

Fundamentos de Redes de Computadores

Tiago Alves

Faculdade UnB Gama
Universidade de Brasília



Camada de Rede

- Introdução
- Circuitos Virtuais e Redes de Datagramas
- Detalhes de um Roteador
- IP: Encaminhamento e Endereçamento
- Algoritmos de Roteamento
- Roteamento na Internet
- Roteamento Broadcast e Multicast



Camada de Rede

A camada de transporte: comunicações entre processos. Confia integralmente nos serviços de comunicação providos pela camada de rede.

- A camada de transporte não se preocupa com a implementação da camada de rede: separação interface X implementação!

Nesse módulo, aprenderemos como a camada de rede implementa os serviços de comunicação entre hosts.

- Diferentemente das camadas superiores, há uma porção da camada de rede em cada um dos hosts e roteadores interligados em rede.
- Devido a sua abrangência, é um dos protocolos mais desafiadores da camada de protocolos.



Proposta de Estudos

Visão geral da camada de rede e dos serviços por ela provido.

Duas grandes abordagens para estruturação da entrega de pacotes em nível de camada de rede: **datagrama** e modelo de circuito virtual.



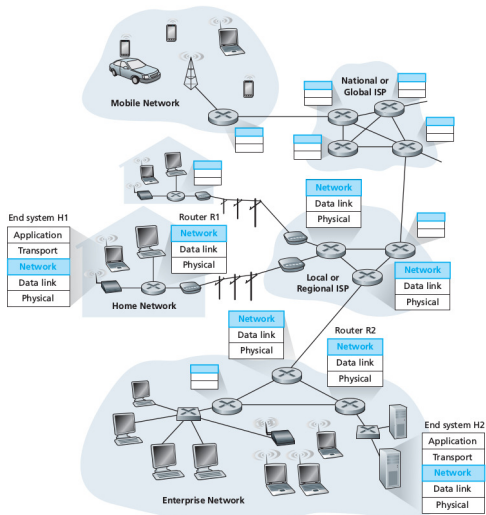
Proposta de Estudos

Divisão em três partes:

- serviços e funções da camada de rede
- encaminhamento: transferência de um pacote de um enlace de chegada para um enlace de saída em um mesmo (ou único) roteador
- roteamento: engloba todos os roteadores interligados na rede. A **interação** coletiva desses elementos, regulada por protocolos de roteamento, determina o caminho que pacotes tomam em seus percursos entre os nós de origem e destino.



Introdução



Cenário

H1 está enviando informação para H2. Análise do papel da camada de rede nesses hosts e nos roteadores de interligação.

- Camada de rede de H1 toma segmentos da camada de transporte de H1, **encapsula** cada segmento em um **datagrama** (pacote da camada de rede) e envia os datagramas para o roteador mais próximo, R1.
- A camada de rede do host H2 recebe os datagramas de seu roteador mais próximo, R2, extrai os segmentos da camada de transporte e entrega os segmentos para a camada de transporte de H2.

O papel primário dos roteadores: encaminhar datagramas de enlaces de chegada para os respectivos enlaces de saída.

A camada de transporte de roteadores pode não ser necessária: às vezes, não representada em diagramas.



Encaminhamento e Roteamento

O papel da camada de rede parece relativamente simples: mover pacotes entre um host emissor e um host receptor. Duas funções básicas:

- **Encaminhamento** (Forwarding): quando um pacote chega por um enlace de um roteador (enlace de chegada), o roteador deve mover (encaminhar) o pacote para o enlace de saída apropriado.
- **Roteamento** (Routing): a camada de rede deve determinar a rota (ou caminho) tomado pelos pacotes em seu trajeto entre o emissor e o receptor. **Atuação dos Algoritmos de Roteamento.**



Encaminhamento e Roteamento

Atenção: autores usam os termos como sinônimos. Nesse curso:

- **encaminhamento**: ação local de um roteador ao transferir um pacote de uma interface de enlace de entrada para a interface apropriada de enlace de saída.
- **roteamento**: processo de abrangência maior, pois considera o comportamento da rede ao determinar os caminhos de ponto de origem ao ponto de destino.



Encaminhamento e Roteamento

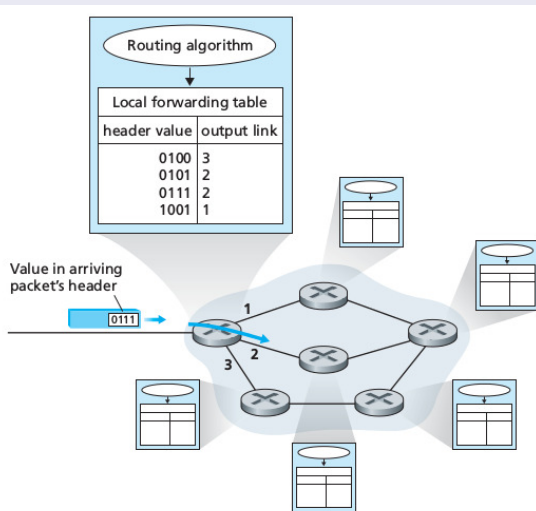
Cada roteador possui uma **tabela de roteamento**

- Um roteador encaminha um pacote depois de examinar o valor de um campo de cabeçalho do pacote que o alcançou. Essa informação de cabeçalho é usada na consulta da tabela de encaminhamento.
- A tabela de encaminhamento indica quais são as interfaces de enlace de saída a serem usadas no processamento dos pacotes.
- Informações de cabeçalho: endereço de destino do pacote, indicação a que conexão pertence o pacote.



Encaminhamento e Roteamento

Consulta à tabela de encaminhamento: tratamento do pacote com cabeçalho **0111**



Encaminhamento e Roteamento

Como **configurar** tabelas de roteamento?

- Cada roteador possui sua própria tabela de roteamento
- Algoritmos de roteamento determinam os valores que são inseridos nas tabelas de roteamento
 - **Algoritmos centralizados:** executados em um ponto central. Distribuição das informações de roteamento para cada um dos roteadores vinculados ao nó central
 - **Algoritmos descentralizados:** algoritmos distribuídos executando em cada um dos roteadores interligados.
- Mensagens de protocolos de roteamento são usadas para configurar as tabelas de roteamento.
- Cenário: roteadores configurados manualmente por humanos



Definições

Packet switch (comutador de pacotes): equipamento de comutação de pacotes que transfere um pacote de uma interface de enlace de entrada para uma interface de enlace de saída, respeitando o valor de um campo no cabeçalho do pacote.

- switches de camada enlace: baseiam decisões de encaminhamento nos valores dos campos de cabeçalho dos **quadros** da camada de enlace. Esses equipamentos são chamados de swiches de camada 2 ou de camada de enlace.
- roteadores: baseiam decisões de encaminhamento nos valores dos campos dos cabeçalhos dos **pacotes** (datagramas) da camada de rede. São dispositivos de camada 3 ou camada de rede, mas também implementam protocolos de camada 2



Encaminhamento e Roteamento

Estabelecimento de conexão

- Algumas arquiteturas de camada de rede (ATM, frame relay, MPLS) demandam que roteadores envolvidos no caminho de interligação confirmem uns aos outros de forma a ajustar um estado inicial antes de os pacotes de camada de rede iniciem seu caminho entre a origem e o destino.



Modelo de Serviços de Rede

Arquitetura em camadas. **Perguntas** típicas sobre a pilha de protocolos:

- A camada de transporte pode confiar na camada de rede para a entrega de pacotes ao host destino?
- Quando diversos pacotes são enviados, eles serão entregues à camada de transporte do host de destino na ordem correta em que eles foram enviados?
- O intervalo de tempo entre o envio de dois pacotes seguidos é o mesmo que será percebido durante a recepção desses pacotes?
- A rede proverá alguma indicação sobre o congestionamento no caminho?
- Qual o modelo abstrato do canal que interliga a camada de transporte entre os hosts emissores e receptores?



Modelo de Serviços de Rede

Respostas: modelo de serviços da rede.

- Define as características do transporte de pacotes fim a fim entre os end systems emissor e receptor.

Categoria de serviços:

- **Entrega garantida:** garante que o pacote eventualmente chegará a seu destino.
- **Entrega garantida com atraso limitado:** garante a entrega, que respeitará um limite de atraso especificado para a transmissão entre os hosts.
- **Entrega ordenada de pacote.**



Modelo de Serviços de Rede

Categoria de serviços:

- **Largura de banda mínima garantida:** nenhum pacote será perdido nem sofrerá atraso longo desde que o host emissor não viole o limite de taxa de transmissão estabelecido para a conexão.
- **Jitter máximo garantido:** o serviço garantirá que o intervalo de tempo entre a transmissão de dois pacotes seguidos no emissor será igual ao intervalo de tempo observado entre a recepção seguida desses dois pacotes no receptor.
- **Serviços de segurança:** serviço de sigilo de comunicações por meio de chave secreta de sessão conhecida apenas pelos integrantes legítimos da comunicação. A camada de rede responsável pela cifração/decifração das informações. Serviços agregados: autenticação e integridade de dados.



Modelo de Serviços de Rede

Internet: melhor esforço

- **não há garantia** de temporização entre os pacotes
- **não há garantia** de ordenação da recepção dos pacotes
- **não há garantia** de entrega, de fato, de pacotes transmitidos

Redes ATM

- CBR: **constant** bitrate
- ABR: **available** bitrate



Camada de transporte

Aplicações eram atendidas por dois tipos de serviços

- sem conexão: UDP
- com conexão: TCP

Camada de rede

Camada de rede: são possíveis as duas abstrações antes tratadas na camada de transporte.

Diferenças:

- os serviços providos são host a host (transporte: processo a processo.)
- camada de rede provê, de forma exclusiva, serviços de comunicação host a host sem conexão (rede de datagramas) ou serviços de comunicação host a host com conexão (circuito virtual), **jamais ambos simultaneamente**.
- as implementações dos serviços voltados a conexão da camada de rede são diferentes daqueles da camada de rede. Esses serviços, no nível de rede, são implementados em todos os roteadores do núcleo da rede bem como nos end systems.

Rede de circuitos virtuais

Muitas arquitetura de redes alternativas à Internet (frame relay e ATM) usam conexões de camada de rede chamadas de **circuitos virtuais**

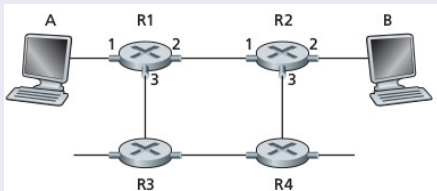
Componentes de circuitos virtuais:

- ❶ **Caminho de dados** (sequência de enlaces e roteadores) entre o emissor e o receptor
- ❷ **Número de VC**, um número para cada enlace do caminho
- ❸ **entradas na tabela de encaminhamento** de cada roteador ao longo do caminho
 - pacote percorrendo circuito virtual carregará número do VC em seu cabeçalho.
 - Cada circuito virtual possui um VC diferente em cada enlace: cada roteador do caminho deve modificar o número VC de cada pacote atravessando o enlace com um novo valor de VC
 - valor de VC é obtido da tabela de roteamento.



Rede de circuitos virtuais

caminho **A-R1-R2-B**. VCs 12, 22 e 32.



Incoming Interface	Incoming VC #	Outgoing Interface	Outgoing VC #
1	12	2	22
2	63	1	18
3	7	2	17
1	97	3	87
...

Rede de circuitos virtuais

Por que cada enlace possui um VC diferente?

- Indicar número de VC particular para cada enlace diminui o tamanho do campo que informará VC no cabeçalho do pacote
- A gestão dos números de VC é simplificada ao permitir um número diferente em cada enlace ser ajustado independentemente dos números escolhidos em outros enlaces pelo caminho \implies Se um mesmo número de VC fosse necessário para todos os enlaces do caminho, roteadores deveriam trocar e processar um número substancial de mensagens de para concordarem em um número VC comum.

Informação de estado de conexão: deve ser mantida pelos roteadores

- A cada conexão estabelecida através do roteador, uma nova entrada de conexão deve ser adicionada à tabela de encaminhamento do roteador.
- A cada conexão finalizada através do roteador, a entrada de conexão deve ser removida da tabela de encaminhamento do roteador.

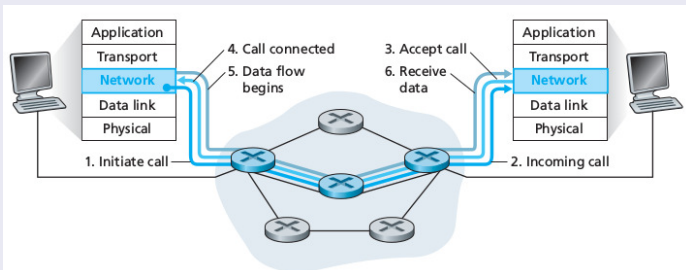


Rede de circuitos virtuais

Fases de um circuito virtual

Ajuste de VC:

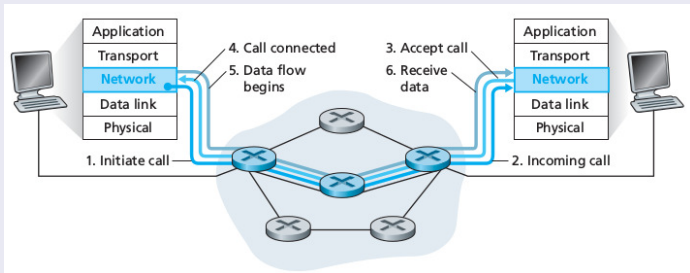
- A camada de transporte emissora comunica a camada de rede, especifica o endereço do receptor e espera pelo ajuste do VC por parte da camada de rede.
- A camada de rede determina o caminho entre o emissor e o receptor: sequência de enlaces e roteadores a serem atravessados pelos pacotes que trafegarão pelo VC.
- A camada de rede determina o número do VC para cada enlace pelo caminho.
- A camada de rede adiciona uma entrada na tabela de encaminhamento de cada roteador ao longo do caminho: alocação de recursos para o VC.



Rede de circuitos virtuais

Fases de um circuito virtual

Transferência de dados: uma vez estabelecido o VC, os dados podem começar a fluir.



Rede de circuitos virtuais

Fases de um circuito virtual

Encerramento de VC:

- Emissor (ou receptor) informa à camada de rede seu desejo de terminar o VC.
- A camada de rede, tipicamente, informará ao end system do outro lado da rede do encerramento da conexão e atualizará as tabelas de roteamento em cada um dos roteadores de pacotes do caminho de forma a sinalizar que o VC não existe mais.



