



Laboratório de Pesquisa em Redes e Multimídia

Equipamentos de Interconexão

Hubs, Pontes e Switches

Prof. José Gonçalves

Departamento de Informática – UFES

zgonc@inf.ufes.br



Universidade Federal do Espírito Santo
Departamento de Informática

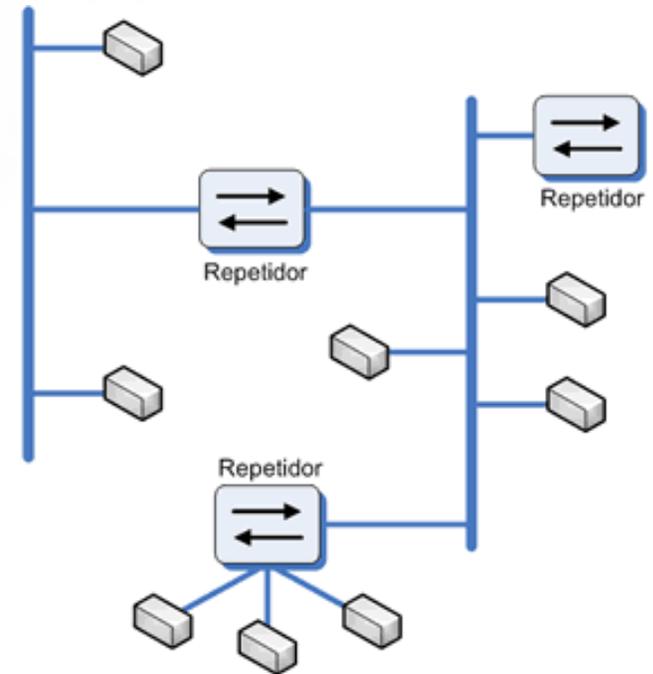
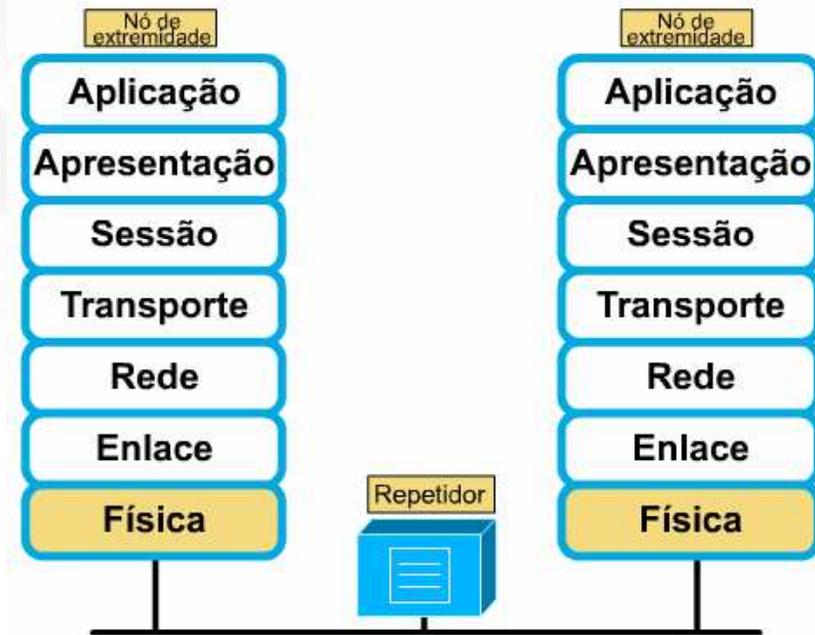
Interconexão de Redes



Repetidor

- Um repetidor (“*repeater*”) é um dispositivo utilizado para estender o alcance de uma LAN.
- Atuando na camada física, o repetidor conecta segmentos de rede, amplificando e transmitindo os sinais elétricos que ocorrem em um segmento para o(s) outro(s).
- Caso haja uma colisão ou interferência elétrica em um dos segmentos, ela se propagará para todos os outros segmentos.
- Não é possível estender os segmentos indefinidamente devido aos atrasos de propagação aceitáveis.
 - Regra 5-4-3 (máximo de cinco segmentos ligados por quatro repetidores, com no máximo três segmentos populados).

Repetidor (cont.)

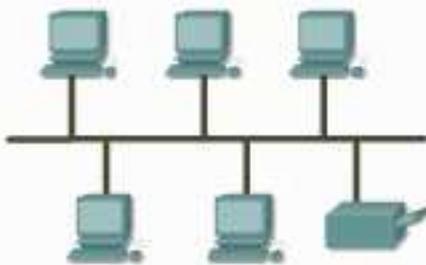


Domínio de Colisão

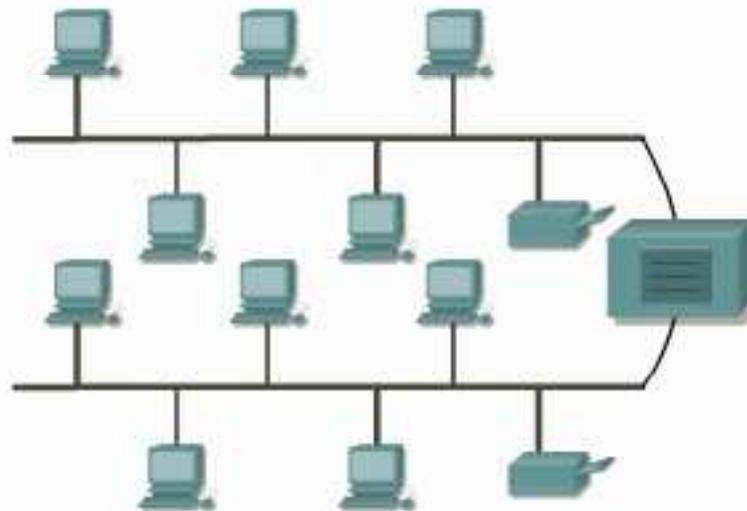
- É uma área lógica onde pacotes podem *colidir* uns contra os outros. Um domínio de colisão pode existir num único segmento da rede, numa porção ou na totalidade de uma rede.
- As estações disputam entre si o meio de transmissão dentro de um domínio de colisão. Uma colisão de pacotes atinge todas as estações do domínio.
- Pacotes *broadcasts* chegam a todas as estações dentro de um domínio de colisão.
- Domínios de colisão devem ser distribuídos de acordo com as características de tráfego (é uma opção de projeto).

Repetidor x Domínio de Colisão

O Acesso Compartilhado é um Domínio de Colisão



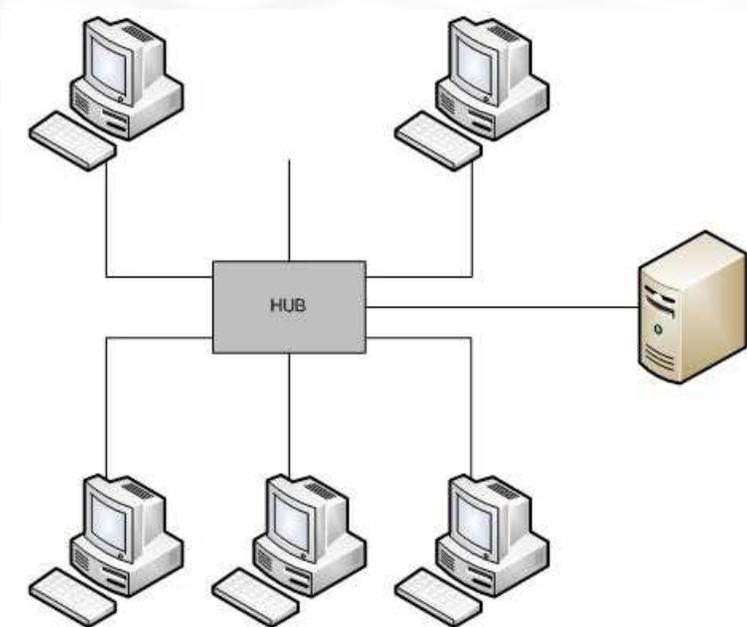
Domínio de Colisão - Estendido pelo Repetidor



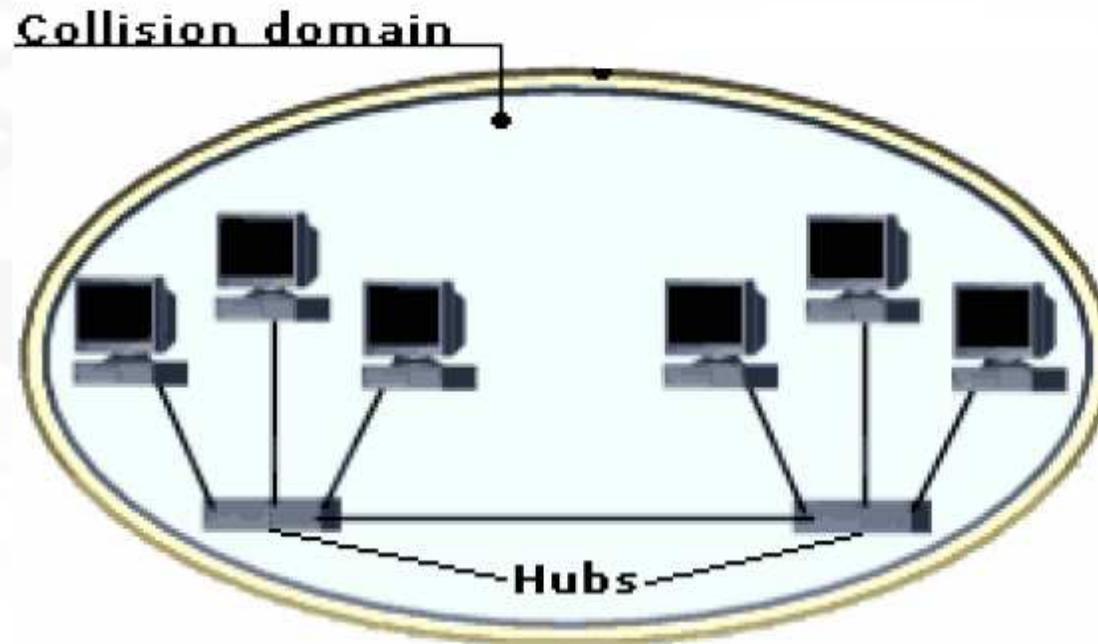
Hub (Network Hub, Repeater Hub)

- Um *hub* é um dispositivo que age como centralizador de conexões de rede. Assim como os repetidores, atuam na camada física do modelo OSI. É uma forma de repetidor multiporta.
- O *hub* permite interconectar múltiplos dispositivos de uma mesma LAN (p.ex. estações *ethernet*, via par trançado ou fibra óptica), agindo como se eles estivessem em um único segmento/barramento de rede.
- Todas as portas de um hub compartilham o mesmo segmento, normalmente com velocidades de 10 ou 100Mb/s. As transmissões são feitas em modo *half-duplex*.
- Quando um pacote chega a uma porta ele é retransmitido para todas as outras portas do *hub*, fazendo com que todas as outras estações também recebam o pacote.
- *Hubs* também participam da detecção de colisão de sinais, propagando um sinal de *jam* para todas as portas ao ser detectada uma colisão.

Hub (cont.)



