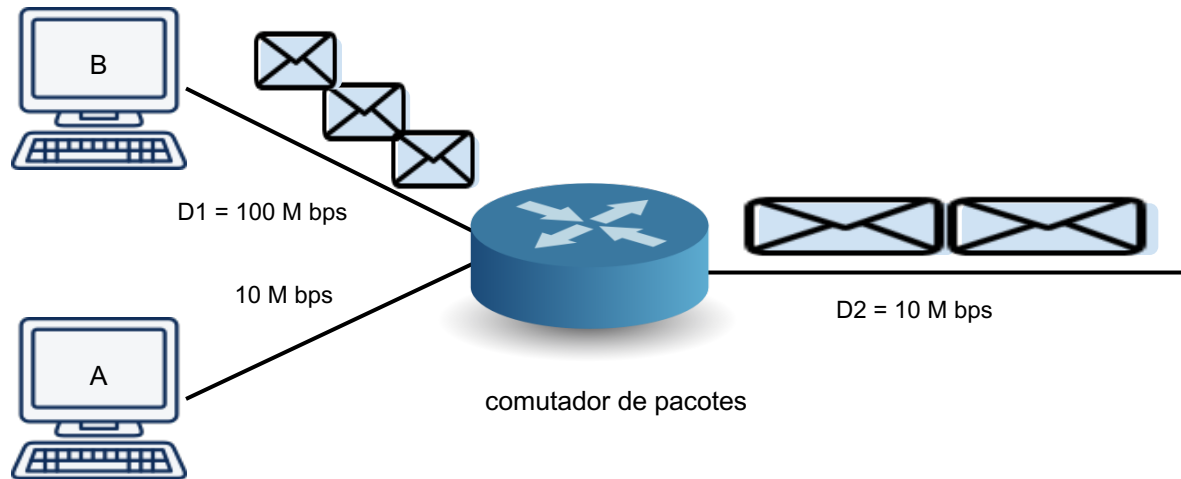
tempo de propagação do canal i +

tempo na fila de espera associada a i

Exemplo com Dois Canais

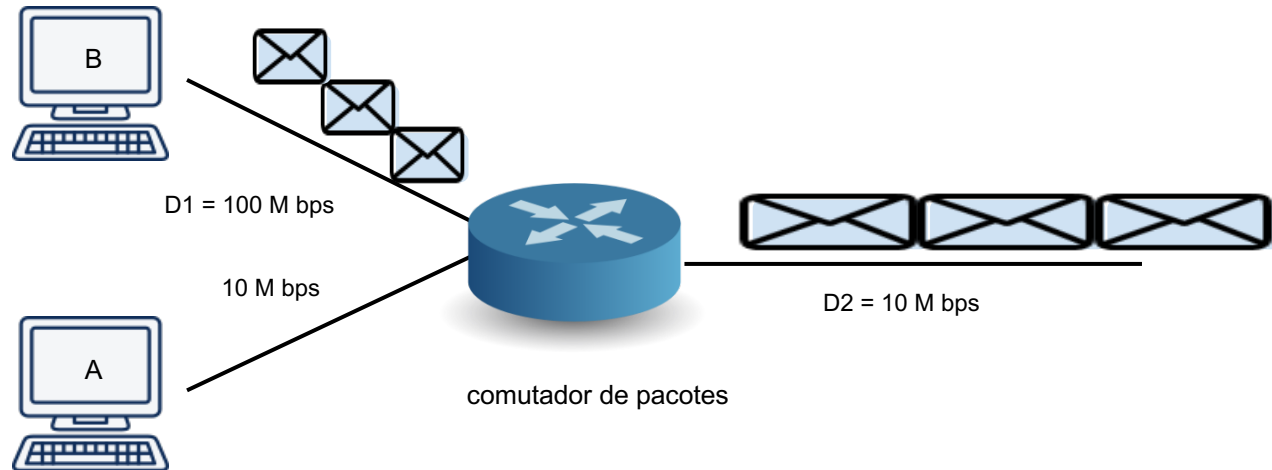


What If ?



Que acontece se o débito D_1 (do canal que liga B ao comutador) é maior que o débito D_2 (do canal à saída do comutador) e o emissor original transmite sem interrupção pacote atrás de pacote?

