

Introdução - Redes de Computadores

Prática em Sistemas Operacionais

João Vicente Ferreira Lima

Departamento de Linguagens e Sistemas de Computação

Universidade Federal de Santa Maria

jvlima@inf.ufsm.br

2021/2

Sumário

1 Fundamentos em Redes

2 Ferramentas

3 Bibliografia

Sumário

1 Fundamentos em Redes

- Introdução
- Protocolos em camadas
- Camada de rede
- Camada de transporte
- Camada de aplicação
- Domain Name Server (DNS)
- Sumário

2 Ferramentas

3 Bibliografia

- Introdução

- Protocolos em camadas
- Camada de rede
- Camada de transporte
- Camada de aplicação
- Domain Name Server (DNS)
- Sumário

- Ping

- Dig
- Nmap
- Sipcalc

3 Bibliografia

- **Desempenho** - latência, taxa de transferência.
- **Escalabilidade** - tamanho, geográfica e administrativa.
- **Confiabilidade** - erros em hardware e software.
- **Segurança** - firewall, criptografia, VPNs (*virtual private network*).
- **Mobilidade** - dispositivos móveis e laptops.
- **Qualidade de Serviço (QoS)** - largura de banda e latência.
- **Multicasting** (*difusão seletiva*) - comunicação um-para-muitos.

- Introdução

- Camada de rede

- Camada de transporte

- Camada de aplicação

- Domain Name Server (DNS)

● Sumário

- Ping

- Dig

- Nmap

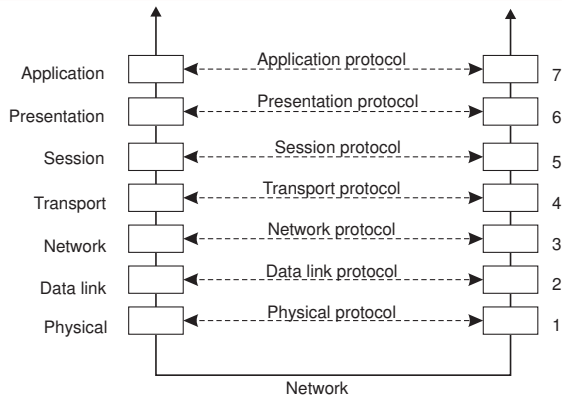
- Sipcalc

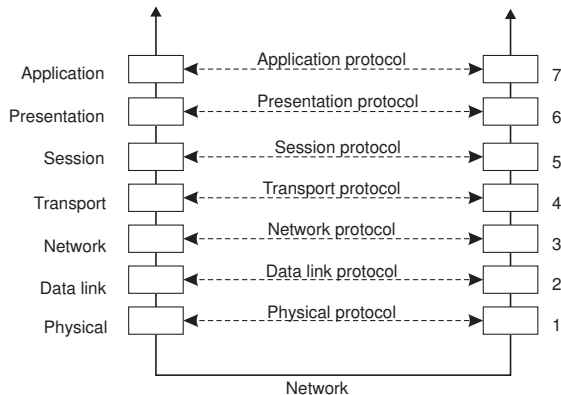
3 Bibliografia

Fundamentos

- Toda comunicação em sistemas distribuídos é baseada em mensagens (de baixo nível)
- Para que a comunicação seja bem sucedida deve haver **regras**
 - Essas regras são **protocolos**
 - Diversos aspectos a considerar (tamanho da mensagem, erros, pacotes, etc)

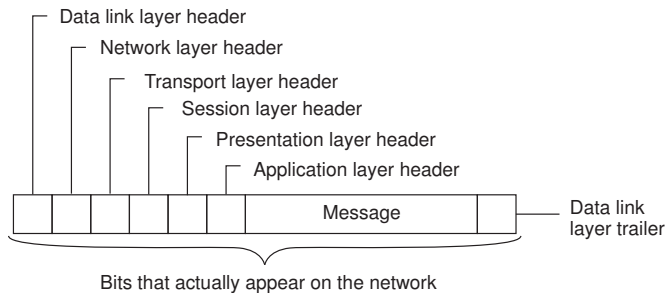
Modelo de referência básico

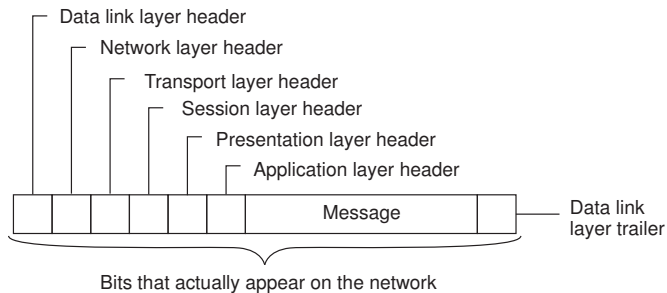




Modelo OSI

- **Modelo de referência para interconexão de sistemas abertos** ou **ISO OSI**
- Nunca foram amplamente utilizados





