

INTERCONEXÃO DE REDES LOCAIS COM ROTEADORES

- ❖ LANs podem ser conectadas entre si com pontes (*bridges*)
- ❖ Por que não usar sempre pontes para construir redes grandes?
- ❖ Pontes implementam uma LAN única e a maioria das LANs oferece *broadcast* (a nível de enlace) como serviço
 - Não faz sentido oferecer *broadcast* numa escala qualquer (imagine toda a rede UFCG!, toda a Internet!!)
- ❖ Pontes usam enchente (*flooding*) em alguns algoritmos de encaminhamento de quadros
 - Enchente não é aplicável em grande escala
- ❖ No ponto onde broadcast e/ou flooding deixa de fazer sentido, outro tipo de equipamento se faz necessário: O ROTEADOR!

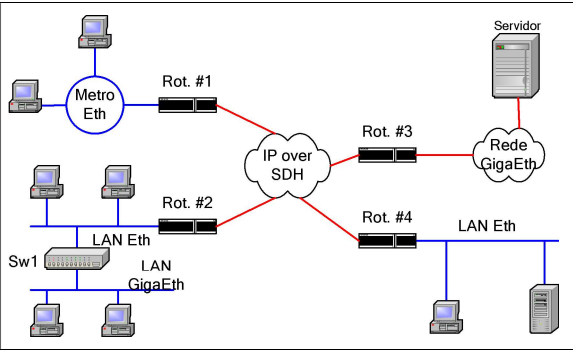
POR QUE NÃO SÓ PONTES?

- ❖ Pontes não fazem conversão de protocolos da camada 3 (só operam na camada 2)
 - Se duas redes com protocolos de rede diferentes precisam ser interconectadas, deve-se usar um equipamento que converta protocolos dessa camada: um roteador multiprotocolo
- ❖ Pontes roteiam baseadas no endereço físico de cada estação
 - Essa solução não tem escala para grandes redes pois as tabelas de roteamento das pontes seriam enormes
- ❖ Limitam-se as estações vistas pelas pontes usando-se outras técnicas de roteamento (roteamento baseado em endereço lógico de estação ou de grupo de estações)
 - Roteadores implementam essas técnicas

CARACTERÍSTICAS BÁSICAS DE UM ROTEADOR

- ❖ Opera na camada de rede (nível 3)
 - Processa o protocolo de rede (p.ex. IP, IPX, X25)
 - Usa informações presentes no cabeçalho do pacote (p.ex. endereço destino)
- ❖ Tem como função básica encaminhar pacotes encolhendo a interface de saída apropriada
 - Usa uma tabela de roteamento
- ❖ Não é transparente
 - Tem um endereço (de rede e de enlace)
 - Estações sabem da existência de roteadores e enviam pacotes para eles encaminharem
 - Encaminhamento nó-a-nó (*hop-by-hop*)

EXEMPLO DE UMA REDE COM PONTES E ROTEADORES



ALGORITMO BÁSICO DE ROTEAMENTO

- ❖ Ao receber um pacote, o roteador:
 - Examina o endereço destino
 - Consulta uma tabela de roteamento
 - Encaminha o pacote para a interface de saída adequada
- ❖ Para evitar crescimento explosivo da tabela de roteamento, o endereço destino é normalmente quebrado em 2 partes
 - Parte vista pelo algoritmo - endereço de rede
 - Parte não vista pelo algoritmo - endereço de estação
- ❖ A tabela de roteamento é indexada apenas com a primeira parte do endereço destino
 - Pode haver exceções onde um ou outro endereço completo seja colocado na tabela de roteamento

- ❖ Supondo que o endereço destino seja quebrado em (Rede, Estação), um algoritmo típico de roteamento seria:
 - Recebe pacote da camada de enlace
 - Extrai o endereço destino
 - Examina a parte Rede
 - Se estiver conectado diretamente a esta rede, mapeia o endereço destino (Rede, Estação) no correspondente endereço físico (MAC) se for tecnologia difusão, monta um quadro apropriado e encaminha o pacote pela interface adequada
 - Senão, Se o endereço destino (Rede, Estação) estiver na tabela de roteamento, Encaminha o pacote pela interface adequada
 - Senão, se a parte Rede estiver na tabela de roteamento, Encaminha o pacote pela interface adequada
 - Senão, encaminha o pacote para um roteador padrão (*default*), se houver; se não houver, acusa um erro de roteamento para o emissor do pacote