A Fila Tenderia para Infinito



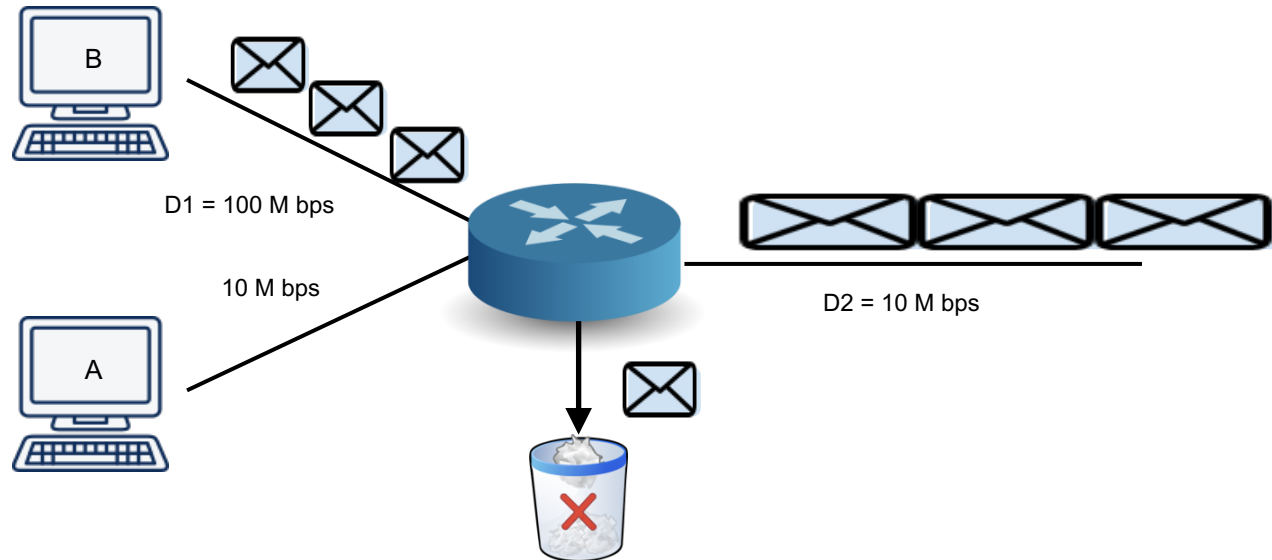
Que acontece se o débito D_1 (do canal que liga B ao comutador) é maior que o débito D_2 (do canal à saída do comutador) e o emissor original transmite sem interrupção pacote atrás de pacote?

A fila de espera tenderia para infinito e o tempo de trânsito também !

$$T = \sum (T_t(i) + T_p(i) + T_{fe}(i))$$

T_{fe} é um fator variável que nesse caso teria tendência para crescer para o infinito

Solução: Limitar a Fila de Espera



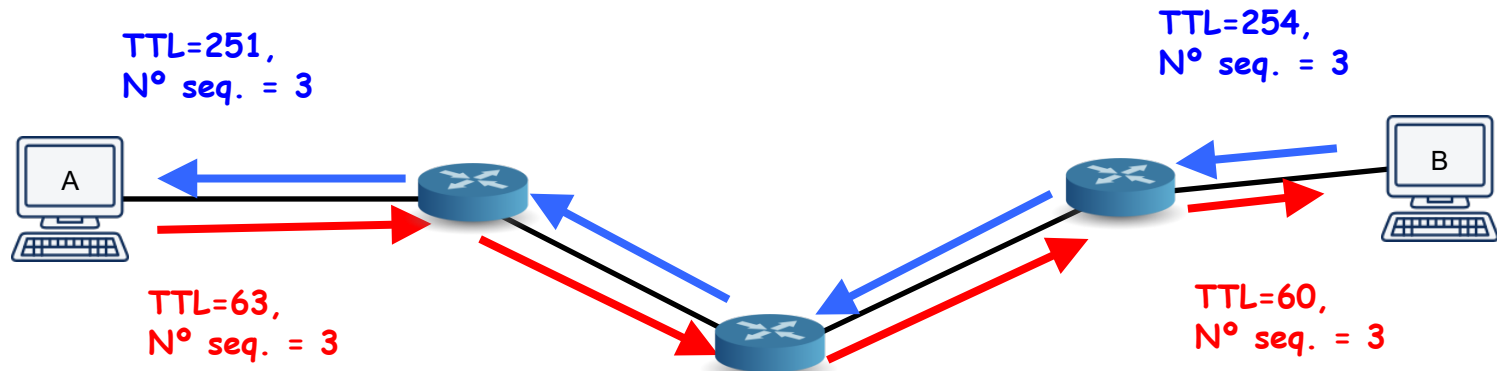
Mais vale suprimir alguns pacotes pois um pacote que chega demasiado tarde é como se se tivesse perdido

$$T_{\text{máximo}} = \sum (T_{\text{t}}(i) + T_{\text{p}}(i) + T_{\text{fe-max}}(i))$$

