REDES DE COMPUTADORES

INTRODUÇÃO / REVISÕES

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

O que se aprende nesta disciplina

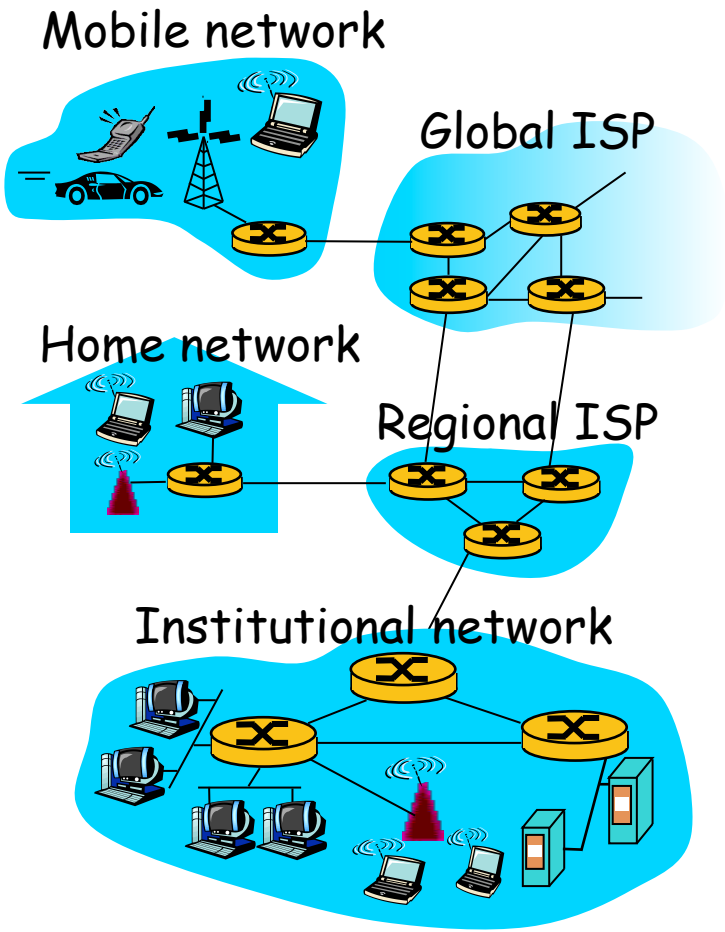
- Saber: Como funciona a Internet
  - Fundamentos das redes de computadores
  - A arquitectura da Internet e os protocolos TCP/IP
  - Aplicações (Web, e-mail, P2P, VoIP, …)
- Fundamentos: conceitos chave em redes
  - Protocolos e algoritmos
  - Camadas, componentes e visão sistema
  - Estruturação das aplicações distribuídas
  - Desempenho e gestão de recursos
  - Designação
- Fazer: programação de aplicações em rede
  - Programação com *sockets*
  - Desenhar e implementar aplicações
  - Analisar e implementar protocolos

Objectivos do capítulo

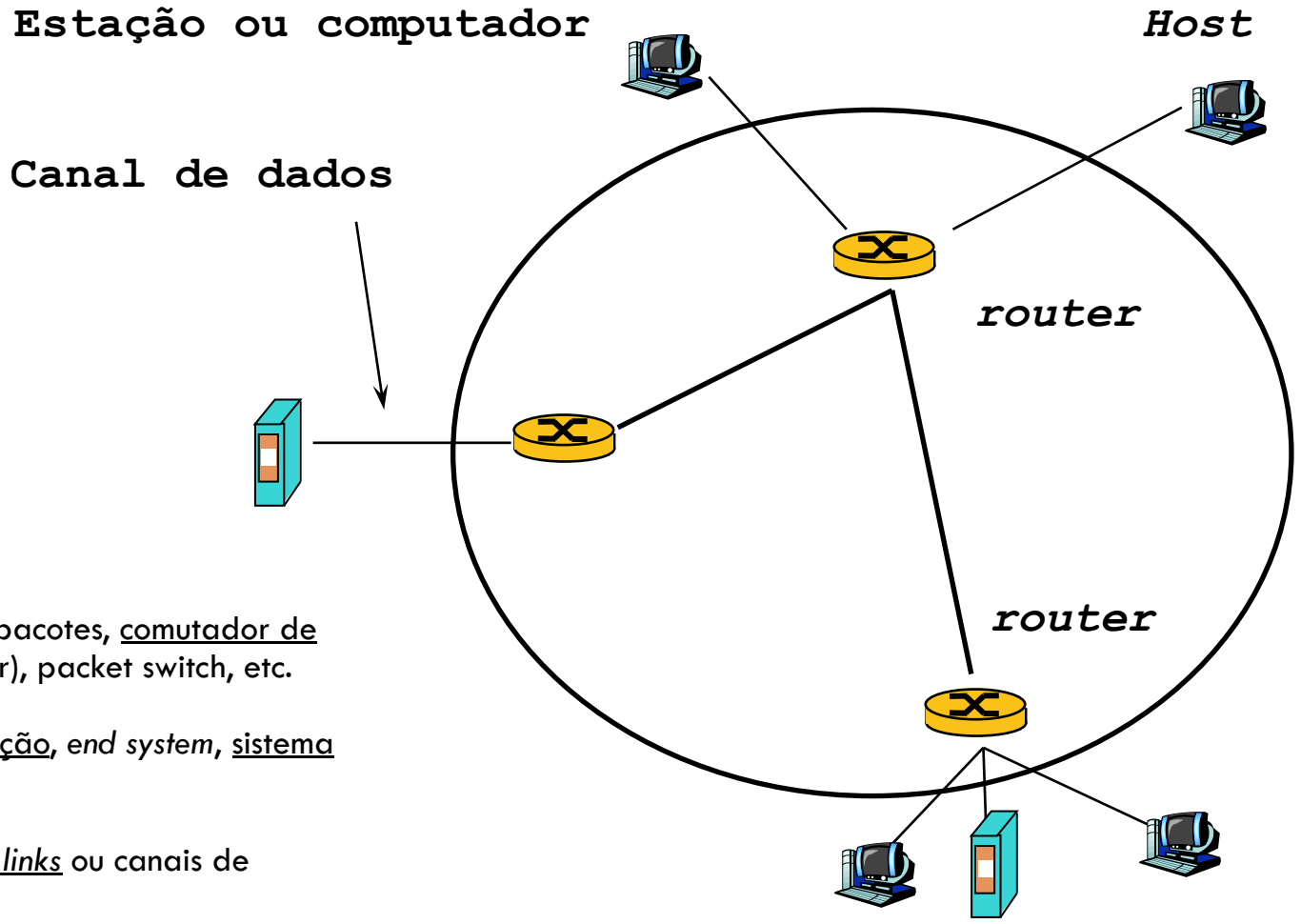
- Perceber o que são as redes de computadores e ter uma primeira aproximação de como funcionam e de para que servem
- Ter uma ideia global da estrutura e componentes de uma rede de computadores, isto é, a visão do sistema
- Introduzir alguns conceitos centrais: comutação de pacotes, protocolos, camadas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

O que é a Internet: *building blocks*

- Milhões de computadores interligados:
  - pc's *workstations*, servidores, PDA's
- Canais de comunicação
  - Baseados em fibra, cobre, rádio, satélite
- Routers que encaminham pacotes (pedaços de informação ou mensagens) de dados através da rede



Terminologia



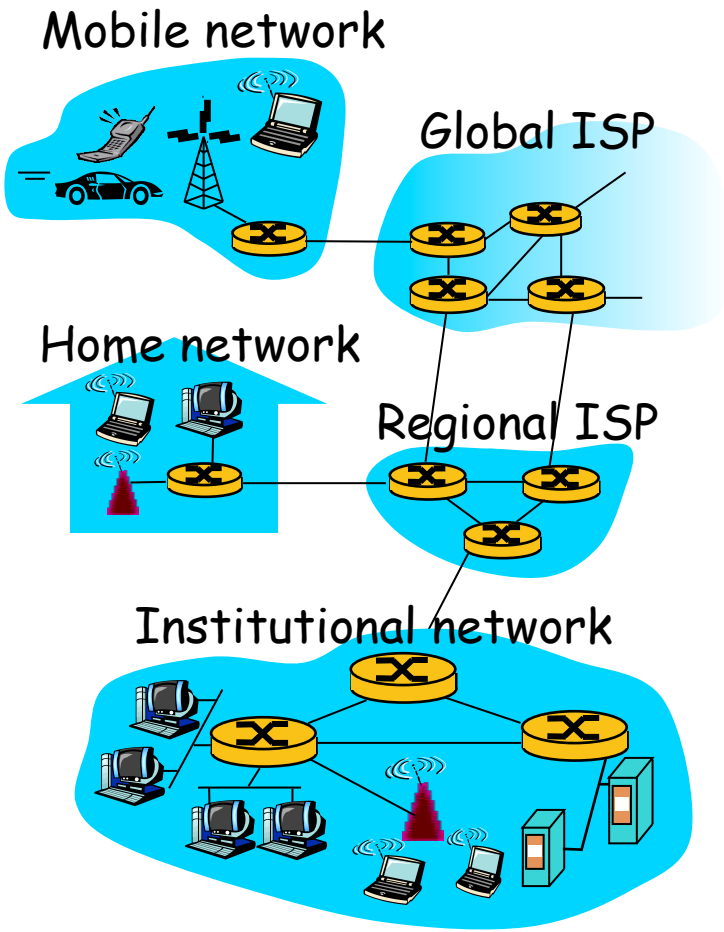
Router, encaminhador de pacotes, comutador de pacotes, *switch* (comutador), packet switch, etc.

