

# Infraestrutura ComputacionalIII

*Conceitos básicos de Redes de  
Computadores*

*Luis C.E. Bona (bona@inf.ufpr.br)*

Slides parcialmente baseados no livro:

*Computer Networking: A Top Down  
Approach. Jim Kurose, Keith Ross*



# Visão geral

## ❖ Neste módulo

- Conceitos principais de redes de computadores
- Configuração dos principais serviços de rede em servidores Linux
- Ferramentas e aplicações de rede
- Conceitos de Nuvens Computacionais e virtualização



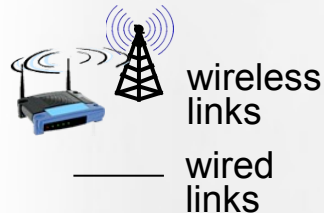
# Hoje

- ❖ O que é a Internet (visão top-down)
- ❖ Principais tecnologias e meios de conexão
- ❖ Noções de protocolos
- ❖ A camada de protocolos da Internet
- ❖ Aplicações: HTTP / DNS / SSH

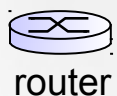
# O que é a Internet



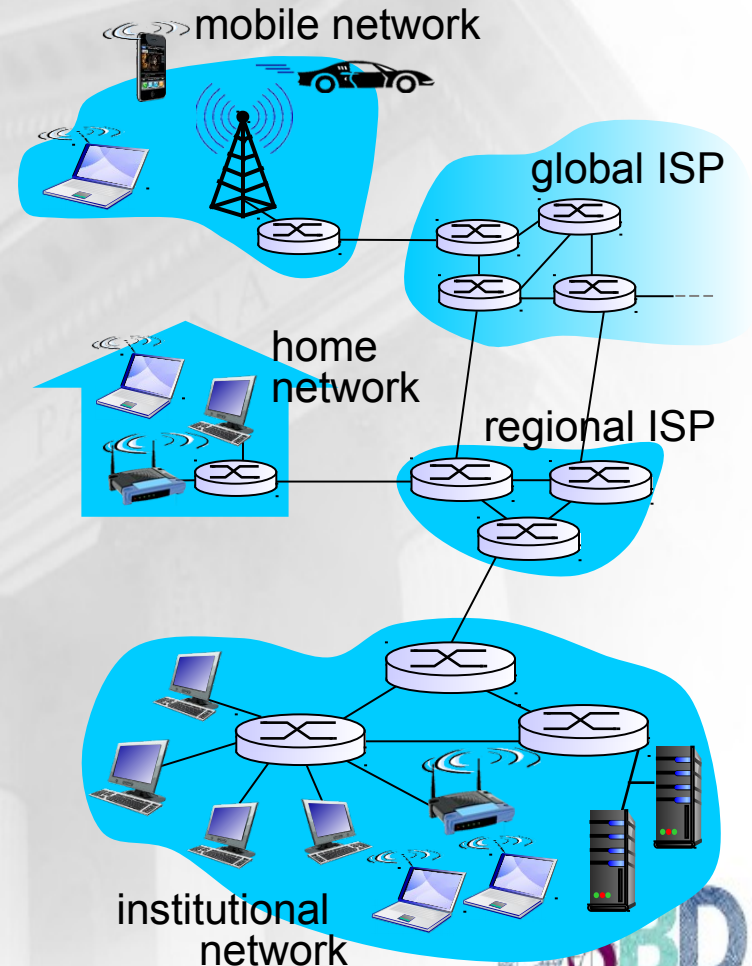
- ❖ Milhões de dispositivos conectados:
  - *hosts*
  - executando *aplicações de rede*



- ❖ *Links de comunicação*
  - Fibra, cobre, rádio, satélite,

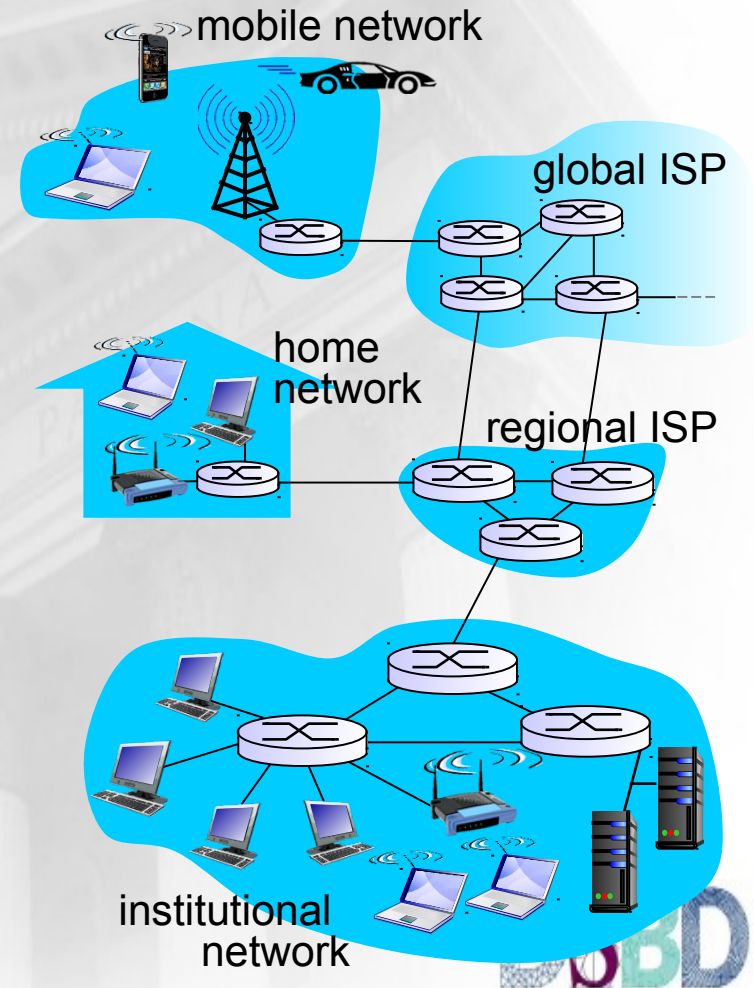


- ❖ *Comutação de pacotes*
  - roteadores e *switches*



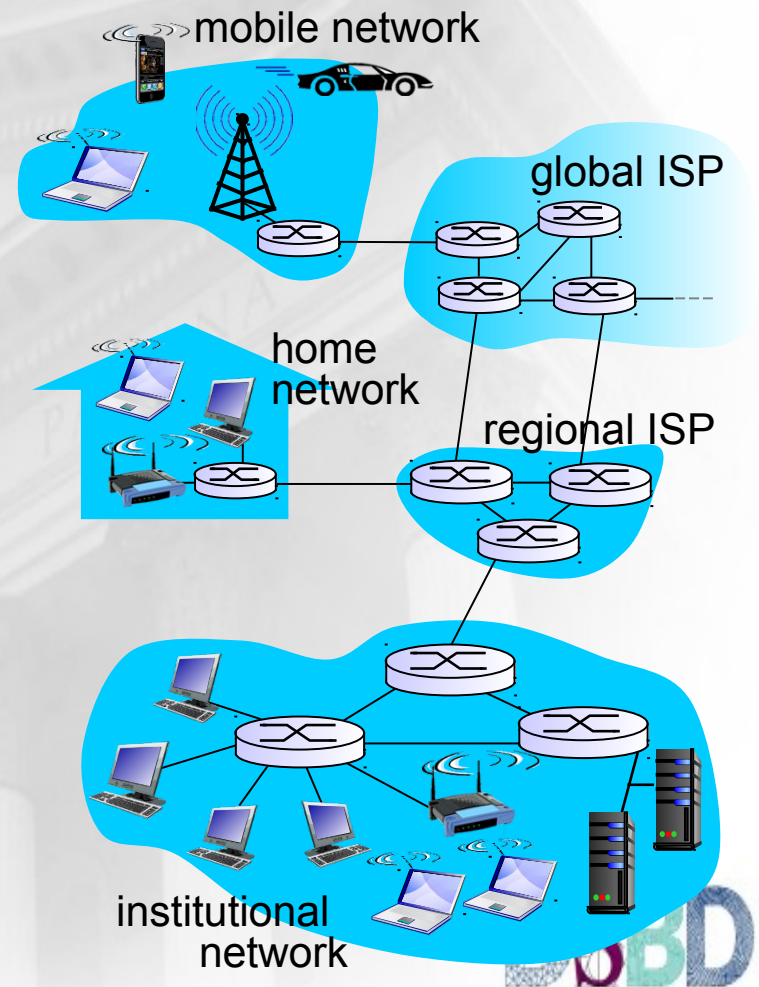
# O que é a Internet?

- ❖ *Internet: “network of networks”*
- ❖ *protocolos*
  - TCP, IP, HTTP, Skype, 802.11
- ❖ *Padrões da Internet*
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force



# O que é a Internet: Visão de Serviço

- ❖ *Infraestrutura que provê serviços para as aplicações*
- ❖ *Proporciona interfaces de programação para as aplicações*
  - Ganchos (*hooks*) que permite a aplicação se conectar na Internet e enviar e receber dados
  - Diferentes opções de serviços



# Protocolos

## *Protocolos humanos:*

- ❖ Diversas regras de comportamento para determinadas situações

## *Protocolos de rede:*

- ❖ Para máquinas
- ❖ Toda a comunicação na Internet é regida por algum protocolo

*protocolos* definem *formatos e ordem das mensagens enviadas e recebidas* entre entidades da rede e *ações realizadas* na transmissão e recepção e *mensagens*