Protocolos em camadas

- Protocolos de níveis mais baixos
- Protocolos de transporte
- Protocolos de níveis mais altos
- Protocolos de middleware

- **Camada física** - especificação e implementação de bits, além da transmissão. Ex: Ethernet, Wi-Fi, etc.
- **Camada de enlace** - efetua a transmissão de uma série de bits agrupados em **quadros** (*frame*), bem como sequencia e verificação de erros. Ex: MAC, ARP, etc.
- **Camada de rede** - descreve como pacotes em uma rede de computadores são **roteados**. Ex: ICMP, IP, etc.



Protocolos de níveis mais baixos

- **Camada física** - especificação e implementação de bits, além da transmissão. Ex: Ethernet, Wi-Fi, etc.
- **Camada de enlace** - efetua a transmissão de uma série de bits agrupados em **quadros** (*frame*), bem como sequencia e verificação de erros. Ex: MAC, ARP, etc.
- **Camada de rede** - descreve como pacotes em uma rede de computadores são **roteados**. Ex: ICMP, IP, etc.

Observação

Para muitos sistemas distribuídos, a camada mais baixa é a camada de rede.

Observação

O protocolo de rede mais utilizado atualmente é o **protocolo de Internet (Internet Protocol - IP)**, sem conexão que faz parte da pilha de protocolos da Internet. Um **pacote** IP pode ser enviado sem preparação alguma.

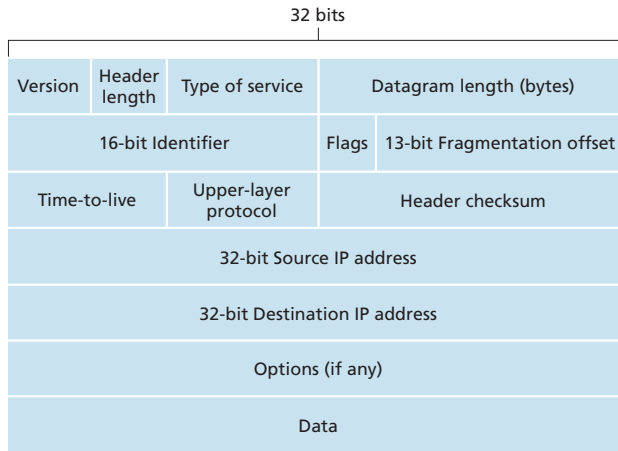


- Introdução
- Protocolos em camadas
- **Camada de rede**
- Camada de transporte
- Camada de aplicação
- Domain Name Server (DNS)
- Sumário

- Ping
- Dig
- Nmap
- Sipcalc

3 Bibliografia





IP version 4 (IPv4)

- Cabeçalho IP tem 20 bytes e TCP 20 bytes = 40 bytes + outras camadas.
- **IP Datagram Fragmentation** - os pacotes IP são fragmentados para transmissão
- O máximo de dados em um quadro da camada de enlace é o *maximum transmission unit (MTU)*
 - O tamanho do frame Ethernet é de **1500 bytes**

- Endereçamento de 32-bit = 2^{32} endereços IP.
- De 0.0.0.0 até 255.255.255.255.

Endereçamento com 32-bit

11000001 00100000 11011000 00001001

equivalente ao IP

193.32.21.216.9

- CIDR generaliza a noção de subrede (subnet).
- Divide um endereço 32-bit em duas partes com formato `a.b.c.d/x` onde `x` indica o número de bits da primeira parte.
- A primeira parte é o prefixo (**prefix**).

