

Camada de Rede

4

Conceitos Básicos
Roteamento

Reinaldo Gomes
reinaldo@dsc.ufog.edu.br

1

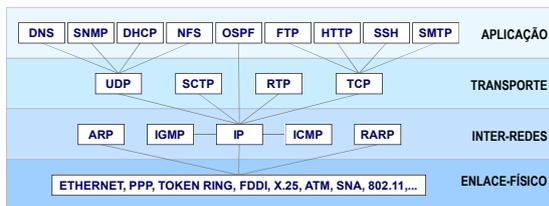
Internet e TCP/IP

- Internet
 - Conjunto de redes de escala mundial, ligadas pelo protocolo IP
- TCP/IP
 - Família de protocolos de comunicação
 - Serviços e acesso independente de tecnologia
 - Permite a interconexão de redes físicas diferentes
 - Interconexão realizada por roteadores
- Protocolo IP
 - Não orientado a conexão, roteamento melhor esforço
 - Não confiável, sem controle de fluxo e de erros → simples
 - Roteamento baseado no endereço da rede de destino

2

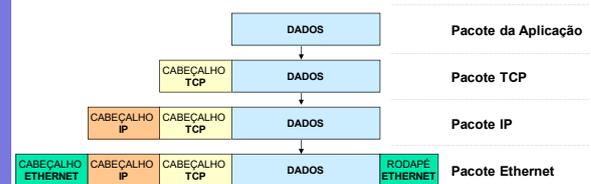
Arquitetura TCP/IP

- TCP/IP = família de protocolos



3

Encapsulamento



4

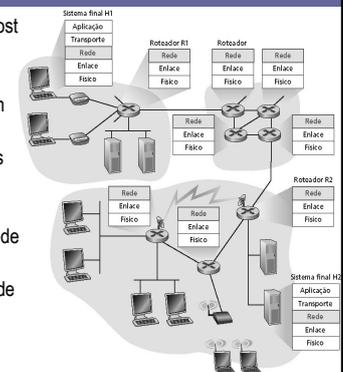
Camada de Rede: Objetivos

- Transferência de pacotes da origem para o destino
- Vários saltos (*hops*) intermediários no caminho
- Elementos de rede: roteadores ou computadores
- Principal função: roteamento (encaminhamento)
- Outras funções
 - Controle de Congestionamento
 - Negociação de QoS
 - Interconexão de redes

5

A camada de Rede

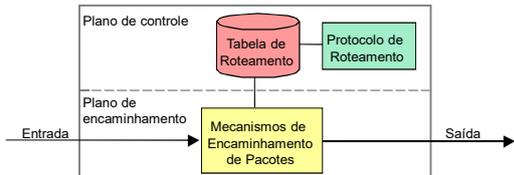
- Transporta segmentos do host transmissor para o receptor
- No lado transmissor, encapsula os segmentos em datagramas
- No lado receptor, entrega os segmentos à camada de transporte
- Protocolos da camada de rede em cada host roteador
- Roteador examina campos de cabeçalho em todos os datagramas IP que passam por ele



6

Roteamento e Encaminhamento

- **Roteamento:** estabelecimento dos melhores caminhos (rotas)
- **Encaminhamento:** processo de despachar cada pacote ao seu destino ou sistema intermediário



7

Questões de projeto

- Serviços oferecidos à camada de Transporte
 - Devem ser independentes da tecnologia da sub-rede
 - Proteção contra tipo, quantidade e topologia das sub-redes
 - Endereços devem ter plano de numeração único
- Tipos de Serviço
 - Orientado a conexão
 - Sem conexão

8

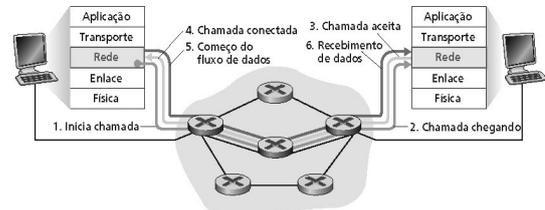
Organização Interna

- Circuitos Virtuais (CVs)
 - Analogia aos circuitos físicos da rede telefônica
 - Rede complexa e segura
 - Geralmente orientada a conexões (conexões na camada de rede são geralmente chamadas de circuitos virtuais)
- Datagramas
 - Analogia com o serviço de "telegramas"
 - Rede simples e não confiável
 - Geralmente é não orientada a conexões (mas pode ser)

9

Circuitos Virtuais

- Usado para estabelecer, manter e encerrar conexões
- Usados em ATM, frame-relay e X-25
- Não é usado na Internet atualmente



10

Circuitos Virtuais

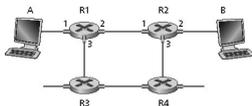


Tabela de comutação no roteador R1:

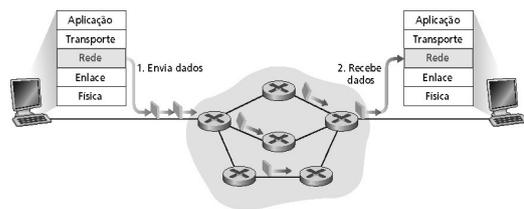
Interface de entrada	CV # de entrada	Interface de saída	CV # de saída
1	12	2	22
2	63	1	18
3	7	2	17
1	97	3	87
...

Roteadores mantêm informações de estado de conexão

11

Rede de Datagrama

- Não é estabelecida conexão na camada de rede
- Roteadores: sem estado sobre conexões fim-a-fim
 - O conceito "conexão" não existe na camada de rede
- Pacotes são encaminhados para o endereço do host de destino
 - Pacotes para o mesmo destino podem seguir diferentes rotas



12

Circuito Virtual X Datagrama

Questão	Subrede de Datagrama	Subrede de Circuito Virtual (VC)
Configuração de Circuito	Não necessária	Requerida
Endereçamento	Cada pacote contém endereços da fonte e do destino	Cada pacote contém um número de CV
Informação de Estado	Subrede não mantém informação de estado	Cada CV requer espaço na tabela da subrede
Roteamento	Cada pacote é roteado independentemente	Rota escolhida quando CV é configurado; todos os pacotes seguem esta rota
Efeito de falhas do roteador	Nenhuma, exceto a perda de pacotes durante a falha	Todos os CVs que passam pelo roteador com falha são Terminados
Controle de congestionamento	Difícil	Fácil se memória suficiente tiver sido alocada a priori para cada CV

13

Roteamento

- Algoritmo de roteamento
 - Parte da camada de rede responsável pela decisão sobre a linha de saída para a transmissão de um pacote
 - Pode ser implementado por um protocolo de roteamento, ou executado de maneira estática
- Protocolo de roteamento
 - Software utilizado pelos roteadores para que eles estabeleçam *tabelas de roteamento* consistentes
 - Qualquer protocolo de roteamento deve comunicar informação da topologia da rede para todos os outros roteadores, para tomar decisões de roteamento

14

Roteamento

- Datagramas
 - Decisão de roteamento deve ser tomada a cada pacote
- Circuitos Virtuais
 - Decisão de roteamento é tomada somente no estabelecimento da conexão
 - Também chamado de *roteamento por sessão*

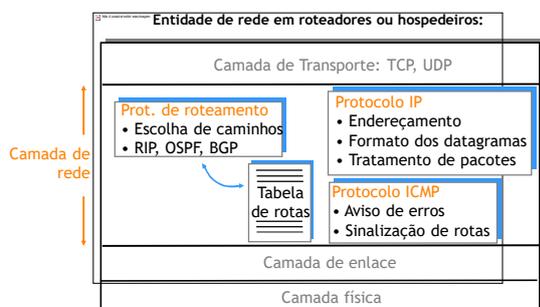
15

Classes de Algoritmos de Roteamento

- Não adaptativo (estático)
 - Decisão do roteamento não é baseada em estimativas de tráfego atual e da topologia
- Adaptativo (dinâmico)
 - Muda decisões de roteamento para refletir mudanças na topologia, bem como, no tráfego
 - Métricas
 - Custo, caminho, carga, tamanho da fila

