

# Administração de Redes Linux



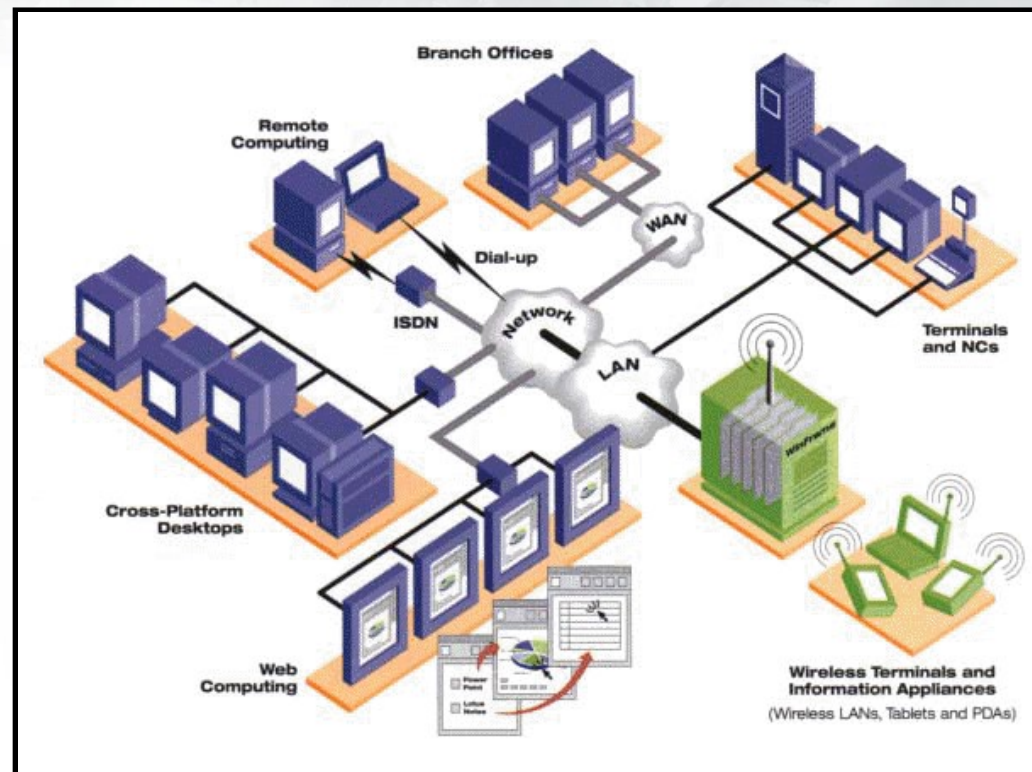


## Objetivos

- Entender os princípios de redes de computadores
- Aprender os tipos de topologias
- Conhecer dispositivos de redes
- Modelo OSI e TCP/IP
- RFC's
- Endereçamento

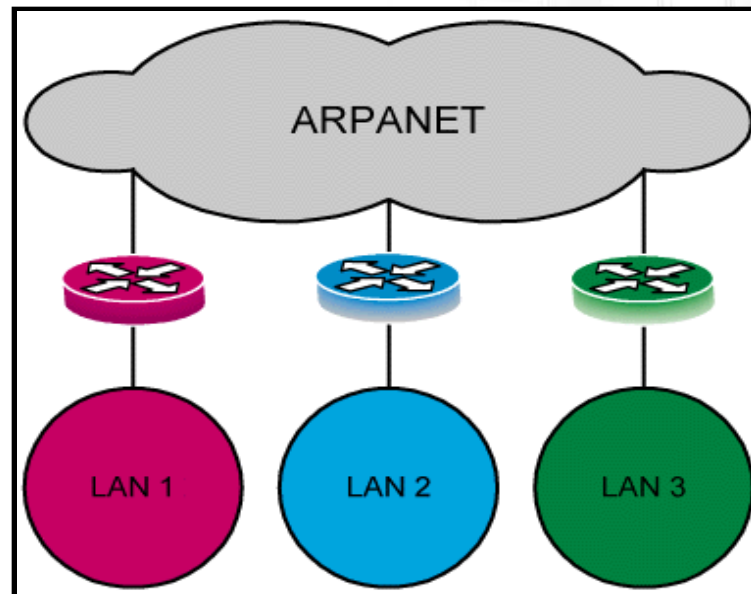
# Introdução

- Compartilhar informações
- Compartilhar serviços
- Disponibilidade
- Interatividade



## O início

- **A Arpanet** – Advanced Research Projects Agency – na década de 60.
- **Transporte de informações secretas.**







# Topologia

- O layout físico de uma rede é denominado topologia de rede.
- A topologia trata da distribuição geográfica dos nós da rede.
- A escolha da topologia depende de vários fatores.
- Estabilidade, velocidade, confiabilidade e custo são os mais importantes.



# Topologia

- Há dois tipos básicos de ligação.
  - Ponto a ponto
  - Multiponto

## Ponto a ponto

- Caracteriza-se pela presença de apenas dois pontos de comunicação um em cada extremidade do enlace ou ligação em questão.



# Multiponto

- Nas ligações multiponto observa-se presença de três ou mais dispositivos de comunicação com possibilidade de utilização do mesmo enlace.







## Utilização do meio físico

- Simplex
- Half-duplex
- Full-duplex

# Simples

- O enlace é utilizado em um dos possíveis sentidos da transmissão.

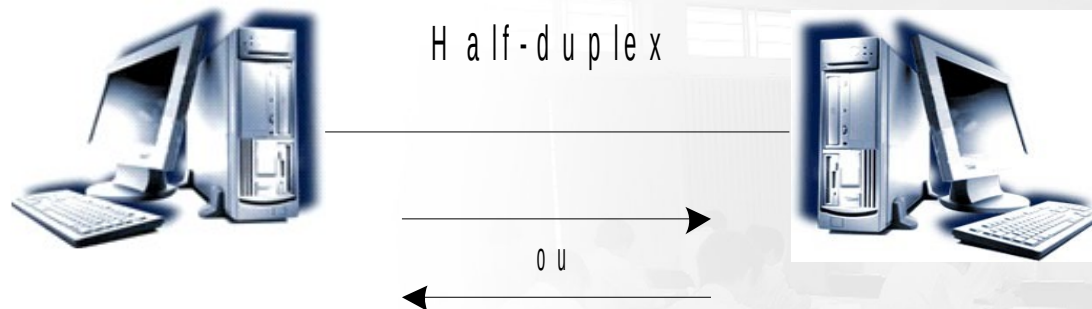


S i m p l e x



# Half-duplex

- O enlace é utilizado nos dois possíveis sentidos da transmissão, porém apenas um por vez.



# Full-duplex

- O enlace é utilizado nos dois possíveis sentidos da transmissão simultaneamente.

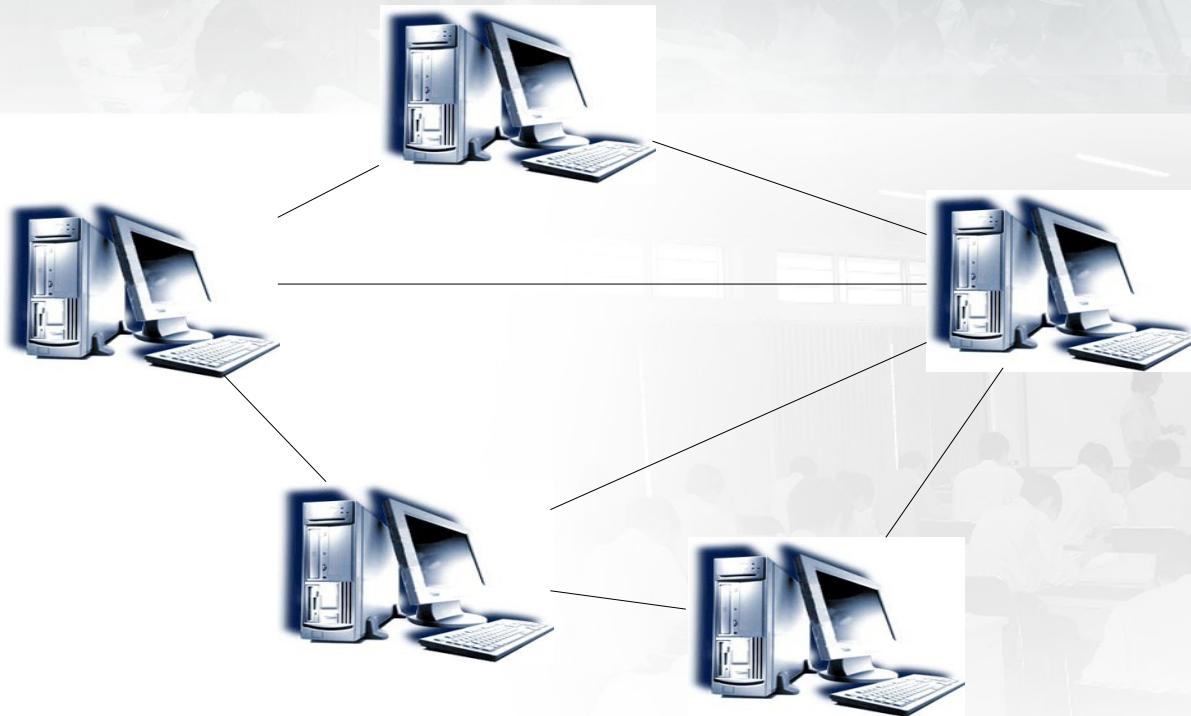




# Topologia totalmente ligada

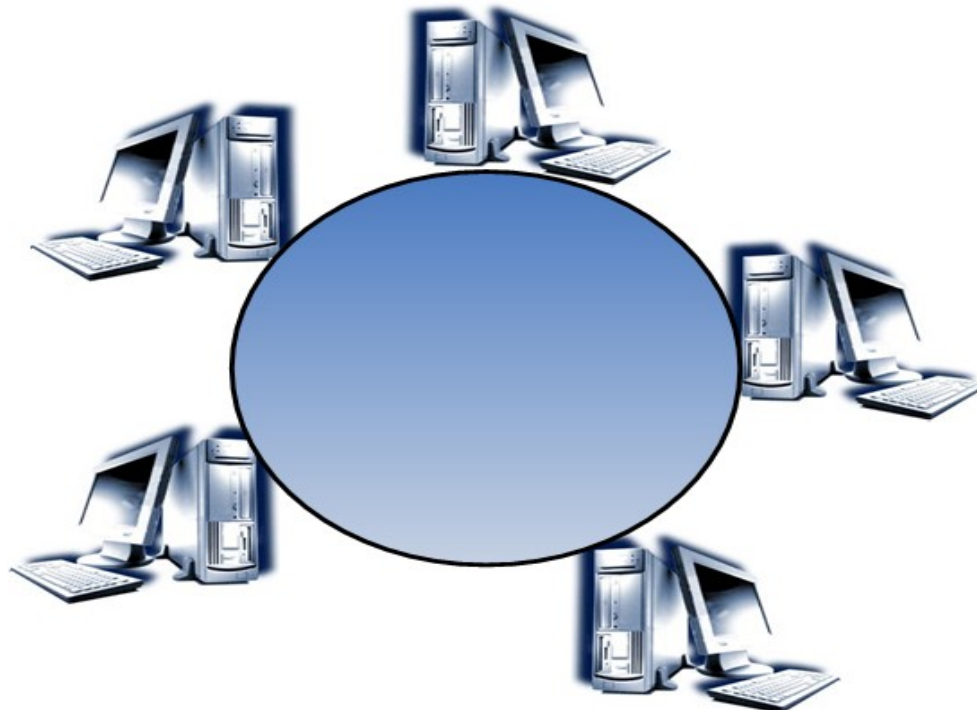


# Topologia parcialmente ligada



## Topologia em anel (ring)

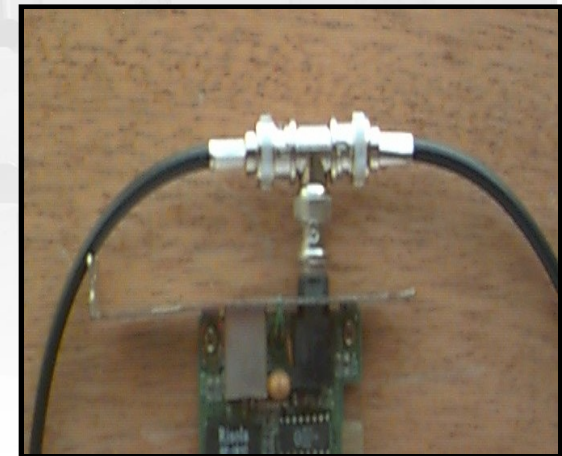
- Nessa topologia os dispositivos são conectados em série, formando um circuito fechado (anel). Os dados são transmitidos unidirecionalmente de nó em nó até atingir seu destino.





## Topologia em barramento (bus)

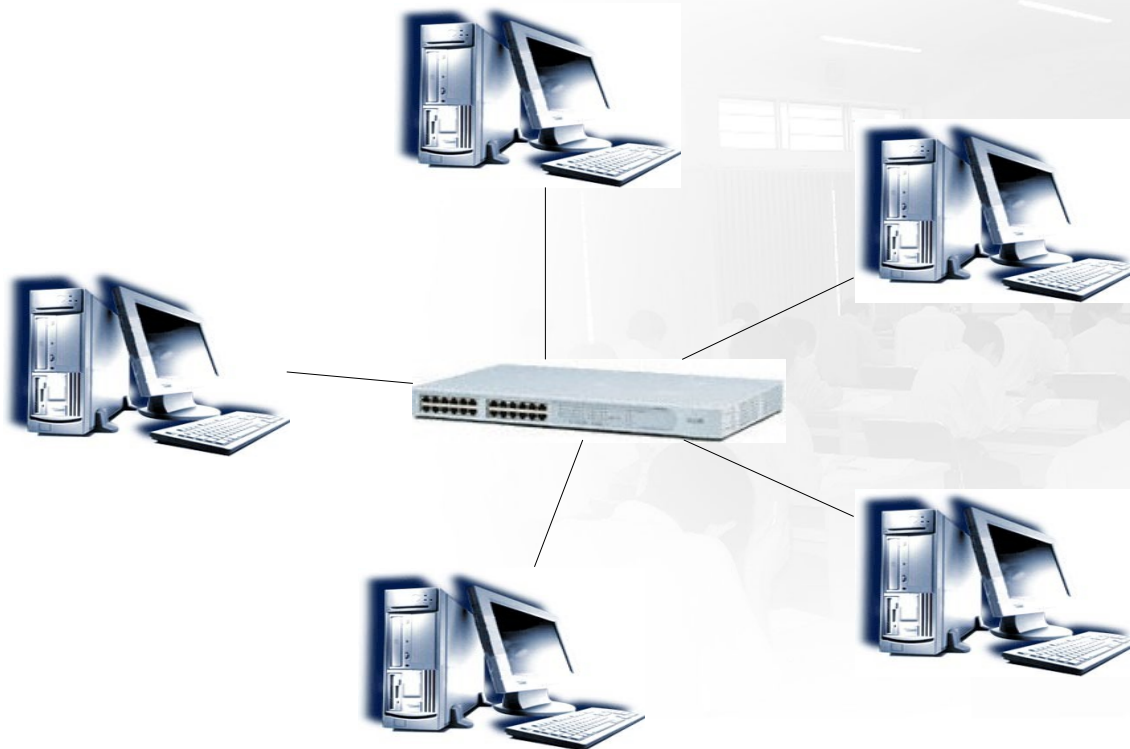
- Nessa topologia, todos os computadores, estão conectados ao mesmo meio físico. Essa estrutura possui um começo, meio e fim, limitada pelo cabo de dados.





## Topologia em estrela (stars)

- Nessa topologia, tem-se que todos os nós encontram-se conectados a um nó central. Nesse caso os periféricos se comunicam através da rede, passando o tráfego através do nó central.





## Topologia de redes quanto a distância

- Redes geograficamente distribuídas
  - **WANS** – Wide Area Networks
- Redes locais
  - **LANs** – Local Area Networks
- Redes metropolitanas
  - **MANs** – Metropolitan Area Networks
- Outras
  - **CANs** – Campus Area Networks
  - **PANs** – Personal Area Networks

## Definição quanto a distância

Distância entre processadores	Processadores localizados no(a) mesmo(a):	Exemplo:
1 m	Metro quadrado	Rede Pessoal
10 m	Sala	Rede Local  Rede Metropolitana Rede Geograficamente Distribuída
100 m	Edifício	
1 km	Campus	
10 km	Cidade	
100 km	País	
1000 km	Continente	A Internet
10000 km e acima	Planeta	



## Dispositivos de Redes

- As redes de computadores são formadas por um emaranhado de dispositivos onde cada um exerce sua função ou completa a funcionalidade de outro dispositivo de rede.
- Estudaremos aqui somente os dispositivos de redes mais comuns, que são encontrados com mais facilidade nas redes de computadores.





## NIC – Network Interface Card

- A função da interface de rede é basicamente fornecer conectividade entre o host e o meio físico da rede.
- Taxa de transmissão: 10 Mbps / 100 Mbps / 1000 Mbps
- Conectada ao computador através de um slot de expansão (ISA ou PCI) ou ainda parte da placa-mãe (On-board).
- O controle de acesso ao meio é feito através de um endereço especial de cada NIC.
- O endereço MAC (Media Access Control) é um número de 48 bits, 24 bits identificam o fabricante e o restante para identificar a placa desse fabricante.

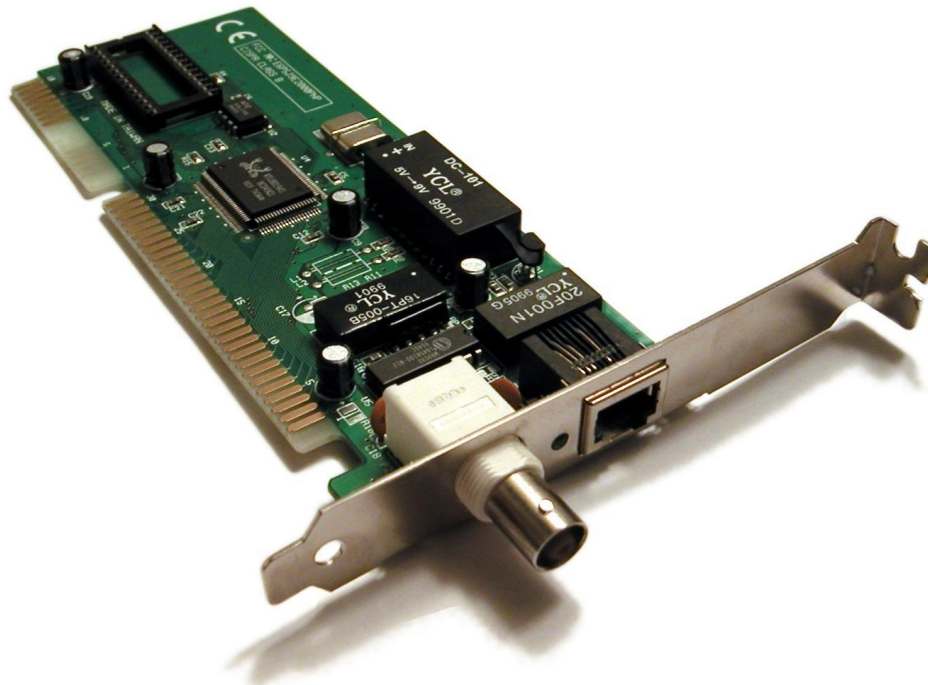


## NIC – Network Interface Card

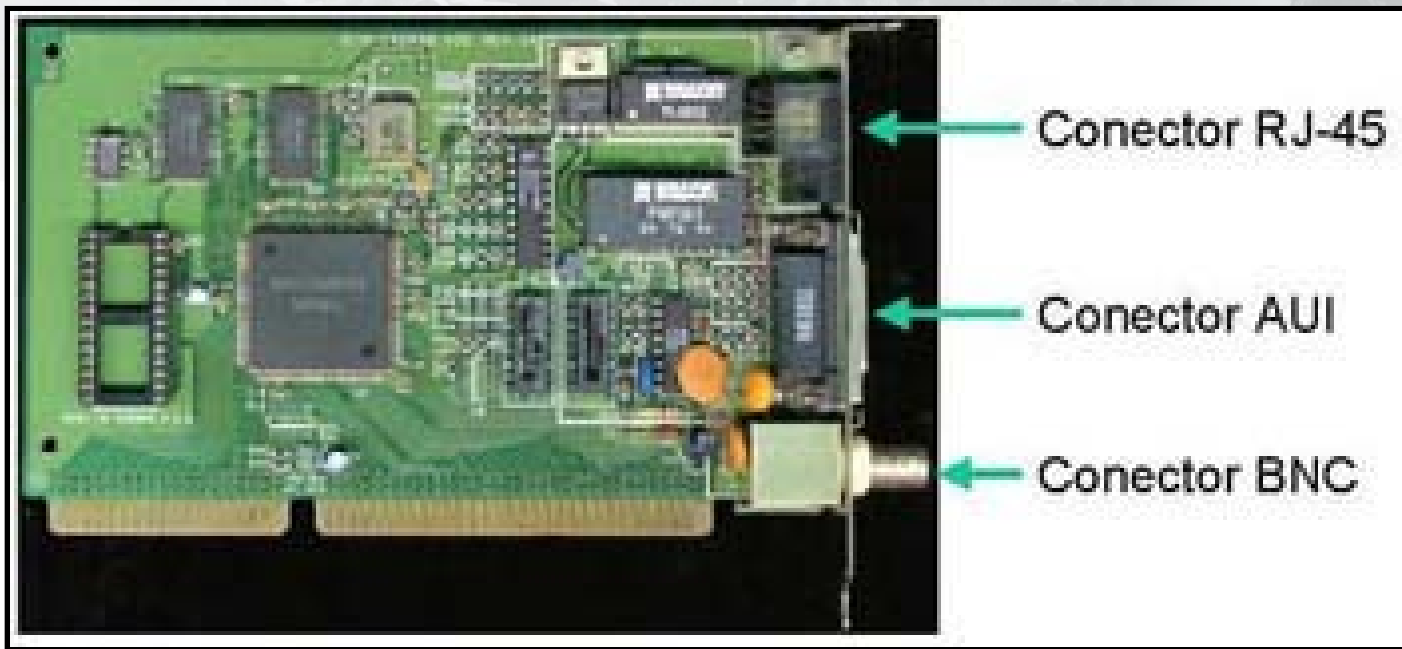
- Para visualizar o endereço MAC digite:

# ifconfig

# NIC – Network Interface Card

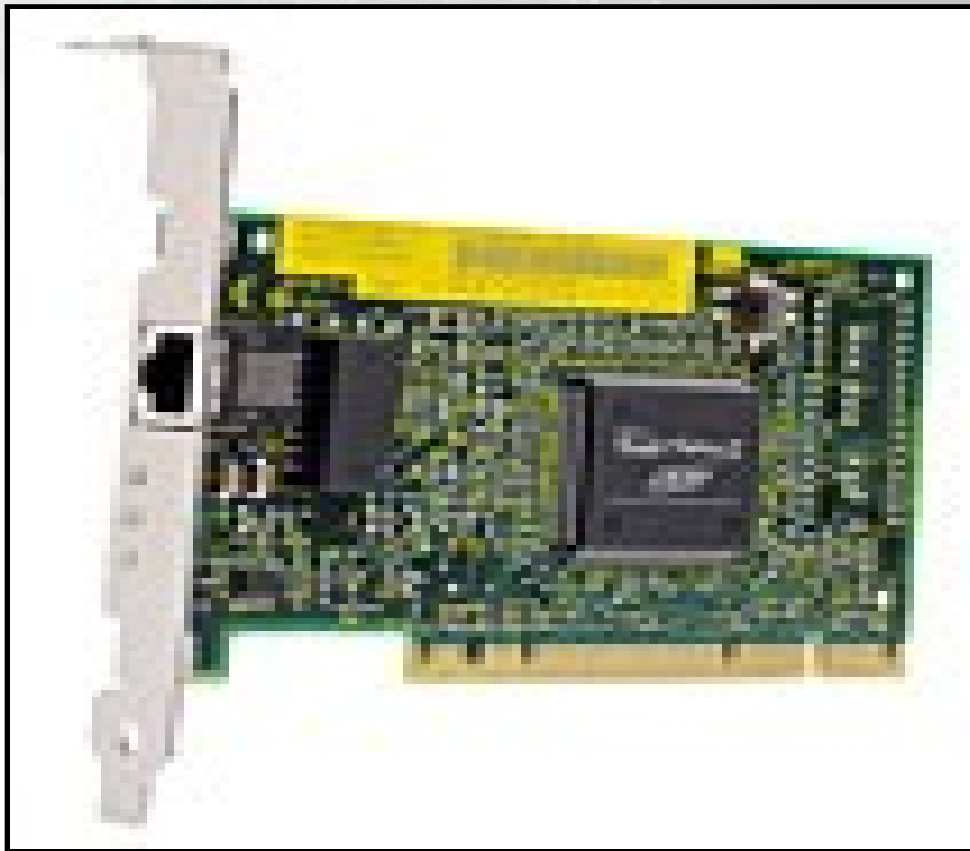


## Ethernet (10 Mbps)





## Fast Ethernet (100 Mbps)



# Gigabit Ethernet (1000 Mbps)



# Hubs

- Interligar os computadores em uma rede.
- Repetidor com várias portas.
- Dispositivo de camada 1 do modelo OSI.
- Broadcast.
- Não consegue determinar para onde enviar o pacote.



## Switch - comutador

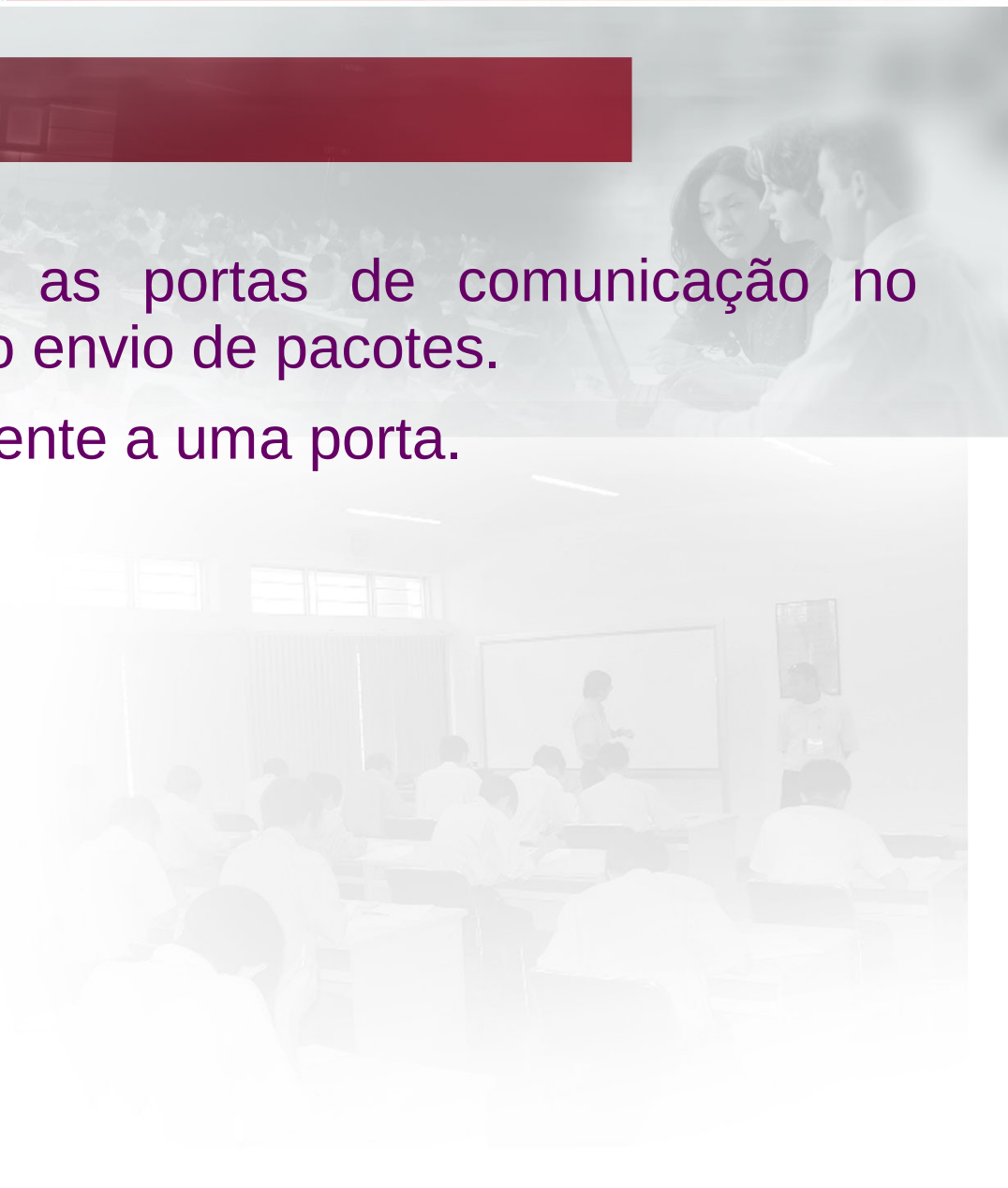
- Interligar os computadores em uma rede.
- Redução de colisões – uso de comutação.
- Dispositivo de camada 2 do modelo OSI.
- Tráfego menos intenso
- Tabelas de endereço MAC.