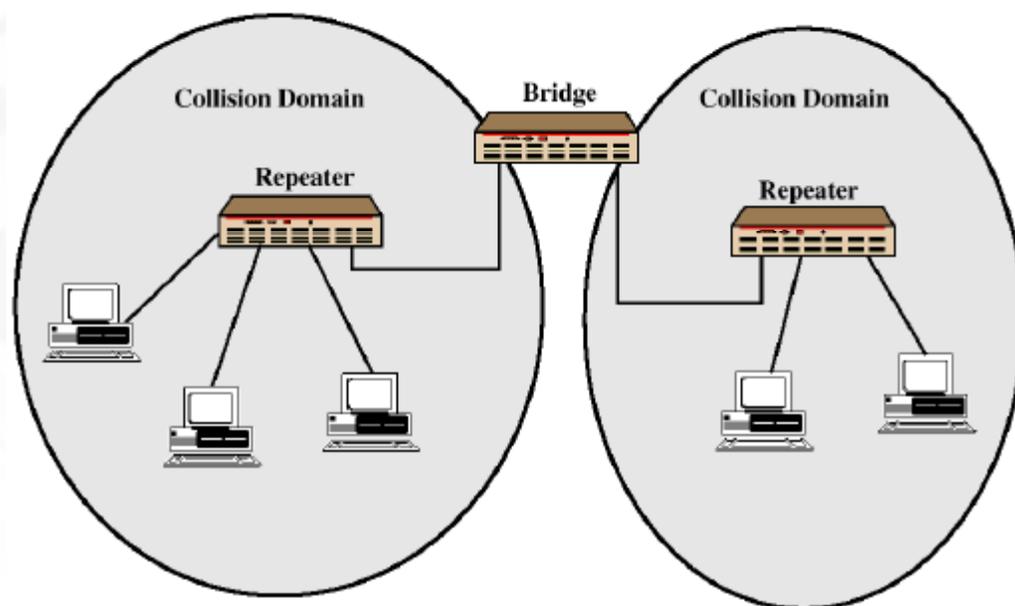
Hub x Domínio de Colisão



Pontes (*"Bridges"*)

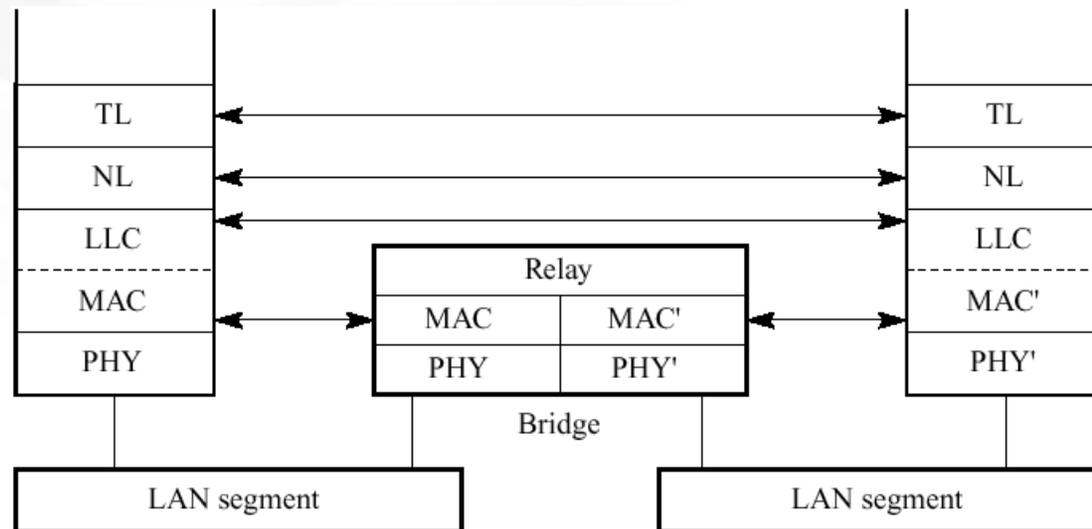
- Numa rede *Ethernet* o meio de transmissão é compartilhado e só um nó pode transmitir a cada vez.
- O aumento do número de nós aumenta a procura pela largura de banda disponível, aumentando a probabilidade de ocorrência de colisões.
- Uma solução é segmentar a rede utilizando para isso o recurso de pontes (*bridges*).
- Uma ponte divide um domínio de colisão em domínios de colisão de menor dimensão. Cada interface da ponte determina um domínio de colisão.
- Em resumo, pontes não propagam as colisões, criando vários "domínios de colisão" independentes.

Pontes (cont.)



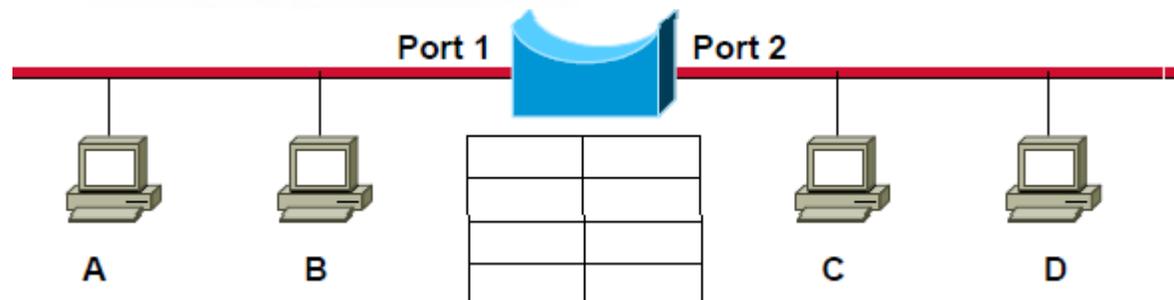
Pontes (cont.)

- Diferentemente dos repetidores e *hubs*, as pontes atuam na camada 2 do modelo OSI.



Pontes (cont.)

- Uma ponte conecta segmentos de uma mesma rede (ou conecta redes com diferentes tecnologias de enlace) e age como um gerente de tráfego:
 - Se o tráfego é destinado ao outro segmento a ponte permite a sua passagem;
 - Se o tráfego é local, o tráfego não é repassado desnecessariamente.



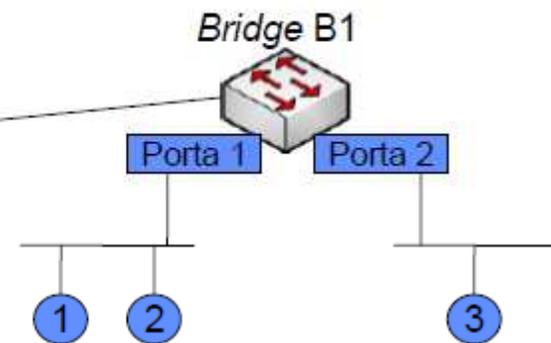
Pontes (cont.)

- A ponte toma a decisão de repassar ou não o tráfego baseada no endereço MAC (endereço físico) das estações.
- Todas as decisões são baseados neste endereço de camada de enlace, não afetando o endereçamento lógico da camada de rede.
- A ponte mantém uma lista das estações ativas em cada segmento e usa esta lista para direcionar o tráfego de uma parte à outra.
- Esta lista é mantida em uma tabela de endereços MAC e portas associadas. A ponte encaminha ou descarta pacotes com base nas entradas desta tabela.

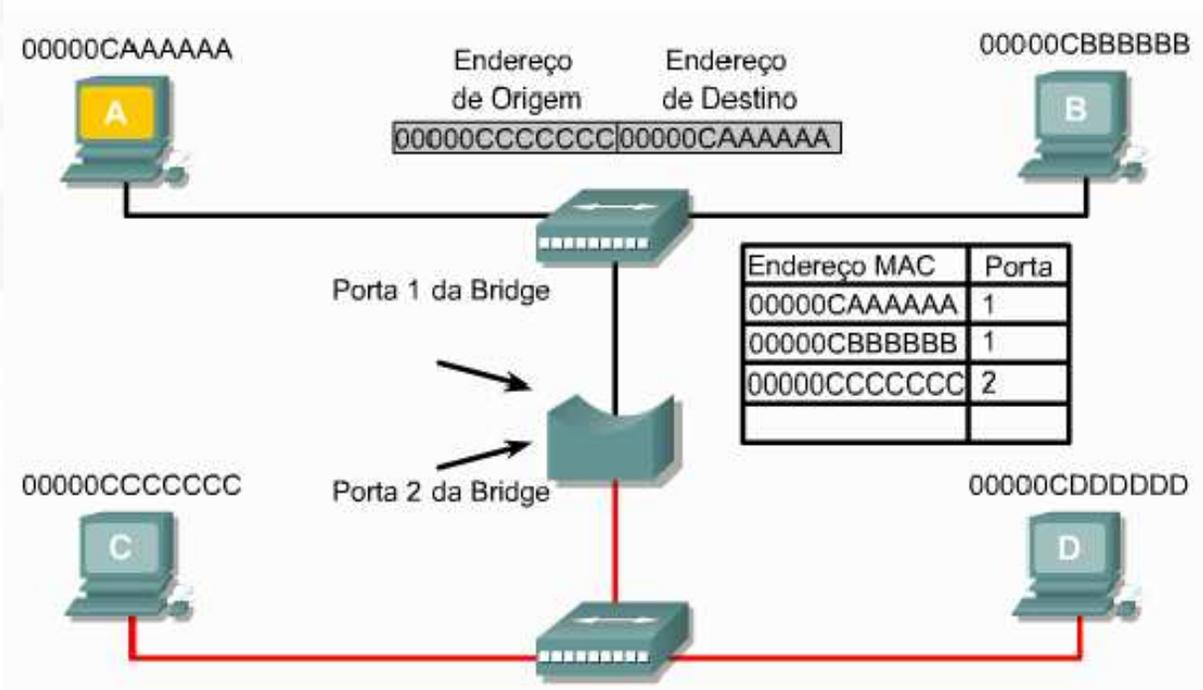
Pontes (cont.)

Tabela de Encaminhamento da *Bridge B1*
(*Forward Database – FDB*)

Endereço	Porta	Tempo
1	1	-
2	1	-
3	2	-

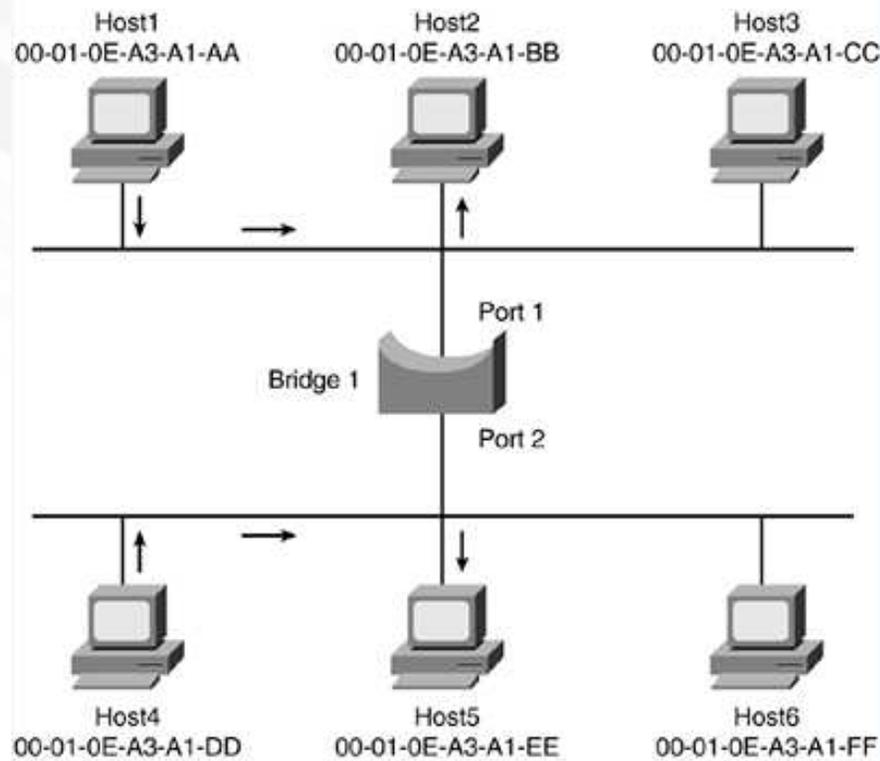


Pontes (cont.)