16

A camada de Rede



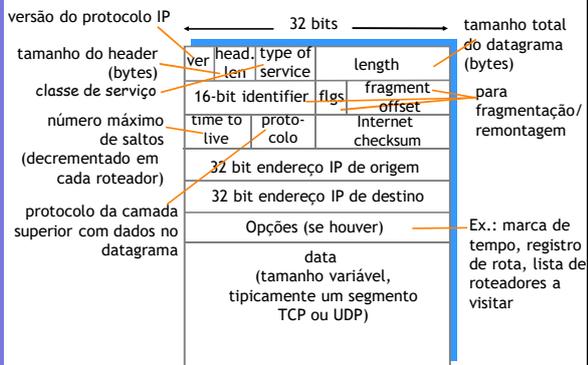
17

Cabeçalho IP

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Version		IHL		TOS				Total length																							
Identification				Flags				Fragment offset																							
TTL				Protocol				Header checksum																							
Source IP address																Destination IP address															
Options and padding																															

18

Cabeçalho IP



19

Campos do cabeçalho IP

Campo	Bits	Descrição
Version	4	Identifica a versão do protocolo IP (IPv4, IPv6)
Header Length	4	Tamanho do cabeçalho (deve ser múltiplo de 32 bits). Se for 20 bytes, deve indicar 5 ($5 \times 32 = 160$ bits, ou 20 bytes).
Type-of-Service Flags	8	Para permitir priorização de pacotes, dependendo da aplicação.
Total Packet Length	16	Tamanho total do pacote em bytes, incluindo cabeçalho e dados
Fragment Identifier	16	Identifica partes de um pacote fragmentado, para ajudar remontagem
Fragmentation Flags	3	Informações e controle sobre fragmentação (ex: instrução para roteadores não fragmentarem o pacote)
Fragmentation Offset	13	Posição dos dados deste pacote (fragmentado) em relação ao pacote original (não fragmentado). Múltiplo de 8 bytes.

20

Campos do cabeçalho IP

Campo	Bits	Descrição
Time-to-Live	8	Indica o número restante de "hops" que o pacote pode seguir antes de ser considerado inválido e descartado.
Protocol Identifier	8	Identifica o protocolo da camada superior, contido no corpo do pacote.
Header Checksum	16	Checksum do cabeçalho IP
Source IP Address	32	Endereço IP do remetente do pacote
Destination IP Address	32	Endereço IP do destinatário do pacote
Options (optional)	varia	Opções adicionais para type-of-service, Source Routing, Timestamp etc. Raramente usado.
Padding (if required)	varia	Se um pacote não for múltiplo de 32 bits, informações nulas são adicionadas para completar o tamanho.
Data	varia	Os dados que o IP deverá carregar (TCP, UDP, Fragmento, outros protocolos, dados puros ICMP, ...)

21

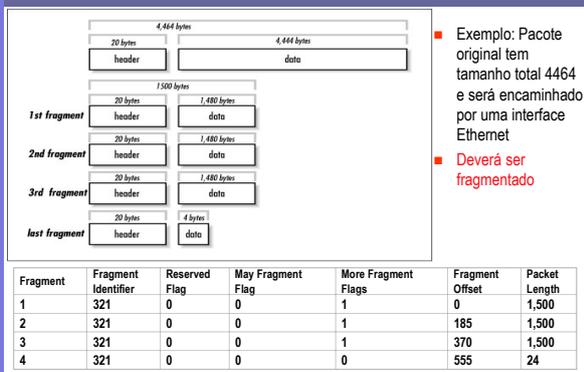
Fragmentação e remontagem

- MTU - *Maximum Transmit Unit* é o tamanho máximo que um pacote pode ter
- MTU é padronizado de acordo com a interface física
- Cada roteador deve fragmentar o pacote antes de encaminhá-lo para uma interface, no caso do tamanho original ser maior que o MTU

Topology	MTU (in bytes)	Defined By
Hyperchannel	65,535	RFC 1374
16 MB/s Token Ring	17,914	IBM
802.4 Token Bus	8,166	RFC 1042
4 MB/s Token Ring	4,464	RFC 1042
FDDI	4,352	RFC 1390
DIX Ethernet	1,500	RFC 894
Point-to-Point Protocol (PPP)	1,500	RFC 1548
802.3 Ethernet	1,492	RFC 1042
Serial-Line IP (SLIP)	1,006	RFC 1055
X.25 & ISDN	576	RFC 1356
ARCnet	508	RFC 1051

22

Fragmentação e remontagem



23

Fragmentação e remontagem

Fragment	Fragment Identifier	Reserved Flag	May Fragment Flag	More Fragment Flags	Fragment Offset	Packet Length
1	321	0	0	1	0	1,500
2	321	0	0	1	185	1,500
3	321	0	0	1	370	1,500
4	321	0	0	0	555	24

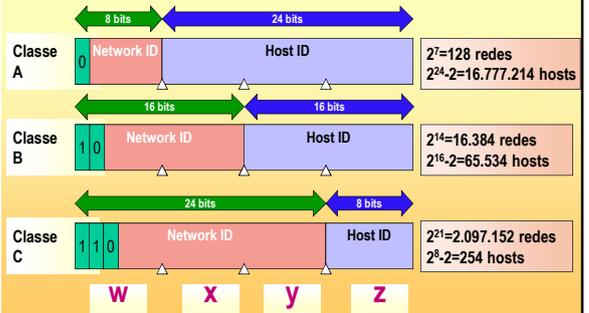
- Exemplo: Pacote original tem tamanho total 4464 e será encaminhado por uma interface Ethernet
- Deverá ser fragmentado
- Cada fragmento pertence a um mesmo pacote original (mesmo valor no campo Fragmentation Identifier)
- O primeiro bit do campo "Flags" deve ser 0 (reservado)
- Fragmentos não podem novamente ser fragmentados, logo "May Fragment" deve ser 0.
- O flag "More Fragments" indica se existe mais fragmentos adiante
- 1o. fragmento → "Fragment Offset" = 0, tam=1500 (20+1480) bytes
- 2o. fragmento → "Fragment Offset" = 185 (1480/8), tam=1500
- 3o. fragmento → "Fragment Offset" = 370 ((1480×2)/8), tam=1500
- 4o. fragmento → "Fragment Offset" = 555 ((1480×3)/8), tam=24 (20+4)