Rede de circuitos virtuais

Diferença entre os ajustes de conexão nas camadas de transporte e de rede

- Camada de transporte: envolve apenas os end systems. São eles que determinam os parâmetros (número de sequência inicial, tamanho da janela de controle de fluxo, ...). Os roteadores são **transparentes** à camada de rede.
- Camada de rede: todos os roteadores do caminho entre os end systems se envolvem no estabelecimento do VC e cada roteador **tem conhecimento de todos os VCs** que o atravessam

Mensagens de sinalização: mensagens que os end systems mandam para a rede para iniciar e terminar VCs, e mensagens encaminhadas entre os roteadores para configurar um VC. Mensagens que **modificam o estado da conexão nas tabelas do roteador**.

Protocolos de sinalização: protocolos usados para trocar mensagens de sinalização.



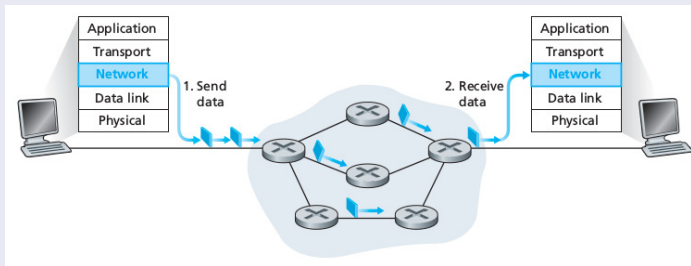
Redes de Datagramas

Em redes de datagramas, cada vez que um end system quer enviar um pacote, ele **grava** no pacote o endereço do destino e **insere** o pacote na rede.

- **Não** há a figura do **estabelecimento de VCs** e os roteadores **não mantêm informação alguma** de estado de VC.
- Um pacote atravessa uma sequência de roteadores ao percorrer o caminho entre o emissor e o receptor
- Cada roteador **usa o endereço de destino** do pacote para realizar encaminhamento
- Cada roteador possui uma **tabela de encaminhamento** que mapeia endereços de destino para endereços de interfaces de enlace
 - quando um pacote chega a um roteador, o roteador usa seu endereço de destino para definir qual é a interface de enlace de saída apropriada a partir de consultas em sua tabela de redirecionamento.
 - roteador encaminha o pacote para aquela interface de enlace.



Redes de Datagramas



Exemplo

Suponha que todos os endereços de destino são números de 32 bits.

Implementação ingênua: tabela de roteamento com entradas para cada um dos endereços de destino possíveis. **4 bilhões de entradas!**

Cenário: roteador com 4 enlaces (0 a 3) com a seguinte tabela de roteamento

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Prefix Match	Link Interface
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
otherwise	3



Exemplo

Cenário: roteador com 4 enlaces (0 a 3)

- O roteador bate/casa um prefixo do endereço de destino do pacote com as entradas da tabela
 - Se houver batimento, o roteador encaminha o pacote para o enlace correspondente
 - Se não houver batimento, há uma saída padrão (interface 3).
- **O batimento de prefixo mais longo precede outros batimentos.**

Prefix Match	Link Interface
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
otherwise	3



Redes de Datagramas

Estado (ou memória):

- Redes de datagramas mantêm informações de **estado de encaminhamento** em sua tabelas de encaminhamento. (Não mantém informações de estado de conexão.)
- Redes de datagramas modificam tabelas de encaminhamento através de **algoritmos de roteamento**
- Mudanças são relativamente lentas: a cada 1 ou 5 min.
- Uma sequência de pacotes enviados de um end system a outro pode seguir caminhos diferentes através da rede, o que os levarão a chegar **fora de ordem**.



História

Circuitos virtuais

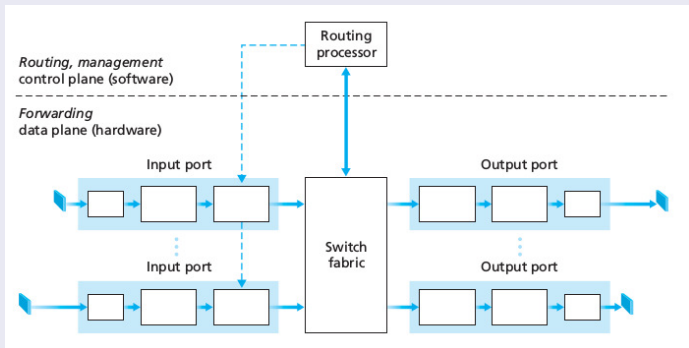
- Redes de telefonia: complexas
- Conectam sistemas terminais extremamente simples (telefones fixos, por exemplo)

Redes de datagramas

- Necessidade de se conectar computadores
- Decisão por fazer os serviços da camada de rede o mais simples o possível. Funcionalidades adicionais serão implementadas em camadas superiores.
- Consequências interessantes desse arcabouço:
 - facilita a interconexão de tecnologias de camada de enlace totalmente distintas: satélite, Ethernet, fibra óptica, rádio.
 - aplicações são implementadas em hosts (servidores) na **borda** da rede.



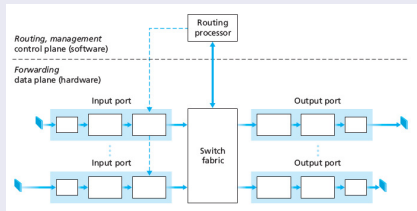
Arquitetura



Arquitetura

Portas de entrada (paralelos com as portas de saída)

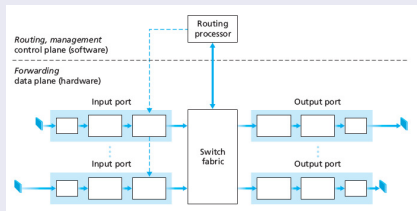
- Realiza a função de terminar um enlace físico em um roteador (caixas da esquerda/extremas)
- Realiza funções de camada de enlace que permitem a interoperabilidade com a camada de enlace do outro lado do enlace (caixas do meio)
- Realiza a função de busca (na tabela de encaminhamento) (caixas da direita/próximo da switch fabric)
- Encaminhamento de pacotes de controle (que carregam informação do protocolo de roteamento) são encaminhados de uma porta de entrada para o processador de roteamento



Arquitetura

Switching fabric

- Conecta as portas de entrada do roteador a suas portas de saída
- Contida no roteador: rede interna ao roteador de rede.

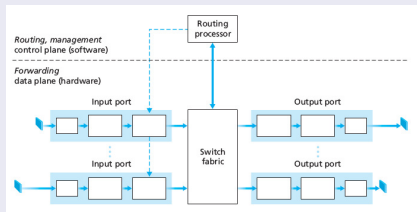


Arquitetura

Portas de saída: Armazena pacotes recebidos da switching fabric e transmite esses pacotes para o enlace de saída realizando as funções necessárias para a camada de enlace e camada física.

Processador de roteamento

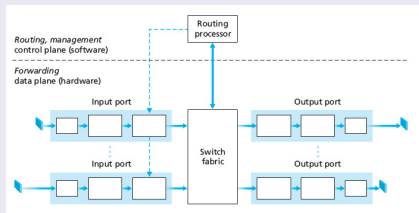
- Executa os protocolos de roteamento, mantém as tabelas de roteamento e os estados dos enlaces conectados ao roteador, calcula a tabela de encaminhamento para o roteador.
- Realiza algumas funções de gerenciamento de rede.



Arquitetura

Planos

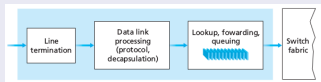
- **Plano de encaminhamento:** portas de entrada, de saída e switching fabric implementam, juntos, a função de roteamento e são quase sempre implementados em hardware. \Rightarrow Requisitos de vazão (largura de banda, velocidade de operação) somente atendida por meio de realizações em HW.
- **Plano de controle:** executa funções de controle, protocolos de roteamento, reage à mudanças de estados de ativação de enlaces conectados ao roteador, realiza funções de gerenciamento. \Rightarrow Funções implementadas em software, executado por um CPU.



Processamento na Entrada do Roteador

Funções

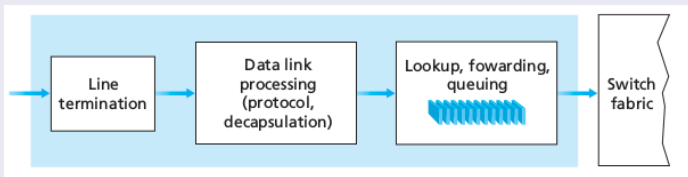
- Terminação de linha da porta de entrada e processamento da camada de enlace: implementa as camadas físicas e de enlace para cada enlace de entrada.
- A busca realizada na porta de entrada é fundamental para a operação do roteador: aqui o roteador usa a tabela de encaminhamento para buscar a porta de saída para onde um pacote que chega será encaminhado por meio da switching fabric.
- A tabela de encaminhamento é calculada e atualizada pelo processador de roteamento: cópia local de tabela é tipicamente armazenada em cada uma das portas de entrada.
 - Barramento especial para comunicação entre o processador de roteamento para os cartões de linha de entrada
 - Ter disponível cópia local economiza interações com o processador de roteamento: evita gargalos de processamento



Processamento na Entrada do Roteador

Desempenho

- Implementações em hardware são necessárias para prover o desempenho esperado
- Arquitetura de memória especial (Ternary Content Address Memory):
 - é empregada em buscas por rotas de saída: consultas em **tempo constante**.
 - o endereço apresentado à memória é um valor de 32 bits (um IP).
 - o valor armazenado em uma célula desse tipo de memória é o conteúdo da tabela de encaminhamento para aquele endereço de IP.



Switching fabric

Definição

- Coração do roteador
- Os pacotes são encaminhados/comutados de uma porta de entrada a uma porta de saída

Switching (comutação) por memória: os primeiros roteadores eram computadores comuns.

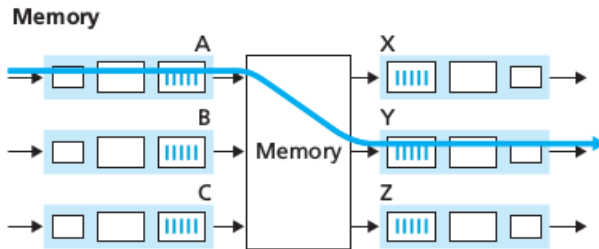
- Comutação entre portas de entrada e saída sob o controle direto da CPU (processador de roteamento)
- Portas: dispositivos tradicionais de IO de um SO.



Switching fabric

Procedimento de **comutação por memória**

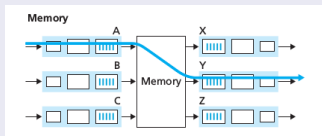
- 1 Porta de entrada sinaliza (interrompe) o processador de roteamento
- 2 Pacote copiado da porta de entrada para a memória do processador
- 3 Processador de roteamento extrai o endereço de destino do cabeçalho, faz o batimento da porta de saída apropriada na tabela de encaminhamento e copia o pacote para o buffer da porta de saída.



Switching fabric

Switching (comutação) por memória

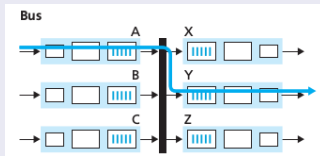
- Dada uma memória com largura de banda de B pacotes/s, a vazão total de roteamento deve ser menor de $B/2$ pacotes/s $\Leftarrow 1$ escrita + 1 leitura da memória.
- Muitos roteadores **modernos** usam dessa arquitetura
 - Diferença do legado: busca pelo endereço de destino e armazenamento do pacote na posição apropriada de memória são realizados pelo processamento no cartão de linha de entrada.
 - Roteadores que comutam por meio de memória são muito parecidos com processadores de memória compartilhada: processador de cartão de linha escreve pacotes na memória da porta correta de saída. Cisco Catalyst 8500.



Switching fabric

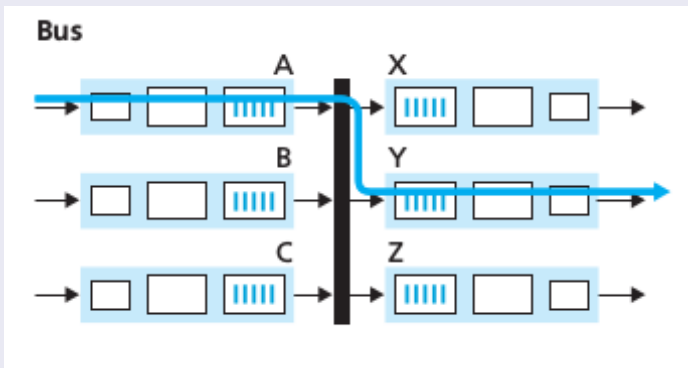
Switching (comutação) por barramento

- Porta de entrada transfere pacote diretamente para a porta de saída através do barramento compartilhado, sem intervenção de processador de roteamento
- Porta de entrada agrega (etiqueta no início do pacote) um rótulo interno de comutação ao pacote indicando a porta local de saída a ser usada e transmite o pacote para o barramento
- O pacote é recebido por todas as portas, mas apenas a porta que reconhece o rótulo interno coletará o pacote
- O rótulo interno é removido na porta de saída: rótulo é usado apenas para comutar no barramento.
- Velocidade de comutação do pacote limitada pela velocidade do barramento: apenas um pacote por vez é permitido no barramento. Cisco 5600: Backplane de barramento de 32 Gbps



Switching fabric

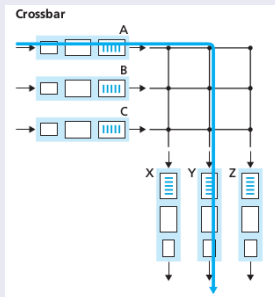
Switching (comutação) por barramento



Switching fabric

Switching (comutação) por meio de rede de interconexões

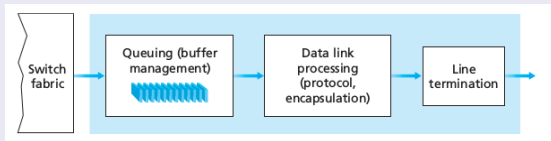
- Medida para superar a limitação de largura de banda de barramento único e compartilhado.
- Crossbar switch
 - 2N barramentos conectando N portas de entrada a N portas de saída
 - Interconexão das portas é controlada pela switching fabric
 - É possível comutar pacotes em paralelo \Rightarrow Cisco 12000



Processamento de saída

Porta de Saída: toma pacotes armazenados na memória da porta de saída e os transmite pelo enlace de saída.

Seleção e desenfileiramento de pacotes para transmissão e funções de transmissão de camadas enlace e física.



Processamento de saída

Onde ocorre enfileiramento?

- Podem acontecer tanto nas portas de entrada como nas portas de saída.
- Onde especificamente e por quanto tempo/tamanho?
 - Carga de tráfego
 - Velocidade relativa da switching fabric
 - Velocidade da linha (enlace)



Processamento de saída

Análise mais detalhada

- Se as filas crescerem demais, a memória do roteador poderá ser exaurida!
- Perdas de pacotes são prováveis nesse cenário: consumo de toda a memória da plataforma eletrônica
- Pacotes serão perdidos em roteadores!