Site, *host*, *worksation*, estação, *end system*, sistema terminal, nó, computador

Communication channels / links ou canais de comunicação

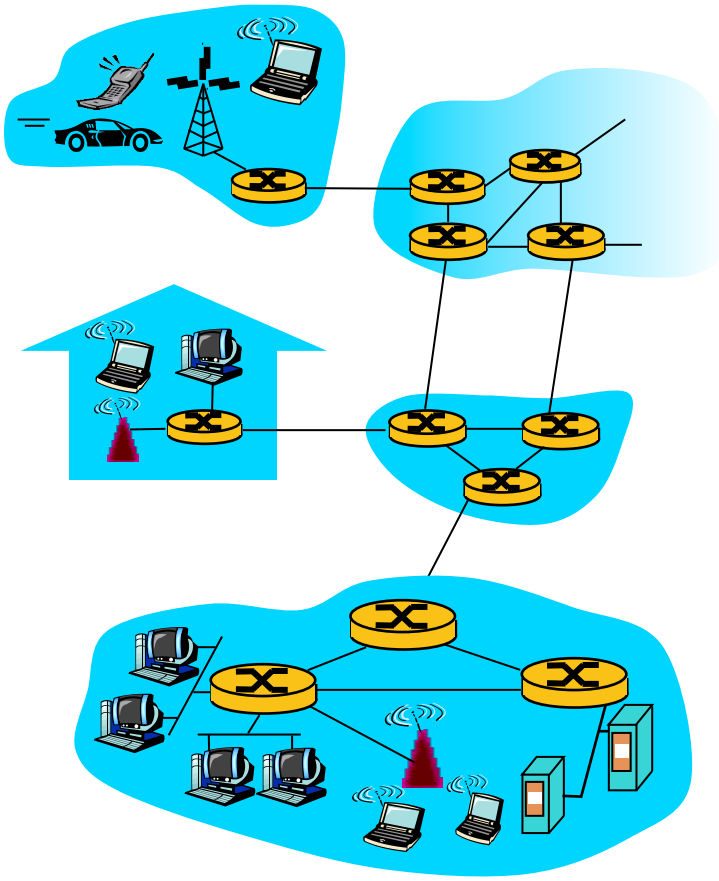
A Internet é uma rede de redes

- *A Internet:* é definida recursivamente
  - Vagamente hierarquizada
  - Composta por muitas subredes
  - Existem também "internets privadas"



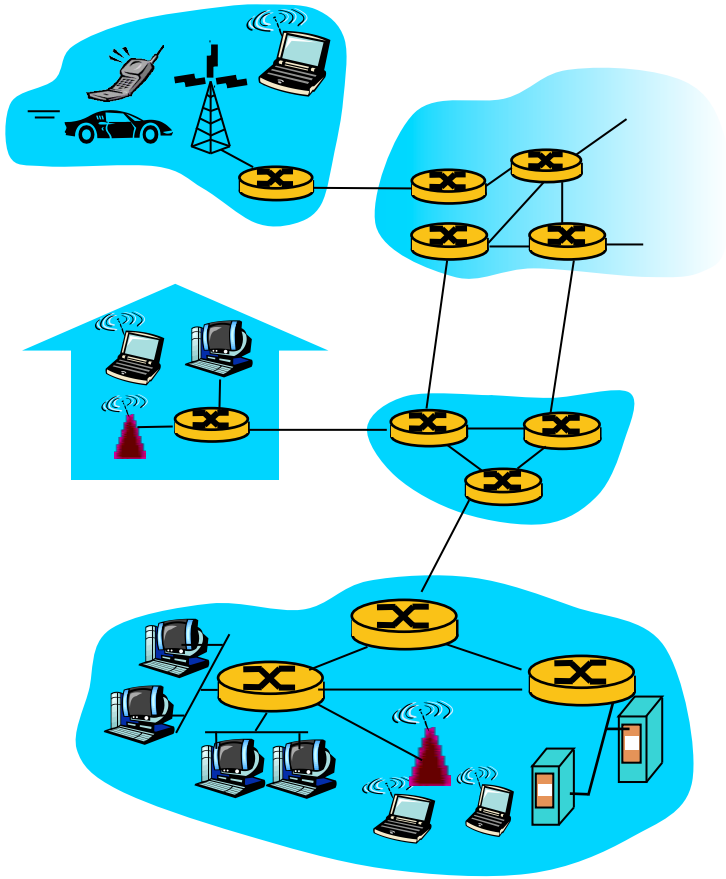
Serviços prestados pela rede

- Serviços de comunicações suportado pelas aplicações
  - Web, VoIP, email, games, e-commerce, file sharing
- Serviços de comunicação providenciados às aplicações
  - Transferência fiável de dados
  - Entrega “best effort” (unreliable) de dados



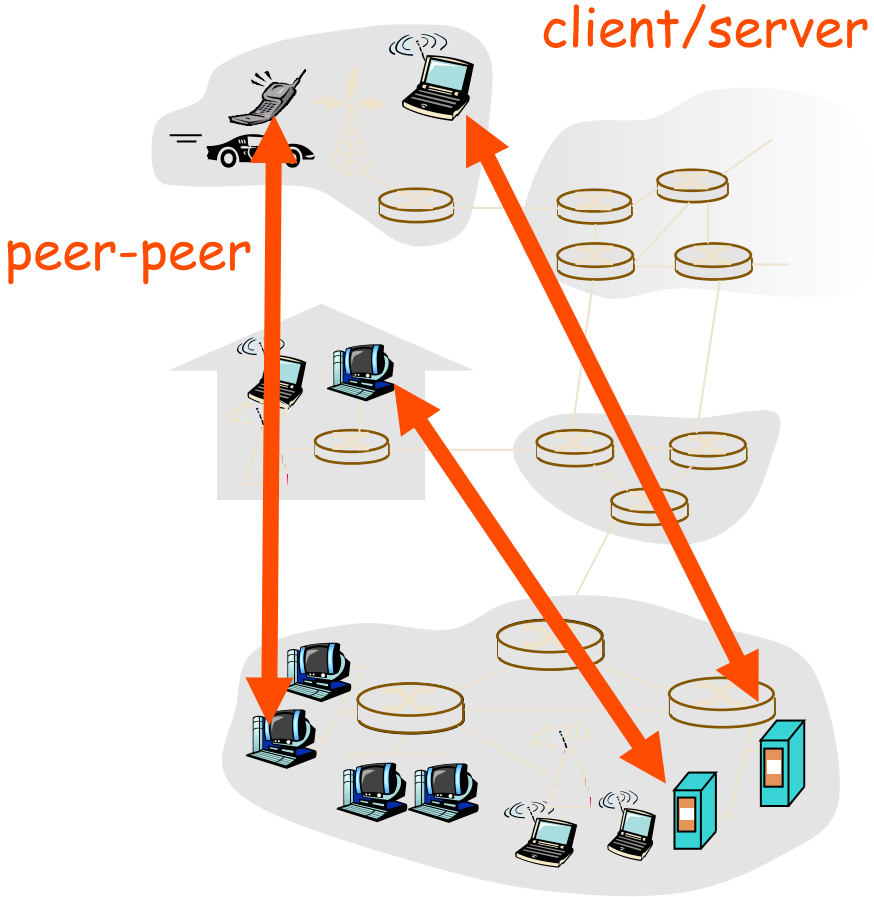
Distribuição de funcionalidades na rede

- A periferia da rede: computadores e aplicações
- O "core" da rede:
  - *Routers*
  - Canais de comunicação
- Redes de acesso: canais de comunicação



A periferia da rede

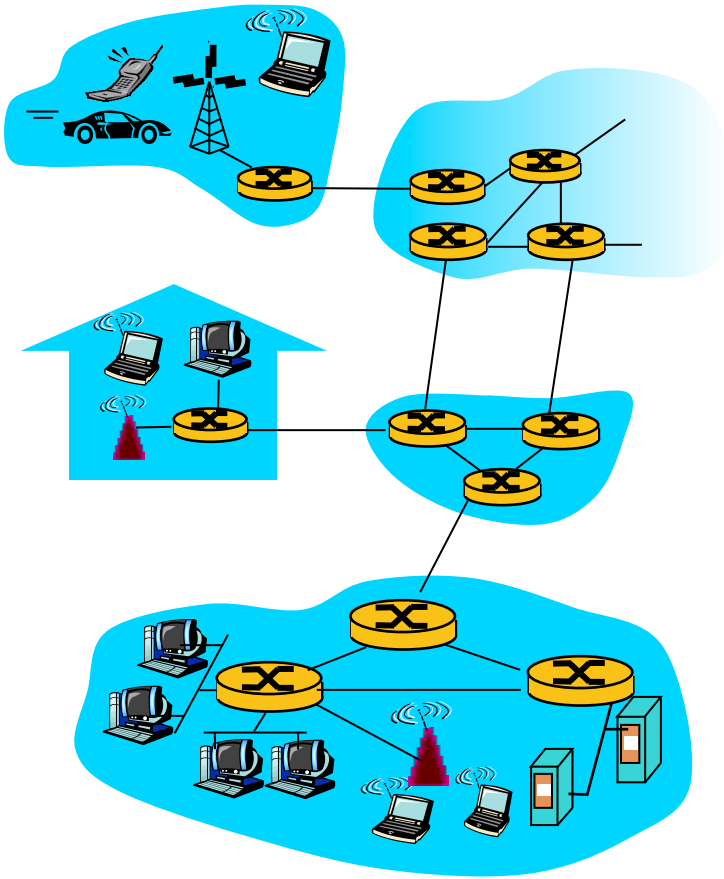
- Computadores
  - Executam os programas aplicação
  - e.g., WWW, correio electrónico
  - Utilizando os serviços da rede
- Modelo cliente/servidor
  - Os clientes solicitam serviços aos servidores
  - e.g., WWW cliente (browser)/ servidor; e-mail cliente/servidor
- Modelo peer-peer (par a par):
  - Interacção simétrica entre computadores
  - e.g.: tele-conferência, troca de ficheiros, ...