24

Endereçamento IP

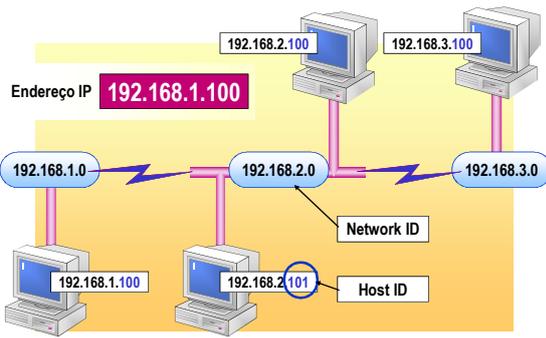
25

Classes de Endereços IP



26

Endereços IP



27

Endereços especiais

- Subredes privadas – Não podem ser publicados na Internet
 - 10.x.x.x (classe A)
 - 172.16.x.x - 172.31.x.x (classe B)
 - 192.168.x.x (classe C)
- Endereços especiais
 - 0.0.0.0 este host
 - 0.0.0.124 host 124 nesta rede
 - 255.255.255.255 todos os hosts desta rede
 - N.N.N.255 todos os hosts da rede N.N.N
 - 127.X.X.X Loopback
- Alguns endereços inválidos para hosts
 - 10.1.0.0 IP do host não pode ser 0
 - 10.1.0.255 IP do host não pode ser 255
 - 10.123.255.4 Subrede não pode ter valor 255
 - 0.12.16.89 Parte do endereço não pode ter valor 0
 - 255.9.56.45 Parte do endereço não pode ter valor 255
 - 10.34.255.1 Parte do endereço não pode ter valor 255

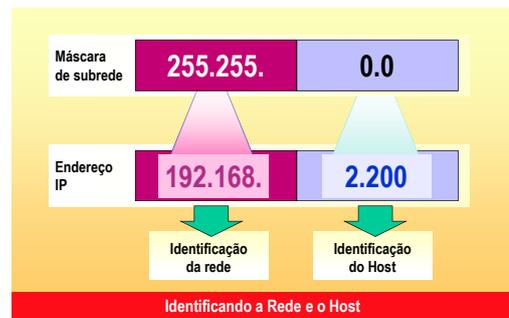
28

Máscara de subrede

- A máscara serve para indicar qual parte do endereço IP identifica o endereço de rede e qual parte identifica o endereço do host
 - os bits 1 indicam a parte do endereço da rede
 - os bits 0 indicam a parte do endereço do host
- Máscaras default:
 - Classe A - 255.0.0.0 11111111.00000000.00000000.00000000
 - Classe B - 255.255.0.0 11111111.11111111.00000000.00000000
 - Classe C - 255.255.255.0 11111111.11111111.11111111.00000000
- Executa-se um AND lógico entre os bits da máscara e endereço IP e obtém o Network Address
- No endereço de rede todos os bits do host são 0
- No endereço de broadcast, todos os bits do host são 1
- Exemplo:
 - 10001100.10110011.11110000.11001000 140.179.240.200 Endereço IP
 - 11111111.11111111.00000000.00000000 255.255.0.0 Máscara classe B
 - 10001100.10110011.00000000.00000000 140.179.0.0 Network Address
 - 10001100.10110011.11111111.11111111 140.179.255.255 Network Broadcast

29

Máscaras de Subrede



30

Máscaras de Subrede

Endereço IP	10.	50.100.200
Máscara de subrede	255.	0.0.0
Identificação da rede	10.	0.0.0

Classe A

31

Máscaras de Subrede

Endereço IP	10.50.	100.200
Máscara de subrede	255.255.	0.0
Identificação da rede	10.50.	0.0

Classe B

32

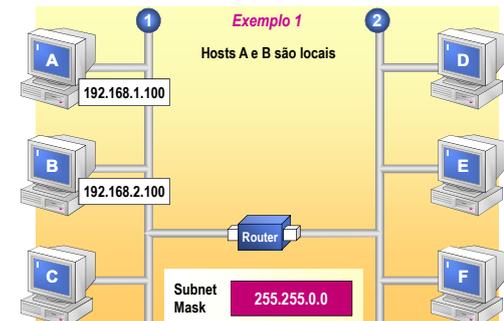
Máscaras de Subrede

Endereço IP	10.50.100.	200
Máscara de subrede	255.255.255.	0
Identificação da rede	10.50.100.	0

Classe C

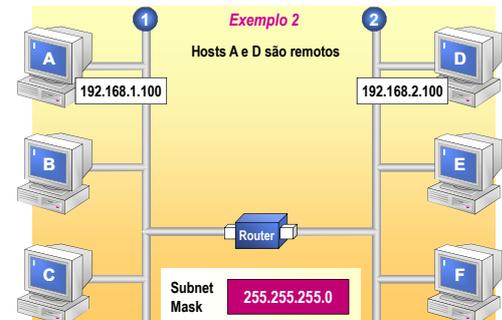
33

Determinando hosts locais e remotos



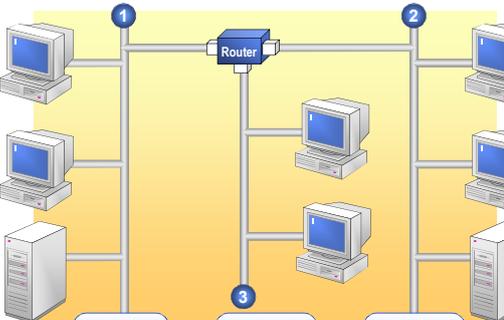
34

Determinando hosts locais e remotos

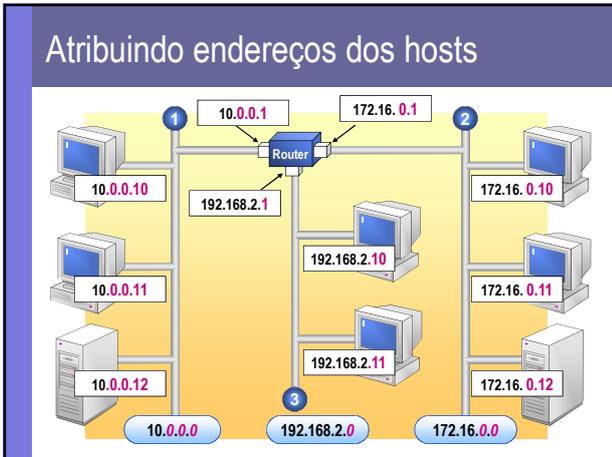


35

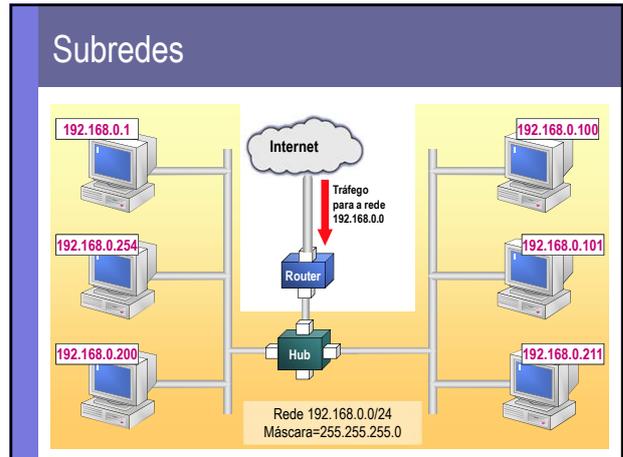
Atribuindo endereços das redes



36



37



38

Dividindo um Classe C em 2 subredes

- Exemplo: Rede 192.168.0.0
- Com máscara de subrede 255.255.255.0
 - Máscara = 11111111.11111111.11111111.00000000
 - 1 subrede, notação 192.168.0.0/24

Network	Hosts		Broadcast Address
	from	to	
192.168.0.0	192.168.0.1	192.168.0.254	192.168.0.255

- Rede 192.168.0.0 com máscara de subrede 255.255.255.128
 - Máscara = 11111111.11111111.11111111.10000000
 - 2 subredes, notação 192.168.0.0/25

Network	Hosts		Broadcast Address
	from	to	
192.168.0.0	192.168.0.1	192.168.0.126	192.168.0.127
192.168.0.128	192.168.0.129	192.168.0.254	192.168.0.255