# Olhando mais afundo

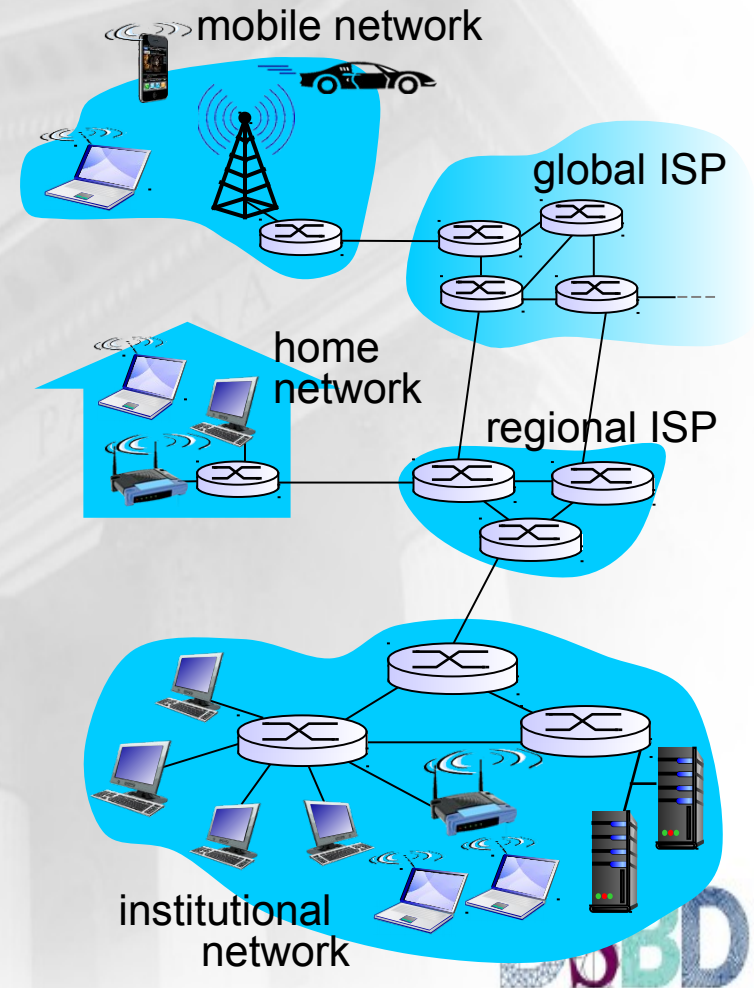
## ❖ *Borda da rede (folhas):*

- hosts: clientes/servidores
- Servidores normalmente em datacenters

## ❖ *Meio físico*

## ❖ *Núcleo da rede*

- roteadores





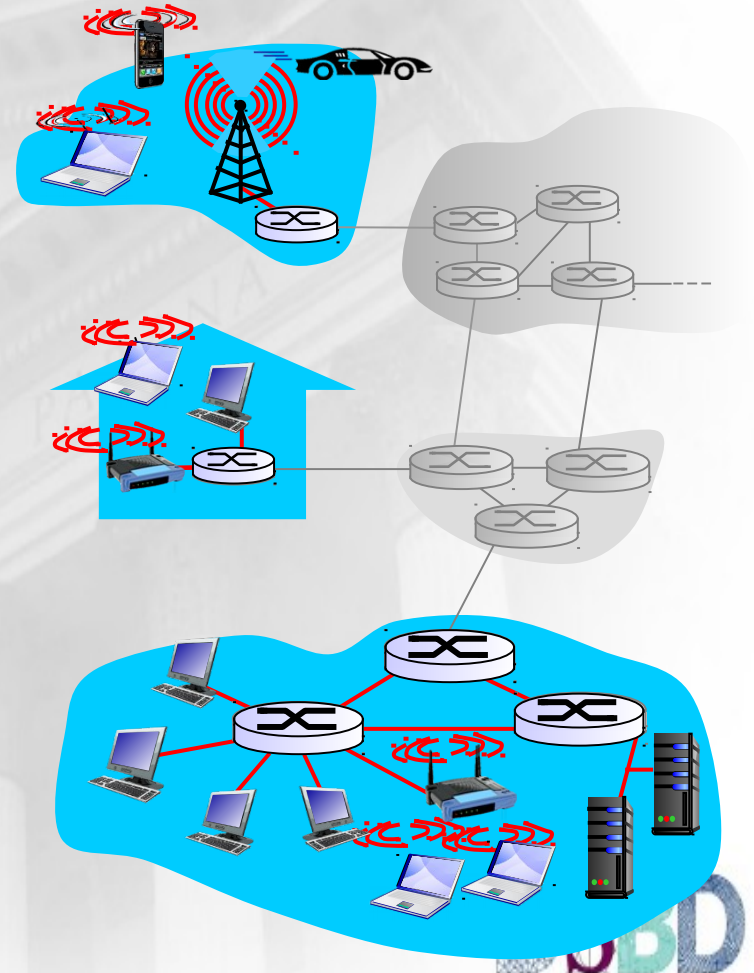
# Redes de acesso

*Q: Como conectar um end host à um roteador da borda?*

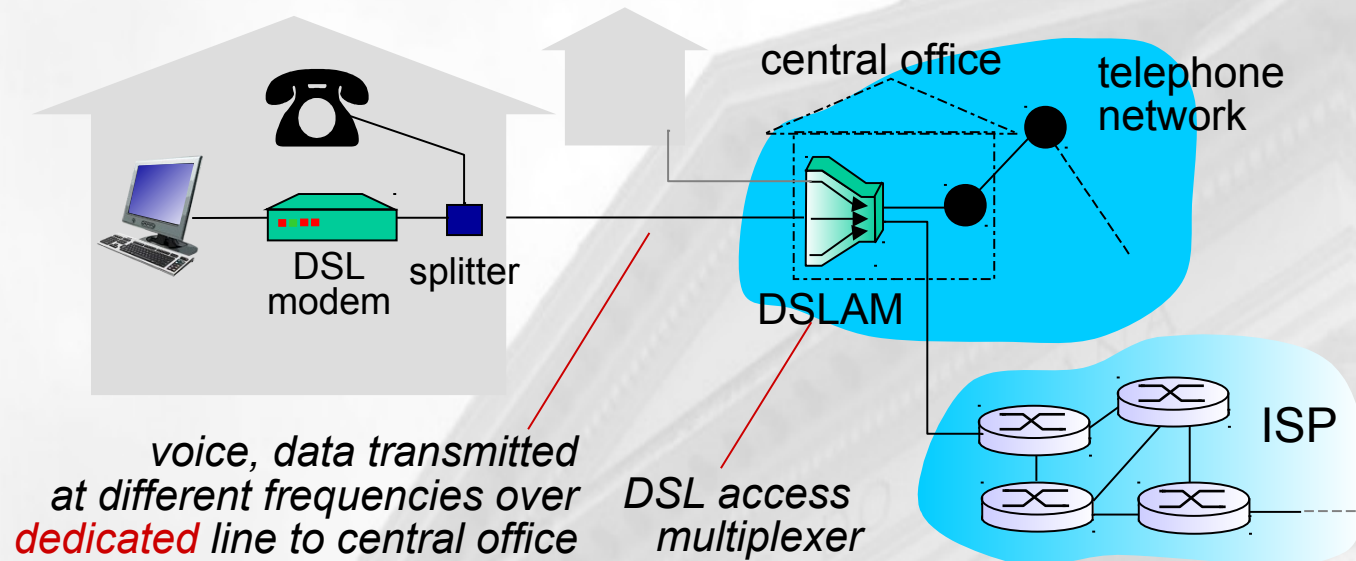
- ❖ Redes de acesso residencial
- ❖ Redes de acesso institucionais
- ❖ Redes de acesso móveis

*Com o que se preocupar?*

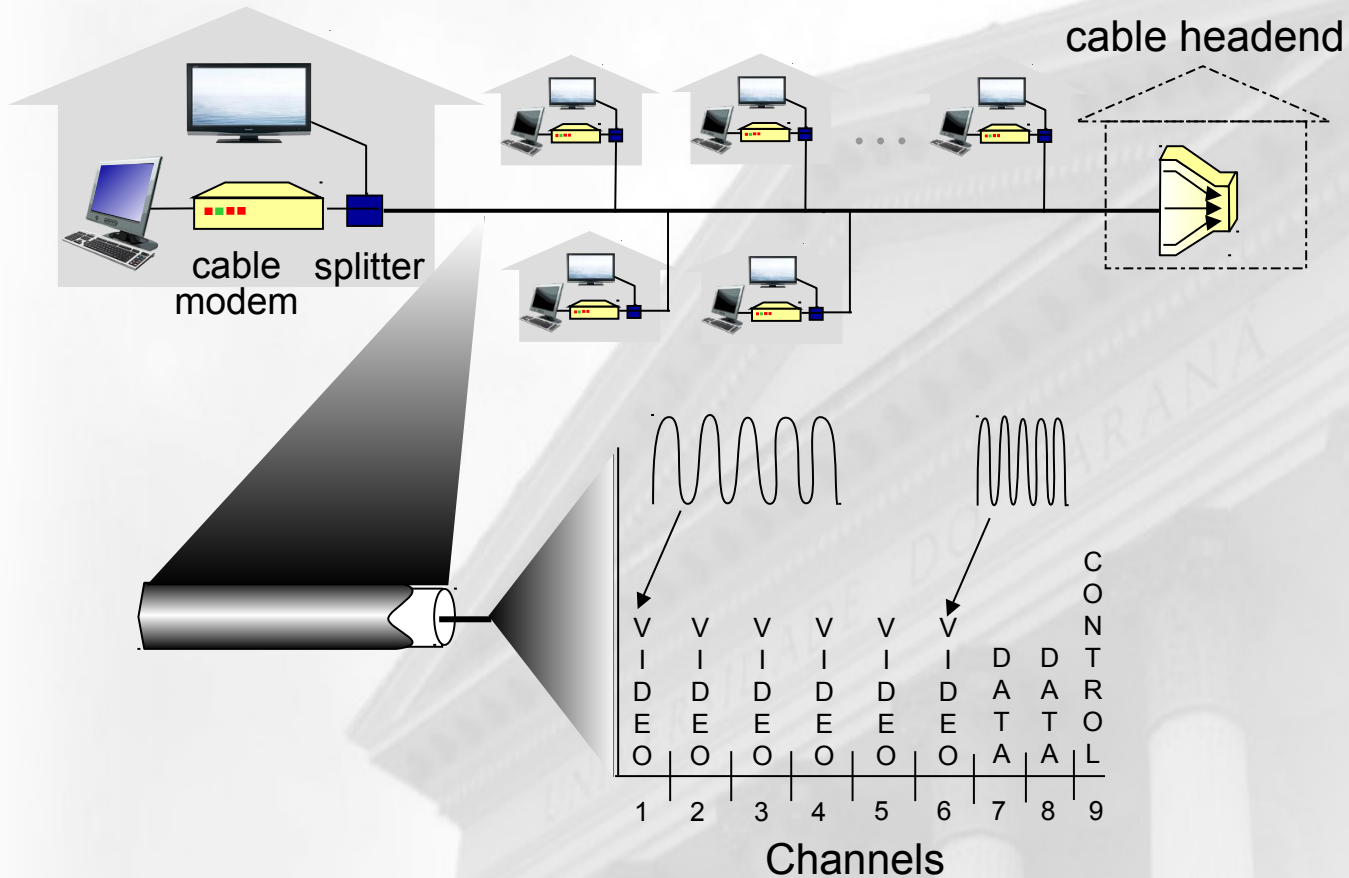
- ❖ Largura de banda
- ❖ Compartilha ou dedicado
- ❖ O que mais?



