# REDES DE COMPUTADORES

# INTRODUÇÃO / REVISÕES

# PARTE 2

Departamento de Informática da Faculdade de Ciências e Tecnologia da UNL
2008 / 2009

# Nota prévia

A estrutura da apresentação é semelhante e utiliza algumas das figuras, textos e outros materiais do livro de base do curso

James F. Kurose and Keith W. Ross, "Computer Networking - A Top-Down Approach Featuring the Internet," Addison Wesley Longman, Inc., 4th Edition, 2007

# Objectivos do capítulo

- Perceber o que são as redes de computadores e ter uma primeira aproximação de como funcionam e de para que servem
- Ter uma ideia global da estrutura e componentes de uma rede de computadores, isto é, a visão do sistema
- Introduzir alguns conceitos centrais: comutação de pacotes, protocolos, camadas
- Introduzir alguns  aspectos essenciais do desempenho (performance)
- Falar de segurança

# Performance ou desempenho



- Quanto tempo leva uma mensagem a percorrer um canal?
- Quanto tempo leva uma mensagem a ir de um computador a outro através de uma rede ?
- Que tipo de necessidades têm as aplicações?
- De que forma a performance da rede influencia o desempenho das aplicações?

# Medidas e unidades



- Velocidade de transmissão
  - Dados (bits) transmitidos por unidade de tempo
  - Canal versus extremo a extremo
  - A velocidade de transmissão determina a duração de transmitir 1 bit no canal
  - Notação:
    - KB = Kbyte = $2^{10}$ bytes
    - Mbps = $10^6$ bits por segundo

# Continuação

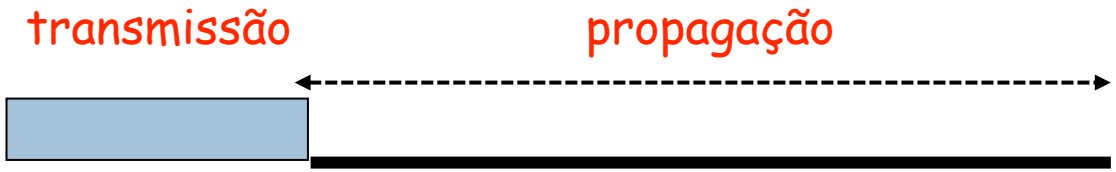


- Latência
  - Tempo necessário para transmitir uma mensagem de um extremo ao outro do canal ou de um computador a outro computador
  - Depende também do tempo de propagação
  - Podemos considerar o tempo num sentido só, ou o RTT
  - Componentes no caso de um canal isolado
    - Latência = Tempo de transmissão + Tempo de propagação
    - Tempo de propagação = Dimensão do canal / Velocidade de propagação
    - Tempo de Transmissão = Dimensão da mensagem / Velocidade de transmissão

# Exemplo



- Cada bit tem de "entrar" no canal (isso depende da velocidade de transmissão)
- Cada bit tem de chegar à outra extremidade do canal (isso depende da velocidade de propagação e da dimensão do canal)
- Exemplo: 1000 bits levam 11 ms a chegar através de um canal a 1 Mbps com 10 ms de tempo de propagação



# Velocidade de transmissão versus de propagação



- À medida que a velocidade de transmissão aumenta, o tempo de propagação (que é constante e imutável) ganha maior relevância

- As aplicações interactivas são cada vez mais sensíveis à distância (1000 bits levam 1 ms a transmitir a 1Mbps, 10 micro segundos a 100 Mbps, 1 micro segundo a 1 Gbps; que importância tem isso num canal com 30 ms de tempo de propagação ?)

# Velocidade de transmissão versus de propagação



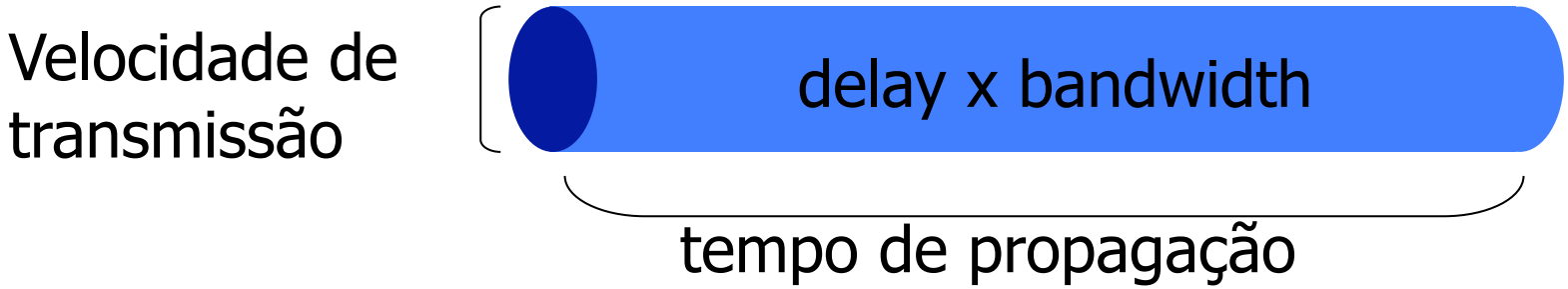
- Mas as aplicações de transferência de dados são sempre sensíveis à velocidade de transmissão

- 3 MB = 24 M bits, levam 240 ms a transmitir a 100 Mbps e 24 s a 1 Mbps; o facto de o tempo de propagação ser 10 ou 100 ms tem pouco ou nenhum peso na latência

# "Volume" = velocidade de transmissão x tempo propagação



- Um canal tem a capacidade de "conter" dados em transito ("*in flight*")
- Um canal a 100 Mbps, com um tempo de transito de 10 ms pode conter 1 M bit
- Para "encher" completamente o canal, o emissor precisa de ter pelo menos o seu "volume" em bits para transmitir