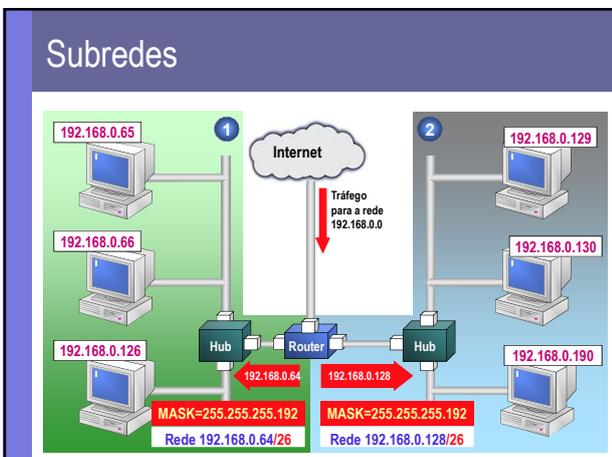
39

Dividindo um Classe C em 4 subredes

- Rede 192.168.0.0 com máscara de subrede 255.255.255.192
 - Máscara = 11111111.11111111.11111111.11000000
 - 4 subredes, notação 192.168.0.0/26

Network	Hosts		Broadcast Address
	from	to	
192.168.0.0	192.168.0.1	192.168.0.62	192.168.0.63
192.168.0.64	192.168.0.65	192.168.0.126	192.168.0.127
192.168.0.128	192.168.0.129	192.168.0.190	192.168.0.191
192.168.0.192	192.168.0.193	192.168.0.254	192.168.0.255

40



41

Endereçamento IP atualmente: CIDR

- CIDR: Classless InterDomain Routing (pronuncia-se como a palavra cider)
- A porção de endereço de rede tem tamanho arbitrário
- Formato do endereço: a.B.C.D/x, em que x é o número de bits na parte de rede do endereço

\leftarrow parte de rede \rightarrow parte de hospedeiro
 11001000 00010111 00010000 00000000
 200.23.16.0/23

42

CIDR: endereços reservados

CIDR	Bloco de Endereços	Descrição	Referência
0.0.0.0/8		Rede corrente (só funciona como endereço de origem)	RFC 1700
10.0.0.0/8		Rede Privada	RFC 1918
14.0.0.0/8		Rede Pública	RFC 1700
39.0.0.0/8		Reservado	RFC 1797
127.0.0.0/8		Localhost	RFC 3330
128.0.0.0/16		Reservado (IANA)	RFC 3330
169.254.0.0/16		Zeroconf	RFC 3827
172.16.0.0/12		Rede Privada	RFC 1918
191.255.0.0/16		Reservado (IANA)	RFC 3330
192.0.0.0/24			
192.0.2.0/24		Documentação	RFC 3330
192.88.99.0/24		IPv6 para IPv4	RFC 3068
192.168.0.0/16		Rede Privada	RFC 1918
198.18.0.0/15		Teste de benchmark de redes	RFC 2544
223.255.255.0/24		Reservado	RFC 3330
224.0.0.0/4		Multicasts (antiga rede Classe D)	RFC 3171
240.0.0.0/4		Reservado (antiga rede Classe E)	RFC 1700
255.255.255.255		Broadcast	

43

Endereçamento IP: CIDR

- CIDR: Classless InterDomain Routing
 - Inicialmente somente grupos de endereços Classe C foram utilizados
 - Como o mesmo procedimento já foi também aplicado às antigas classes A e B, pode-se dizer que de fato o endereçamento em classes está descaracterizado e completamente substituído pelo CIDR (obs.: vários endereços de antigas classes A e B foram progressivamente realocados)



44

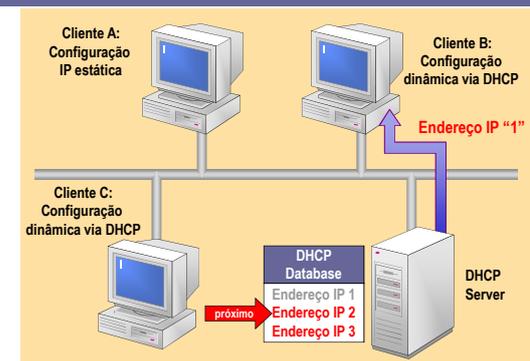
Como obter um endereço IP

- Como um sistema final pode obter um endereço IP?
 - Configuração estática
 - DHCP: Dynamic Host Configuration Protocol: obtém dinamicamente endereços IP de um servidor
 - "plug-and-play"
- Como uma rede obtém a parte de sub-rede do endereço IP?
 - Obtém a porção alocada no espaço de endereço do seu provedor ISP (ICANN: internet corporation for assigned names and numbers)
 - Aloca endereços, gerencia DNS, atribui nomes de domínios, ...

bloco do ISP	11001000	00010111	00010000	00000000	200.23.16.0/20
Organização 0	11001000	00010111	00010000	00000000	200.23.16.0/23
Organização 1	11001000	00010111	00010010	00000000	200.23.18.0/23
Organização 2	11001000	00010111	00010100	00000000	200.23.20.0/23
...

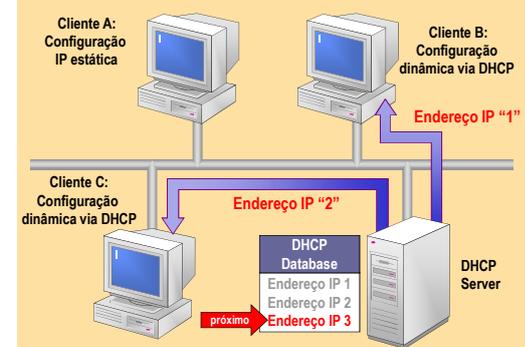
45

Como funciona o DHCP



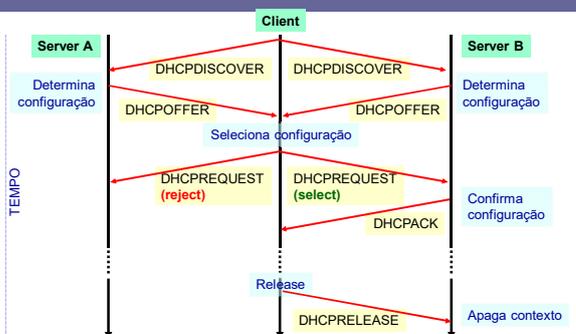
46

Como funciona o DHCP



47

DHCP – o que acontece na rede



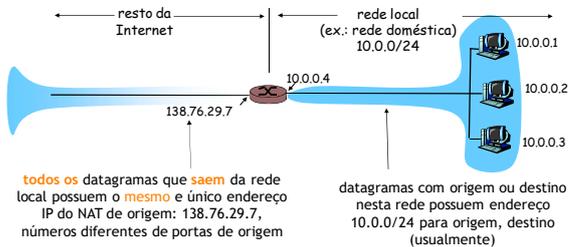
48

DHCP: Alguns Problemas

- **Segurança:**
 - Mensagens não são autenticadas
 - Alguém pode forjar um DHCP server ou um cliente
 - Cliente não pode confiar no servidor e vice-versa
- **Configuração:**
 - Para redes com mais de um servidor
 - Servidores na rede não podem trocar informações
 - Não existe um protocolo server-server
 - Devem ter espaços de endereçamento disjuntos para evitar distribuição de IP duplicados
 - Servidores são configurados manualmente

49

NAT: Network Address Translation



50

NAT: Network Address Translation

- **Motivação:** redes locais podem utilizar apenas um endereço IP:
 - Não é preciso alocar uma gama de endereços do ISP, apenas um endereço IP é usado para todos os dispositivos
 - Podem-se alterar os endereços dos dispositivos na rede local sem precisar notificar o mundo exterior
 - Pode-se mudar de ISP sem alterar os endereços dos dispositivos na rede local
 - Dispositivos da rede local não são explicitamente endereçáveis ou visíveis pelo mundo exterior (um adicional de segurança).