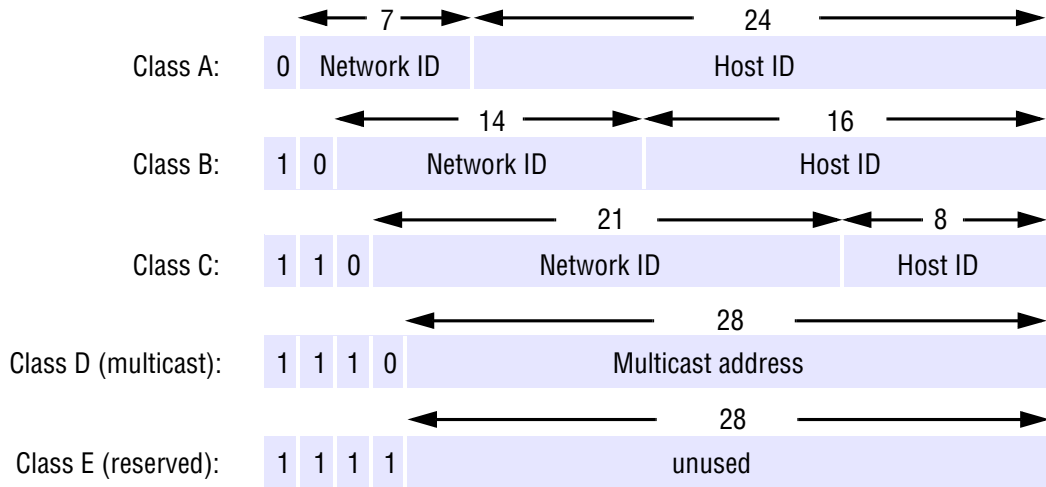
- 10.1.1.0/24 denota todos os endereços entre 10.1.1.0 e 10.1.1.255.
- O primeiro endereço (10.1.1.0) é reservado para identificar a rede (*network address*).
- O último endereço (10.1.1.255) é reservado para broadcast (*broadcast address*).
- Os outros podem ser usados para computadores.
- Normalmente, o 10.1.1.1 é o router da rede.

- 223.1.1.32/28 denota todos os endereços entre 223.1.1.32 e 223.1.1.47.
- O primeiro endereço é reservado para identificar a rede (*network address*).
- O último endereço é reservado para broadcast (*broadcast address*).
- Os outros podem ser usados para computadores.

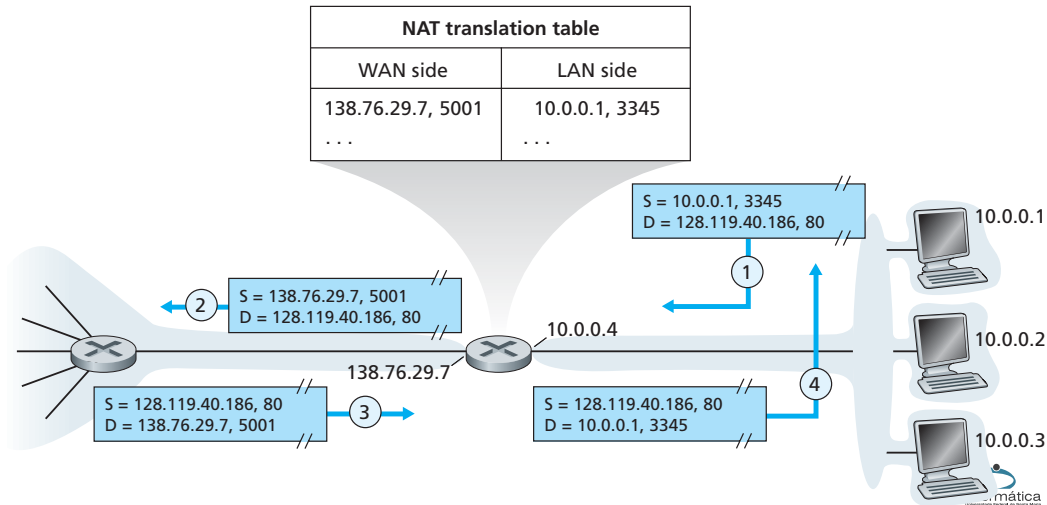
IP Classless InterDomain Routing (CIDR)