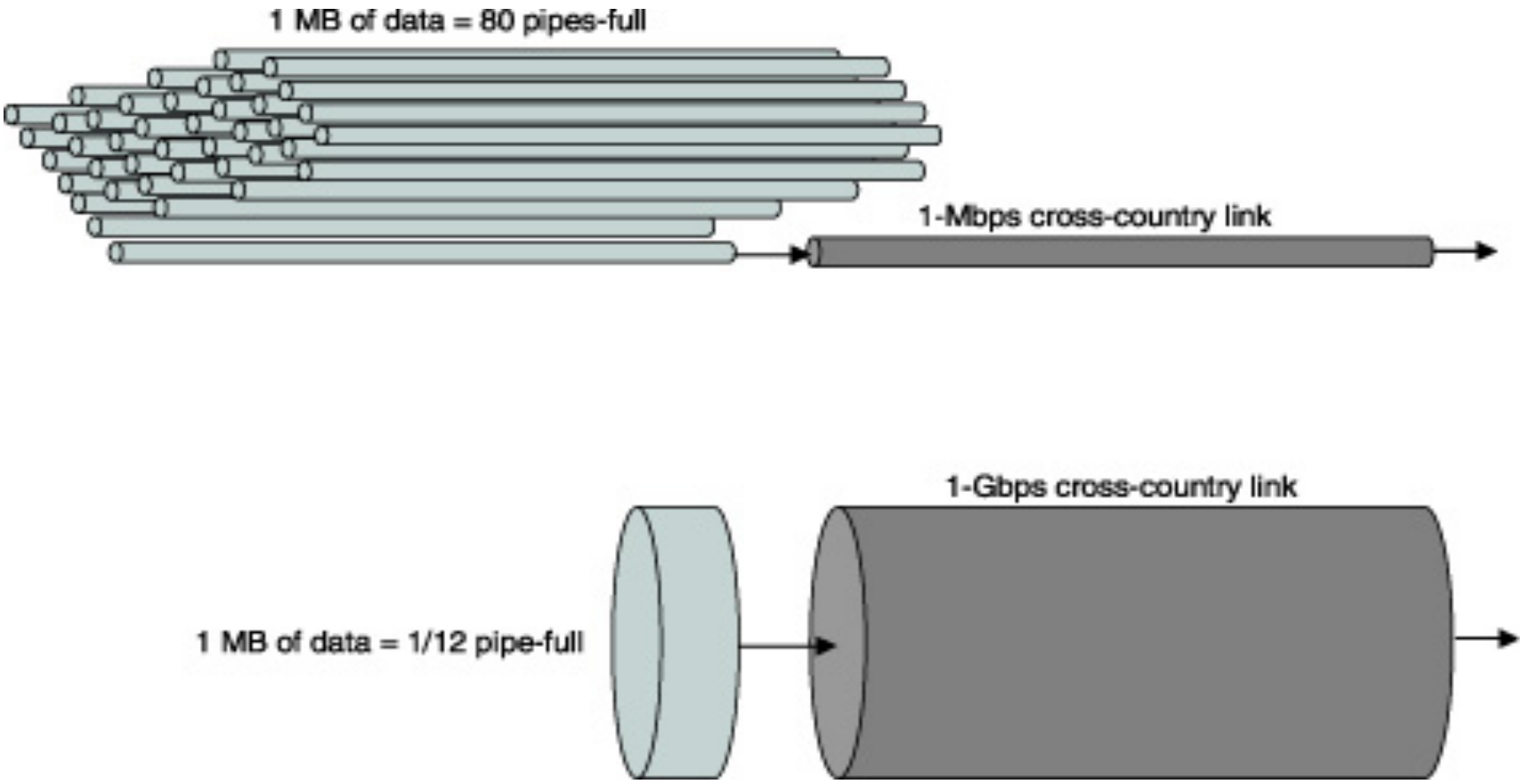
# Exemplo



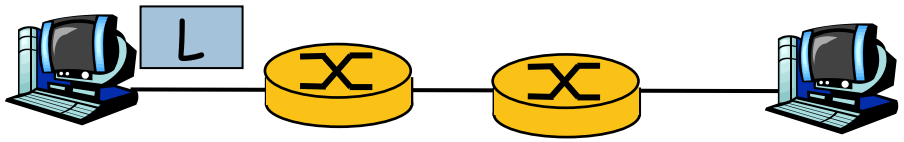
- Quando o emissor só pode passar à mensagem seguinte depois de receber autorização do receptor
- 1 K bit a transmitir a 1 Mbps leva 1 ms; se o canal tiver um RTT de 10 ms, em cada 11 ms o emissor só transmite durante 1 ms, isto é, durante1/11≈ 9% do tempo total
- Se o canal tiver a velocidade de 100 Mbps leva 0,01 ms a transmitir a mensagem, em cada 10,01 ms o emissor emite durante 0,01 ms, isto é, durante 1/1001 ≈ 0,1% do tempo total

# Efeito da alta velocidade





# Store and Forward (Memorizar e Retransmitir)




- Leva $D/V_t$ segundos a transmitir um pacote de D bits por um canal com a velocidade de transmissão de $V_t$ bps
- *store and forward:* o pacote inteiro tem de chegar ao router antes de poder ser transmitido para o próximo
- Tempo de chegada ao destino ou latência de extremo a extremo = $3(D/V_t)$ - se desprezarmos o tempo de propagação

Exemplo:

- D = 3 Mbits
- $V_t$ = 1.5 Mbps
- Latência de extremo a extremo = 6 segundos (admitindo que as filas de espera estão sempre vazias e desprezando o tempo de propagação)

# Latência de extremo a extremo na Internet

```
$ ping  google.com
PING google.com (72.14.207.99) 56(84) bytes of data.
64 bytes from 72.14.207.99: icmp_seq=1 ttl=234 time=229 ms
64 bytes from 72.14.207.99: icmp_seq=2 ttl=234 time=225 ms
64 bytes from 72.14.207.99: icmp_seq=3 ttl=234 time=263 ms
64 bytes from 72.14.207.99: icmp_seq=4 ttl=234 time=271 ms
64 bytes from 72.14.207.99: icmp_seq=5 ttl=234 time=247 ms
64 bytes from 72.14.207.99: icmp_seq=6 ttl=234 time=304 ms
64 bytes from 72.14.207.99: icmp_seq=7 ttl=234 time=232 ms

--- google.com ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6005ms
rtt min/avg/max/mdev = 225.494 / 253.414 / 304.605 / 26.461 ms
```

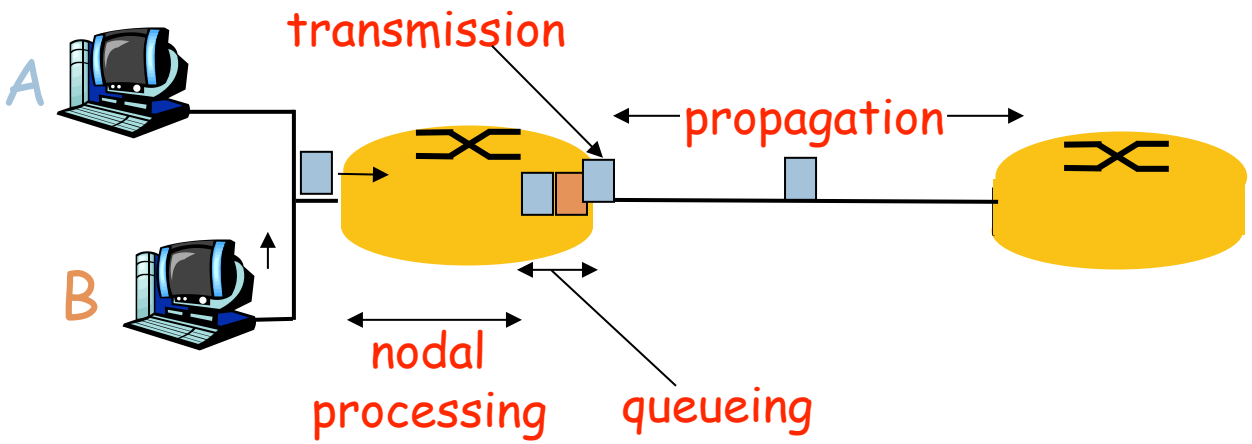
## A mesma medida tomada noutro ponto da Internet

```
$ ping google.com
PING google.com (72.14.207.99): 56 data bytes
64 bytes from 72.14.207.99: icmp_seq=0 ttl=239 time=158.811 ms
64 bytes from 72.14.207.99: icmp_seq=1 ttl=239 time=162.902 ms
64 bytes from 72.14.207.99: icmp_seq=2 ttl=239 time=159.659 ms
64 bytes from 72.14.207.99: icmp_seq=3 ttl=239 time=158.153 ms
64 bytes from 72.14.207.99: icmp_seq=4 ttl=239 time=156.678 ms
64 bytes from 72.14.207.99: icmp_seq=5 ttl=239 time=157.992 ms
64 bytes from 72.14.207.99: icmp_seq=6 ttl=239 time=161.316 ms
^C
--- google.com ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max/stddev = 156.678 / 159.359 / 162.902 / 1.974 ms
```