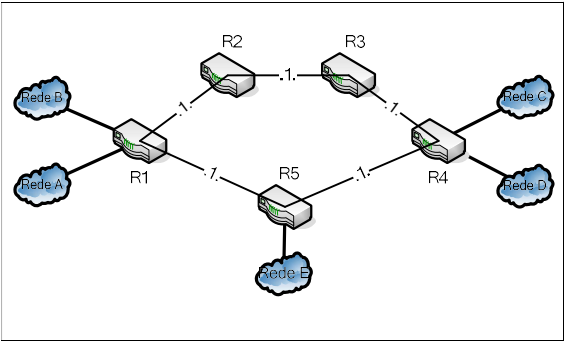
ATUALIZAÇÃO DE TABELAS DE ROTEAMENTO

- ❖ Tabela de roteamento pode ser criada/mantida de forma:
 - **Estática** (em redes pequenas; através de configuração manual feita pelo administrador da rede)
 - **Dinâmica** (em redes maiores ou onde haja rotas alternativas (anéis) para um mesmo destino, através de configuração automática feita por programas / protocolos)
- ❖ Vamos examinar duas técnicas para roteamento dinâmico
 - Vetor Distância (*Vector Distance - VD*)
 - Estado de Enlace (*Link State - LS*)
- ❖ Três algoritmos são usados na prática (p. ex. na Internet):
 - RIP (*Routing Information Protocol*) - VD
 - OSPF (*Open Shortest Path First*) - LS
 - BGP-4 (*Border Gateway Protocol* - 4)

ROTEAMENTO COM VETOR DE DISTÂNCIA

- ❖ Usado pelo protocolo RIP (ainda em uso na sua versão 2 – RIPv2)
- ❖ Cada roteador mantém uma tabela com a menor "distância" conhecida até cada rede destino e que conexão usar para chegar lá
- ❖ A métrica de distância pode ser
 - Número de nós (*hops*) ou enlaces a atravessar (mais comum)
 - Número de pacotes em fila até o destino
 - Atraso (em milisegundos)
- ❖ Cada roteador sabe a distância até cada vizinho
 - É simples descobrir isso, qualquer que seja a métrica usada
- ❖ Periodicamente, cada roteador envia sua tabela de roteamento completa para todos os seus vizinhos
 - Ao receber uma tabela de um vizinho, o roteador determina a melhor rota para cada destino possível (que ele já conhece ou está aprendendo agora) e atualiza sua tabela de roteamento

EXEMPLO DE ROTEAMENTO COM VETOR DE DISTÂNCIA



INTERCONEXÃO DE REDES DE COMPUTADORES

- ❖ Inicialmente (assim que é ligado), cada roteador conhece apenas as redes diretamente conectadas a ele
 - R1 só conhece a Rede A, a Rede B e as redes R1-R2 e R1-R5
 - R5 só conhece a Rede E, a rede R1-R5 e a rede R4-R5
 - E assim por diante
- ❖ Na sua primeira exportação de rotas, R1 envia sua tabela de rotas para R2 e R5

Rede Destino	Próximo Nó	Custo
A	R1-d	0 (zero)
B	R1-d	0
R1-R2	R1-d	0
R1-R5	R1-d	0
Obs. Rx-d indica diretamente conectado na rede		

- ❖ R2 e R5 recebem a tabela de R1 e, juntando com o que já conheciam, ficam com as tabelas de roteamento seguintes

Rede Destino	Próximo Nó	Custo
A	R1	1
B	R1	1
R1-R2	R2-d	0
R1-R5	R1	1
R2-R3	R2-d	0

Rede Destino	Próximo Nó	Custo
A	R1	1
B	R1	1
E	R5-d	0
R1-R2	R1	1
R1-R5	R5-d	0
R4-R5	R5-d	0