## Switch - comutador

- Caminhos virtuais entre as portas de comunicação no período necessário para o envio de pacotes.
  - Sinal enviado exclusivamente a uma porta.
  - Métodos
    - Cut-through
    - Store and Forward
    - Fragment Free
- 

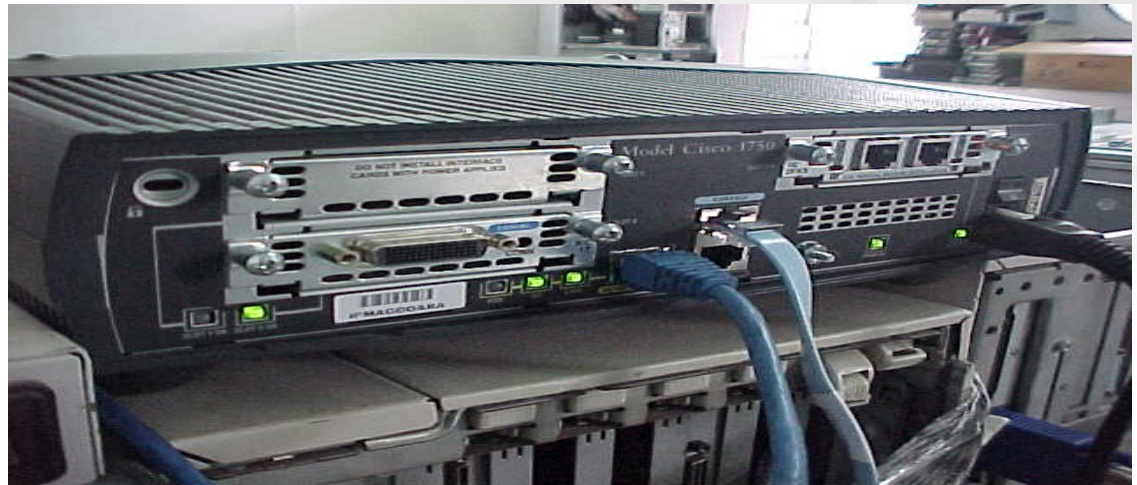


## Switch - comutador

- Cut-Through
  - Lê o MAC e manda o pacote diretamente para o destino antes mesmo de terminar de chegar;
- Store and Forward
  - Lê todo o pacote, verificando erros. Depois envia para o destino, livre de erros, descarta-o
- Fragmente-Free
  - Mistura de Cut-Through e Store and Forward, lendo apenas 64 bytes iniciais.

## Router (Roteador)

- Dispositivo que permite interconexão entre diferentes redes, mesmo geograficamente distantes e até mesmo com protocolos diferentes.
- Dispositivo que opera na camada 3 do modelo OSI.
- Equipamento mais importante para o funcionamento da internet.





## Modelo OSI e TCP/IP

- O tráfego na rede é gerado quando ocorre uma solicitação na rede.
- A solicitação tem que ser alterada daquilo que o usuário vê para um formato que possa ser utilizado na rede.
- Essa transformação é possível por meio do modelo de referência OSI e TCP/IP.
- Regula a transmissão de dados desde o meio físico até o aplicativo para o usuário.

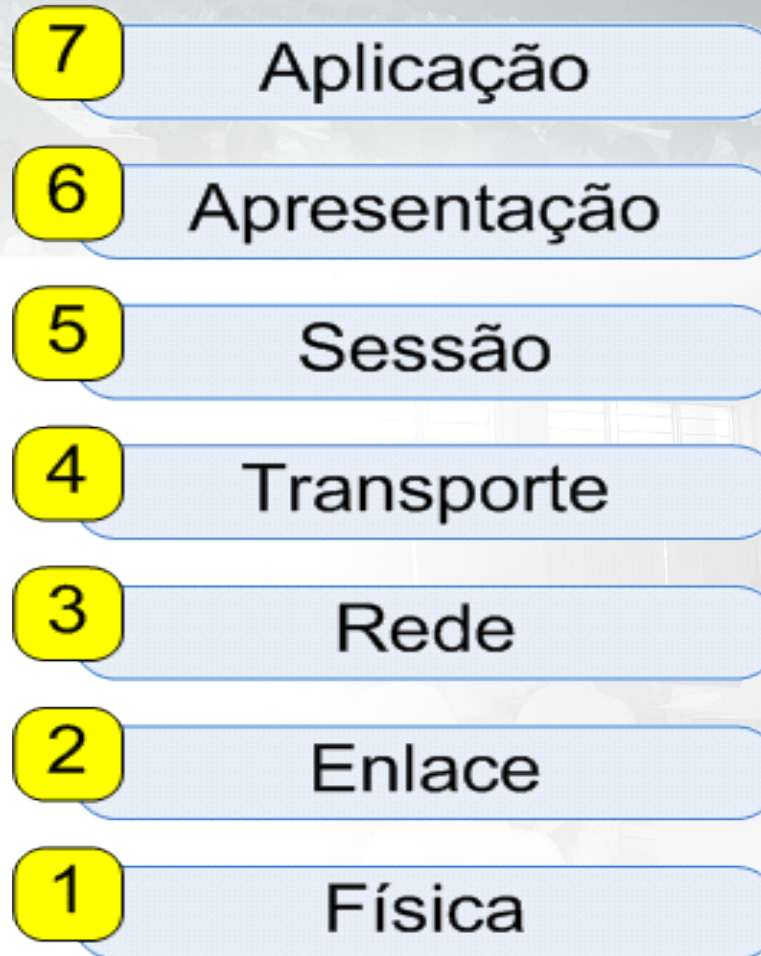




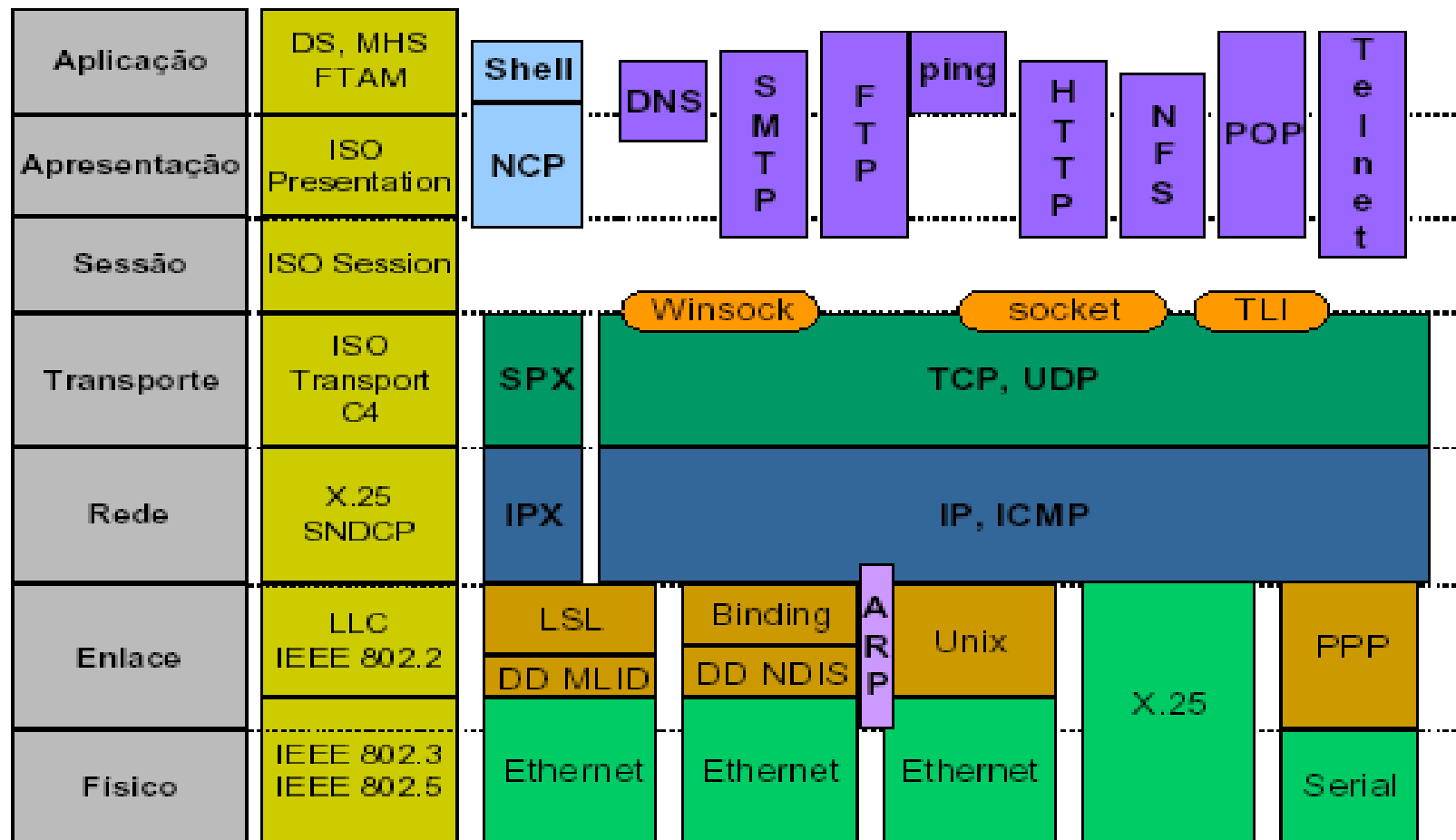
## Modelo OSI e TCP/IP

- Conceitualmente, a comunicação em redes de computadores é implementada em níveis, ou camadas. Nesse esquema, cada camada é responsável por uma parte do processo de comunicação, seja ele implementado em hardware ou software.
- O modelo de camadas mais comum e genérico é o proposto pela International Organization for Standardization (ISO), conhecido como ISO's Reference Model of Open System Interconnection, geralmente referenciado como "Modelo ISO/OSI". Este define uma arquitetura genérica com 7 camadas.

# Modelo OSI



# Modelo OSI





## Por que dividir em camadas?

- Os engenheiros que criaram a pilha de protocolos TCP/IP decidiram adotar o processo de camadas para que fosse possível padronizar e facilitar a criação e manutenção de cada processo envolvido na comunicação de redes.
- Por exemplo, a camada de acesso a rede determina como as interfaces de rede devem formatar os dados para enviar no meio físico.
- Isto permite que existam interfaces de rede do mesmo padrão (ethernet) provenientes de diferentes fabricantes.
- Dessa forma, cada componente envolvido na comunicação de redes está situado em uma ou mais camadas do modelo de referência.



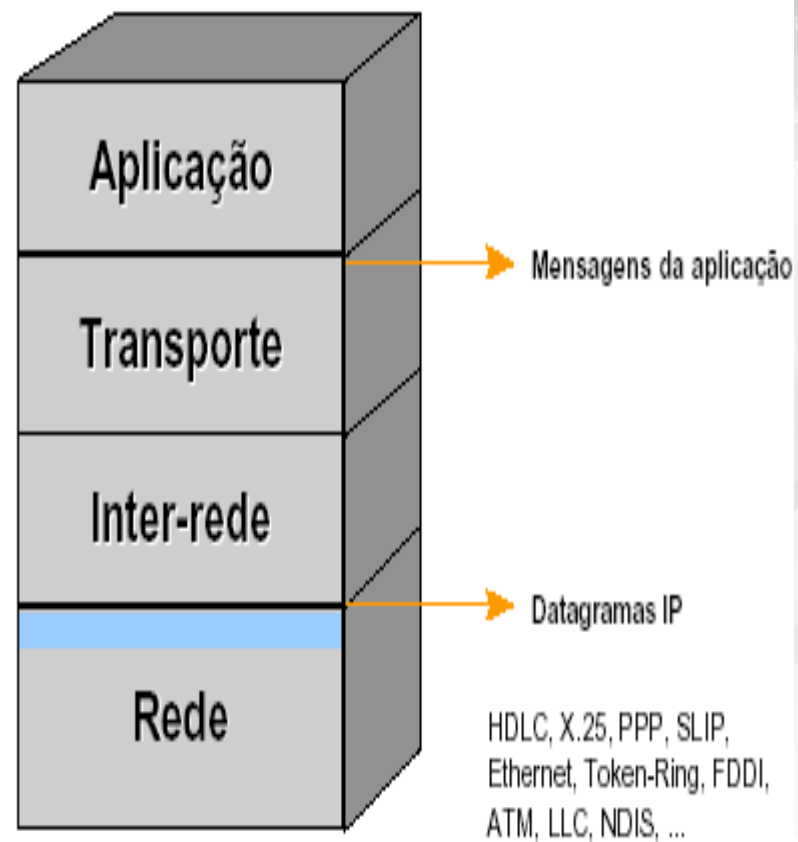


## Modelo OSI e TCP/IP

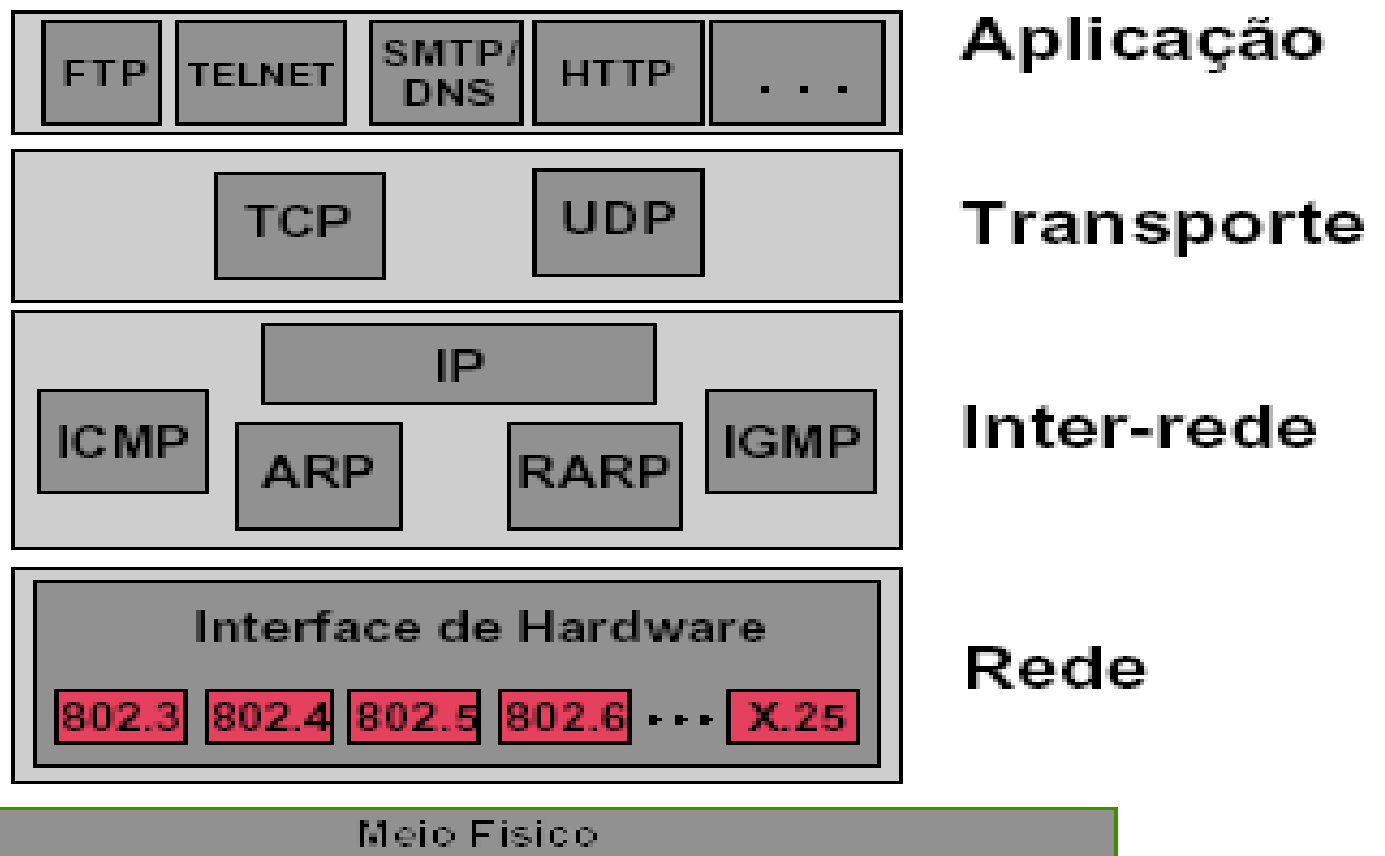
- O modelo TCP/IP foi criado no início pelo projeto DARPA.
- Foi e vem sendo utilizado como padrão para muitas implementações de redes.
- O modelo TCP/IP tem 5 camadas.
- Com o tempo, novos protocolos foram sendo criados, seguindo o processo natural de desenvolvimento da tecnologia de comunicação de dados.
- O modelo OSI expandiu o modelo TCP/IP em mais camadas.
- O sistema de relacionamento inter-camadas foi mantido, porém o modelo passou a contar com 7 camadas.

# Modelo TCP/IP

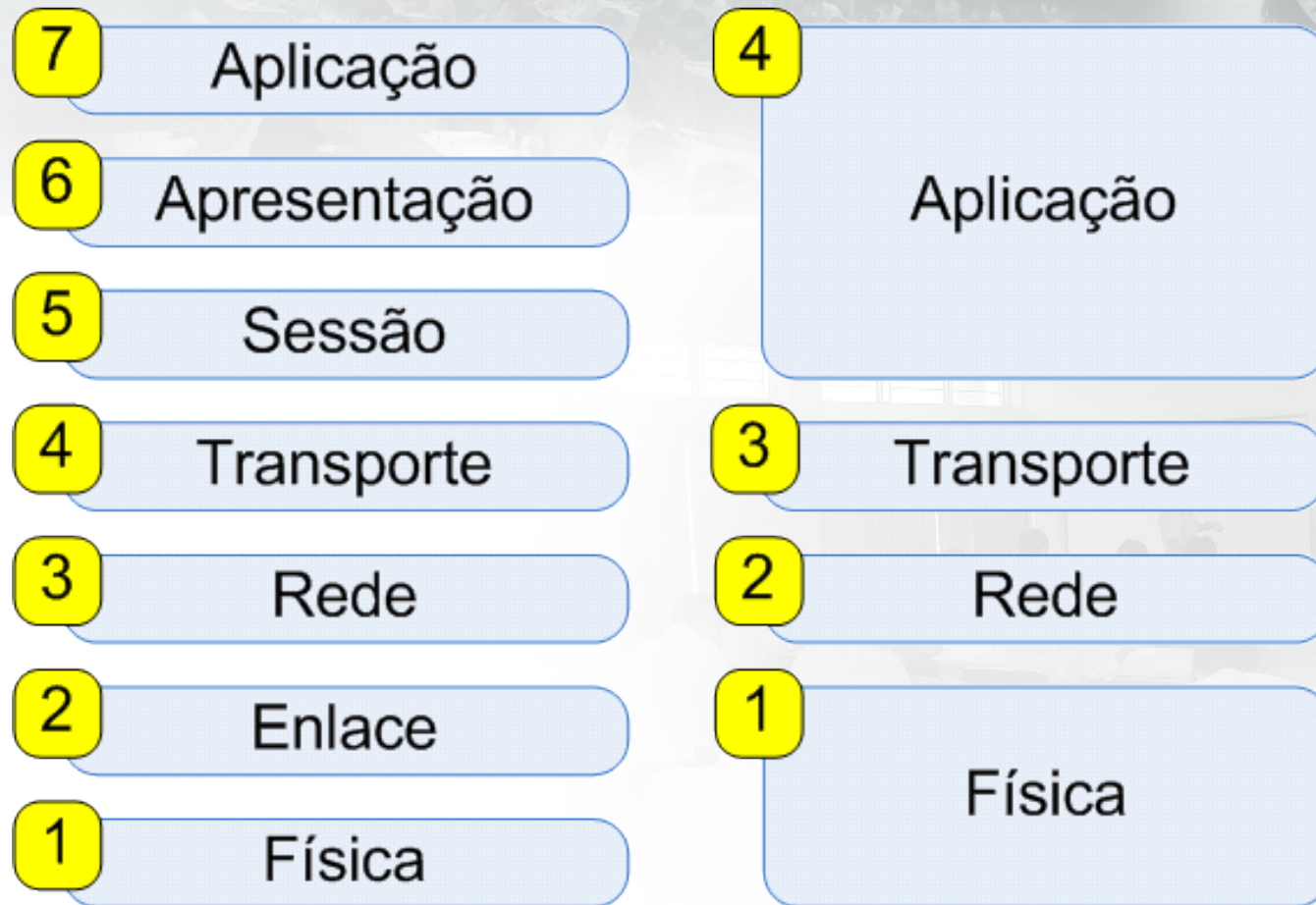
A figura 1 ilustra a divisão em camadas da arquitetura TCP/IP:



# Modelo TCP/IP



# Modelo OSI e TCP/IP



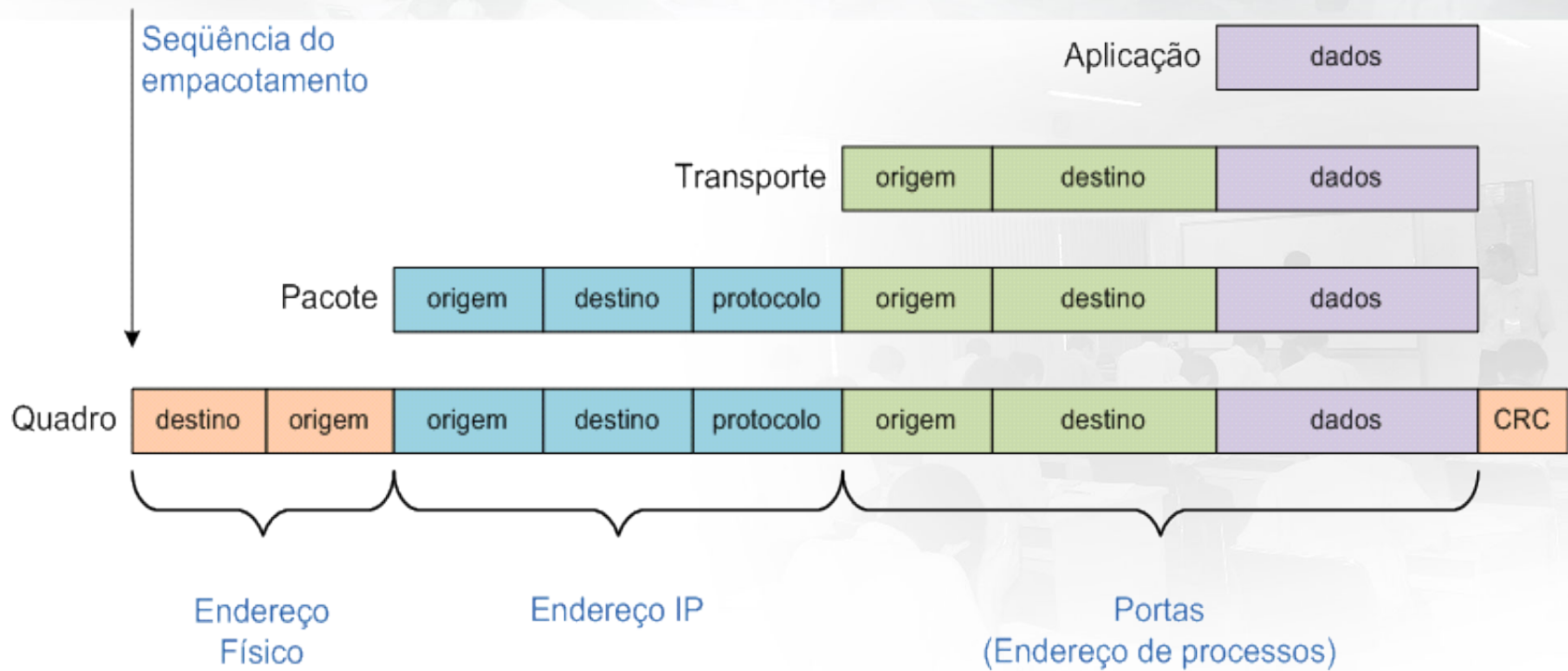
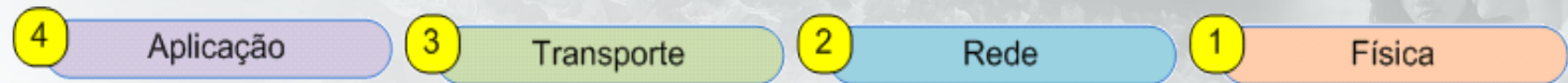




## Modelo OSI e TCP/IP

- O tráfego de rede é enviado na forma de pacotes de dados.
- Uma pacote de dados é a informação de um usuário transformado em um formato entendido pela rede.
- Cada camada adicionará informações ao pacote de dados.
- As informações adicionadas a um pacote são chamadas de cabeçalho.
- O cabeçalho de uma camada é simplesmente a informação que detalha o formato do pacote de dados

# Modelo OSI e TCP/IP





# Encapsulamento

**7. APPLICATION LAYER**

**6. PRESENTATION LAYER**

**5. SESSION LAYER**

**4. TRANSPORT LAYER**

**3. NETWORK LAYER**

**2. DATA LINK LAYER**

**1. PHYSICAL LAYER**



## Request for Comment

- Solicitação para comentários.
- São documentos distribuídos gratuitamente pela internet e continuam a evoluir conforme surgem novas tecnologias e técnicas.
- <http://www.rfc-editor.org>



# Administração de Redes Linux Parte II





# Protocolos

- Aurélio: Conjunto de regras, padrões especificações técnicas que regulam a transmissão de dados entre computadores por meio de programas específicos, permitindo a detecção e correção de erros.
- Os protocolos são como “frases” que uma interface de rede tem que dizer para poder se comunicar com as outras.
- A primeira providência que um protocolo de redes deve tomar é declarar qual protocolo estamos falando. Deve haver em algum lugar no início da mensagem um indicador de protocolo.



## Redes Internet – Protocolos TCP/IP

- A denominação TCP/IP é dada a redes internet dado o uso de dois principais protocolos: o protocolo de transporte TCP (Transport Control Protocol) e o protocolo de rede IP (Internet Protocol). Ambos, juntamente com outros protocolos serão vistos em detalhes mais adiante.
- A comunicação entre as máquinas pertencentes a rede ocorre independente da arquitetura e características de cada máquina.
- É possível a comunicação entre diferentes redes (sub-redes), de maneira transparente, independente de seus tamanhos, topologia e organização.



## Redes Internet – Protocolos TCP/IP

- É dita uma rede de pacotes, pois a informação é dividida em pacotes individuais que são reconstruídos e reordenados no destino.
- Cada pacote pode inclusive percorrer um caminho diferente.
- Falhas em uma subrede não comprometem o funcionamento da rede como um todo.