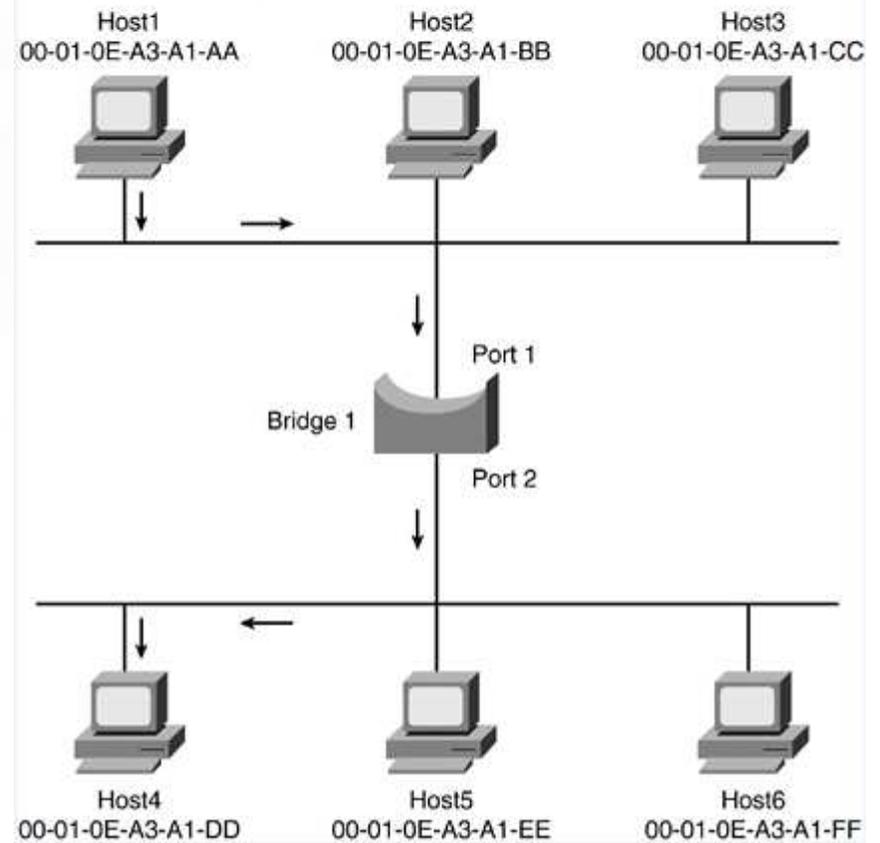
Pontes (cont.)

Host1 Communicating to Host2



Host4 Communicating to Host5

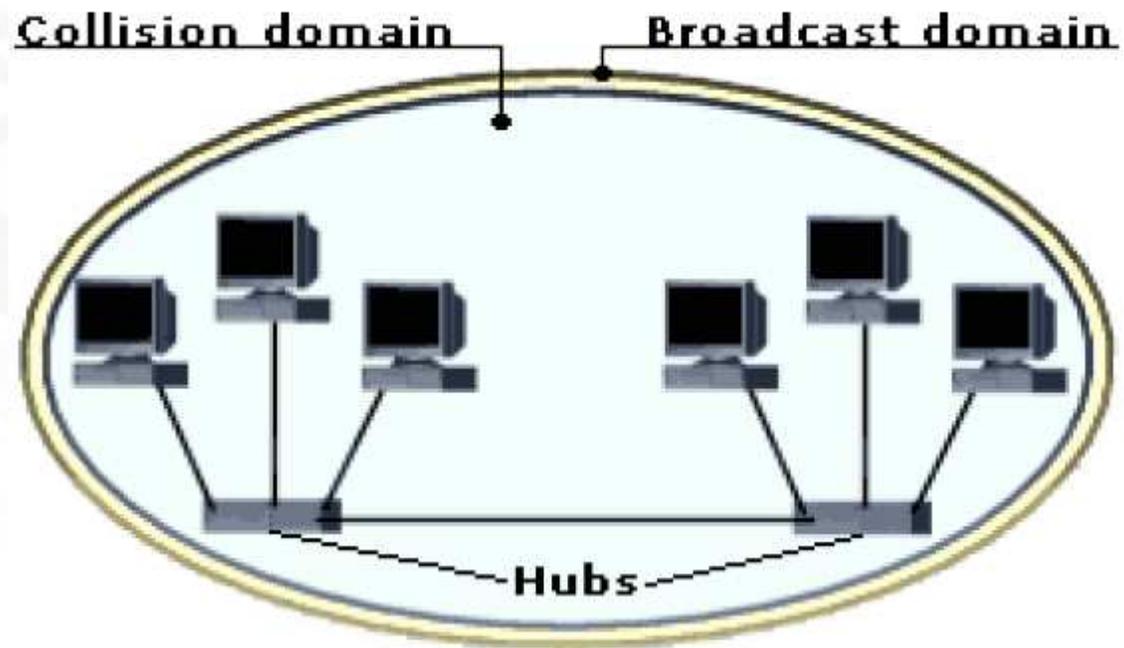
Host1 Communicating to Host4



Domínio de Broadcast

- Um *domínio de broadcast* é um segmento lógico de uma rede de computadores em que um computador ou qualquer outro dispositivo conectado à rede é capaz de se comunicar com outro sem a necessidade de utilizar um dispositivo de roteamento.

Hub x Domínio de Broadcast x Domínio de Colisão

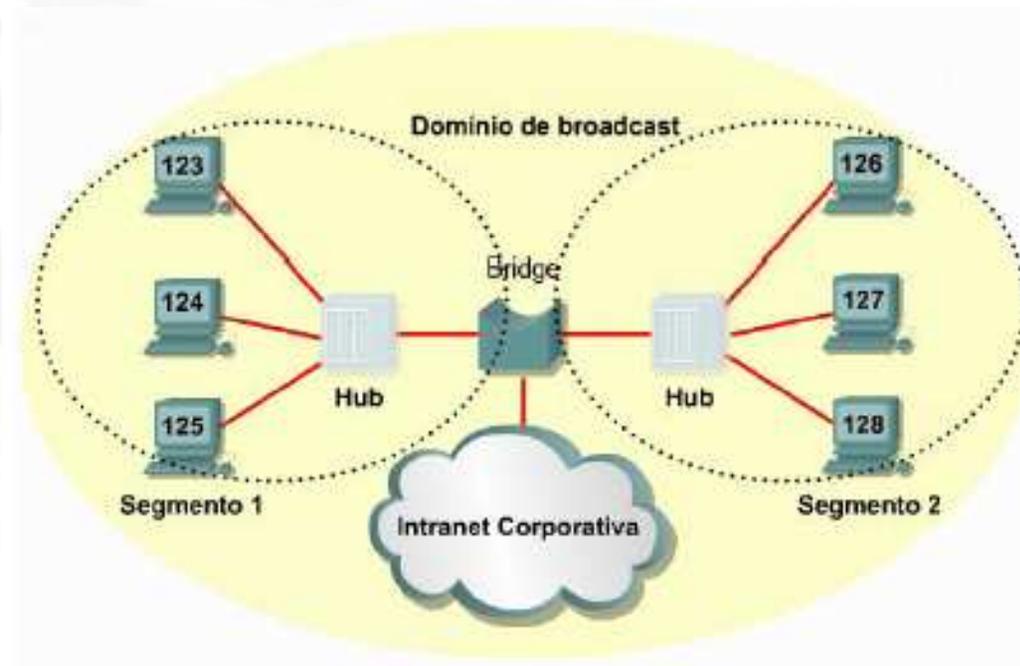


Pontes x Domínio de Broadcast

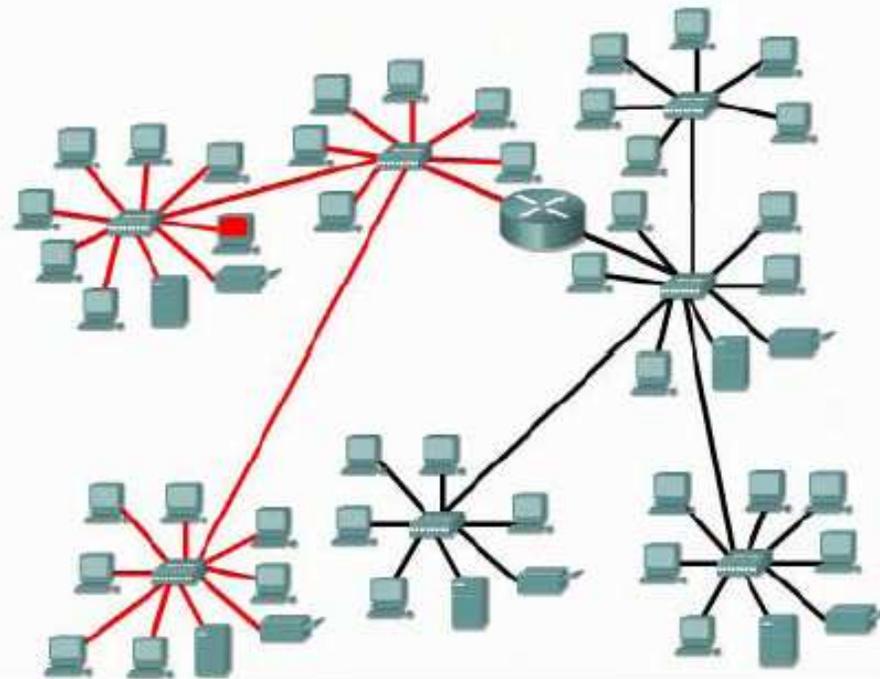
- Pontes não têm qualquer efeito no domínio de *broadcast* (isto é, pacotes de *broadcast* atravessam as pontes).
- Não importa quantas pontes existam em uma rede. A não ser que haja um dispositivo como um roteador, que funciona com endereçamento de camada 3, a rede inteira compartilhará o mesmo espaço de endereço lógico de *broadcast*.
- Em resumo, uma ponte cria mais domínios de colisão mas não adiciona domínios de *broadcast*.

Pontes x Domínio de Broadcast x Domínio de Colisão

(cont.)



Pontes x Domínio de Broadcast x Domínio de Colisão (cont.)



Com a utilização de um roteador em lugar de um dispositivo de bridging um broadcast de camada dois pode ser contido. Os dispositivos da camada três são os únicos dispositivos que podem conter broadcasts.

Características Adicionais das Pontes

- Permitem conversão de tipos de interfaces.
- Acomodam mudanças de máquinas.
 - Temporização associada às posições na tabela.
- Transmitem os *broadcasts* recebidos para todas interfaces:
 - Separa domínios de colisão mas não separa domínios de *broadcast*.
 - Estendem o alcance das redes (regra 5-4-3): 5 segmentos de distância um do outro, 4 bridges, 3 segmentos com estações.

Pontes Transparentes (cont.)

- Como visto, redes são segmentadas visando prover maior largura de banda por usuário. Pontes fornecem essa maior banda para os usuários ao reduzirem o número de estações competindo pela largura de banda do segmento.
- Devido ao método de operação padronizado para as pontes, uma rede segmentada através destes dispositivos se apresenta como uma única rede lógica (isto é, no nível IP, nível de rede) já que a única separação de tráfego é feita apenas no nível MAC.
 - Conceito de "*ponte transparente*" (padrão IEEE 802.1d): as estações não sabem da existência da ponte e acham que todas as outras estações estão no mesmo segmento (transparência).
 - Redes locais interligadas por pontes transparentes não sofrem nenhuma modificação ao serem interconectadas por estes equipamentos, que são "transparentes" para os nodos da rede.