O Core (centro ou miolo)

- Malha de *comutadores de pacotes* interligados através de canais

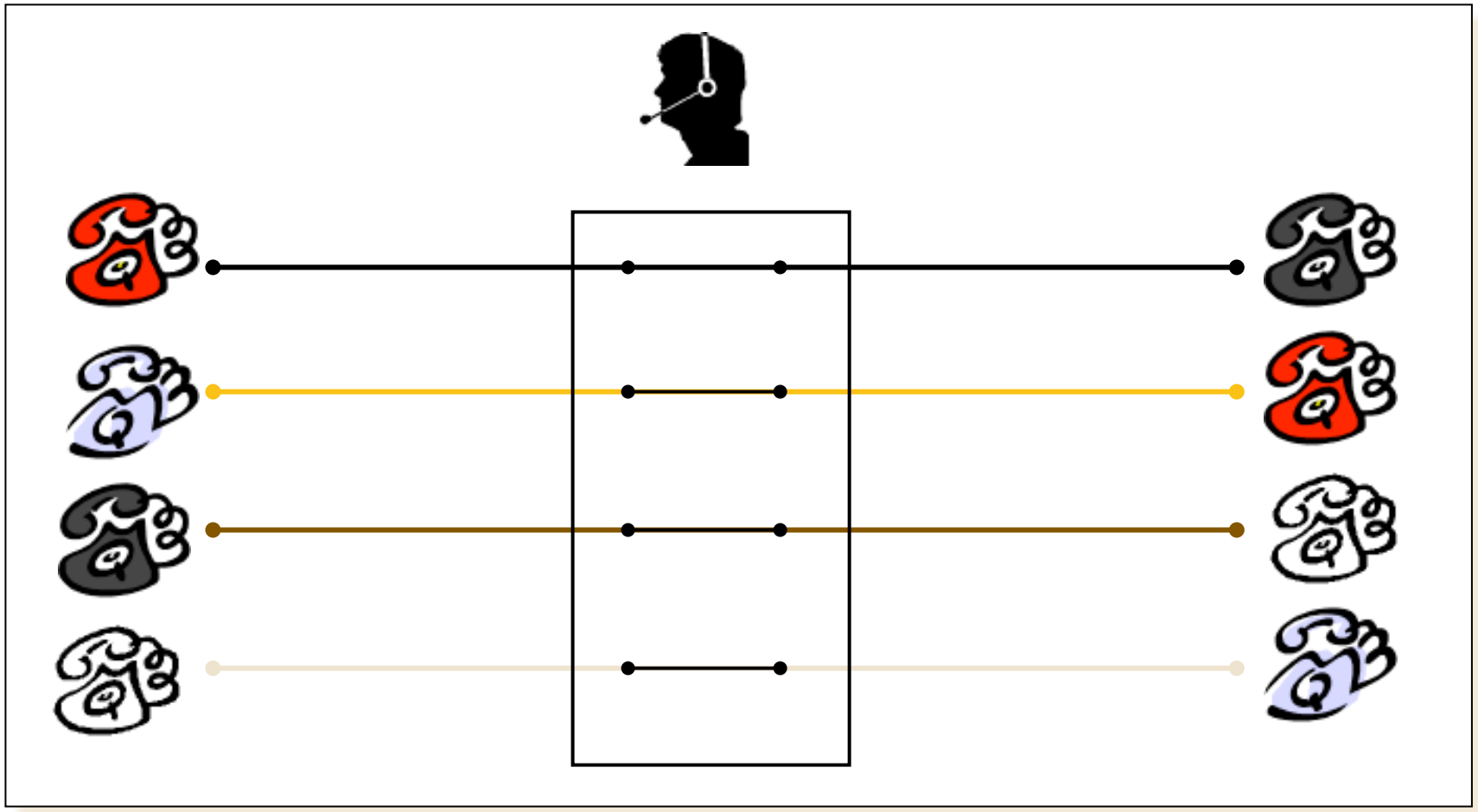
- Informação transferida desde a origem até ao destino através de caminhos dentro da malha



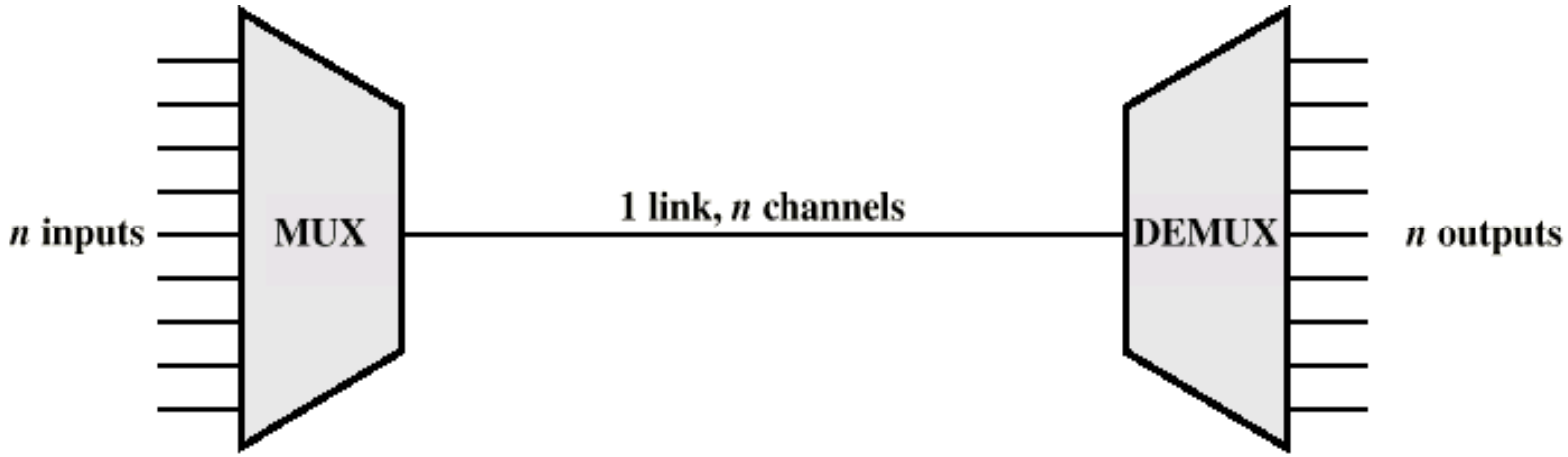
Multiplexagem dos canais

- N interlocutores pretendem comunicar com M interlocutores distantes
- Instalamos N x M canais ponto a ponto a ligar cada par deles ?
- Não, *multiplexamos* uma ou mais ligações de uma malha mais restrita pois é mais económico
- Logicamente, *multiplexar* consiste em disponibilizar vários canais que partilham o mesmo meio de transmissão (*link*)
- O termo utiliza-se em todos os contextos onde existe desdobramento em várias vias ....

Comutação manual de circuitos



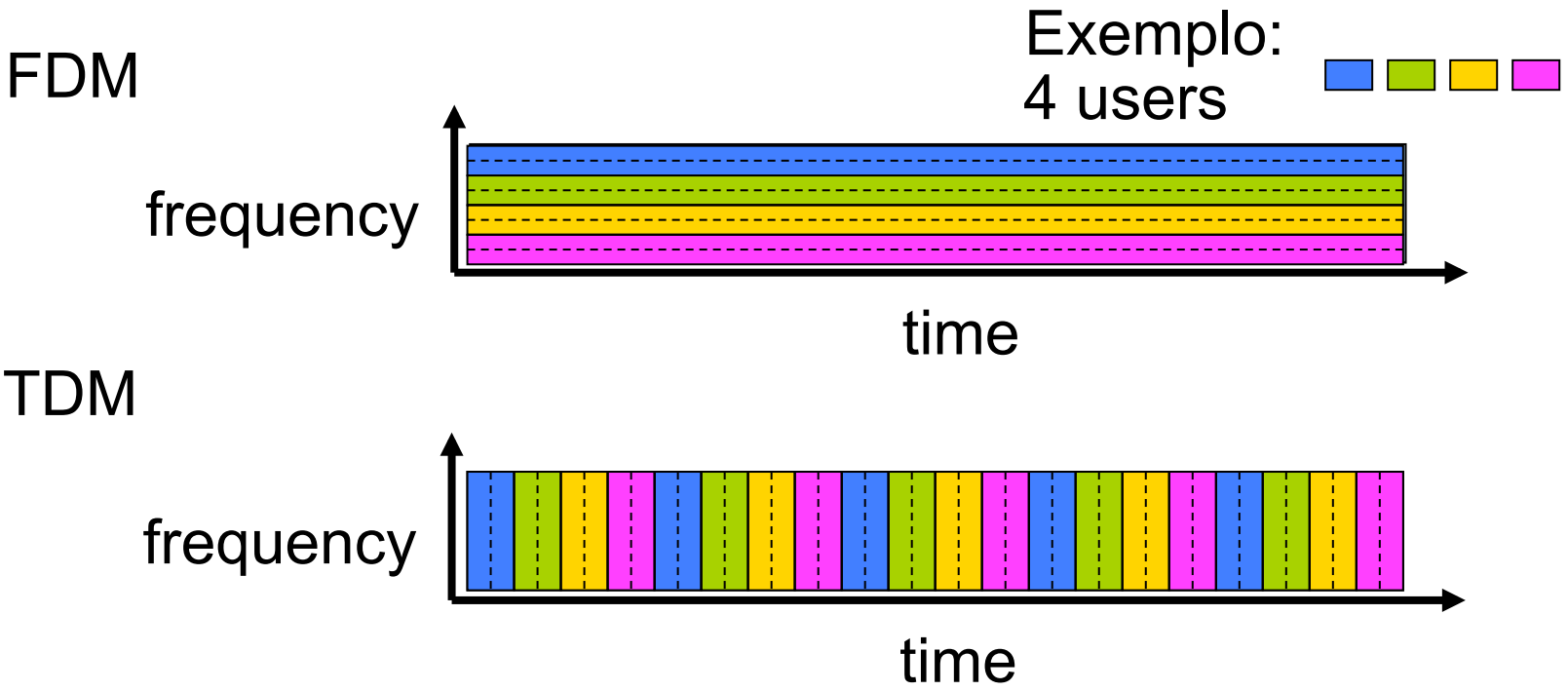
N canais que partilham o mesmo suporte (*link*)



Técnicas de *multiplexagem estáticas*

A multiplexagem directa do suporte de transmissão faz-se por:

- FDM - divisão em frequência (*frequency-division multiplexing*)
- TDM - divisão no tempo (*time-division multiplexing*)