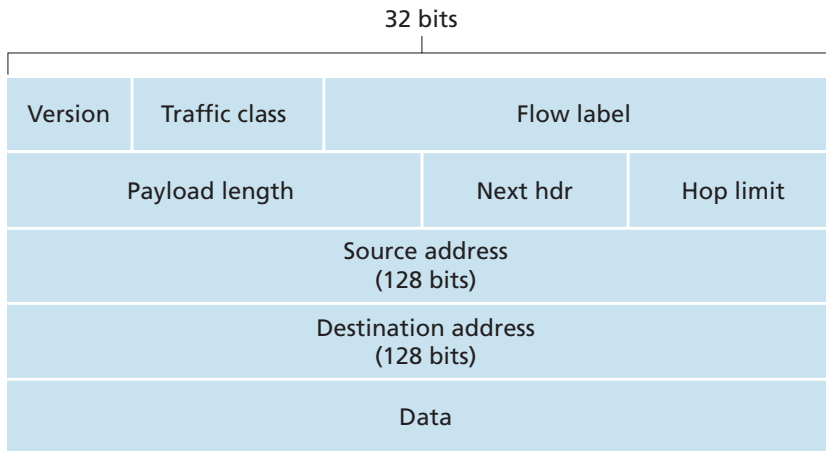
- **Classe A** - redes muito grandes e longa distância em nível nacional.
- **Classe B** - organizações com mais que 255 computadores.
- **Classe C** - usados nos demais tipos de redes.
- **Classe D** - reservado para multicasting.
- **Classe E** - reservado para uso futuro.

IP Classless InterDomain Routing (CIDR)









IP version 6 (IPv6)

Principais avanços

- 1 **Espaço de endereçamento** - 2^{128} (de 2^{32} do IPv4).
- 2 **Melhor desempenho de roteamento** - sem soma de verificação e sem fragmentação.
- 3 **Suporte a tempo real** e outros serviços.
- 4 **Evolução futura** - campo *próximo cabeçalho* (*next hdr*).
- 5 Difusão seletiva (*multicast*) e **não-seletiva** (**anycast**).
- 6 **Segurança** - até agora, só suportado na camada de aplicação.

Sumário

- 1 Fundamentos em Redes
 - Introdução
 - Protocolos em camadas
 - Camada de rede
 - Camada de transporte
 - Camada de aplicação
 - Domain Name Server (DNS)
 - Sumário

- 2 Ferramentas
 - Ping
 - Dig
 - Nmap
 - Sipcalc

- 3 Bibliografia

Protocolos de transporte

Última camada básica: fornece todas as funcionalidades que os níveis abaixo não fornecem para um uso razoável por uma aplicação de rede.

Protocolos de Internet *de facto*

- **TCP (Transmission Control Protocol)** - orientado a conexão, confiável, controle de fluxo, sequencial.
- **UDP (Universal Datagram Protocol)** - sem conexão, não confiável, não ordenado.

Nota

Ambos são cliente-servidor.

Protocolos de transporte

Última camada básica: fornece todas as funcionalidades que os níveis abaixo não fornecem para um uso razoável por uma aplicação de rede.

Protocolos de Internet *de facto*

- **TCP (Transmission Control Protocol)** - orientado a conexão, confiável, controle de fluxo, sequencial.
- **UDP (Universal Datagram Protocol)** - sem conexão, não confiável, não ordenado.

Nota

Ambos são cliente-servidor.

Níveis de dados

- ❶ **Nível 1 (física)** - bit stream.
- ❷ **Nível 2 (enlace)** - frames.
- ❸ **Nível 3 (rede)** - packets.
- ❹ **Nível 4 (transporte)** - TPDU (*Transport Protocol Data Unit*)
 - Segmentos (TCP).
 - Datagramas (UDP).

Portas e Sockets

Port - um número de 1 – 65535, regidos pela IANA.

- Portas bem conhecidas ficam entre 0 – 1024.
- Portas registradas ficam entre 1024 – 49151.
- Portas dinâmicas/privadas entre 49152 – 65535.

Socket

- Identifica um ponto final (*endpoint*) em uma comunicação.
- Caracterizado pela combinação:
 - Tipo de protocolo (TCP, UDP,).
 - Endereço IP local.
 - Porta local.