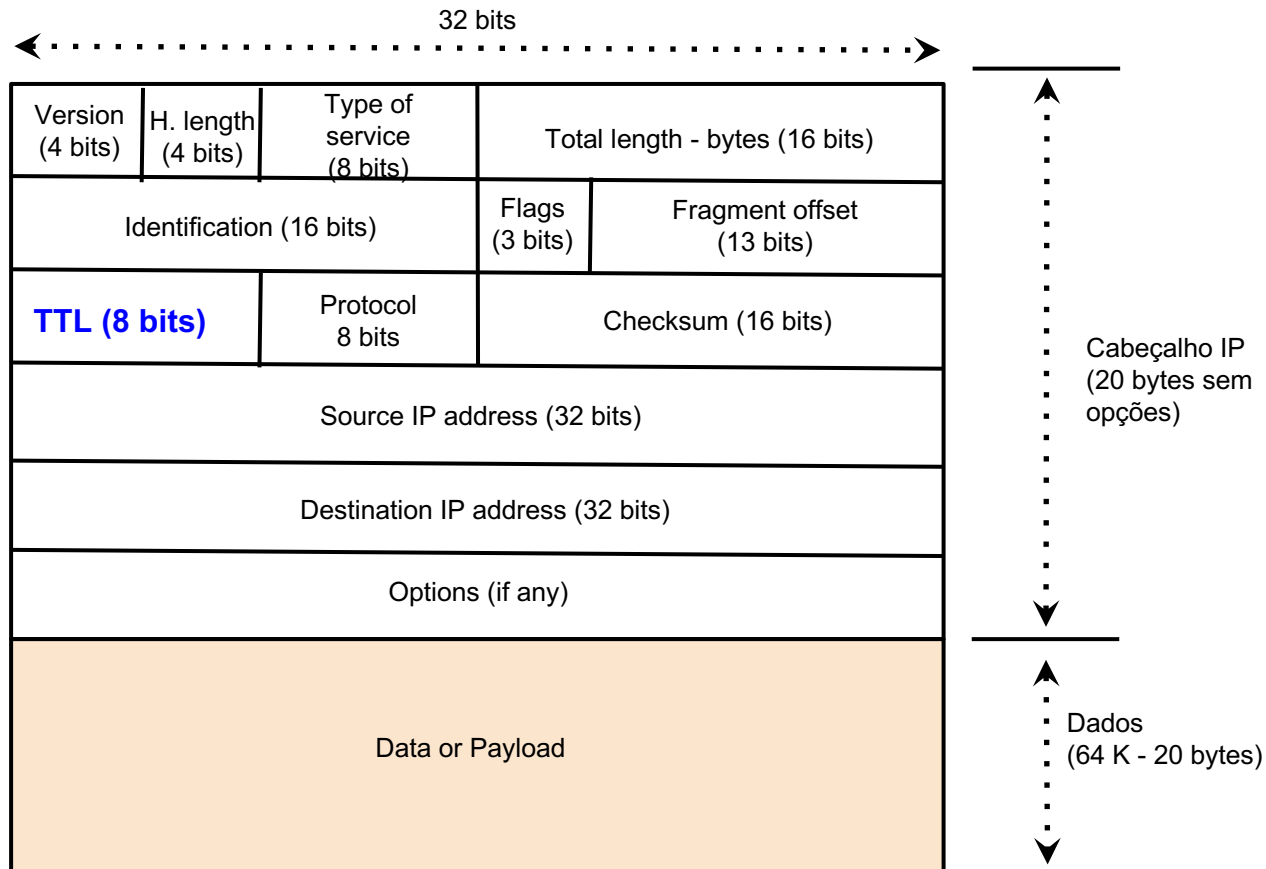
$T_{\text{fe-max}}$ é o tempo máximo de espera na fila de espera dos pacotes que passam. Geralmente $T_{\text{fe}} \leq T_{\text{fe-max}}$

Uso do programa Ping para Medir o RTT

- **A envia periodicamente pacotes a B**
 - Cada vez que envia um pacote registra o valor do relógio
 - Cada pacote tem um n.º de sequência crescente
- **Cada vez que B recebe um pacote responde a A**
 - Com o n.º de sequência recebido
- **Cada vez que A recebe um pacote de B**
 - Calcula o tempo de trânsito *end-to-end* (RTT - *Round Trip Time*)