Processamento de saída

Cenário

- linhas de entrada e de saída com a mesma taxa de transmissão R_{line} pacotes por segundo.
- N portas de entrada e N portas de saída
- Pacotes de mesmo tamanho e chegada síncrona (cadenciada) às portas de entrada.
 - tempo necessário para enviar um pacote por qualquer enlace é igual ao tempo necessário para receber um pacote por qualquer enlace.
 - durante esse intervalo, zero ou um pacote pode chegar por um enlace de entrada.
- Taxa de transmissão da switching fabric R_{switch} : taxa em que os pacotes poderão ser comutados de uma porta de entrada para uma porta de saída.



Processamento de saída

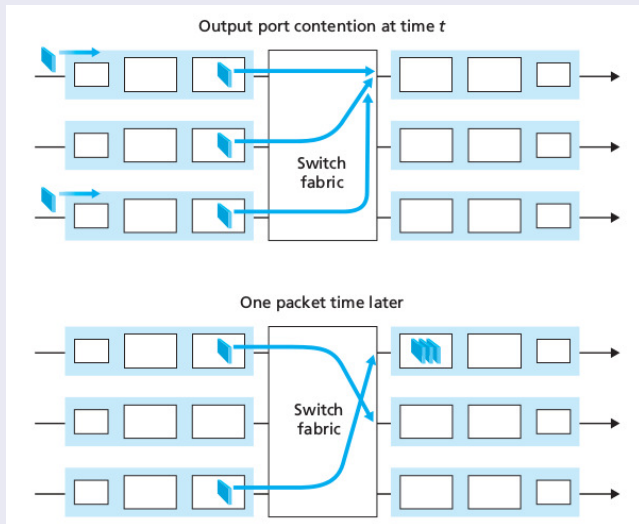
Cenário:

- Se $R_{\text{switch}} = N \cdot R_{\text{line}}$, enfileiramento irrisório nas portas de entrada. (Atenção: N é o número de portas de entrada ou de saída)
 - Pior caso: N entradas despachando para uma única saída. A switching fabric operará no limite, mas será capaz de mitigar enfileiramentos.
- Portas de saída: efeito da taxa de transmissão dos enlaces de saída.
 - Considerando as mesmas condições anteriores: $R_{\text{switch}} = N \cdot R_{\text{line}}$ e pacotes destinados a uma única porta de saída.
 - Enquanto a porta de saída despacha um pacote (em uma unidade de tempo), N novos pacotes chegarão na porta de saída e deverão **esperar** pela transmissão na fila do enlace de saída.
 - Enquanto isso acontece, mais N novos pacotes chegam à porta de saída. E assim por diante.
 - Exaustão de memória: crescimento rápido das filas na porta de saída.
 - Contingência: **descarte de pacotes**.



Processamento de saída

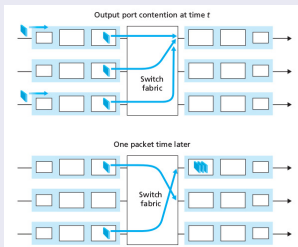
Cenário: processamento na porta de saída.



Processamento de saída

Cenário: processamento na porta de saída.

- no tempo t , pacotes chegaram a cada uma das portas de entrada, destinados à porta de saída superior
- assumindo velocidade de enlaces de entrada $1/3$ da velocidade da switching fabric, uma unidade de tempo depois, os 3 pacotes originais foram transferidos para a porta de saída e são enfileirados aguardando a transmissão.
- no próximo intervalo de tempo, um desses três pacotes será transmitido pelo enlace de saída.
- Ainda nesse contexto, dois novos pacotes chegaram por enlaces de entrada e um desses deverá ser encaminhado pelo enlace de saída.



Processamento de saída

Qual é uma contramedida efetiva para se proteger de enfileiramentos variáveis?

- Uso de buffers

Qual deverá ser o tamanho adequado de buffer para absorver flutuações na carga de tráfego?

- Por anos (RFC 3934), a regra geral para definição do tamanho do buffer era que o tamanho do armazenamento temporário (B) deveria ser igual a um RTT médio (RTT de 250 ms é um bom valor) multiplicado pela capacidade C do enlace.
- Parâmetros derivados de análises de dinâmicas de enfileiramento em número pequeno de fluxos TCP [Villamizar 1994].
- Portanto: enlace 10 Gbps, RTT de 250 ms $\implies B = RTT \cdot C = 2,5$ Gbits de buffers.



Processamento de saída

Novos estudos teóricos e experimentais [Appenzeller 2004] adequaram a heurística

- Quando o número de fluxos TCP N atravessando um enlace é grande, recomenda-se o seguinte valor: $B = RTT \cdot \frac{C}{\sqrt{N}}$
- Isso é importante quando um número muito grande de fluxos atravessam um backbone de grande porte de enlaces de roteadores: o valor de N grande ajuda a diminuir o valor de espaço de armazenamento temporário.



Processamento de saída

Escalonador de pacotes

- Elemento componente da porta de saída que escolhe um dos pacotes dentre aqueles enfileirados para transmissão.
- Sistemáticas
 - FCFS: First-come-first-served
 - WFQ: Weighted fair queueing: compartilhamento “justo” de um enlace de saída por diferentes conexões fim-a-fim que possuam pacotes enfileirados para transmissão.
- QoS.



Processamento de saída

O que acontece se não houver mais memória para armazenar um pacote de entrada a ser enfileirado?

Políticas de descarte de pacotes

- **Drop-tail:** remover o pacote mais novo da fila.
- Remoção de um ou mais pacotes presentes na fila para reservar espaço para os pacotes mais novos que entram na fila da porta de saída
- Às vezes convém descartar (ou marcar o cabeçalho de) pacote antes de o buffer encher de forma a prover um sinal de congestão para o emissor/transmissor.
- AQM: active queue management Algorithms. Random Early Detection (RED)



Processamento de saída

RED: Random Early Detection

- média ponderada de comprimento da fila de saída é monitorada
 - 1 se a média for menor que min_{th} e um pacote chegar, ele será enfileirado.
 - 2 se a fila estiver cheia ou se seu comprimento atual for maior que max_{th} , o pacote é marcado ou, simplesmente, descartado
 - 3 se o comprimento da fila estiver no intervalo $[min_{th}, max_{th}]$:
 - pacote é marcado
 - ou pacote é descartado com probabilidade que é tipicamente uma função do comprimento médio da fila, min_{th} e max_{th} .
- Há várias referências sobre o assunto.



Processamento de saída

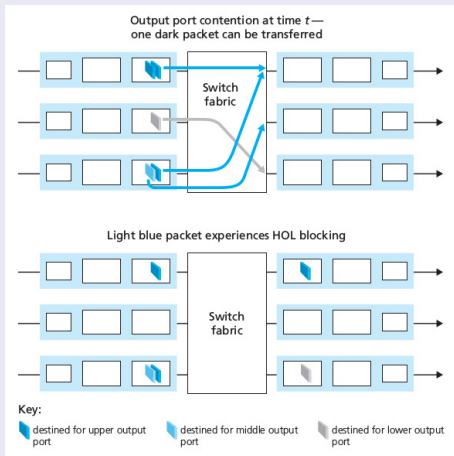
Cenário: switching fabric não é rápida o suficiente em comparação à taxa de transmissão dos enlaces de entrada:

- Enfileiramento nas portas de entrada.
- Exemplo: crossbar switching fabric
 - 1 as velocidades dos enlaces são iguais
 - 2 para um pacote ser transferido de qualquer porta de entrada para uma porta de saída demanda a mesma quantidade de tempo necessária para que ele seja recebido em um enlace de entrada
 - 3 pacotes são movidos de uma determinada fila de entrada para sua fila de saída usando a heurística FCFS. Vários pacotes poderão ser transferidos simultaneamente pela switching fabric desde que duas portas de saída sejam diferentes. Se dois pacotes em estado pronto para comutação forem destinados à mesma porta de saída, um dos pacotes será escolhido como o primeiro enquanto o outro aguardará enfileirado na fila de entrada.



Processamento de saída

Cenário: switching fabric não é rápida o suficiente em comparação à taxa de transmissão dos enlaces de entrada:



Bloqueio **head-of-the-line**.

Processamento de saída

Cenário: switching fabric não é rápida o suficiente em comparação à taxa de transmissão dos enlaces de entrada:

- HOL Blocking: **Head-of-the-line** blocking: um pacote enfileirado em uma fila de entrada deve aguardar a transferência através da fabric (malha) mesmo que sua porta de destino esteja livre porque está bloqueado por outro pacote na frente da linha.
- HOL Blocking poderá se manifestar com um enfileiramento na entrada que cresce indefinidamente (perdas de pacotes significativas) tão logo a taxa de chegada de pacotes nos enlaces de entrada alcance o nível de 58% de suas capacidades.



O plano de controle de roteamento

O plano de controle de roteamento reside e opera completamente em um processador de roteamento dentro do roteador.

O plano de controle de uma rede inteira é descentralizado:

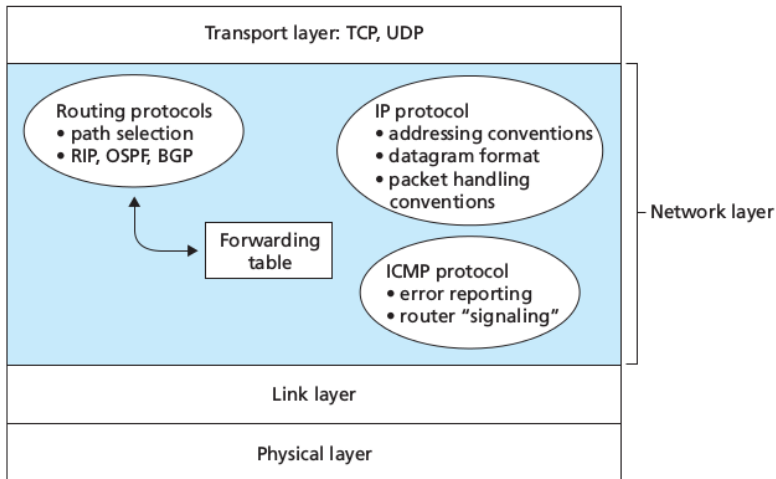
- Diferentes peças/atores (algoritmos de roteamento) executam em roteadores diferentes e interagem por meio de troca de mensagens de controle.
- Os fabricantes de roteadores e switches integram o hardware do plano de controle e o software do plano de controle em um conjunto fechado (mas interoperável) de plataformas.



Introdução

Duas versões de IP em uso: IPv4 [RFC 791] (maior foco dos estudos) e IPv6.

Visão em detalhes da camada de rede da Internet.



Introdução

Componentes da camada de rede

- Protocolo IP
- Componente (protocolos) de roteamento: determina como um datagrama se desloca entre uma origem e um destino: tabelas de encaminhamento
- Protocolos de relatório de erros e de estados da internet: ICMP (Internet Control Message Protocol) provê mecanismos para relatar erros em datagramas e responder a determinados pedidos de informações em nível de camada de rede

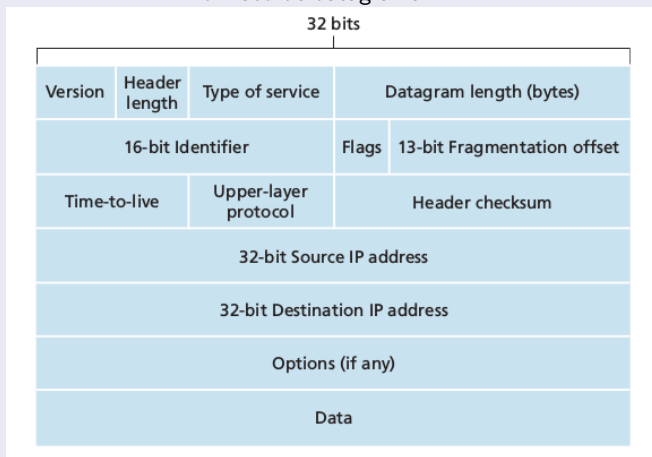


Formato de um datagrama

Unidade básica da camada de rede: datagrama

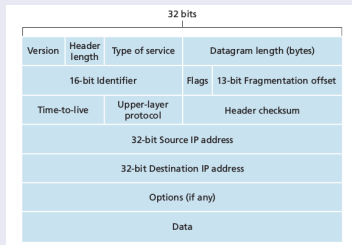
Cada profissional que interage com ativos de rede precisa ver, interpretar e dominar a interpretação de um datagrama!

Formato de datagrama IPv4



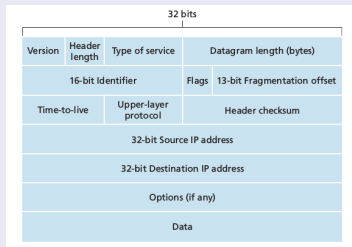
Formato de um datagrama

- Número de versão: 4 bits para indicar o número de versão. É a partir dessa informação que um roteador consegue determinar como interpretar o restante do datagrama IP.
- Comprimento de cabeçalho (**em palavras de 32 bits**): devido ao fato de um datagrama IPv4 poder possuir um número variado de opções (incluídas no cabeçalho do datagrama IPv4), esses 4 bits são necessários para determinar onde os dados (payload) do datagrama IP se iniciam.
 - A maioria dos datagramas não contém opções.
 - Cabeçalho típico de 20 bytes.



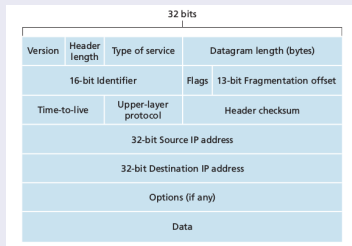
Formato de um datagrama

- TOS (Type of Service): incluído no cabeçalho para permitir a sinalização de diferentes tipos de datagramas IP (datagramas de baixa latência, alta vazão ou confiabilidade) de forma que sejam discriminados entre si. \Rightarrow Marcação para distinção de pacotes de tráfego tempo-real (VoIP) de tráfego comum (FTP).



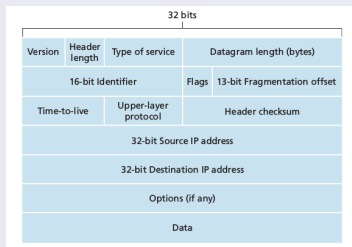
Formato de um datagrama

- Comprimento do datagrama: comprimento total do datagrama IP (cabeçalho mais dados), medido em bytes. Como são reservados 16 bits para esse campo, o comprimento máximo que pode ser gravado é 65535 bytes.
 - Datagramas tipicamente possuem comprimento de 1500 bytes.
- Identificadores, flags e offset de fragmentação: tratam da fragmentação IP. O protocolo IPv6 não permite a fragmentação de datagramas em roteadores.
- TTL (time-to-live): campo incluído para garantir que datagramas não circulem indefinidamente (por exemplo, numa rota circular) pela rede. É um campo cujo valor é decrementado cada vez que o datagrama é processado em um roteador: quando o datagrama alcança o valor 0, é descartado.