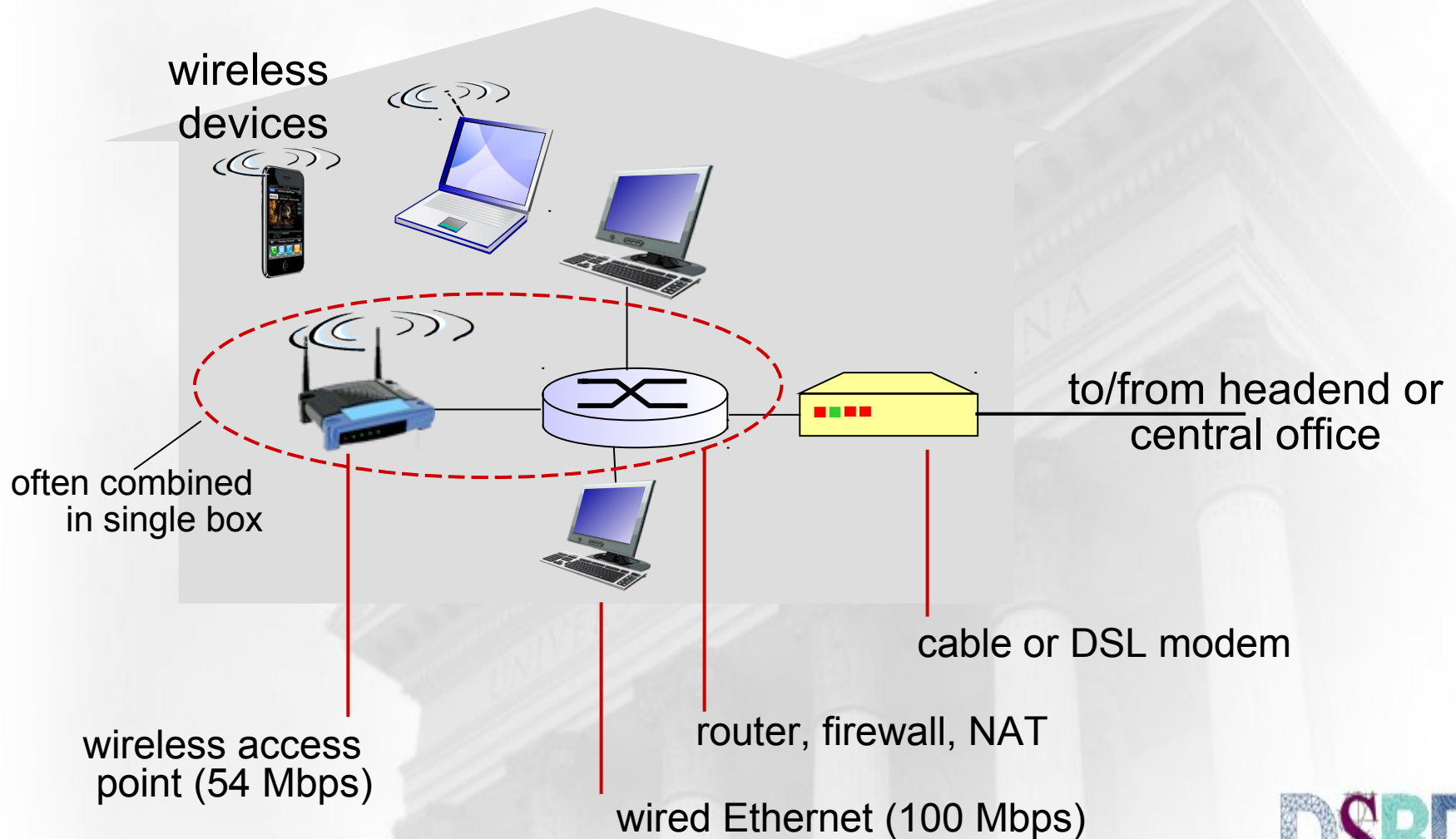
# ADSL



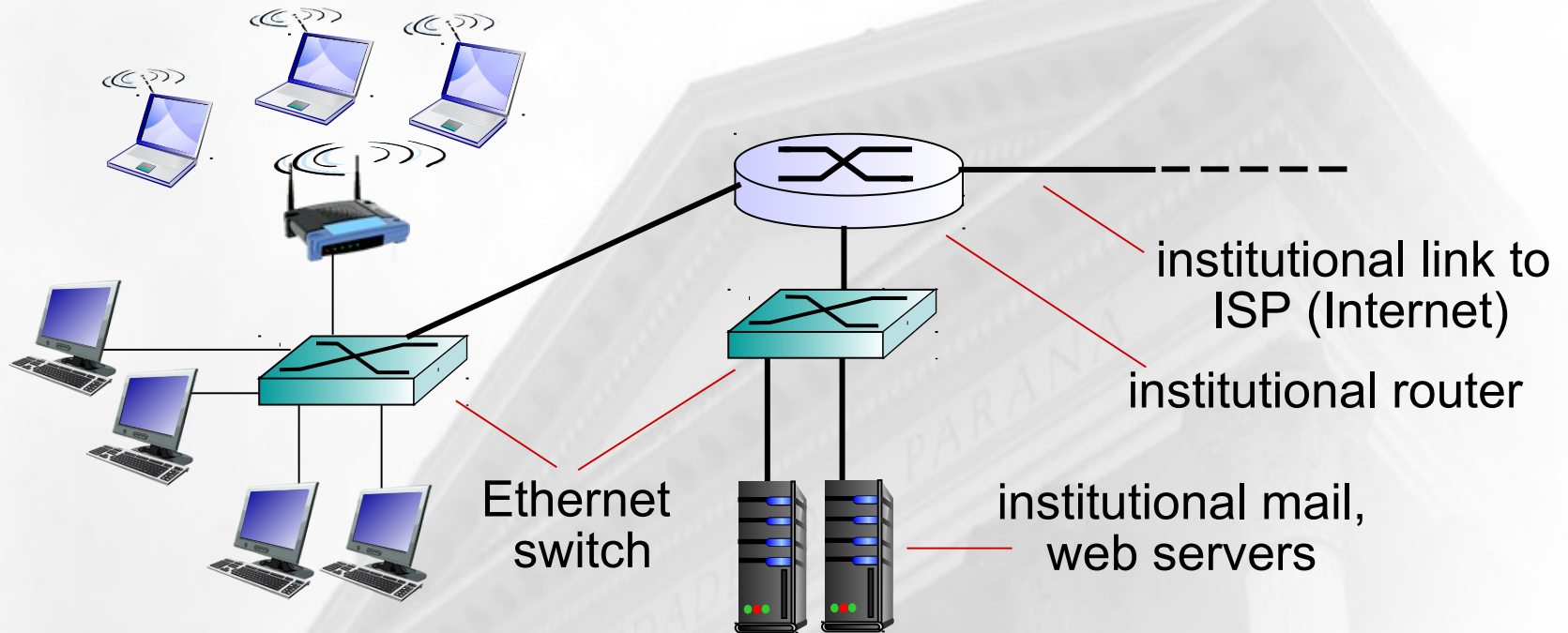
# Sobre a TV a cabo



# Rede residencial



# Rede corporativa



# Rede sem fio

- ❖ Access point (Base station)

## *wireless LANs:*

- 802.11 b/g/n/ac (WiFi)



*to Internet*

## Wireless WAN

- 3G, 4G



*to Internet*

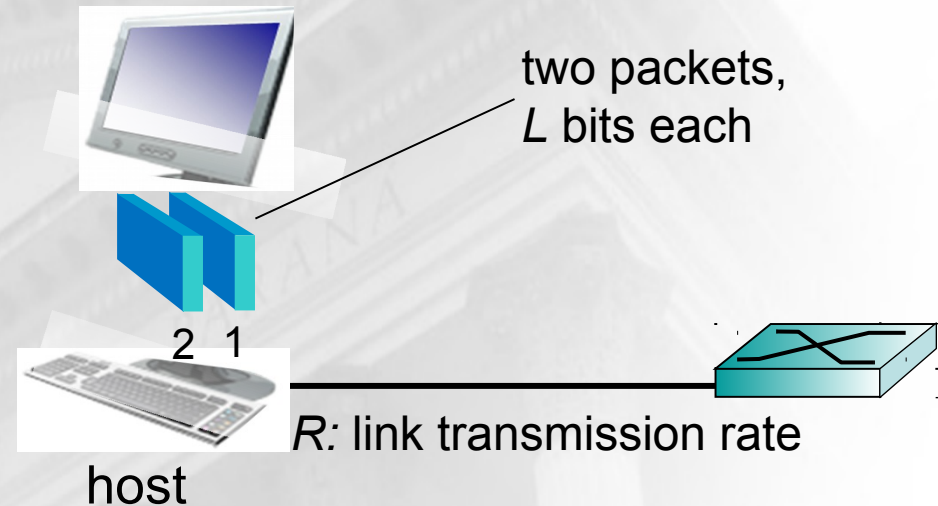
DSBD



# Host: Envia *pacotes* de dados

Host:

- ❖ Recebe as mensagens das aplicações
- ❖ Quebra em pedaços, chamados de pacotes, com um determinado número de bits (L)
- ❖ Transmite os pacotes na rede de acessos em uma taxa R
  - Taxa de transmissão do link, ou capacidade, ou largura de banda (*bandwidth*)



$$\begin{array}{l} \text{Atraso} \\ \text{De transmissão} \end{array} = \frac{\text{Tempo para transmitir}}{\text{L bits no canal}} = \frac{L \text{ (bits)}}{R \text{ (bits/sec)}}$$

# Meio físico

- ❖ Guiados e não guiados
- ❖ Determinam os limites de comunicação



# Meio físico: TP

*twisted pair (TP)*

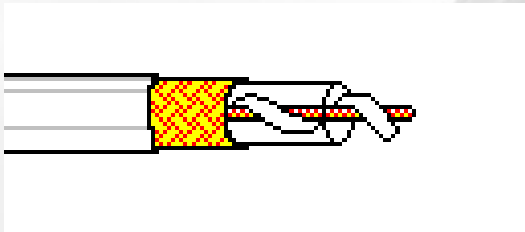
- UTP/STP
- Diversas categorias
- Ethernet 100M (CAT 3); 1GB (CAT 5e); 10GB (CAT 6a)



BD

# Meio físico

## *Cabo coaxial*



## *Fibra ótica*

- ❖ Grande largura de banda
- ❖ Baixa taxa de erro
  - Poucos repetidores
  - Imunidade à Ruído

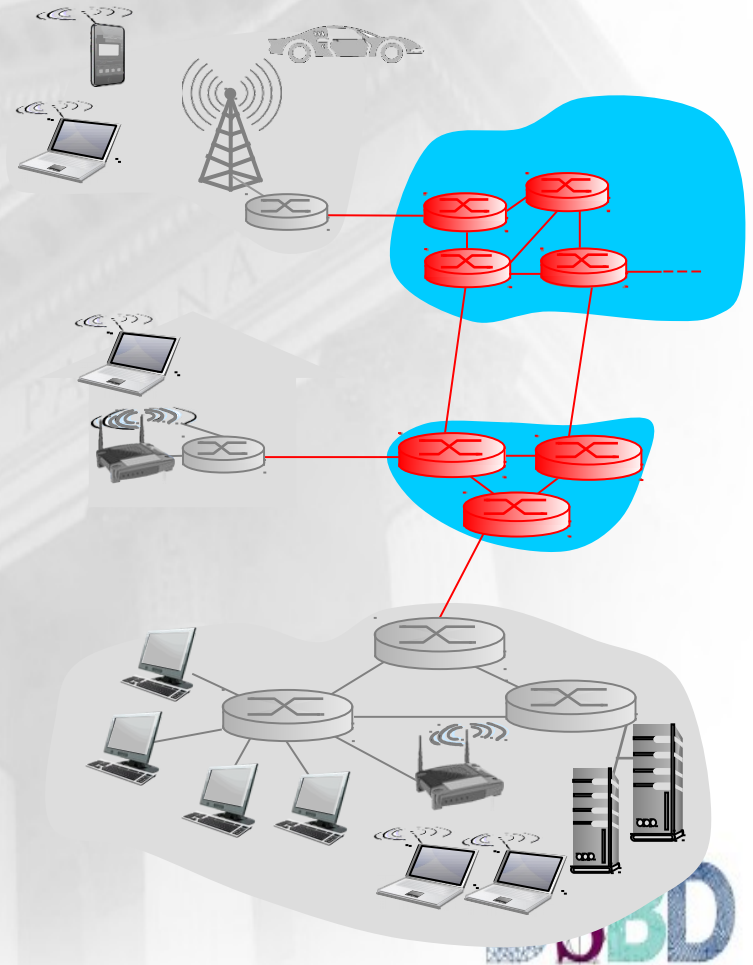
# Meio físico: Rádio

## *Alguns exemplos:*

- ❖ LAN (WiFi)
  - 11Mbps, 54 Mbps (2.4GHz / 5 GHz)
- ❖ WAN (cellular)
- ❖ Satélite
  - 270 msec atraso ponta-a-ponta

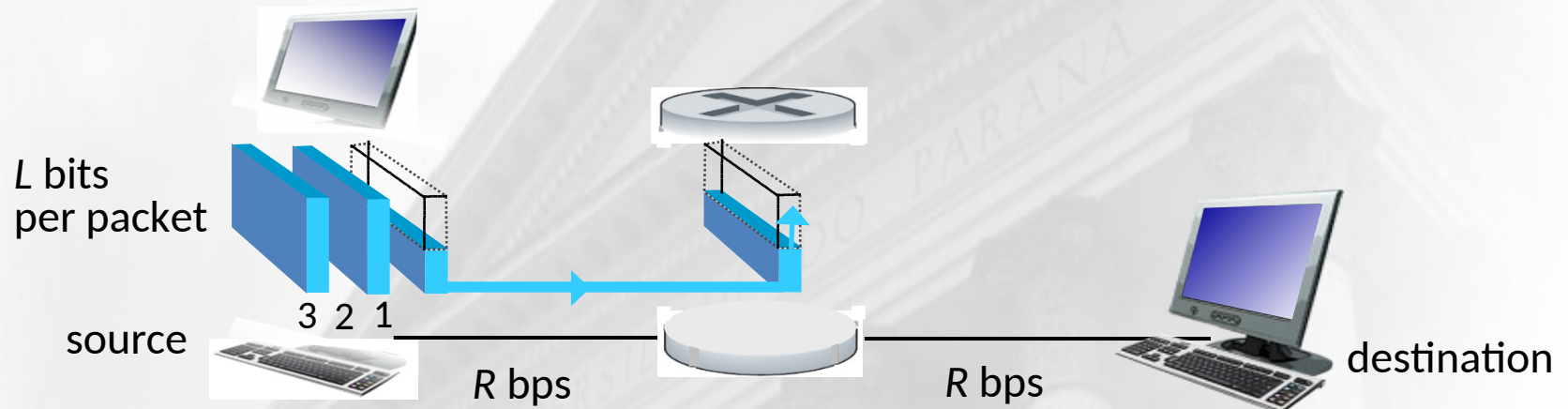
# Núcleo da rede

- ❖ Malha de roteadores interconectados
- ❖ packet-switching: hosts quebram as mensagens das aplicações em pacotes
  - Encaminham (*forward*) pacotes de um roteador para o próximo através dos enlaces (*links*) da origem até o destino
  - Cada pacotes transmits na melhor velocidade possível