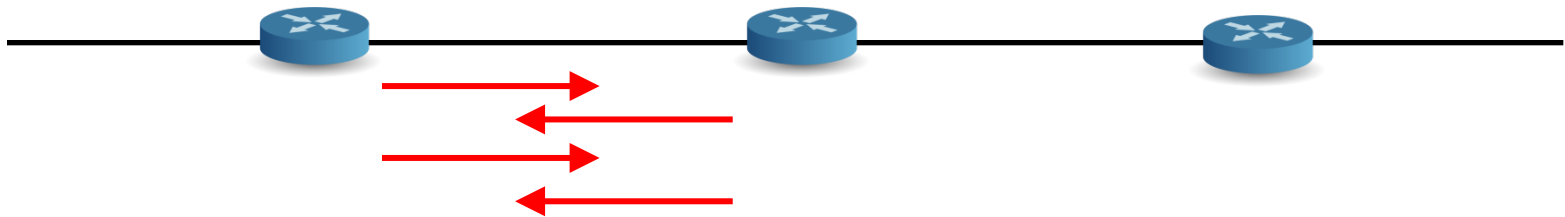


Formato de um Pacote IP



Time-to-Live (TTL)

- Mecanismo de segurança se existirem problemas
 - Ciclos de encaminhamento por erros ou instabilidade
 - Saturam completamente os canais em jogo



- O campo *time-to-live* do cabeçalho IP
 - O campo é decrementado sempre que o pacote chega a um nó de comutação
 - Se chega a 0 é suprimido ...
 - ...e uma mensagem "*time exceeded*" é enviada à origem

Ping para Paris - 2000 Km

```
$ ping -c 6 www.inria.fr
```

```
PING ezp3.inria.fr (128.93.162.84): 56 data bytes
```

```
64 bytes from 128.93.162.84: icmp_seq=0 ttl=51 time=39.508 ms
```

```
64 bytes from 128.93.162.84: icmp_seq=1 ttl=51 time=38.203 ms
```

```
64 bytes from 128.93.162.84: icmp_seq=2 ttl=51 time=38.367 ms
```

```
64 bytes from 128.93.162.84: icmp_seq=3 ttl=51 time=37.888 ms
```

```
64 bytes from 128.93.162.84: icmp_seq=4 ttl=51 time=38.262 ms
```

```
64 bytes from 128.93.162.84: icmp_seq=5 ttl=51 time=37.895 ms
```

```
--- ezp3.inria.fr ping statistics ---
```

```
6 packets transmitted, 6 packets received, 0.0% packet loss
```

```
round-trip min/avg/max/stddev = 37.888/38.354/39.508/0.546 ms
```

Lisboa / Paris são 1454 Km em linha reta, a que corresponde um tempo de propagação directo de 7 ms, mas o RTT médio medido é de 39.508 ms (o RTT teórico é 14 ms) e o TTL é 51

Ping para Los Angeles - 10.000 Km

```
$ ping -c 6 ucla.edu
```

```
PING ucla.edu (128.97.27.37): 56 data bytes
```

```
64 bytes from 128.97.27.37: icmp_seq=0 ttl=50 time=177.351 ms
```

```
64 bytes from 128.97.27.37: icmp_seq=1 ttl=50 time=176.195 ms
```

```
64 bytes from 128.97.27.37: icmp_seq=2 ttl=50 time=175.938 ms
```

```
64 bytes from 128.97.27.37: icmp_seq=3 ttl=50 time=176.594 ms
```

```
64 bytes from 128.97.27.37: icmp_seq=4 ttl=50 time=175.894 ms
```

```
64 bytes from 128.97.27.37: icmp_seq=5 ttl=50 time=175.830 ms
```

```
--- ucla.edu ping statistics ---6 packets transmitted, 6 packets received, 0.0% packet loss  
round-trip min/avg/max/stddev = 175.830/176.300/177.351/0.535 ms
```

Lisboa / Los Angeles são 9155 Km em linha reta, a que corresponde um tempo de propagação directo de 43 ms, mas o RTT médio medido é de 198 ms (o RTT teórico é 86 ms) e o TTL é 47

Ping to Auckland (NZ) - 20000 Km

```
$ ping -c 6 www.auckland.ac.nz
PING www.auckland.ac.nz (130.216.159.127): 56 data bytes
64 bytes from 130.216.159.127: icmp_seq=0 ttl=231 time=294.663 ms
64 bytes from 130.216.159.127: icmp_seq=1 ttl=231 time=293.014 ms
64 bytes from 130.216.159.127: icmp_seq=2 ttl=231 time=293.077 ms
64 bytes from 130.216.159.127: icmp_seq=3 ttl=231 time=292.939 ms
64 bytes from 130.216.159.127: icmp_seq=4 ttl=231 time=293.009 ms
64 bytes from 130.216.159.127: icmp_seq=5 ttl=231 time=292.044 ms

--- www.auckland.ac.nz ping statistics ---
6 packets transmitted, 6 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 292.044/293.124/294.663/0.774 ms
```