Formato de um datagrama

- Protocolo: usado apenas quando o datagrama alcança seu destino final.
 - Indica o protocolo específico da camada de transporte que deverá receber o payload do datagrama IP.
 - Valor 6: TCP
 - Valor 17: UDP
 - Lista de todos os valores possíveis: [IANA Protocol Numbers 2012]
 - Papel similar ao feito pelo campo de número de porta no segmento de camada de transporte.
 - A cola que emenda as camadas de transporte e de rede, ao passo que a porta da camada de transporte é a cola entre a camada de transporte e a camada de aplicação



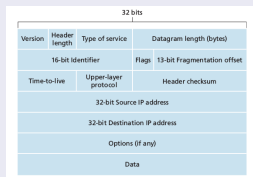
Formato de um datagrama

- Checksum de cabeçalho
 - Ajuda o roteador a identificar erros de bits em um datagrama IP.
 - Cálculo
 - Cada dois bytes do cabeçalho é tratado como um número, que são somados usando aritmética de complemento de 1.
 - Precisa ser recalculado e armazenado em cada roteador antes do encaminhamento, pois alguns campos de cabeçalho são alterados durante o encaminhamento (TTL, campo de opções, ...).
 - Um roteador calcula o checksum do header e o compara com o checksum declarado no cabeçalho do datagrama: incompatibilidade de valores sinaliza erro.
 - Descarte de datagrama no caso de erro. (tipicamente)
 - Dois checksum: IP (cabeçalho) e TCP/UDP (segmento completo) \implies Nem sempre um segmento TCP será transportado por IP (ATM, por exemplo).



Formato de um datagrama

- IP de origem e de destino
 - Quando um emissor cria um datagrama, ele insere seu endereço de IP no campo de IP de origem e insere o endereço de IP do destino no campo IP de destino.
 - Um host toma conhecimento do IP de seu destino por meio do DNS.
- Opções (raramente): permite que um cabeçalho IP seja estendido.
 - Campo projetado para ser usado raramente: nem sempre incluído nos datagramas para não implicar overhead
 - Cabeçalho de comprimento variável: nem sempre é possível saber, a priori, onde começará o campo de dados (payload)
 - A presença de um campo de opções no cabeçalho pode causar variação no tempo de processamento de um datagrama
 - Foi removido do cabeçalho IPv6

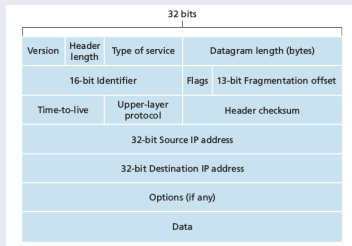


Formato de um datagrama

- Dados (payload): contém um segmento da camada de transporte (TCP, UDP) ou mensagens ICMP.

No total, 20 bytes, sem opções!

- datagrama que carrega segmento TCP possuirá 40 bytes de cabeçalho



Fragmentação de datagramas IP

Nem todos os protocolos de camada de enlace são capazes de carregar pacotes da camada de rede de mesmo tamanho

- Ethernet: 1500 bytes de dados
- Enlaces WAN: 512 bytes de dados

Tamanho máximo (em bytes) de dados que um quadro da camada de enlace é capaz de conduzir: MTU.

- Em função do encapsulamento, o MTU da camada de enlace insere um limite fixo no tamanho de um datagrama IP

Tamanhos fixos não constituem um problema. O problema é ter um conjunto de **valores diferentes** desses tamanhos **fixos** no caminho de um pacote.



Fragmentação de datagramas IP

Cenário: Um roteador que interage com vários enlaces, cada um operando com um protocolo de camada de enlace com MTU diferente.

- O roteador recebe um datagrama IP de um enlace
- Em seguida, confere a tabela de encaminhamento para determinar o enlace de saída, que, por sua vez, possui um MTU menor que o comprimento do datagrama.
- **Fragmentação**: separação dos datagramas IP em fragmentos, e cada fragmento encapsulado em um quadro da camada de enlace. Envio de cada um desses quadros **separadamente** pelo enlace de saída



Fragmentação de datagramas IP

Cenário: Um roteador que interage com vários enlaces, cada um operando com um protocolo de camada de enlace com MTU diferente.

Fragmentos precisam ser **remontados** antes de serem entregues à camada de transporte no destino!

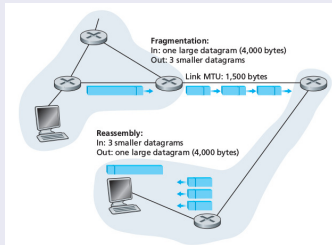
- TCP e UDP esperam pacotes completos.
- Decisão de projeto de protocolo: o trabalho de remontagem dos datagramas fragmentados acontece nos end systems de forma a liberar do núcleo da rede (roteadores) dessa atividade.
- Quando um destinatário recebe uma série de datagramas do mesmo emissor, é necessário determinar se alguns dos datagramas são fragmentos de um datagrama maior
- Se alguns datagramas forem fragmentos, deve ser decidido quando foi recebido o último fragmento e como os fragmentos recebidos devem ser unidos para compor o datagrama original



Fragmentação de datagramas IP

Cenário: Um roteador que interage com vários enlaces, cada um operando com um protocolo de camada de enlace com MTU diferente.

- Projetistas do IP puseram uma flag de identificação e um campo de offset de fragmentação no cabeçalho dos datagramas IP.
- Quando um datagrama é criado, o **emissor marca** o datagrama com um número de identificação bem como insere as informações de endereço de origem e de destino.
- Tipicamente, o host emissor incrementa o número de identificação a cada datagrama que envia
- Quando um roteador necessita fragmentar um datagrama, cada datagrama resultante da fragmentação é criado usando as mesmas informações de cabeçalho do datagrama original: número de identificação, IPs de origem e de destino.



Fragmentação de datagramas IP

Cenário: Um roteador que interage com vários enlaces, cada um operando com um protocolo de camada de enlace com MTU diferente.

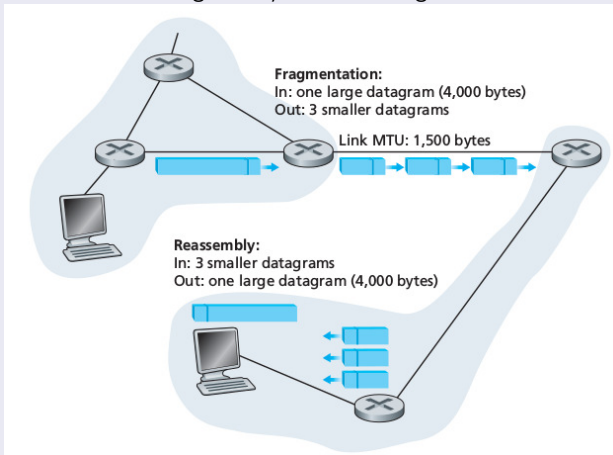
- Projetistas do IP puseram uma flag de identificação e um campo de offset de fragmentação no cabeçalho dos datagramas IP.
 - Quando o receptor recebe uma série de datagramas do mesmo host emissor, ele examina se os números de identificação para determinar se os datagramas são completos ou são, na verdade, **fragmentos** de um datagrama maior.
 - Pelo fato de IP ser um serviço não-confiável, é possível que alguns datagramas/fragmentos nunca alcancem o host de destino.
 - Para que o host de destino tenha absoluta certeza de que recebera o último fragmento de um datagrama maior, o último fragmento tem um dos bits de flag do cabeçalho marcado com 0, ao passo que todos os outros fragmentos possuem, em seu cabeçalho, o bit de flag marcado com 1.
 - Para que o host de destino determine se um fragmento está faltando (e para que seja capaz de remontar os fragmentos na devida ordem) o campo de offset é usado para especificar onde o fragmento se encaixa na montagem do datagrama IP original.



Fragmentação de datagramas IP

Cenário: Um roteador que interage com vários enlaces, cada um operando com um protocolo de camada de enlace com MTU diferente.

Fragmentação e remontagem.



Fragmentação de datagramas IP

Cenário: Um roteador que interage com vários enlaces, cada um operando com um protocolo de camada de enlace com MTU diferente.

- Datagrama de 4000 bytes: 20 bytes de cabeçalho e 3980 bytes de payload
 - enlace com MTU de 1500 bytes
 - fragmentação: 3 porções
 - número de identificação: 777
 - comprimentos: múltiplo de 8 bytes \Rightarrow offset de 8 bytes
- No destino, o datagrama é encaminhado à camada de transporte apenas depois de o IP ter reconstruído completamente o datagrama IP original.
 - Se um ou mais fragmentos não alcançarem o destino, o datagrama incompleto é totalmente descartado e não é encaminhado para a camada de transporte
 - Fica a cargo do TCP se recuperar desse evento de perda provocando uma **retransmissão** de dados pelo emissor para reenvio do datagrama perdido.



Fragmentação de datagramas IP

Fragmentos IP

Fragment	Bytes	ID	Offset	Flag
1st fragment	1,480 bytes in the data field of the IP datagram	identification = 777	offset = 0 (meaning the data should be inserted beginning at byte 0)	flag = 1 (meaning there is more)
2nd fragment	1,480 bytes of data	identification = 777	offset = 185 (meaning the data should be inserted beginning at byte 1,480. Note that $185 \cdot 8 = 1,480$)	flag = 1 (meaning there is more)
3rd fragment	1,020 bytes (= $3,980 - 1,480 - 1,480$) of data	identification = 777	offset = 370 (meaning the data should be inserted beginning at byte 2,960. Note that $370 \cdot 8 = 2,960$)	flag = 0 (meaning this is the last fragment)



Endereçamento IPv4

Tópico de importância central à Internet.

Como roteadores e host são/estão conectados à Internet

- Tipicamente, um host possui apenas um enlace para a rede: protocolo IP usa esse enlace para enviar dados à rede.
- Interface: fronteira entre o host e o enlace físico.
- Roteador: múltiplas interfaces interligadas por diferentes enlaces
- Cada host e roteador deve possuir seu próprio endereço IP
 - o IP é associado à **interface**, não ao host/roteador



Endereçamento IPv4

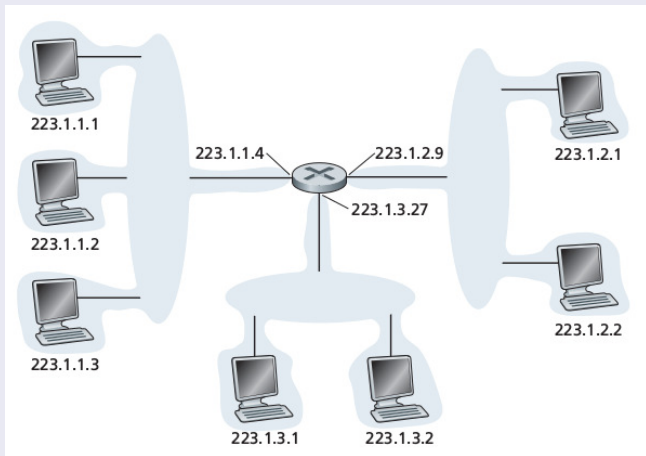
Endereços

- 32 bits de comprimento (4 bytes) $\Rightarrow 2^{32}$ endereços possíveis (~ 4 bilhões)
- Notação “dígito ponto”: cada **byte** do endereço é escrito na forma decimal e é separado por um ponto (.)
 - $193.32.216.9 \Rightarrow 11000001\ 00100000\ 11011000\ 00001001$
- Cada interface em cada host e roteador da Internet deve possuir seu número de IP global e exclusivo (exceção: NAT)
- Uma porção do endereço de IP da interface será determinado pela **subrede** à qual a interface está conectada.



Endereçamento IPv4

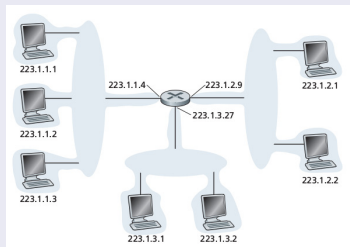
Um roteador com 3 interfaces interligando 7 hosts



Endereçamento IPv4

Um roteador com 3 interfaces interligando 7 hosts

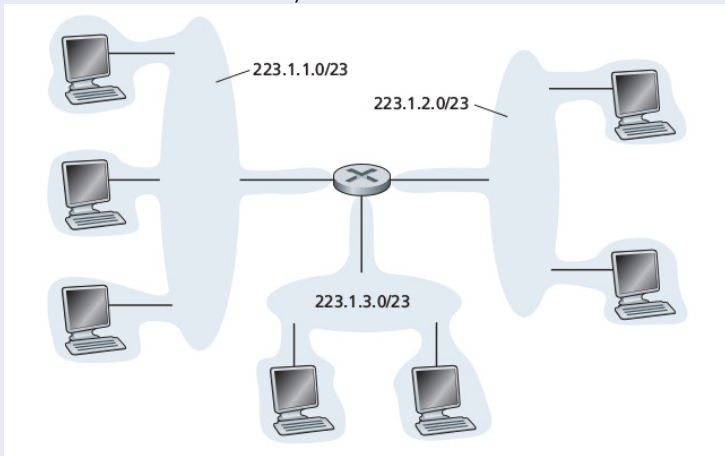
- 223.1.1.xxx : os mesmos 24 bits do endereço IP se repete na rede da esquerda, servida pela interface com IP 223.1.1.4
- Interconexão uns aos outros por meio de uma “rede” que não contém roteadores: LAN Ethernet (provida por switch Ethernet) ou um AP WiFi.
- SUBREDE [RFC 950], rede IP ou, simplesmente, rede.
 - 223.1.1.0/24, máscara de subrede: os 24 bits da esquerda
 - Elementos 223.1.1.1, 223.1.1.2, 223.1.1.3 e a interface de roteador em 223.1.1.4.
 - Qualquer outro elemento que venha fazer parte desta subrede deve ter um endereço na forma 223.1.1.xxx.



Endereçamento IPv4

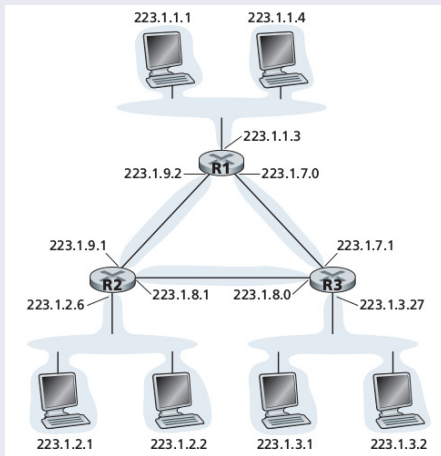
Um roteador com 3 interfaces interligando 7 hosts

Endereçamento de subredes.