## Operação lógica AND (&)

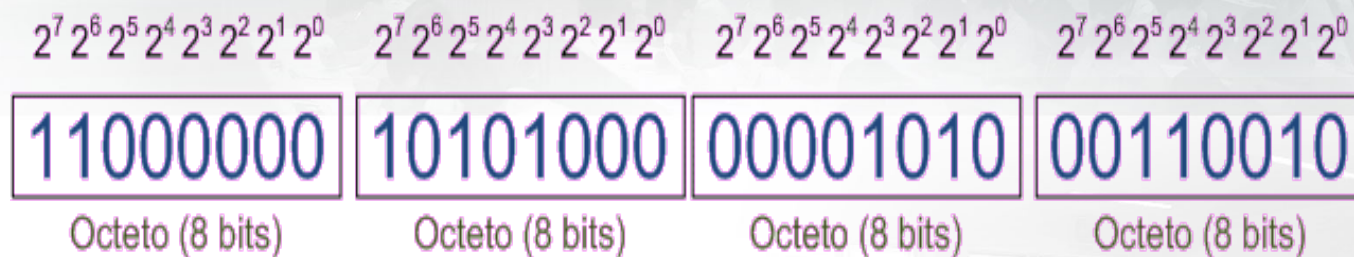
Bit	Bit	Resultado
0	1	0
1	0	0
0	1	0
1	1	1



## Endereçamento IP

- Em uma rede TCP/IP, cada máquina conectada à rede é chamada de host. Cada host, por sua vez, é identificado por um número inteiro de 32 bits, chamado endereço IP.
- É através do uso de endereços IP que toda a comunicação em uma rede internet é feita. A maneira mais comum de se representar um endereço IP é através da notação decimal pontuada.

# Endereços IPV4



$$2^7 + 2^6 = 192$$

$$2^7 + 2^5 + 2^3 = 168$$

$$2^3 + 2^1 = 10$$

$$2^5 + 2^4 + 2^1 = 50$$

192.168.10.50

notação  
decimal pontuada

notação  
binária

- 2<sup>7</sup>=128
- 2<sup>6</sup>=64
- 2<sup>5</sup>=32
- 2<sup>4</sup>=16
- 2<sup>3</sup>=8
- 2<sup>2</sup>=4
- 2<sup>1</sup>=2
- 2<sup>0</sup>=1



## Endereçamento IP

- Conceitualmente, um endereço IP é composto de duas partes: o **endereço da rede** e o **endereço do host**. Este esquema é utilizado para permitir a comunicação entre redes diferentes (através de roteamento). Esta separação é feita através da utilização de classes de endereços IP ou de máscara de rede.
- Os endereços IP foram originalmente divididos em classes conforme o número de bits utilizado para endereçamento da rede e do host, sendo que os primeiros bits do endereço identificam a classe.



## Redes Internet – Protocolos TCP/IP

<b>Classe A</b>	0	1	1	1	1	1	1	1
Multiplica por:	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
equivale a:	128	64	32	16	8	4	2	1
Multiplicação:	0x128	1x64	1x32	1x16	1x8	1x4	1x2	1x1
Resulta em:	0	64	32	16	8	4	2	1
Somando tudo:	<b>0+64+32+16+8+4+2+1</b>							
Resulta em:	<b>127</b>							

## Redes Internet – Protocolos TCP/IP

<b>Classe B</b>	1	0	1	1	1	1	1	1
Multiplica por:	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
equivale a:	128	64	32	16	8	4	2	1
Multiplicação:	1x128	0x64	1x32	1x16	1x8	1x4	1x2	1x1
Resulta em:	128	0	32	16	8	4	2	1
Somando tudo:	<b>128+0+32+16+8+4+2+1</b>							
Resulta em:	<b>191</b>							

## Redes Internet – Protocolos TCP/IP

Classe C	1	1	0	1	1	1	1	1
Multiplica por:	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
equivale a:	128	64	32	16	8	4	2	1
Multiplicação:	1x128	1x64	0x32	1x16	1x8	1x4	1x2	1x1
Resulta em:	128	64	0	16	8	4	2	1
Somando tudo:	128+64+0+16+8+4+2+1							
Resulta em:	223							

# Redes Internet – Protocolos TCP/IP

0	Identificação de Rede ( 7 bits )		Identificação do Host ( 24 bits )		Classe A		
1	0	Identificação de Rede ( 14 bits )		Identificação do Host ( 16 bits )		Classe B	
1	1	0	Identificação de Rede ( 21 bits )		Identificação do Host ( 8 bits )		Classe C
1	1	1	0	Identificação de Grupo para MultiCast ( 28 bits )			Classe D
1	1	1	0	Reservado para futuro uso ( 27 bits )			Classe E



## Redes Internet – Protocolos TCP/IP

Classe	Gama de Endereços	N.º Endereços por Rede
<b>A</b>	1.0.0.0 até 126.0.0.0	16 777 216
<b>B</b>	128.0.0.0 até 191.255.0.0	65 536
<b>C</b>	192.0.0.0 até 223.255.255.0	256
<b>D</b>	224.0.0.0 até 239.255.255.255	<i>multicast</i>
<b>E</b>	240.0.0.0 até 255.255.255.255	<i>multicast reservado</i>

$$2^n - 2 = N_n/N_h$$



## Redes Internet – Protocolos TCP/IP

- Endereços privados (RFC 1918)
- Dentro das classes A,B e C foram reservadas 3 redes, normatizados pela RFC 1918, que são conhecidas como endereços de rede privados, usados em intranets:

Classe A: 10.0.0.0 - 10.255.255.255

Classe B: 172.16.0.0 - 172.31.255.255

Classe C: 192.168.0.0 - 192.168.255.255

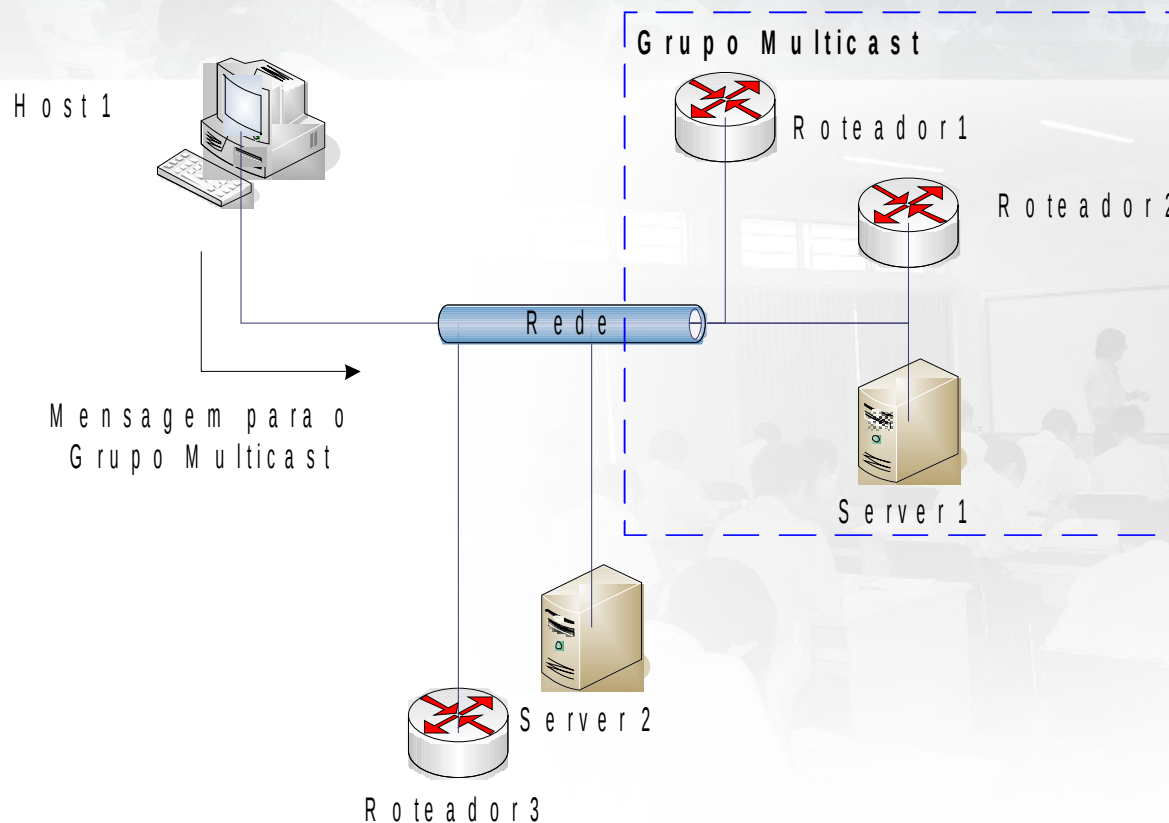


## Máscara de Sub-Rede

- 32 bits em notação decimal pontuada.
- bits 1 indicam o endereço da sub-rede.
- bits 0 o endereço do host.
- Máscara default
- **Classe A: 255.0.0.0 ou /8 ou**
  - » 11111111.00000000. 00000000.  
00000000
- **Classe B: 255.255.0.0 ou /16 ou**
  - » 11111111.11111111. 00000000.  
00000000

# Endereçamento de Multicast

- É a entrega de informação para múltiplos destinatários simultaneamente.







## Endereçamento CIDR – Máscara variável

- Com o crescimento da internet, veio a necessidade de ampliação e flexibilização das classes e da distribuição de endereços IP, o que motivou a criação de um novo esquema, chamado CIDR (Classless Inter-Domain Routing).
- Com isso ganhamos mais liberdade na determinação de endereços de redes/hosts e um melhor aproveitamento do espaço de endereços IP.

## Endereçamento CIDR – Máscara variável



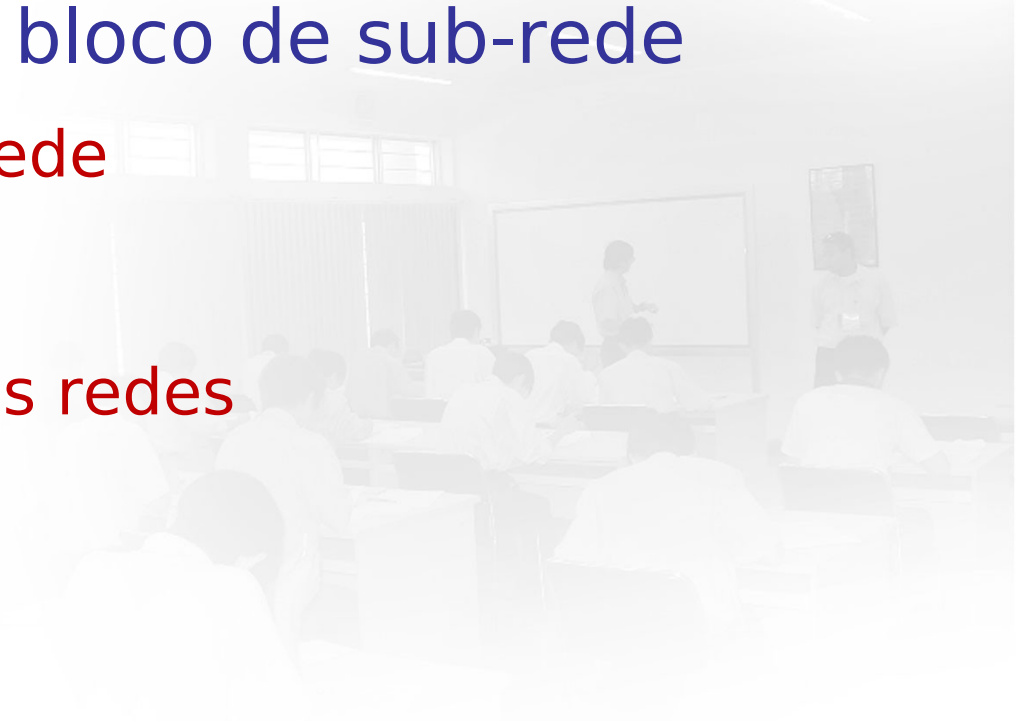


## Endereços especiais

- 0.0.0.0 RFC 1700 – Indica a própria rede
- 10.0.0.0/8 RFC 1928 – Endereçamento reservado para uso em redes internas e privadas
- 127.0.0.0/8 RFC 1700 – Rede dita loopback que aponta sempre para a própria máquina
- 172.16.0.0/12 RFC 1918 – Idem rede 10.0.0.0/8
- 192.168.0.0/16 RFC 1918 – Idem acima

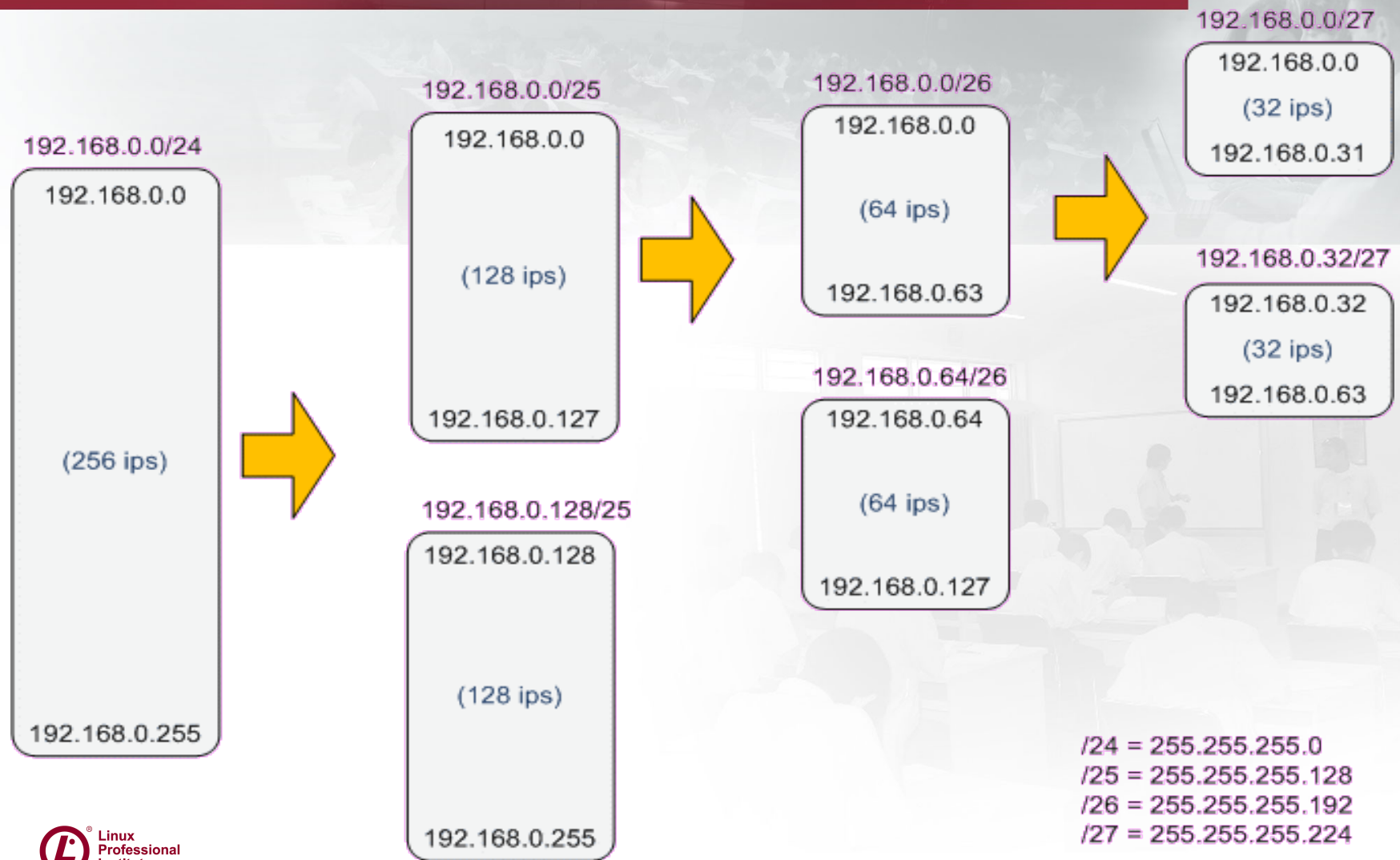


## Endereços especiais

- Primeiro endereço do bloco de sub-rede
    - Identificador da sub-rede
  - Último endereço do bloco de sub-rede
    - Broadcast para a sub-rede
  - 255.255.255.255
    - Broadcast para todas as redes
- 



# Sub-Redes



# Cálculo de Sub-Redes

	$2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$	$2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$	$2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$	$2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$
IP	<b>11000000</b>	<b>10101000</b>	<b>00001010</b>	<b>01100110</b>
	$2^7+2^6=192$	$2^7+2^5+2^3=168$	$2^3+2^1=10$	$2^6+2^5+2^2+2^1=102$

AND (&)

Máscara de sub-rede	<b>11111111</b>	<b>11111111</b>	<b>11111111</b>	<b>00000000</b> /24
	$2^7+2^6+2^5+2^4+2^3+2^2+2^1+2^0=255$	$2^7+2^6+2^5+2^4+2^3+2^2+2^1+2^0=255$	$2^7+2^6+2^5+2^4+2^3+2^2+2^1+2^0=255$	0

IP da rede	<b>11000000</b>	<b>10101000</b>	<b>00001010</b>	<b>00000000</b>
	$2^7+2^6=192$	$2^7+2^5+2^3=168$	$2^3+2^1=10$	0

192.168.10.0

$2^7=128$   
 $2^6=64$   
 $2^5=32$   
 $2^4=16$   
 $2^3=8$   
 $2^2=4$   
 $2^1=2$   
 $2^0=1$

$1 \& 0 = 0$   
 $0 \& 1 = 0$   
 $0 \& 0 = 0$   
 $1 \& 1 = 1$

Quantidade de redes =  $2^{\text{numero de bits de rede}}$   $\longrightarrow 2^{24}=16.777.216$

Quantidade de ips =  $2^{\text{numero de bits de ips}}$   $\longrightarrow 2^8=256$

Quantidade de hosts =  $(2^{\text{numero de bits de ips}}) - 2 \longrightarrow (2^8) - 2=254$

# Cálculo de Sub-Redes

	$2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$	$2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$	$2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$	$2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$	
IP	11000000	10101000	00001010	01100110	
	$2^7+2^6=192$	$2^7+2^5+2^3=168$	$2^3+2^1=10$	$2^6+2^5+2^2+2^1=102$	
	AND (&)				
Máscara de sub-rede	11111111	11111111	11111111	11000000	/26
	$2^7+2^6+2^5+2^4+2^3+2^2+2^1+2^0=255$	$2^7+2^6+2^5+2^4+2^3+2^2+2^1+2^0=255$	$2^7+2^6+2^5+2^4+2^3+2^2+2^1+2^0=255$		
IP da rede	11000000	10101000	00001010	01000000	
	$2^7+2^6=192$	$2^7+2^5+2^3=168$	$2^3+2^1=10$	$2^6=64$	
	192.168.10.64				

$2^7=128$   
 $2^6=64$   
 $2^5=32$   
 $2^4=16$   
 $2^3=8$   
 $2^2=4$   
 $2^1=2$   
 $2^0=1$

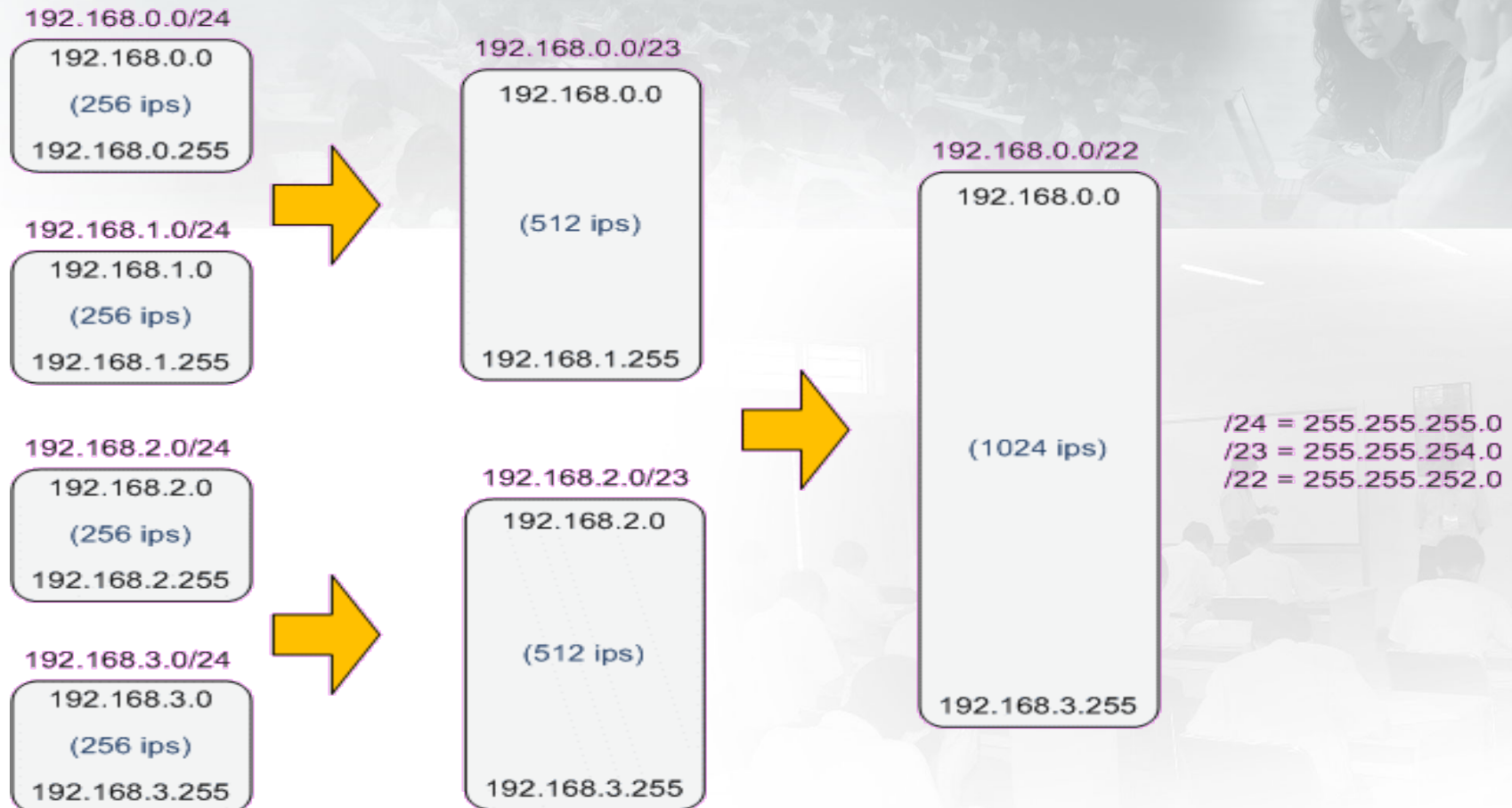
$1 \& 0 = 0$   
 $0 \& 1 = 0$   
 $0 \& 0 = 0$   
 $1 \& 1 = 1$

Quantidade de redes =  $2^{\text{numero de bits de rede}}$   $\longrightarrow 2^{26}=67.108.864$

Quantidade de ips =  $2^{\text{numero de bits de ips}}$   $\longrightarrow 2^6=64$

Quantidade de hosts =  $(2^{\text{numero de bits de ips}}) - 2 \longrightarrow (2^6) - 2=62$

# Super-Redes



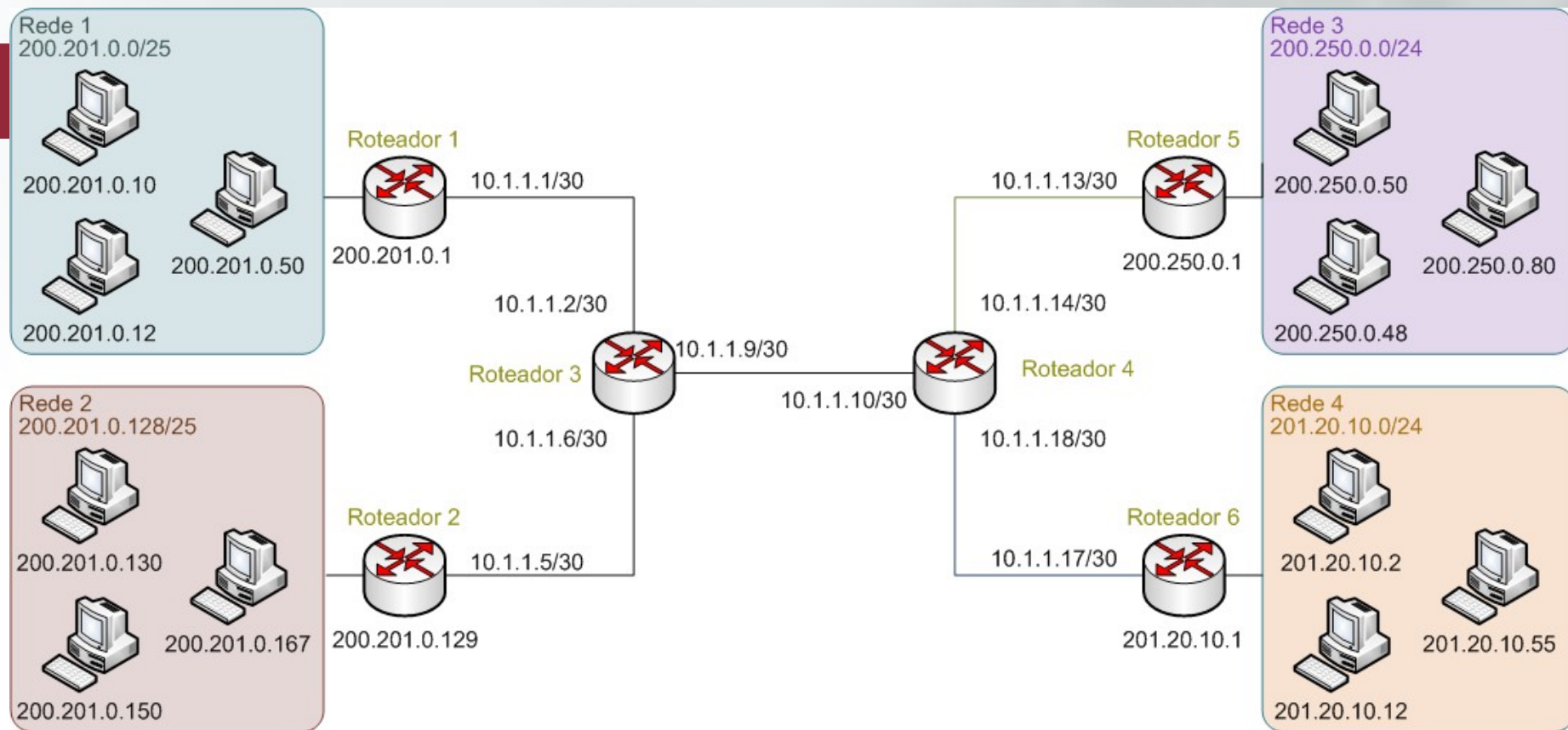




## Roteamento

- É na camada 3 (Rede) que são tomadas as decisões de roteamento com base em endereços lógicos como o IP.
- Protocolos roteáveis
- A função do roteamento é interligar redes ou sub-redes diferentes.
- Compara o ID da rede origem com o ID da rede do destino.
- Se o resultado for o mesmo, significa que os hosts estão na mesma rede, assim o host de origem pode entregar o pacote diretamente.