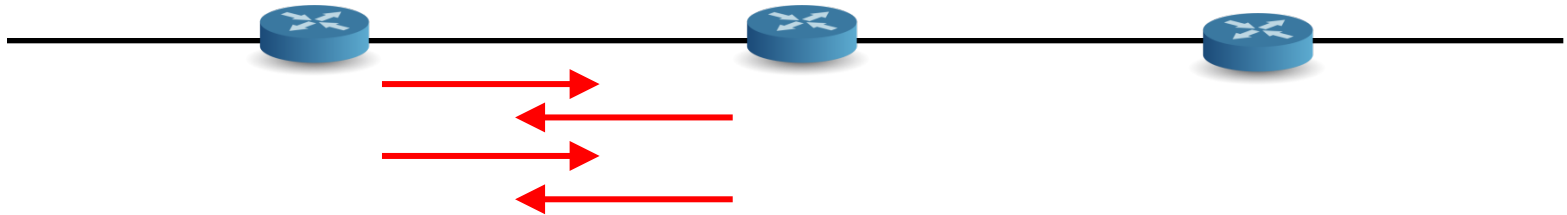
Lisboa / Auckland são 19680 Km em linha reta, a que corresponde um tempo de propagação directo de 92 ms, mas o RTT médio medido é de 294 ms (o RTT teórico é 184 ms) e o TTL é 231

Conclusões

- A distância e o tempo de propagação têm uma influência decisiva no tempo de trânsito de extremo a extremo
- A influência das filas de espera nota-se na variação do tempo de trânsito
- Quanto maiores as variações maior variabilidade deve existir na ocupação destas filas de espera quando os pacotes do ping chegaram
- No entanto, existe também a influência do tempo de processamento por cada equipamento intermédio

Time-to-Live (TTL)

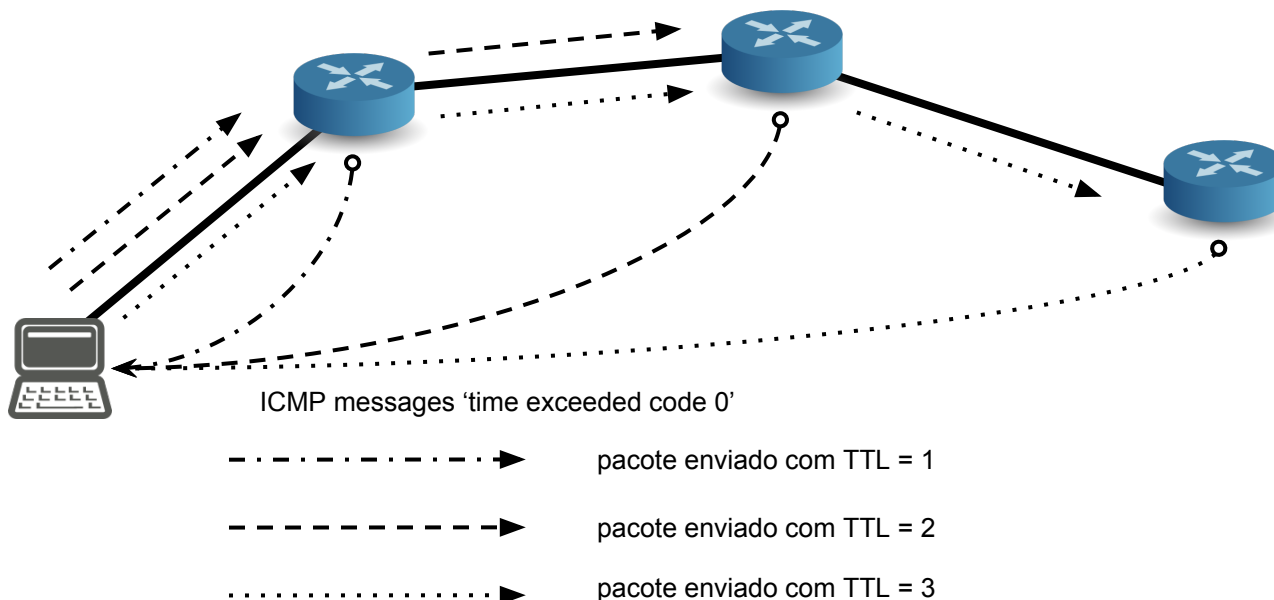
- Mecanismo de segurança se existirem problemas
 - Ciclos de encaminhamento por erros ou instabilidade
 - Saturam completamente os canais em jogo