Vantagens da comutação de circuitos

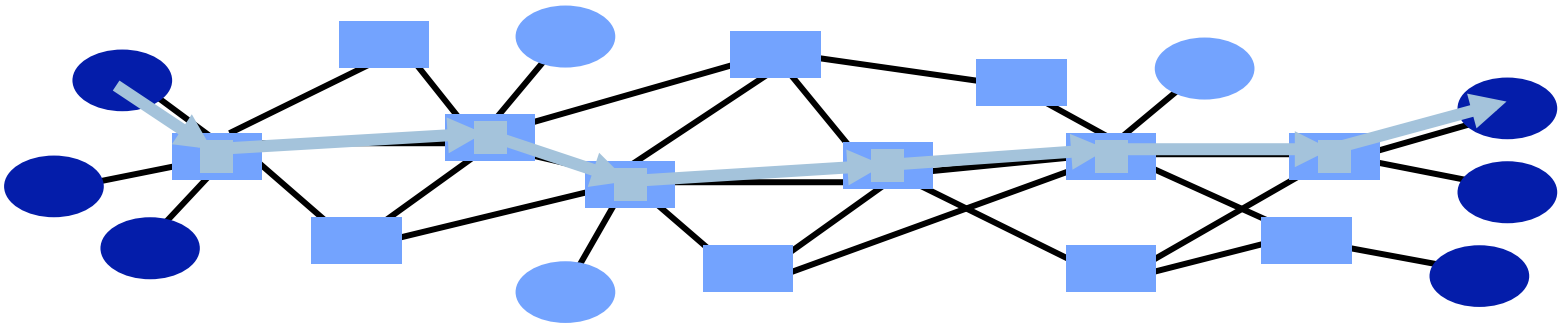
- Capacidade garantida
  - Capacidade de comunicação claramente definida
  - Garantia de capacidade
- Abstracção muito simples
  - Semelhante à de um canal directo
- Encaminhamento simplificado
  - Encaminhamento baseado em time slots ou frequências
  - Não é necessário inspeccionar o cabeçalho das mensagens
- Pequeno overhead por mensagem
  - Encaminhamento baseado em time slots ou frequências

Desvantagens da comutação de circuitos

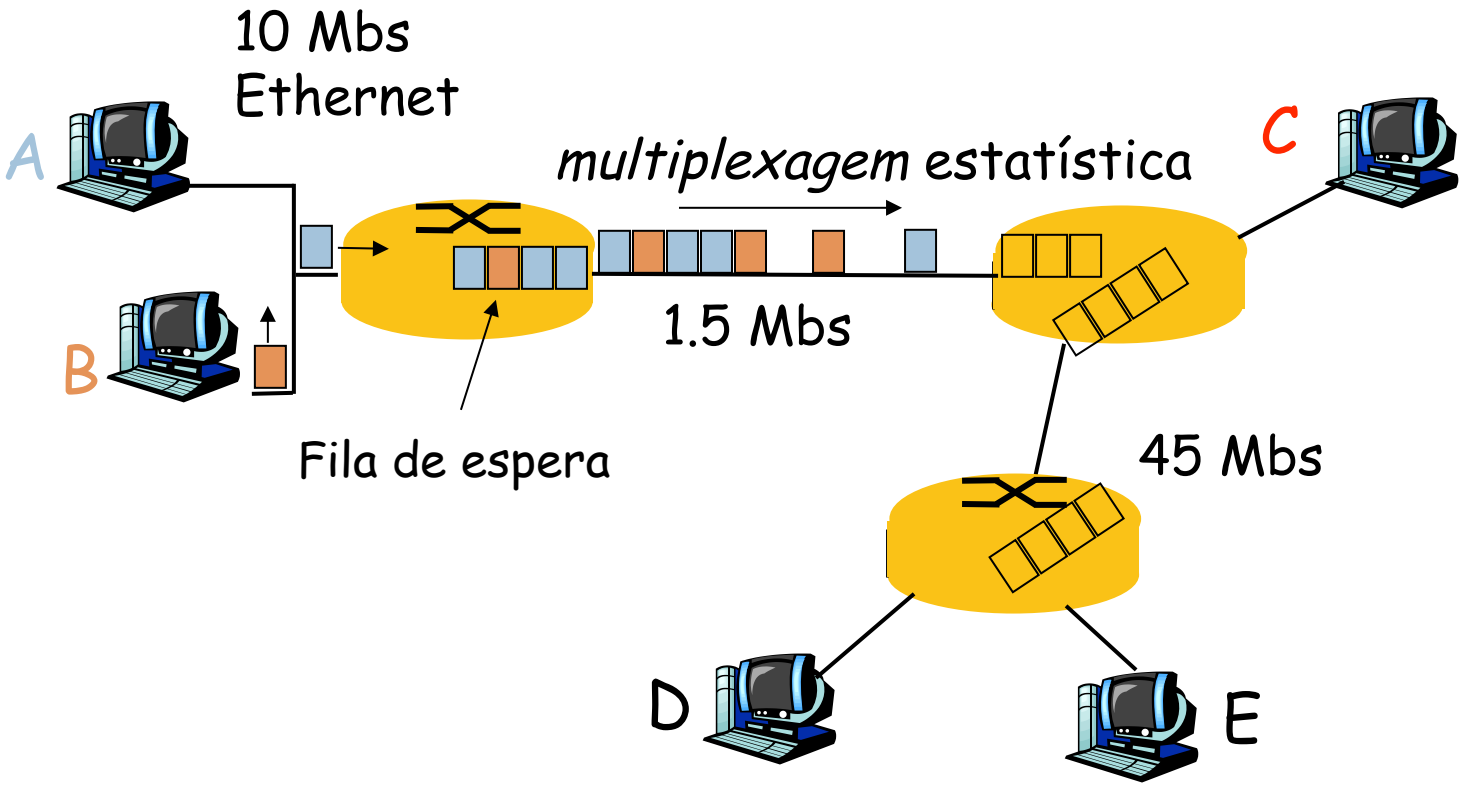
- Desperdício de capacidade de transmissão
  - Tráfego irregular desperdiça a capacidade durante os períodos de acalmia
  - Gestão inflexível da capacidade dos canais
- As conexões podem ser bloqueadas
  - Quando não há recursos disponíveis bloqueia o estabelecimento de conexões
  - Incapaz de fornecer serviço a todos os clientes
- É necessário esperar pelo estabelecimento da conexão
  - Não é possível comunicar até o circuito estar estabelecido
  - Conduz a um desperdício significativo para comunicações curtas
- Estado na rede
  - Os nós da rede mantêm estado sobre os circuitos estabelecidos

Comutação de pacotes (Packet Switching)

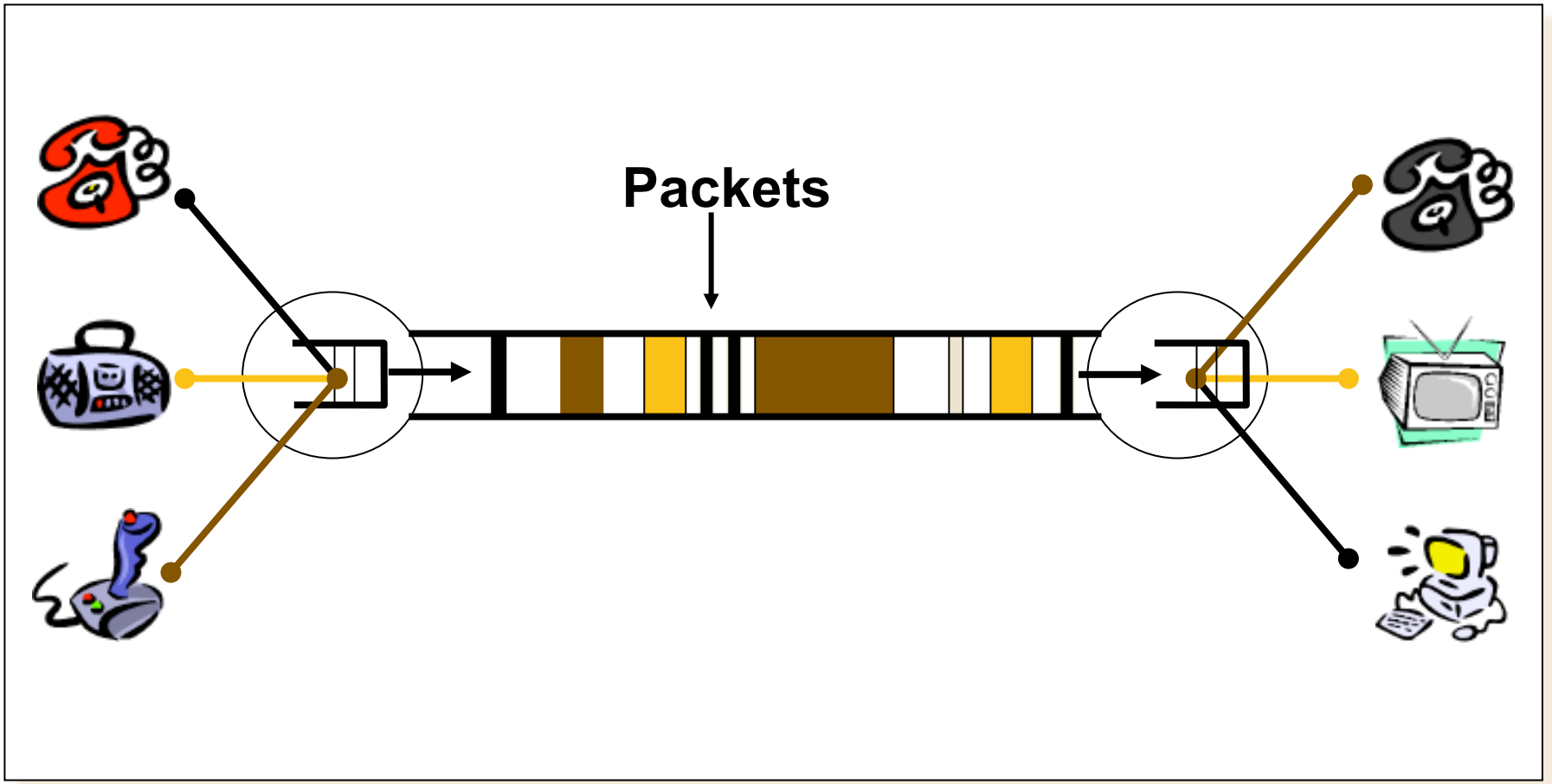
- O tráfego é dividido em pequenos pacotes (de bits)
  - Cada pacote tem um cabeçalho com o endereço do destino
- Os pacotes atravessam a rede de forma flexível
  - O encaminhamento é baseado no endereço
  - Os comutadores podem memorizar os pacotes momentaneamente
- O computador de destino final reconstrói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