## Tabela de Roteamento (forma genérica)

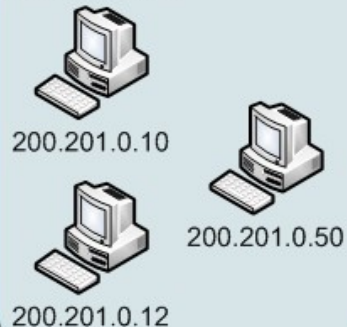
Rede de destino: 200.250.0.0/24

Gateway ou Next-Hop: 200.201.0.1

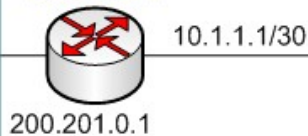
Interface: eth0 ou 200.201.0.10

Custo: 1

Rede 1  
200.201.0.0/25



Roteador 1



10.1.1.2/30

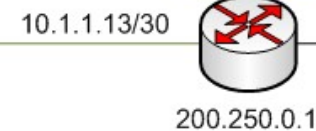
Roteador 3

10.1.1.9/30

10.1.1.10/30

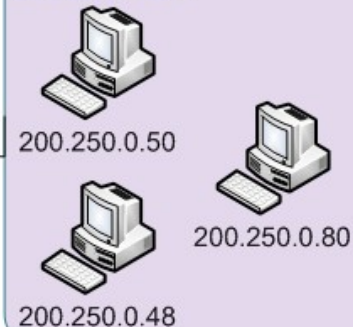
Roteador 4

Roteador 5



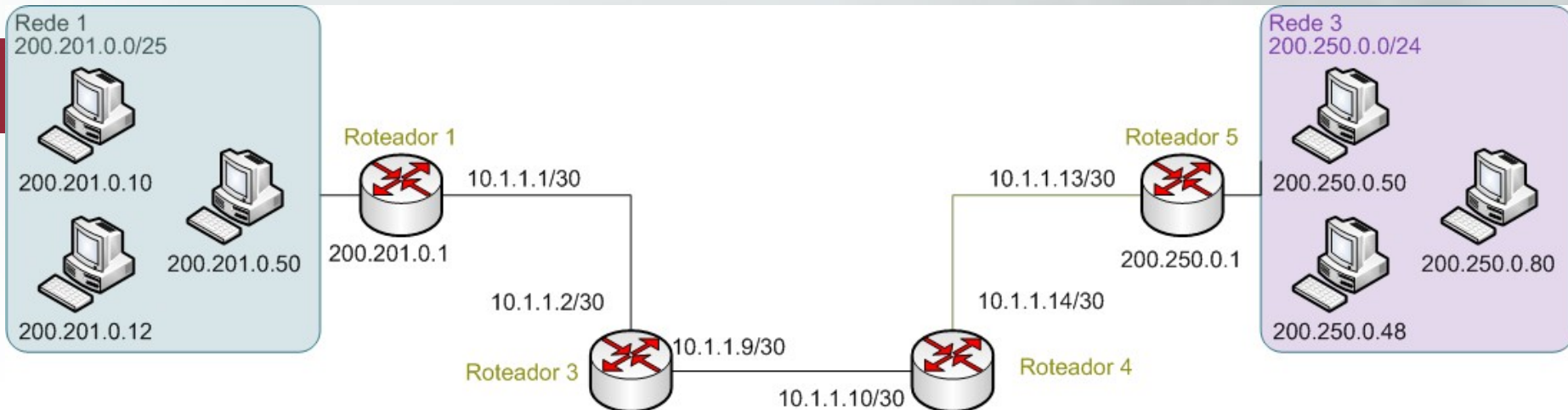
10.1.1.14/30

Rede 3  
200.250.0.0/24



## Tabela de roteamento do Roteador 1

Rede	Gateway	Interface	Custo
200.201.0.0/25	---	eth0	0
10.1.1.0/30	---	Serial 0	0
200.250.0.0/24	10.1.1.2	Serial 0	1



### Tabela de roteamento do Roteador 3

Rede	Gateway	Interface	Custo
10.1.1.0/30	---	Serial 0	0
10.1.1.8/30	---	Serial 1	0
200.201.0.0/25	10.1.1.1	Serial 0	1
200.250.0.0/24	10.1.1.10	Serial 1	1

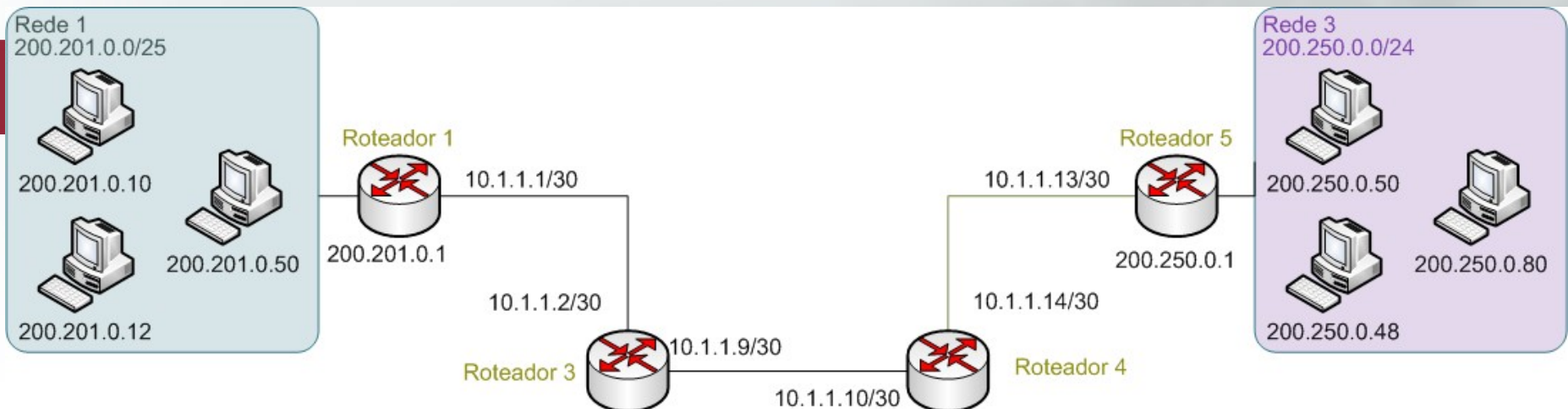
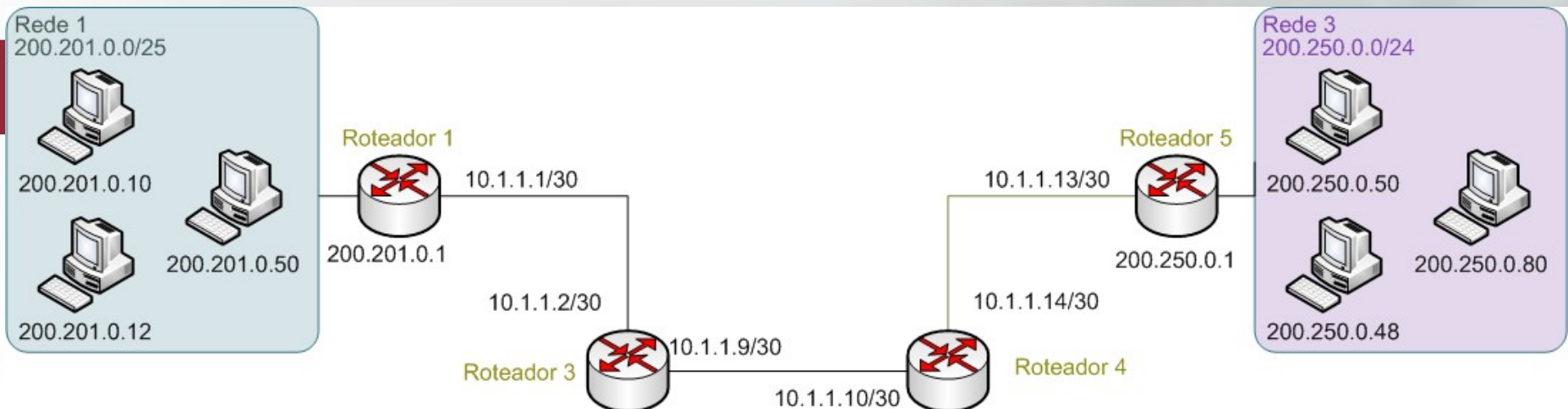


Tabela de roteamento do Roteador 4			
Rede	Gateway	Interface	Custo
10.1.1.8/30	---	Serial 0	0
10.1.1.12/30	---	Serial 1	0
200.201.0.0/25	10.1.1.9	Serial 0	1
200.250.0.0/24	10.1.1.13	Serial 1	1





**Tabela de roteamento do Roteador 5**

Rede	Gateway	Interface	Custo
10.1.1.12/30	---	Serial 0	0
200.250.0.0/24	---	eth0 ou 200.250.0.1	0
200.201.0.0/25	10.1.1.14	Serial 0	1





## Roteamento

- Se o resultado for diferente, o host de origem irá analisar sua tabela de roteamento em busca de uma entrada que especifique para qual host (geralmente um roteador) os dados devem ser entregues para alcançar determinada sub-rede.
- Caso a busca na tabela de roteamento não encontre a entrada adequada, o host de origem enviará o pacote de dados para o **gateway padrão**.



## Endereçamento Ethernet

- é o mais utilizado na conexão de redes internas
- No padrão Ethernet, cada dispositivo de rede fabricado tem uma identificação única através de um número de 48 bits chamado endereço MAC.

□ **00:50:DA:C3:F1:9C**

- É composto de 6 bytes, onde os 3 primeiros identificam o fabricante da placa.
- Os bytes normalmente são exibidos em notação hexadecimal.



## Protocolo ARP

- É responsável pela resolução (tradução) de endereços IP em endereços MAC.
- Essa tradução é necessária pois os dispositivos de redes se identificam somente através de endereços MAC.
- O protocolo IP antes de enviar um pacote, determina através da máscara se o endereço de destino pertence a mesma rede.
- Caso o endereço não pertença a mesma rede ele envia o pacote para a porta do roteador, ambos usam ARP

# Protocolo ARP

Seqüência do  
empacotamento

7 Aplicação

6 Apresentação

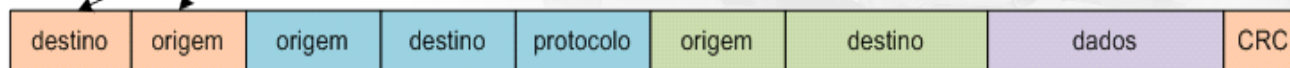
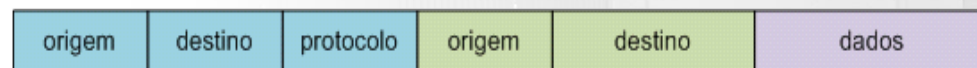
5 Sessão

4 Transporte

3 Rede

2 Enlace

1 Física





# Protocolo ARP

- A máquina remetente envia uma requisição ARP que basicamente pergunta para toda rede quem é que tem o endereço IP desejado.

00:04:76:eb:b5:21 → f f:f f:f f:f f:f f:f f      ARP Who has 10.0.5.1? Tell 10.0.5.17

00:01:02:66:e7:90 → 00:04:76:eb:b5:21 ARP 10.0.5.1 at 00:01:02:66:e7:90

- A máquina 10.0.5.17 construiria agora a sua tabela arp

IP de origem	10.0.5.17	IP de destino	10.0.5.1
MAC de origem	00:04:76:eb:b5:21	MAC de destino	00:01:02:66:e7:90





## Protocolo ARP

- Para evitar ter que enviar as requisições ARP repetidas vezes, o sistema operacional faz um cache temporário das respostas (geralmente 30 segundos).
- `# arp -n`





## Resolução de Nomes

- A idéia de resolução de nomes é permitir que, a partir de uma nome de host como [www.terra.com.br](http://www.terra.com.br) , chegue-se ao endereço IP 200.154.56.80.
- Nos primórdios na internet, todos os nomes de hosts existentes eram armazenados em um arquivo texto que era distribuído entre as máquinas inter-conectadas.
- Surgimento do DNS.



## Protocolo IP

- IP (*Internet Protocol*) é o protocolo fundamental no qual se baseia uma Rede Internet
- É oferecer um serviço de entrega de pacotes simples e eficiente que possa ser utilizado como base para os outros protocolos que compõem uma rede internet.
- É um protocolo “não confiável”, no sentido de que não há garantias de que os dados transmitidos realmente chegaram ao seu destino.
- As mensagens não seguem um caminho pré-determinado entre origem e destino.
- Assim, não há garantias que a mensagem chegue ao destino nem na mesma ordem em que foram enviadas.



## Protocolo IP

- Sempre que o conjunto de dados a ser transmitido for muito grande para o protocolo inferior (ethernet, por exemplo), ele é fragmentado em tamanhos menores
- **MTU** (Maximal Transmission Unit).
  - Cada fragmento desses recebe uma identificação que permite ao destino remontar os fragmentos no datagrama IP original.
- **Roteamento IP**
- **TTL** (Time To Live)
  - Cada vez que um datagrama IP passa por um roteador, um campo especial do seu cabeçalho é decrementado.





## Protocolo IP

- **TTL (Time To Live)**
  - A função desse campo é limitar o tempo de vida do pacote, caso contrário ele poderia ficar sendo roteado para sempre entre as diversas redes existentes nunca desaparecendo.
  - Quando esse valor chega a zero, o pacote é descartado e uma mensagem ICMP é enviada para o remetente indicando que o TTL expirou.
  - Este recurso é utilizado por programas como traceroute para mapear o caminho percorrido por um pacote da origem ao destino.
  - *# cat /proc/sys/net/ipv4/ip\_default\_ttl*





# Protocolo ICMP

- ICMP (Internet Control Message Protocol)
  - É um protocolo simples, encapsulado no protocolo IP que foi criado para o envio de mensagens de erro ou prover informações a respeito da disponibilidade de serviços.
  - Seu objetivo é apontar erros com a rede IP, em especial problemas de roteamento e disponibilidade de serviços.



# Protocolo ICMP

- Mensagens mais importantes
  - Destination unreachable: o pacote não pode ser entregue.
  - Redirect: um roteador avisando uma estação que existe outra rota para o destino desejado.
  - TTL exceeded: campo TTL chegou a zero
  - echo request e echo reply: utilizado principalmente para saber se uma máquina ou serviço esta ativo.



## Protocolo UDP – RFC 768

- UDP (User Datagram Protocol)
  - Protocolo de nível de transporte orientado à transmissão de mensagens sem a necessidade do estabelecimento de uma conexão
  - Não há garantia da entrega da mensagem, muito menos sua ordenação.
  - Usado para mensagens curtas ou onde não vale o estabelecimento de uma conexão
  - Consultas DNS
  - TFTP (Trivial FTP)
  - Streaming de áudio/vídeo

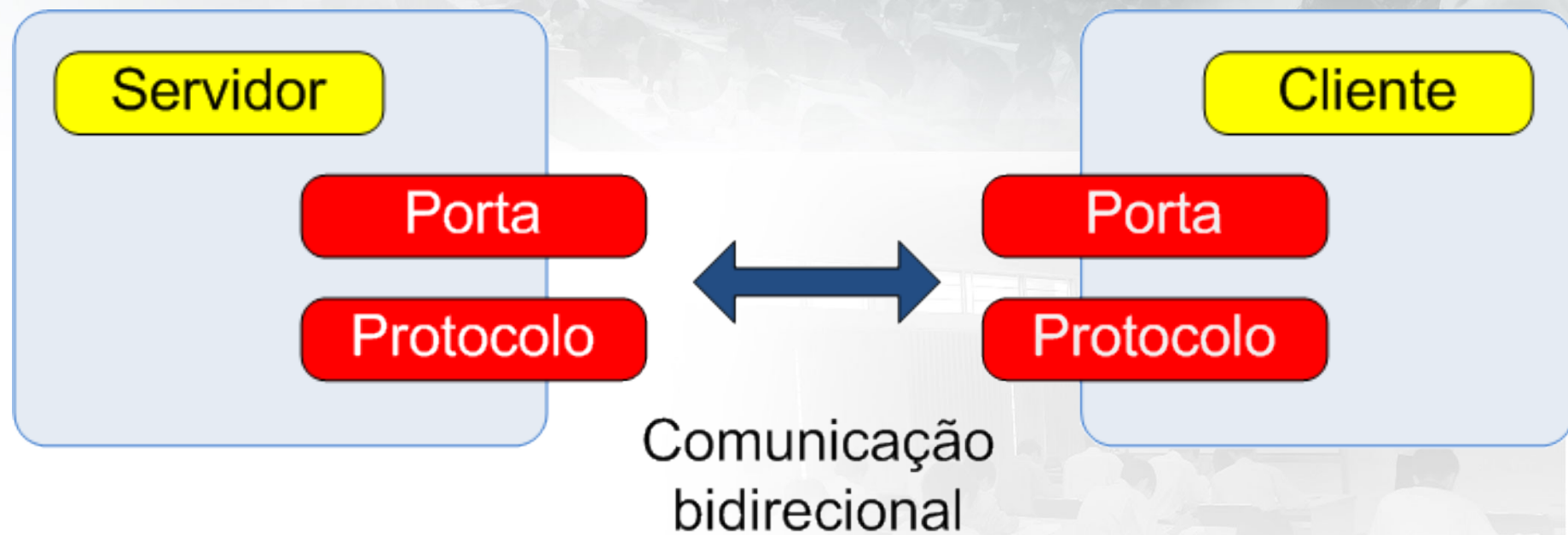


## Protocolo UDP

- O UDP utiliza o conceito de portas para identificar origem e destino.
- Uma empresa possui um número telefônico geral (o endereço IP), e funcionários e setores possuem ramais (as portas).
- Quando alguém liga para o número geral (IP), pede ou disca um ramal (porta) e consegue falar com a pessoa (aplicação) desejada.



# Protocolo UDP





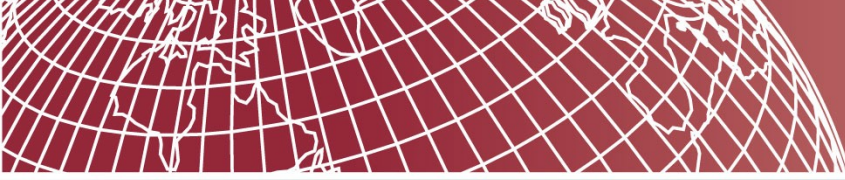
## Protocolo TCP – RFC 793

- TCP (Transmission Control Protocol)
  - É um protocolo de nível de transporte orientado a conexão.
  - Como tal oferece uma comunicação confiável entre a origem e o destino.



## Principais características do TCP

- Controle de fluxo
- Recuperação de erros
- Transmissão full duplex (transmite e recebe ao mesmo tempo).
- Reordenação de segmentos.
- Assim como o UDP, utiliza o conceito de portas.
- Opera com o conceito de fluxo de dados, a aplicação não precisa se preocupar com detalhes como tamanho de pacotes. O protocolo se encarrega disso automaticamente.



# Campos do pacote TCP







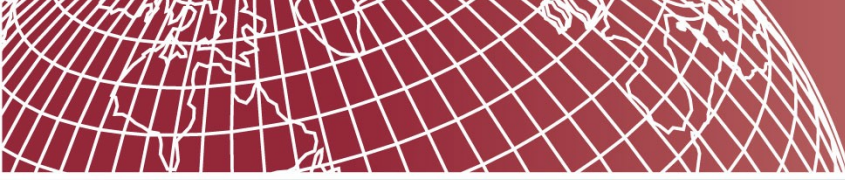
## Principais características do TCP

- Porta de Origem/Destino
- Sequence Number: O emissor determina seu próprio número de sequência
- Acknowledgement number (ACK): O receptor confirma o recebimento

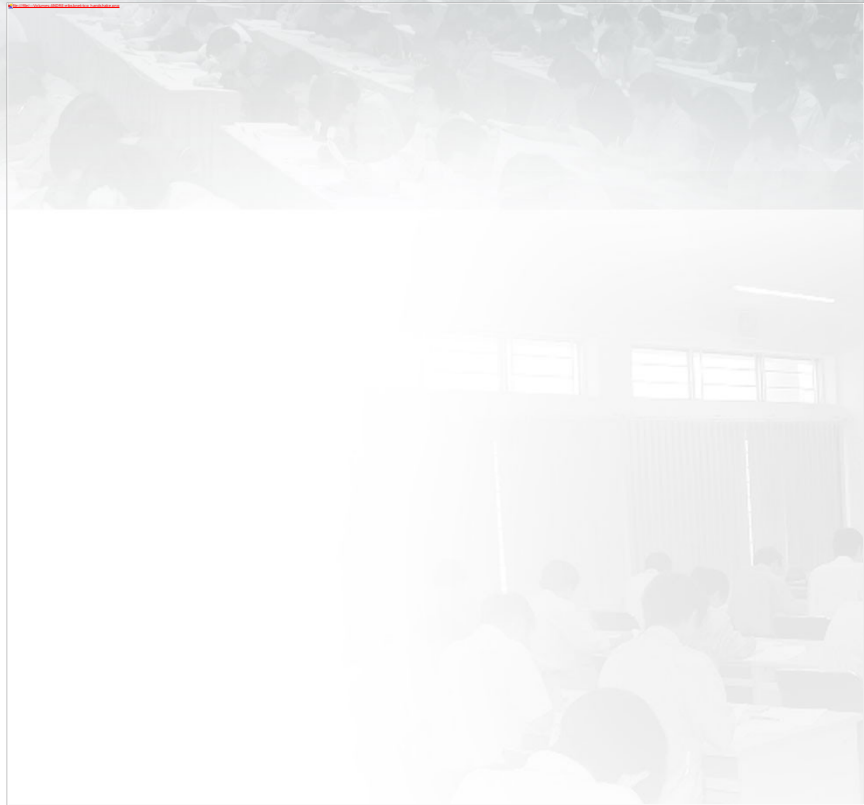


## Principais características do TCP

- A flag URG indica que o segmento contém dados urgentes e que deveriam ser processados pela máquina destino o quanto antes.
- A flag ACK indica que o segmento contém dados no campo de confirmação.
- A flag PSH indica que o segmento atual está sendo construído contém dados que devem ser entregues imediatamente.
- A flag RST é usada quando um evento causa uma desconexão indesejada.
- A flag SYN indica o pedido de abertura de uma conexão.



# Conexões – three way-handshake





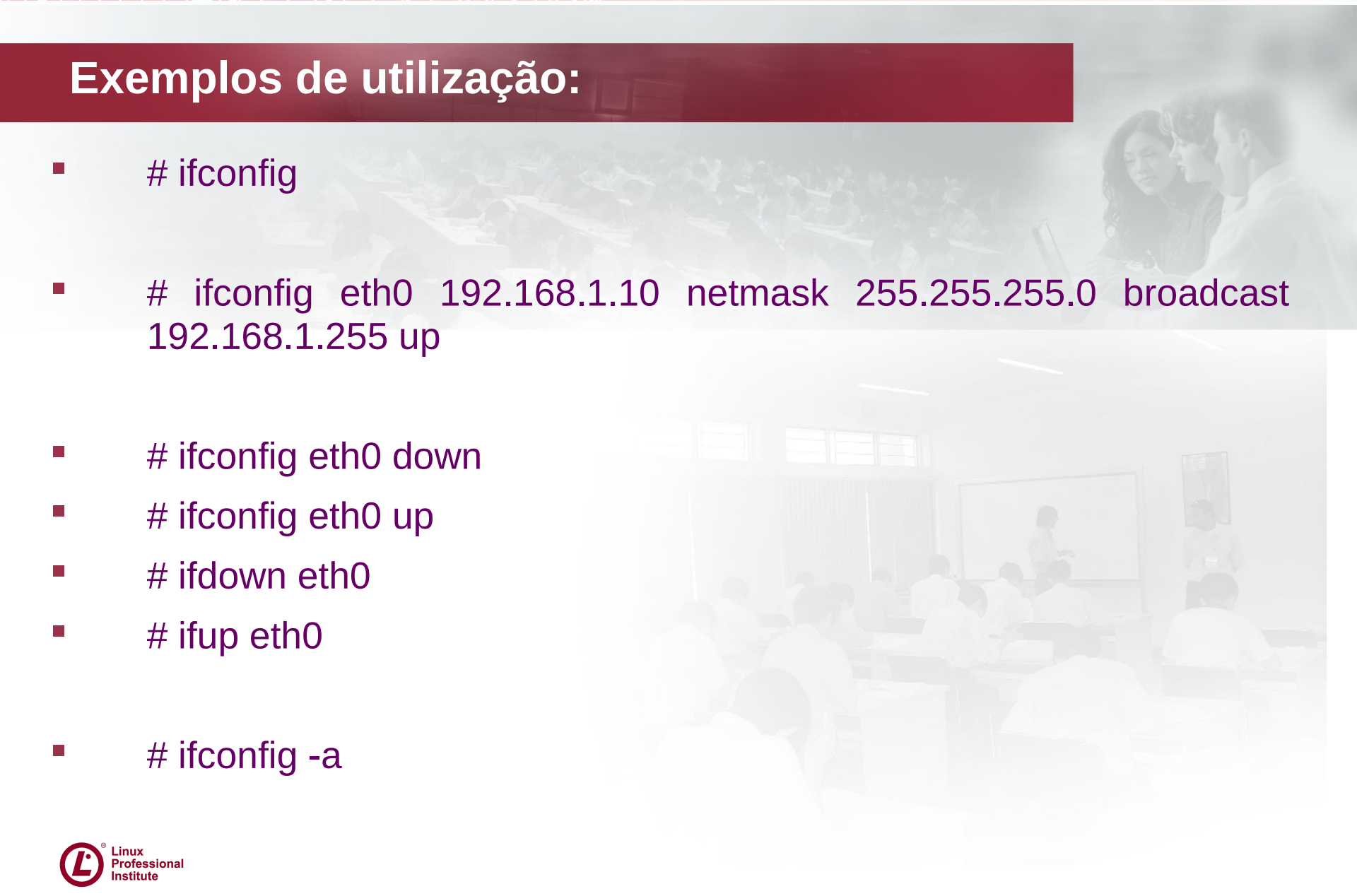
## Configurando o TCP/IP no Linux

- ifconfig: é utilizado tanto para configurar uma interface de rede como para consultar seu estado.
- **Uso:**
  - ifconfig [interface] endereço [opções] [up/down]
  - interface: a interface que será utilizada (eth0,ppp0)
  - up/down: Ativa/desativa interface
  - [endereço]: Endereço IP da interface
  - netmask : máscara de rede da interface
  - broadcast: Endereço de broadcast da rede





## Exemplos de utilização:

- `# ifconfig`
  - `# ifconfig eth0 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255 up`
  - `# ifconfig eth0 down`
  - `# ifconfig eth0 up`
  - `# ifdown eth0`
  - `# ifup eth0`
  - `# ifconfig -a`
- 



## Arquivos de configuração do TCP/IP no Linux

- `/etc/network/interfaces`: é o arquivo onde se configura as interfaces de rede no Debian (e seus derivados).
  - » `## /etc/network interfaces`
  - » `auto lo`
  - » `iface lo inet loopback`
  
  - » `auto eth0`
  - » `iface eth0 inet dhcp`



# Arquivos de configuração do TCP/IP no Linux

- auto lo
- iface lo inet loopback
- auto eth0
- iface eth0 inet static
  - address 172.16.8.200
  - netmask 255.255.0.0
  - network 172.16.0.0
  - broadcast 172.16.255.255
  - gateway 172.16.0.1
  - dns-nameservers 172.16.0.1



## Arquivos de configuração do TCP/IP no Linux

- **auto:** Inicia o NIC durante a inicialização.
- **iface:** Nome da interface.
- **inet:** O nome da família do endereço; inet=ipv4. Outras escolhas são ipx e inet6.
- **static:** O nome do método usado para configurar a interface, tanto estática (ip fixo) ou dhcp






## Configurando o TCP/IP no Linux

- Deve-se reiniciar os serviços de rede no Linux:
- `# /etc/init.d/networking restart`  
    » ou
- `# ifdown eth0`
- `# ifup eth0`
- `# invoke-rc.d networking restart`



## Configurando o TCP/IP no Linux

- O arquivo `/etc/resolv.conf` contém os servidores DNS utilizados pela máquina.
  - `## /etc/resolv.conf`
  - `nameserver 192.168.200.1`
  - `nameserver 172.16.0.1`
- 



## Configurando o TCP/IP no Linux

- O arquivo `/etc/hosts` é utilizado para resolução de nomes locais. Muito utilizado no início da internet.
- `## /etc/resolv.conf`
- `127.0.0.1 localhost`
- `172.16.8.200 instrutor.andre.net instrutor`



## Configurando rotas no Linux

- **route:** o comando route configura e exibe informações a respeito das rotas IP do sistema.
- **Uso:**
  - route [add/del] [-net/host endereço] [opções]
  - -n: não resolve nome
  - add/del: Adiciona remove uma rota
  - -net/-host IP: cria uma rota para uma rede ou host.
  - default: especifica a criação de uma rota padrão.
  - gw IP: especifica o endereço do roteador.