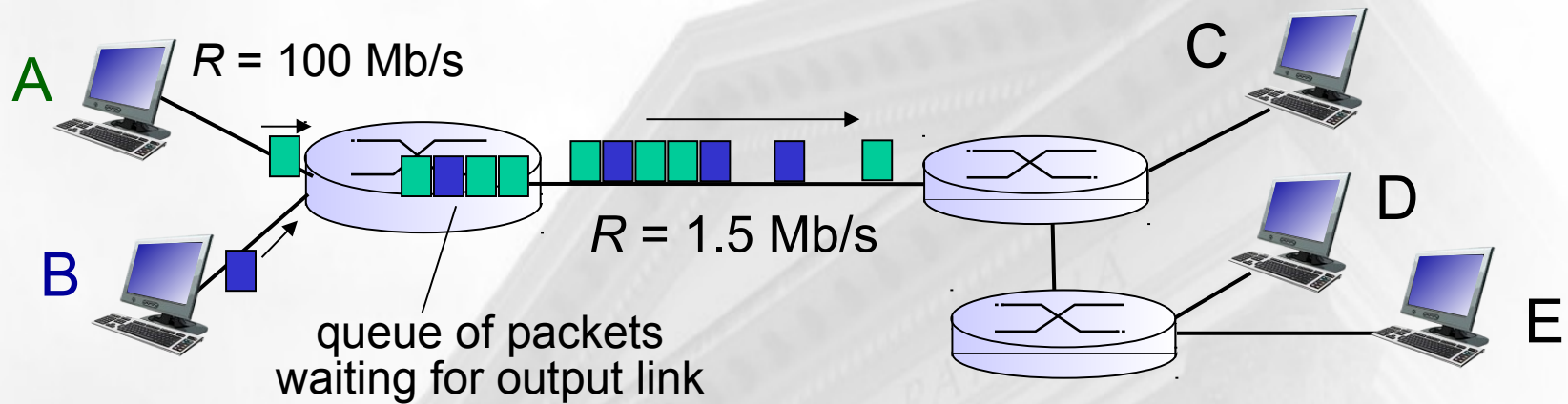




# Packet-switching



# Packet Switching

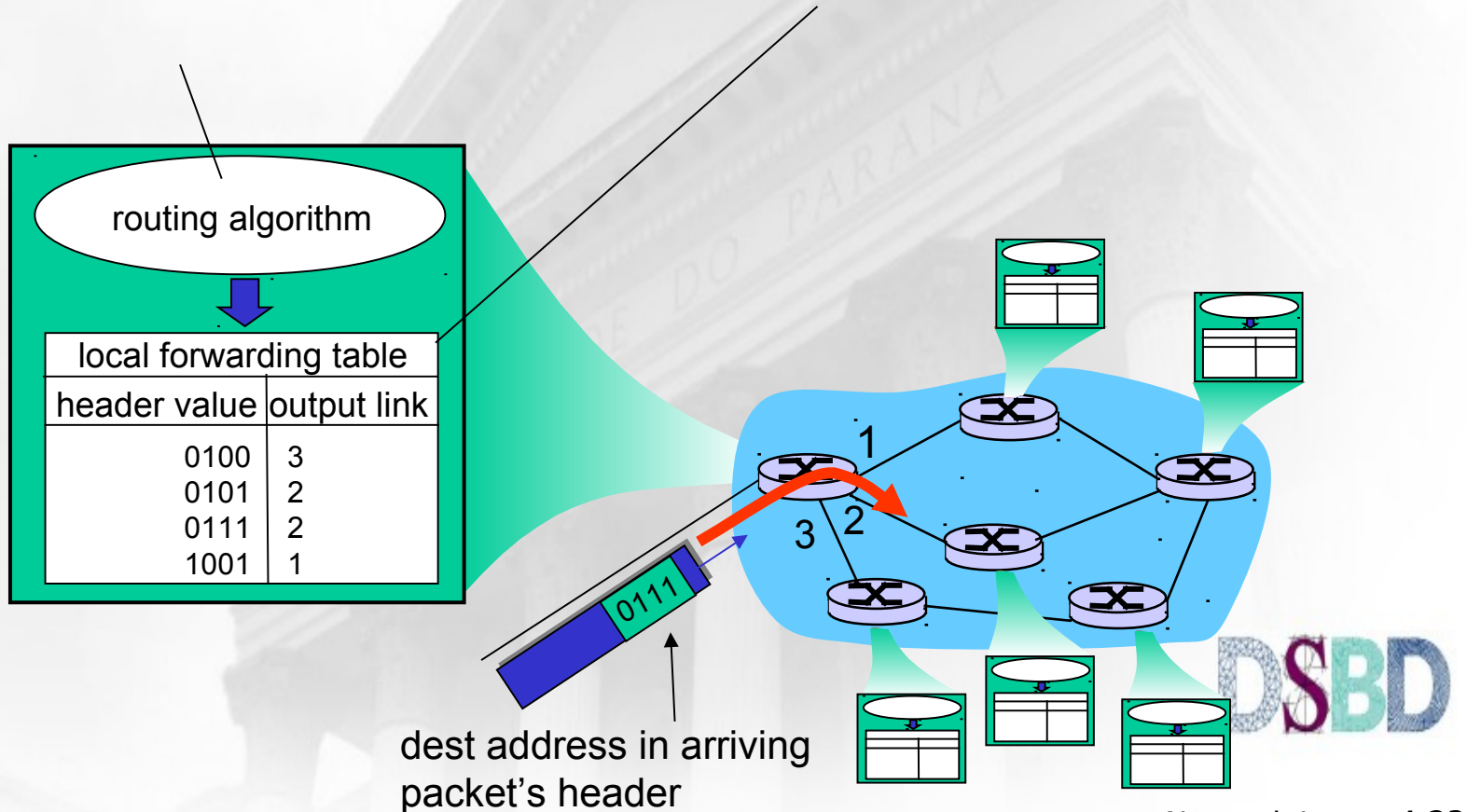


Enfileiramento e perda

# Funções principais

# routing

*forwarding*

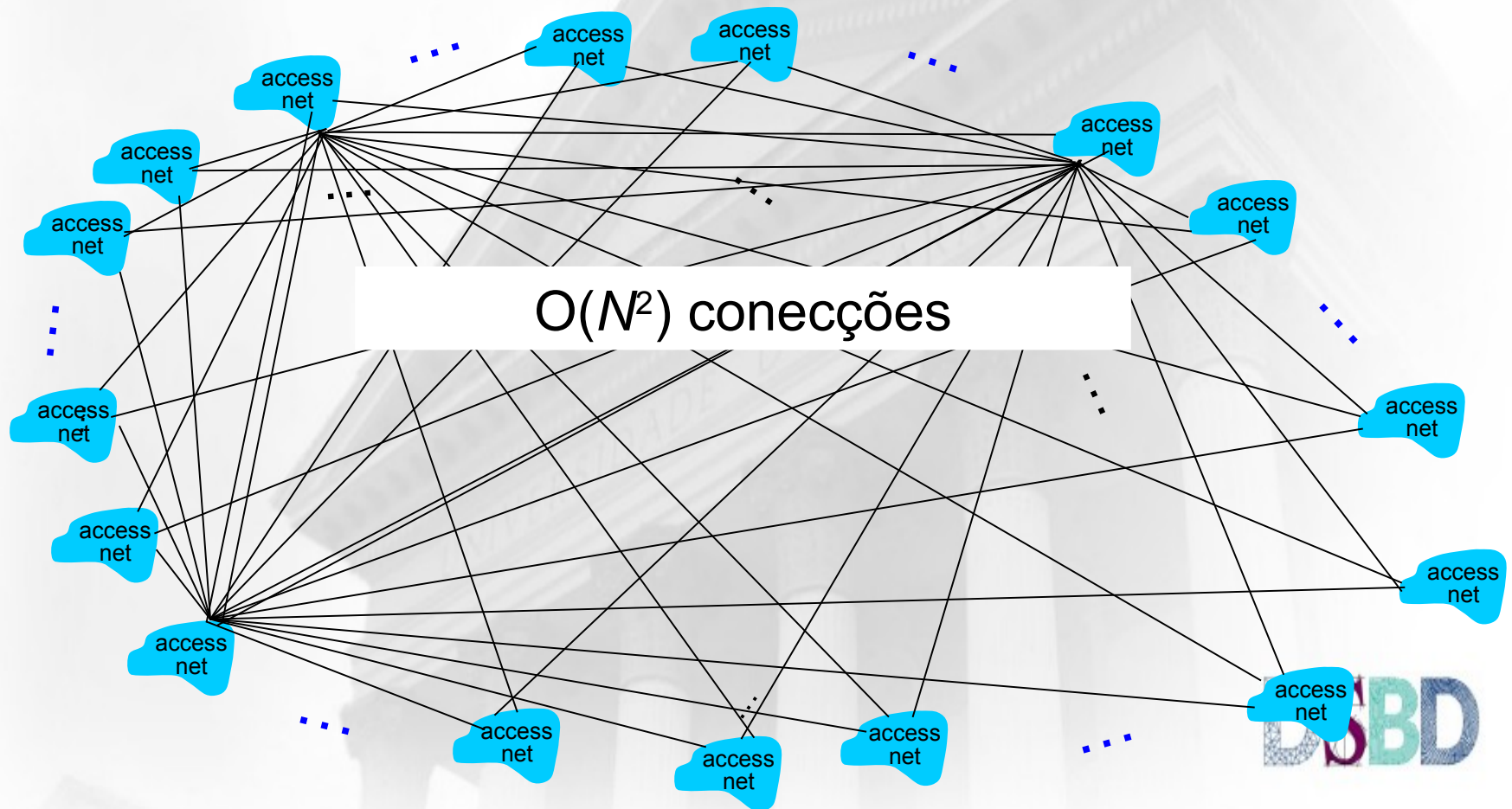


# Estrutura da Internet

- ❖ Hosts se conectam à Internet via provedores de serviço de Internet (**ISPs** - Internet Service Providers)
- ❖ ISPs precisam se interconectar. Mas como?