Constantes

```
1 socket.socket(family=AF_INET, type=SOCK_STREAM, proto=0, fileno=None)
```

- **Family:**

- AF_UNIX, AF_LOCAL - comunicação local
- AF_INET - protocolos de internet IPv4
- AF_INET6 - protocolos de internet IPv6

- **Type:**

- SOCK_STREAM - TCP (orientando a conexão)
- SOCK_DGRAM - UDP

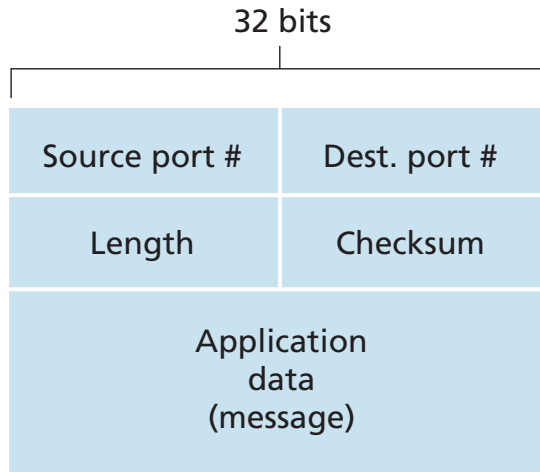
Protocolo UDP

- Mais rápido em termos de desempenho porque não mantém conexões.
- Adiciona muito pouco ao IP.
- **Não é confiável** - não garante envio de mensagens nem sua ordem.
- Uso - NFS, DNS, DHCP, jogos, streaming de audio e video.

Vantagens

- 1 **Controle fino** de dados enviados e recebidos.
- 2 **Sem conexões** (*connectionless*) - sem custo para estabelecer conexões.
- 3 **Sem estado de conexão.**
- 4 **Header pequeno** - TCP adiciona 20 bytes, enquanto o UDP adiciona 8 bytes.

Protocolo UDP



```
1  #!/usr/bin/env python3
2  import socket
3
4  udp_ip = '127.0.0.1'
5  udp_port = 5005
6  message = 'Hello_hello'
7
8  print("UDP_target_IP:" + udp_ip)
9  print("UDP_target_port:" + str(udp_port))
10 print("Message:" + message)
11
12 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
13 sock.sendto(str.encode(message), (udp_ip, udp_port))
```

UDP em Python

Parte do recebimento:

```
1  #!/usr/bin/env python3
2  import socket
3
4  udp_ip = '127.0.0.1'
5  udp_port = 5005
6  message = 'Hello_hello'
7
8  sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
9  sock.bind((udp_ip, udp_port))
10
11 while True:
12     data, addr = sock.recvfrom(1024)
13     print("Received:_" + str(data))
```

- ## Características

- 1 **Full-duplex** - transmissão nos dois sentidos.
- 2 **Point-to-point** - único remetente e único destinatário.
- 3 **Three-way handshake** - estabelece conexão em três passos.
- 4 **Ordenado** - ordem é garantida.
- 5 **Controle de fluxo** - caso o receptor é mais lento.
- 6 **Não tem** - timer, vazão mínima garantida, segurança.



Servidor TCP

```
1  #!/usr/bin/env python3
2  import socket
3
4  host = 'localhost'
5  port = 50007
6  s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
7  s.bind((host, port))
8  s.listen(1)
9  conn, addr = s.accept()
10 with conn:
11     print('Connected_by', addr)
12     while True:
13         data = conn.recv(1024)
14         if not data:
15             break
16         conn.sendall(data)
```



```
1  #!/usr/bin/env python3
2  import socket
3
4  host = 'localhost'
5  port = 50007
6
7  s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
8  s.connect((host, port))
9  s.sendall(b'Hello, world')
10 data = s.recv(1024)
11 print('Received', repr(data))
```

Sumário

- 1 Fundamentos em Redes
 - Introdução
 - Protocolos em camadas
 - Camada de rede
 - Camada de transporte
 - Camada de aplicação
 - Domain Name Server (DNS)
 - Sumário

- 2 Ferramentas
 - Ping
 - Dig
 - Nmap
 - Sipcalc

- 3 Bibliografia

Protocolos de níveis mais altos

- Na prática, acima da camada de transporte, somente a camada de aplicação é usada.
- Tudo acima dessa camada foi agrupado.

Protocolos de mais alto nível

- **Sessão** - Versão aprimorada da camada de transporte, que faz controle de diálogo e sincronização, raro suporte.
- **Apresentação** - Significado dos bits, define formato da mensagem.
- **Aplicação** - Pela OSI era para conter um conjunto de aplicações padronizadas de rede como email, transferência de arquivos e terminal. Mas ela se tornou repositório para todas as aplicações e protocolos restantes.

Protocolos de níveis mais altos

- O que falta nessa camada é uma clara distinção entre **aplicações**, **protocolos específicos de aplicação** e **protocolos de uso geral**.