Endereçamento IPv4

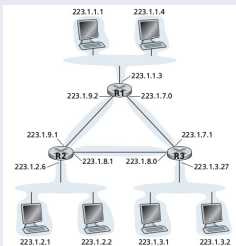
3 roteadores interligando 6 subredes.



Endereçamento IPv4

3 roteadores interligando 6 subredes.

- 3 subredes: 223.1.1.0/24, 223.1.2.0/24 e 223.1.3.0/24
- 3 subredes adicionais:
 - 223.1.9.0/24: interfaces entre R1 e R2;
 - 223.1.8.0/24: interfaces entre R2 e R3;
 - 223.1.7.0/24: interfaces entre R3 and R1.



Endereçamento IPv4

Regra para definir subredes

- Desconecte cada interface de seu host ou roteador e crie ilhas de redes isoladas.
- As interfaces delimitarão os end points das redes isoladas
- Cada uma dessas redes isoladas é considerada uma subrede.

Redes corporativas

- Compostas por múltiplos segmentos Ethernet, enlaces ponto-a-ponto.
- Todos os dispositivos de uma mesma subrede possuem o mesmo endereço de subrede.



Endereçamento IPv4

CIDR: Classless Interdomain Routing [RFC 4632]

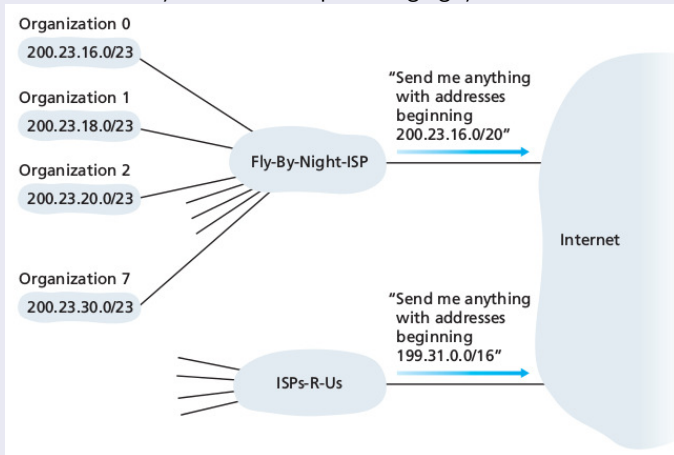
- Endereço dividido em duas porções: a.b.c.d/x, x representa a quantidade de bits da primeira porção do endereço
 - os x bits iniciais são a porção que define o número da subrede: prefixo ou prefixo de rede do endereço IP.
- Organizações recebem, em geral, um bloco contíguo de endereços IP com o mesmo prefixo comum.
 - agregação de endereços, de rotas ou sumarização de rotas: ISP informa que é capaz de rotear para uma determinada subrede.
 - Roteamento pelo prefixo mais longo: o caminho preferencial é determinado pela máscara que marca mais bits



Endereçamento IPv4

CIDR: Classless Interdomain Routing [RFC 4632]

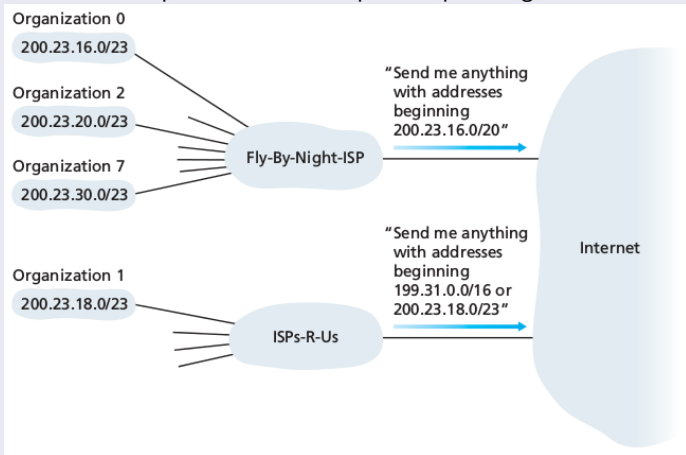
Endereçamento hierárquico e agregação de rotas.



Endereçamento IPv4

CIDR: Classless Interdomain Routing [RFC 4632]

ISPs-R-Us possui rota mais específica para Organization 1



Endereçamento IPv4

CIDR: Classless Interdomain Routing [RFC 4632]

- Fazer casamento para roteamento baseado apenas nos x primeiros bits simplifica a implementação dos serviços de rede:
 - esconde complexidades da topologia adotada internamente na organização para interligar seus ativos;
 - diminui o tamanho das tabelas de roteamento
- Os $(32 - x)$ bits restantes do endereçamento servem para distinguir dispositivos na organização
 - São considerados no roteamento intra-organizacional
 - É possível criar outras subredes dentro da organização



Endereçamento IPv4

Classfull addressing: legado

- endereços rede IP restritos a 8, 16 ou 24 bits: redes classes A, B e C
- Problemas quando o número de hosts interligado pelas redes aumentou subitamente
 - Classe C, 24 bits, 254 hosts: 2 dos 2^8 endereços são reservados para usos especiais. Isso é pouco para muitas organizações.
 - Classe B (/16), 65534 hosts
- Endereço de Broadcast: 255.255.255.255



Endereçamento IPv4

Obtendo um bloco de endereços

Passos:

- 1 Administrador deve contactar o ISP
- 2 ISP provê endereços de um bloco maior já alocado para gerência por parte do ISP
 - Bloco maior: 200.23.16.0/20
 - Clientes
 - Bloco do ISP 200.23.16.0/20 11001000 00010111 00010000 00000000
 - Organization 0 200.23.16.0/23 11001000 00010111 00010000 00000000
 - Organization 1 200.23.18.0/23 11001000 00010111 00010010 00000000
 - Organization 2 200.23.20.0/23 11001000 00010111 00010100 00000000
 -
 - Organization 7 200.23.30.0/23 11001000 00010111 00011110 00000000



Endereçamento IPv4

Quem provê IPs para o ISP?

- Internet Corporation for Assigned Names and Numbers (ICANN)
 - Organização sem fins lucrativos, aloca endereços de IP e mantém os root servers DNS
 - Atua em arbitragem de disputas por nomes de domínios



Endereçamento IPv4

Obtendo endereço de IP para um equipamento: **DHCP**

- Roteadores são, em geral, configurados manualmente
- Como a quantidade de hosts numa organização pode ser muito grande, é inviável configurá-los individual e manualmente
 - DHCP: Dynamic Host Configuration Protocol [RFC 2131]
 - Configurações permitem que um mesmo equipamento receba o mesmo IP todas as vezes em que se conecta à rede ou um IP temporário diferente a cada conexão.
 - Configurações adicionais automáticas: máscara de subrede, endereço do roteador do primeiro hop (gateway) e o endereço do DNS local
 - Reúso de números de IP



Endereçamento IPv4

DHCP: descrição

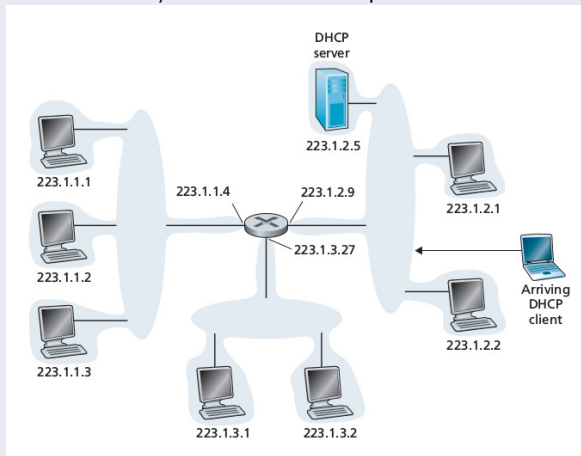
- protocolo cliente-servidor
- DHCP relay agent: registro no roteador



Endereçamento IPv4

DHCP: descrição

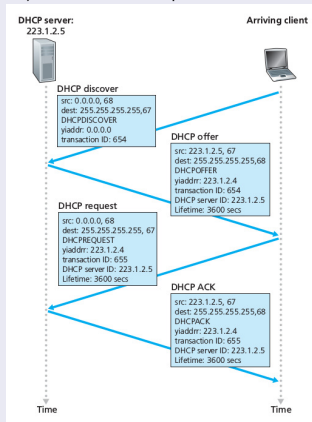
Cenário de interações cliente-servidor para o Protocolo DHCP



Endereçamento IPv4

DHCP: descrição

Interações cliente-servidor para o Protocolo DHCP

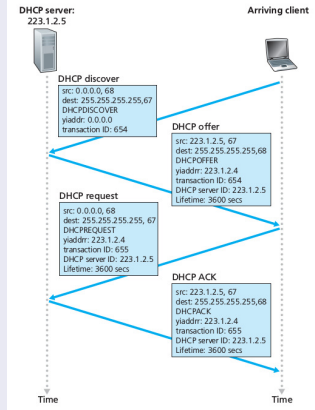


Endereçamento IPv4

DHCP: passos

- ❶ DHCP server discovery: pacote UDP na porta 67.
 - Endereço de destino broadcast: 255.255.255.255.;
 - Endereço de origem: 0.0.0.0
- ❷ DHCP server offers
 - Endereço de destino broadcast
 - No caso de mais de um servidor DHCP: **transaction ID** correlacionado com a mensagem de **discover**
 - Formato da resposta: **transaction-ID** do **request**, endereço de IP proposto, máscara de rede e o lease time do endereço IP: algumas horas ou dias

Interações cliente-servidor para o Protocolo DHCP



Endereçamento IPv4

NAT: Network Address Translation

- A Internet não pode ser mais chamada de “Rede Mundial de Computadores”: os equipamentos conectados à Internet são dos mais diversos tipos, mas não são somente **computadores**.
- Como implementar uma rede de computadores se o seu ISP só lhe provê um endereço de IP válido?
 - NAT: RFC 3022



Endereçamento IPv4

Roteador com NAT

- Subrede doméstica: 10.0.0.0/24
 - 10.0.0.0/8: uma das três porções do espaço de endereçamento IP que é reservado para redes privadas [RFC 1918]
 - realm (reino) com endereços privados: endereços fazem sentido apenas naquela rede. **Não há roteamento** de fora da rede para dentro da rede para que solicitações sejam respondidas pelos endereços da faixa 10.0.0.0/8.
 - **Q:** Como o roteador vai entregar a resposta a um pacote enviado por uma máquina local (endereço na faixa 10.0.0.0/8)? **A:** Roteador com NAT.



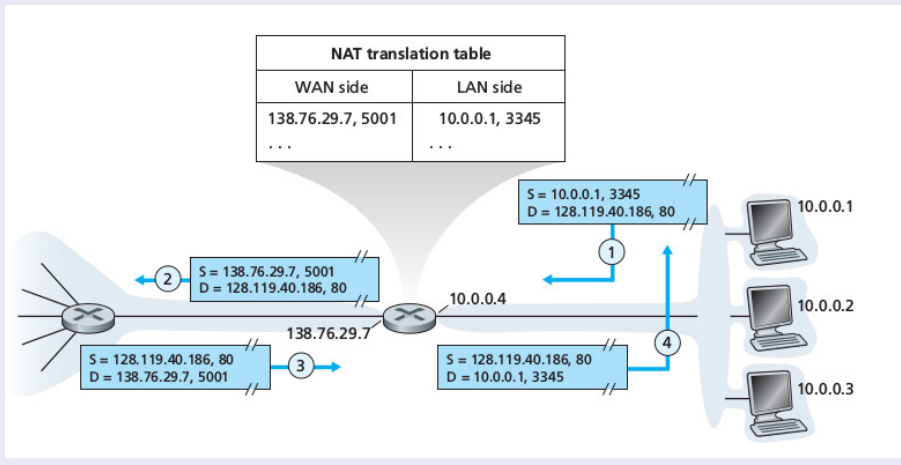
Protocolo IP: Encaminhamento e Endereçamento na Internet

Endereçamento IPv4

0.0.0.0/8	0.0.0.0 – 0.255.255.255	16,777,216	Software	Used for broadcast messages to the current ("this") network as specified by RFC 1700 , page 4.
10.0.0.0/8	10.0.0.0 – 10.255.255.255	16,777,216	Private network	Used for local communications within a private network as specified by RFC 1918 .
100.64.0.0/10	100.64.0.0 – 100.127.255.255	4,194,304	Private network	Used for communications between a service provider and its subscribers when using a Carrier-grade NAT , as specified by RFC 6598 .
127.0.0.0/8	127.0.0.0 – 127.255.255.255	16,777,216	Host	Used for loopback addresses to the local host, as specified by RFC 990 .
169.254.0.0/16	169.254.0.0 – 169.254.255.255	65,536	Subnet	Used for link-local addresses between two hosts on a single link when no IP address is otherwise specified, such as would have normally been retrieved from a DHCP server, as specified by RFC 3927 .
172.16.0.0/12	172.16.0.0 – 172.31.255.255	1,048,576	Private network	Used for local communications within a private network as specified by RFC 1918 .
192.0.0.0/24	192.0.0.0 – 192.0.0.255	256	Private network	Used for the IANA IPv4 Special Purpose Address Registry as specified by RFC 5736 .
192.0.2.0/24	192.0.2.0 – 192.0.2.255	256	Documentation	Assigned as "TEST-NET" in RFC 5737 for use solely in documentation and example source code and should not be used publicly.
192.88.99.0/24	192.88.99.0 – 192.88.99.255	256	Internet	Used by 6to4 anycast relays as specified by RFC 3068 .
192.168.0.0/16	192.168.0.0 – 192.168.255.255	65,536	Private network	Used for local communications within a private network as specified by RFC 1918 .
198.18.0.0/15	198.18.0.0 – 198.19.255.255	131,072	Private network	Used for testing of inter-network communications between two separate subnets as specified in RFC 2544 .
198.51.100.0/24	198.51.100.0 – 198.51.100.255	256	Documentation	Assigned as "TEST-NET-2" in RFC 5737 for use solely in documentation and example source code and should not be used publicly.
203.0.113.0/24	203.0.113.0 – 203.0.113.255	256	Documentation	Assigned as "TEST-NET-3" in RFC 5737 for use solely in documentation and example source code and should not be used publicly.
224.0.0.0/4	224.0.0.0 – 239.255.255.255	268,435,456	Internet	Reserved for multicast assignments as specified in RFC 5771 . 233.252.0.0/24 is assigned as "MCAST-TEST-NET" for use solely in documentation and example source code.