- O campo *time-to-live* do cabeçalho IP
 - O campo é decrementado sempre que o pacote chega a um nó de comutação
 - Se chega a 0 é suprimido ...
 - ...e uma mensagem "*time exceeded*" é enviada à origem

Uso do TTL pelo Traceroute

- Utilização do campo TTL
 - A origem envia um pacote com TTL de n
 - Cada nó decrementa o valor do TTL
 - Se chega a 0 envia uma mensagem "TTL exceeded"
- O programa *traceroute* explora esta faceta



Exemplo

tracert to ucla.edu (128.97.27.37), 64 hops max, 52 byte packets

```
1 10.0.1.1 (10.0.1.1)  2.046 ms  1.879 ms  1.517 ms
2  dsldevice.lan (192.168.1.254)  5.250 ms  3.066 ms  1.972 ms
3  2.96.54.77.rev.vodafone.pt (77.54.96.2)  9.896 ms  9.579 ms  9.998 ms
4  21.93.30.213.rev.vodafone.pt (213.30.93.21)  9.525 ms  9.862 ms  9.677 ms
5  ae5-100-ucr1.lis.cw.net (195.10.57.9)  15.140 ms  12.979 ms  12.717 ms
6  ae5-xcr1.mal.cw.net (195.2.30.230)  23.257 ms  26.014 ms  23.604 ms
7  et-1-3-0-xcr2.prp.cw.net (195.2.24.189)  36.238 ms  38.611 ms  36.070 ms
8  lag-26.ear1.paris1.level3.net (212.73.242.237)  39.280 ms  39.765 ms  37.218 ms
9  * * *
10 * * *
11 cenic.ear1.losangeles1.level3.net (4.35.156.66)  179.252 ms  179.903 ms  179.225 ms
12 * * *
13 bd11f1.anderson--cr01f1.anderson.ucla.net (169.232.4.6)  181.289 ms
   bd11f1.anderson--cr01f2.csb1.ucla.net (169.232.4.4)  180.015 ms
   bd11f1.anderson--cr01f1.anderson.ucla.net (169.232.4.6)  180.748 ms
14 cr01f2.csb1--sr02f2.csb1.ucla.net (169.232.8.7)  179.717 ms
   cr01f1.anderson--sr02fb.jsei.ucla.net (169.232.8.53)  178.705 ms  180.013 ms
15 128.97.27.37 (128.97.27.37)  179.559 ms !Z 177.996 ms !Z 178.304 ms
```

Como Viver com o *Jitter*

- jitter |'jitər| informal

noun

- 1 (*jitters*) feelings of extreme nervousness: *a bout of the jitters*.
- 2 slight irregular movement, variation, or unsteadiness, esp. in an electrical signal or electronic device.

- **Aplicações elásticas**

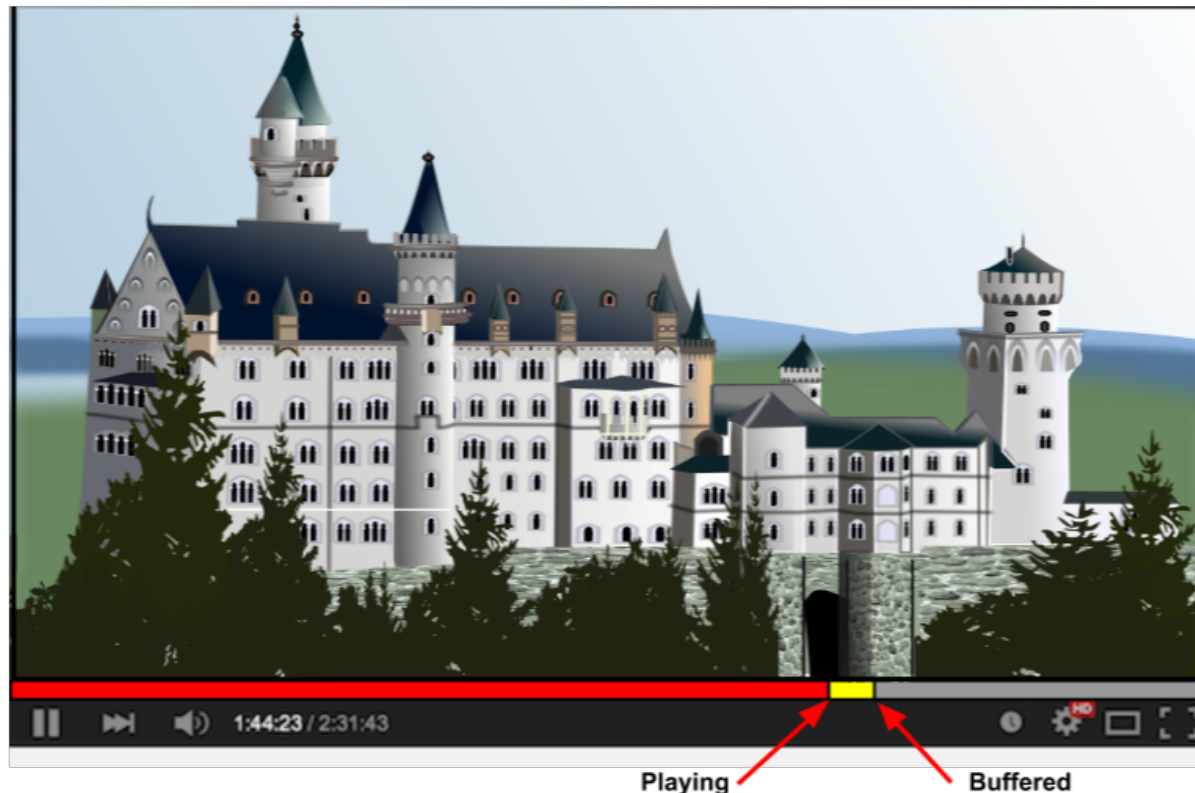
- Aplicações que podem funcionar corretamente com tempos de trânsito e capacidade extremo a extremo variáveis