Como funciona a comutação de pacotes



Comutação de pacotes: multiplexagem estatística



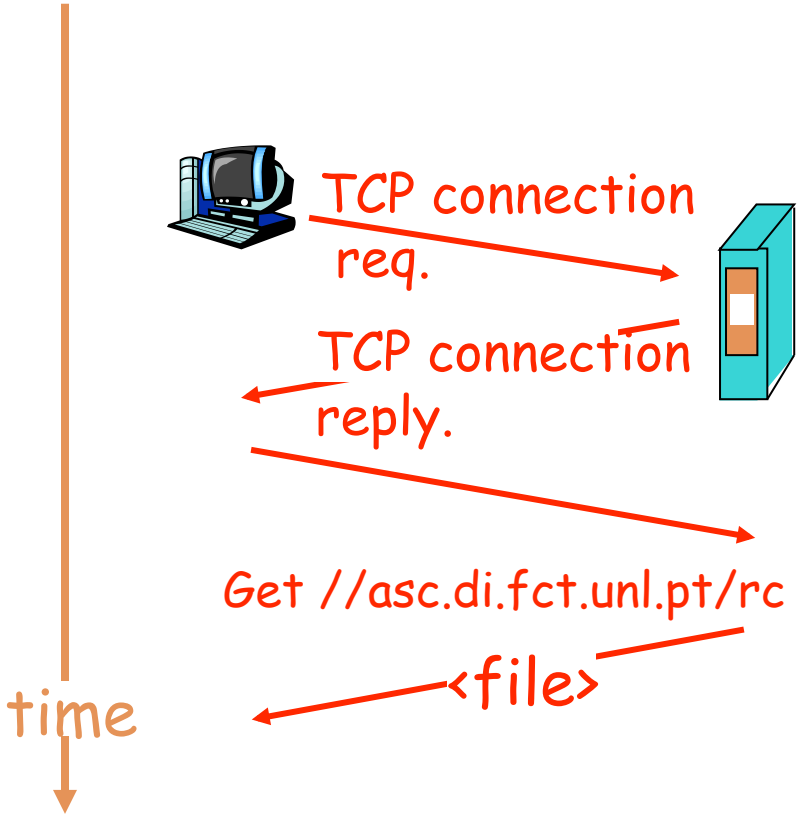
Protocolos de Comunicação

- Quando várias entidades distribuídas colaboram para atingirem um fim, ao conjunto de mensagens trocadas e às acções associadas às mesmas chama-se um “protocolo”.
- Por exemplo, ir ao médico, este solicitar análises, voltar ao médico, ir à farmácia, telefonar ao médico, etc. é um exemplo de um protocolo executado por humanos
- Os protocolos de comunicação são executados entre máquinas em vez de humanos, utilizando a rede para trocarem mensagens
- Toda a actividade da Internet é governada por protocolos de comunicação normalizados

O que é um protocolo

- Um protocolo de comunicações define o formato de todas as mensagens trocadas entre entidades na rede, a sua ordem relativa, e as acções a executar após a recepção e a emissão dessas mensagens

- Um protocolo também define o comportamento dos participantes perante falhas.



Exemplo: HTTP — HyperText Transfer Protocol

```
GET /courses/archive/spring06/cos461/ HTTP/1.1
Host: www.cs.princeton.edu
User-Agent: Mozilla/4.03
CRLF
```

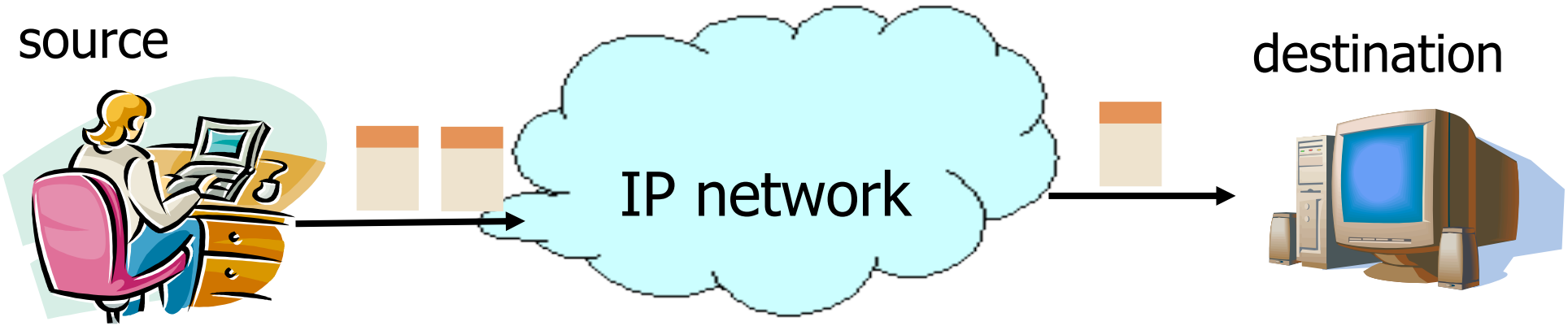
Request

Response

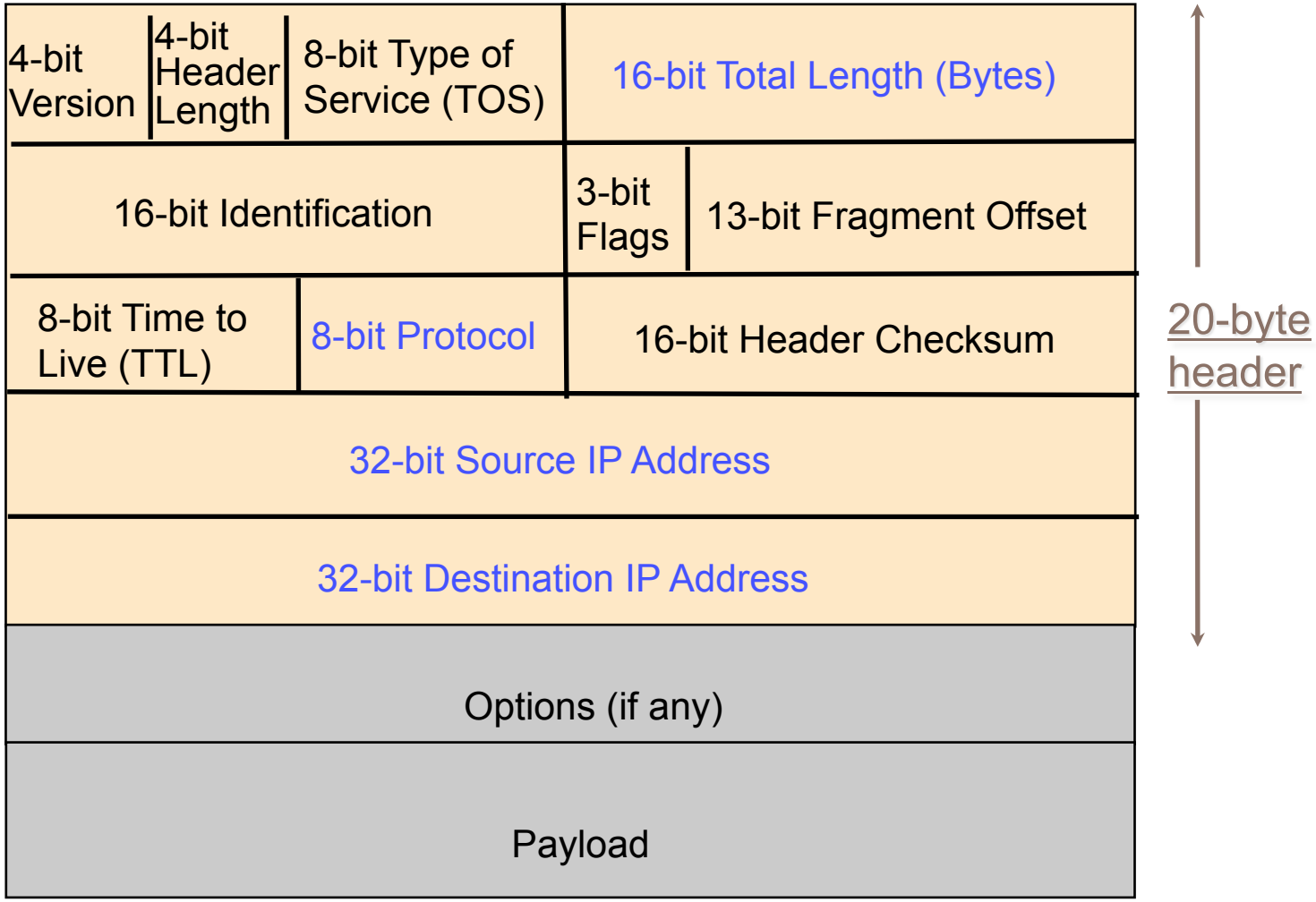
```
HTTP/1.1 200 OK
Date: Mon, 6 Feb 2006 13:09:03 GMT
Server: Netscape-Enterprise/3.5.1
Last-Modified: Mon, 6 Feb  2006 11:12:23 GMT
Content-Length: 21
CRLF
Site under construction
```

Exemplo: IP - Best-Effort Packet Delivery

- Comutação de pacotes (packet switching)
  - Os dados são enviados em pequenas mensagens (pacotes)
  - O cabeçalho tem os endereços origem e destino
- Entrega baseada no melhor esforço (best-effort delivery)
  - Os pacotes podem perder-se
  - Os pacotes podem corromper-se
  - Os pacotes podem chegar por outra ordem

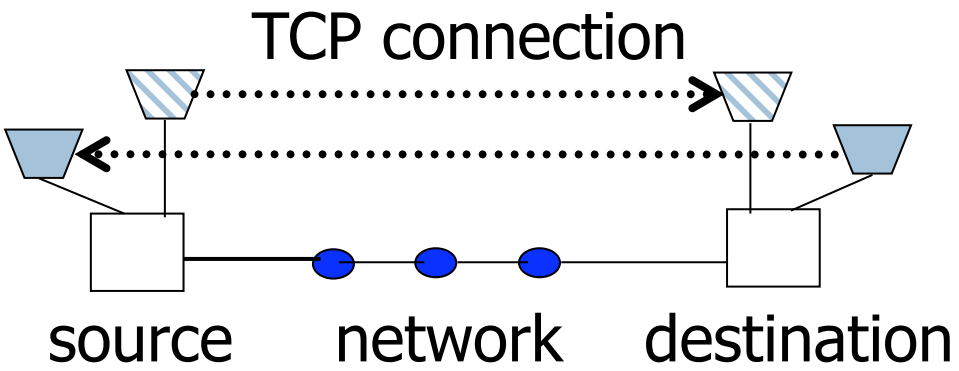


Exemplo: Pacote IP



Exemplo: TCP - Transmission Control Protocol

- Serviço de comunicação (*socket*)
  - Sequência de bytes ordenada (*ordered, reliable byte stream*)
  - Bidireccional
- Os mecanismos essenciais estão nos computadores
  - Retransmissão de pacotes
  - Suprimir duplicados e reordenar os pacotes
  - Controlo de fluxos para não afogar o receptor
  - Controlo de saturação para se adaptar à capacidade da rede