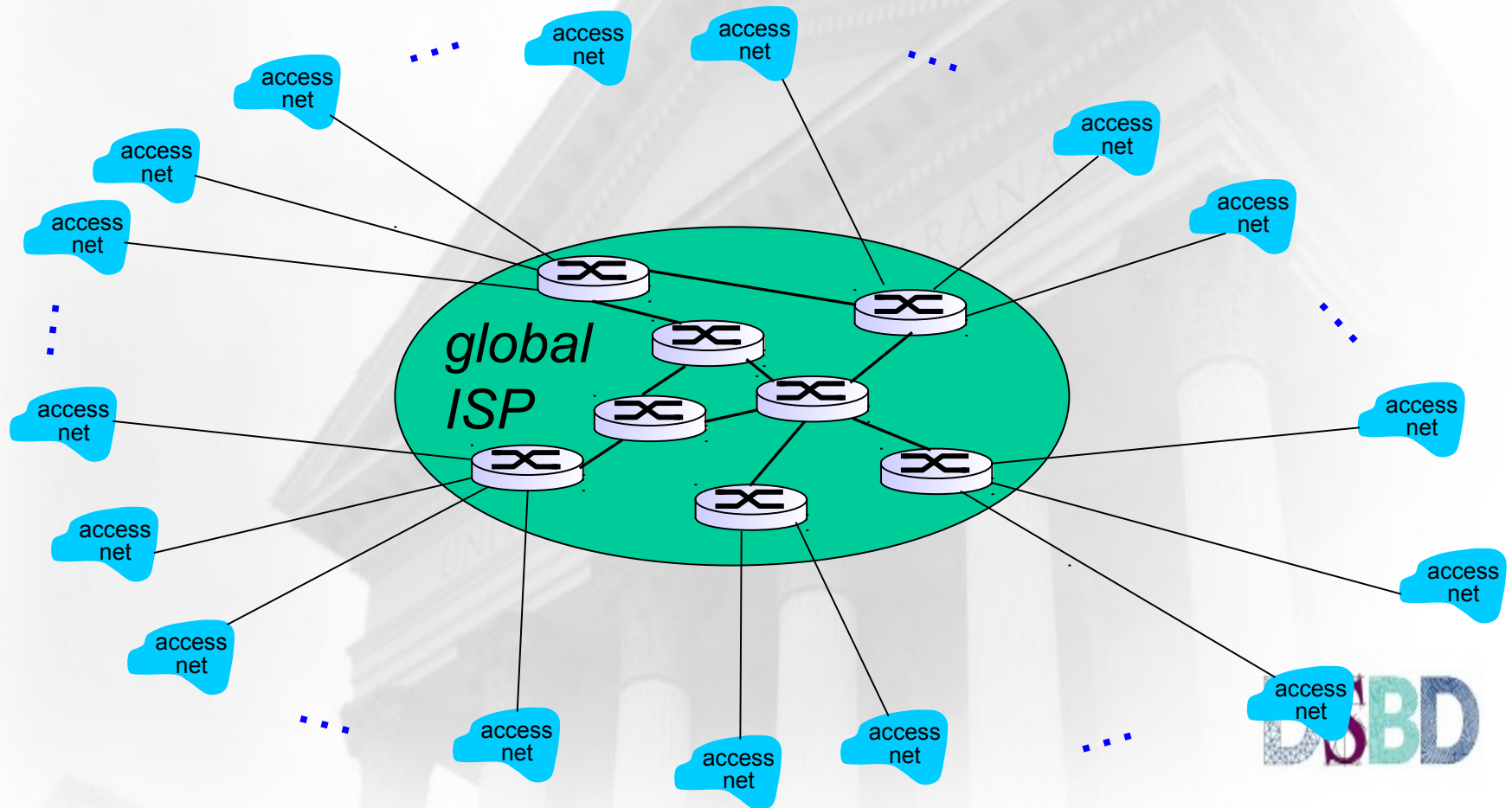
# Estrutura da Internet

*Conectar cada ISP com todos os outros*



# Estrutura da Internet

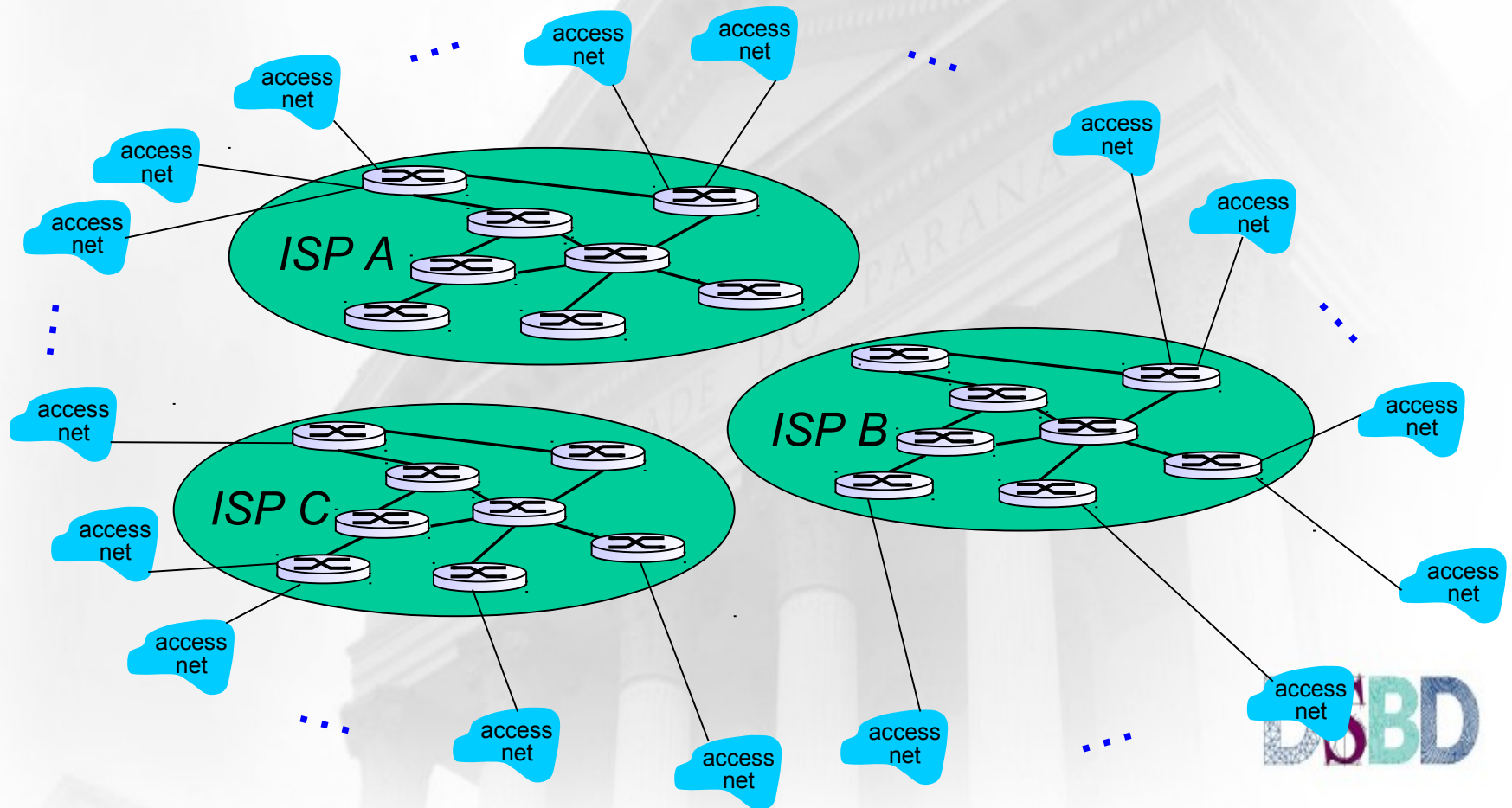
*Criar um ISP global*





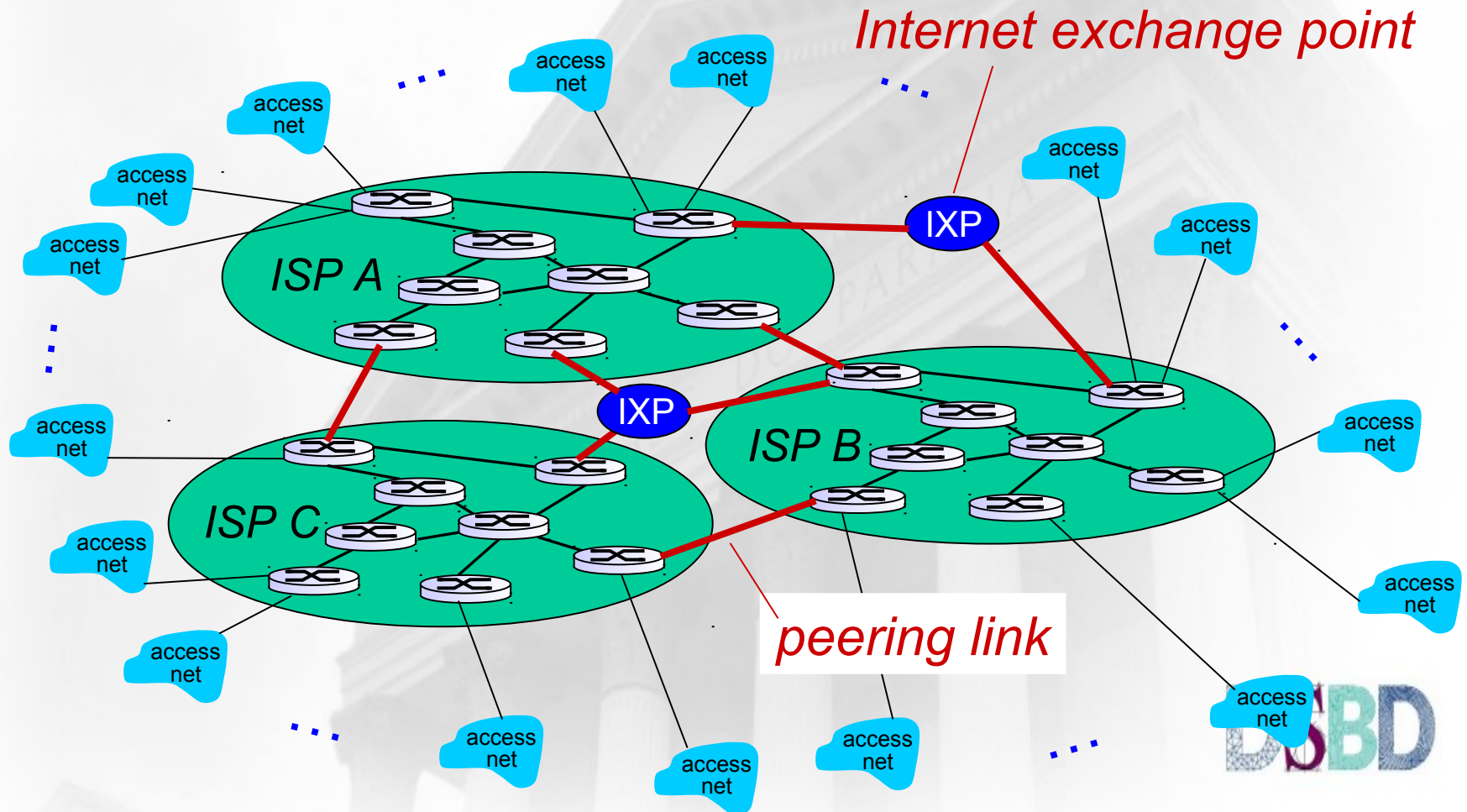
# Estrutura da Internet

Por que não vários?