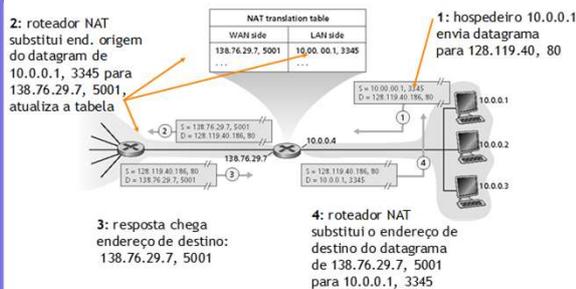
51

NAT: Network Address Translation

- **Implementação:** o roteador NAT deve:
 - Tratar datagramas que saem: substituir (endereço IP de origem, porta #) de cada datagrama para (endereço IP do NAT, nova porta #)
 - ... clientes/servidores remotos responderão usando (endereço IP do NAT, nova porta #) como endereço de destino
 - Lembrar (na tabela de tradução do NAT) cada (endereço IP de origem, porta #) para o par de tradução (endereço IP do NAT, nova porta #).
 - Tratar datagramas que chegam: substituir (endereço IP do NAT, nova porta #) nos campos de destino de cada datagrama pelos correspondentes (endereço IP de origem, porta #) armazenados da tabela NAT

52

NAT: Network Address Translation



53

NAT: Network Address Translation

- **Campo número de porta com 16 bits:**
 - Possibilidade de 65536 conexões simultâneas com um único endereço de LAN!!!
- **NAT é controverso:**
 - Roteadores deveriam processar somente até a camada de rede (Layer 3): Violação do argumento fim-a-fim
 - A possibilidade de NAT deve ser levada em conta pelos desenvolvedores de aplicações: ex.: aplicações P2P
 - A escassez de endereços deveria ser resolvida pelo IPv6

54

IPv6

55

Cabeçalho IPv6

- **Motivação inicial:** o espaço de endereços de 32 bits estava próximo de ser completamente alocado.
- Motivação adicional:
 - Melhorar o formato do cabeçalho para permitir maior velocidade de processamento e de transmissão
 - Mudanças no cabeçalho para incorporar mecanismos de controle de serviço (i.e., *Quality-of-Service*, QoS)
- **Formato do datagrama IPv6:**
 - Cabeçalho fixo de 40 Bytes
 - Não é permitida fragmentação (a fragmentação e remontagem tomam muito tempo, retirando essa funcionalidade dos roteadores acelera o repasse de datagramas IP)

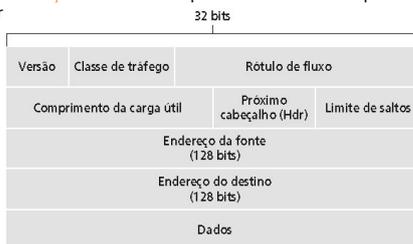
56

Cabeçalho IPv6

Classe de tráfego: permitir definir prioridades diferenciadas para vários fluxos de informação

Rótulo de Fluxo: identifica datagramas do mesmo “fluxo” (conceito de “fluxo” não é bem definido).

Próximo Cabeçalho: identifica o protocolo da camada superior ou um header

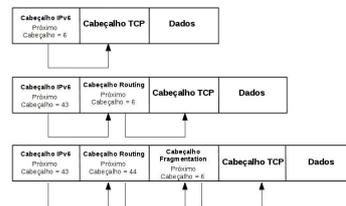


57

Cabeçalho IPv6

Podemos definir cabeçalhos de extensão

- Hop-by-Hop Options
- Destination Options (routing)
- Routing
- Fragmentation
- Authentication Header
- Encapsulating Security Payload
- Destination Options
- Mobility



58

Outras mudanças do IPv4

- **Endereços** de 128 bits (i.e., 16 Bytes)
- **Checksum:** **removido** inteiramente para reduzir o tempo de processamento em cada salto
- **Options:** são permitidas, mas são alocadas em cabeçalhos suplementares, indicados pelo campo “Next header”
- **ICMPv6:** nova versão de ICMP
 - Tipos de mensagens adicionais, ex.: “Packet Too Big”
 - Funções de gerenciamento de grupos multicast

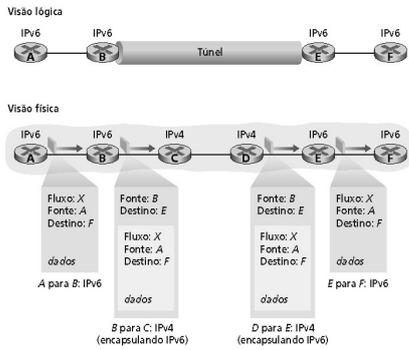
59

Transição do IPv4 para IPv6

- Nem todos os roteadores poderão ser atualizados simultaneamente
- **Não haverá um dia da vacinação!!!!**
- Como a rede irá operar com roteadores mistos de IPv4 e IPv6?
- **Tunelamento:** IPv6 transportado dentro de pacotes IPv4 entre roteadores IPv4