## Exemplos de utilização:

- `# route -n`
- `# route del -net 10.0.20.0/24`
- `# route del default`
- `# route del -net 0.0.0.0`
- `# route add default gw 172.16.0.1`
- `# route add -net 0.0.0.0 gw 172.16.0.1`
- `# route add -net 192.168.200.0/24 gw 10.0.5.1`
- `# route add -net 192.168.200.0 netmask 255.255.255.0 gw 10.0.5.1`



## Exibindo conexões de rede

- **netstat:** o comando netstat é usado para exibir o estado das conexões de rede.
- **Uso:**
  - netstat [opções]
  - -n: não resolve nome
  - -A: Especifica a família de protocolos a ser utilizada.
  - -e: Exibe informações adicionais a respeito de cada conexão.
  - -l: exibe apenas conexões em estado Listen
  - -p: exibe o nome dos processos a quem pertencem as conexões.
  - -u: protocolo udp      -t: protocolo tcp



## Exemplos de utilização:

- `# netstat -A inet`
  - `# netstat -A inet -elp`
  - `# netstat -A inet -n`
  - `# netstat -nat`
  - `# netstat -ntlp`
  - `# netstat -putan | grep <IP/porta>`
- 
- The background of the slide features a faded image of a classroom. In the foreground, the backs of several students' heads are visible as they sit at desks. In the background, a teacher is standing near a whiteboard, and other students are visible at their desks.



## Ferramentas de diagnóstico de rede

- **Traceroute:** tenta descobrir a rota percorrida por um pacote IP para chega a um determinado destino.
- O resultado dos testes nem sempre são confiáveis, já que rotas IP são potencialmente dinâmicas e é comum alguns roteadores não responderem aos pacotes IP por ele utilizados.
- **Uso: `traceroute [opções] endereço_destino`**
  - `-n` : não resolve nomes
  - `-w <tempo>`: especifica o tempo em segundos de espera de resposta para pacote enviado.





## Exemplos de utilização:

- # traceroute -n [www.brasil.gov.br](http://www.brasil.gov.br)
- # traceroute [www.caixa.gov.br](http://www.caixa.gov.br)



## Ferramentas de diagnóstico de rede

- **ping:** envia uma requisição ECHO ICMP (ICMP\_ECHO\_REQUEST) e mede o tempo de resposta (ICMP\_ECHO\_RESPONSE). Utilizado para verificar disponibilidade e responsividade de máquinas ou serviços na rede.
- **Uso: ping [opções] endereço\_destino**
  - -c : quantidade de requisições a serem enviadas
  - -i <segs>: o intervalo em segundo entre o envio de requisições. Padrão 1 segundo.



## Exemplos de utilização:

- # ping [www.brasil.gov.br](http://www.brasil.gov.br)
- # ping -c 4 -i 10 172.16.8.200
- # ping -f 172.16.0.1



## Ferramentas de diagnóstico de rede

- **mtr**: combina as funcionalidades do ping e do traceroute em uma simples ferramenta de diagnostico de rede
- **Uso: mtr [opções] endereço\_destino**
  - -n : não resolve nomes
- **# mtr www.caixa.gov.br**





## Ferramentas de diagnóstico de rede

- **telnet:** foi originalmente criado para efetuar logins remotos utilizando o protocolo TELNET. É comumente usado para efetuar conexões em modo texto em portas de diversos protocolos permitindo a depuração de problemas com protocolos que trafeguem em texto (HTTP, POP, IMAP, SMTP, etc)
- **Uso: telnet endereço\_destino porta**
  - # telnet 172.16.0.1 8080



## Ferramentas de diagnóstico de rede

- **tcpdump:** é um dos primeiros e mais flexíveis sniffers disponíveis. Trabalha em modo texto e precisa ser executado como root para capturar tráfego de rede. Utiliza o formato é o libcap é bastante comum, sendo também utilizado por outros sniffers.
- **Uso: tcpdump [opções] [expressão]**
  - -i <interface>: especifica a interface na qual o tcpdump deve capturar os pacotes.
  - -n: para não resolver nomes.
  - -w: especifica um arquivo para qual o tráfego capturado deve ser gravado.



## Ferramentas de diagnóstico de rede

- -s <bytes>: especifica o tamanho em bytes da captura a ser feita. Normalmente o tcpdump captura somente os cabeçalhos dos pacotes.
- -X: mostra os dados capturados em hexadecimal e ASCII.
- -p: coloca a placa de rede em modo promíscuo.
- [expressão]: o tcpdump tem uma “linguagem” onde é possível especificar filtros de pacotes (baseando-se em informações como portas, protocolos, hosts, etc). Leia o manual do tcpdump, muito importante.



## Exemplos do tcpdump

- **# tcpdump -i eth0 -w teste.pcap**
  - Executa o tcpdump conectado a interface eth0, enviando o tráfego capturado para o arquivo teste.pcap
- **# tcpdump -n “port 22 and host 172.16.8.200”**
  - Não resolve nomes e captura o tráfego apenas da e da porta 22 que tenho como origem ou destino o host 172.16.8.200.
- **# tcpdump -i eth0 “src host 172.16.8.200 and not port 22”**
  - Captura o tráfego que tenha origem o host 172.16.8.200 e que não seja da ou para a porta 22



# Administração de Redes Linux Serviços





## Acesso remoto: telnet

- O telnet era usado antigamente para conexão remota a servidores e equipamentos. Ainda hoje, alguns equipamentos só permitem acesso via telnet. Sua segurança é fraca pois os dados são transmitidos em texto puro.
- Instalar o servidor telnet:  
`# aptitude install telnetd`
- Verificar se a porta 23 abriu  
`# netstat -nat | grep :23`



## Acesso remoto

- Conecte em uma máquina
  - # telnet 172.16.8.x
- Testando uma conexão através do telnet
  - # telnet 172.16.8.200 80



## Acesso remoto: ssh

- O openssh provê uma implementação aberto do protocolo SSH (Secure Shell), que permite a execução de um shell remoto (login remoto) ou outros comandos de maneira segura através de canais criptografados.
- Foi originado do openbsd.
- Possui algumas ferramentas úteis para gerenciamento de arquivos pela rede.
- Sua instalação é simples:
  - `# aptitude install openssh-server`





## Etapas de uma conexão ssh

- **1.** O Cliente conecta via TCP no servidor ssh.
- **2.** Servidor aceita a conexão TCP e responde com algumas informações.
  - Parâmetros de criptografia
  - Fingerprint (Identificação criptográfica) do servidor.
- **3.** O cliente verifica o fingerprint recebido contra o arquivo `~/.ssh/known_hosts`, podendo acontecer uma das alternativas abaixo:
  - Host conhecido e fingerprint correto: pode continuar
  - Host desconhecido: deixa o usuário decidir-se pode continuar ou não.
  - Host conhecido mas fingerprint incorreto: emite aviso de possível ataque e interrompe a conexão.





## Etapas de uma conexão ssh

- 4. Sessão criptografada estabelecida. Somente agora que o canal de comunicação está protegido é que o usuário tem a chance de inserir sua senha ou utilizar algum outro método de autenticação.
- Ao fazer a conexão ssh a um servidor pela primeira vez, o fingerprint será gerado e exibido na tela, e pergunta para o usuário se pode continuar.



## Conexão ssh

- `ssh [opções] [usuário@] <host> [comandos remotos]`
- `# ssh 172.16.8.x`
- `# ssh -l root 172.16.8.x`
- `# ssh root@172.16.8.x`
- `# ssh 172.16.8.x ls /etc`
- `# ssh 172.16.8.x ls /etc > /tmp/saida.txt`
- `# ssh 172.16.8.x tar czf - /var > backup_var.tar.gz`



## Envio/recebimento de arquivos - scp

- Dentro do pacote do openssh temos a ferramenta scp (secure copy). Permite que arquivos sejam enviados entre máquinas utilizando os mesmos recursos de autenticação e criptografia que o ssh.
- **scp [opções] <origem> <destino>**
- -r: faz uma cópia recursiva, ou seja, entrando em diretórios.
- -v: verboso, exibe informações adicionais.
- -C: ativa compactação de dados.
- -P: porta



## Envio/recebimento de arquivos - scp

- **# scp arquivo 172.16.8.x:/tmp**
  - Envia “arquivo” do diretório atual para o diretório /tmp na máquina remota.
- **# scp root@172.16.8.x:/dados/\* .**
  - Cópia todos os arquivos do diretório /dados da maquina remota para o diretório atual no cliente.
- **# scp -rC 172.16.8.x:/dados /tmp**
  - Copia recursivamente o conteúdo do diretório /dados da máquina remota para o diretório /tmp local utilizando compressão dos dados durante a transmissão.





## Arquivo de configuração do servidor ssh

- Editar o arquivo `/etc/ssh/sshd_config`
- **Port:** Especifica em qual porta o servidor deve “escutar”.
- **Protocol:** Especifica a versão do protocolo SSH a ser disponibilizada por este servidor.
- **PermitRootLogin:** Permite que sejam efetuados logins como usuário root. Não recomendável. Inserir valor “no”.
- **ListenAdress:** Indica qual interface poderá aceitar conexões.
- **AllowUsers:** Informa qual usuário pode efetuar login remoto.
- **AllowGroups:** Informa os membros do grupo que podem acessar o servidor remotamente.



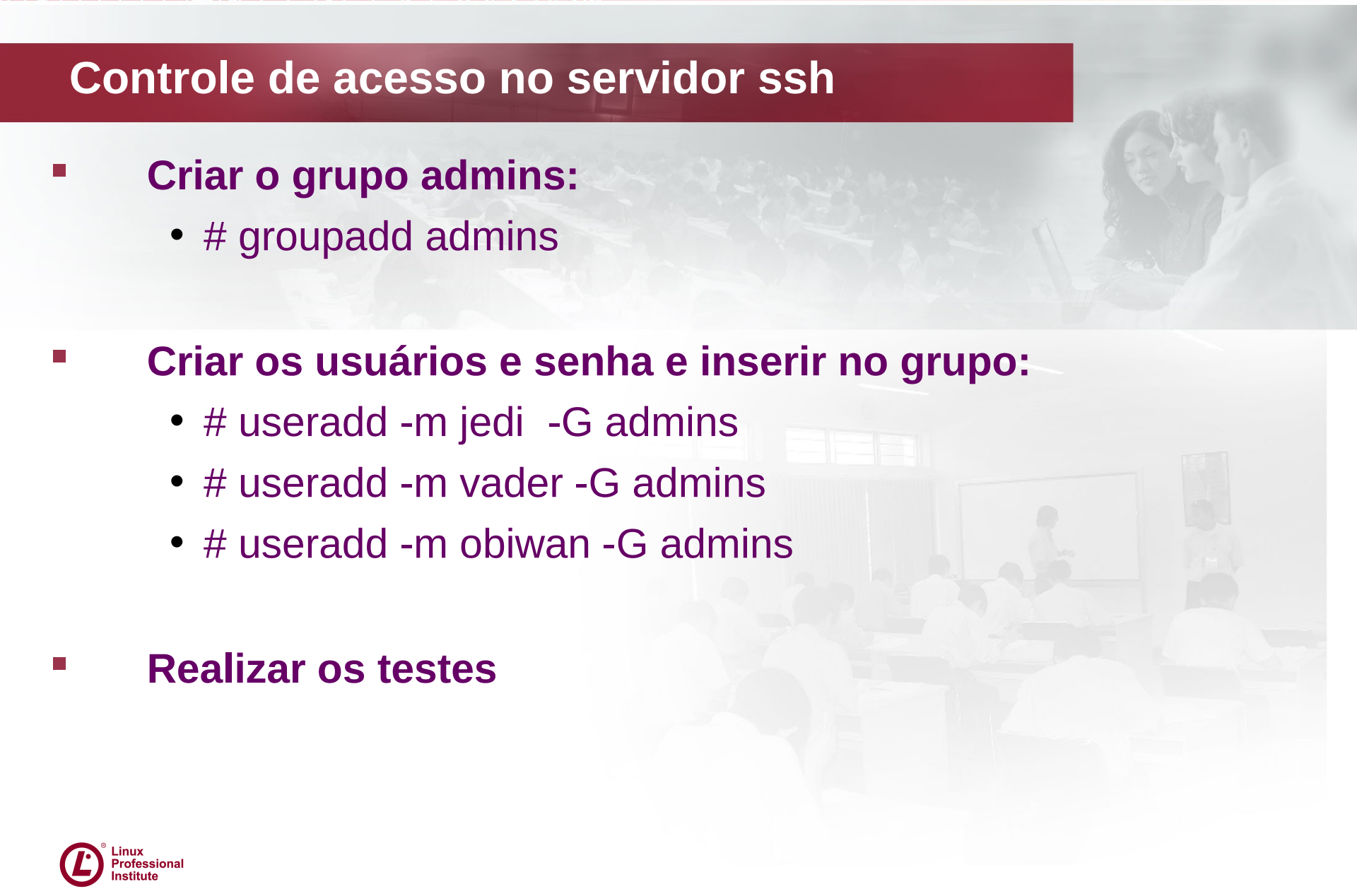


## Controle de acesso no servidor ssh

- **Port 65123**
- **ListenAddress 172.16.8.x**
- **PermitRootLogin no**
- **AllowGroups admins**
- **DenyUsers obiwan**



## Controle de acesso no servidor ssh

- **Criar o grupo admins:**
    - `# groupadd admins`
  - **Criar os usuários e senha e inserir no grupo:**
    - `# useradd -m jedi -G admins`
    - `# useradd -m vader -G admins`
    - `# useradd -m obiwan -G admins`
  - **Realizar os testes**
- 



## FTP seguro - sftp

- Trata-se de um sistema semelhante ao ftp em termos de comandos, mas utilizando os recursos de criptografia e autenticação do ssh.
- Para ativá-lo, basta habilitar essa opção no arquivo de configuração do servidor:
  - **/etc/ssh/sshd\_config**
  - Subsystem sftp /usr/lib/openssh/sftp-server
- `# sftp jedi@10.1.20.201`



## Autenticação por chave

- Funcionalidade muito interessante do OpenSSH, onde ao invés de utilizar uma simples senha para acesso (que potencialmente tem poucos caracteres), utiliza-se um arquivo com uma chave criptográfica (que contém algo como 1024 ou 2048 bits).
- O objetivo aqui é criar um conjunto de chaves no cliente para acessar via ssh servidores remotos sem ter que digitar a senha e sim digitar uma frase utilizando a determinada chave, ou seja, estabelecer uma relação de confiança entre cliente e servidor remoto via chaves públicas.
- O usuário deve ser o mesmo no cliente e no servidor.



# Autenticação por chave

- Criar o usuário toor no servidor e no cliente. Gerar a chave pública que vai ser compartilhada.
- **1. No cliente**
  - # su – toor
  - \$ ssh-keygen -t rsa
  - \$ cd .ssh
  - \$ ssh-copy-id -i id\_rsa.pub toor@172.16.8.x





# Autenticação por chave

- **2. No Servidor**
  - `# cd /home/toor/.ssh`
  - `# ls`
  - `# cat authorized_keys`
- `# vim /etc/ssh/sshd_config`
- `ChallengeResponseAuthentication no`
- Reinicie o ssh



## Autenticação por chave

- **2. No cliente**
  - `$ ssh 172.16.8.x`
- Agora faça um teste, crie o usuário yoda no servidor e no cliente, e na geração da frase senha deixe em branco e efetue o procedimento.



## Servidor NFS

- O servidor NFS (Network File system) – sistema de arquivos remotos – é o serviço de compartilhamento de arquivos padrão do UNIX e Linux.
- O servidor NFS tem 2 pacotes relacionados:
  - **nfs-common**: traz binários e documentação necessários tanto para o cliente quanto para o servidor NFS.
  - **nfs-kernel-server**: implementação do servidor NFS diretamente no kernel do Linux.



## Daemons e utilitários para o NFS

- **Portmap:**
  - Um servidor para o serviço de diretório RPC.
- Os daemons RPC relacionados ao NFS: são tipicamente iniciados automaticamente em tempo de boot através do comando `/etc/init.d/nfs start` e consistindo em:
  - **rpc.nfsd:** suporta o serviço de arquivos.
  - **rpc.statd** e **rpc.lockd:** suporta o sistema de bloqueio.
  - **rpc.rpquotad:** gerencia quotas
  - **rpc.mountd:** verifica requisições de montagem e disponibiliza o suporte de acesso.



## Instalando Servidor NFS

- **# aptitude install nfs-kernel-server**
  - Existe apenas um arquivo de configuração para o servidor NFS.
  - Nele será inserido o diretório que você deseja compartilhar com suas respectivas opções. Ex:
  - **# mkdir /dados**
  - **# vim /etc/exports**
- /dados \*(rw,sync,no\_root\_squash,no\_subtree\_check)**





## Configurando Servidor NFS

- **rw** (read write): permite que o cliente grave na pasta.
- **ro** (read only): permite que o cliente somente leia na pasta.
- **async** : permite transferência de dados assíncrona, e portanto, mais rápida.
- **sync** : permite transferência de dados síncrona.
- **root\_squash** : bloquear o acesso ao root, ou seja, modifica o UID das requisições de root para o usuário nobody antes de eles irem ao sistema de arquivos.
- **no\_root\_squash** : permite que o root remoto possa alterar arquivos no compartilhamento com as mesmas permissões do root local. O UID mantém.



## Configurando Servidor NFS

- **all\_squash** : Rebaixa todos os usuários a usuário nobody.
- **no\_subtree\_check** : permite que o volume todo que foi compartilhado seja exportado.



## Configurando Servidor NFS

- O arquivo `/etc/exports` será lido quando o servidor NFS iniciar.
- `# /etc/init.d/nfs-kernel-server start`
- Sempre que fizer alguma alteração no arquivo e o servidor já estiver rodando, utilize o comando `exportfs` para atualizar a lista de compartilhamentos disponíveis no servidor.]
  - `# exportfs -ar`



## Opções do comando exportfs

- -a : A operação afetará todos os diretórios exportados.
- -r : Re-exporta todos os diretórios mencionados em /etc/exports. Usado quando o arquivo de configuração sofreu alguma alteração.
- -u: Remove a exportação do diretório.
- -v: Modo detalhado.
- -o: Opções NFS a serem usadas para o diretório em particular.



## Configurando Servidor NFS

- Para verificar quais os compartilhamentos disponíveis execute o comando:
- `# showmount -e localhost`
- `# showmount -a localhost`





## Configurando Cliente NFS

- Instalar os pacotes:
  - `# aptitude install nfs-common portmap`
- `# /etc/init.d/portmap restart`
- Ver quais diretórios estão sendo exportados, compartilhados.
  - `# showmount -e 172.16.8.200`



## Configurando Cliente NFS

- Montar o diretório remoto na máquina local.
- `# mount -t nfs 172.16.8.200:/dados /mnt`
- `# cd /mnt`
- Deixando automático:
- `# vim /etc/fstab`
- `172.16.8.200:/dados /mnt nfs auto,users,exec 0 0`



## Servidor FTP

- O File Transfer Protocol – FTP – é um protocolo utilizado para transferência de arquivos entre estações.
- O intuito do FTP é unicamente prover uma forma de transferência de arquivos na internet, ou na rede local.
- Deve-se ter um usuário e uma senha válida.
- Baseado nesse login, ele direciona o usuário para o diretório pessoal do usuário indicado, onde ele tem permissão de acesso.



## Servidor FTP

- Usuário anonymous
  - Conta especial para acesso público, com permissão apenas de leitura no servidor FTP, ou seja, ele só consegue baixar arquivos do servidor.
- Instalar o servidor FTP
  - `# aptitude install vsftpd`
- Editar o arquivo `/etc/vsftpd.conf`



## Servidor FTP

- Efetuando login em servidor FTP
  - # ftp 172.16.8.200
  - ftp> help





## Servidor DHCP

- Conceitos de DHCP (Dynamic Host Configuration Protocol):
- é baseado no protocolo bootp. Assim como o bootp, ele permite que um administrador defina dinamicamente características aos clientes que conectarem a rede.
- Isto elimina a necessidade de se configurar informações de rede como DNS, gateway e endereços IP aos clientes. Este protocolo é muito útil por exemplo, se utilizado com laptops e notebooks que são utilizados em várias redes diferentes.

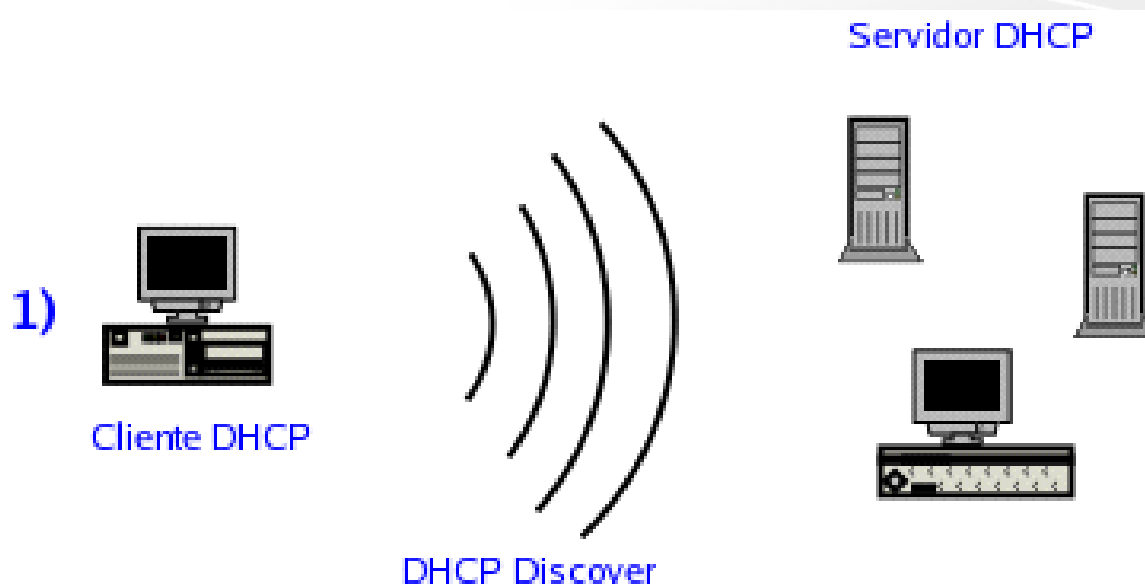


## Servidor DHCP

- O DHCP também é vital quando utilizado em grandes redes, onde manter sob controle todos os endereços e configurar novos clientes pode gerar uma grande dor de cabeça.
- Outra vantagem é a reutilização de endereços IP: tão logo um cliente se desconecte da rede, o mesmo endereço usado por ele pode ser utilizado para o próximo novo cliente.

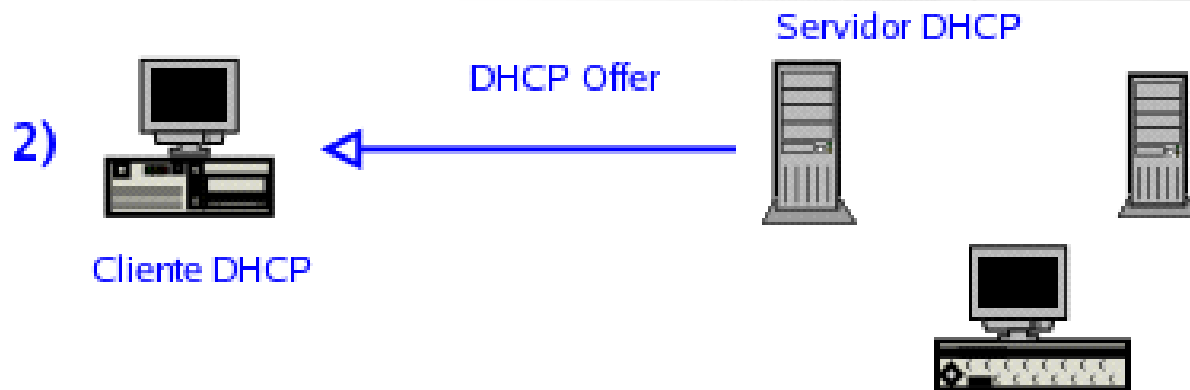
## Como o DHCP funciona?

1. O cliente DHCP envia um pacote broadcast para a rede chamado DHCP Discover. Este pacote tem como destino o endereço IP 255.255.255.255



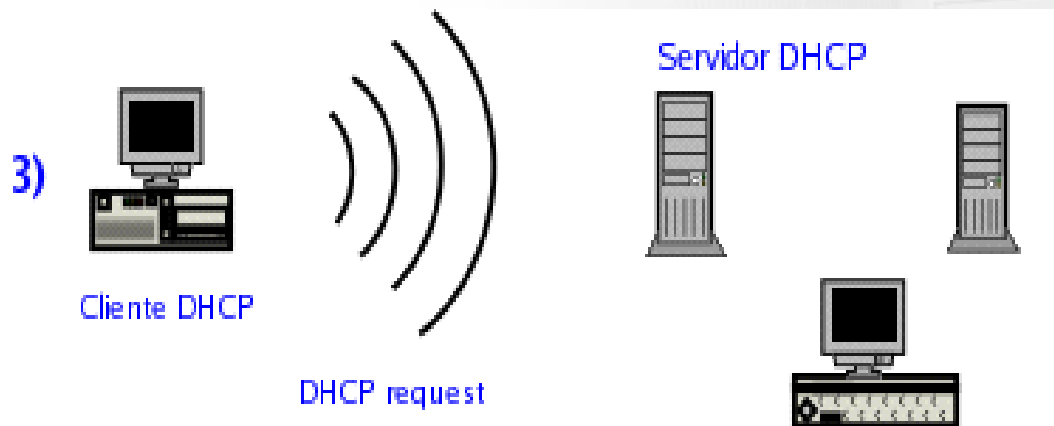
## Como o DHCP funciona?

2. O servidor DHCP, ao receber o pacote, consulta sua tabela de configuração e prepara uma resposta para a máquina cliente chamado de DHCP Offer. Esta resposta já contém uma oferta de configuração para o cliente, mas nada foi reservado ainda.



## Como o DHCP funciona?

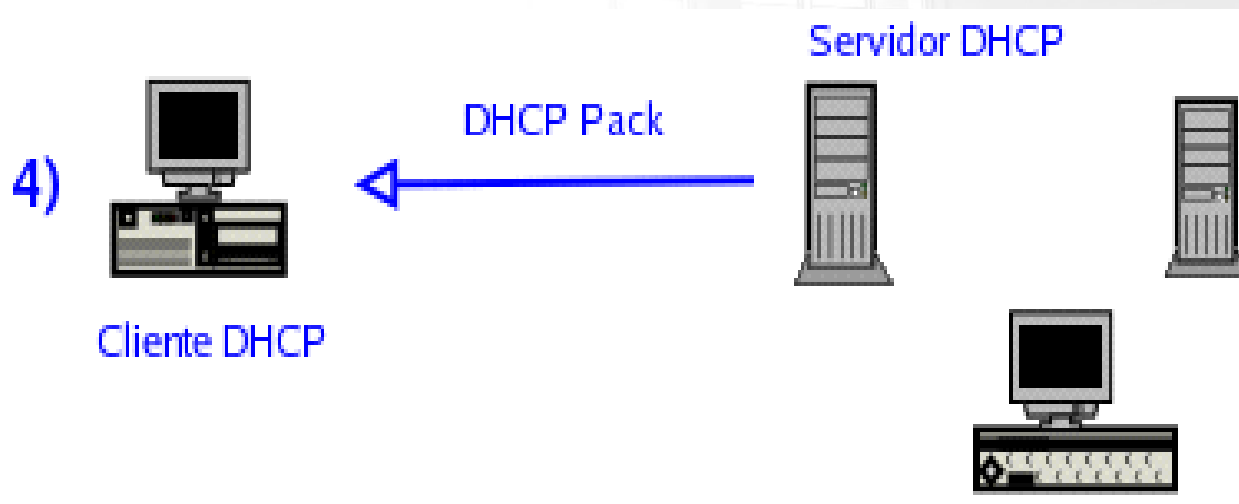
3. O cliente, se concordar com a oferta do servidor, fará um pedido “formal” de configuração específico para este servidor. Este pedido é feito através de um pacote chamado DHCP REQUEST que basicamente contém informações que o servidor disse ter disponíveis.





## Como o DHCP funciona?

- Se as informações ainda estiverem válidas, o servidor responderá com um pacote chamado DHCP PACK, e gravará em sua base de dados local que este endereço IP oferecido deixou de estar vago .





# Instalando Servidor DHCP

- `# aptitude install dhcp3-server`
- O cliente já vem instalado na grande maioria dos sistemas operacionais.