# Tempo de encaminhamento dos pacotes

- Os pacotes são entregues ao destino após um certo período de tempo com origem em diversas formas de atraso no processamento dos mesmos
- Em cada nó (*host* ou *router*) seguido de um canal os pacotes sofrem atrasos por diversas razões:
  - Tempo de processamento (análise do cabeçalho, ...)
  - Período na fila de espera
  - Tempo de transmissão
  - Tempo de propagação

# Origens do tempo de trânsito



# Quantificação dos tempos

- **Tempo de processamento**. Geralmente é constante e quase sempre desprezável salvo em cenários especiais
- **Tempo de transmissão**. Depende da velocidade de transmissão do canal por onde o pacote é transmitido. É dado pela equação:
  - $T_t$ = dimensão do pacote (bits) / velocidade de transmissão (bps), ou $T_t$ = Dp / $V_t$
- **Tempo de propagação**. Depende da dimensão do canal. É dado pela equação:
  - $T_p$ = dimensão do canal ( Kms) / velocidade de propagação do sinal (Km / s) ou $T_p = D_c / V_p$
  - $V_p$ é da ordem de grandeza de 200.000 (canais guiados = fios) a 330.000 Km / segundo (velocidade de propagação da luz no vácuo)

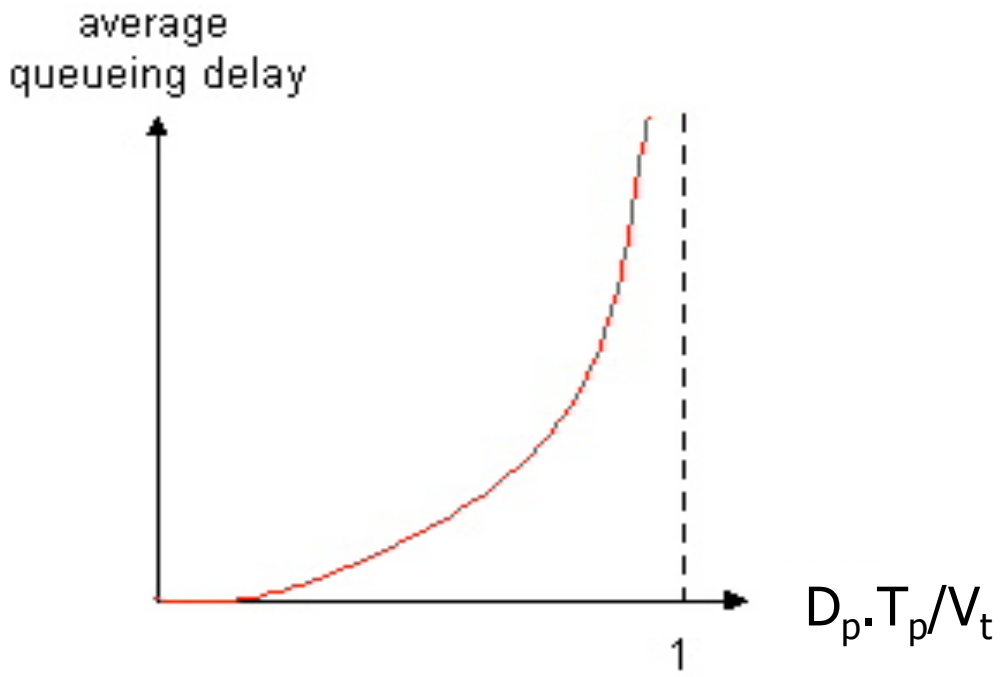
# Filas de espera (*queueing delay*)

- $V_t$ = Velocidade de transmissão (bps)
- $D_p$ = Dimensão média do pacote (bits)
- $T_p$ = taxa (ritmo) médio de chegada de pacotes

Intensidade de tráfego = $D_p$ . $T_p$ / $V_t$

- $D_p.T_p/V_t \sim 0$: em média o queueing delay é baixo
- $D_p.T_p/V_t \rightarrow 1$: torna-se significativo
- $D_p.T_p/V_t > 1$: está a chegar mais trabalho do que aquele que pode ser servido, o tempo de espera tende para infinito

# *Queueing delay*




Intensidade de tráfego = $D_p.T_p/V_t$

# A mesma medida tomada noutro ponto da Internet

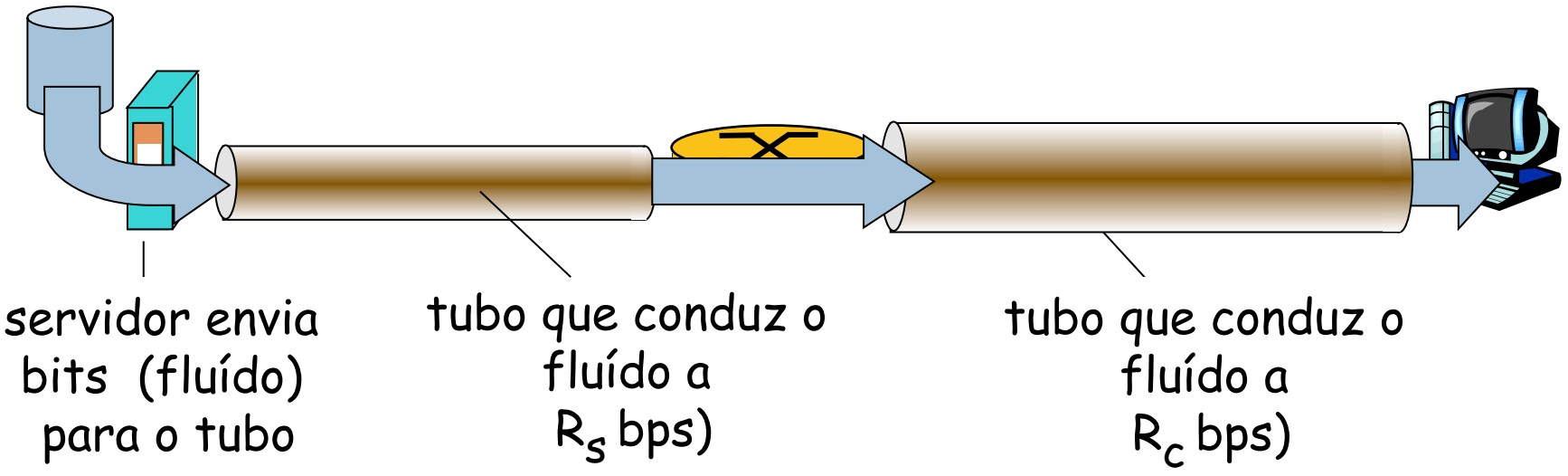
```
$ ping google.com
PING google.com (72.14.207.99): 56 data bytes
64 bytes from 72.14.207.99: icmp_seq=0 ttl=239 time=158.811 ms
64 bytes from 72.14.207.99: icmp_seq=1 ttl=239 time=162.902 ms
64 bytes from 72.14.207.99: icmp_seq=2 ttl=239 time=159.659 ms
64 bytes from 72.14.207.99: icmp_seq=3 ttl=239 time=158.153 ms
64 bytes from 72.14.207.99: icmp_seq=4 ttl=239 time=156.678 ms
64 bytes from 72.14.207.99: icmp_seq=5 ttl=239 time=157.992 ms
64 bytes from 72.14.207.99: icmp_seq=6 ttl=239 time=161.316 ms
^C
--- google.com ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max/stddev = 156.678 / 159.359 / 162.902 / 1.974 ms
```