Endereçamento IPv4

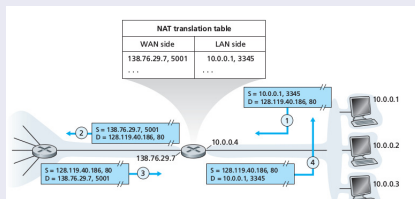
Roteador com NAT



Endereçamento IPv4

Roteador com NAT

- Roteador com NAT conecta-se à rede como se fosse apenas um dispositivo: ele “personifica” as requisições de sua subrede, dessa forma, escondendo da Internet que há atrás daquele IP uma subrede.
 - Na chegada das respostas, ele faz o encaminhamento adequado, pois trata o estado de roteamento: NAT translation table
- Mas, para isso há um roteamento executado acima da CAMADA 3!
- Essa é uma medida de contingência resultante da escassez de endereços IPv4: pode ser desconsiderado em IPv6
- Aplicações P2P sofrem com a presença de NAT: um host ligado através de um NAT nem sempre é capaz de responder a um TCP accept \implies connection reversal (P2P) ou relays de aplicação



Endereçamento IPv4

UPnP: Universal Plug and Play

- Problemas com segurança: roteador aceita comandos diretos de um host (NÃO através de interface de gerenciamento) para mapear uma porta externa ao IP/porta do host interno.

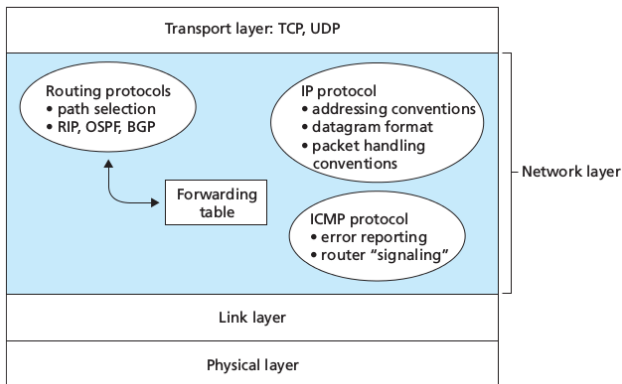


ICMP: Internet Control Message Protocol

Composição da camada de rede da Internet

- Protocolo IP
- Protocolos de roteamento: RIP, OSPF e BGP
- ICMP [RFC 792]

Visão em detalhes da camada de rede da Internet.



ICMP: Internet Control Message Protocol

Função básica: relatório de erro.

- Mensagem obtida em camada de aplicação “Destination network unreachable” vem do ICMP
 - Um roteador do caminho foi incapaz de encontrar a rota para o endereço de destino especificado pela aplicação.
 - Nesse ponto, o roteador envia uma mensagem ICMP type-3
- Mensagens ICMP trafegam como **payload** de datagramas IP Composição:
 - campos de tipo
 - campo de código
 - cabeçalho com os primeiros 8 bytes do datagrama IP que causou o erro.
- **ping**: ICMP tipo 8, código 0
 - Host que recebe a mensagem responde com ICMP tipo 0, código 0.
 - O ping “server” não é um processo: vai implementado no kernel do SO.



ICMP: Internet Control Message Protocol

Tabela de tipos de mensagens

ICMP Type	Code	Description
0	0	echo reply (to ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	6	destination network unknown
3	7	destination host unknown
4	0	source quench (congestion control)
8	0	echo request
9	0	router advertisement
10	0	router discovery
11	0	TTL expired
12	0	IP header bad

ICMP: Internet Control Message Protocol

Traceroute: implementado com mensagens ICMP

- Cria uma série de datagramas IP com segmentos arbitrários UDP com número de porta “improvável”
- Varia os valores de TTL dos datagramas e anota o horário de envio de cada um desses datagramas.
- Roteadores que recebem TTL 0 irão gerar uma mensagem ICMP tipo 11 código 0
 - Resposta indica o nome do roteador e o seu endereço IP
 - Emissor recebe a resposta e pode calcular o valor da RTT (sequência de três pacotes é enviada com o mesmo TTL para essa estimativa).
- Quando o host de destino recebe o datagrama UDP com uma porta fictícia, outra mensagem ICMP é gerada: tipo 3, código 3.
 - Nesse ponto, o traceroute pára de criar pacotes com TTLs cada vez maiores: já alcançou seu destino.



IPv6

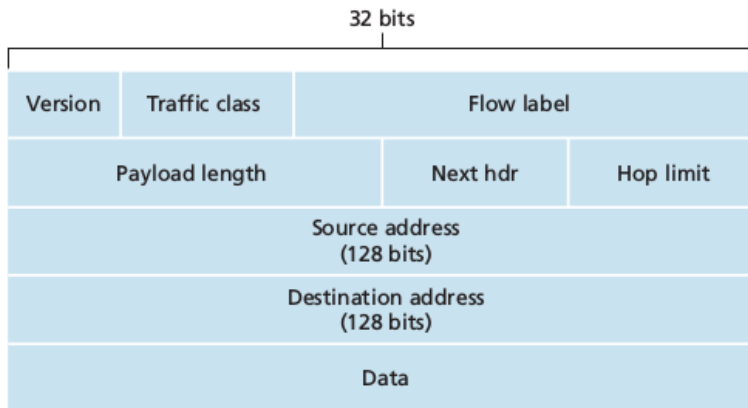
Iniciativa de padronização do começo da década de 1990, IETF.

- Contramedida em relação à escassez de endereços IPv4.
- Novo formato de datagrama IPv6.



IPv6

Formato de datagrama IPv6



IPv6

Iniciativa de padronização do começo da década de 1990, IETF.

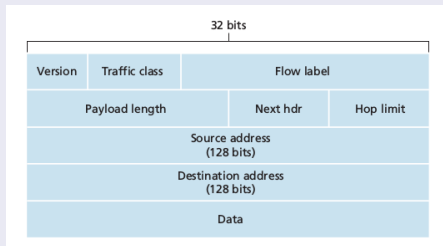
- Espaço de endereçamento aumentado
 - Endereços de IP com 128 bits: cada grão de areia do planeta pode ter seu próprio endereço de IP!
 - Endereço anycast.
- Cabeçalho com **tamanho fixo** de 40 bytes
- Rotulação de fluxo e prioridade
 - RFC 1752: definição de fluxo. Permite usar a marcação para diferenciar pacotes. **QoS**



IPv6

Novos campos

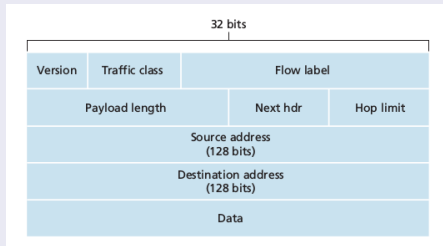
- Versão: 4 bits. (valor 6)
- Classe de tráfego: 8 bits. Similar ao TOS do IPv4
- Rótulo de fluxo: 20 bits, mascaramento de fluxos
- Comprimento de payload: 16 bits (unsigned integer) indicando o tamanho (em bytes) do payload carregado pelo datagrama



IPv6

Novos campos

- Next header: Indica a qual protocolo o payload do datagrama deverá ser entregue. (Campo protocolo do IPv4)
- Hop limit: função semelhante ao TTL. Evita que o pacote circule indefinidamente pela rede.
- Endereços de origem e destino: 128 bits!
- Data: payload



IPv6

Campos ausentes:

- **Fragmentação/remontagem** inexistente no IPv6. Mensagem “packet too big” sinaliza ao emissor a impossibilidade de um roteador IPv6 encaminhar um pacote. O emissor deverá fazer o controle do tamanho de pacote.
- Checksum de cabeçalho: camadas superiores realizam o checksum. Verificação foi **removida** da camada de rede implementada em IPv6
- Campo de Opções: **removido**. Trocado pelo campo “Próximo Cabeçalho”



IPv6

Há uma nova versão de ICMP para IPv6 [RFC 4443]

- Novos tipos e códigos adicionados.
- As funcionalidades do protocolo IGMP foram englobadas pelo IPv6



IPv6

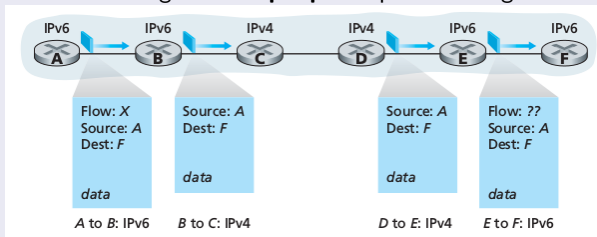
Migração IPv4 para IPv6

- Os sistemas que reconhecem IPv6 são capazes de cooperar com equipamentos IPv4
 - Mas, o que acontece quando um pacote IPv6 encontrar um roteador/equipamento IPv4 pelo caminho?
 - Que tal Flag day? \implies Não funciona!
 - RFC 4213: recomendações de um processo gradual
- **Dupla pilha:** equipamentos IPv6 implementam IPv4 e usam esse último quando reconhecerem que estão conversando com outro equipamento IPv4. Efeito colateral: dois endereços por interface do equipamento: IPv4 e IPv6 (apoio do DNS)
 - Downgrade IPv4, mesmo no modo híbrido.
 - Perdem-se algumas flags IPv6, pois se copiam apenas os payloads dos pacotes IPv6.



IPv6

Abordagem de **dupla pilha**: perda de flags



IPv6

Migração IPv4 para IPv6

- Tunelamento
 - Nesse cenário, encapsula-se um pacote IPv6, temporariamente, em um pacote IPv4.



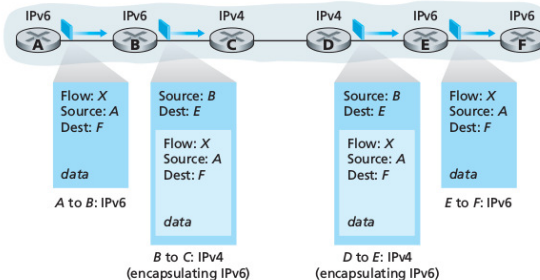
IPv6

Abordagem de tunelamento: overhead

Logical view



Physical view



IPv6

Segurança no protocolo IP

- Opções: VPN, IPsec
- IPsec:
 - 1 Estabelecimento de sessão IPsec entre hosts
 - 2 Segmentos TCP e UDP recebem serviços de segurança providos pelo IPsec
 - cifração, adição de campos de segurança e encapsulamento em datagrama IP comum
 - IPsec de destino decifra o segmento e passa o segmento em claro para a camada de transporte.



IPv6

Segurança no protocolo IP

- IPsec:
 - Serviços
 - Acordo quanto a primitivas criptográficas
 - Cifração de payloads dos datagramas
 - Garantia de integridade dos dados
 - Autenticação da origem

Roadwarrior



Conceitos

Tipicamente um host está conectado diretamente a um roteador, o roteador **default** (ou roteador de primeiro hop)

- Todas as vezes que um pacote precisa ser enviado à rede, ele é inicialmente transferido para o roteador padrão.
- Como levar o pacote entre o roteador origem e o roteador destino?



Conceitos

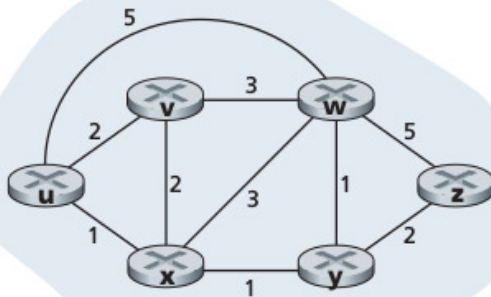
Função de um algoritmo de roteamento: dado um conjunto de roteadores interligados por enlaces, determinar a “melhor rota” entre o roteador fonte e o roteador destino

- Grafo: $G = (N, E)$, N é um conjunto de nós e E é uma coleção de arestas, onde cada aresta interliga um par de nós de N .
 - N : conjunto de roteadores
 - E : conjunto de enlaces



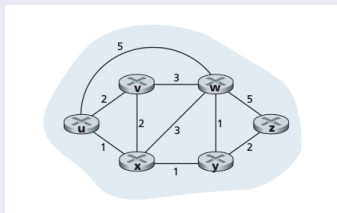
Conceitos

Modelo abstrato de rede: grafo.



Conceitos

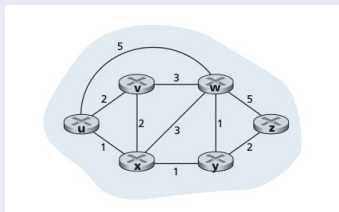
- Cada aresta possui um valor associado: **custo**.
 - Função de custo: comprimento físico do enlace, velocidade de transmissão ou custo financeiro de operação do enlace.
 - Para cada aresta (x, y) em E (coleção de enlaces/arestas), define-se custo $c(x, y)$ o custo da aresta entre os nós x e y .
 - Se $(x, y) \notin E \implies c(x, y) = \infty$
- Grafos sem direcionamento (não direcionados) $\implies c(x, y) = c(y, x)$
- x e y são vizinhos $\iff (x, y) \in E$



Conceitos

Algoritmo de roteamento deverá identificar as rotas de menor custo entre as origens e destinos das redes.

- Caminho ou rota: sequência de nós $x_1, x_2, x_3, \dots, x_p$ tal que $(x_1, x_2), (x_2, x_3), \dots, (x_{p-1}, x_p)$ pertencem a E .
 - Custo na rota: $c(x_1, x_2) + c(x_2, x_3) + \dots + c(x_{p-1}, x_p)$
 - Várias rotas entre x e y , mas há uma **rota de menor custo**.
- Rota de menor custo entre u e w : (u, x, y, w)
- Para enlaces (arestas) com mesmo custo, a menor (e melhor) rota será aquela que percorre o menor número de enlaces (ou menor número de HOPs).



Conceitos

Classificação de algoritmos de roteamento

- **Algoritmo de roteamento global**

- Conhecimento integral da rede
- Execução em duas etapas: de coleta de topologia e de cálculo
- Cálculo centralizado ou distribuído
- Algoritmos do tipo LS (Link State)



Conceitos

Classificação de algoritmos de roteamento

- **Algoritmo de roteamento descentralizado**

- Cálculo da rota de forma descentralizada
- Nenhum nó tem conhecimento integral da topologia/estado da rede
- Cada nó inicia o processo com estado sobre os custos dos enlaces aos quais está conectado diretamente
- Processo iterativo de cálculo e troca de mensagens leva à descoberta das rotas de menor custo
- Algoritmos do tipo DV (Distance Vector)



Conceitos

Classificação de algoritmos de roteamento

- **Algoritmos de roteamento estático:** rotas que mudam muito lentamente com o passar do tempo
- **Algoritmos de roteamento dinâmico**
 - Alteram as rotas de acordo com o tráfego da rede
 - Mais suscetíveis a loops e oscilações nas rotas
- **Algoritmos sensíveis ao carregamento da rede:** sensibilizam-se quanto ao congestionamento da malha

Os algoritmos da Internet (modo geral) são insensíveis ao congestionamento: o custo de um enlace não se altera em função do congestionamento.