# Configurando Servidor DHCP

- Editar o arquivo `/etc/dhcp3/dhcpd.conf`
- `# vim /etc/dhcp3/dhcpd.conf`



## Configurando Servidor DHCP

# Modo de atualização do servidor DNS ( não utilizado  
# nesse exemplo mas necessário estar presente )

***ddns-update-style none;***

# Este servidor é a autoridade na rede. Se existirem outros  
# servidores DHCP, os comandos deste tem prioridade

***authoritative;***



## Configurando Servidor DHCP

# Tempo padrão (em segundos) de empréstimo de um IP  
**default-lease-time 28800;**

# Tempo máximo de empréstimo (também em segundos)  
**max-lease-time 43200;**

# Endereço do servidor DNS  
**option domain-name-servers 192.168.200.1**





## Configurando Servidor DHCP

# Vamos configurar a subrede 192.168.200.0

```
subnet 192.168.200.0 netmask 255.255.255.0 {  
option broadcast-address    192.168.200.255;  
option routers              192.168.200.1;  
option subnet-mask          255.255.255.0;  
range 192.168.200.100      192.168.200.180;  
}
```



## Iniciando o Servidor DHCP

- `# /etc/init.d/dhcp3-server start`
- Buscando IP nas estações:
  - `# dhclient`
- Visualize as estações solicitando endereço:
  - `# tail -f /var/log/messages`
- Verifique a reserva de endereços no arquivo:
  - `# cat /var/lib/dhcp3/dhcpd.leases`



## Configurando interface de rede para DHCP

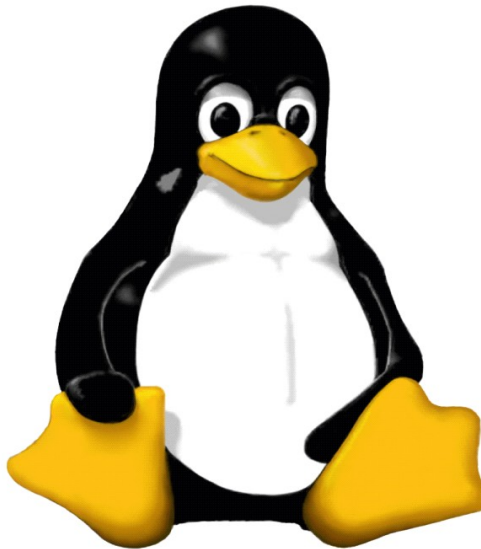
- **# vim /etc/network/interfaces**
- **iface eth0 inet dhcp**
- **auto eth0**
- **Salve o arquivo e reinicie o serviço de rede**
- **# /etc/init.d/networking restart**



# Configurando Servidor DHCP

- Editar o arquivo `/etc/dhcp3/dhcpd.conf`
- `# vim /etc/dhcp3/dhcpd.conf`

# Integrando redes heterôgeneas







## Tópicos abordados

- O que é o SAMBA
- História do SAMBA
- Instalação do SAMBA
- Construindo um servidor de arquivos





## Tópicos abordados

- Entender a história do samba
- Instalar e entender o arquivo de configuração
- Compartilhar um diretório na rede
- Criar um servidor de arquivos na rede PDC





## O que é o SAMBA

- O SAMBA é um servidor e um conjunto de ferramentas que permite que máquinas Linux e Windows se comuniquem entre si, compartilhando serviços (arquivos, diretórios, impressão) através do protocolo smb(Server Message Block) e atualmente com o CIFS(Common Internet File System).





## O que pode ser feito com o Samba

- Construir domínios completos
- Controle de acesso no nível de usuário
- Compartilhamento de arquivos e diretórios
- Servidor Wins
- Servidor de domínio
- Impressão





## História do Samba

- Andrew Tridgell desenvolveu o SAMBA porque precisava montar um volume Unix em sua máquina DOS.
- No início utilizou o NFS, mas o aplicativo precisava ter suporte NetBIOS.
- Escreveu um sniffer de pacotes, para analisar e auxiliá-lo a interpretar todo o tráfego NetBIOS da rede.
- Ele escreveu o primeiro código, que fez o servidor Unix aparecer como um servidor de arquivos Windows para sua máquina DOS.







## História do Samba

- Deixou de lado por quase dois anos
- Um dia ele resolveu testar a máquina windows de sua esposa com sua máquina Linux , e ficou maravilhado com o funcionamento do programa que criou e veio a descobri que o protocolo era documentado e resolveu levar o trabalho adiante.
- Site do projeto samba: [www.samba.org](http://www.samba.org)





## Utilitários importantes

- **smbclient** – ferramenta para navegação e gerenciamento de arquivos, diretórios, impressoras compartilhadas por servidores Windows ou SAMBA
- **smbfs** – pacote que possui ferramentas para o mapeamento de arquivos e diretórios compartilhados por servidores Windows ou SAMBA em um diretório local.
- **winbind** – Daemon que resolve nomes de usuários e grupo através de um servidor NT/SAMBA e mapeia os UIDs/GIDs deste servidor como usuários locais.



## Instalação do Samba

- No debian instalar os pacotes:
  - samba
  - winbind
- `# aptitude install samba winbind`
- `# dpkg -l | grep samba`
- `# dpkg -l | grep winbind`



## Configurando o Samba

- O arquivo de configuração do samba fica dentro de /etc/samba e se chama smb.conf.
- `# vim /etc/samba/smb.conf`





## Seções do Samba

- As seções têm o objetivo de organizar os parâmetros para que tenham efeito somente em algumas configurações de compartilhamento do servidor.
- Alguns nomes de seções são reservadas para configurações específicas do SAMBA.
- **[global]** : definem configurações que afetam o servidor SAMBA como um todo, fazendo efeito em todos os compartilhamentos existentes na máquina.
- **[homes]**: especifica opções de acesso a diretórios pessoais de usuários. O diretório home é disponibilizado somente para seu dono, após se autenticar no sistema.





## Seções do Samba

- **[printers]** : Define opções gerais para controle das impressoras do sistema. Este compartilhamento mapeia os nomes de todas as impressoras encontradas no /etc/printcap
- **[profile]**: Define um perfil quando o servidor SAMBA é usado como Controlador de Domínio.
- **[netlogon]**: Define o caminho onde o SAMBA irá executar os scripts de logon.



## Parâmetros do Samba

- Um parâmetro é definido no formato nome = valor
- Definem opções do servidor samba
- Podem conter as seguintes opções:
  - 0 ou 1
  - yes ou no
  - true ou false
- Assim as seguintes configurações são válidas
  - master browse = 0
  - master browse = no
  - master browse = false



## Missão 1: Compartilhando um diretório

Utilizar o Samba Server para autenticar usuários antes de compartilhar os arquivos de sua rede.

Editar o arquivo de configuração do samba

```
# vim /etc/samba/smb.conf
```



## Missão 1: Compartilhando um diretório

[**global**]

workgroup = CLASSIC

netbios name = servidor samba

encrypt passwords = Yes

security = user

os level = 65

preferred master = yes

domain master = no

local master = yes





## Missão 1: Compartilhando um diretório

Vamos cadastrar os usuários que acessarão o servidor.  
Primeiro vamos criar um grupo específico:

```
# groupadd -g 201 maquinas
```

```
# useradd -d /home/arquivos -s /bin/false -g maquinas  
miranda
```

```
# passwd miranda
```

Criar o usuário no Samba

```
# smbpasswd -a miranda
```





## Missão 1: Compartilhando um diretório

Criar os diretórios que os usuários poderão visualizar ao se conectarem no sistema.

```
# vim /etc/samba/smb.conf
```

```
[homes]
```

```
read only = no
```

```
browseable = no
```



## Missão 1: Compartilhando um diretório

Criar um compartilhamento público:

[publico]

path = /home/publico

browseable = yes

read only = no

write list = miranda

write list = @maquinas



## Missão 1: Compartilhando um diretório

```
# mkdir /home/publico  
chmod 777 /home/publico
```

```
# /etc/init.d/samba restart
```

```
# ps -aux | grep samba
```

```
# smbstatus
```

```
# smbclient -L 192.168.x.x
```

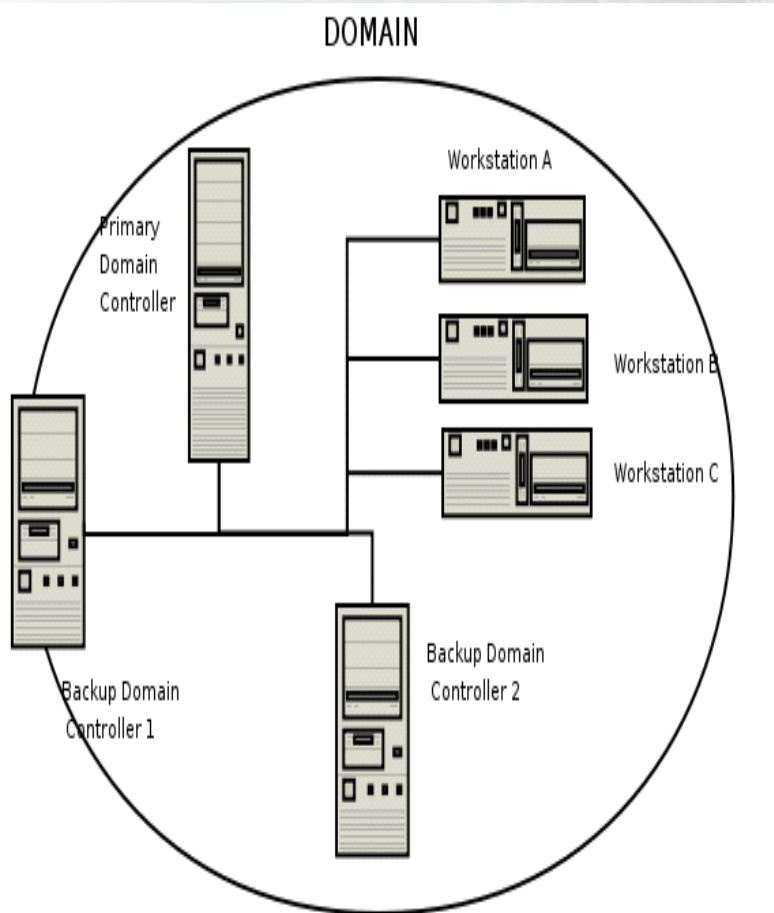


## Missão 1: Compartilhando um diretório

### Testando o samba

- Para verificar se você digitou tudo certinho, o Samba oferece a ferramenta testparm, que mostra erros dentro do arquivo de configuração. Esta ferramenta é bastante útil para encontrar erros sempre que criamos ou modificamos alguma coisa. Use-a!

## Missão 2: SAMBA como PDC



- Controlador de Domínio Primário
- Centralizar os serviços de armazenamento de arquivos





## Missão 2: SAMBA como PDC

### Resolução de Nomes

- As conexões dentro da rede com o samba devem saber de onde e para onde devem seguir.
- O Samba oferece suporte para WINS, cuja resolução de nomes ocorre de maneira fácil e prática com a ajuda do arquivo `/etc/hosts`



## Missão 2: SAMBA como PDC

### Configurando o /etc/hosts

192.168.20.1	chefe
192.168.20.2	win01
192.168.20.3	lin01

# ping chefe

# ping win01



## Missão 2: SAMBA como PDC

### Editar o arquivo `/etc/samba/smb.conf`

**[global]**

workgroup = AMS

server string = Servidor de Arquivos

passdb backend = smbpasswd:/etc/samba/smbpasswd

username map = /etc/samba/smbusers

name resolve order = wins bcast hosts

printcap name = CUPS



## Missão 2: SAMBA como PDC

printing = CUPS

logon drive = H:

logon script = scripts\logon.bat

logon path = \\chefe\profile\%U

domain logons = yes

preferred master = yes

wins support = yes





## Missão 2: SAMBA como PDC

```
add user script = /usr/sbin/useradd -m '%u'
delete user script = /usr/sbin/userdel -r '%u'
add group script = /usr/sbin/groupadd '%g'
delete group script = /usr/sbin/groupdel '%g'
add user to group script = /usr/sbin/usermod -G '%g' '%u'
add machine script = /usr/sbin/useradd -s /bin/false -d
/var/lib/nobody '%u'
```





## Missão 2: SAMBA como PDC

[homes]

comment = Pastas pessoais

valid users = %S

read only = no

browseable = no



## Missão 2: SAMBA como PDC

[printers]

comment = Gráfica

path = /var/spool/samba

printable = yes

guest ok = yes

browseable = no



## Missão 2: SAMBA como PDC

```
add user script = /usr/sbin/useradd -m '%u'
delete user script = /usr/sbin/userdel -r '%u'
add group script = /usr/sbin/groupadd '%g'
delete group script = /usr/sbin/groupdel '%g'
add user to group script = /usr/sbin/usermod -G '%g' '%u'
add machine script = /usr/sbin/useradd -s /bin/false -d
/var/lib/nobody '%u'
```



## Missão 2: SAMBA como PDC

[netlogon]

comment = Netlogon

path = /servidor/netlogon

valid users = %U

read only = no

browseable = no



## Missão 2: SAMBA como PDC

**[profile]**

comment = Perfis de usuario

path = /servidor/profiles

valid users = %U

create mode = 0600

directory mode = 0700

writable = yes

browseable = no





## Missão 2: SAMBA como PDC

[geral]

comment = Arquivos da Empresa

path = /servidor/geral

read only = no



## Missão 2: SAMBA como PDC

### Adicionando Usuários

```
# smbpasswd -a root
```

```
# vi /etc/samba/smbusers
```

```
root = Administrator
```

```
# useradd batman -m -G users
```

```
# passwd batman
```

```
# smbpasswd -a batman
```



## Missão 2: SAMBA como PDC

### Sincronizando os Grupos

O mapeamento entre os grupos padrão do Windows e os do Linux é essencial para que as estações de trabalho pensem que o Samba é na verdade um servidor Windows NT.



## Missão 2: SAMBA como PDC

### Sincronizando os Grupos

```
# groupadd domadmin
```

```
# net groupmap add ntgroup="Domain Admins"  
unixgroup=domadmin rid=512 type=d
```

```
# net groupmap add ntgroup="Domain Users"  
unixgroup=users rid=513 type=d
```

```
# net groupmap add ntgroup="Domain Guests"  
unixgroup=nobody rid=514 type=d
```



## Missão 2: SAMBA como PDC

### Iniciando o samba

```
# /etc/init.d/samba restart
```

```
# /etc/init.d/winbind restart
```





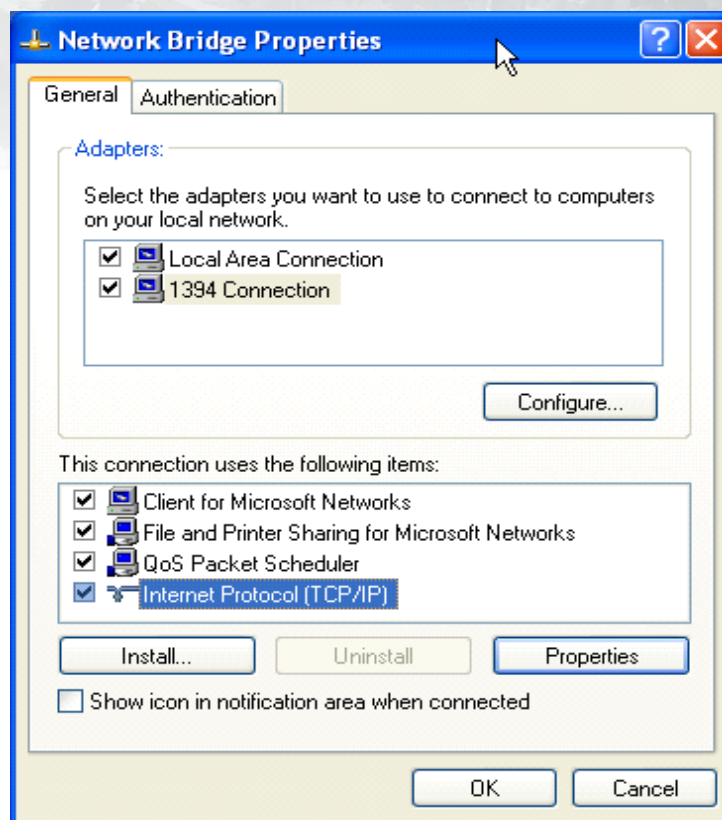
## Missão 2: SAMBA como PDC

### Criando diretórios

```
# mkdir -p /servidor/geral  
# chown -R root:users /servidor/geral  
# chmod -R ug+rw, o+rx-w /servidor/geral  
  
# mkdir /servidor/publico
```

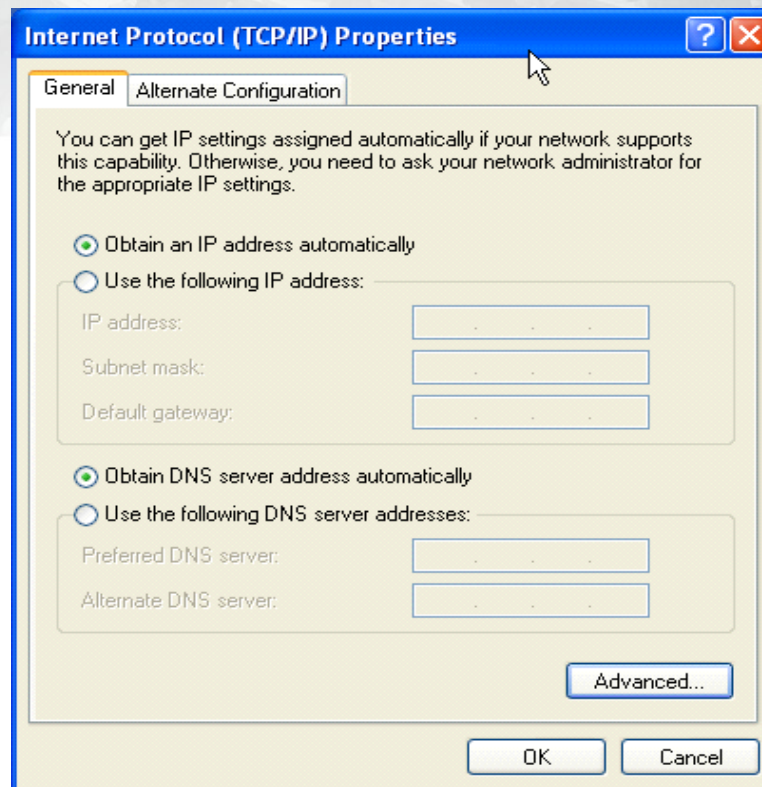
## Missão 2: Configurando o Windows.

### Sem comentários



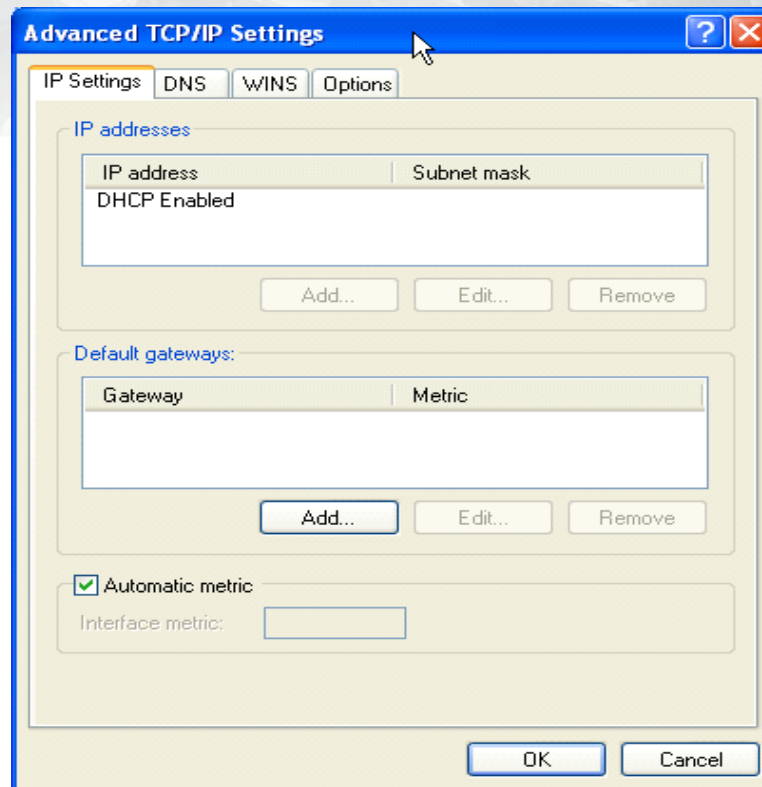
# Missão 2: Configurando o Windows.

## Sem comentários



# Missão 2: Configurando o Windows.

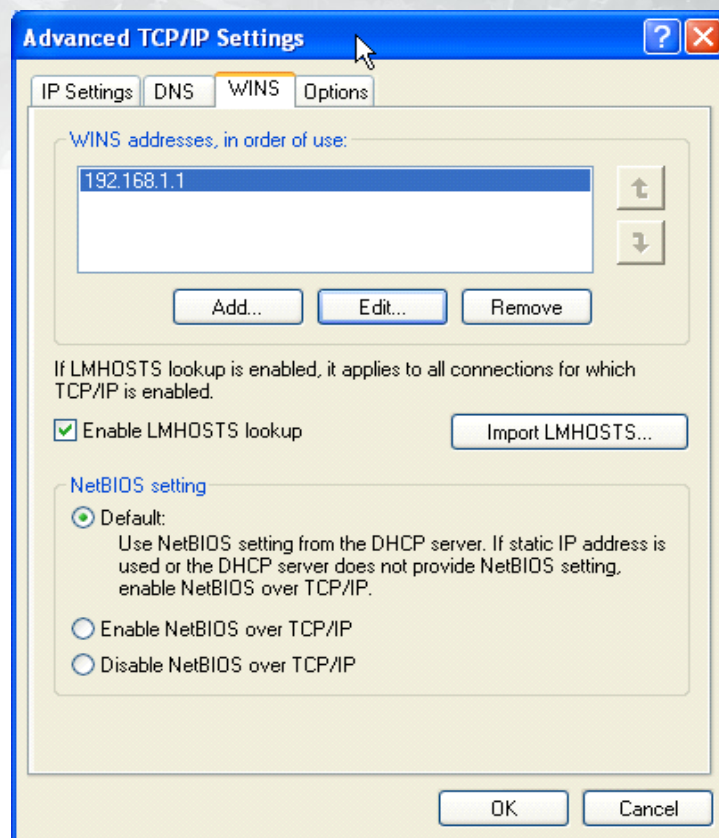
## Sem comentários





# Missão 2: Configurando o Windows.

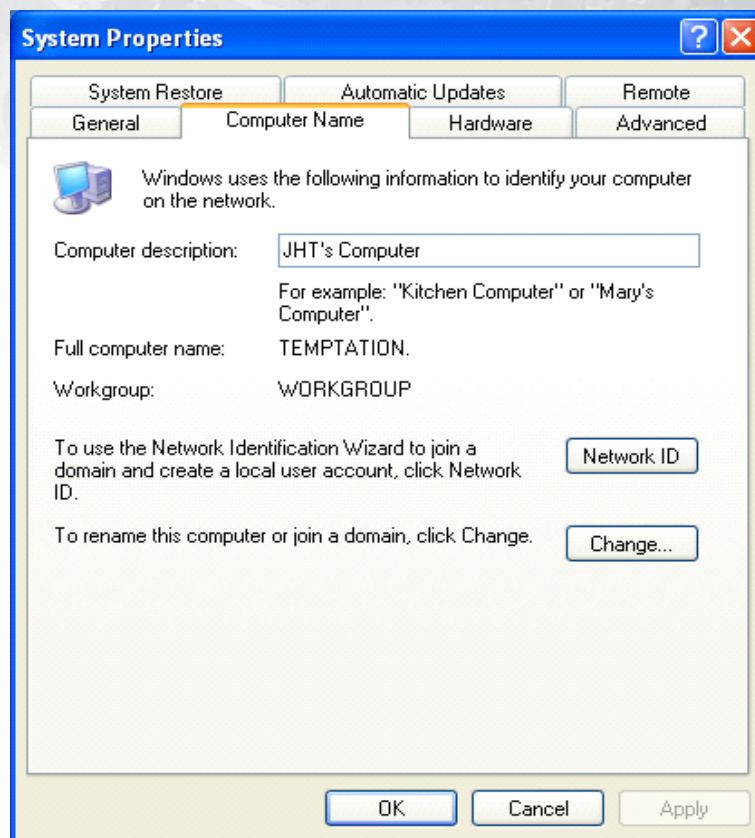
## Sem comentários





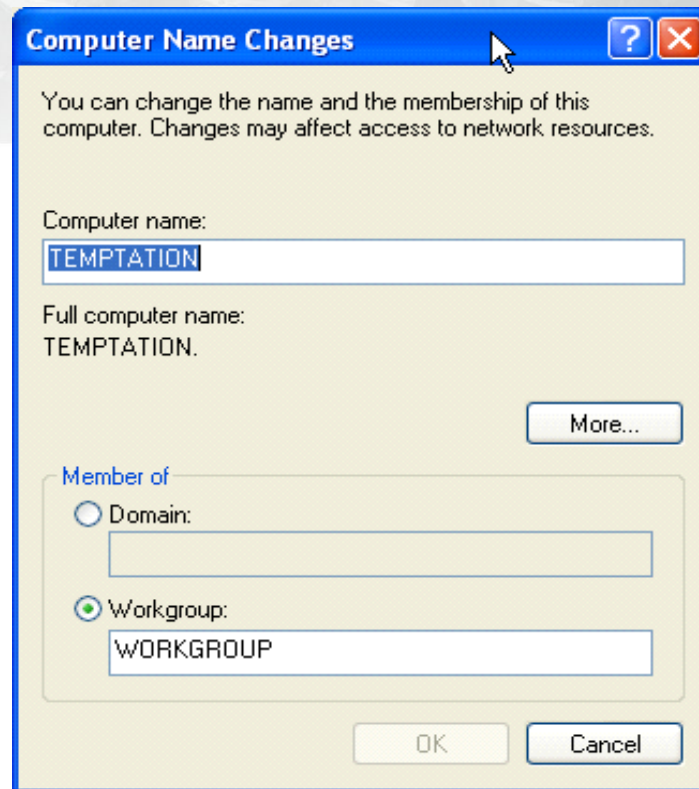
# Missão 2: Configurando o Windows.

## Sem comentários



## Missão 2: Configurando o Windows.

### Sem comentários



## Missão 2: Configurando o Windows.

Sem comentários





## Sistema de Resolução de Nomes

- Com o passar dos anos, aliado ao crescimento da Internet, tornou-se inviável que os usuários tivessem que saber o endereço IP para cada recurso que pretendiam acessar.
- É mais fácil referenciar uma pessoa pelo seu nome do que pelo seu número de celular.
- É exatamente isso que um sistema de resolução de nomes faz: Traduzir nomes para endereços IP.
- O primeiro sistema de resolução de nomes criado foi o hosts.





## Sistema de Resolução de Nomes: hosts

- Esse sistema de resolução que desencadeou a criação de todos os outros sistemas de resolução posteriores.
- Funcionamento simples
- No Linux é o arquivo `/etc/hosts`
  - `127.0.0.1 localhost`
  - `172.16.8.200 instrutor.eib.net instrutor`
- **Problemas:** Com o passar do tempo, o aumento do número de hosts cresceu muito, os computadores precisavam baixar um arquivo de hosts para poder utilizar a resolução de nomes oferecida pelo sistema.





# DNS

- A sigla DNS significa, em inglês, *Domain Name System*, ou Sistema de Nomes de domínio.
- Na internet representa um gigantesco catálogo de endereços e serviços e é a base para diversas tecnologias que hoje temos como básicas.



## Nomes e domínios

- Um domínio nada mais é do que uma subarvore do espaço de nomes de domínio.
- O nome de um domínio é o nome do ramo que está naquele domínio.
- Cada subarvore é considerada parte de uma domínio.
- Os domínios localizados nas pontas dos ramos representam máquinas individuais.