# Perca de pacotes (*packet loss*)

- Se a intensidade do tráfego é tal que as filas de espera enchem, o router descarta pacotes (estes pacotes perdem-se)
- Fila de espera demasiado grandes podem ser contraproducentes pois introduzem uma latência de extremo a extremo enorme, o que "confunde" o emissor
- Também os erros dos canais poderão conduzir à perca de pacotes
- Os pacotes perdidos têm de ser transmitidos de novo

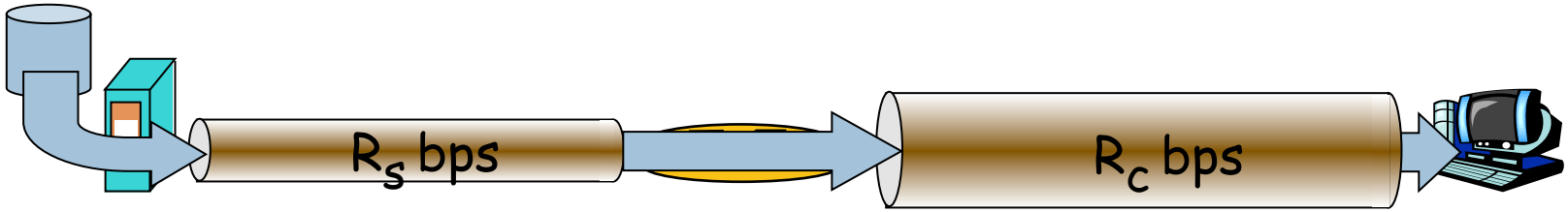
## Velocidade média (taxa) de transmissão extremo a extremo

- *throughput:* velocidade média de transmissão de extremo a extremo, taxa ou ritmo médio (bits/unidade de tempo) com que os bits são transferidos entre o emissor e o receptor
  - *Instantâneo*: num dado momento
  - *Médio:* medido num intervalo

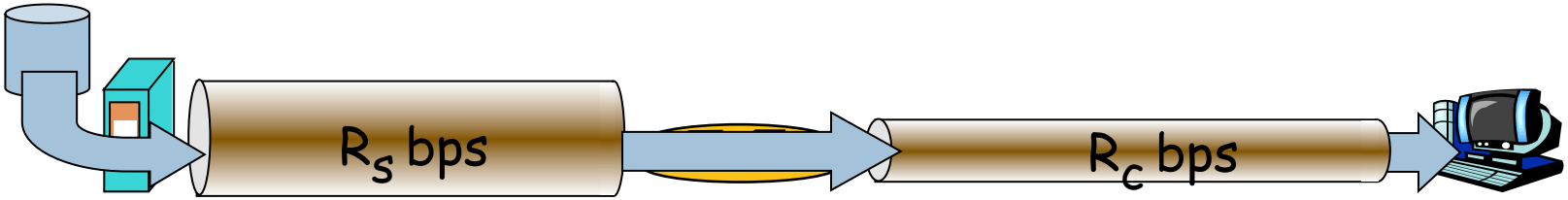


# Continuação

$R_s < R_c$ Qual é o ritmo médio extremo a extremo?




$R_s > R_c$ Qual é o ritmo médio extremo a extremo?

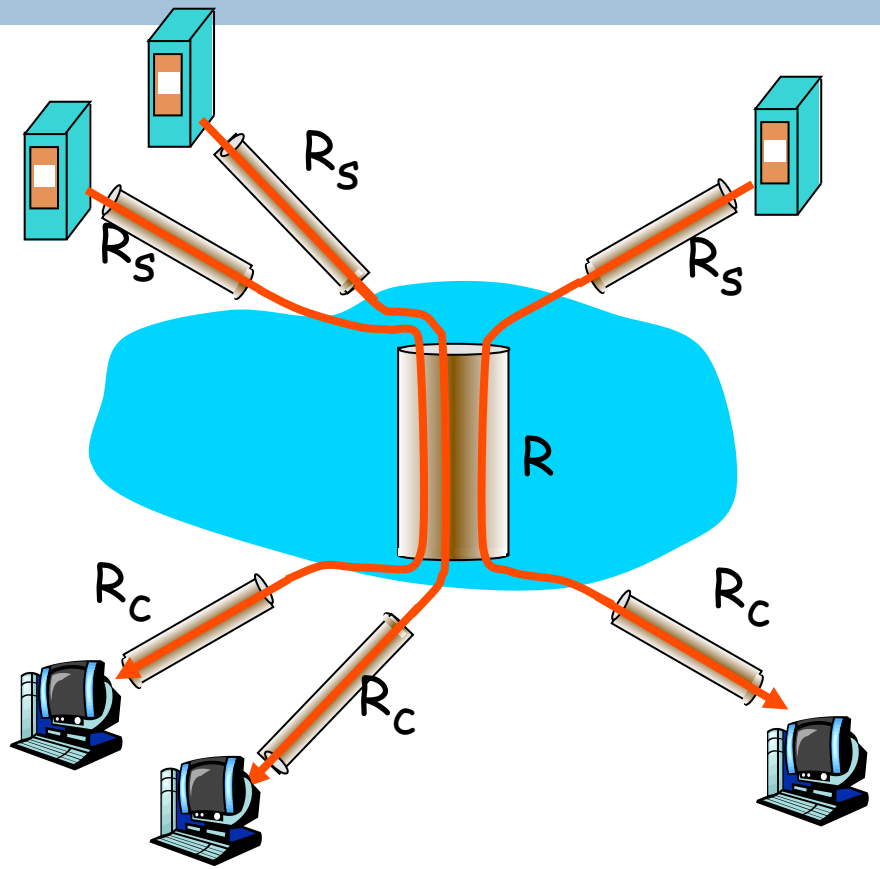



*bottleneck link ou canal gargalo*

Canal que condiciona de facto a velocidade de transmissão

# Que se passa de facto na Internet ?

10 conexões partilham equitativamente a rede com um canal bottleneck (gargalo) a R bps

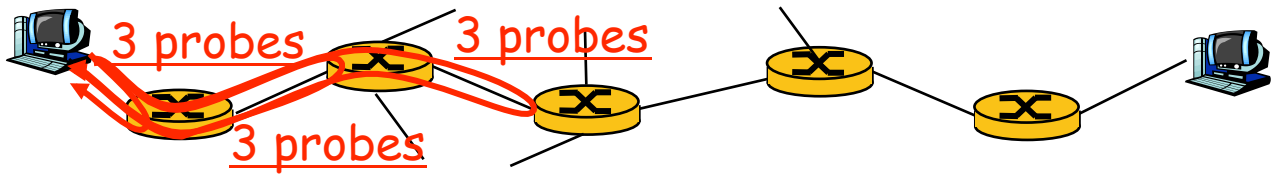



- Cada conexão obtêm: $\min(R_c, R_s, R/10)$
- Na prática: $R_c$ ou $R_s$ são o *bottleneck* (estrangulamento)

# Tempos e rotas reais na Internet



- O programa ping permite medir a latência de extremo a extremo (RTT). Como ver também as rotas?
- **Traceroute** mede a latência (RTT) entre a origem e cada ponto intermédio de uma rota. For all *i*:
  - sends three packets that will reach router *i* on path towards destination
  - router *i* will return packets to sender
  - sender times interval between transmission and reply.