Algoritmo de roteamento LS

Premissas:

- Cada host deverá difundir pacotes com informações sobre os estados dos enlaces para todos os outros nós que compõem a rede.
- Cada um desses pacotes conterá identificação e custos dos enlaces associados
- OSPF: link-state broadcast
- Com a difusão, todos os nós constituintes da rede têm conhecimento integral do estado da rede
- Cada nó, dessa forma, é capaz de calcular os mesmos custos, pois usam o mesmo algoritmo e os mesmos estados. Roteamento **global**, cálculo **descentralizado**.



Algoritmo de roteamento LS

Algoritmo de Dijkstra

- Calcula a rota de menor custo de um nó (origem, u) para qualquer outro nó da rede
- Algoritmo iterativo: Depois da k -ésima iteração, as rotas de menor custo para k nós de destino já foram determinadas.
 - Entre as rotas de menor custo para ligar os nós da rede, essas k rotas são as que possuem os menores custos.
- Notação
 - $D(v)$, custo da rota de menor custo do nó fonte para o destino v como saída da iteração do algoritmo;
 - $p(v)$, nó anterior da rota de menor custo do nó fonte para o destino v como saída da iteração do algoritmo;
 - N' , subconjunto de nós; v pertence a N' se a rota de menor custo da origem para v já for conhecida/calculada pelo algoritmo.



Algoritmo de roteamento LS

Algoritmo de Dijkstra

Initialization:

```
N' = {u}
for all nodes v
  if v is a neighbor of u
    then  $D(v) = c(u,v)$ 
  else  $D(v) = \infty$ 
```

Loop

```
find w not in N' such that D(w) is a minimum
add w to N'
update D(v) for each neighbor v of w and not in N':
   $D(v) = \min( D(v), D(w) + c(w,v) )$ 
/* new cost to v is either old cost to v or known
least path cost to w plus cost from w to v */
until N' = N
```

Algoritmo de roteamento LS

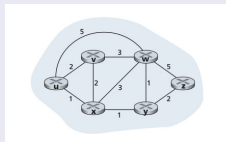
Algoritmo de Dijkstra

- **Inicialização:**

- As rotas de menor custo de u para seus vizinhos v , x e w são inicializadas com 2, 1 e 5.
- Embora o custo para w seja definido inicialmente com 5, há uma rota de custo menor que interliga u a w .
- Custos para y e z são definidos ∞ , pois não possuem ligação direta a u .

- **Iteração 1:**

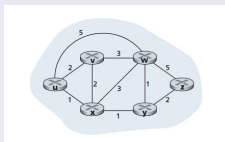
- Conferem-se os nós que não foram ainda adicionados ao conjunto N' .
- Identifica-se o nó com menor custo que foi resultado da última iteração $\Rightarrow x$, com custo 1.
- Adiciona-se x a N' .
- Na linha 12 do algoritmo LS, realiza-se a atualização $D(v)$ de todos os nós v . Conferir segunda linha da tabela de evolução do algoritmo.



Algoritmo de roteamento LS

Evolução de estados

step	N'	$D(v), p(v)$	$D(w), p(w)$	$D(x), p(x)$	$D(y), p(y)$	$D(z), p(z)$
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



Algoritmo de roteamento LS

Algoritmo de Dijkstra

- **Iteração 1** (continuação):

- O custo do caminho para v não foi alterado. Por sua vez, o custo para w modifica-se, pois passa-se a considerar um caminho de menor curso através de x , que resulta em custo 4 \implies Uma rota de menor curso é selecionada e o antecessor ao longo da menor rota partindo de u é ajustado para x .
- Similarmente, o custo para y , através de x , é calculado como sendo 2 \implies Atualiza-se a tabela em decorrência desse novo valor de custo.

- **Iteração 2:**

- Na segunda iteração, os nós v e y possuem uma rota de menor custo de intensidade 2. \implies Adicionamos y ao conjunto N' que agora contém u , x e y .
- O custo dos nós restantes ainda não inseridos em N' , os nós v , w e z , são atualizados de acordo com a linha 12 do algoritmo LS. Isso gera a terceira linha da tabela de evolução dos estados.

- **Iteração $n+1$:**



Algoritmo de roteamento LS

Algoritmo de Dijkstra

- Ao fim do algoritmo LS, nós temos, para cada nó, um predecessor ao longo da rota de menor custo partindo do nó origem:
 - Para cada predecessor, temos outro predecessor
 - Constrói-se o caminho completo da fonte a todos os destinos
 - A tabela de encaminhamento é montada pelo armazenamento (pares) de próximos passos da rota de menor curso de u ao seu destino
 - Complexidade $O(n^2)$, $(n+1)n/2$



Algoritmo de roteamento LS

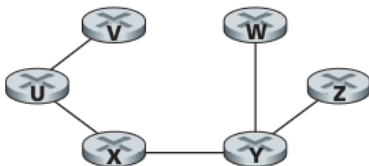
Evolução de estados para a rede da Figura anterior.

<i>step</i>	<i>N'</i>	$D(v),p(v)$	$D(w),p(w)$	$D(x),p(x)$	$D(y),p(y)$	$D(z),p(z)$
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					

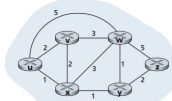


Algoritmo de roteamento LS

Tabela de rotas e topologia com ligações de menor custo.



Destination	Link
v	(u, v)
w	(u, x)
x	(u, x)
y	(u, x)
z	(u, x)



Algoritmo de roteamento LS

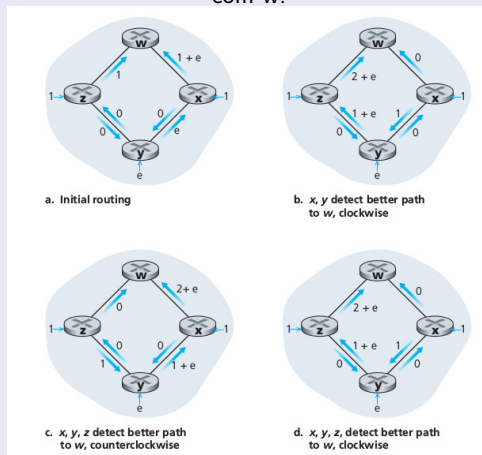
Algoritmo de Dijkstra

- Casos patológicos: Não se recomenda a execução do mesmo algoritmo LS em todos os nós do sistema.
- Oscilações são efeito colateral de esquema de roteamento sensível ao congestionamento.



Algoritmo de roteamento LS

Oscilações derivadas de roteamento sensível ao congestionamento: z, x e y comunicam com w.



Algoritmo DV (Distance Vector)

Definições

- Algoritmo iterativo, assíncrono e distribuído
 - Distribuído: cada nó recebe informações de um ou mais de seus vizinhos diretamente conectados, executa cálculos e distribui os resultados dos cálculos para os vizinhos
 - Iterativo: continua até que mensagens contendo informações pertinentes ao algoritmo de roteamento deixam de fluir.
 - Assíncrono: não demanda que todos os nós calculem simultaneamente as rotas.



Algoritmo DV (Distance Vector)

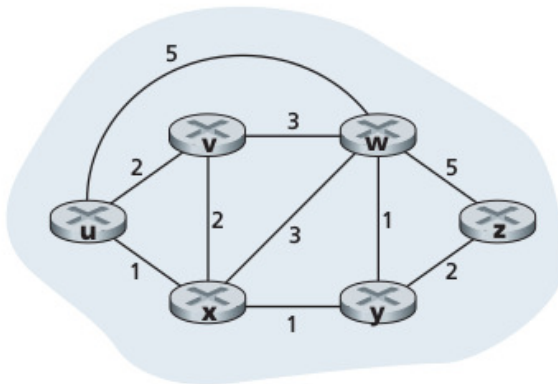
Termos

- $d_x(y)$ é o custo da rota de menor custo entre os nós x e y
- Equação de Bellman-Ford: $d_x(y) = \min_v \{c(x, v) + d_v(y)\}$, onde a minimização é realizada entre todos os vizinhos de v .



Algoritmo DV (Distance Vector)

Modelo abstrato de rede: grafo.



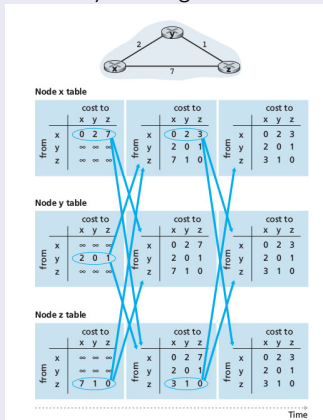
Algoritmo de roteamento LS

Algoritmo DV (Distance Vector)

```
1 Initialization:
2   for all destinations y in N:
3      $D_x(y) = c(x,y)$  /* if y is not a neighbor then  $c(x,y) = \infty$  */
4   for each neighbor w
5      $D_w(y) = ?$  for all destinations y in N
6   for each neighbor w
7     send distance vector  $D_x = [D_x(y) : y \text{ in } N]$  to w
8
9 loop
10  wait (until I see a link cost change to some neighbor w or
11    until I receive a distance vector from some neighbor w)
12
13  for each y in N:
14     $D_x(y) = \min_v \{c(x,v) + D_v(y)\}$ 
15
16  if  $D_x(y)$  changed for any destination y
17    send distance vector  $D_x = [D_x(y) : y \text{ in } N]$  to all neighbors
18
19 forever
```

Algoritmo de roteamento LS

Evolução do algoritmo DV.



Comparação entre LS e DV

- Em DV, cada nó conversa apenas com os vizinhos diretamente conectados, mas provê a seus vizinhos as estimativas de rotas de menor custo partindo dele para todos os outros nós (de que tem conhecimento) da rede.
- Em LS, cada nó conversa com TODOS os outros nós (difusão), mas avisa à eles apenas os custos de seus enlaces de ligação direta com vizinhos.
- Complexidade de mensagens
 - LS: $O(|N||E|)$ \implies Quando o custo de um enlace é modificado, o novo custo deve ser difundido para todos os outros nós
 - DV necessita de troca de mensagens entre vizinhos interligados a cada iteração \implies Uma mensagem de mudança de custo é gerada apenas se uma alteração de custo provocar impactos na determinação da rota de menor custo



Comparação entre LS e DV

- Velocidade de convergência
 - LS: $O(|N|^2)$ e $O(|N||E|)$ mensagens.
 - DV, convergência lenta e pode provocar loops de roteamento enquanto o algoritmo está convergindo. \Rightarrow Pode sofrer do problema contagem ao infinito
- Robustez/Resiliência
 - LS, há uma certa resiliência quanto a falha de roteadores (descarte/corrupção da mensagem de difusão)
 - DV poderá difundir um cálculo incorreto de rota por toda a rede



Roteamento Hierárquico

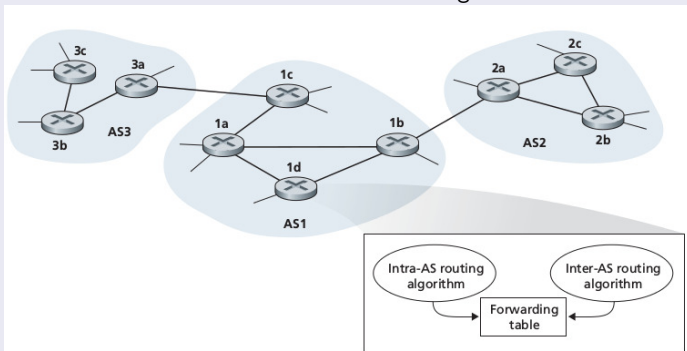
O cenário estudado no cálculo de rotas de menor custo é um tanto inadequado para o mundo real

- Escala: na medida em que o número de roteadores cresce, a latência gasta no cálculo, armazenamento e comunicação de informação de roteamento torna-se proibitivo
 - LS: há envolvimento de todos os nós!
 - DV: nunca convergirá a uma solução.
- Autonomia administrativa: alguém cuida da infra de rede. Dividir para conquistar!
- AS: sistemas autônomos.
 - Pertencem a um ISP e cooperam administrativamente
 - Protocolo de Roteamento Intra-Sistema Autônomo
 - Gateways: equipamentos de borda



Roteamento Hierárquico

Sistemas autônomos interligados



Roteamento Hierárquico

Comunicações entre ASs da Internet

- BGP4
- Hot-potato: se é possível sair por mais de um Gateway dentro de um aglomerado AS para alcançar determinado destino, o roteador interno escolherá enviar o pacote para o Gateway cuja rota de ligação possuir menor custo



RIP: Roteamento Intra AS [RFCs 1058 e 2453]

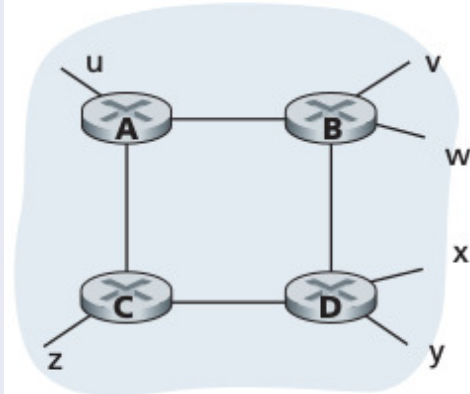
Introdução

- Protocolos Internos de Gateway \Rightarrow RIP (Routing Information Protocol) e OPSF (Open Shortest Path First)
- RIP foi um dos primeiros protocolos Intra-AS
- Tipo DV
 - Métrica de custo: contagem de hops
 - Custos calculados entre origem e rede de destino
 - HOP: número de subredes atravessadas na rota de menor curso do roteador de origem à rede de destino



RIP: Roteamento Intra AS [RFCs 1058 e 2453]

Número de hops de um roteador origem A para várias subredes.