Exemplos

- **File Transfer Protocol (FTP)** - transferência de arquivos entre cliente e servidor. Não confundir com a aplicação `ftp`.
- **HyperText Transfer Protocol (HTTP)** - transferência de páginas Web. Implementado em browsers Web (cliente) e servidores Web (servidor). Porém não é somente vinculado a Web como em objetos remotos em Java.

Protocolos de middleware

Observação

Middleware é concebido para prover **protocolos de uso geral** que podem ser usados por diferentes aplicações.

Um rico conjunto de **protocolos de comunicação**

- **(Un)marshaling** de dados, para sistemas integrados.
- **Protocolos de nomeação** para compartilhamento fácil de recursos.
- **Protocolos de segurança** para comunicação segura.
- **Mecanismo de escalabilidade** como replicação e caching.

Middleware é concebido para prover **protocolos de uso geral** que podem ser usados por diferentes aplicações.

- Chamada de procedimentos remotos.
- Invocar objetos remotos.
- Estabelecer e sincronizar fluxos para dados em tempo real.
- Oferecer serviços multicast confiáveis.

Sumário

- 1 Fundamentos em Redes
 - Introdução
 - Protocolos em camadas
 - Camada de rede
 - Camada de transporte
 - Camada de aplicação
 - Domain Name Server (DNS)
 - Sumário

- 2 Ferramentas
 - Ping
 - Dig
 - Nmap
 - Sipcalc

- 3 Bibliografia

Domain Name Server (DNS)

- Traduzir nomes (*hostname*) para endereços IP.
- Serviço de diretório, ou banco de dados distribuído, em hierarquia.
- Executa sobre o protocolo TCP ou UDP porta 53.

Outros serviços

- **Host aliasing** - “apelidos” de um host (o original é o **canonical hostname**).
- **Mail server aliasing** - servidor Web e Mail podem ter o mesmo nome.
- **Load distribution** - distribuição de trabalho, ou seja, quando um cliente consulta um nome com vários IPs, ele retorna todos e alterna a ordem deles. O cliente normalmente usa o primeiro IP.

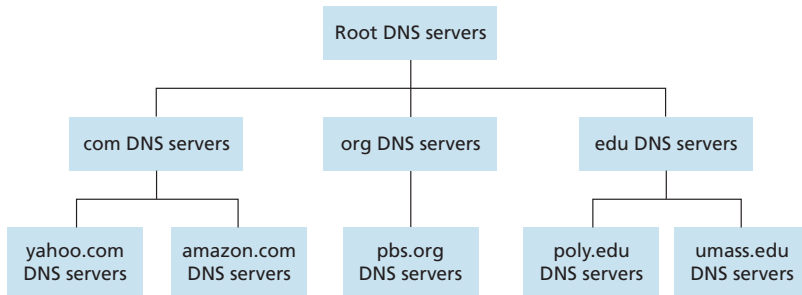
Domain Name Server (DNS)

- Traduzir nomes (*hostname*) para endereços IP.
- Serviço de diretório, ou banco de dados distribuído, em hierarquia.
- Executa sobre o protocolo TCP ou UDP porta 53.

Outros serviços

- **Host aliasing** - “apelidos” de um host (o original é o **canonical hostname**).
- **Mail server aliasing** - servidor Web e Mail podem ter o mesmo nome.
- **Load distribution** - distribuição de trabalho, ou seja, quando um cliente consulta um nome com vários IPs, ele retorna todos e alterna a ordem deles. O cliente normalmente usa o primeiro IP.