60

Tunelamento

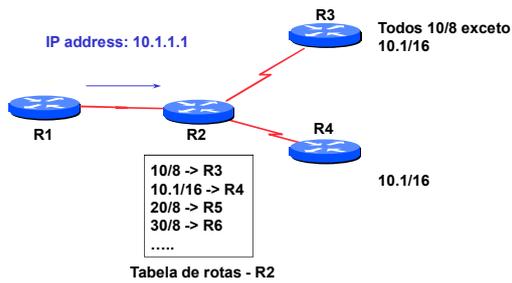


61

Encaminhamento

62

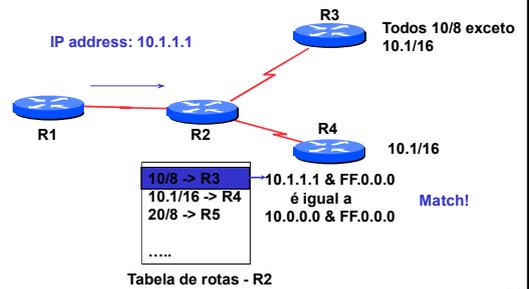
Encaminhamento de Pacotes



63

63

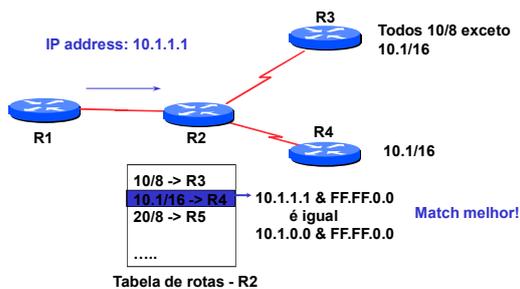
Encaminhamento de Pacotes



64

64

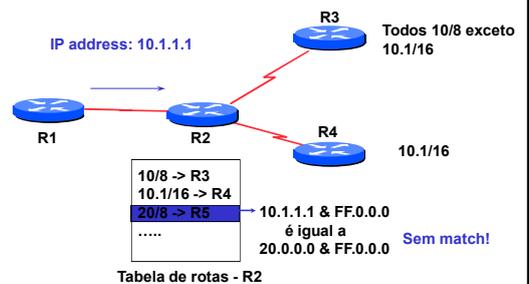
Encaminhamento de Pacotes



65

65

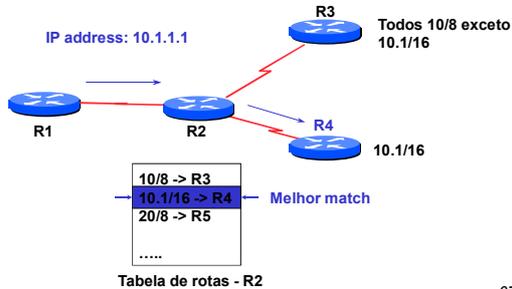
Encaminhamento de Pacotes



66

66

Encaminhamento de Pacotes



67

Roteamento

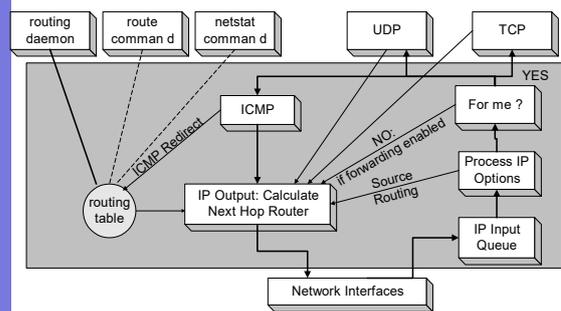
68

Roteamento

- **Relembrando:** O roteamento IP consiste em duas fases:
 1. Como repassar um pacote de uma interface de entrada para uma interface de saída de um roteador (encaminhamento de pacotes)?
 2. Como localizar e configurar uma rota?
- A primeira parte já foi discutida
- Há duas abordagens para o cálculo da tabela de roteamento:
 - Roteamento Estático
 - Roteamento Dinâmico: Rotas calculadas pelos protocolos de roteamento

69

Roteamento IP



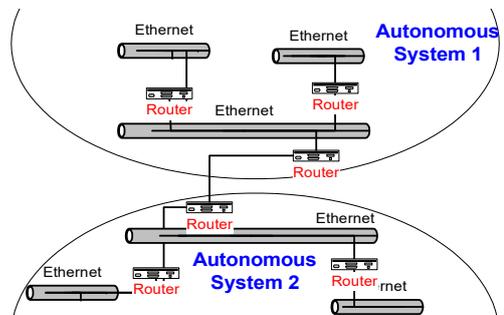
70

Sistemas Autônomos

- Um **sistema autônomo** é uma região da Internet que é administrada por uma entidade única.
- Exemplos:
 - Rede da RNP
 - Rede de Backbone da Embratel
 - ISP Regional
- Roteamento é feito de maneira diferente se estamos dentro de um sistema autônomo (**roteamento intra-domínio**) ou se devemos permitir que sistemas autônomos se comuniquem (**roteamento inter-domínio**)

71

Sistemas Autônomos (AS)



72

Roteamento Intradomínio e Interdomínio

Roteamento Intradomínio

- Roteamento dentro de um AS
- Ignora o que está fora do AS
- Protocolos para roteamento intradomínio também são chamados de **Interior Gateway Protocols** ou **IGP's**.
- Protocolos mais populares são:
 - RIP (simples, antigo)
 - OSPF (melhor)

Roteamento Interdomínio

- Roteamento entre AS's
- Assume que a Internet consiste de uma coleção de AS's interconectados
- Geralmente, há um roteador dedicado em cada AS que trata o tráfego interdomínio.
- Protocolos para roteamento interdomínio também são chamados **Exterior Gateway Protocols** ou **EGP's**.
- Protocolos de Roteamento:
 - EGP
 - BGP (mais recente)

73

Componentes de um Algoritmo de Roteamento

- Um procedimento para enviar e receber informações de alcançabilidade sobre a rede para outros roteadores
- Um procedimento para computação de rotas ótimas
 - Rotas são calculadas utilizando algoritmo de menor caminho (*shortest path*):
 - **Objetivo:** Dada uma rede, a cada enlace é atribuído um custo. Procurar o caminho de menor custo entre duas redes com custo mínimo.
- Um procedimento para reagir e anunciar mudanças na topologia

74

Abordagens para o Roteamento de Menor Caminho

- Existem duas estratégias de roteamento na Internet

1. Roteamento Vetor de Distância

- Cada nó conhece a distância (=custo) para seus vizinhos diretamente conectados
- Um nó envia periodicamente uma lista de atualizações de roteamento para seus vizinhos.
- Se todos os nós atualizam seus custos, as tabelas de roteamento eventualmente convergem
- Novos nós anunciam sua chegada a seus vizinhos

2. Roteamento Estado de Enlace

- Cada nó sabe a distância para seus vizinhos
- A informação de distância (=estado de enlace) é difundida para todos os nós da rede
- Cada nó calcula sua tabela de roteamento de forma independente

75

Algoritmos de Roteamento na Internet

Vetor de Distância

- Routing Information Protocol (RIP)
- Gateway-to-Gateway Protocol (GGP)
- Exterior Gateway Protocol (EGP)
- Interior Gateway Routing Protocol (IGRP)

Estado de Enlace

- Intermediate System - Intermediate System (IS-IS)
- Open Shortest Path First (OSPF)

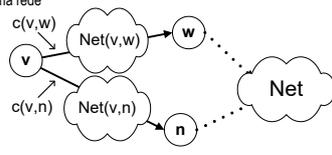
76

Uma rede é como um grafo

- Daqui em diante, redes são representadas como grafos:

- Nós estão conectados na rede
 - Rede pode ser um enlace ou uma LAN
- a interface da rede tem um custo
- redes são destinos
- $Net(v,w)$ é um endereço IP de uma rede

- Para facilitar a notação, normalmente as nuvens entre os nós são substituídas por enlaces.

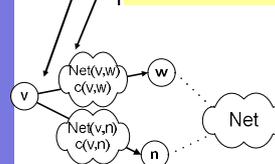


77

Algoritmo Vetor de Distância: Tabela de Roteamento

$c(v,w)$: custo para transmitir na interface para a rede $Net(v,w)$

$Net(v,w)$: Endereço de rede da rede entre v e w
A rede pode ser um enlace, mas também pode ser uma LAN



Routing Table of node v

Dest	via (next hop)	cost
Net	n	$c(v,Net)$

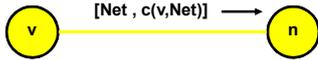
78

Algoritmo Vetor de Distância: Mensagens

RoutingTable of node v

Dest	via (next hop)	cost
Net	n	$c(v, \text{Net})$

Nós enviam mensagem para seus vizinhos que contém entradas na tabela de roteamento



Uma mensagem que tem o formato: $[\text{Net}, c(v, \text{Net})]$ significa "Meu custo para ir a Net é $c(v, \text{Net})$ "

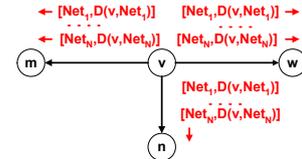
79

Algoritmo Vetor de Distância: Enviando Atualizações

RoutingTable of node v

Dest	via (next hop)	cost
Net ₁	m	$c(v, \text{Net}_1)$
Net ₂	n	$c(v, \text{Net}_2)$
...
Net _i	w	$c(v, \text{Net}_i)$

Periodicamente, cada nó v envia o conteúdo de sua tabela de roteamento para seus vizinhos:

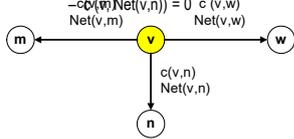


80

Iniciando uma Tabela de Roteamento I

- Suponha que o nó v torna-se ativo.
- O custo para acessar as redes diretamente conectadas é zero:

- $c(v, \text{Net}(v, m)) = 0$
- $c(v, \text{Net}(v, w)) = 0$
- $c(v, \text{Net}(v, n)) = 0$ $c(v, w)$



RoutingTable

Dest	via (next hop)	cost
Net(v, m)	m	0
Net(v, w)	w	0
Net(v, n)	n	0

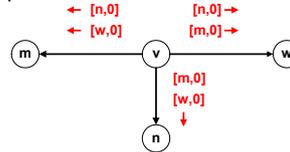
81

Iniciando uma Tabela de Roteamento II

RoutingTable

Dest	via (next hop)	cost
Net(v, m)	m	0
Net(v, w)	w	0
Net(v, n)	n	0

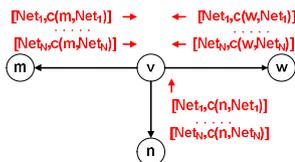
- O novo Nó v envia as entradas da tabela de roteamento para todos os vizinhos:



82

Iniciando uma Tabela de Roteamento III

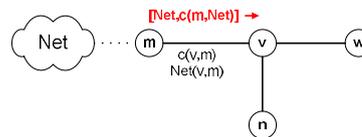
- Nó v recebe as tabelas de roteamento de outros nós e constrói sua tabela de roteamento



83

Atualizando as Tabelas de Roteamento I

Suponha que o nó v recebe uma msg de um nó m: $[\text{Net}, c(m, \text{Net})]$



O Nó v atualiza sua tabela de roteamento e envia mensagens para outros nós se a mensagem reduz o custo de uma rota

84

Atualizando as Tabelas de Roteamento II

Antes de receber a mensagem:

Suponha $c(m,Net) + c(v,m) < c(v,Net)$:

Routing Table

Dest	via (next hop)	cost
Net	??	$c(v,Net)$

Routing Table

Dest	via (next hop)	cost
Net	m	$c^{new}(v,Net)$

85

Exemplo

Assumir: - custo do enlace é 1, i.e., $c(v,w) = 1$
 - atualização de todos, atualiza simultaneamente
 - Inicialmente, cada roteador conhece apenas o custo das interfaces conectadas

Net	via	cost									
t=0: 10.0.1.0	-	0	t=0: 10.0.2.0	-	0	t=0: 10.0.3.0	-	0	t=0: 10.0.4.0	-	0
10.0.2.0	-	0	10.0.3.0	-	0	10.0.4.0	-	0	10.0.5.0	-	0
t=1: 10.0.1.0	-	0	t=1: 10.0.2.1	1	1	t=1: 10.0.3.1	1	1	t=1: 10.0.4.1	1	1
10.0.2.0	-	0	10.0.2.0	-	0	10.0.3.0	-	0	10.0.4.0	-	0
10.0.3.0	10.0.2.2	1	10.0.3.0	-	0	10.0.4.0	-	0	10.0.5.0	-	0
			10.0.4.0	10.0.3.2	1	10.0.5.0	10.0.4.2	1			
t=2: 10.0.1.0	-	0	t=2: 10.0.1.0	10.0.2.1	1	t=2: 10.0.1.0	10.0.3.1	2	t=2: 10.0.1.0	10.0.4.1	2
10.0.2.0	-	0	10.0.2.0	-	0	10.0.2.0	10.0.3.1	1	10.0.2.0	10.0.4.1	1
10.0.3.0	10.0.2.2	1	10.0.3.0	-	0	10.0.3.0	-	0	10.0.3.0	10.0.4.1	1
10.0.4.0	10.0.2.2	2	10.0.4.0	10.0.3.2	1	10.0.4.0	-	0	10.0.4.0	-	0
			10.0.5.0	10.0.3.2	2	10.0.5.0	10.0.4.2	1	10.0.5.0	-	0

86

Exemplo

Net	via	cost									
t=2: 10.0.1.0	-	0	t=2: 10.0.1.0	10.0.2.1	1	t=2: 10.0.1.0	10.0.3.1	2	t=2: 10.0.2.0	10.0.4.1	2
10.0.2.0	-	0	10.0.2.0	-	0	10.0.2.0	10.0.3.1	1	10.0.3.0	10.0.4.1	1
10.0.3.0	10.0.2.2	1	10.0.3.0	-	0	10.0.3.0	-	0	10.0.4.0	-	0
10.0.4.0	10.0.2.2	2	10.0.4.0	10.0.3.2	1	10.0.4.0	-	0	10.0.5.0	-	0
			10.0.5.0	10.0.3.2	2	10.0.5.0	10.0.4.2	1			
t=3: 10.0.1.0	-	0	t=3: 10.0.1.0	10.0.2.1	1	t=3: 10.0.1.0	10.0.3.1	2	t=3: 10.0.1.0	10.0.4.1	3
10.0.2.0	-	0	10.0.2.0	-	0	10.0.2.0	10.0.3.1	1	10.0.2.0	10.0.4.1	2
10.0.3.0	10.0.2.2	1	10.0.3.0	-	0	10.0.3.0	-	0	10.0.3.0	10.0.4.1	1
10.0.4.0	10.0.2.2	2	10.0.4.0	10.0.3.2	1	10.0.4.0	-	0	10.0.4.0	-	0
10.0.5.0	10.0.2.2	3	10.0.5.0	10.0.3.2	2	10.0.5.0	10.0.4.2	1	10.0.5.0	-	0

Agora, as tabelas de roteamento convergiram!

87

Características do Roteamento Vetor de Distância

- **Atualizações Periódicas:** Atualização das tabelas são realizadas num certo período de tempo. Um valor típico é 90 segundos.
- **Atualizações Disparadas:** Se uma métrica muda em um enlace, um roteador imediatamente envia uma mensagem de atualização sem esperar o período de atualização.
- **Atualização da Tabela de Roteamento Completa:** A maioria dos protocolos de vetor de distância enviam a seus vizinhos a tabela de roteamento inteira (não apenas as entradas modificadas).
- **Temporizadores para Invalidação de Rotas:** Entradas nas tabelas de roteamento são inválidas se não são atualizadas. Um período típico para invalidar uma entrada não atualizada é depois de 3-6 mensagens de atualização.

88

O Problema de Contagem ao Infinito

A's Routing Table

to	via (next hop)	cost
C	B	2

B's Routing Table

to	via (next hop)	cost
C	C	1

Link entre B e C cai

Diagram illustrating the infinite count-to-infinity problem: B's table updates to C via C (cost 00), then C via A (cost 03), then C via B (cost 04), then C via C (cost 00), then C via A (cost 03), then C via B (cost 04), then C via C (cost 00), and so on.

89

Contagem ao Infinito

- A razão para o problema da contagem ao infinito é que cada roteador tem apenas "next-hop-view" não através de quem está indo
- Por exemplo, no primeiro passo, A não percebe que sua rota (com custo 2) para C vêm através do nó B
- Como resolver o problema de Contagem ao Infinito?