Normalização dos protocolos

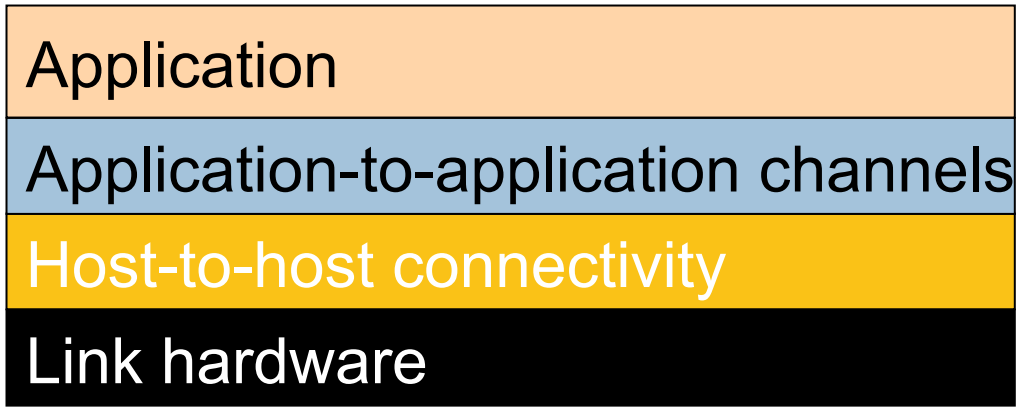
- Os computadores ao comunicarem têm de "falar" o mesmo protocolo
  - A normalização permite existirem diferentes implementações
  - Senão teriam sempre de ser os mesmos a escrever o software
- Quem trata das normas: Internet Engineering Task Force
  - Baseada em grupos de trabalho
  - Produz "Request For Comments" (RFCs)
    - Que promovem normas baseadas em consenso e código executável
    - E.g., RFC 1945 on "HyperText Transfer Protocol – HTTP/1.0"
  - IETF via a web http://www.ietf.org
- Normas de facto: software escrito sempre pelos mesmos
  - P2P file sharing, Skype, <your protocol here>…

Estruturação interna das redes

- Numa rede estão envolvidas muitas componentes (*routers*, canais, hosts, protocolos, aplicações, software, ...)
- Até aqui temos insistido na estruturação física da rede, mas a mesma está também estruturada de forma lógica
- As redes são estruturadas logicamente por níveis ou camadas
- Cada camada representa um conjunto de funcionalidades bem definidas oferecendo uma interface e um conjunto de serviços bem definidos à camada superior
- Para funcionar, cada camada utiliza os serviços da camada inferior

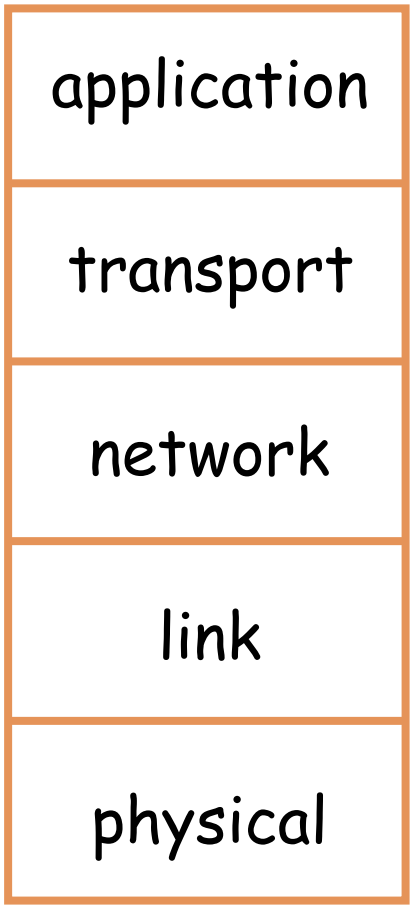
Porquê a estruturação por camadas ?

- Modularidade - sub-divide o problema
  - Cada camada assenta nos serviços (interface) da camada inferior
  - Cada camada exporta serviços para a camada seguinte
- A interface entre camadas define os serviços
  - Esconde os detalhes de implementação
  - Permite a evolução independente de cada camada

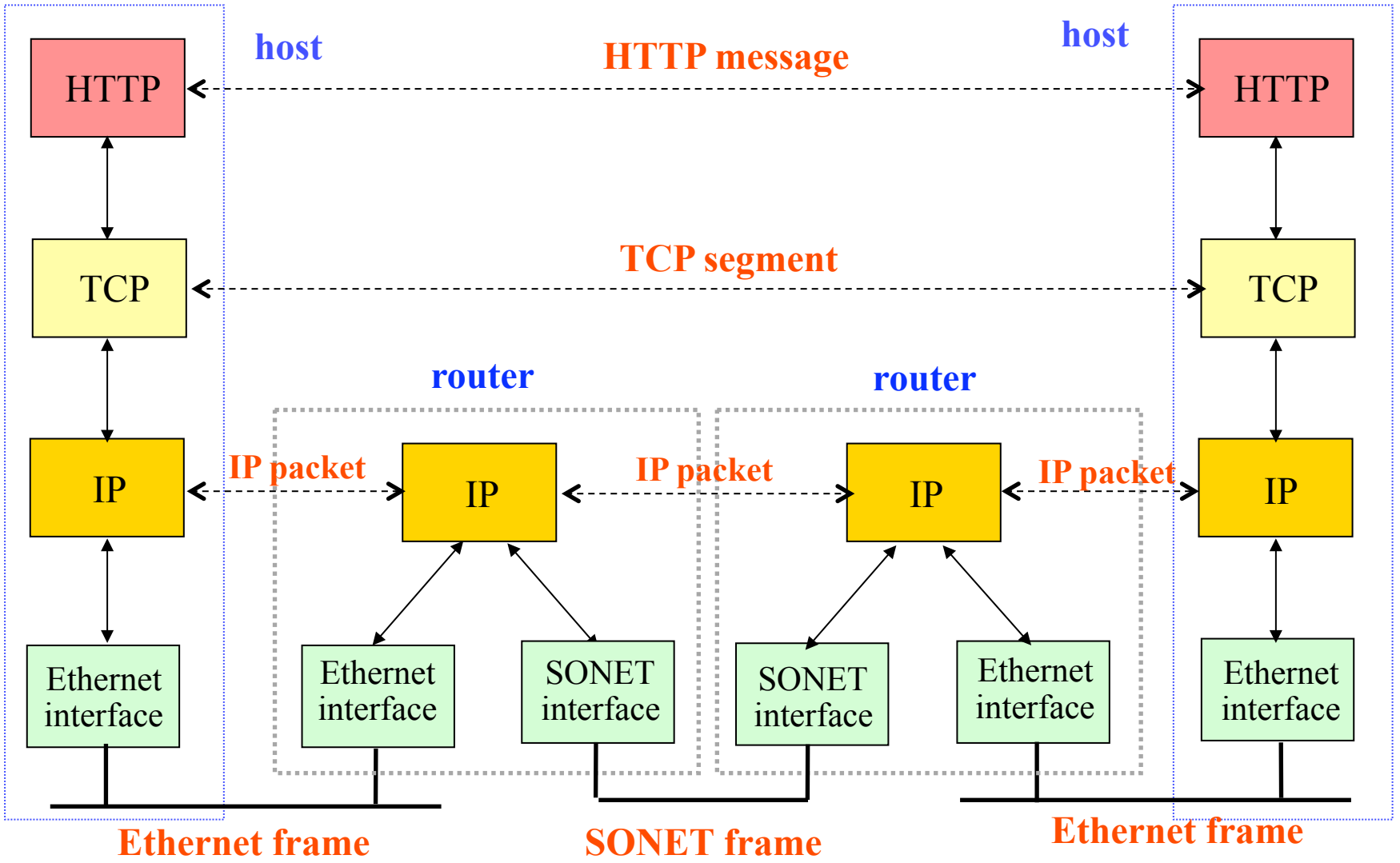


A pilha de protocolos TCP/IP

- **aplicação:** o suporte das aplicações de rede
  - **ftp, smtp, http,** …
- **transporte:** transferência de dados host-host
  - **TCP, udp**, …
- **rede:** encaminhamento de datagrams da origem até ao destino
  - **IP**, protocolos de encaminhamento
- **link:** transferência de dados entre elementos de rede contíguos
  - Ethernet, PPP
- **Nível físico:** transmissão directa de bits sobre o meio de suporte da ligação



Message, Segment, Packet, and Frame



Terminologia

Os protocolos são implementados através da troca de unidades lógicas de informação entre elementos de rede. Nós designaremos por:

"Frames" ou Tramas as unidades trocadas ao nível *data-link*

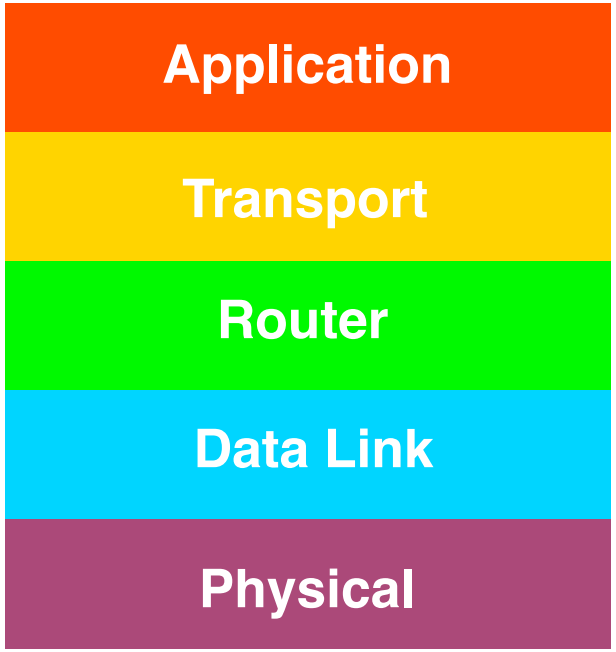
Pacotes ou Datagramas as unidades trocadas ao nível rede