INTERCONEXÃO DE REDES DE COMPUTADORES

- ❖ Depois de decorrido o tempo necessário para a troca de tabelas de roteamento entre todos os roteadores (o chamado **tempo de convergência**), todos os roteadores aprendem como chegar a todas as redes
- ❖ A tabela de roteamento de R1, por exemplo, ficaria:

Rede Destino	Próximo Nó	Custo
A	R1-d	0
B	R1-d	0
C	R5	2
D	R5	2
E	R5	1
R1-R2	R1-d	0
R1-R5	R1-d	0
R2-R3	R2	1
R4-R5	R5	1
R3-R4	R2	2
R3-R4	R5	2

- ❖ Como ficariam as tabelas de roteamento dos outros roteadores?
- ❖ Como não leva em conta a largura de banda disponível e outras características dos enlaces, algoritmos mais modernos usam outra técnica:
 - *Link State Routing*
 - *Distance Vector*, porém, ainda é muito usado (muita gente usa RIPv2 na Internet)

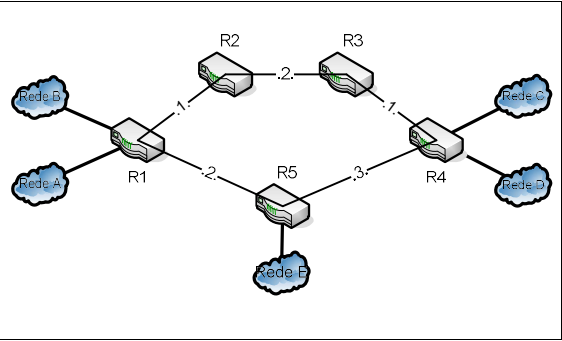
INTERCONEXÃO DE REDES DE COMPUTADORES

LINK STATE ROUTING

- ❖ Usado pelo OSPF, por exemplo
 - Uso recomendado pelo IETF para a Internet
- ❖ Essencialmente, a toda a topologia e todos os custos em cada enlace são tornados conhecidos para todos os roteadores (na primeira vez; depois disso, somente as alterações percebidas em cada roteador são enviadas para os roteadores)
- ❖ Cada roteador pode então determinar a melhor rota para cada destino

LINK STATE ROUTING - ALGORITMO

- ❖ **PASSO 1:** Descobrir os roteadores vizinhos
 - Quando um roteador entra no ar, envia um pacote *Hello* em cada linha de saída para descobrir os vizinhos
 - No caso de uma LAN, usa *broadcast* ou *multicast*
- ❖ **PASSO 2:** Medir/calcular o custo (p. ex., atraso) em cada linha
- ❖ **PASSO 3:** Construir pacotes de anúncio de estado de linha (*Link State Advertisement - LSA*)
 - Contém informação sobre todos os vizinhos, com respectivos custos



Roteador	Rede	Custo para a Rede ou do Enlace
R1	A	0
	B	0
	R1-R2	1
R2	R1-R2	1
	R2-R3	2
	R2-R5	2
R3	R3-R2	2
	R3-R4	1
	R3-R5	3
R4	C	0
	D	0
	R4-R3	1
R5	E	0
	R5-R1	2
	R5-R4	3

LSAs criados pelos roteadores (inicialmente)

- ❖ **PASSO 4:** Distribuir os LSAs
 - São enviados inicialmente e posteriormente quando algo muda no entorno de um roteador (custo dos enlaces)
 - Usa *flooding*
 - Algoritmos especiais são usados para evitar explosão de LSAs (usando números de sequência - pacote com sequência igual ou inferior ao atual é descartado, idade dos pacotes, etc.)
 - Algoritmos especiais são usados para evitar inconsistências dado que cada roteador executa o algoritmo com informações diferentes dos demais
 - Para evitar geração de *loops*, máquinas não alcançáveis, etc.
- ❖ **PASSO 5:** Calcula as novas rotas
 - Usa algoritmo de Dijkstra para calcular os caminhos mais curtos até os outros roteadores
 - O melhor caminho entre um roteador Ri e um roteador Rj acaba indicando o melhor caminho entre o roteador Ri e todas as redes diretamente conectadas ao roteador Rj
 - Os caminhos mais curtos são registrados na tabela de roteamento e o roteador volta à operação normal