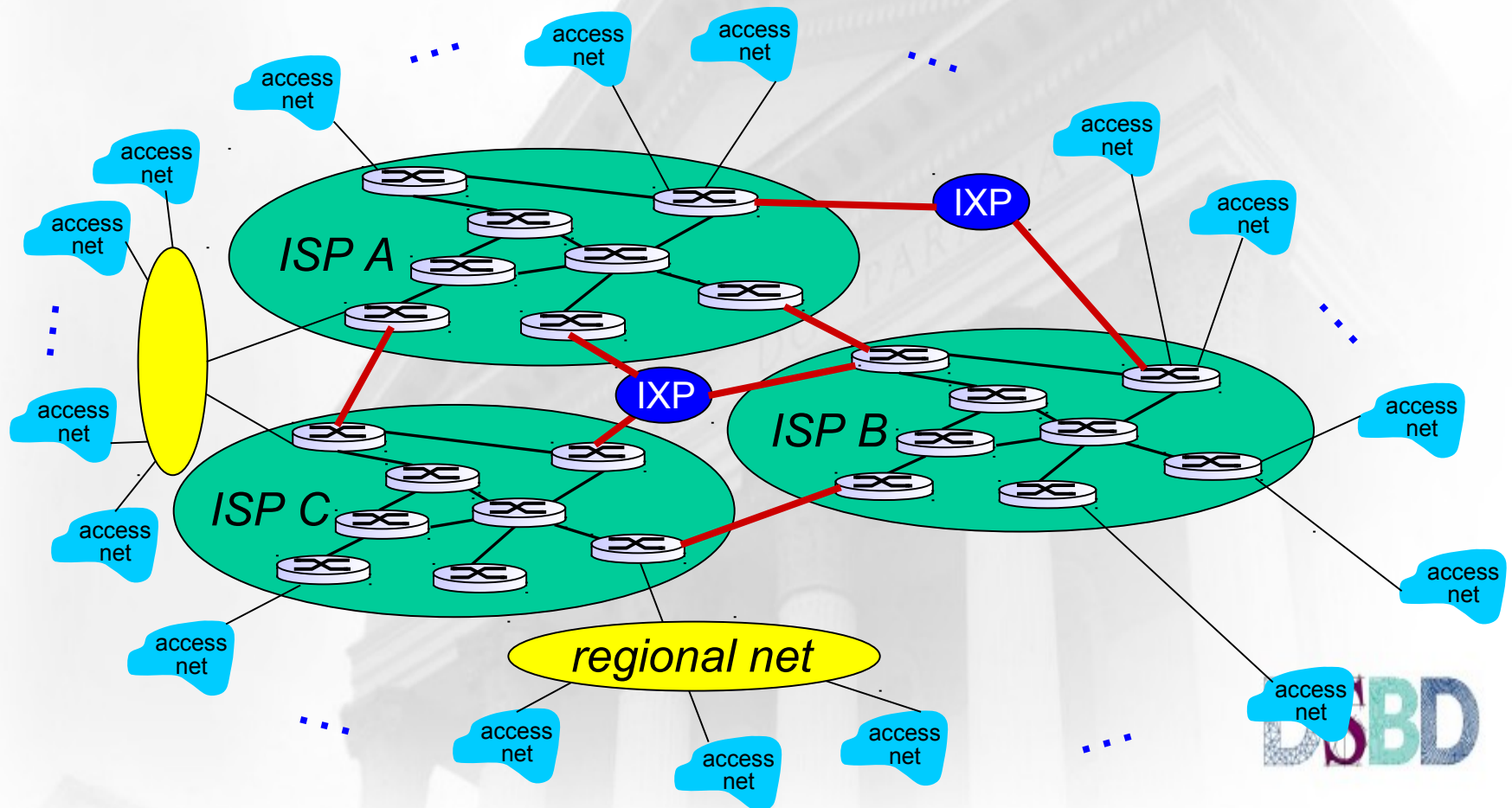
# Estrutura da Rede

Mas agora esses ISPs precisam se conectar



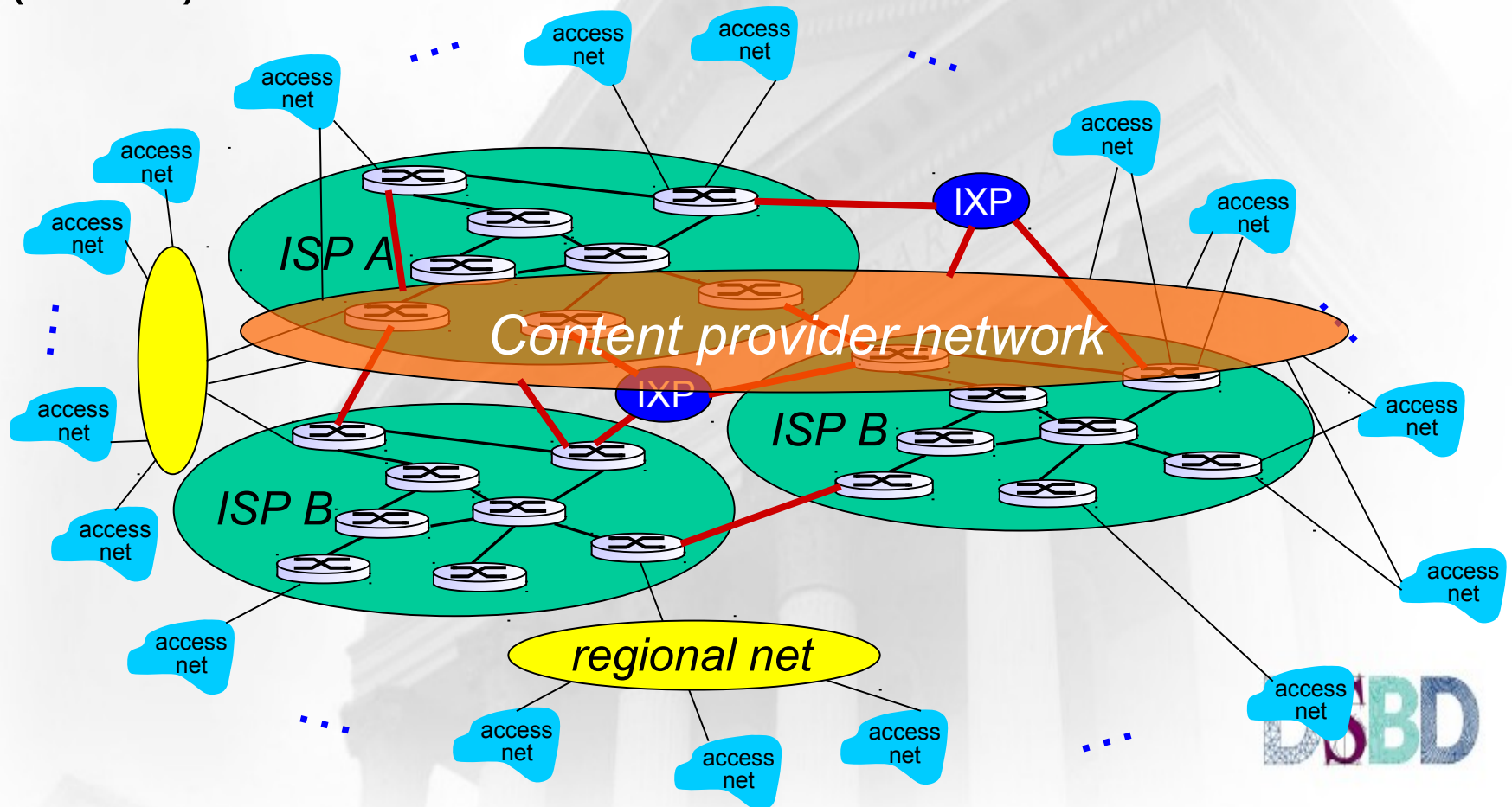
# Estrutura da Internet

Podemos acrescentar redes regionais

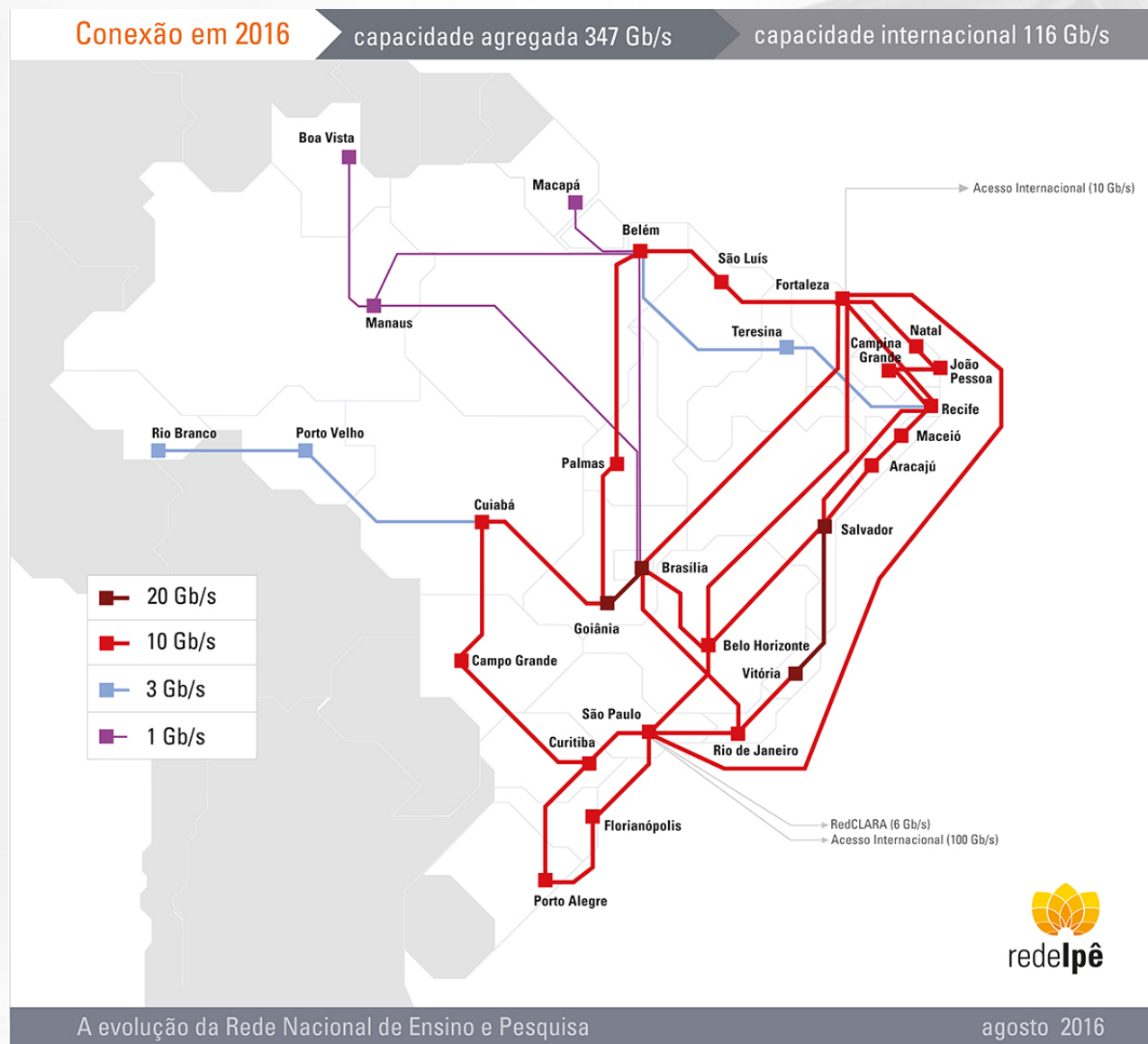


# Estrutura da Internet

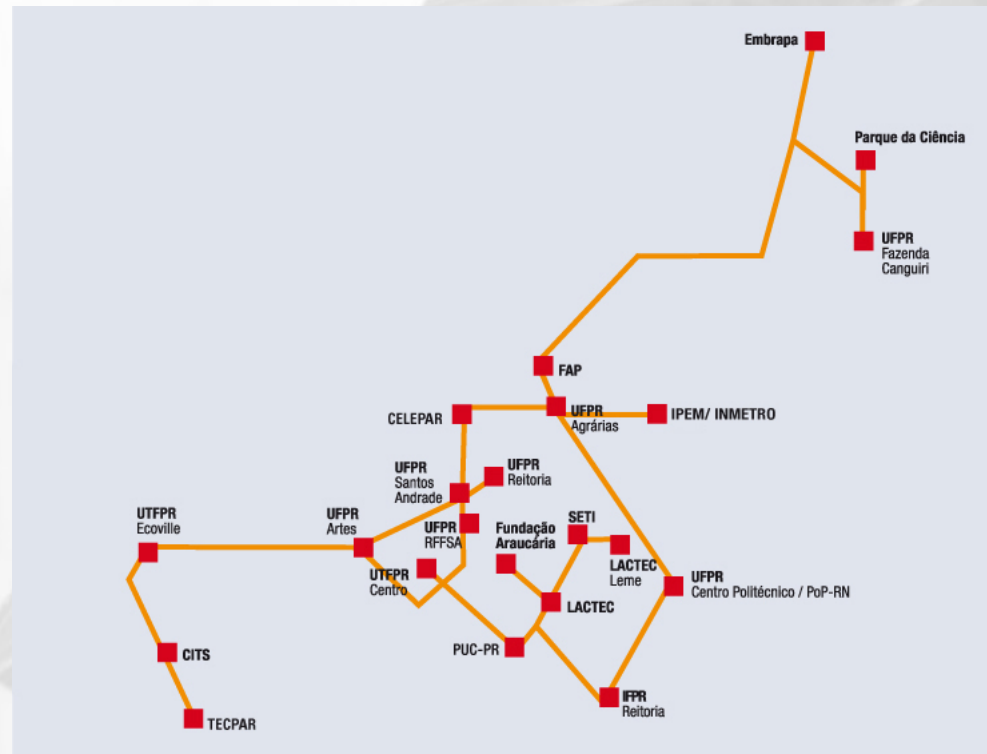
E ainda provedores de conteúdo para com sua rede trazer o conteúdo próximo aos seus usuários (CDNs)