# Exemplo



Traceroute / tracert : gaia.cs.umass.edu to www.eurecom.fr

Three delay measurements from gaia.cs.umass.edu to cs-gw.cs.umass.edu

```
1  cs-gw (128.119.240.254)  1 ms  1 ms  2 ms
2  border1-rt-fa5-1-0.gw.umass.edu (128.119.3.145)  1 ms  1 ms  2 ms
3  cht-vbns.gw.umass.edu (128.119.3.130)  6 ms 5 ms 5 ms
4  jn1-at1-0-0-19.wor.vbns.net (204.147.132.129)  16 ms 11 ms 13 ms
5  jn1-so7-0-0-0.wae.vbns.net (204.147.136.136)  21 ms 18 ms 18 ms
6  abilene-vbns.abilene.ucaid.edu (198.32.11.9)  22 ms  18 ms  22 ms
7  nycm-wash.abilene.ucaid.edu (198.32.8.46)  22 ms  22 ms  22 ms
8  62.40.103.253 (62.40.103.253)  104 ms 109 ms 106 ms
9  de2-1.de1.de.geant.net (62.40.96.129)  109 ms 102 ms 104 ms
10  de.fr1.fr.geant.net (62.40.96.50)  113 ms 121 ms 114 ms
11  renater-gw.fr1.fr.geant.net (62.40.103.54)  112 ms  114 ms  112 ms
12  nio-n2.cssi.renater.fr (193.51.206.13)  111 ms  114 ms  116 ms
13  nice.cssi.renater.fr (195.220.98.102)  123 ms  125 ms  124 ms
14  r3t2-nice.cssi.renater.fr (195.220.98.110)  126 ms  126 ms  124 ms
15  eurecom-valbonne.r3t2.ft.net (193.48.50.54)  135 ms  128 ms  133 ms
16  194.214.211.25 (194.214.211.25)  126 ms  128 ms  126 ms
17  * * *
18  * * *
19  fantasia.eurecom.fr (193.55.113.142)  132 ms  128 ms  136  ms
```

trans-oceanic link

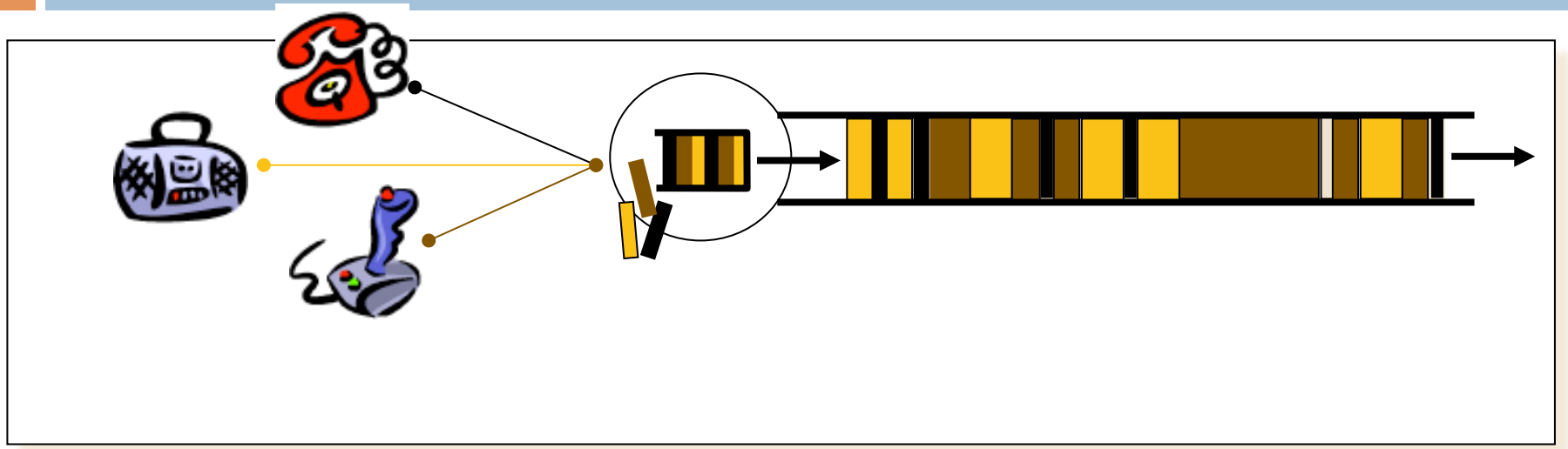
* means no response (probe lost, router not replying)

# Variação do tempo de transito ("Jitter")



- Se se pretende enviar vídeo codificado a 4 Mbps (em média), tanto faz que o canal seja de 6 ou de 100 Mbps
- No entanto, um canal de 4 Mbps dará problemas
- Por outro lado se o tempo de transito variar muito de pacote para pacote, os problemas serão ainda maiores pois os pacotes atrasados são equivalentes a pacotes perdidos

# Gestão de recursos na rede: filas de espera



- Dentro da rede, um canal é um recurso limitado partilhado entre diversos fluxos
- Caso mais simples: first-in-first-out queue (FIFO)
  - Serve os pacotes pela ordem por que chegam
  - Quando o canal está ocupado, os pacotes vão para a fila de espera
  - Quando a fila está cheia, os pacotes são suprimidos
- Mais sofisticado: dar prioridade a certo tipo de pacotes