Algoritmo

Learning / Aging

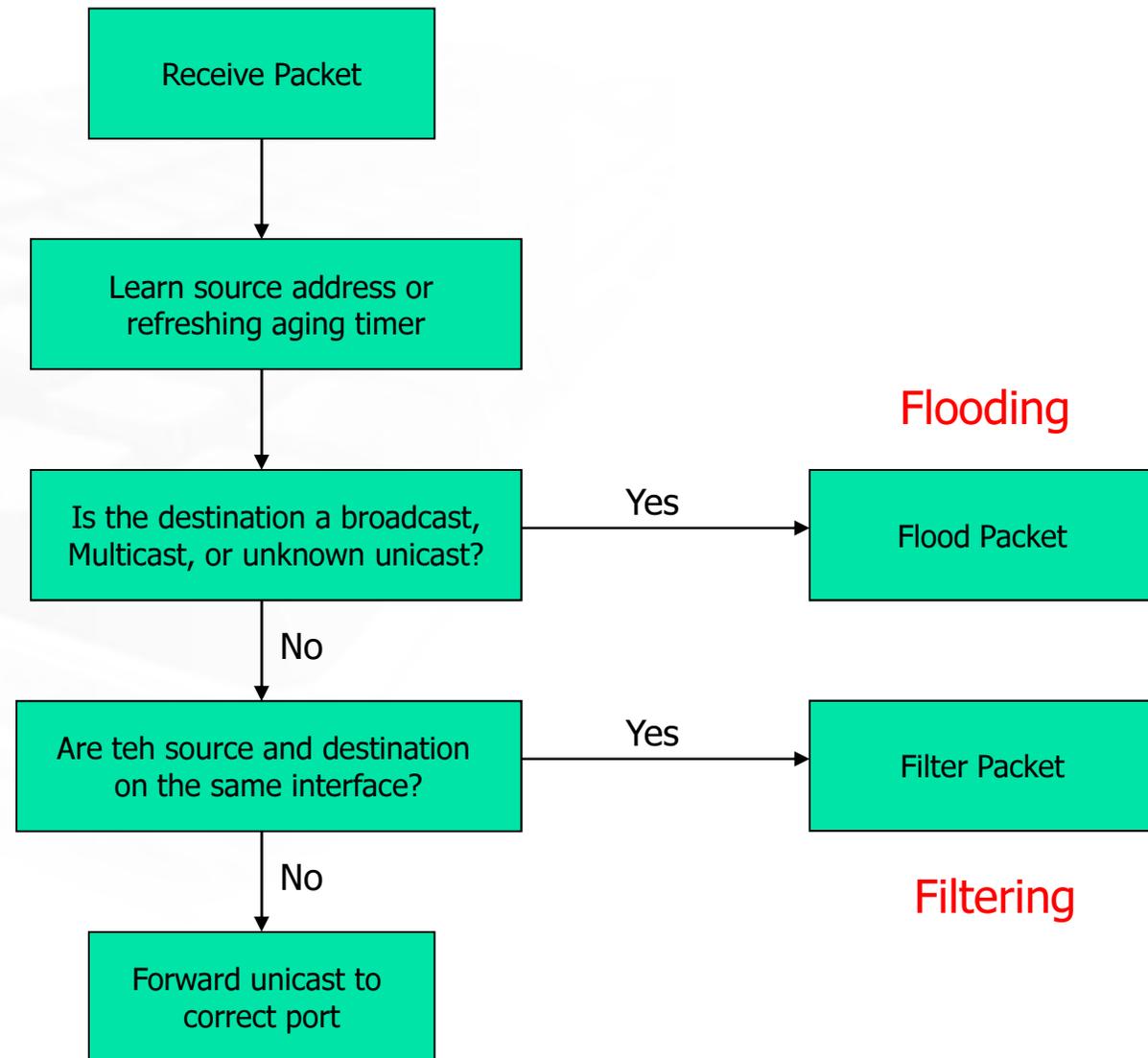
Flooding

Flood Packet

Filter Packet

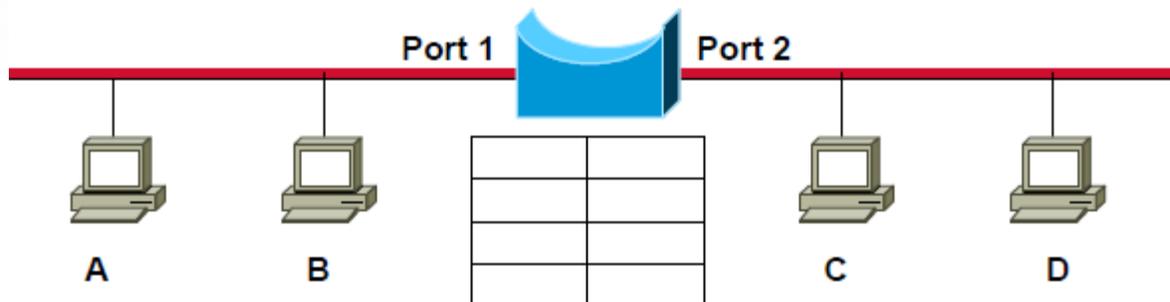
Filtering

Forwarding



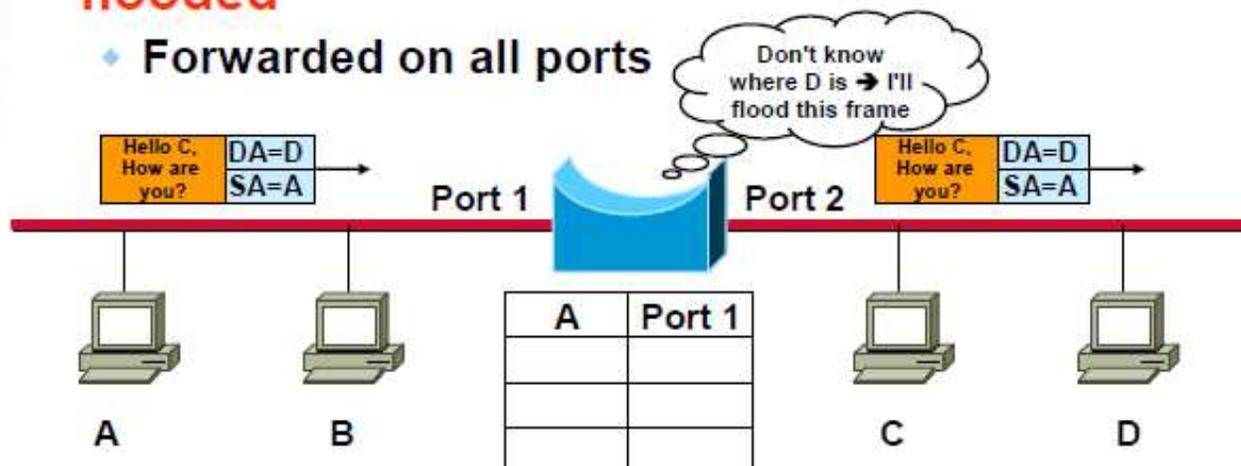
Operação de uma Bridge

- Transparent bridging is like "plug & play"
- Upon startup a bridge knows nothing
- Bridge is in **learning mode**



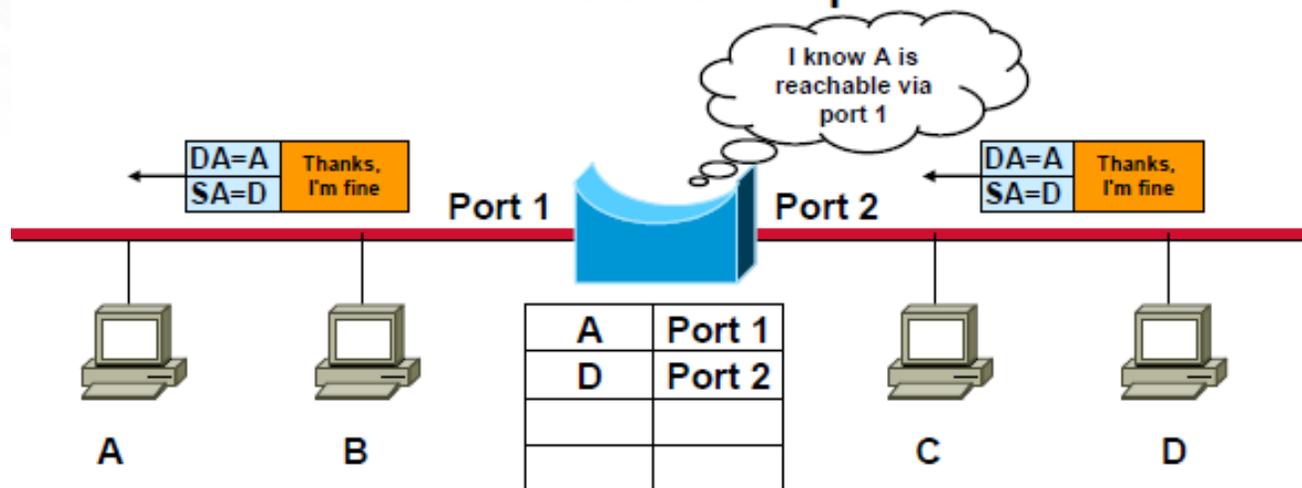
Operação de uma Bridge (cont.)

- Once stations send frames the bridge notices the **source** MAC address
 - ♦ Entered in bridging table
- Frames for unknown destinations are **flooded**
 - ♦ Forwarded on all ports



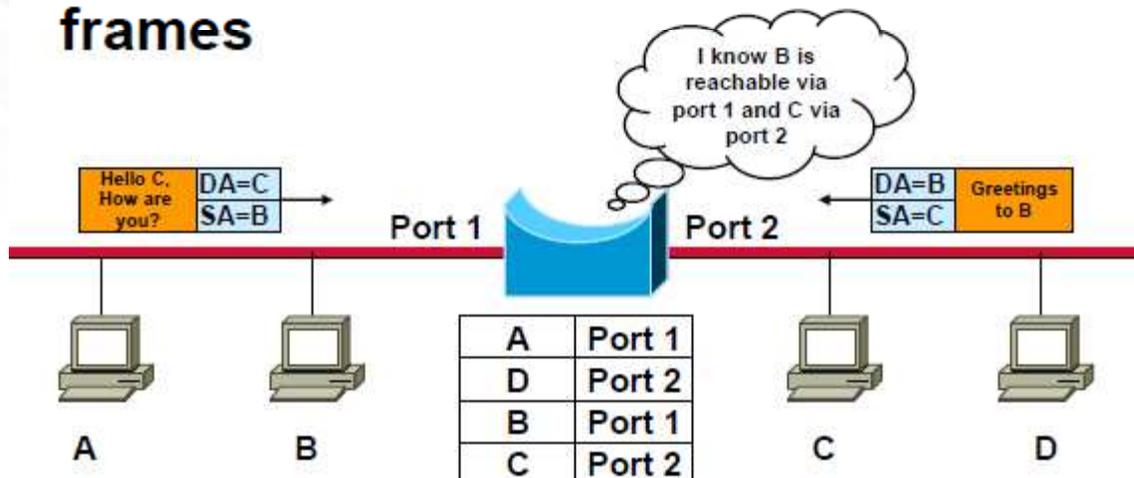
Operação de uma Bridge (cont.)