❖ A tabela de roteamento de R1, por exemplo, ficaria:

Rede Destino	Próximo Nó	Custo
A	R1-d	0
B	R1-d	0
C	R2	4
D	R2	4
E	R5	2
R1-R2	R1-d	0
R1-R5	R1-d	0
R2-R3	R2	1
R3-R4	R2	3
R4-R5	R5	2

❖ Como ficariam as tabelas de roteamento dos outros roteadores?

- ❖ Observação importante: Na arquitetura TCP/IP
 - Rede Destino sempre é indicada pelo par Endereço de Rede e Máscara de Rede
 - Próximo Nó sempre é indicado pelo endereço IP do próximo nó

LINK STATE ROUTING - Conclusão

- ❖ O algoritmo tem vários problemas (contornáveis)
 - Roteador que diz que tem uma linha, mas não tem
 - Roteador que esquece de informar uma linha que tem
 - Roteador que não encaminha os pacotes
 - Roteador que corrompe os pacotes antes de enviar
 - Roteador que calcula erradamente
 - Roteador que não tem memória suficiente para calcular rotas
- ❖ Tais problemas podem ocorrer quando a rede tem dezenas de milhares de roteadores!
- ❖ O que normalmente não acontece!!

CARACTERÍSTICAS AVANÇADAS DE ROTEADORES

- ❖ Uso de enlace alternativo em situação de pico de tráfego
 - *Bandwidth on Demand*
- ❖ Priorização de tráfego
 - IP (TCP, UDP, SMTP, TELNET, FTP), IPX, ...
- ❖ *Firewall* simples e QoS simples
 - Filtro de pacote por endereço de rede, por endereço de transporte, por tipo de protocolo de transporte
 - Controle de banda por endereço de rede, transporte, protocolo
 - Auto-proteção para ataques Denial of Service (DOS)
- ❖ Roteador atuando também como ponte
 - Com ou sem propagação de *broadcast*
- ❖ Compressão de dados
- ❖ Balanceamento de carga (*Load Balancing*) por interface, protocolo
- ❖ Troca a quente de componentes (*Hot Swapping*)
- ❖ Multiprotocolo (conversão de protocolo)

FATORES NA ESCOLHA DE UM ROTEADOR

- ❖ Protocolos de rede suportados
- ❖ Capacidade e tipo de interfaces para LAN (metálica, óptica, Gbic, SFP - Ethernet, Fast Eth, Giga Eth, etc.)
- ❖ Capacidade e tipo de interfaces WAN (E1, E2, STM-1, etc. – metálica, óptica)
- ❖ Desempenho de roteamento
 - Poucos milhares a vários milhões sw pacotes por segundo(p.ex. 20.000 a 10.000.000)
- ❖ Funcionalidade
 - Gerência
 - Bridging
 - Compressão
 - Filtragem
- ❖ Custo
 - US\$ 1.000 A US\$ 100.000

ONDE COLOCAR ROTEADORES?

- ❖ Entre 2 segmentos de LAN ou para se conectar a uma WAN
- ❖ Temos 2 alternativas para organização da rede:
 - Rede com roteadores espalhados
 - Rede com roteadores centralizados (*Collapsed Backbone*) com topologia em estrela única ou múltipla
- ❖ Razões para *Collapsed backbone*
 - Servidores estão voltando a ser centralizados para diminuir o Custo Total de Propriedade - CTP (*Total Cost of Ownership* - TCO)
 - Suporte, manutenção, teste de cabeamento, etc., tudo é mais fácil/barato
 - Seria a volta do CPD (Centro de Processamento de Dados) - do Aquário?
 - Os grupos de trabalho (workgroups) não aguentaram dar suporte aos servidores das redes locais
 - Criam-se depósitos/fazendas de servidores (*server farms*)
 - Todos os segmentos das LANs têm de passar pelo ponto central
 - Dado que os servidores estão nesse local