# Como se lida com os pacotes perdidos?

Problem: Lost Data

GET index.html

Internet

Solution: Timeout and Retransmit

GET index.html

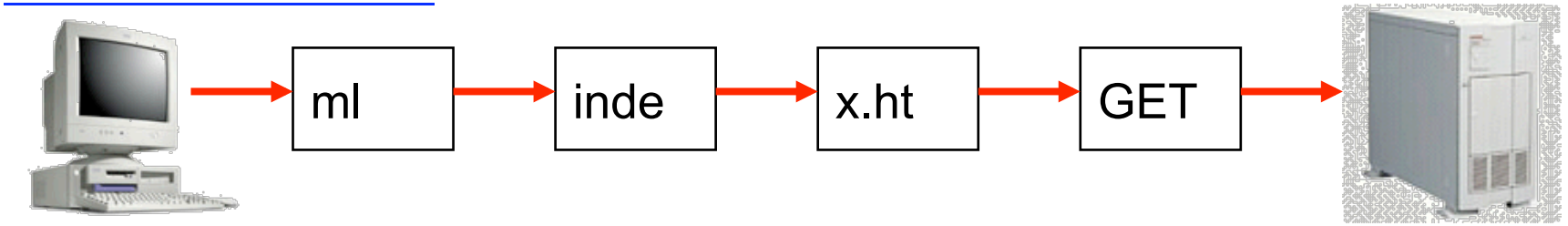
GET index.html

Internet

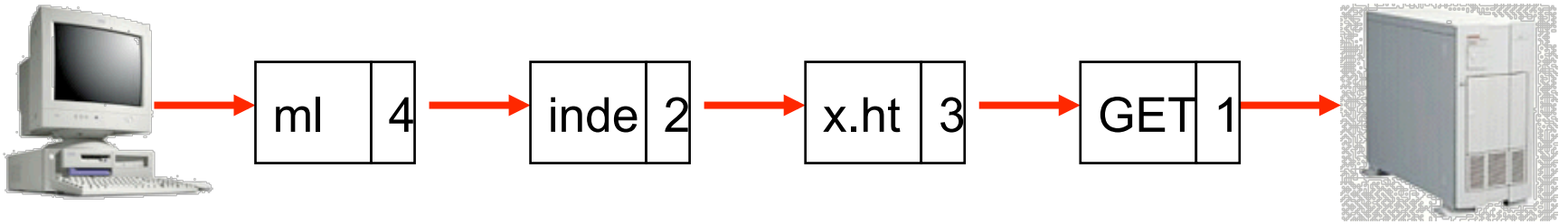
GET index.html

# Como se lida com dados fora de ordem?

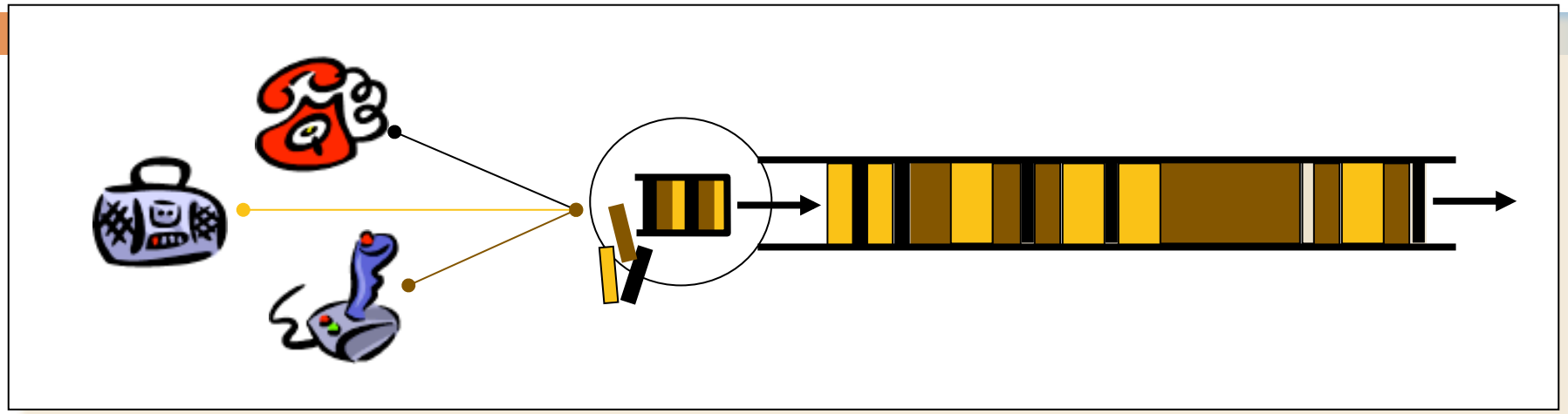
Problem: Out of Order




Solution: Add Sequence Numbers



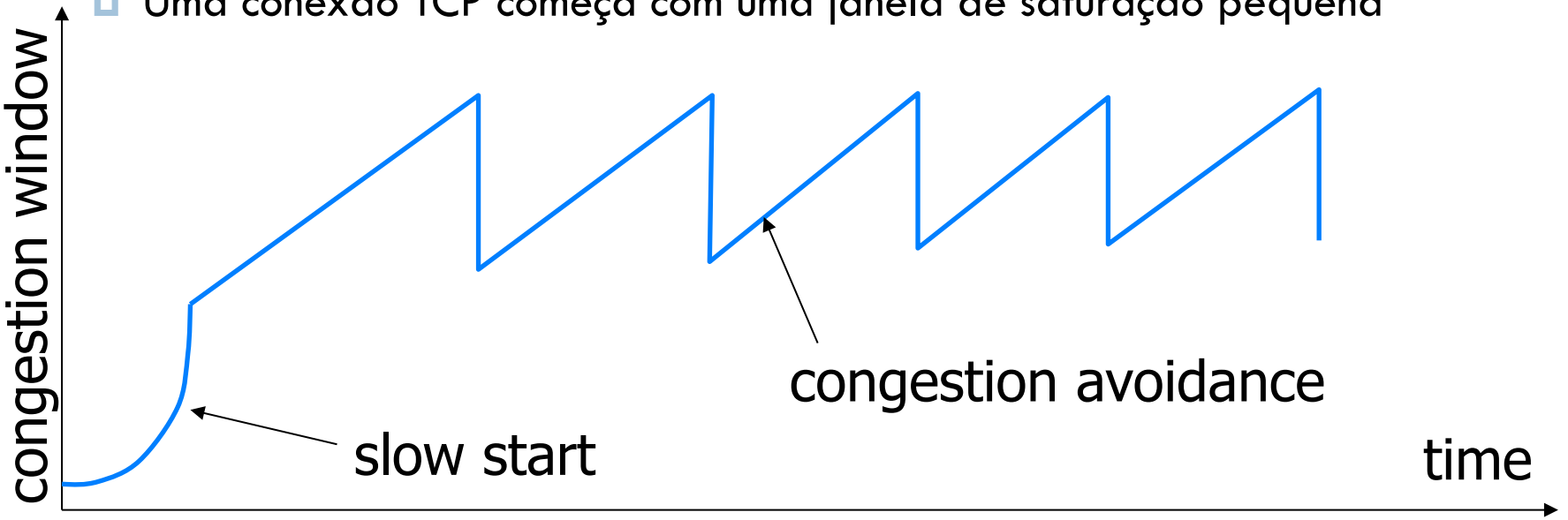
# Gestão de recursos na rede: controlo da saturação



- Que acontece se muitos utilizadores estão a usar a rede?
  - Os emissores devem diminuir o ritmo de emissão
  - … em resposta aos seus pacotes serem suprimidos
- Esta é a essência do controlo de saturação do protocolo TCP
  - A qual é fundamental para evitar o colapso da Internet
- Mas que não chega para lidar com fluxos de vídeo