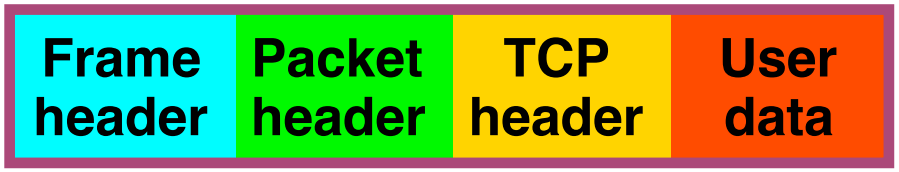
Segmentos as unidades trocadas ao nível transporte

Mensagens as unidades trocadas aos níveis superiores

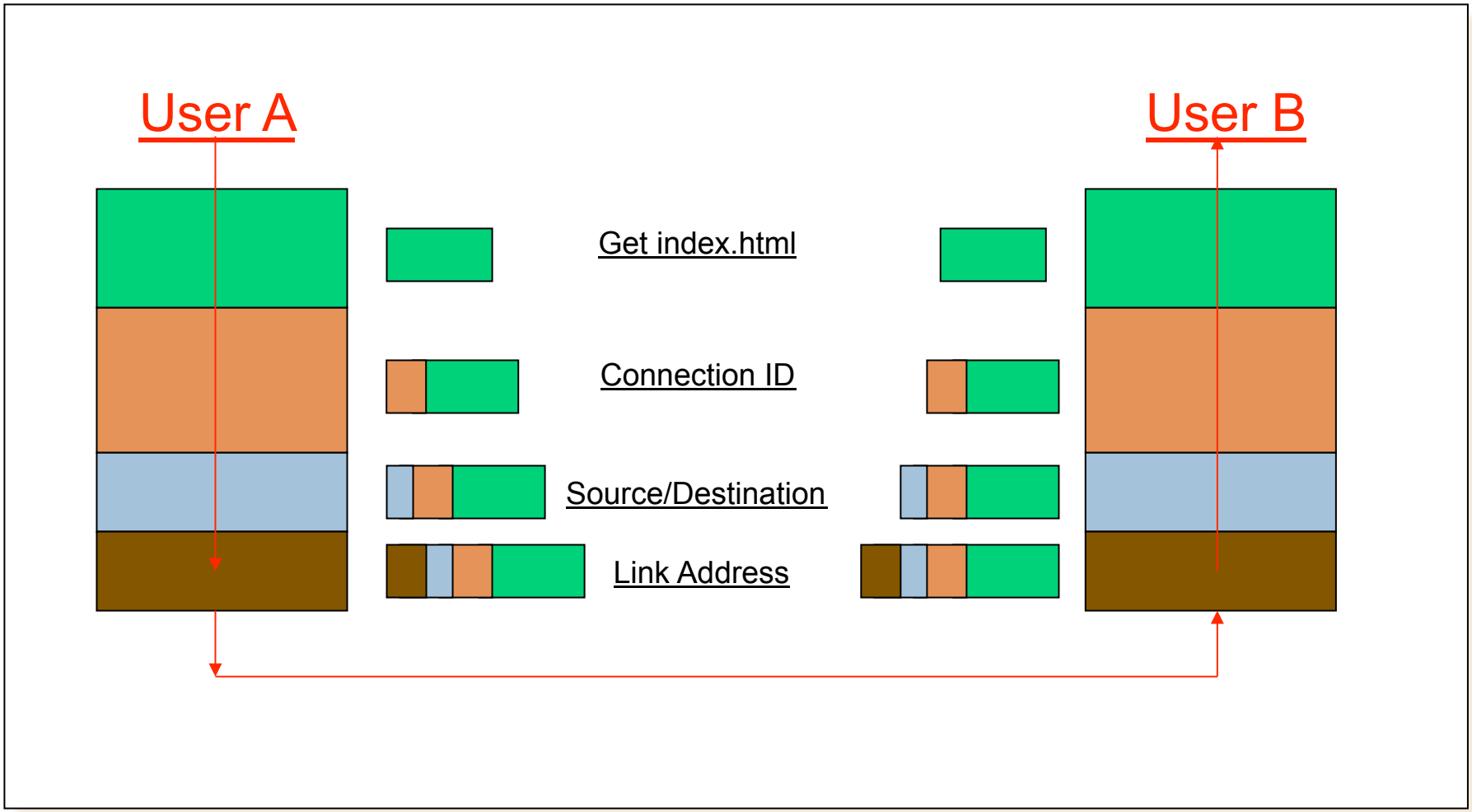
Tratamento dos dados nos diferentes níveis

- Equipamentos diferentes fazem coisas diferentes
  - Physical layer: electrical signals (repeaters and hubs)
  - Link layer: frames (bridges and switches)
  - Network layer: packets (routers)