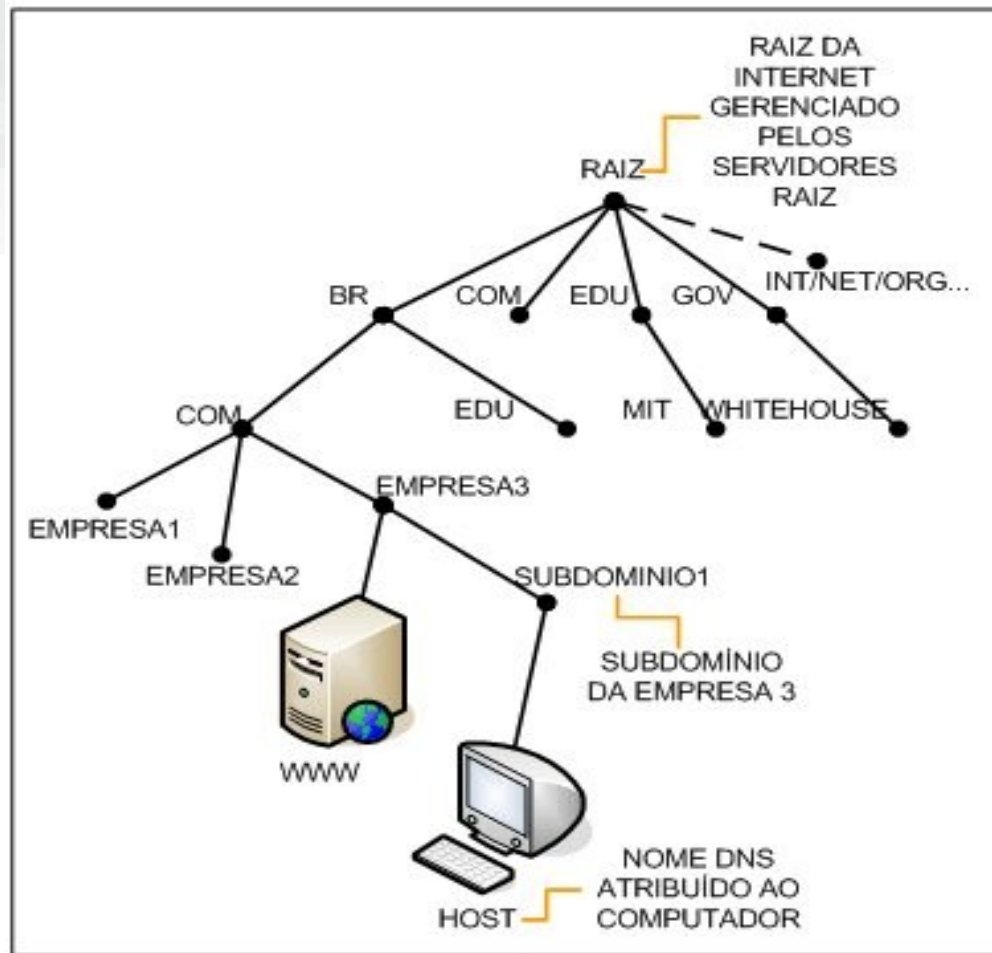
[www.terra.com.br](http://www.terra.com.br)



# Nomes e domínios



# Nomes e domínios





## Nomes e domínios

- No caso do terra, por exemplo, o domínio de primeiro nível é “br”.
- Sempre da direita para a esquerda.
- A segunda parte, é o domínio de segundo nível.
- A última parte (www) é o nome do host sob aquele domínio, e tudo que estiver entre o nome do host e o domínio de segundo nível são subdomínios de seu precedente na hierarquia.
- O conjunto completo de um nome DNS é chamado FQDN (Fully Qualified Domain Name).





## Domínios de primeiro nível

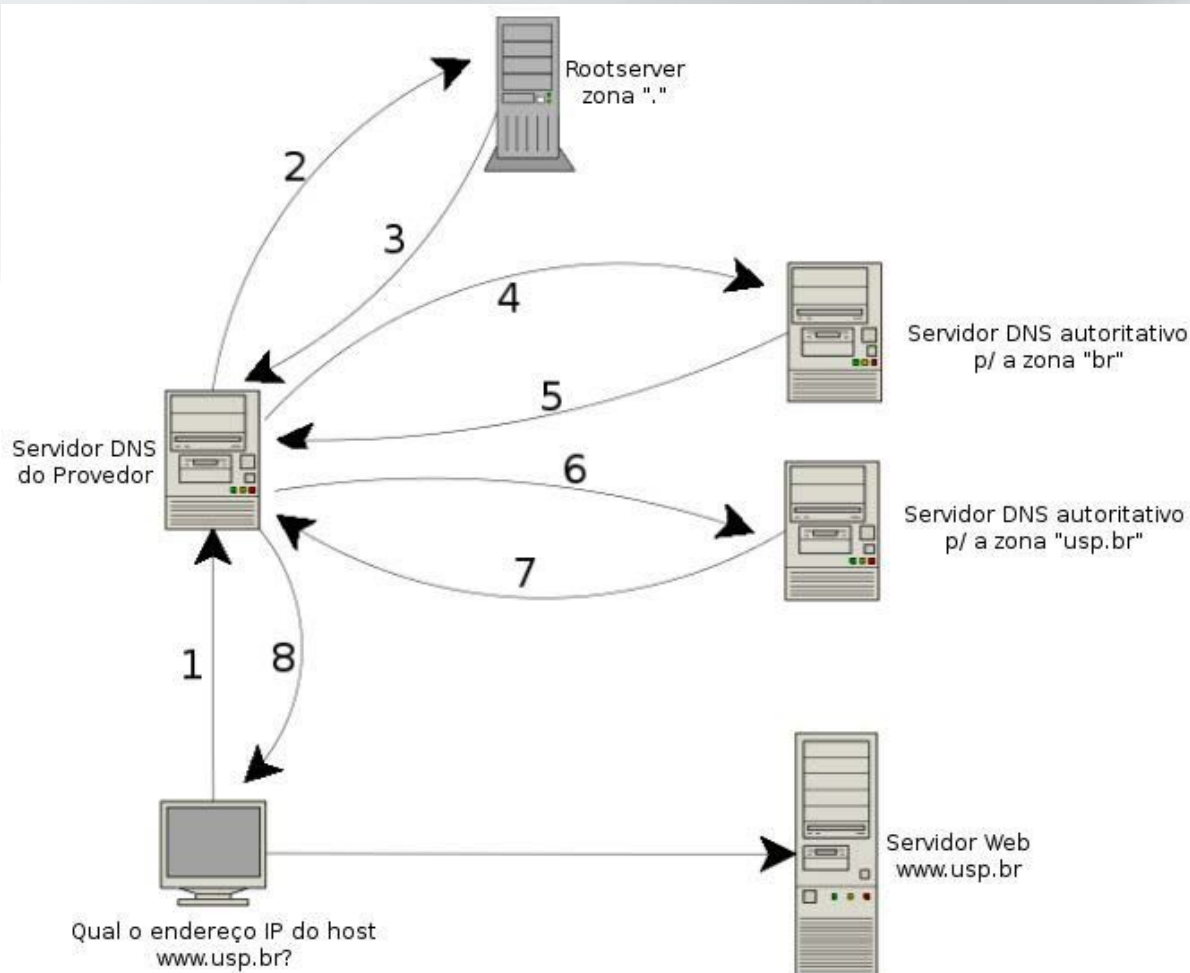
- Inicialmente a internet foi dividida em seis domínios, por tipo de organização:
- **com**: organizações comerciais
- **edu**: organizações de ensino
- **gov**: organizações governamentais
- **mil**: organizações militares
- **net**: organizações da rede
- **org**: entidades não-governamentais
- <http://www.norid.no/domenenavnbaser/domreg.html>



## Delegação

- A delegação de domínios é similar a hierarquia de uma empresa.
- O trabalho vai sendo mandado para o nível mais baixo.
- No Brasil, o órgão responsável pelo registro de domínios é a FAPESP, através do registro.br.

# Processo de resolução DNS





# Instalação do DNS no Linux

- No Linux, o software chamado “bind” provê a implementação de resolução de nomes DNS. O bind é amplamente utilizado em servidores Linux, Unix e BSD.
- `# aptitude install bind9`



## Instalação do DNS no Linux

- Após a instalação do pacote bind9, o daemon ***named*** será iniciado e o bind já estará em funcionamento
- ***# netstat -natup | grep :53***
- Os arquivos de configuração do servidor de nomes bind estão localizados sob o diretório ***/etc/bind*** e os arquivos de zonas a serem criados deverão ser colocados sob o diretório ***/var/cache/bind***, conforme o valor do parâmetro ***directory*** na seção ***options*** no arquivos de configuração principal do servidor de nomes bind, o arquivo ***/etc/bind/named.conf.options***





## Servidor de nomes somente para cache

- A idéia é somente aproveitar o cache de resolução de nomes do servidor.
- Não é necessário criar mapas de zonas e lidar com nenhum tipo adicional de configuração.
- Quando se instala o pacote bind9 no debian já é iniciado com um funcionamento adequado para atuar como um servidor de nomes somente para cache logo após sua instalação.



## Servidor de nomes autoritativo

- É um servidor de nomes que possui autoridade para um ou mais domínios e, sendo assim, responde as pesquisas de resoluções de nomes para hosts que fazem parte dos domínios em questão.
- Requer a configuração do bind, e adicionalmente a criação dos mapas de zonas para resolução comum e reversa para o domínio.
- Agora iremos configurar nosso servidor DNS.
- O domínio para qual desejamos que o bind responda pesquisas de resolução de nomes deverá ser cadastrado no arquivo de configuração do bind.



## Configurando DNS

- `# cd /etc/bind/`
- `# vim named.conf`

```
zone "andre.com.br" {  
    type master;  
    file "db.andre.com.br";  
    allow-transfer { 10.1.20.202; };  
    notify yes;  
};
```



# Configurando DNS

```
zone "20.1.10.in-addr.arpa" {  
    type master;  
    file "db.rev";  
};
```





## Servidor de nomes autoritativo

- O parâmetro **type** especifica o tipo de servidor de nomes será. **master** ou **slave**. **master** indica que será o servidor principal do domínio em questão e **slave** indica que nosso servidor de nomes será um servidor escravo, que somente receberá atualizações de mudanças feitas na zona.
- **file** indica o nome do arquivo onde a zona será definida.
- **allow-transfer** especifica os endereços Ips dos servidores de nomes slaves (escravos).
- **notify** especifica se o bind deverá enviar notificações de mudanças nos dados da zone para seus servidores de nomes escravos.





## Configurando DNS

- Iremos editar o `/etc/hosts`

```
# vim /etc/hosts
```

```
127.0.0.1    localhost
```

```
127.0.1.1    dns1
```

```
172.16.8.200 dns1.andre.com.br dns1
```

Salve o arquivo e teste:

```
# ping dns1
```

```
# ping dns1.andre.com.br
```



## Criação dos mapas de zonas

- Vamos criar os arquivos de zona
- `# cd /var/cache/bind`
- `# vim db.andre.com.br`



## Criação dos mapas de zonas

\$TTL 43200

\$ORIGIN andre.com.br.

```
@      IN      SOA ns1.andre.com.br. root.ns1.andre.com.br. (  
    2009090901 ;      serial  
    3600      ;      refresh  
    900       ;      retry  
    1209600   ;      expire  
    43200     ;      default_ttl  
    )
```

```
@      IN      MX 5   mx01
```

```
@      IN      NS     syslog-server
```

```
@      IN      A       172.16.8.200
```



## Criação dos mapas de zonas

```
@      IN      MX 5    mx01
@      IN      NS      ns1
@      IN      A       172.16.8.200
```

```
ns1     IN      A       172.16.8.200
mx01     IN      A       10.1.20.202
www      IN      CNAME   ns1
ftp      IN      CNAME   ns1
smtp     IN      CNAME   ns1
pop      IN      CNAME   ns1
```



## Criação dos mapas de zonas

- `# cd /var/cache/bind`
- `# vim db.andre.com.br`



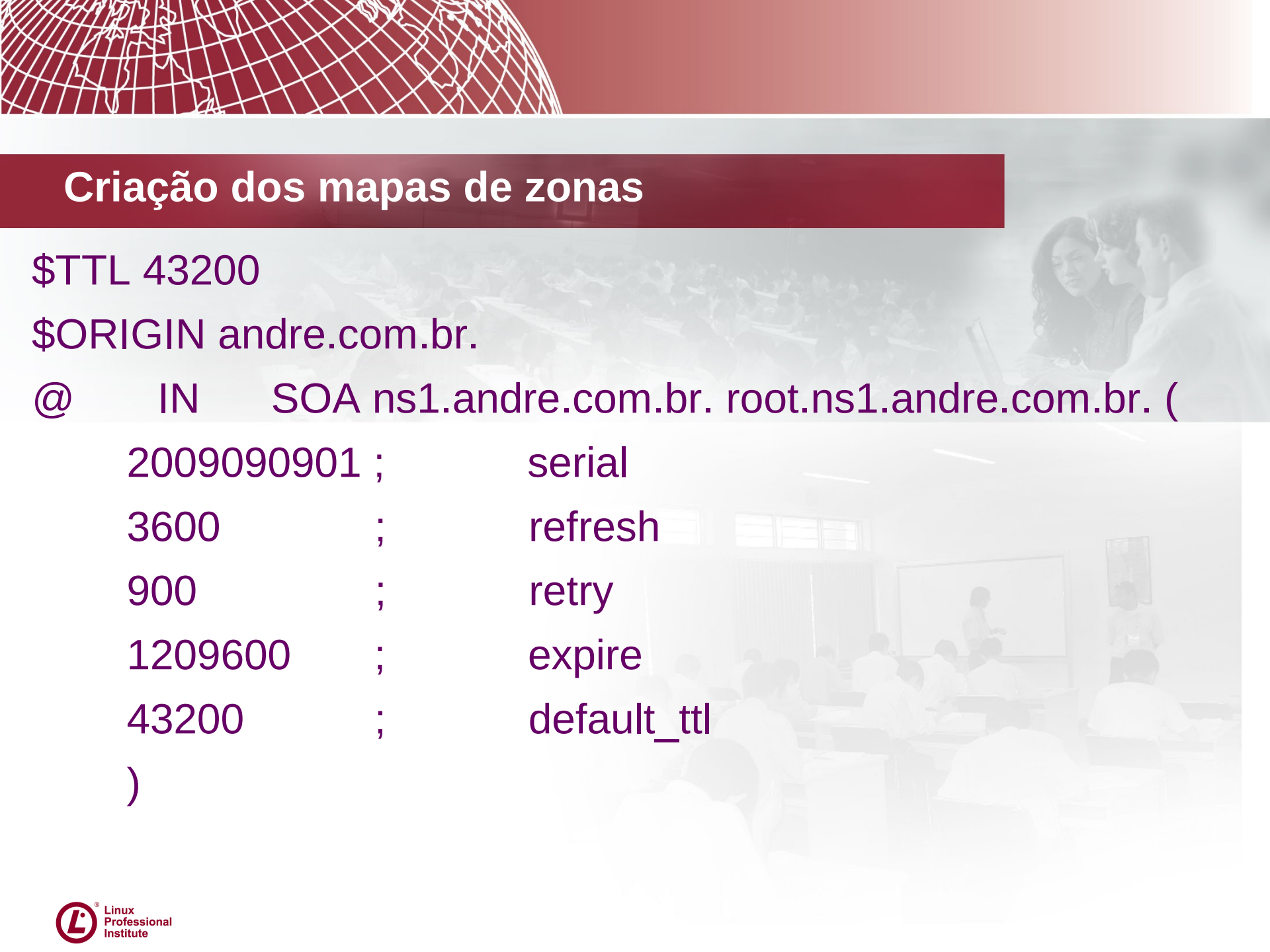


## Criação dos mapas de zonas

\$TTL 43200

\$ORIGIN andre.com.br.

```
@      IN      SOA ns1.andre.com.br. root.ns1.andre.com.br. (
    2009090901 ;      serial
    3600       ;      refresh
    900        ;      retry
    1209600    ;      expire
    43200      ;      default_ttl
)
```





## Criação dos mapas de zonas

@	IN	NS	ns1.andre.com.br.
201	IN	PTR	ns1.andre.com.br.
202	IN	PTR	mx01.andre.com.br.



## Criação dos mapas de zonas

- **\$TTL**: Cada registro em uma zona é conhecido também como um *RR*, ou *Registro de Recurso* (do inglês *Resource Record*). Cada *RR* pode possuir um "tempo de vida", definido como um registro inteiro de 32 bits representado em unidades de segundos. Um *TTL* define um limite de tempo que o registro deve ser mantido em cache.
- **SOA**: O registro SOA indica o início de uma zona com autoridade. Esse registro é composto de vários campos, como e-mail do administrador, servidor principal, tempo máximo de cache e outros.



## Criação dos mapas de zonas

- **mx** : define o servidor de e-mail responsável pelas mensagens do domínio `andre.com.br`
- **ns** : define que nossa zona terá um servidor de nome que poderá responder com autoridade.
- **A**: fornece o endereço IP de um dado nome
- **PTR**: fornece o nome de um dado IP (reverso)
- **CNAME**: usado para criar apelidos para nomes.





## Criação dos mapas de zonas

- **Serial** : O campo serial deve conter um número inteiro que deve ser incrementado a cada modificação feita na zona. Quando os servidores de nomes escravos checam a zona em busca de modificações, o campo *serial* é o campo checado. Se o mesmo possui um valor maior que o valor da cópia que possuem localmente, os servidores de nomes escravos entendem que a zona foi modificada e que uma nova transferência de zona é necessária.
- Um bom padrão para o serial costuma ser:

AAAAMMDDXX





## Criação dos mapas de zonas

- Esses campos controlam o período de tempo em que os servidores de nomes secundários irão buscar por informações atualizadas sobre a zona do servidor.
- **Refresh:** é tempo do ciclo de atualização.
- **Retry:** este é o período entre tentativas.
- **Expire:** tempo de vida útil de um banco de dados
- **Minimum TTL:** este é o tempo mínimo de vida que uma zona terá antes de ser descartada.



## Testando a configuração

- É muito importante realizar os testes nos arquivos de configuração antes de reiniciar os serviços do bind
- Primeiro iremos testar o arquivo de configuração principal:
  - `# named-checkconf /etc/bind/named.conf`
- Segundo testar os arquivos de configuração das zonas:
  - `# named-checkzone andre.com.br /var/cache/bind/db.andre.com.br`
  - `# named-checkzone andre.com.br /var/cache/bind/db.rev`



## Iniciando o bind

- Agora iremos editar o arquivo `/etc/resolv.conf`
- `# vim /etc/resolv.conf`  
search andre.com.br  
nameserver 172.16.8.200  
Nameserver 192.168.200.1
- `# /etc/init.d/bind9 restart`
- `# netstat -natup | grep :53`



## Iniciando o bind

- Agora iremos editar o arquivo `/etc/resolv.conf`
- `# vim /etc/resolv.conf`  
search andre.com.br  
nameserver 172.16.8.200  
Nameserver 192.168.200.1
- `# /etc/init.d/bind9 restart`
- `# netstat -natup | grep :53`



## Testando o bind

- # ping [www.andre.com.br](http://www.andre.com.br)
- # dig MX smaff.com.br
- # dig NS smaff.com.br
- # dig @ns1.smaff.com.br axfr smaff.com.br





## Testando o bind

- # nslookup
  - > set type=ns
  - > set all
  - >andre.com.br
- 
- # nslookup
  - > set type=mx
  - > set all
  - >andre.com.br



## Testando o bind

- # nslookup
- > set type=soa
- >andre.com.br

- # nslookup
- > set type=ptr
- > set all
- >172.16.8.200



## Testando o bind

- `# host -t any andre.com.br`
- `# host -t any uol.com.br`
- `# host -t mx andre.com.br`
- `# host -v -t soa uol.com.br`
- `# dig -t mx andre.com.br`
- `# dig -t ns andre.com.br`

# Segurança em Linux Firewall Proxy





## Conceitos básicos: Definindo a Segurança

- Privacidade
- Confiabilidade
- Integridade
- Disponibilidade
- Autenticação





# Tipos de Atacantes

- Script Kiddies (Lammers, etc)
- Hackers;
- Crackers;
- Carders;



# Tipos de Ataques

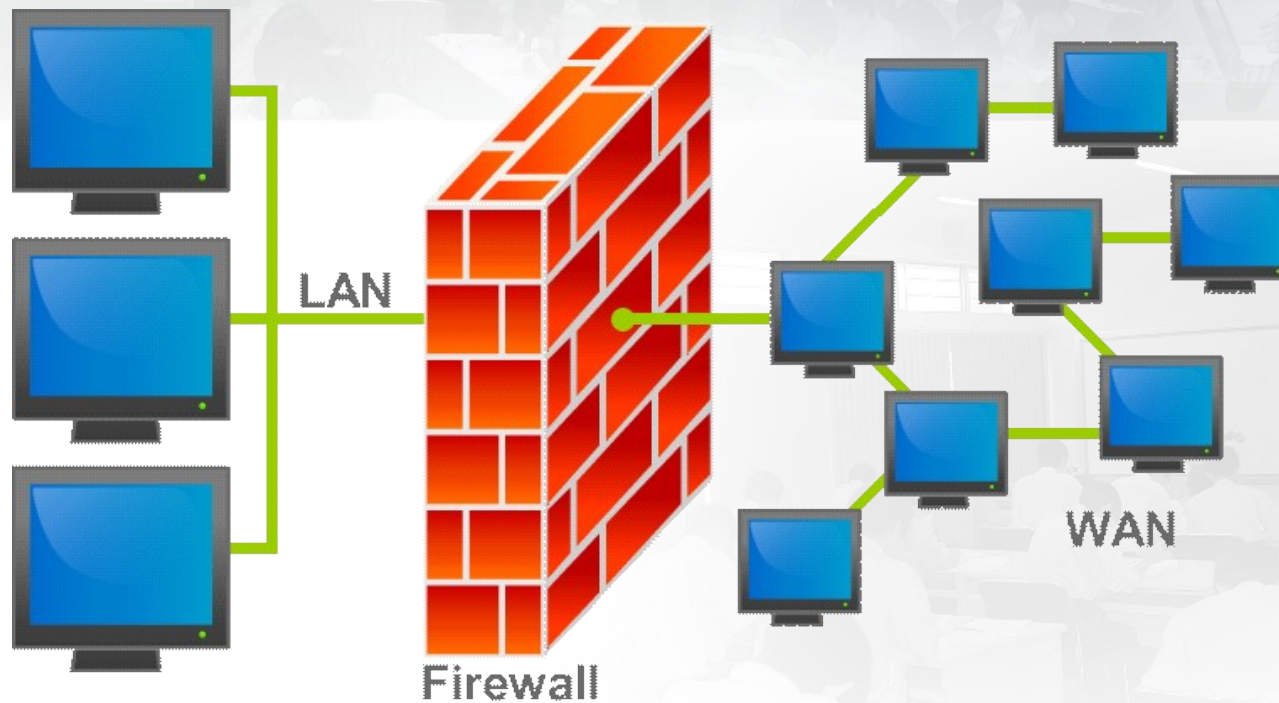
- Levantamento de informações
- Exploração
- Paralisação de serviços
- Worms / Vírus



## O que é um firewall

“Um firewall é um dispositivo de hardware, software ou híbrido que tem como principal objetivo proteger as informações e/ou filtrar o acesso as mesmas. Pode servir ainda como elemento de interligação entre duas redes distintas.”

# O que é um firewall





## Tipos de firewall

- **Filtros de Pacote;**
- **Filtros de Aplicação;**



## Tipos de firewall: Filtro de pacote

- Inspeção somente do cabeçalho dos pacotes

### Cabeçalho IP

Endereço de Origem  
Endereço de Destino,  
TTL,  
Checksum

### Cabeçalho TCP

Número de Sequência  
Porta de Origem,  
Porta de Destino,  
Checksum

### Camada de Aplicação

????????????????????  
????????????????????  
????????????????????  
????????????



## Tipos de firewall

“A filtragem de pacotes é um dos principais mecanismos, que mediante regras definidas pelo administrador em um firewall, permite ou não a passagem de datagramas IP em uma rede. Poderíamos filtrar pacotes para impedir o acesso a um serviço de Telnet , ou ainda a um chat ou até mesmo a uma Homepage.”

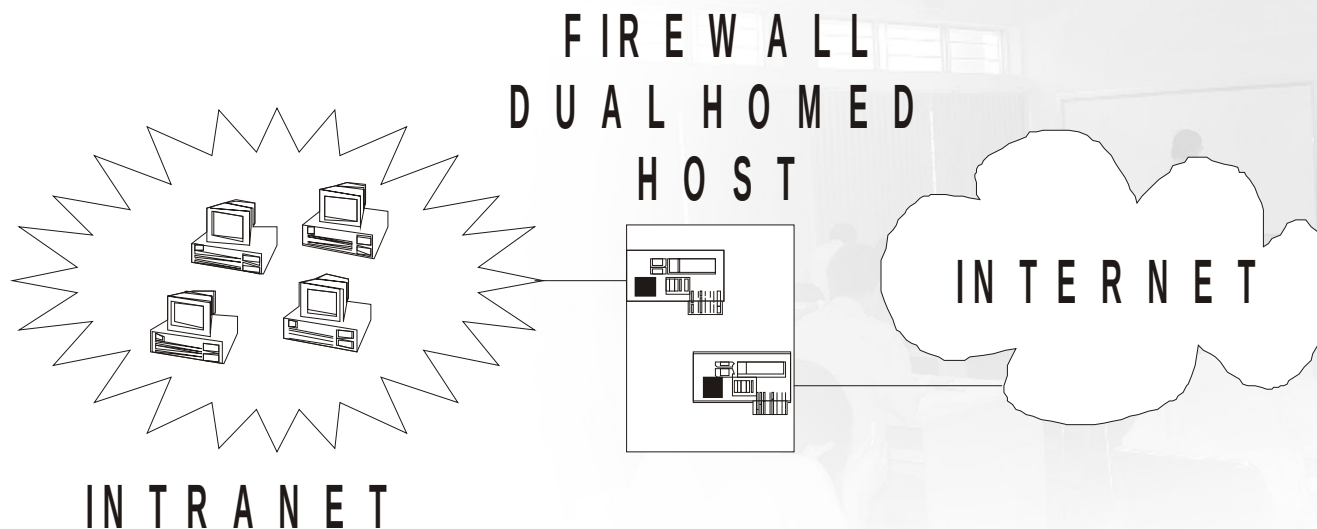


## Tipos de firewall

- **Endereço de origem** : de onde está vindo a requisição;
- **Endereço de destino** : para onde está sendo feita a requisição;
- **Tipo de Protocolo** : TCP, UDP e ICMP;
- **Porta** : numeração da porta TCP/UDP;
- **Tipos de Datagrama** : ICMP Echo Request, SYN/ACK, FIN, etc.