90

Contagem ao Infinito

- A razão para o problema da contagem ao infinito é que cada nós tem apenas "next-hop-view" não através de quem está indo
- Por exemplo, no primeiro passo, A não percebe que sua rota (com custo 2) to C vêm através do nó B
- Como resolver o problema de Contagem ao Infinito?
- **Solução 1:** Sempre passar o caminho inteiro na mensagem de atualização (**Path vectors ou Vetor de Distância por Caminho**)
 - Se as tabelas são grandes, as mensagem de roteamento vão requerir largura de banda substancial
 - BGP usa esta solução

91

Contagem ao Infinito

- A razão para o problema da contagem ao infinito é que cada nós tem apenas "next-hop-view" não através de quem está indo
- Por exemplo, no primeiro passo, A não percebe que sua rota (com custo 2) to C vêm através do nó B
- Como resolver o problema de Contagem ao Infinito?
- **Solução 2:** Nunca anunciar o custo para um vizinho se este é o próximo salto no caminho atual (**Horizonte Dividido**)
 - Exemplo: A não enviaria a primeira atualização de rota para B, visto que B é o próximo salto na rota atual de A para C (A não manda antes de B)
 - Também falha, dependendo da situação

92

Algoritmo de Estado de Enlace

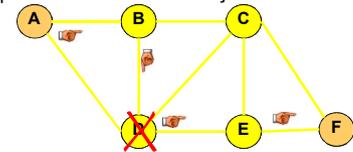
Reinaldo Gomes
reinaldo@dsc.ufog.edu.br

93

Distance Vector vs. Link State

- Usando roteamento de vetor de distância cada nó possui apenas a informação do próximo salto a ser alcançado

- Node A: para F vá por B
- Node B: para F vá por D
- Node D: para F vá por E
- Node E: vá direto para F

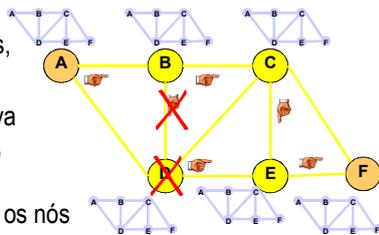


- Roteamento vetor de distância toma decisões pobres se os caminhos não se encontram completamente corretos (ex.: algum nó caiu)
- Se alguma informação se encontra incorreta o encaminhamento também será incorreto até que o algoritmo consiga convergir novamente

94

Distance Vector vs. Link State

- No roteamento por estado de enlace cada nó possui um mapa completo da topologia da rede
- Em caso de falhas, cada nó pode recalcular uma nova rota para o destino
- **Dificuldade:** Todos os nós precisam ter uma visão consistente da rede



95

Roteamento por Estado de Enlace

- **Propriedades**
 - Cada nó necessita da informação completa a respeito da topologia da rede
 - As informações a respeito do estado dos enlaces da rede deve ser inundada (*flooding*) para todos os nós
 - Garante a convergência das informações de roteamento

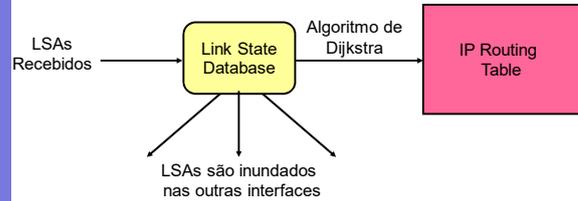
96

Roteamento por Estado de Enlace

- Princípios básicos
 - Cada roteador estabelece um relacionamento com seus vizinhos – adjacência
 - Cada roteador gera anúncios do estado do enlace (*link state advertisements - LSAs*) que são distribuídos para todos os roteadores - LSA = (link id, estado, custo, vizinho)
 - Cada nó mantém um banco de dados de todos os LSAs recebidos (*topological database* or *link state database*) que irá descrever o grafo que representa a topologia da rede
 - Cada roteador usa esse banco de dados para rodar o algoritmo de menor caminho (Dijkstra) para calcular as rotas para cada rede

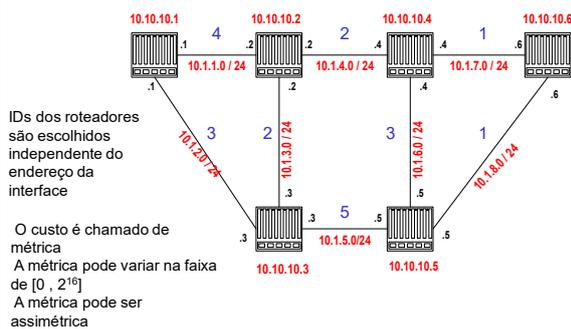
97

Operação do algoritmo de estado de enlace



98

Exemplo de rede



99

Link State Advertisement (LSA)

- O LSA do roteador 10.10.10.1 será:

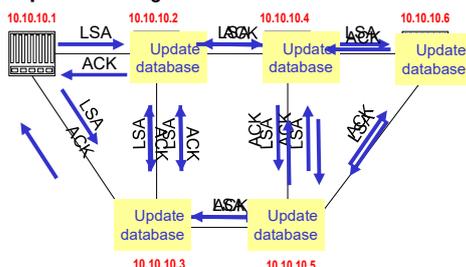
- Link State ID: 10.10.10.1 = ID do Roteador
- Advertising Router: 10.10.10.1 = ID do Roteador
- Número de links: 3 = 2 links mais o próprio roteador
- Descrição do Link 1: Link ID = 10.1.1.1, Métrica = 4
- Descrição do Link 2: Link ID = 10.1.2.1, Métrica = 3
- Descrição do Link 3: Link ID = 10.10.10.1, Métrica = 0

Cada roteador envia seu LSA para todos os demais roteadores da rede

100

Distribuição de Informações de Roteamento

- LSA-Updates são transmitidos através de um **Flooding Confiável (Reliable Flooding)**
- Exemplo: Flooding do LSA do roteador 10.10.10.1



101

Armazenamento das Informações

Cada roteador possui um banco de dados que contém os LSAs dos outros roteadores

LS Type	Link StateID	Adv. Router	Checksum	LS SeqNo	LS Age
Router-LSA	10.1.10.1	10.1.10.1	0x9b47	0x80000006	0
Router-LSA	10.1.10.2	10.1.10.2	0x219e	0x80000007	1618
Router-LSA	10.1.10.3	10.1.10.3	0x6b53	0x80000003	1712
Router-LSA	10.1.10.4	10.1.10.4	0xe39a	0x8000003a	20
Router-LSA	10.1.10.5	10.1.10.5	0xd2a6	0x80000038	18
Router-LSA	10.1.10.6	10.1.10.6	0x05c3	0x80000005	1680

102

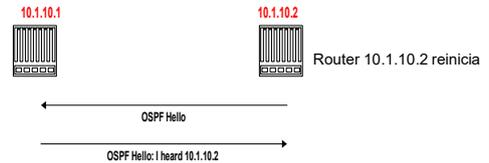
Link State Database

- O conjunto de todos os LSAs recebidos pelos nós é chamado de **link-state database**
- Cada roteador possui uma base de dados idêntica
 - Cada roteador necessita da descrição completa da rede para calcular as rotas
- Se roteadores vizinhos se comunicam pela primeira vez eles irão trocar as informações de suas bases de dados
- As bases de dados são sincronizadas através do mecanismo de disseminação dos dados do algoritmo (flooding confiável)

103

Descoberta de Vizinhos

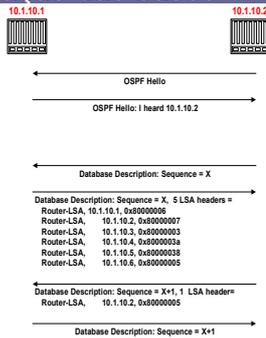
- Os roteadores transmitem pacotes de hello para todas as interfaces controladas pelo OSPF
- Se dois roteadores compartilham um link eles se tornam vizinhos e estabelecem a adjacência



- Depois de se tornarem vizinhos os roteadores trocam suas bases de informações

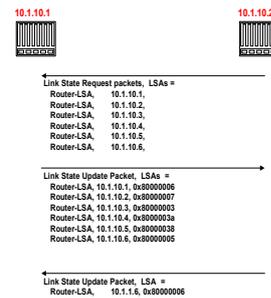
104

Descobre de vizinhos e sincronização das bases



105

Troca de LSAs



106