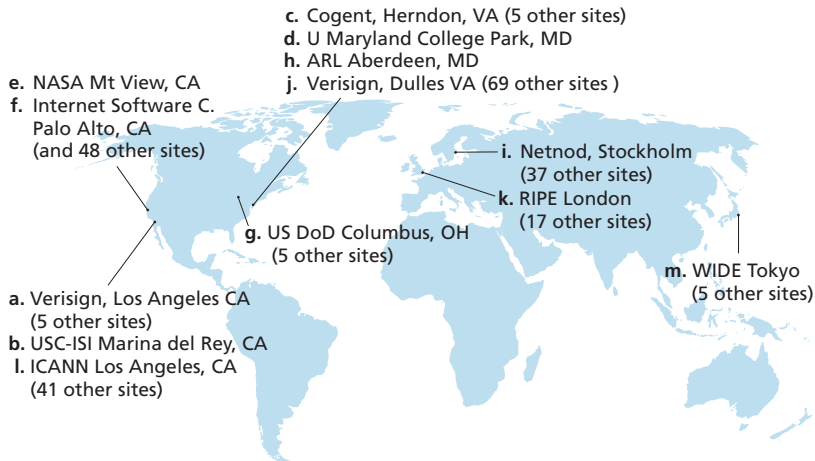
- **SOA** - *Start of Authority*, informações de domínio como serial number, e prazos.
- **NS** - *name server*.
- **A ou AAAA** - *IP address* IPv4 ou IPv6.
- **MX** - *SMTP mail exchangers*, servidor de email do domínio.
- **PTR** - *reverse DNS lookup*.
- **CNAME** - *domain name aliases*.



Hierarquia DNS

- **Root DNS servers** - 13 servidores raízes, maioria na América do Norte. Porém cada servidor é uma rede de duplicatas sendo todos aproximadamente 247 servidores.
- **Top-level domain (TLD) servers** - Domínios `com`, `org`, `net`, `edu` e `gov`, e de países como `uk`, `fr`, `ca`, `br`, etc.
- **Authoritative DNS servers** - Organizações com acesso aberto devem prover entradas DNS. Universidades e grandes companhias mantem seus próprios servidores DNS.

Servidores raiz do DNS



Sumário

1 Fundamentos em Redes

2 Ferramentas

- Ping
- Dig
- Nmap
- Sipcalc

3 Bibliografia

Sumário

- 1 Fundamentos em Redes
 - Introdução
 - Protocolos em camadas
 - Camada de rede
 - Camada de transporte
 - Camada de aplicação
 - Domain Name Server (DNS)
 - Sumário

- 2 Ferramentas
 - Ping
 - Dig
 - Nmap
 - Sipcalc

- 3 Bibliografia

Ping

- Envia pacotes ICMP (*Internet Control Message Protocol*).
- **TTL** - *Time to live* ou *hop limit*
- Round-trip time

```
ping -c 5 8.8.8.8
```

```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
```

```
64 bytes from 8.8.8.8: icmp_seq=1 ttl=52 time=28.7 ms
```

```
64 bytes from 8.8.8.8: icmp_seq=2 ttl=52 time=26.5 ms
```

```
64 bytes from 8.8.8.8: icmp_seq=3 ttl=52 time=26.7 ms
```

```
64 bytes from 8.8.8.8: icmp_seq=4 ttl=52 time=26.5 ms
```

```
64 bytes from 8.8.8.8: icmp_seq=5 ttl=52 time=26.7 ms
```

```
--- 8.8.8.8 ping statistics ---
```

```
5 packets transmitted, 5 received, 0% packet loss, time 11ms
```

```
rtt min/avg/max/mdev = 26.467/27.015/28.702/0.868 ms
```


Sumário

- 1 Fundamentos em Redes
 - Introdução
 - Protocolos em camadas
 - Camada de rede
 - Camada de transporte
 - Camada de aplicação
 - Domain Name Server (DNS)
 - Sumário

- 2 Ferramentas
 - Ping
 - Dig
 - Nmap
 - Sipcalc

- 3 Bibliografia

Domain Information Groper (dig)

● Consultas ao registro DNS.

```
$ dig www.inf.ufsm.br
; <<>> DiG 9.11.4-3ubuntu5.3-Ubuntu <<>> www.inf.ufsm.br
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 38319
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;www.inf.ufsm.br. IN A

;; ANSWER SECTION:
www.inf.ufsm.br. 7193 IN A 200.18.42.2

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
```

Domain Information Groper (dig)

```
$ dig inf.ufsm.br ns
; <>> DiG 9.11.4-3ubuntu5.3-Ubuntu <>> inf.ufsm.br ns
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 8007
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;inf.ufsm.br. IN NS

;; ANSWER SECTION:
inf.ufsm.br. 6241 IN NS dns.inf.ufsm.br.

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53)
;; WHEN: seg mai 06 14:42:01 -03 2019
```