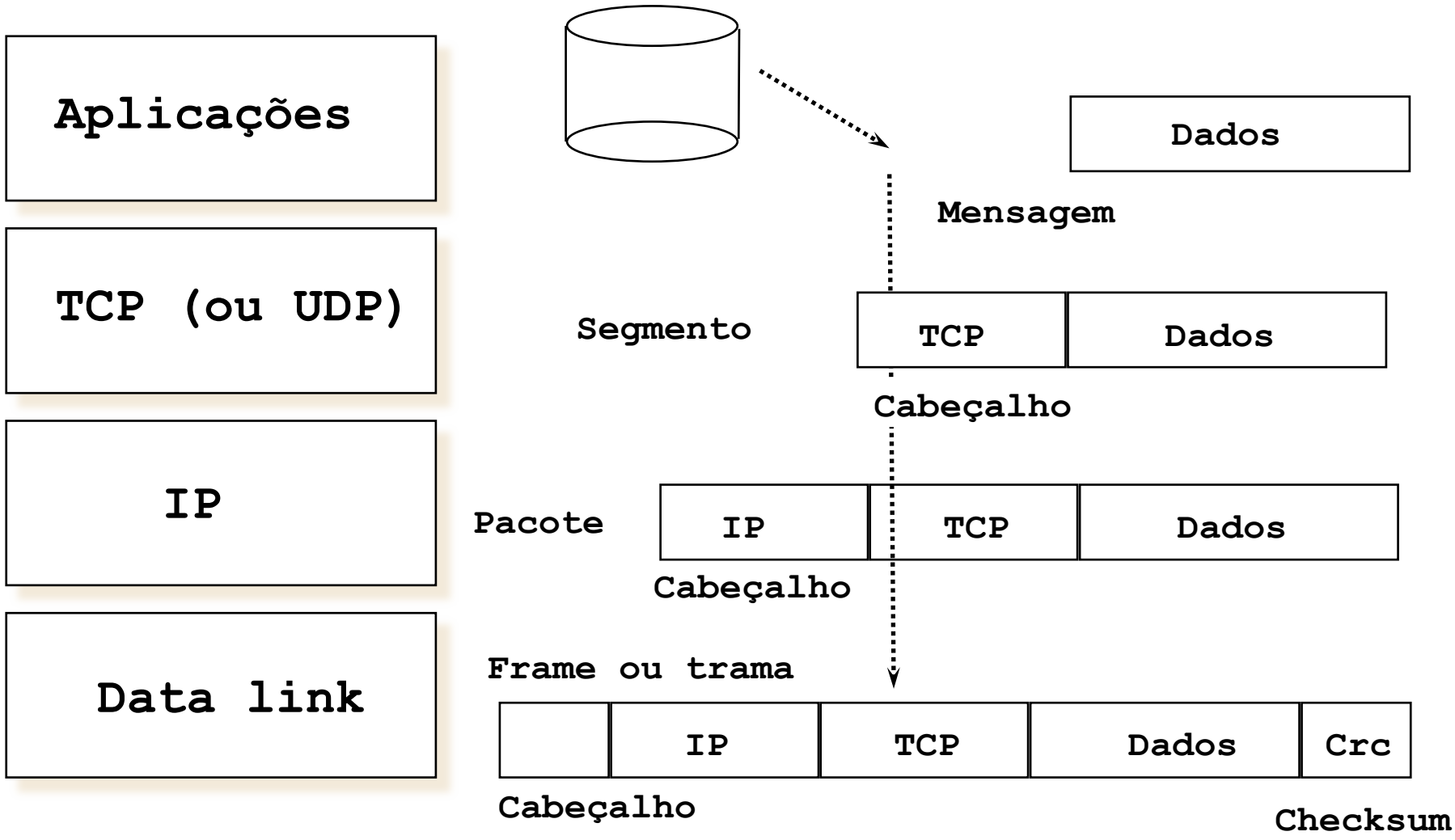




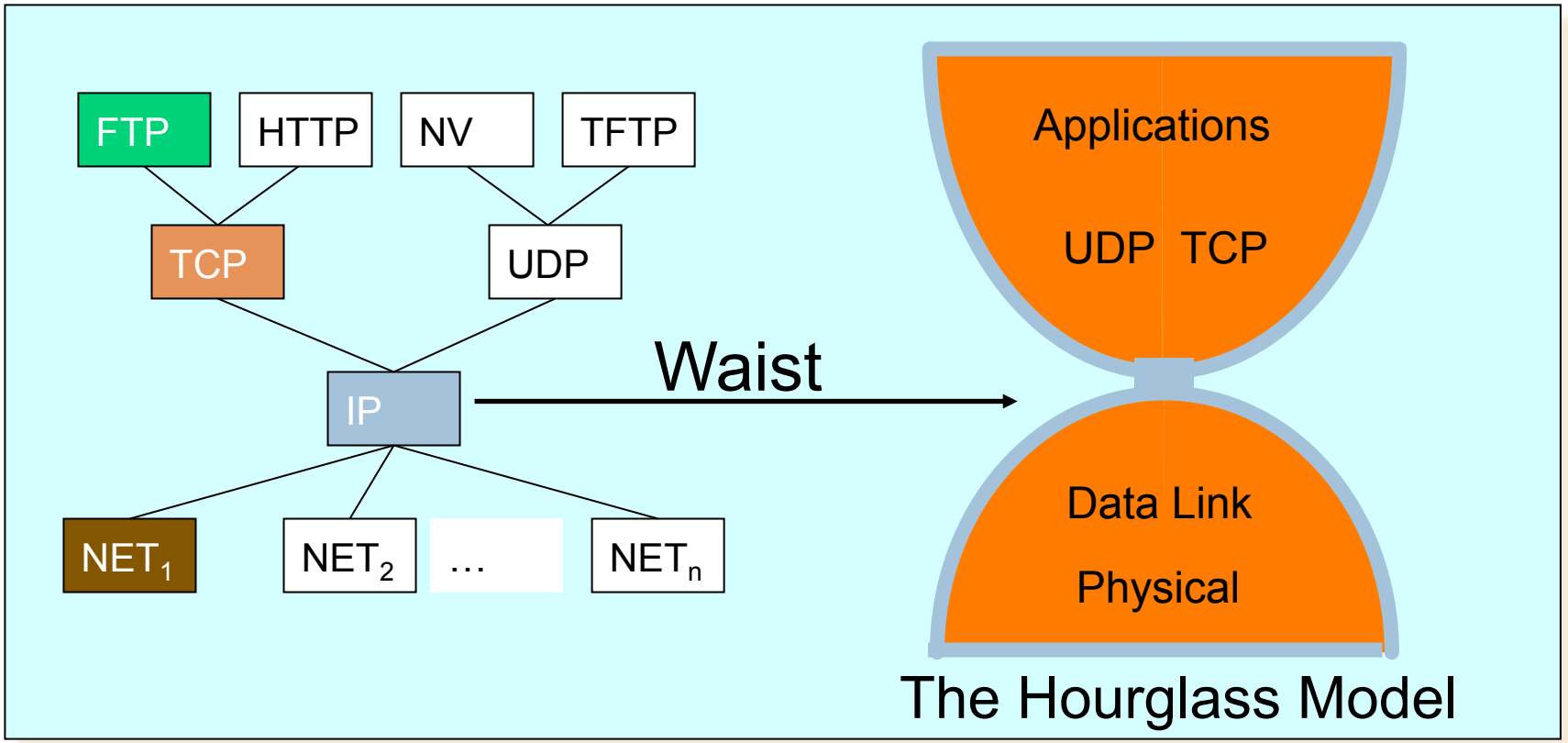
Encapsulamento por camadas



Dados e camadas de protocolos



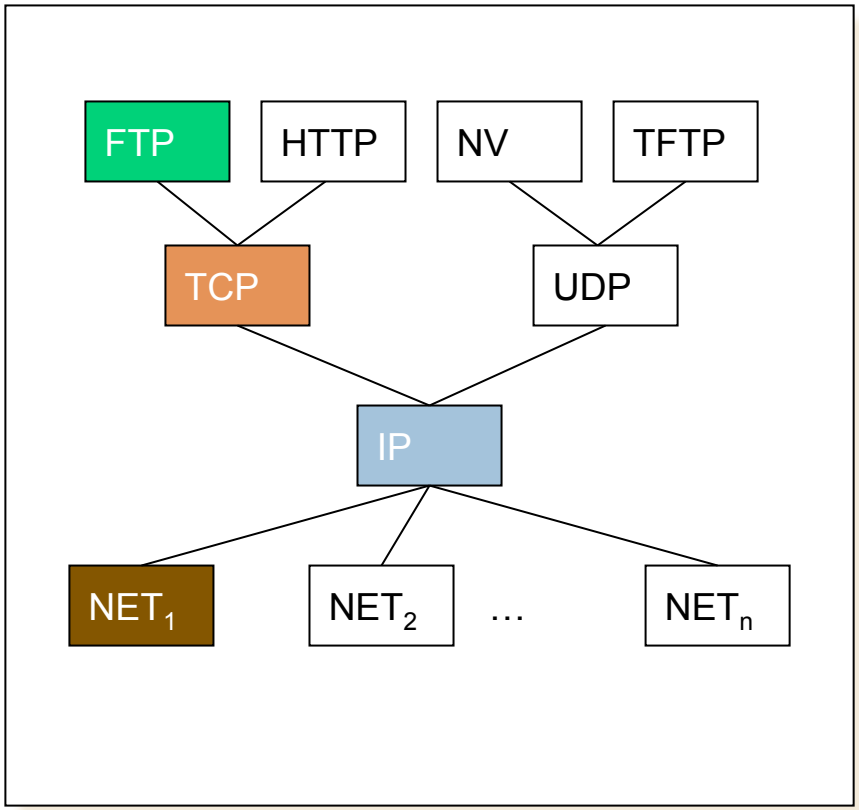
A Suite de protocolos Internet

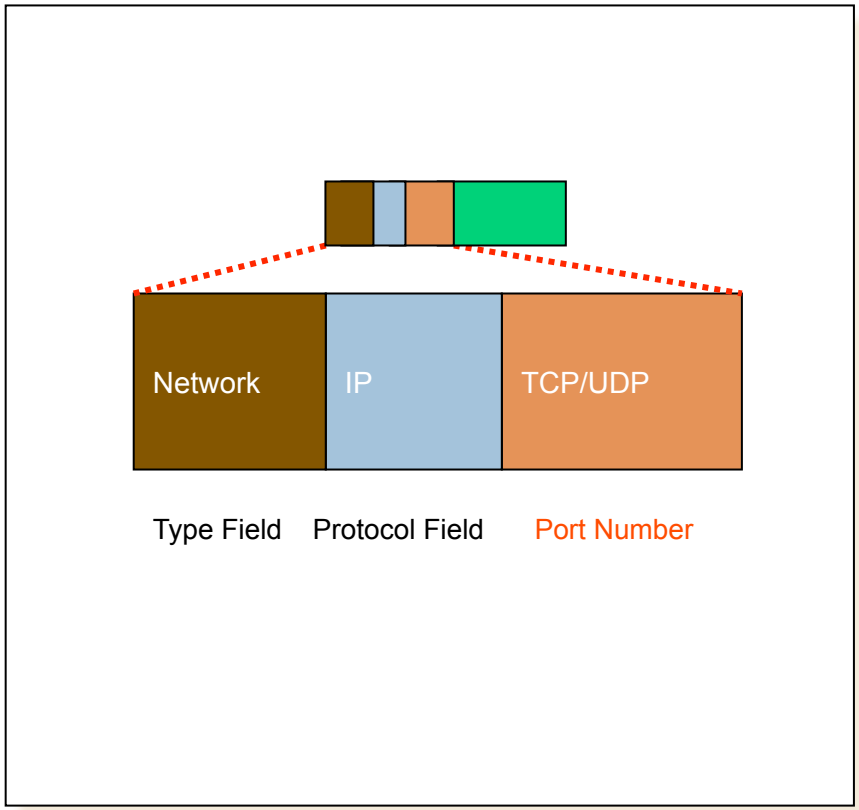


O gargalo facilita a inter-operação

Protocol demultiplexing

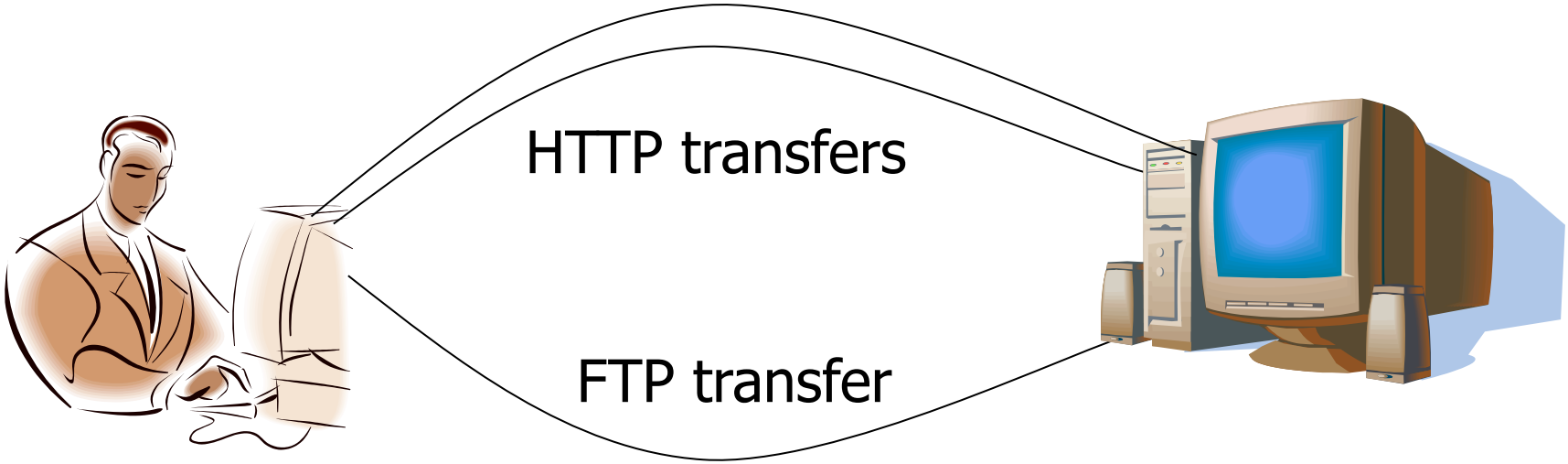
- Multiplicidade de escolha a diferentes níveis





Demultiplexing: números das portas

- Permite diferenciar diferentes transferências
  - Conhecer a origem e o destino não é suficiente
  - Necessita-se de um identificador diferente para cada uma
- As portas permitem distinguir diferentes serviços no mesmo host
  - E.g., HTTP server running on port 80
  - E.g., FTP server running on port 21



As camadas não têm defeitos?

- A camada N pode duplicar funcionalidades das inferiores
  - E.g., tratamento de erros e retransmissões
- Algumas camadas necessitam de informação de outras
  - E.g., timestamps, maximum transmission unit size (MTU)
- A adesão estrita ao modelo pode ter pior desempenho
  - E.g., porque se escondem detalhes sobre o que se passa de facto
- Algumas camadas não têm uma fronteira bem definida
  - Existem inter-dependências devido a necessidades de desempenho
  - Existem mesmo dependências que estão normalizadas (header checksums)
- Os cabeçalhos começam a ser realmente enormes
  - Às vezes maiores que o conteúdo dos pacotes (payload)