- **Aplicações não elásticas**

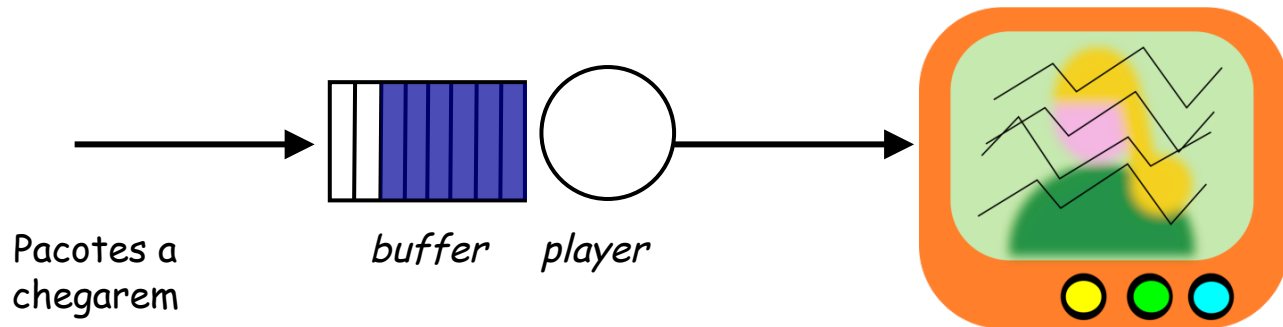
- As que requerem condições específicas para funcionarem corretamente

Exemplo

- As aplicações com som ou vídeo são não elásticas e usam *buffers* com dados de avanço