# REDE Ipe - RNP



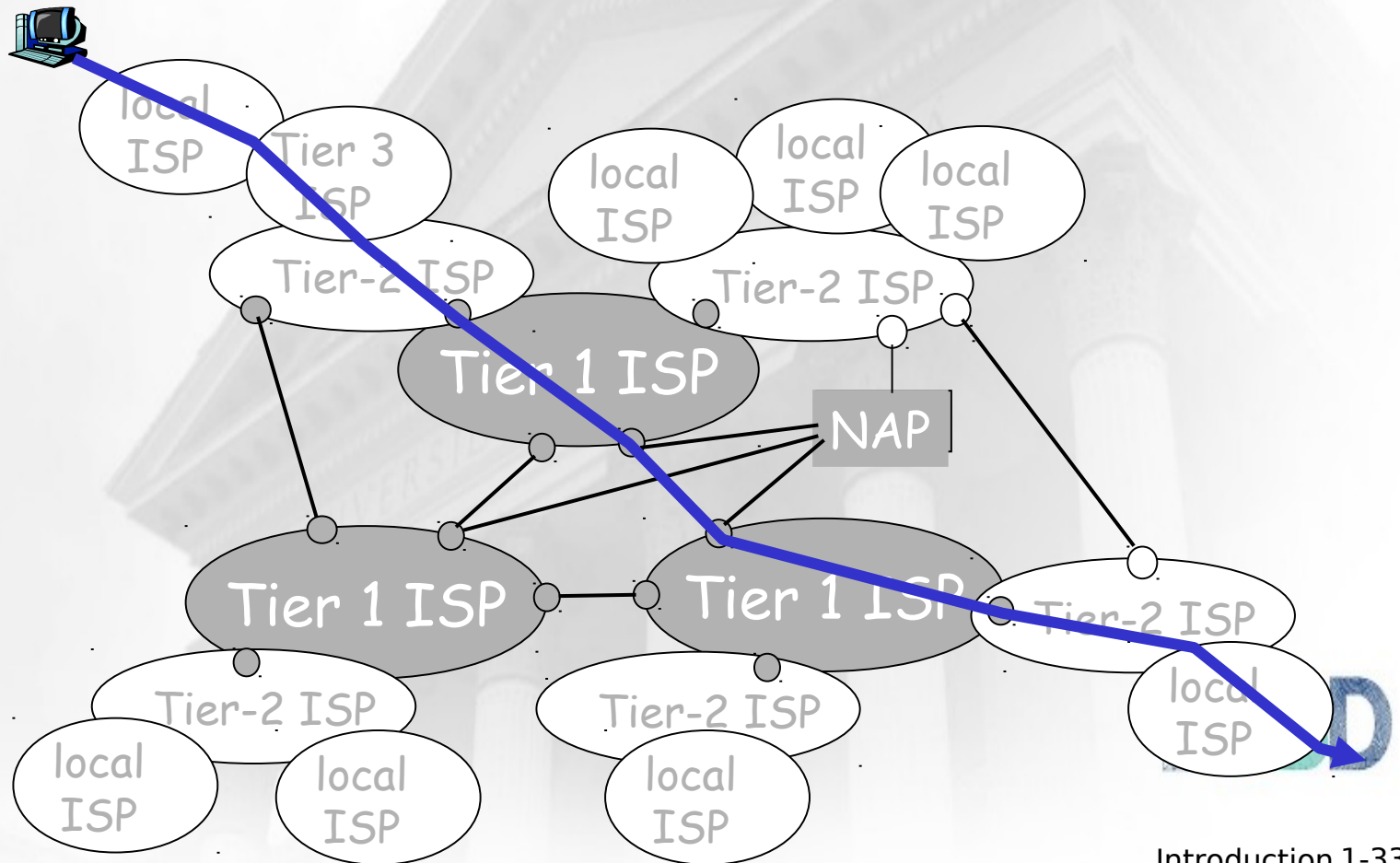
# Rede COMEP





# Estrutura da Internet

- ❖ Como observar parte desses caminhos?
- ❖ Traceroute / tracepath



# “Camadas” de protocolo

*Redes are complexas com muitas “partes”*

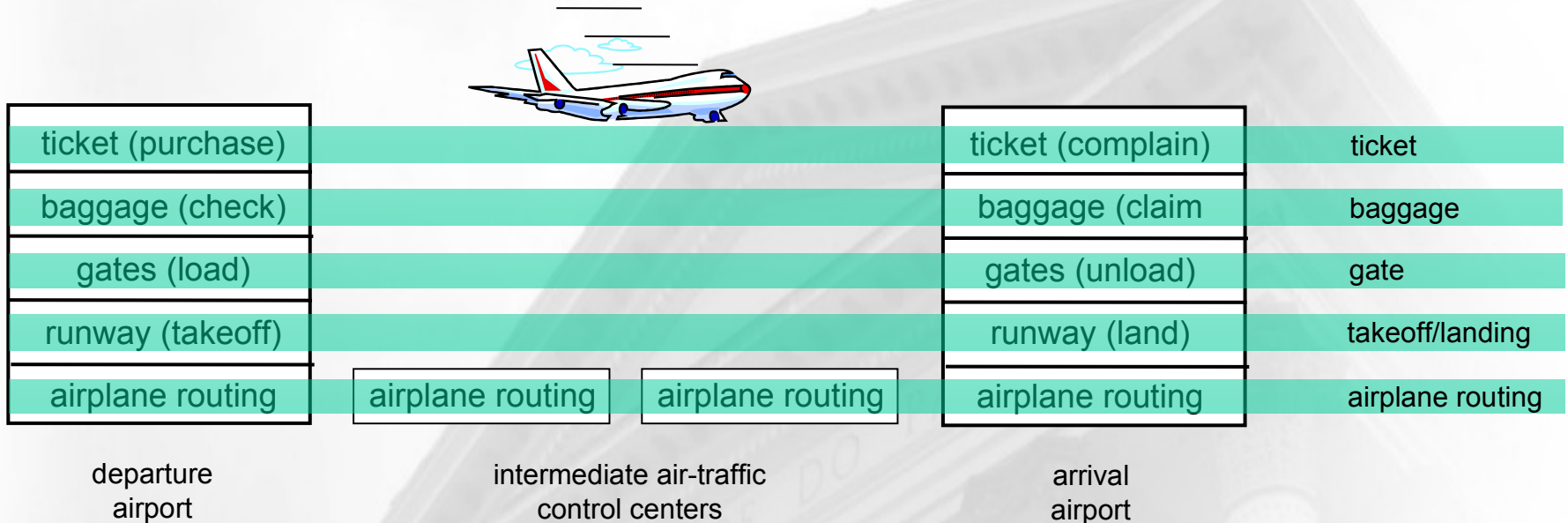
- *Meios físicos*
- *Hosts*
- *Aplicações*
- *Protocolos*
- *Hardware*
- *...*

*Questão: \_*

Como organizar essa complexidade?



# Camadas (layers)



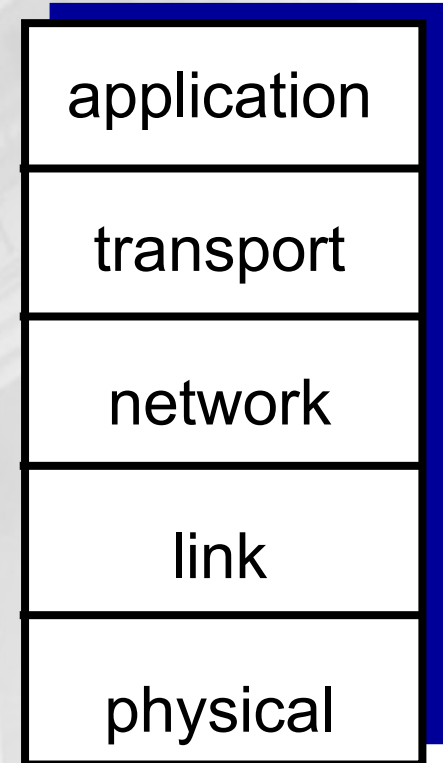
*camadas: cada camada implementa um serviço*

- Através de suas funções internas
- Confiando nos serviços proporcionados pelas camadas inferiores

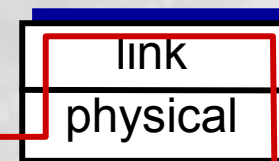
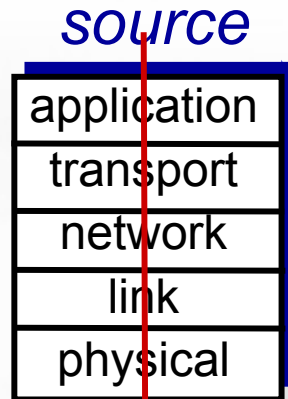
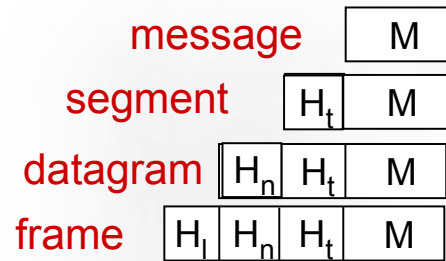


# Pilha de protocolos da Internet

- ❖ *Aplicação*
  - FTP, SMTP, HTTP
- ❖ *Transporte:* process-process
  - TCP, UDP
- ❖ *Rede:* roteamento
  - IP, roteamento
- ❖ *Enlace:* Interface com outras redes
  - Ethernet, 802.11 (WiFi), PPP
- ❖ *Físico:* Bits no “fio”

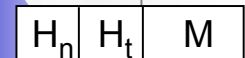
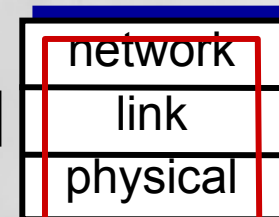
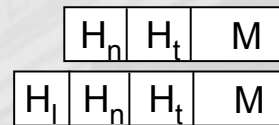
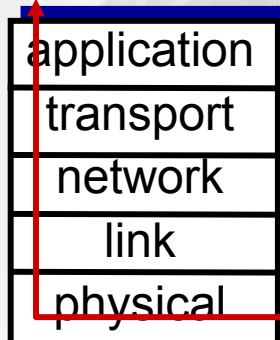


# Encapsulamento



switch

*destination*



router

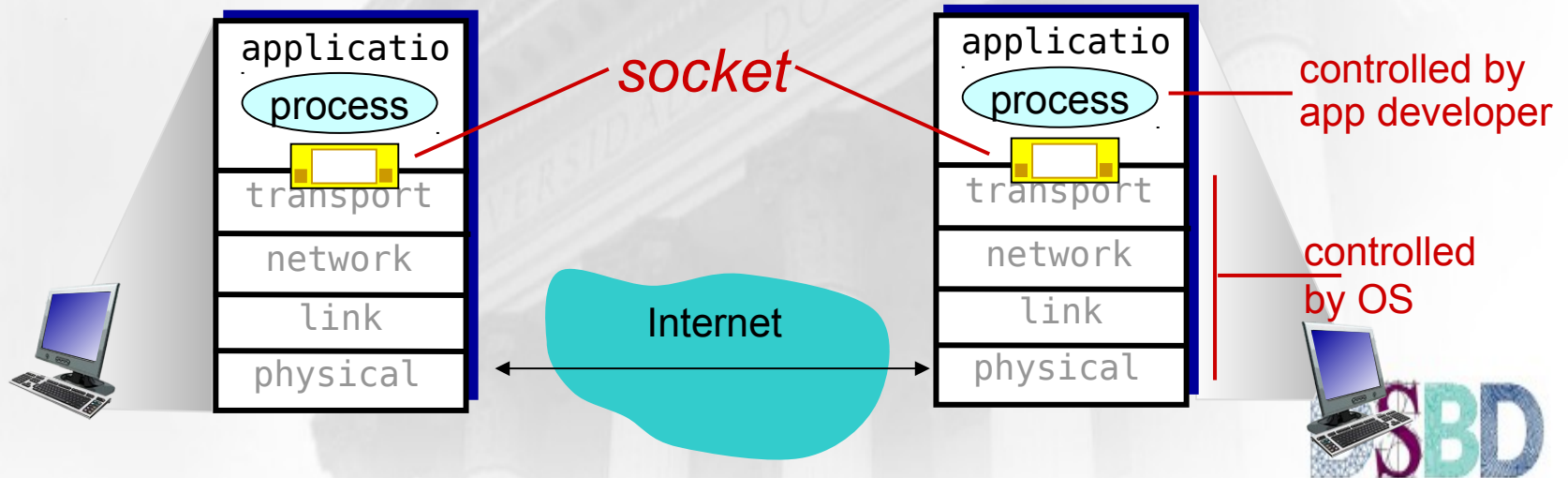
DSBD

# Camada de Aplicação

- ❖ Infinitude de aplicações que se comunicam utilizando um determinado protocolo pela rede
- ❖ HTTP, Whatsapp, Torrent, DNS, HTTP, Webservices, Rsync, SSH, NFS, Skype, SMTP, SIP
- ❖ Abstração para o programador: *sockets*
- ❖ Vamos chamar de processos as aplicações executando nos *hosts*

# Sockets

- ❖ Sockets operam usando os serviços da camada de transporte





# Endereçando “processos”

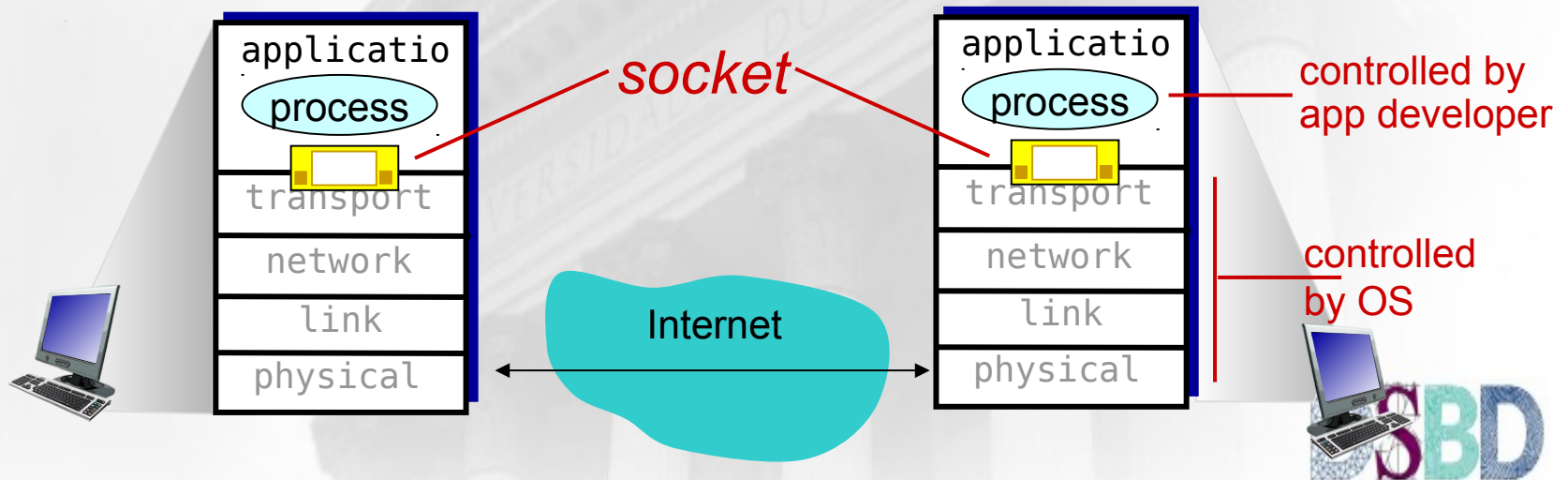
- ❖ Todo host tem um endereço IP
- ❖ Mas um único *host* pode abrigar diferentes processos
- ❖ Conceito de porta (port) resolve esse problema

# Endereçando “processos”

- ❖ Alguns serviços da rede tem portas padrões para operar
  - SSH 22
  - HTTP 80
  - TELNET 23
  - FTP 21
  - DNS 53
  - IMAP2 143 / IMAPS 993
- ❖ IANA (Internet Assigned Numbers Authority)

# Sockets

- ❖ Sockets operam a cima da camada de transporte



# Sockets

- ❖ A camada de transporte oferece dois serviços básico
- ❖ UDP e TCP

# HTTP

- ❖ Um dos protocolos mais populares
- ❖ Porta *default* 80
- ❖ Servidor: apache, nginx, ...
- ❖ Cliente: Navegadores web, mas pode ser linha de comando (wget)