- If the destination address matches a bridging table entry, this frame can be actively
 - ♦ **forwarded** if reachable via other port
 - ♦ **filtered** if reachable on same port



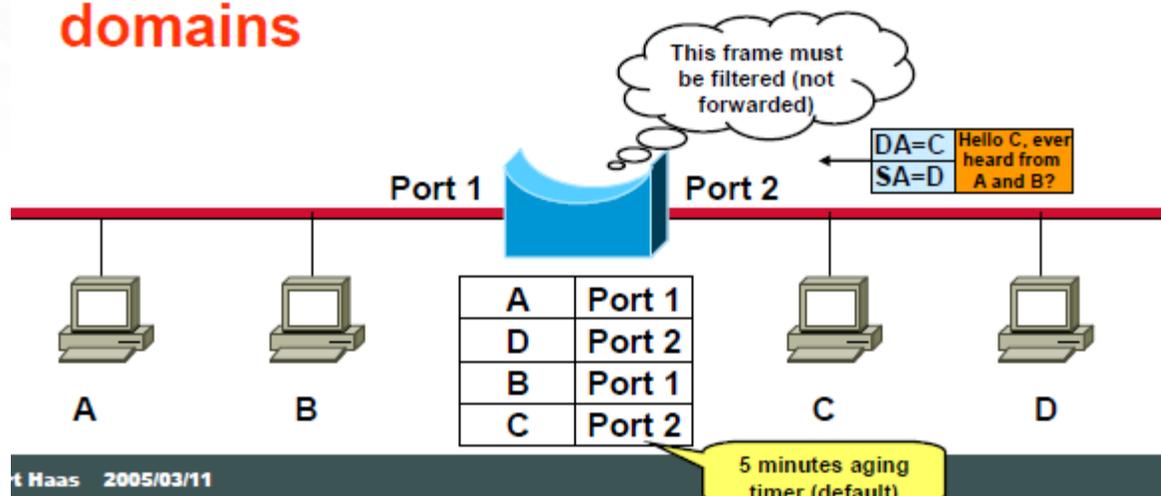
Operação de uma Bridge (cont.)

- After some time the location of every station is known – simply by listening!
- Now only **forwarding** and **filtering** of frames



Operação de uma Bridge (cont.)

- Frames whose source and destination address are reachable over the same bridge port are filtered
- LAN separated into **two collision domains**



Switches

- Um *switch* é, essencialmente, uma ponte mais complexa, com múltiplas interfaces ("*multiport bridge*"). É um dispositivo de interconexão que comuta (encaminha) quadros entre segmentos de uma LAN de acordo com o endereço MAC das estações. Assim como as *bridges*, operam na camada 2 do modelo RM-OSI.
- OBS: um LAN Switch tem um comportamento bem diferente de um WAN Switch. Este último é orientado a conexão e definitivamente não é transparente aos nós da rede.



Switches (cont.)

- Assim como a ponte, o *switch* segmenta a rede internamente.
- A cada porta corresponde um segmento diferente, o que significa que não há colisões entre pacotes de segmentos diferentes — ao contrário dos *hubs*, cujas portas partilham o mesmo domínio de colisão.
- O *switch* permite que estações de trabalho sejam conectadas diretamente as suas portas individuais, possibilitando a cada um dos dispositivos experimentarem o máximo da largura de banda disponível.

Switches (cont.)

- Como cada interface (porta) do *switch* define um domínio de colisão separado, as estações podem transmitir ao mesmo tempo. Por exemplo, pode-se ter duas sessões de comunicação concorrentes, uma entre as portas 1 e 2 e a outra entre as portas 7 e 8 do *switch*.
- Normalmente, *switches* possuem portas 10/100 Mbps mas também oferecem interfaces Gigabit. São, portanto, uma alternativa mais barata do que mudar todas as interfaces de rede para alternativas de maior velocidade.
- Como numa bridge, um *switch* aprende quais são os endereços MAC associados a cada uma das suas interfaces e só encaminha quadros para mais de uma porta quando o endereço de destino é desconhecido. Caso a origem e o destino estejam no mesmo segmento o *switch* bloqueia a passagem do quadro.

Switches (cont.)

- Os *switches* possuem tabelas internas (chamadas de *source address tables* - SAT) que armazenam os endereços MAC "conhecidos" e sua correspondente porta de origem. Estes são endereços das estações de trabalhos, de *hubs* "inteligentes", de outros *switches*, *bridges* ou roteadores.
- Sempre que chega um quadro cujo endereço MAC não consta na SAT é necessário que este quadro seja enviado a todas as portas do *switch*. Esta ação acentua drasticamente o tráfego na rede, e pode provocar um número considerável de colisões. Uma vez que a estação de destino responde à transmissão, seu endereço MAC é "aprendido" e armazenado nas SAT.

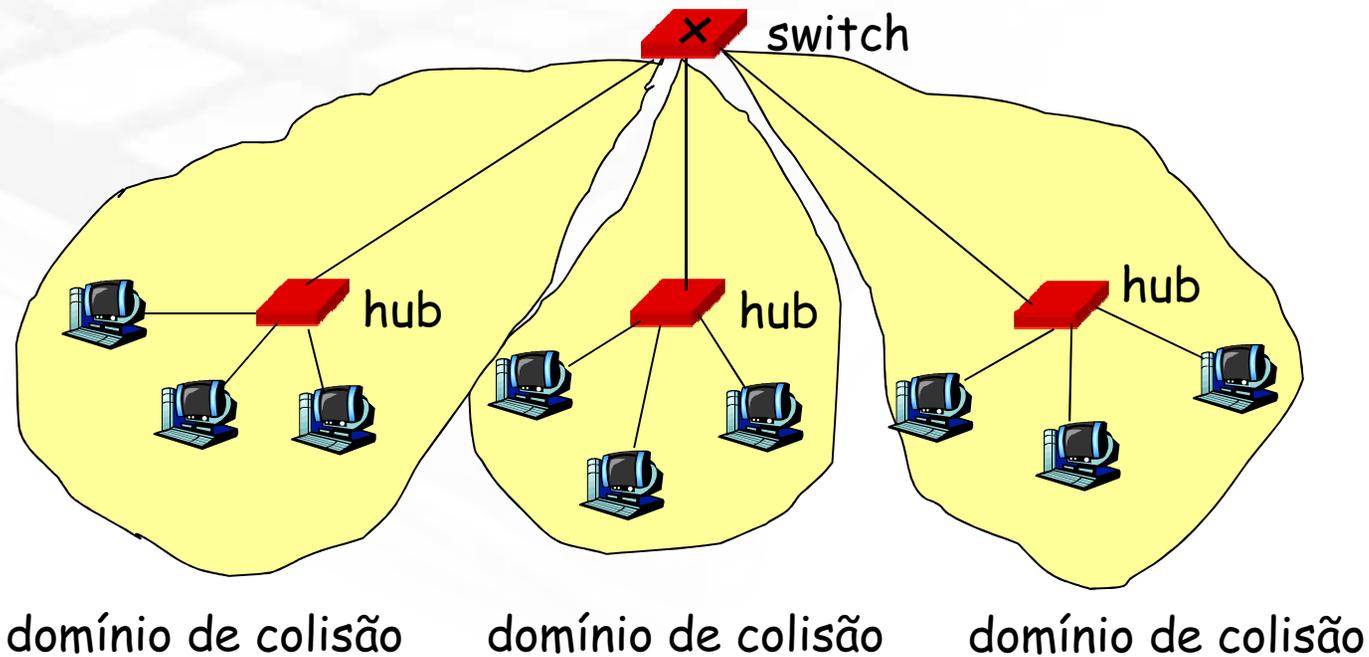
Switches (cont.)

- Se as tabelas dos *switches* possuírem uma capacidade de aprendizagem de endereços MAC inferior ao número de dispositivos da rede, é possível que estas já estejam cheias. Neste caso uma das entradas da SAT deverá ser descartada para a armazenagem do novo endereço aprendido.
- O critério para descarte do endereço na tabela varia de fabricante ou modelo, sendo mais comuns o uso de uma fila FIFO ou um critério estatístico em que se descarta aqueles que em uma média temporal geraram um menor tráfego. De qualquer modo, a necessidade de se descartar entradas na tabela acabará por acarretar no aumento do tráfego "*broadcast*" da rede, o que é altamente indesejável.

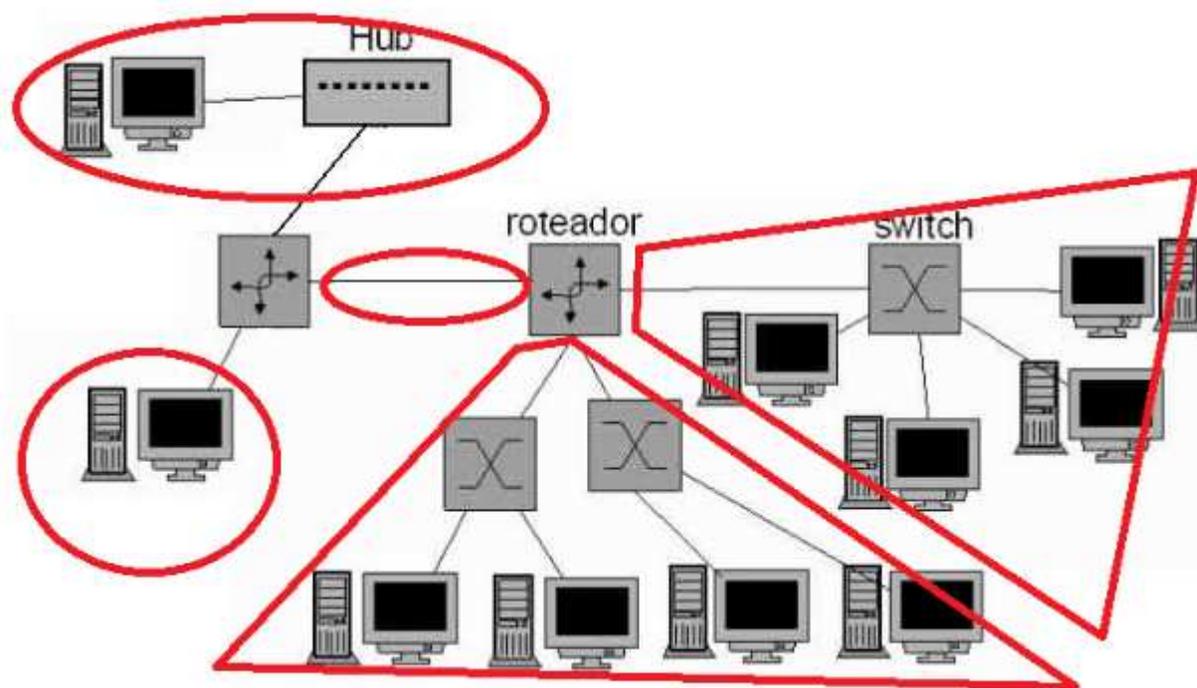
Switches (cont.)

- Por esta razão, ao se escolher um *switch*, recomenda-se dimensionar o tamanho da rede e escolher um modelo cuja capacidade de armazenamento de endereços seja igual ao maior ao número de dispositivos da mesma.
- Características adicionais:
 - Propagam *broadcasts* para todas as suas portas.
 - Permitem definir múltiplos domínios de *broadcast*. Cada um destes domínios define o que é denominado de VLAN (Virtual LAN).
 - Fazem detecção de erros. Para isso, analisam o FCS (*frame check sequence*), que se encontra no final do quadro *ethernet*.
 - Podem operar em modo *full-duplex* (a estação pode enviar e receber dados ao mesmo tempo – violação do padrão Ethernet), o que dobra a largura de banda disponível.