# TCP - Transmission Control Protocol

- Controlo de fluxos baseado numa janela
  - O emissor limita o n.º de bytes "in flight" (dimensão da janela) à capacidade do receptor
- Controlo de saturação: reacção à perca de pacotes
  - *Janela de saturação* usada para tentar não saturar a rede (aumenta se os pacotes chegam, decresce no caso contrário)
  - Uma conexão TCP começa com uma janela de saturação pequena



# Segurança da rede

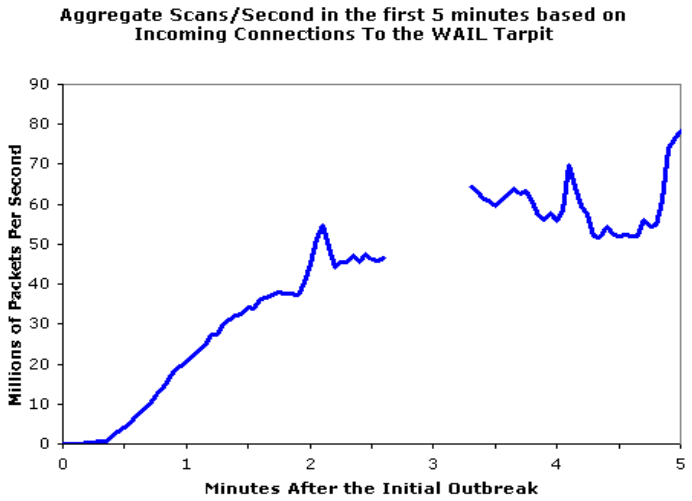
- **Internet não foi concebida com a segurança no topo das prioridades**
  - *Visão original:* "um grupo de de pessoas de confiança usam uma uma rede de investigação para um fim comum" ☺
  - Os protocolos Internet foram desenhados em grande parte em resposta às necessidades que apareceram. Inicialmente não havia uma ideia clara do que seria a rede no futuro.
  - Existem problemas de segurança em todos os níveis!

- **São possíveis ataques à periferia e ao centro:**
  - Ataques a hosts: malware, spyware, worms, acesso não autorizado (user accounts, roubo de dados)
  - Negação de serviços: impedir o acesso a recursos (servidores, capacidade de comunicação)

# Ataques através de programas: malware

- **Spyware:**
  - Infecção através do download de uma página web com spyware
  - Memoriza as teclas carregadas, os sites visitados, envia o resultado para o atacante
- **Vírus e Worms ("Minhocas"):**
  - A infecção progride através de um objecto com código que é recebido por um host e é executado imediatamente ou mais tarde e auto-propaga-se para outros hosts

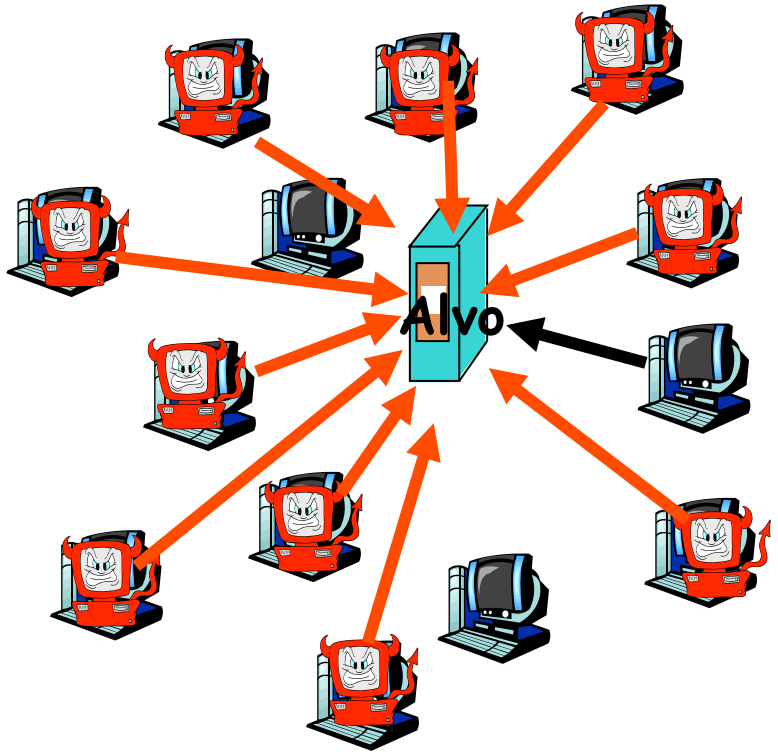
Sapphire Worm: aggregate scans/sec
in first 5 minutes of outbreak (CAIDA, UWisc data)



# Ataques de negação de serviço

- O atacante torna os recursos inacessíveis através de uma saturação artificial dos mesmos

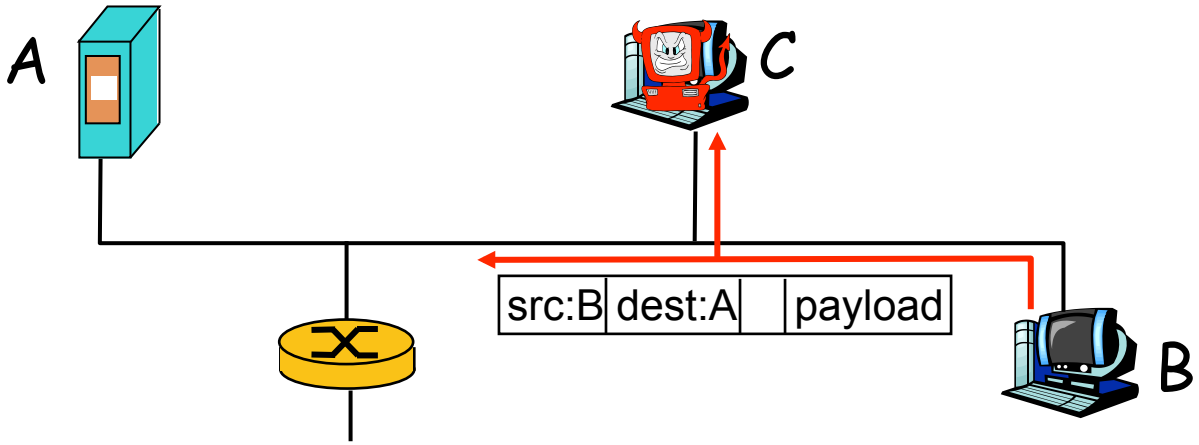
1) Contamina tantos hosts quanto puder (malware)
2) Selecciona o alvo
3) Envia pacotes para o alvo (existem várias formas e fazer um ataque deste tipo)