# HTTP

- ❖ Uma página web é composta de objetos
- ❖ Objetos podem ser HTML, arquivos, imagens, etc
- ❖ Um arquivo base (index) que referencia vários outros objetos
- ❖ Cada objeto possui uma identificação (URL)
- ❖ `hostname{:port}/path&parameters`
- ❖ Mensagens de requisição (request) e resposta (response)

# HTTP request

```
GET /index.html HTTP/1.1\r\n
Host: www-net.cs.umass.edu\r\n
User-Agent: Firefox/3.6.10\r\n
Accept: text/html,application/xhtml+xml\r\n
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7\r\n
Keep-Alive: 115\r\n
Connection: keep-alive\r\n
\r\n
```



# HTTP response

```
GET /index.html HTTP/1.1\r\n
Host: www-net.cs.umass.edu\r\n
User-Agent: Firefox/3.6.10\r\n
Accept: text/html,application/xhtml+xml\r\n
Accept-Language: en-us,en;q=0.5\r\n
Accept-Encoding: gzip,deflate\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7\r\n
Keep-Alive: 115\r\n
Connection: keep-alive\r\n
\r\n
```

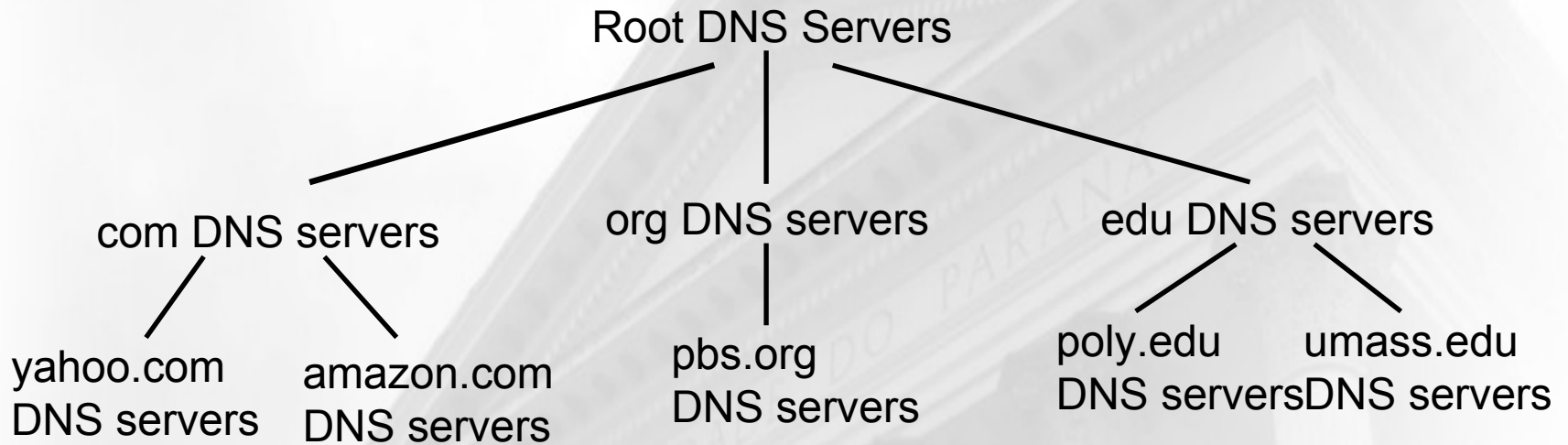
# HTTP

- ❖ Como é um protocolo em ASCII, você pode ser um “cliente”
- ❖ TELNET

# DNS

- ❖ Necessidade de ter nomes para os “hosts”
- ❖ Na Internet inicial: */etc/hosts*
- ❖ Domain Name System
  - É uma aplicação, parte da filosofia da Internet
  - Banco de dados distribuídos

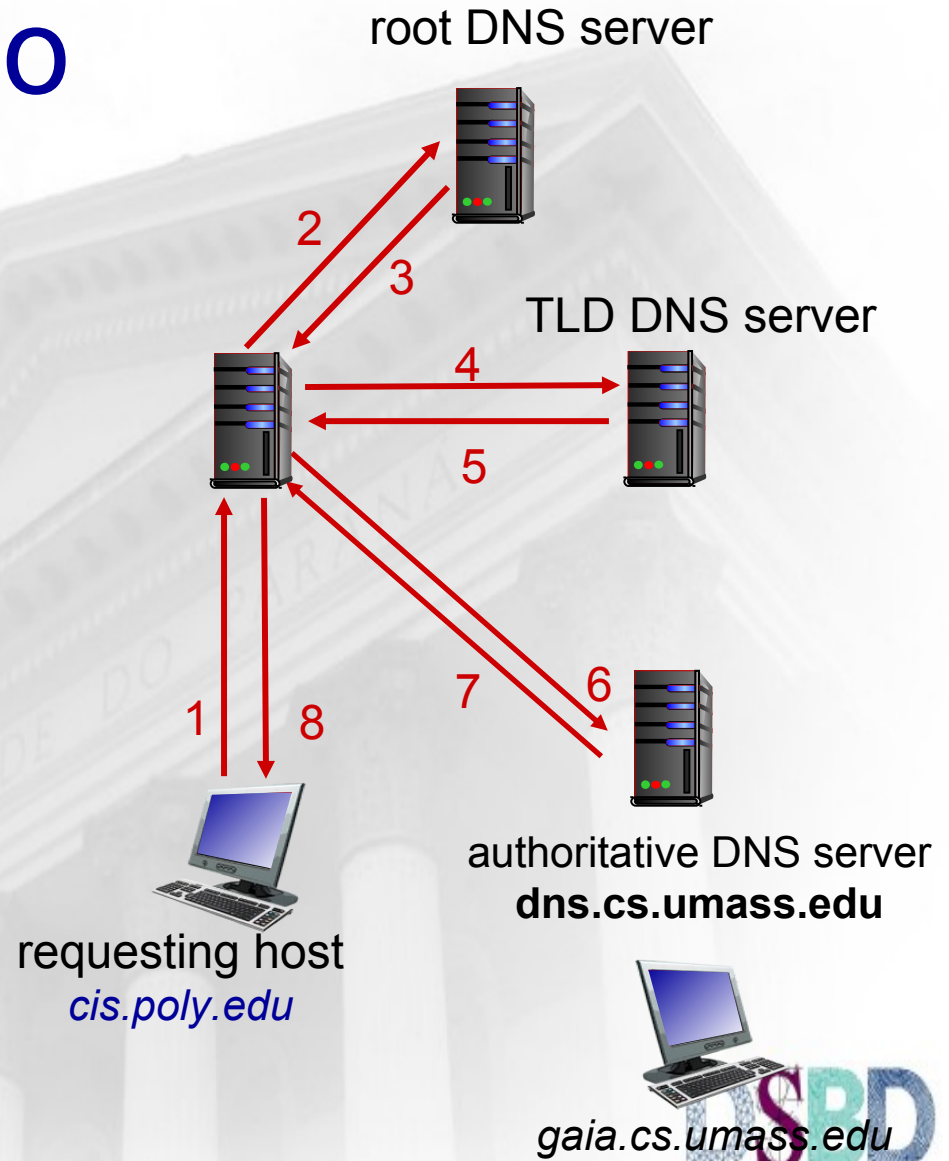
# DNS, organização



# DNS

- ❖ Top-level domain (TLD) servers
  - Responsáveis pelos domínios de países, e outros como com, org, net, edu
- ❖ Authoritative DNS
  - Organizações que possuem seus próprios servidores DNS
- ❖ DNS local
  - Normalmente o chamado “default name server”

# DNS, exemplo



# DNS: cache

- ❖ Como garantir escalabilidade?
  - Diminuindo o número de requisições
- ❖ Introduzir um tempo de vida para cada resposta (TTL)
- ❖ Quanto deve ser esse tempo?



# DNS Dinâmico

- ❖ Existem serviços gratuitos e pagos na rede que oferecem serviços de DNS dinâmico
- ❖ Este serviço permite que você registre um nome e oferecem uma interface para atualização do IP
- ❖ A dificuldade é o fato do IP ser dinâmico
  - Ainda existem outras quanto ao Firewall, IPv4 ou IPv6. Mas isso veremos depois...

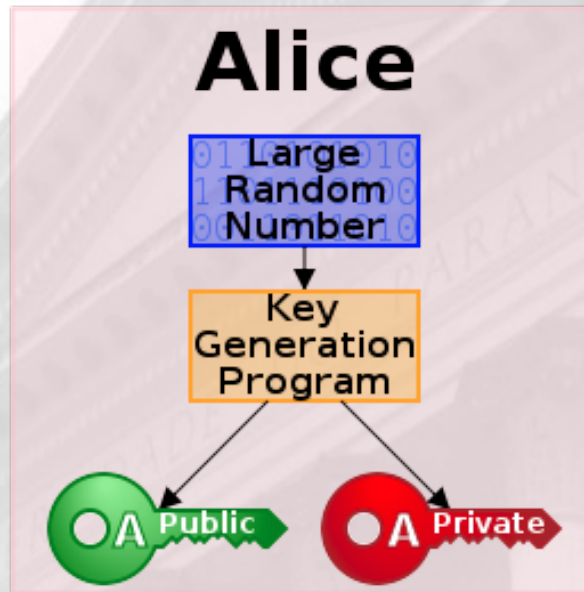
# SSH

- ❖ Serviço de rede seguro sobre uma rede insegura
- ❖ Principal serviço é de “login” remoto (secure shell) (substituto do *telnet*)
- ❖ Mas pode prover muitos outros serviços
  - Sistema de arquivo remoto
  - Tunelamento de portas
  - Base para VPNs
  - FTP seguro

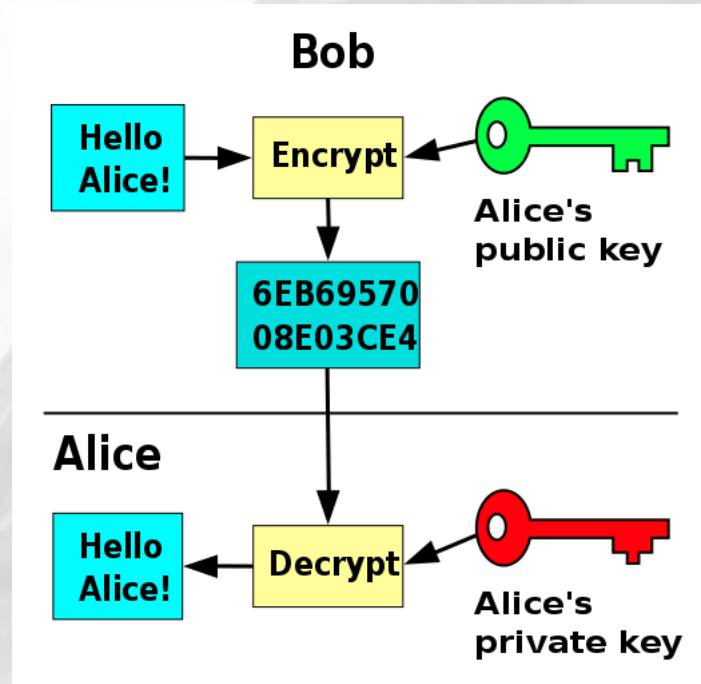
# SSH

- ❖ Vários métodos de autenticação
- ❖ O principal é por senha ou por chaves públicas e privadas
- ❖ Entretanto senha pode ser considerado um modo bastante vulnerável, já que sua senha pode ser capturada ou “advinhada”

# SSH – Chaves Assimétricas



# SSH – Chaves Assimétricas



# SSH

## ❖ Autenticação via chaves

- Cliente envia identificação para servidor SSH
- Servidor busca a chave do usuário e cria um desafio (criptografa uma mensagem)
- O Cliente vence o desafio usando sua chave privada

# SSH

## ❖ Na prática

- No host (seu computador) de onde se deseja fazer o acesso
  - `ssh-keygen -t rsa`
    - \* você vai entrar uma passphrase
    - \* será gerado o par em `/home/user/.ssh/`
- No host que será acessado
  - Copiar a chave pública para dentro do arquivo `.ssh/authorized_keys`

Se você cuida de um servidor exposto na Internet é uma boa ideia desabilitar o acesso por password



# SSH FS

- ❖ Disponibilizar arquivos remotos localmente
- ❖ Na prática
  - No host (seu computador) de onde se deseja fazer o acesso
    - sshfs **conta@host**:caminho ponto\_montagem
  - Ao terminar
    - fusermount -u ponto\_montagem
- ❖ Também pode usar ssftp e comandos scp

# SSH – Outros truques

- ❖ Algumas ferramentas fazem o tunelamento “automático” através do ssh
  - VNC
  - X11
  - Sockets5 proxy
  - Rsync