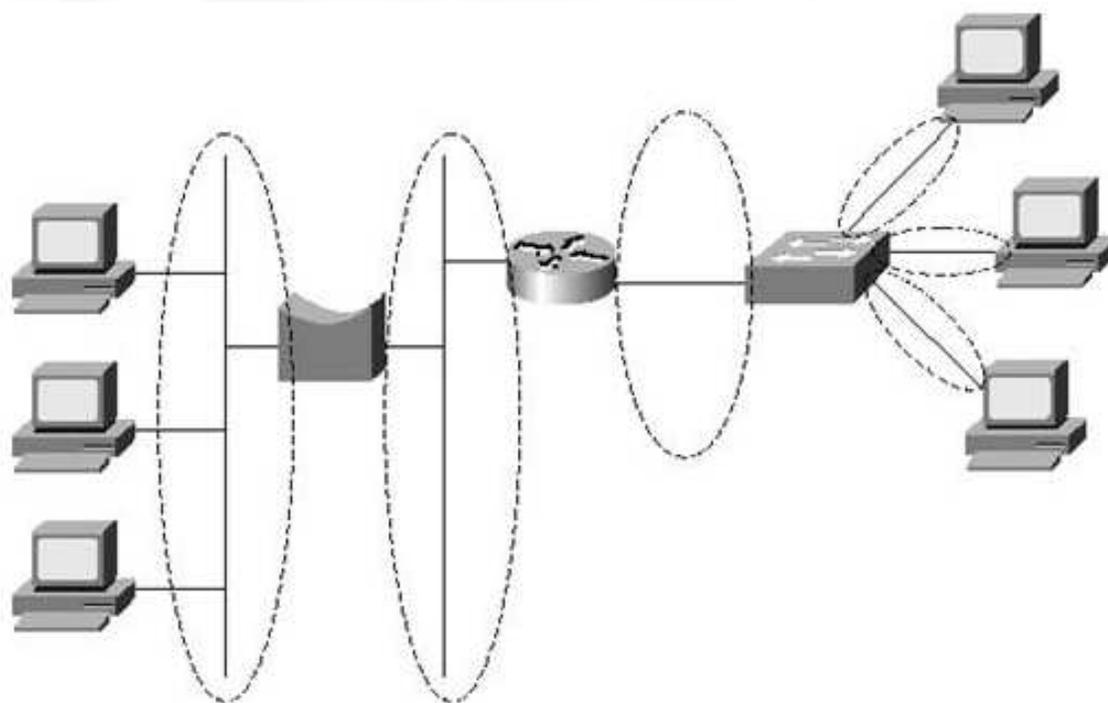
Switch x Domínio de Colisão



Switch x Domínio de Broadcast



Exemplo



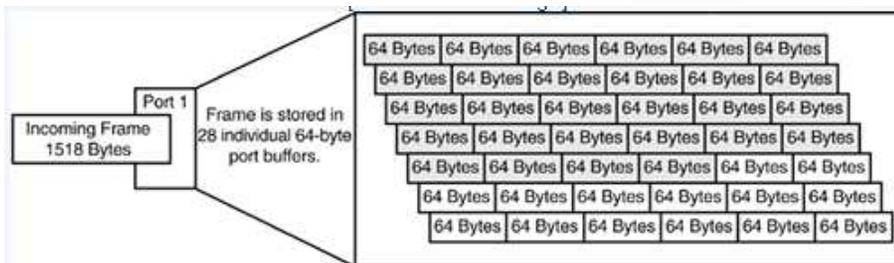
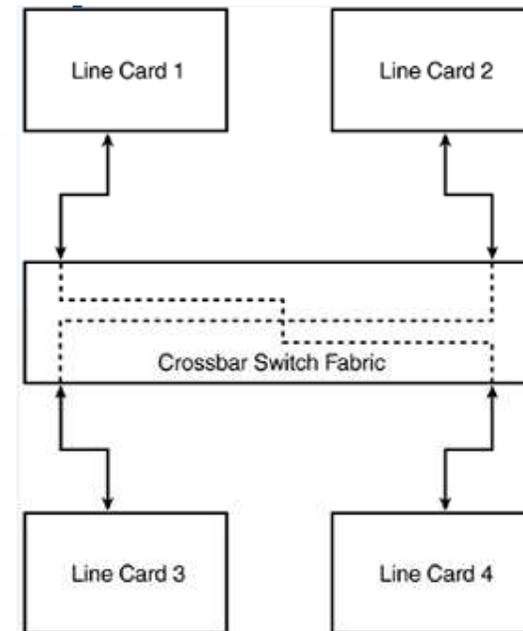
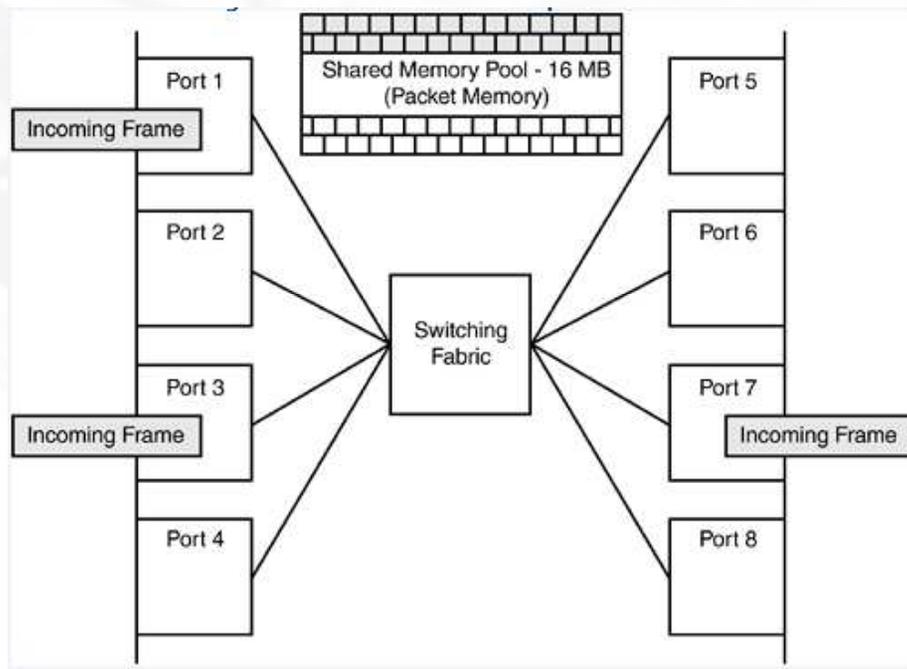
Switch Backplane

- O *backplane* é nada mais do que o "*switch fabric*", ou seja, o elemento que permite a comunicação simultânea entre as portas do *switch*.
- A capacidade de repasse de pacotes do *backplane* de um *switch* deverá ser de, pelo menos, a metade da soma das taxas máximas de transmissão de todas as portas do *switch*, se estas forem *half duplex*.
- Se as portas do *switch* puderem operar em *full duplex*, a capacidade de repasse dos pacotes deverá ser igual ou maior à soma das taxas máximas de transmissão das portas do *switch*.

Switch Backplane (cont.)

- Por exemplo, um *switch* de 12 portas Fast Ethernet *half duplex* deverá possuir um *backplane* com a capacidade de efetuar o repasse dos quadros a uma velocidade mínima de 600 Mbps, o que corresponde à situação crítica de haver 6 portas recebendo quadros, e estes sendo redirecionados às outras 6 portas.
- Se o *backplane* não suporta o fluxo agregado de 600 Mbps que está recebendo, terá que guardar em memória alguns dos quadros, a fim de evitar o seu descarte. Neste caso o *backplane* torna-se o gargalo da rede.
- Um *switch* que, por maior que seja o tráfego recebido, o *backplane* nunca será o gargalo da rede é chamado *Non Blocking*.

Switch Backplane (cont.)

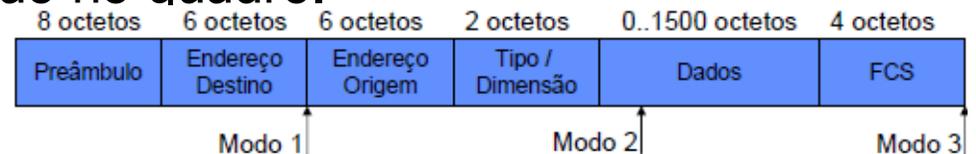


Características dos Switches Modernos

- Diferentes taxas de dados suportadas simultaneamente (10, 100, 1000, 10000 Mbps).
- Operação full duplex.
- QoS para garantia de *delays* máximos em aplicações de tempo real.
 - Mecanismos de filas
 - Controle de fluxo
- Segurança
 - Mapeamento estático (DA associado com source port)
 - Porta segura (número limitado de usuários pré-definidos por porta)
- Diferentes esquemas de *forwarding* (modos de comutação)
 - Store-and-forward, Cut-throught e Fragment-free
- Suporte a VLAN (trunking).
- Spanning tree (deteção e eliminação de *loops*).

Modos de Comutação

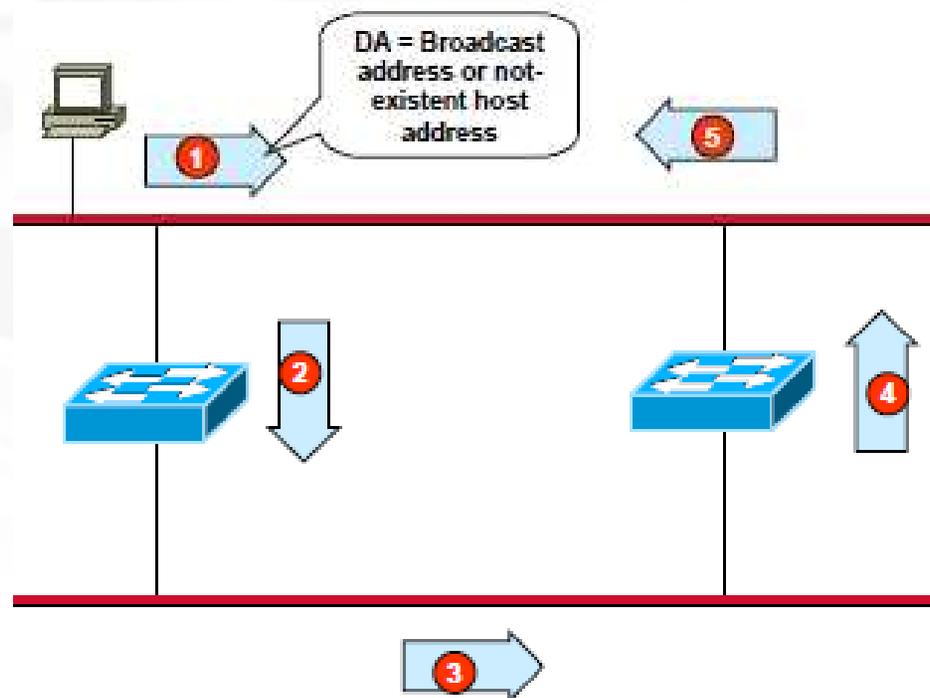
- Cut-through
 - Envia logo após já ter o endereço MAC destino disponível. O frame é encaminhado através do *switch* antes que todo o frame seja recebido.
 - Este modo diminui a latência de transmissão mas não garante erros e nem colisão.
- Store-and-forward
 - O frame inteiro é recebido antes que qualquer encaminhamento seja feito. Os endereços destino e fonte são lidos e filtros são aplicados antes do frame ser encaminhado. CRC Check é realizado.
 - Garante que não houve erros e nem colisão no quadro.
- Fragment-Free
 - Envia logo após receber 64 bytes (dimensão mínima do quadro).
 - Garante que não houve colisão no quadro.