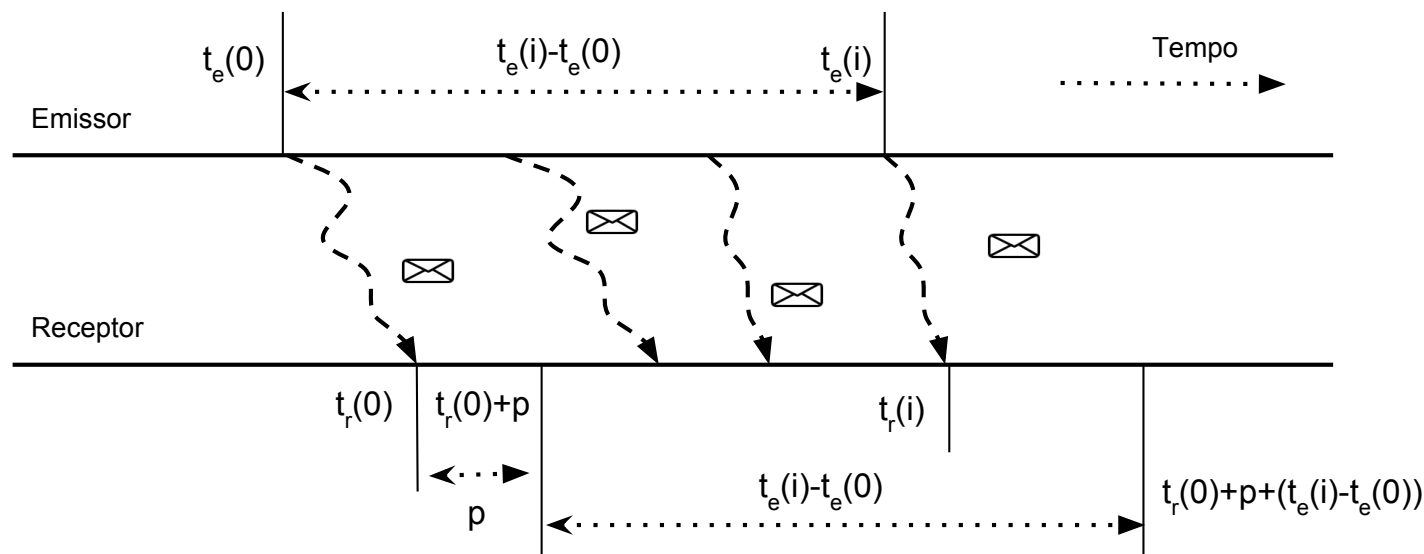


Compasso de Espera (Playout Delay)



- Os pacotes chegam com um ritmo variável ao *buffer*
- Mas o *player* processa-os ao seu ritmo — o tempo de permanência no *buffer* é variável o que acomoda o *jitter*
- Genericamente, ao intervalo de tempo que medeia entre a receção e a utilização da informação chama-se o **compasso de espera para processamento ou playout delay**

O Compasso de Espera é Variável



- O tempo de permanência no *buffer* é variável para acomodar o *jitter*
- De forma mais rigorosa: à diferença máxima entre a recepção e a utilização da informação chegada chama-se *playout delay* (p na figura)

Escolha do Valor do *Playout Delay*

- Quanto maior melhor ?
 - Nas aplicações não elásticas uni-direccionais (*progressive streaming*) um grande *playout delay* apenas provoca uma grande espera no início
 - Mas se for pequeno provoca *rebuffering* frequente
 - No limite poder-se-ia fazer o download integral do vídeo e só depois o visualizar
- O valor pode ser arbitrário ?
 - Nas aplicações interactivas (telefone ou conferência) valores superiores a algumas centenas de milissegundos (e.g. 150 ms) tornam-se perceptíveis e podem tornar a qualidade de serviço insustentável

Indicadores de Desempenho

- Tempo de trânsito extremo a extremo
 - tempo que medeia desde que um pacote foi emitido até que chega ao destino (i.e., é completamente recebido)
- Variabilidade do tempo de trânsito (*jitter*)
- Taxa de transferência extremo a extremo
 - débito médio da chegada de informação ao destino medido num dado intervalo de tempo (bits por segundo tal como o débito dos canais)
- Taxa de perda de pacotes extremo a extremo
 - valor médio do número de pacotes que não chegam ao destino, medido num dado intervalo de tempo
- Taxa de utilização de um canal
 - fração de tempo em que o canal está a transmitir

Tipos de Aplicações

- Aplicações elásticas

- Não são sensíveis ao tempo de trânsito e adaptam-se à disponibilidade da rede
- Geralmente usam TCP
- Exemplos: transferência de ficheiros, e-mail, ...
- No entanto, se forem interactivas, estão limitadas pela "paciência do utilizador"

- Aplicações NÃO elásticas

- São sensíveis ao tempo de trânsito e impõem limites superiores ao mesmo
- Algumas até sacrificam a fiabilidade ao tempo de trânsito (usam UDP ao invés de TCP)
- Exemplos: IPTV (televisão sobre pacotes IP), VoIP (Voz sobre IP), vídeo em geral, jogos, etc.

Conclusões Sobre o Jitter

- O tempo de trânsito extremo a extremo depende de vários fatores. Alguns são constantes
 - Tempo de transmissão
 - Tempo de propagação
- Outro é variável, nomeadamente
 - Tempo nas filas de espera ou *queueing time*
 - Tempo de processamento ou *processing time*
- As aplicações não elásticas usam técnicas especiais para lidar com o *jitter*
 - Nomeadamente *buffers* e *playout delays*

Conclusões

- Para que N sistemas comuniquem entre si usam-se
 - Nós de comutação
 - Agregam-se diversas comunicações para partilha dos canais (*multiplexing*)
- A comunicação de dados na Internet baseia-se em
 - transmissão e comutação de pacotes
 - *statistical multiplexing*
 - nós de comutação de pacotes que funcionam segundo o modo *store & forward*
- Mas a comutação de pacotes introduz
 - Tempo de trânsito extremo a extremo variável
 - No caso normal introduz igualmente perda eventual de pacotes e débito de extremo a extremo também variável