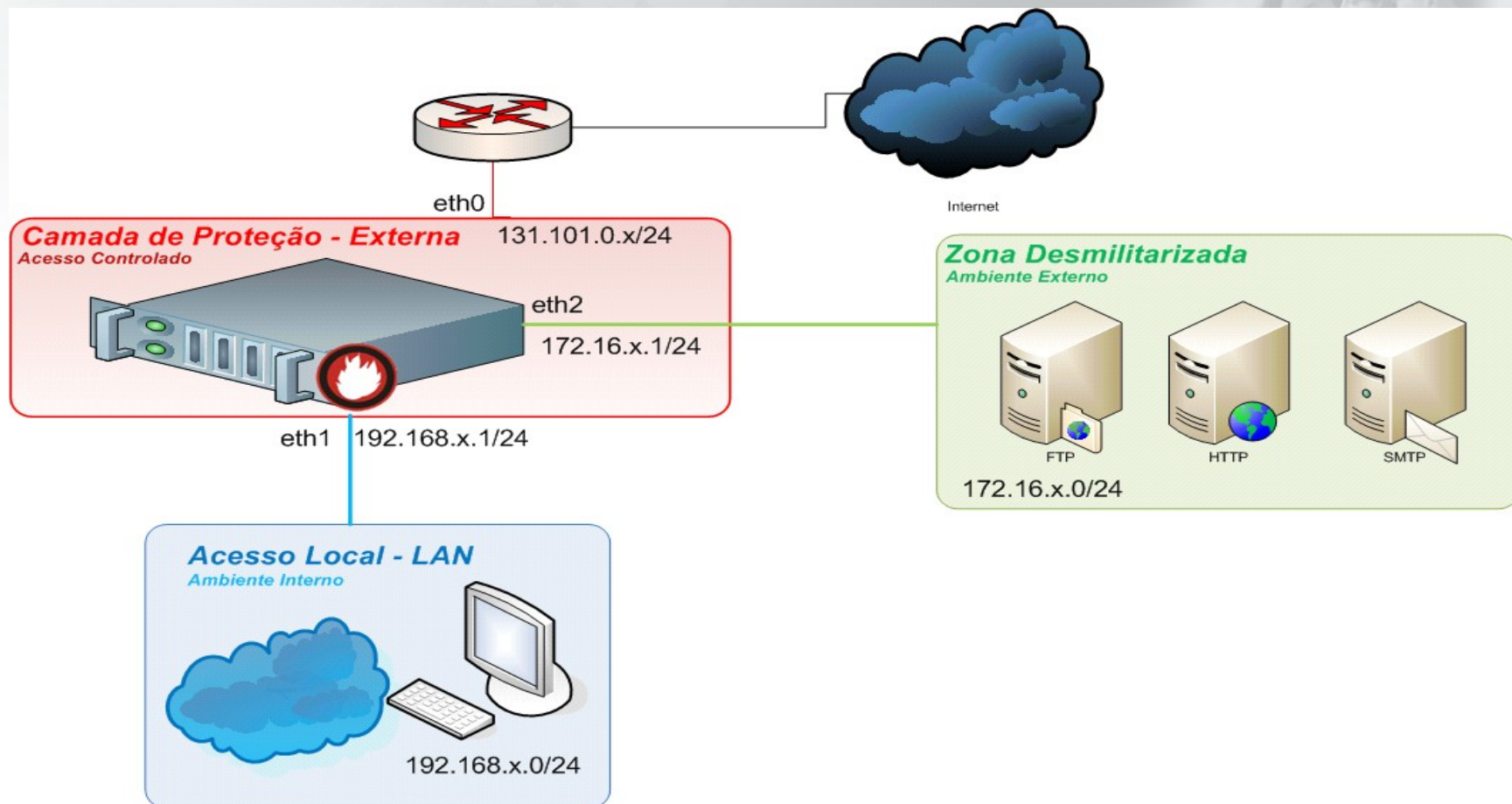
## Ligação entre Redes

- O modelo mais simples de firewall é conhecido como o dual homed system, ou seja um sistema que interliga duas redes distintas.





# DMZ





## Tipos de firewall: Filtro de aplicação. Layer 7

- Inspeção no cabeçalho dos pacotes e no campo de dados do pacote

### Cabeçalho IP

Endereço de Origem  
Endereço de Destino,  
TTL,  
Checksum

### Cabeçalho TCP

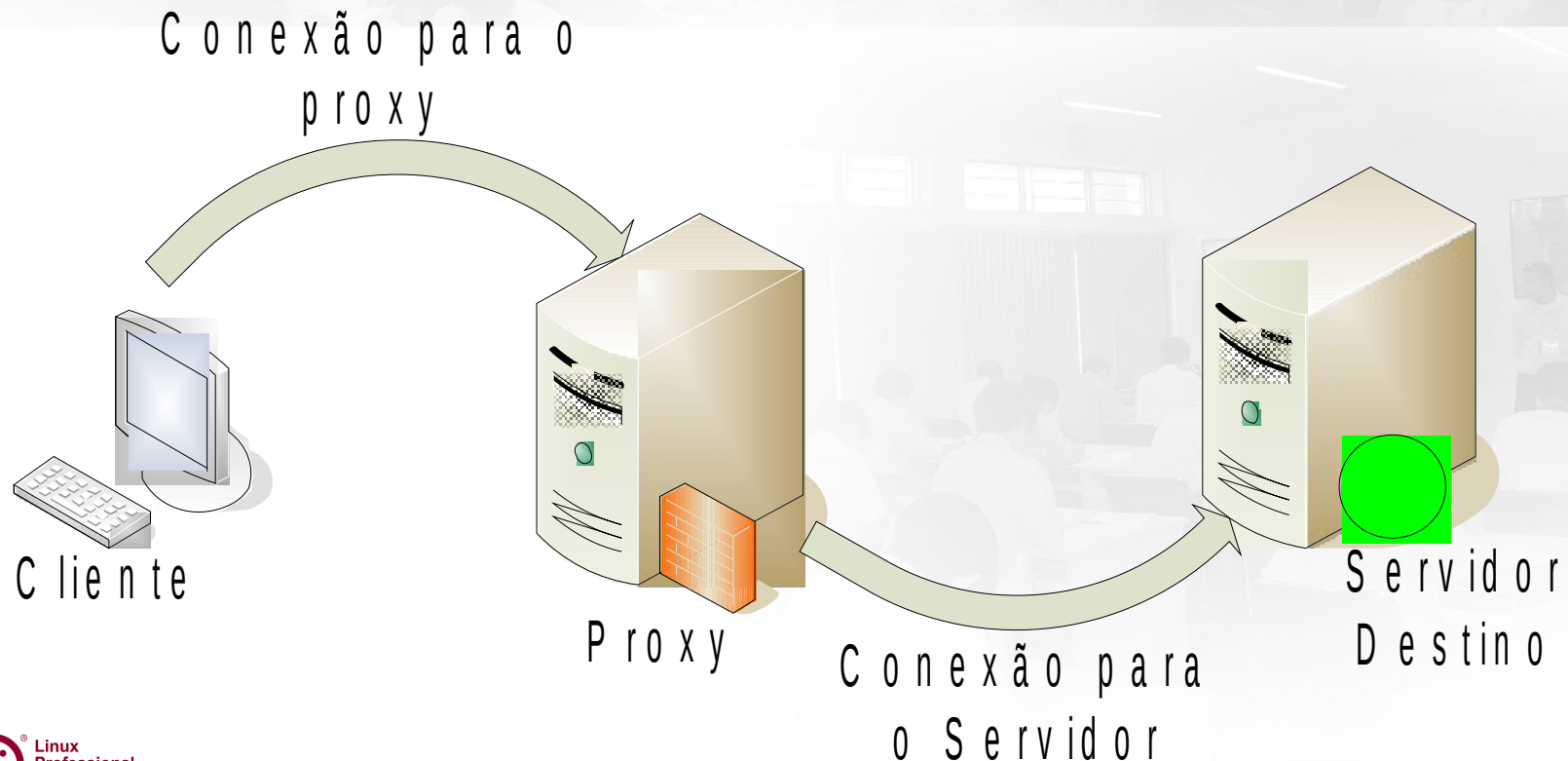
Número de Sequencia  
Porta de Origem,  
Porta de Destino,  
Checksum

### Camada de Aplicação

```
<html><head><meta http-equiv =  
"content-type" content="text/html;  
charset= UTF-8"> <title> MSNBC -  
MSNBC FrontPage </title> <link  
rel="stylesheet"
```

# Proxy

- “Permite executar a conexão ou não a serviços em uma rede de modo indireto, utilizando um cliente específico para esta conexão. Exemplo : Proxies e Navegadores”



# Proxy Transparente





# Softwares de Firewall Linux

- **Ipfwadm**
- **Ipchains**
- **Iptables**





## Ipfwadm

- “O IP Firewall Administration, ou simplesmente ipfwadm foi a ferramenta padrão para construção de regras de firewall, para o kernel anterior a versão 2.2.0. Para muitos o ipfwadm era extremamente complexo e causava certa confusão ao administradores de sistema. Nos dias de hoje ainda existem firewalls baseados neste serviço..”





## ipchains

- “O ipchains foi a solução, ou melhor a atualização, feita para o Kernel 2.2 do ipfwadm. A idéia do ipchains foi ter o poder do ipfwadm, mas com uma simplicidade no que diz respeito a criação de regras. A idéia do ipchains além de prover esta facilidade é criar uma compatibilidade com o ipfwadm através do utilitário ipfwadm-wrapper.”



# iptables

- “A nova geração de ferramentas de firewall para o Kernel 2.4 do Linux. Além de possuir a facilidade do ipchains, a idéia do criador Paul Russel foi implentar uma série de facilidades como NAT e filtragem de pacotes mais flexível que o Ipchains.”

# Operacionalização Do Firewall





# Regras

- As regras de um firewall são a forma de aplicação de filtragem dos pacotes em seu funcionamento. Por exemplo, para impedir que os usuários façam FTP, é necessário criar uma regra que impeça este tipo de operação, ou melhor bloqueie o envio e recebimento de pacotes na porta 21 e 20 utilizadas pelo FTP.



## Tipos de Regras

- **DENY** - bloqueia o acesso, impedindo a passagem de um pacote e não manda nenhum tipo de aviso ao endereço que requisitou o acesso. Ideal para um firewall seguro.
- **ACCEPT** - aceita a passagem de um pacote.
- **REJECT** - bloqueia o acesso, impedindo a passagem de um pacote manda um aviso ao endereço que requisitou o acesso.





## Observações

- As regras são como filtros aplicados ao iptables para que o mesmo implemente o que chamamos de filtro de pacote de acordo com o endereço IP/porta de origem/destino, interface de origem/destino, etc.
- As regras são armazenadas dentro dos chamdos chains e processadas na ordem que são inseridas. Estas mesmas regras são armazenadas no kernel, o que significa que quando o sistema é reinicializado as mesmas são perdidas.



## Sintaxe

A sintaxe de uma regra é a seguinte :  
iptables comando parametros extensões

Algo similar a isto na prática :

```
iptables -A INPUT -p tcp -s 10.0.0.1 -j DROP .
```



## Tabelas

- **Tabela filter** – Considerada a tabela padrão, contém 3 chains básicos :
  - **INPUT** - Consultado para pacotes que chegam na própria máquina
  - **OUTPUT** - Consultado para pacotes que saem da própria máquina.
  - **FORWARD** - Consultado para pacotes que são redirecionados para outra interface de rede ou outra estação. Utilizada em mascaramento.



## Tabelas

- **Tabela nat** - Usada para passagem de pacotes que pode gerar outra conexão. Um exemplo clássico é o mascaramento (masquerading), nat, port forwarding e proxy transparente são alguns. Possui 3 chains básicas :
- **PREROUTING** - Consultado quando os pacotes precisam ser redirecionados logo que chegam. Por exemplo um pacote smtp que vai ser direcionado para um endereço interno da rede. ( chain ideal para realização do chamado Destination NAT (DNAT)





## Tabela NAT

- **OUTPUT** - Consultado quando os pacotes gerados localmente precisam ser redirecionados antes de serem roteados. Este chain somente é utilizada para conexões que se originam de IPs de interfaces de rede locais.
- **POSTROUTING** - Consultado quando os pacotes precisam ser modificados após o tratamento de roteamento. É o chain utilizado para realização de SNAT e mascaramento(IP Masquerading).





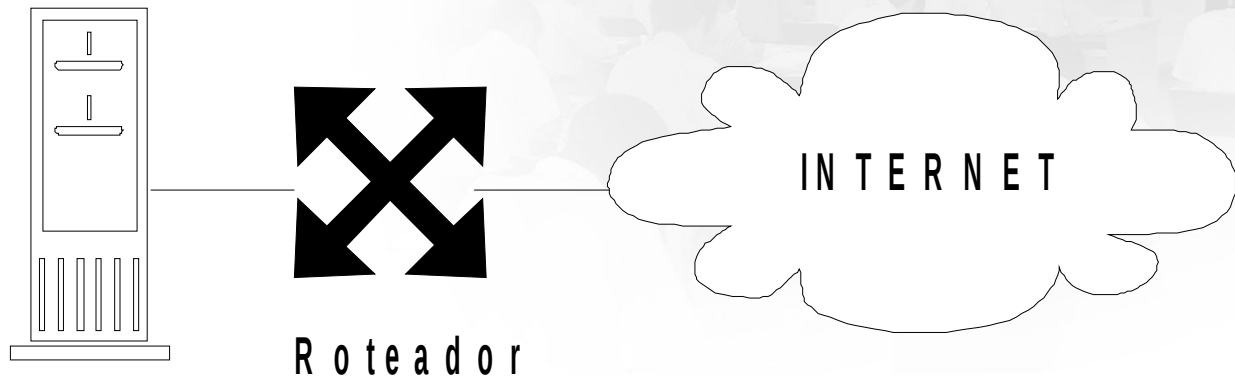
## Tabela mangle

- **Tabela mangle** - Utilizada para alterações especiais de pacotes como por exemplo modificar o tipo de serviço (TOS) de um pacote. Ideal para produzir informações falsas para scanners Possui 2 chains padrões:
  - **PREROUTING** - Consultado quando os pacotes precisam ser redirecionados logo que chegam.
  - **OUTPUT** - Consultado quando os pacotes gerados localmente precisam ser redirecionados antes de serem roteados.

## Nat e mascaramento

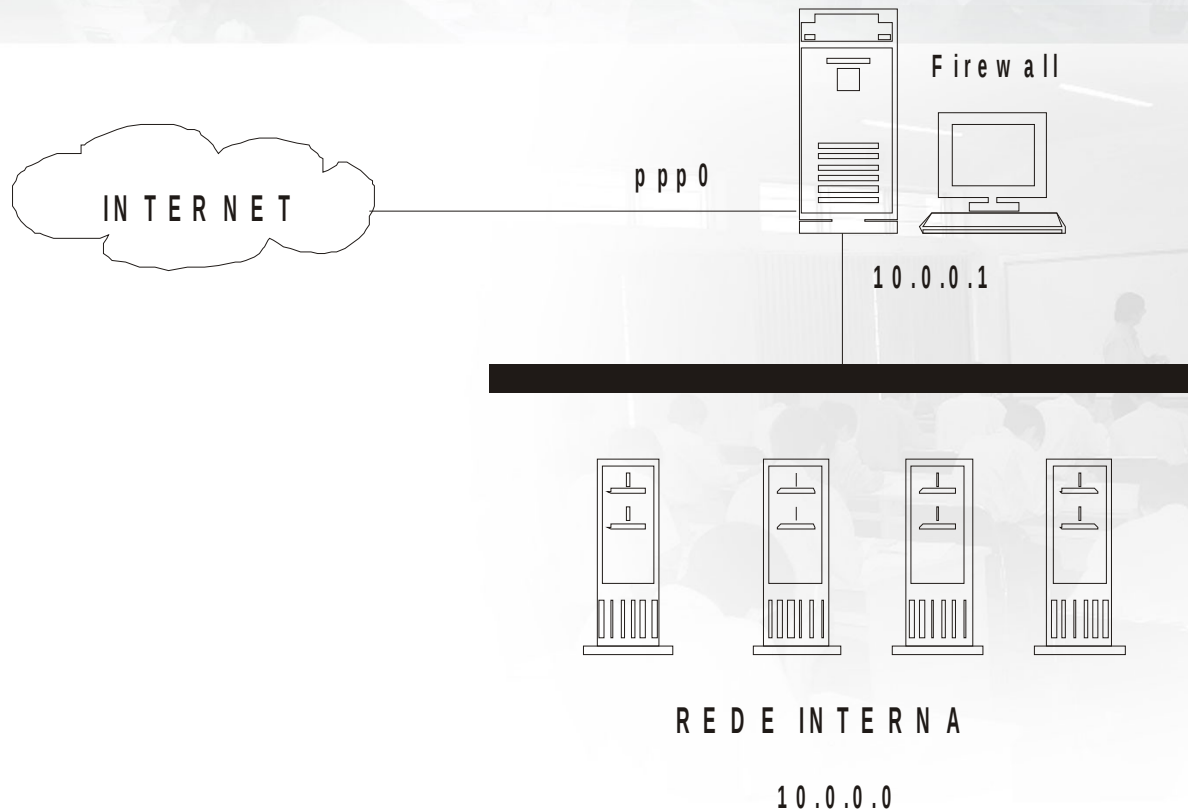
- O NAT (Network Address Translator) é a técnica da qual roteadores, traduzem um endereço falso dentro de uma rede uma rede classe A com endereços 10.0.0.X, para um endereço válido para a Internet ou para uma outra rede. No esquema abaixo, vemos isto de forma mais clara :

R e d e I n t e r n a 1 0 . 0 . 0 . X - R e d e E x t e r n a 2 0 0 . 0 . 0 . X



# NAT

Um exemplo de mascaramento para um usuário caseiro de ADSL, ou modem.





# Tabelas

- **filter:** INPUT, FORWARD e OUTPUT
- **nat:** PREROUTING, POSTROUTING e OUTPUT
- **mangle:** PREROUTING e OUTPUT



## Ações

- **ACCEPT:** o pacote é aceito por essa cadeia e segue em frente.
- **DROP:** o pacote é rejeitado pela cadeia, não existe uma mensagem ICMP.
- **REJECT:** semelhante ao DROP, porém retorna uma mensagem ICMP.
- **REDIRECT:** altera o endereço IP de destino do pacote e será usado unicamente na tabela nat.





## Comandos do iptables

- -A: anexa regras ao final de uma cadeia.
- -D: apaga uma ou mais regras da cadeia especificada.
- -I: insere uma ou mais regras no começo da cadeia
- -L: lista todas as regras em uma cadeia
- -F: remove todas as regras de uma cadeia
- -Z: restaura os contadores de datagramas e de bytes em todas as regras das cadeias especificadas para zero, ou para todas as cadeias se nenhuma for especificada.
- -P: define a política padrão para uma cadeia dentro de uma política especificada.



## Comandos do iptables

- -p protocolo: define o protocolo ao qual a regra se aplica
- -s address: define a origem do pacote ao qual a regra se aplica.
- -d address: define o destino do pacote ao qual a regra se aplica.
- -j alvo: define um alvo para o pacote caso ele se encaixe nesta regra.
- -i interface: define o nome da interface por onde o datagrama foi recebido.
- -o interface: define o nome da interface por onde o datagrama será transmitido.



## Extensões do iptables

- **Extensão TCP: usada com -p tcp**
- --sport: especifica a porta que a origem do datagrama usa.
- --dport: especifica a porta que o destino do datagrama usa.
- --syn: especifica que a regra deve encontrar somente datagramas com o bit SYN ligado e os bits ACK e FIN desligados
- **Extensão UDP: usada com -p udp**
- --sport: idem tcp.
- --dport: idem tcp.



# SEGURANÇA EM SERVIDORES

- Hardening
- Segurança no terminal
- Gerenciamento de privilégios
- PAM
- Protegendo o serviço SSH
- Segurança no boot loader
- NESSUS





## Hardening

- **Programas desnecessários**
- ABNT NBR ISO/IEC 17799:2005 diz, no item 11.5.4, que devemos remover todo utilitário desnecessário do sistema após uma checagem.
- `# mkdir /root/auditoria`
- `# dpkg -l | awk '{print $2,$3}' | sed '1,7d' > /root/auditoria/pacotes`
- `# aptitude purge wget`





## Arquivos com permissão de Suid Bit

- Por recomendação da norma ABNT NBR ISO/IEC 17799:2005, no item 11.6.1, o acesso à informação e às funções dos sistemas de aplicações por usuário e pessoal de suporte deve ser restrito, de acordo com o definido da política de controle de acesso.
- `# find / -perm 4000 > /root/auditoria/lista.suid`



## Segurança no Sistema de Arquivos

- No que diz respeito à segurança em sistemas de arquivos, a norma ABNT NBR ISO/IEC 17799:2005 recomenda no item 10.4 e item 10.4.1 que devemos proteger a integridade do software e da informação e ter um controle contra códigos maliciosos.
- NOSUID
- # adduser teste
- # cp /bin/sh /home/teste/
- # chmod 4755 /home/teste/sh
- # cd /home/teste/
- # su - teste
- \$ /sh



# Segurança no Sistema de Arquivos

- `# mount -o remount,rw,nosuid /home`
- `# mount`
- `# su – teste`
- `$ ./sh`
- `$ id`



# Segurança no Sistema de Arquivos

- NO EXEC
- `# mount -o remount,rw,noexec /home`
- `# mount`
- `# su – teste`
- `$ ./sh`



## Segurança no terminal

- No que diz respeito a segurança no terminal, a norma ABNT NBR ISO/IEC 17799:2005, no item 11.5.5 e item 11.5.6 que devemos ter controle sobre os acessos no sistema a fim de evitar acessos não-autorizados e para não fornecer informações desnecessárias.
- Desabilitando o uso do CTRL+ALT+DEL
- Limitando o uso dos terminais texto
- Bloqueando o terminal com a variável TMOUT
- Bloqueando o terminal com o programa vlock



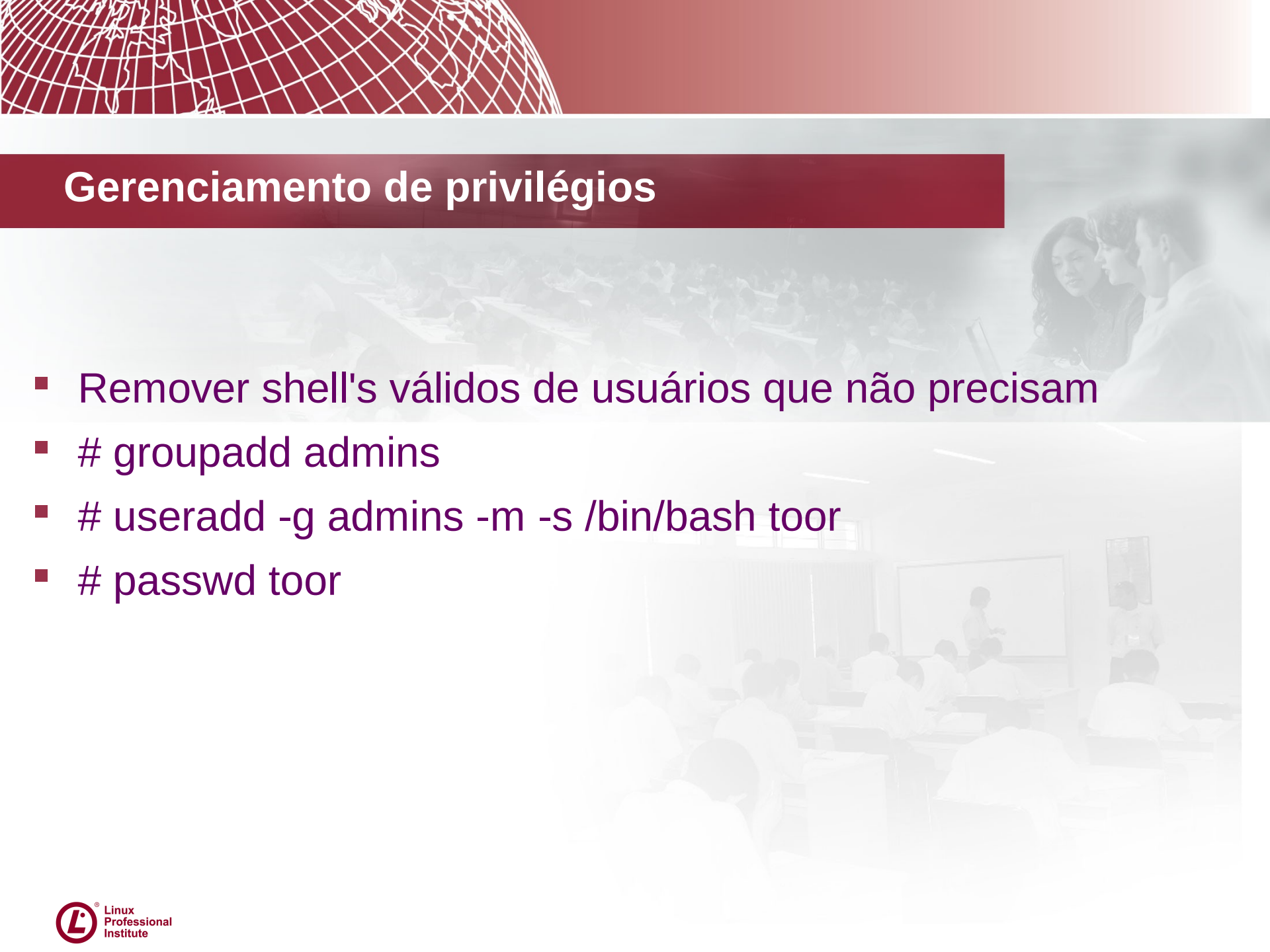


## Gerenciamento de privilégios

- Por recomendação da norma ABNT NBR ISO/IEC 17799:2005, no item 11.2.2 a concessão e o uso de privilégios devem ser restritos e controlados.
- Bloqueando o login de root nos terminais texto
- `# vim /etc/securetty`
- Determinar datas de expiração para contas de usuários
- `# chage -l usuario`
- `# chage -M 30 -W 5 -I 2 usuario`



## Gerenciamento de privilégios

- Remover shell's válidos de usuários que não precisam
  - # groupadd admins
  - # useradd -g admins -m -s /bin/bash toor
  - # passwd toor
- 



# PAM

- PAM significa Pluggable Authentication Modules.
- Conjunto de bibliotecas compartilhadas que permitem ao administrador do sistema local definir como as aplicações autenticam os usuários, sem a necessidade de modificar e recompilar programas.
- Diretório de /etc/pam.d/
- <tipo de módulo> <palavra de controle> <módulo>  
<parâmetros>



## Tipos de módulos

- **auth:** faz autenticação dos usuários, pedindo e verificando senhas e liberando o acesso a estes.
- **account:** garante a autenticação, verificando se a conta do usuário em questão não expirou e se este pode acessar o sistema nos horários predeterminados.
- **password:** é usado para criação de senhas e a sua configuração.
- **session:** é usado para gerenciar a sessão de um usuário que foi autenticado no sistema.





## Palavras de controle

- **required:** este módulo deve ser verificado para permitir autenticação.
- **requisite:** este módulo deve ser verificado para que a autenticação seja bem sucedida.
- **sufficient:** A falha na verificação de um módulo não implica a falha da autenticação como um todo. Mas se a verificação de um módulo marcado como sufficient for bem sucedida e nenhum módulo required tiver falhado, então os módulos restantes do mesmo tipo de módulo não são verificados e o usuário é autenticado.





## Palavras de controle

- **optional:** A falha na verificação do módulo não implica a falha da autenticação como um todo. A única ocasião em que um módulo marcado como optional é necessária para uma autenticação bem sucedida é quando a verificação de nenhum outro módulo dessa classe falhou ou funcionou. Neste caso, um módulo marcado como optional determina a autenticação para um módulo desse tipo.
- `pam_securetty` e `pam_nologin`
- `# ls -l /lib/security`



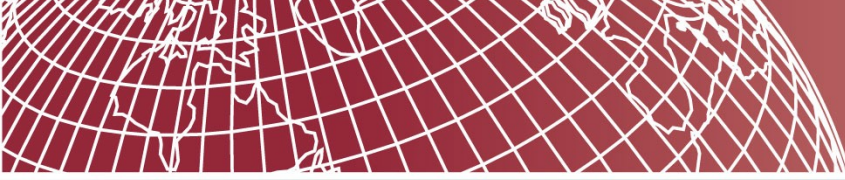
# PAM

- Editar o arquivo `/etc/pam.d/login`
- `account requisite pam_time.so`
- Editar o arquivo `/etc/security/time.conf`
- `login;*;root;!A10000-2400`
- `services;tty;users;times`



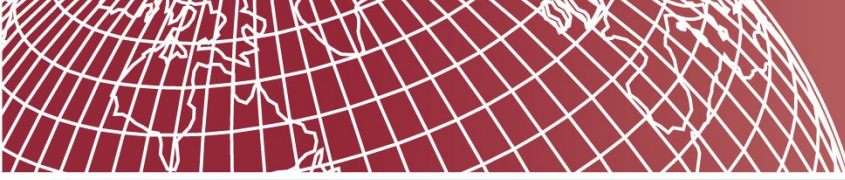
# Regras





# Regras





# Regras

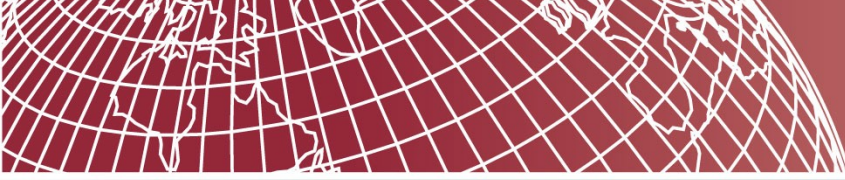






# Regras





# Regras





# Regras





# Regras