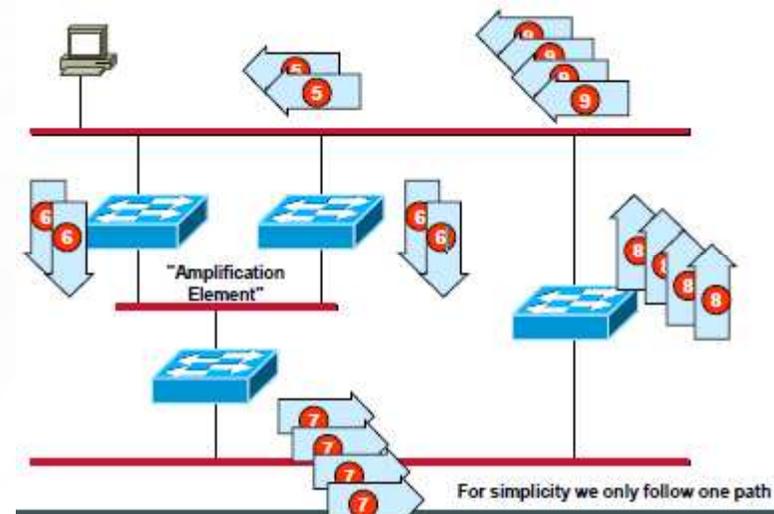
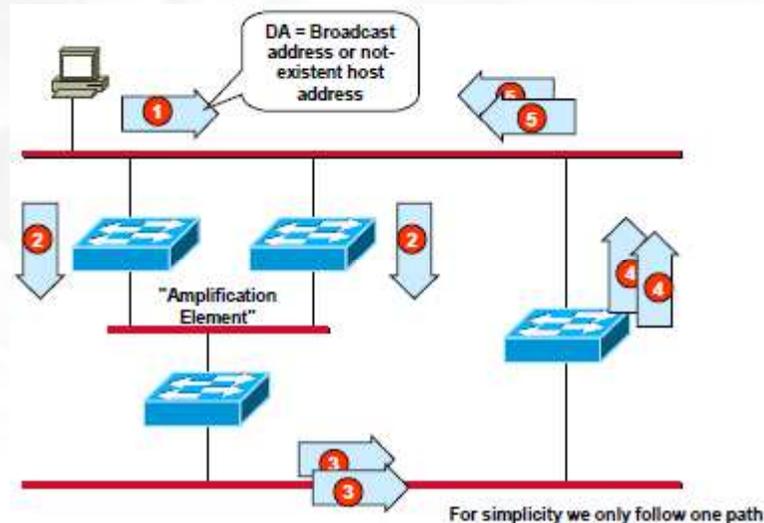
O Problema dos Loops em Pontes/Switches

- Como pode ser notado, pontes e switches não sabem (não aprendem) a topologia da rede. Sabem apenas associar um endereço destino associado a uma porta. Não existe, por exemplo, maneira de se determinar o melhor caminho numa rede segmentada.
- Adicionalmente, *frames* podem entrar em *loop* infinito. *Frames broadcast*, em particular, não têm destino definido e poderiam ficar sendo encaminhados sobre todos os caminhos paralelos indefinidamente".
 - Observe que não existe TTL ou "hop count" em um frame Ethernet.

Loop Infinito ("Endless Circling")

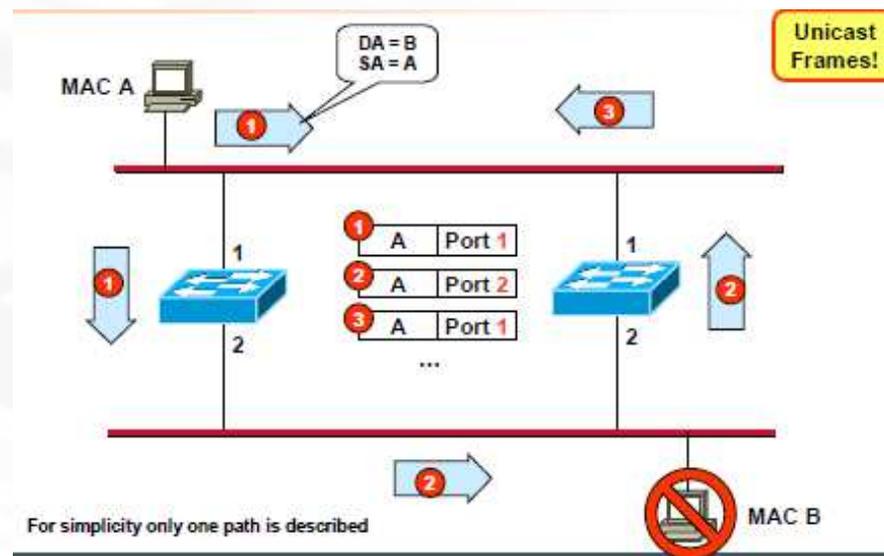


Broadcast Storm



- "Broadcast storm" pode ser vista como uma situação mais crítica de *loop* infinito ("*enhanced endless circling problem*"), ocasionada pela presença de um "elemento amplificador" (vide figura).
- Em condição de *broadcast storm*, o tráfego pode chegar rapidamente a dezena de milhares de frames por segundo. Nesta situação, todos os equipamentos têm queda de desempenho. Esta situação pode ocorrer em redes complexas, onde ciclos podem ser fechados através de equipamentos com múltiplas placas de rede ligadas a diferentes switches.

Mutual Table Rewriting



- Host A envia um frame unicast para o host B. O host A é associado à porta 1 nos dois switches. Como o frame não é consumido pelo host B (subitamente desconectado da rede), a ponte encaminha este frame para o segmento superior.
- A tabela da ponte é reescrita, agora com host A associado à porta 2.
- Este ciclo continua ...

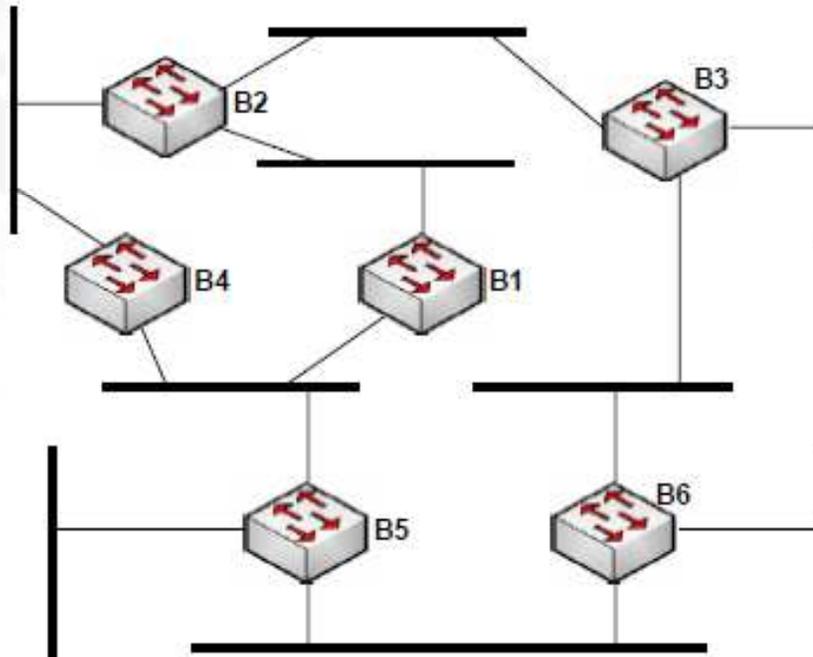
O Protocolo Spanning Tree (STP)

- Como visto, a existência de caminhos paralelos leva a problemas severos em redes segmentadas por bridges/switches.
- Radia Perlman (DEC) criou uma solução fácil para o problema da redundância: o protocolo *Spanning Tree*.
- Basicamente, o STP detecta e elimina *loops* em redes segmentadas, ao mesmo tempo que mantém a possibilidade de conexões redundantes entre *switches* (o que implica em maior tolerância a falhas).
- Na ocorrência de malhas fechadas, algumas portas são desativadas para abrir eliminar os loops (caminhos redundantes de maior custo são eliminados do grafo).

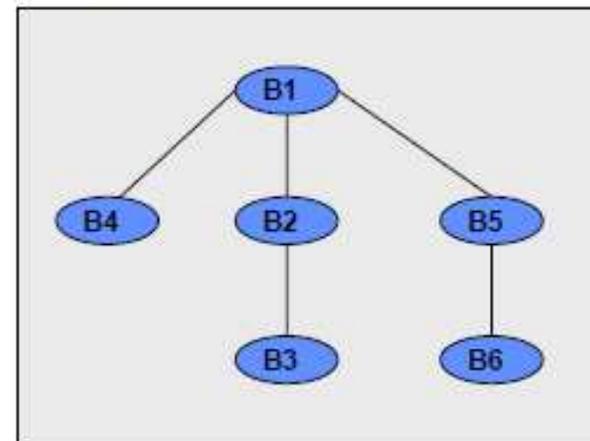
O Protocolo Spanning Tree (STP) (cont.)

- É criada uma árvore - um sub-conjunto do grafo original - que não contém ciclos. Interfaces (portas) que não fazem parte desta árvore são desconectadas e *loops* são removidos.
- Em caso de falha em um caminho ativo, o STP automaticamente ativa caminhos redundantes.
- O STP é implementado apenas nas *bridges* e *switches*, e não em *hosts* e é padronizado pelo IEEE (IEEE 802.1d).

O Protocolo Spanning Tree (STP) (cont.)



Topologia de Rede em Árvore



Ingredientes (Conceitos) do SPT

- Bridge ID
- Path Cost
- BPDU
- Root Bridge (Root Port)



Bridge ID

- É uma combinação de um número de prioridade de 16 bits e o menor endereço MAC existente em qualquer porta da bridge.
- O Bridge ID é determinado automaticamente usando a prioridade default de 32768.
 - Cisco Catalyst 5000/6000 usam um dos 1024 endereços MAC associados ao *backplane* ou ao módulo supervisor.

Bridge Priority	MAC
2 bytes. Range: 0 a 65536 (default: 32768)	6 bytes. From backplane

Path Cost

- Serve para determinar quão próximo da *bridge* raiz da árvore STP encontra-se uma particular *bridge*.

- A cada porta está associada um custo, também determinado automaticamente pela fórmula

$$\text{Port Cost} = 1000 / \text{Bandwidth (em Mbps)}$$

- STP Costs padronizados pelo IEEE:

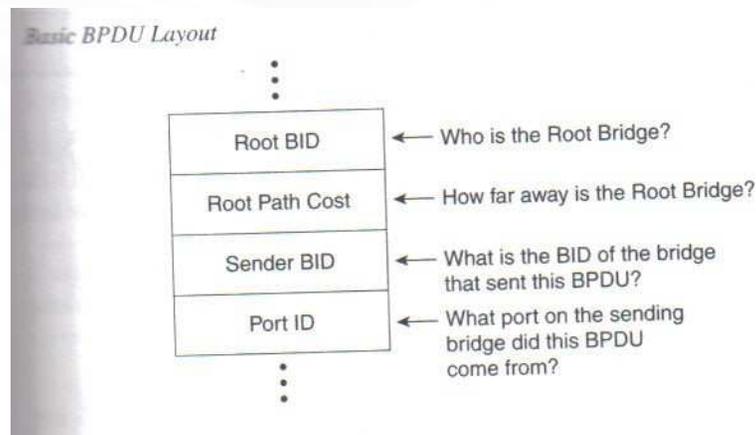
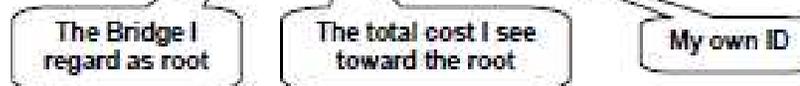
Bandwidth (Mbps)	Custo
4	250
10	100
16	62
45	39
100	19
155	14
622	6
1.000	4
10.000	2

BPDU – Bridge Protocol Data Unit

- BPDUs constituem as mensagens básicas do protocolo.
- Cada *bridge* envia periodicamente (hello time default = 2 segundos) BPDUs encapsuladas em frames multicast Ethernet e transmitidas entre os switches/pontes.
- As BPDUs contém todas as informações necessárias para se construir a *spanning tree*.
- A troca de BPDUs resulta no seguinte:
 - Eleição da *root bridge*;
 - Eleição de uma *designated bridge* para cada um dos segmentos da LAN;
 - A remoção de *loops* colocando portas redundantes do *switch* em um estado de *backup*.