# Sniffing (cheirar), modificar, suprimir pacotes

***Packet sniffing:***

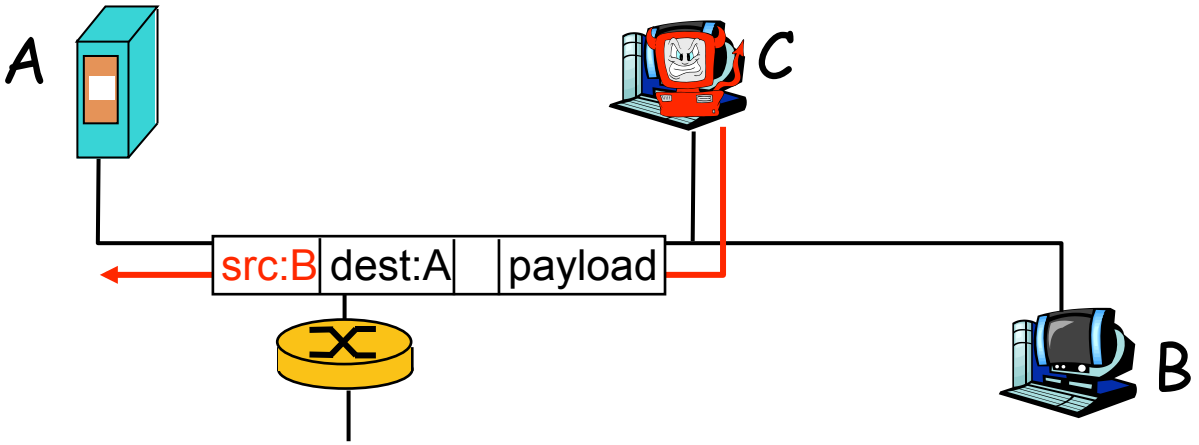
- Quando um canal é multiponto (shared Ethernet, wireless)
- Uma interface promíscua pode ler os pacotes que atravessam o canal e gravar os seus dados (incluindo as passwords!)




- Ethereal é um exemplo de software de packet-sniffer

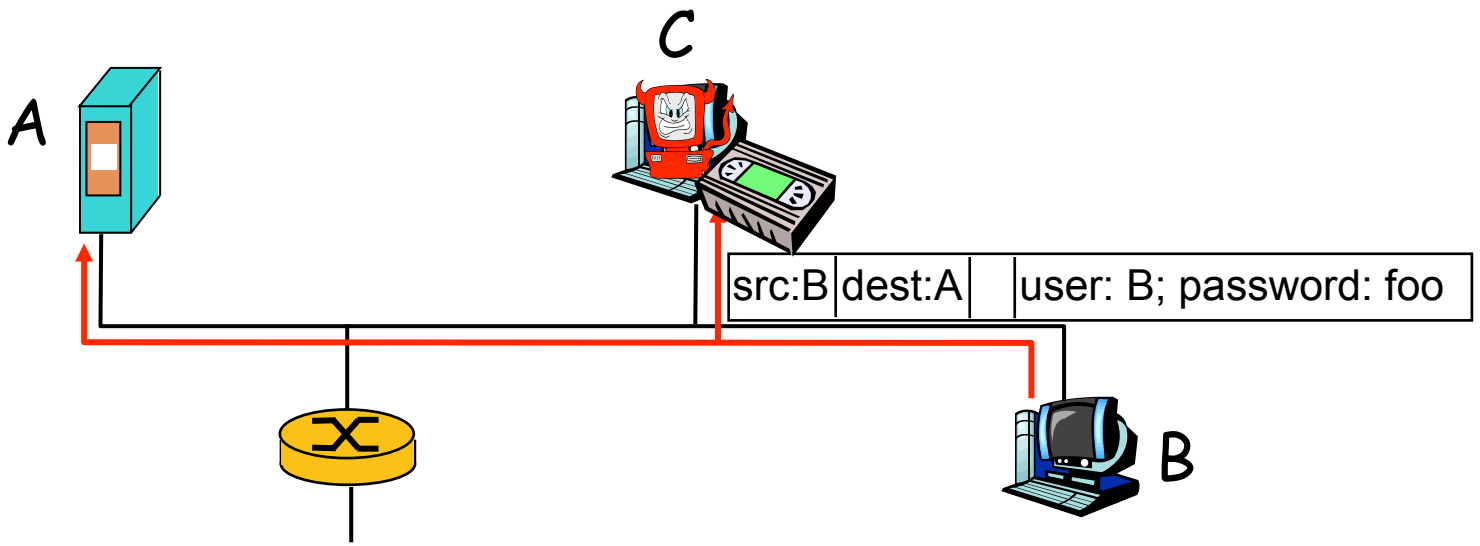
# Disfarçar-se ou mascarar-se

- ***IP spoofing:*** enviar um pacote com um endereço origem falso



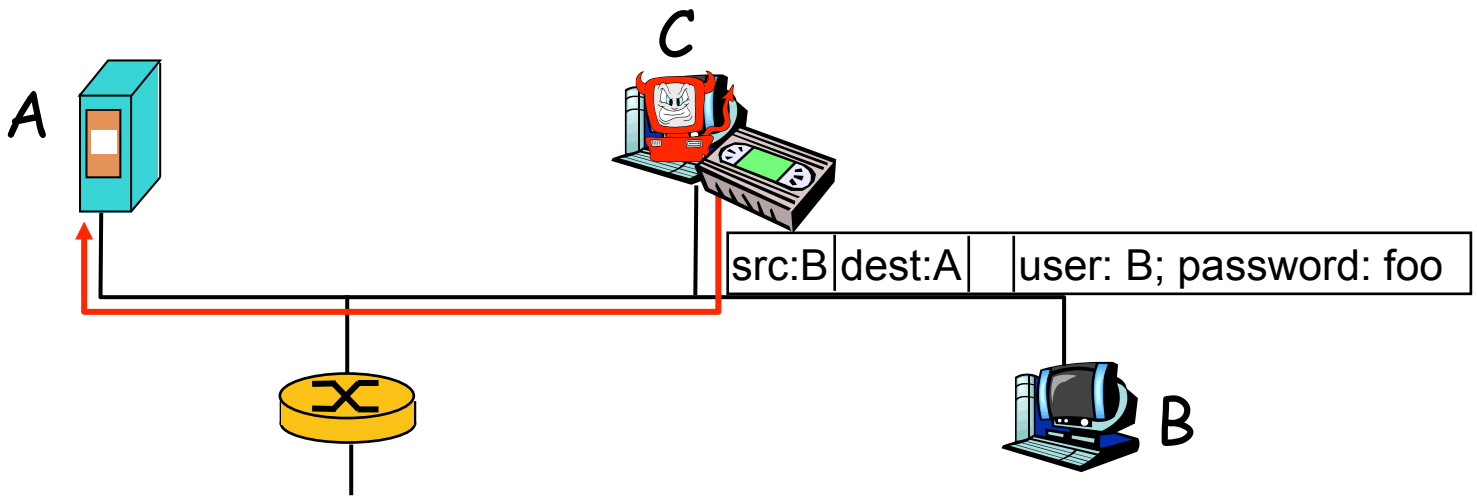
# Mais sofisticado ainda

- ***Record-and-playback*: memorizar a informação crítica (por exemplo a password), e reutilizá-la mais tarde**
  - Do ponto do vista do sistema o utilizador é quem se identifica e conhece a password
  - As passwords também podem ser obtidas por (**phishing**)



# Continuação

*Mais tarde .....*



# O que vimos neste capítulo

- Percebemos o que são as redes de computadores e temos uma primeira aproximação de como funcionam e de para que servem
- Temos uma ideia global da estrutura e componentes de uma rede de computadores
- Discutimos alguns conceitos centrais: comutação de pacotes, protocolos, camadas
- Discutimos alguns aspectos essenciais do desempenho (performance)
- Falámos de segurança