Domain Information Groper (dig)

```
dig inf.ufsm.br mx
```

```
;; ANSWER SECTION:
```

```
inf.ufsm.br. 200 IN MX 5 alt2.aspmx.l.google.com.
```

```
inf.ufsm.br. 200 IN MX 1 aspmx.l.google.com.
```

```
inf.ufsm.br. 200 IN MX 10 aspmx3.googlemail.com.
```

```
inf.ufsm.br. 200 IN MX 5 alt1.aspmx.l.google.com.
```

```
inf.ufsm.br. 200 IN MX 10 aspmx2.googlemail.com.
```

```
inf.ufsm.br. 200 IN MX 10 aspmx5.googlemail.com.
```

```
inf.ufsm.br. 200 IN MX 10 aspmx4.googlemail.com.
```

```
;; Query time: 1 msec
```

```
;; SERVER: 127.0.0.53#53(127.0.0.53)
```

```
;; WHEN: seg mai 06 15:05:12 -03 2019
```

```
;; MSG SIZE rcvd: 219
```

Sumário

- 1 Fundamentos em Redes
 - Introdução
 - Protocolos em camadas
 - Camada de rede
 - Camada de transporte
 - Camada de aplicação
 - Domain Name Server (DNS)
 - Sumário
- 2 Ferramentas
 - Ping
 - Dig
 - **Nmap**
 - Sipcalc
- 3 Bibliografia

Network Mapper

- Scanner de rede com diversas funcionalidades.

```
$ map 8.8.8.8
```

```
Starting Nmap 7.60 ( https://nmap.org ) at 2019-05-06 15:08 -03
```

```
Nmap scan report for google-public-dns-a.google.com (8.8.8.8)
```

```
Host is up (0.039s latency).
```

```
Not shown: 998 filtered ports
```

```
PORT      STATE SERVICE
```

```
53/tcp    open  domain
```

```
443/tcp   open  https
```

```
Nmap done: 1 IP address (1 host up) scanned in 4.82 seconds
```

Network Mapper

- Scanner de rede com diversas funcionalidades.

```
$ nmap -p 22 8.8.8.8
```

```
Starting Nmap 7.60 ( https://nmap.org ) at 2019-05-06 15:37 -03
Nmap scan report for google-public-dns-a.google.com (8.8.8.8)
Host is up (0.034s latency).
```

```
PORT      STATE      SERVICE
22/tcp    filtered  ssh
```

```
Nmap done: 1 IP address (1 host up) scanned in 0.42 seconds
```

Sumário

- 1 Fundamentos em Redes
 - Introdução
 - Protocolos em camadas
 - Camada de rede
 - Camada de transporte
 - Camada de aplicação
 - Domain Name Server (DNS)
 - Sumário

- 2 Ferramentas
 - Ping
 - Dig
 - Nmap
 - Sipcalc

- 3 Bibliografia

IP Subnet Calculator

```
$ sipcalc 10.1.1.20/24  
-[ipv4 : 10.1.1.20/24] - 0
```

[CIDR]

Host address - 10.1.1.20

Host address (decimal) - 167837972

Host address (hex) - A010114

Network address - 10.1.1.0

Network mask - 255.255.255.0

Network mask (bits) - 24

Network mask (hex) - FFFFFFF0

Broadcast address - 10.1.1.255

Cisco wildcard - 0.0.0.255

Addresses in network - 256

Network range - 10.1.1.0 - 10.1.1.255

Usable range - 10.1.1.1 - 10.1.1.254

IP Subnet Calculator

```
$ sipcalc 10.1.1.100/30
-[ipv4 : 10.1.1.100/30] - 0

[CIDR]
Host address - 10.1.1.100
Host address (decimal) - 167838052
Host address (hex) - A010164
Network address - 10.1.1.100
Network mask - 255.255.255.252
Network mask (bits) - 30
Network mask (hex) - FFFFFFFC
Broadcast address - 10.1.1.103
Cisco wildcard - 0.0.0.3
Addresses in network - 4
Network range - 10.1.1.100 - 10.1.1.103
Usable range - 10.1.1.101 - 10.1.1.102
```

Sumário

1 Fundamentos em Redes

2 Ferramentas

3 Bibliografia

Bibliografia

- **Sistemas Distribuídos - Princípios e Paradigmas**, A. Tanenbaum e M. Steen, Cap. 4
- **Sistemas Distribuídos - Conceitos e Projeto**, G. Coulouris, J. Dollimore, T. Kindberg e G. Blair, Cap. 3
- **Computer Networking: A Top-Down Approach**, J. F. Kurose, K. W. Ross.