Destination	Hops
u	1
v	2
w	2
x	3
y	3
z	2

RIP: Roteamento Intra AS [RFCs 1058 e 2453]

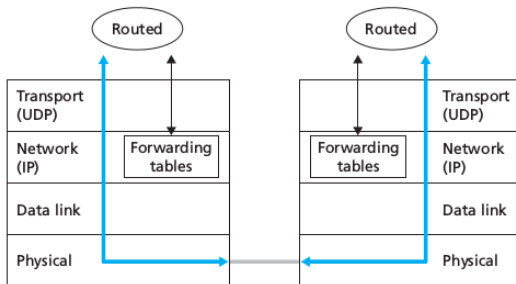
Introdução

- Tipo DV
 - O custo máximo de uma rota é limitado a 15: um AS terá diâmetro menor que 15 hops
 - Mensagens de atualização de roteamento são trocadas a cada 30 s entre vizinhos:
 - RIP response message: contém lista de até 25 redes de destino no AS, bem como a distância do emissor para cada uma dessas subredes
 - RIP advertisements
 - Vizinhos que não se “anunciam” depois de 180 s podem ser considerados desligados.
 - Mensagens fluem em datagramas UDP porta 520
- Protocolo executa em camada de **aplicação** sobre UDP



RIP: Roteamento Intra AS [RFCs 1058 e 2453]

Implementação de RIP por meio do daemon routed.



OSPF: Roteamento Intra AS [RFCs 2328]

Introdução

- Executado em ISPs de camada alta (upper-tier)
- Considerado sucessor do RIP
 - LS: flooding de mensagens de estado de enlace e algoritmo de Dijkstra para cálculo de rota de menor custo
 - Custos individuais de enlace são configurados pelo administrador do sistema
 - Um roteador difunde informações de roteamento para todos os outros roteadores no sistema autônomo
 - Isso acontece sempre que houver mudança no estado de enlaces
 - Estado de enlaces são anunciados a cada 30 min, em caso de estabilidade.



OSPF: Roteamento Intra AS [RFCs 2328]

Introdução

- Considerado sucessor do RIP
 - Características avançadas (em relação a RIP)
 - Segurança: trocas entre roteadores OSPF podem ser “autenticadas”
 - Caminhos múltiplos com mesmo valor de menor custo:
 - Suporte integrado para roteamento unicast e multicast
 - Suporte para hierarquia dentro de um único domínio de roteamento



Introdução

Os exemplos até então explorados envolveram comunicações ponto-a-ponto: um emissor e um receptor

Broadcast e Multicast

- broadcast: serviço de entrega com difusão. Um pacote é entregue a todos os componentes de uma rede.
- multicast: difusão com restrição. Um subconjunto é alvo.

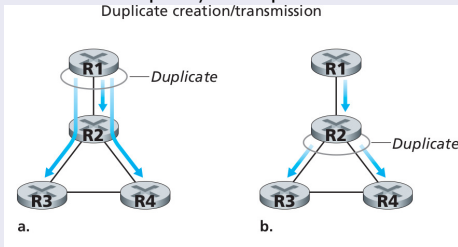


Algoritmos de roteamento broadcast

Primeira proposta

- nós despacha um lote com cópias do pacote de difusão
- N -way unicast: não necessita nem protocolo de roteamento da camada de rede, não há serviço de cópia de pacote ou serviço diferenciado de encaminhamento.
- Desvantagens: **ineficiência!** Melhor seria se nós da borda do primeiro hop notassem a difusão e somente a partir da borda da rede fosse feita uma multiplicação dos pacotes.

Duplicação de pacotes
Duplicate creation/transmission



Algoritmos de roteamento broadcast

Como um nó emissor tomará conhecimento de seus destinatários viáveis? (quem está na lista de recepção?)

- Mecanismos adicionais de protocolo: difusão de membresia.
- Protocolo inicialmente simples passa a crescer em complexidade
- Mecanismos de roteamento LS: dependem de difusão. Uma boa medida será “crescer um protocolo” em cima do unicast.



Algoritmos de roteamento broadcast

Abordagem mais refinada

- Grafos: $G = (N, E)$, N é um conjunto de nós e E é o conjunto de arestas

Flooding descontrolado

- Abordagem viável: um nó receptor recebe o pacote, mas encaminha uma cópia para todos os seus vizinhos (menos para o vizinho emissor do pacote)
- Com grafo conectado, a cópia alcançará todos os nós da rede
- Desvantagem: presença de ciclos no grafo provocará circulação perpétua de pacotes: inundação (flooding)
- **tempestade de broadcast!**



Algoritmos de roteamento broadcast

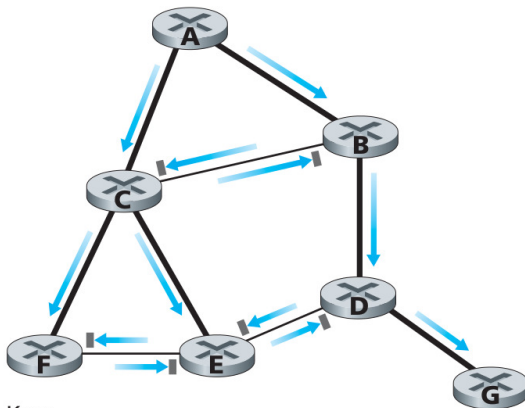
Flooding controlado

- Os nós da rede passam a analisar quando iniciar uma torrente.
- Flooding com controle de número de sequência
 - nó insere seu endereço (ou identificador único) e número de sequência de broadcast e encaminha pacotes aos vizinho
 - vizinhos fazem controle de pacote: recebido, duplicado, encaminhado. Todo pacote recebido é primeiro conferido na tabela de controle: Gnutella
- Encaminhamento de rota inversa (Reverse path forwarding)
 - roteador recebe pacote e o transmite por todos seus enlaces de saída (menos o enlace de chegada, naturalmente) se, se somente se, o pacote alcançou o roteador pelo enlace que está contido na rota unicast de menor custo para o emissor
 - OU, roteador simplesmente descarta o pacote
 - isso impede encaminhamento perpétuo e tempestade de broadcast.



Algoritmos de roteamento broadcast

Reverse Path Forwarding: difusão partindo de A



Key:

→ pkt will be forwarded

→■ pkt not forwarded beyond receiving router

Algoritmos de roteamento broadcast

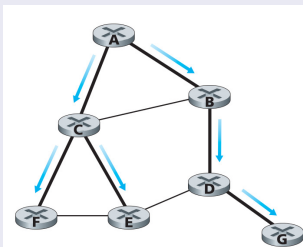
Spanning-Tree Broadcast

- As abordagens acima evitam situações indesejadas que poderiam quebrar a rede.
- Porém, ainda sofrem com ineficiências: há nós que poderão receber mais de um pacote!
- Figura: árvore de difusão permite que cada nó seja alcançado por apenas uma cópia do pacote
 - Visão do grafo em forma de árvore: $G = (N, E)$ é um grafo. $G' = (N, E')$, E' é subconjunto de E , G' é conectado porém não contém ciclos e, naturalmente, contém N .
 - Se cada aresta (enlace) possui seu respectivo custo e se o custo da árvore é a soma dos custos das arestas (enlaces), a spanning tree de custo mínimo obtida de um grafo G é a spanning tree mínima.

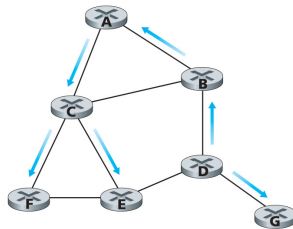


Algoritmos de roteamento broadcast

Árvore de difusão permite que cada nó seja alcançado por apenas uma cópia do pacote



a. Broadcast initiated at A



b. Broadcast initiated at D



Algoritmos de roteamento broadcast

Abordagem:

- Constrói-se a spanning tree
- Quando nó quer difundir pacote, envia pacote para todos os enlaces incidentes que pertencem à árvore de difusão.
- Nó da árvore encaminha cópia a todos os seus vizinhos (exceto ao emissor original, naturalmente)
- Problema: **complexidade**. Criação e manutenção da spanning tree.

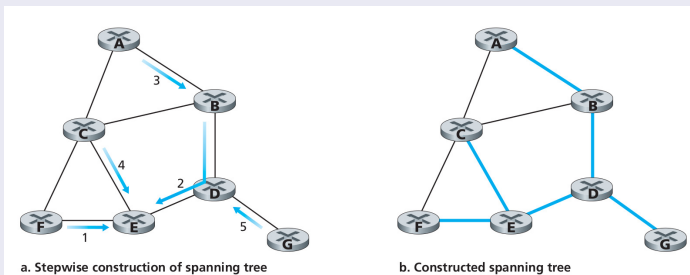
Abordagem center-based

- ponto central é definido (núcleo)
- nós vizinhos mandam mensagem de inclusão na árvore: encaminhada unicast até o nó núcleo
- ramo da árvore definido pelo caminho percorrido: é atualizado quando se tem conhecimento de nó que solicita inclusão na árvore



Algoritmos de roteamento broadcast

Abordagem center-based



Algoritmos de roteamento broadcast

Protocolos de difusão: na prática

- Gnutella: broadcast em camada de aplicação para difundir mensagens entre os peers
⇒ Enlace entre peers provido pelo TCP
- OSPF link-state advertisements e Intermediate-System-to-Intermediate-System (IS-IS)



Algoritmos de roteamento multicast

Conceito: um subconjunto restrito de nós da rede deve ser alcançado pela cópia do pacote

Cenário

- transferência de uma atualização de software para lote de PCs
- streaming
- aplicações de compartilhamento de dados
- feeds
- atualizações de caches de rede



Algoritmos de roteamento multicast

Problemas

- Quem serão os receptores?
- Como endereçar um pacote para que ele alcance um nó da rede multicast?
 - Em unicast, há destinatário explícito
 - Em broadcast, não é necessário indicar destinatário: são todos os componentes de uma rede
 - E em multicast? Ingenuamente: uma lista de endereços no pacote: OVERHEAD!



Algoritmos de roteamento multicast

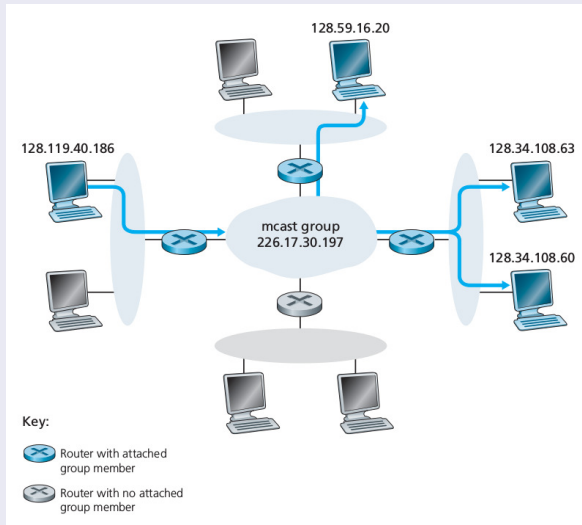
Endereçamento

- address indirection: identificador único usado pelo grupo de receptores. Cópia é entregue a todos os receptores associados àquele identificador.
- endereço IP classe D multicast. Cada host ainda possui seu respectivo endereço IP unicast



Algoritmos de roteamento multicast

Grupo multicast



Algoritmos de roteamento multicast

Perguntas práticas

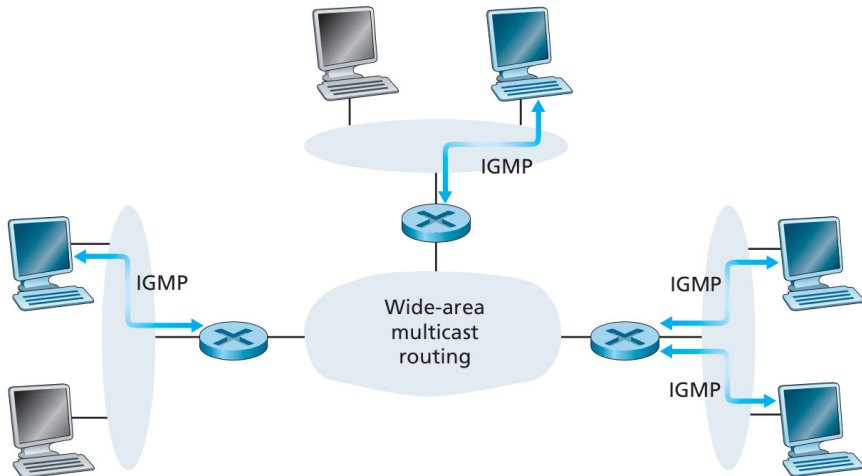
- Como um grupo é determinado e como ele é desfeito?
- Como é escolhido o endereço do grupo?
- Como novos hosts são adicionados ao grupo?
- A membresia do grupo é restrita?
- Membros do grupo sabem o endereço IP dos outros membros?
- Como é a atividade colaborativa do grupo multicast para que os pacotes fluam?

Resposta: **IGMP**.



Algoritmos de roteamento multicast

IGMP



Algoritmos de roteamento multicast

IGMP

- Opera entre o host e seu roteador mais próximo
- Host interage com seu roteador mais próximo e informa que deseja ingressar num grupo multicast específico
- Há protocolo de interação entre os roteadores envolvidos no encaminhamento dos pacotes entre os componentes do grupo



Algoritmos de roteamento multicast

IGMP: Arquitetura

- Mensagens IGMP: três tipos, encapsuladas dentro de datagrama IP (protocolo de número 2)
 - membership_query: roteador manda para determinar o conjunto de todos os grupos multicast que foram estabelecidos pelos hosts acessados através de determinada interface
 - membership_report: resposta dos hosts. Ou mensagem disparada por host quando a aplicação indica a sua inclusão em grupo multicast.
 - leave_group: saída (opcional). Roteador pode inferir a saída: falha em enviar os membership_report. Protocolo soft-state



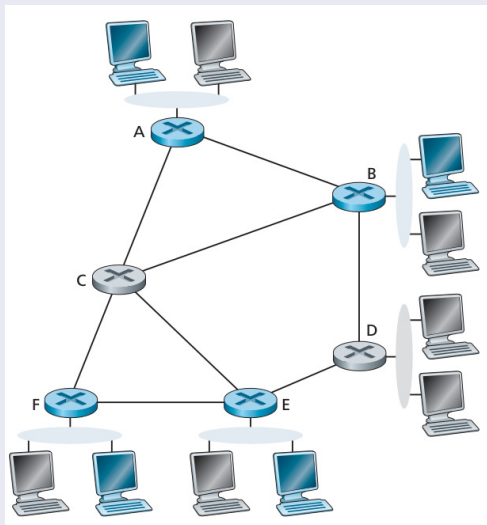
Algoritmos de roteamento multicast

Algoritmos de roteamento

- Objetivo de algoritmo: encontrar uma árvore de enlace que conecta todos os roteadores de próximos (dos hosts multicast) e de interligação
- Tipos
 - multicasting routing usando árvore de compartilhamento em grupo: árvore compartilhada de roteamento para rotear todos os pacotes de todos os emissores
 - multicasting routing usando árvore de com foco na fonte: árvore de roteamento individual para cada host emissor



Algoritmos de roteamento multicast



Algoritmos de roteamento multicast