Formato da BPDU

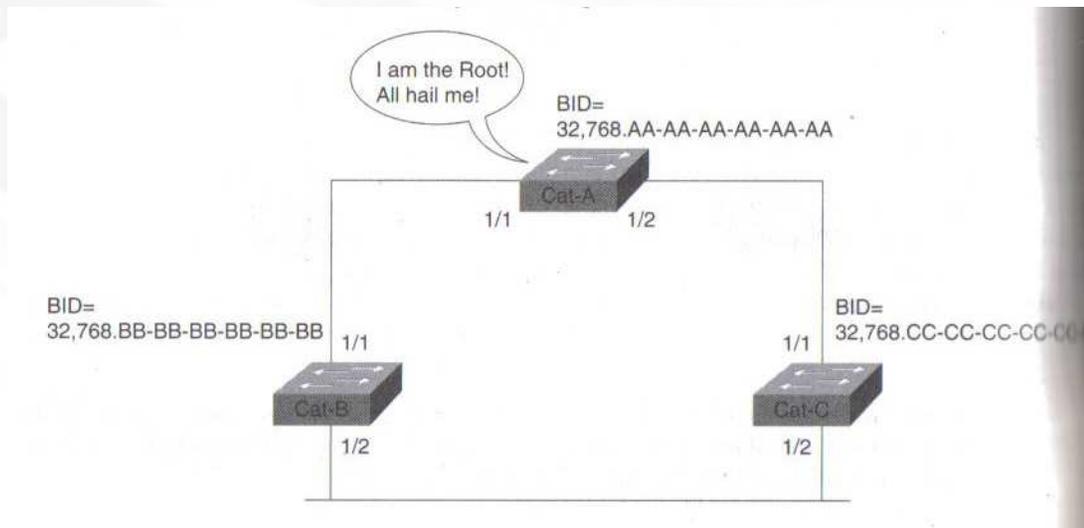
Prot. ID	Prot. Vers.	BPDU Type	Flags	Root ID	Root Path Costs	Bridge ID	Port ID	Mess. Age	Max Age	Hello Time	Fwd. Delay
2 Byte	1 Byte	1 Byte	1 Byte	8 Byte	4 Byte	8 Byte	2 Byte	2 Byte	2 Byte	2 Byte	2 Byte



Passos Básicos do STP

- Passo 1: eleição da *root bridge*
 - Inicialmente, a rede segmentada elege uma única *root bridge*.
- Passo 2: eleição das *root ports*
 - Cada non-root bridge elege uma única root port, que é a porta mais próxima da root bridge (caminho de menor custo até a *root bridge*);
- Passo 3: eleição das *designated ports*
 - Lowest sender BID

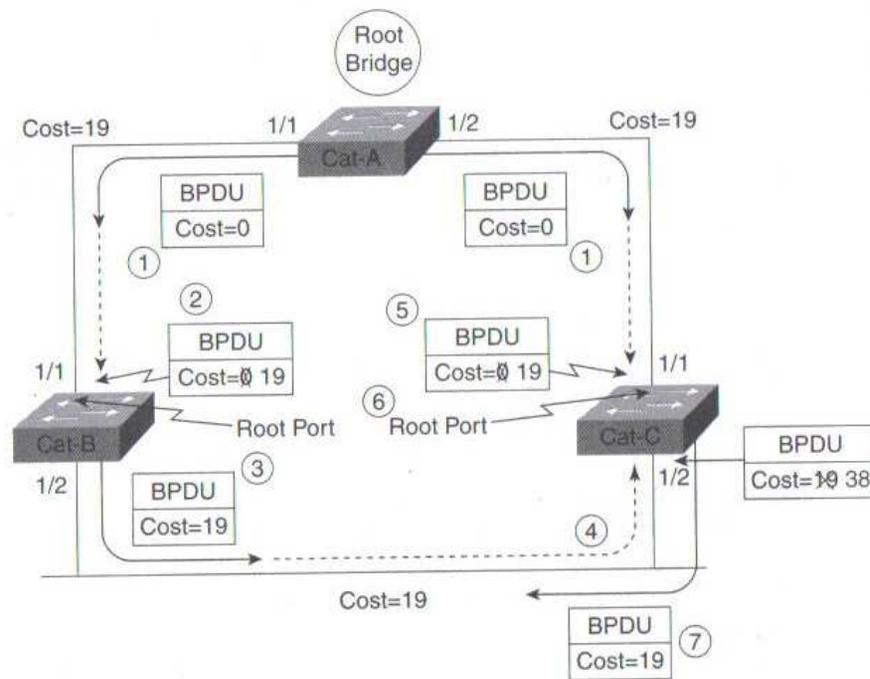
Passo 1: Eleição da Root Bridge



Passo 1: Eleição da Root Bridge

- A *bridge* com o menor Bridge ID (BID) da rede é sempre escolhida como a *root bridge*.
- Inicialmente, cada *bridge* da rede assume a si mesma como *root*. Em outras palavras, ao ser iniciada (*boot*) ela coloca o seu Bridge ID em ambos os campos Root BID e Sender BID na BPDU correspondente.
- Suponha que Cat-B se anuncie como Root Bridge (isto é, envie uma correspondente BPDU a cada 2 segundos). Poucos minutos depois Cat-C se anuncia como Root Bridge. Quando a Cat-C BPDU chega a Cat-B, este descarta a BPDU porque ele possui um menor BID. Ao receber o Cat-B BPDU, Cat-C percebe que existe outra bridge mais importante e começa então a transmitir Cat-C BPDUs que listam Cat-B como Root ID e Cat-C como Sender BID.
- Minutos depois, Cat-A se inicia, assume que é Root Bridge e se anuncia (envia BPDUs) como tal. Logo que esses BPDUs chegarem a Cat-B e Cat-C eles abdicarão da posição de Root Bridge em favor de Cat-A. A partir deste momento, todos os três switches/pontes estarão anunciando Cat-A como Root Bridge e eles próprios como Sender ID.

Passo 2: Eleição das Root Ports



Passo 2: Eleição das Root Ports (cont.)

- Cada *non-Root Bridge* deve eleger uma única *root port* dentre as suas interfaces.
- As *root ports* são usadas no cálculo do caminho de menor custo até a *root bridge* (é a porta mais próxima da *root bridge* em cada non-root bridge).
- Quando Cat-A envia BPDUs o campo de Root Path Cost é preenchido com o valor zero já que Cat-A é a root bridge.
- Quando Cat-B recebe esses BPDUs ele adiciona o Path Cost da porta 1/1 ao Root Path Cost contido na BPDU recebida de Cat-A ($0+19=19$). Cat-B passa a enviar BPDUs pela porta 1/2 com o campo de Root Path Cost tendo o valor igual a 19.

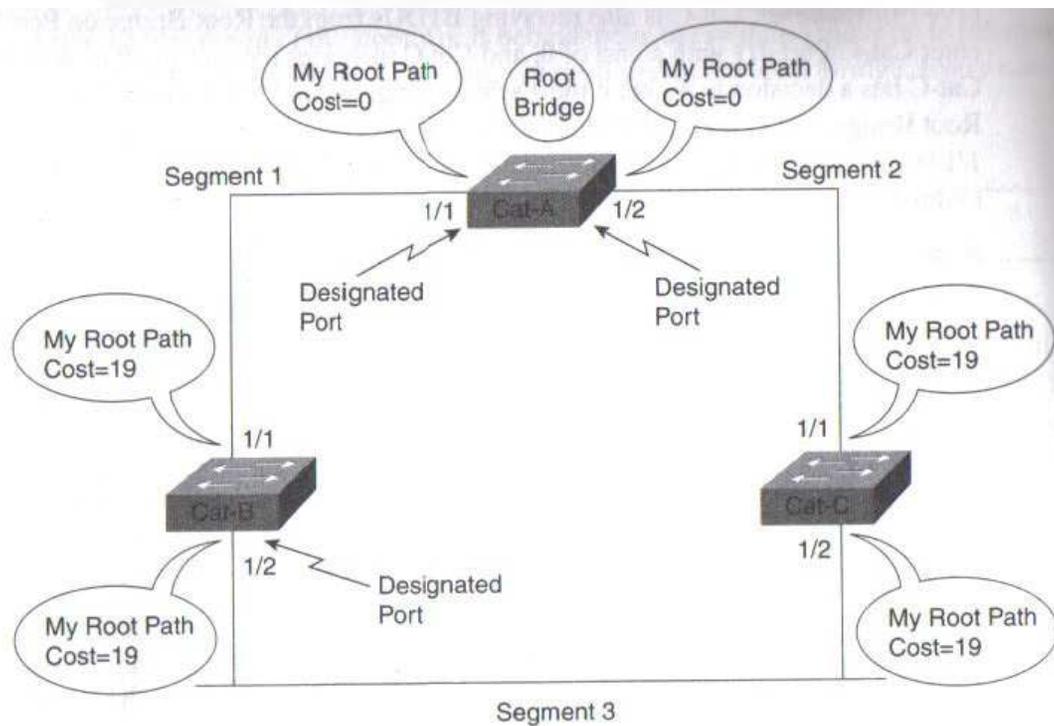
Passo 2: Eleição das Root Ports (cont.)

- Quando Cat-C recebe esses BPDUs de Cat-B ele incrementa internamente o seu Root Path Cost para 38 ($19+19=38$). Como Cat-C também está recebendo BPDUs com custo igual zero da Root Bridge na Port 1/1 (e internamente ele está somando o custo de 19 da porta) ele tem uma decisão a fazer: ele tem que selecionar uma única Root Port, isto é, a porta que está mais próxima da root bridge. Cat-C compara um Root Path Cost de 19 na porta 1/1 com um Root Path Cost igual a 38 na porta 1/2. A porta 1/1 (de menor custo) é eleita a Root Port. Cat-C passa então a anunciar um Root Path Cost de 19.
- Da mesma maneira, a porta 1/1 se torna a Root Path Cost da ponte Cat-B. Observe que STP costs são incrementados à medida que BPDUs são recebidos numa porta e não quando eles são enviados pela porta.

Passo 2: Eleição das Root Ports (cont.)

- Reforçando a diferença entre *Path Cost* e *Root Path Cost*:
 - *Path Cost* é um valor assinalado a cada porta, sendo dependente da largura de banda do link. Ele é adicionado ao BPDU recebido naquela porta a fim de calcular o custo do caminho da bridge até a root bridge (ou seja, o Root Path Cost).
 - *Root Path Cost* é definido como o custo acumulado até a Root Bridge. Em uma BPDU, este é o valor transmitido no campo de custo. Este valor é calculado adicionando-se o Path Cost que está sendo recebido na porta ao valor contido na BPDU.

Passo 3: Eleição das Designated Ports



Passo 3: Eleição das Designated Ports (cont.)

- A prevenção de loops do STP torna-se mais evidente no passo 3, com a eleição das *Designated Ports* de cada segmento.
- Afinal, numa rede rodando STP:
 - Root e Designated Ports encaminham tráfego;
 - Non-Designated Ports bloqueiam tráfego.

Passo 3: Eleição das Designated Ports (cont.)

- Segmento 1:
 - Existem duas portas no segmento: Cat-A:Port 1/1 (com Root Cost igual a 0) e Cat-B:Port 1/1 (com Root Cost igual a 19).
 - Cat-A:Port 1/1 é selecionada como Designated Port do segmento 1 por ter o menor Root Path Cost. Observe que toda porta ativa em uma Root Bridge se torna uma Designated Port.
- Segmento 2:
 - Raciocínio semelhante. Cat-A:Port 1/2 é eleita a Designated Port do segmento 2.
- Segmento 3:
 - Existe um empate: tanto Cat-B:Port 1/2 quanto Cat-C:Port 1/2 possuem um Root Path Cost de 19. Neste caso o desempate é feito usando a seguinte sequência de prioridade: menor Root BID, menor Path Cost até a Root Bridge, menor Sender BID e menor Port ID. Usando o terceiro critério, Cat-B:Port 1/2 é eleita.
 - Cat-C:Port 1/2 se torna um non-Designated Port.

Resumindo ...

Typical loop problems:

- Broadcast Storm
- Bridge Table Corruption, endless unicast circulation

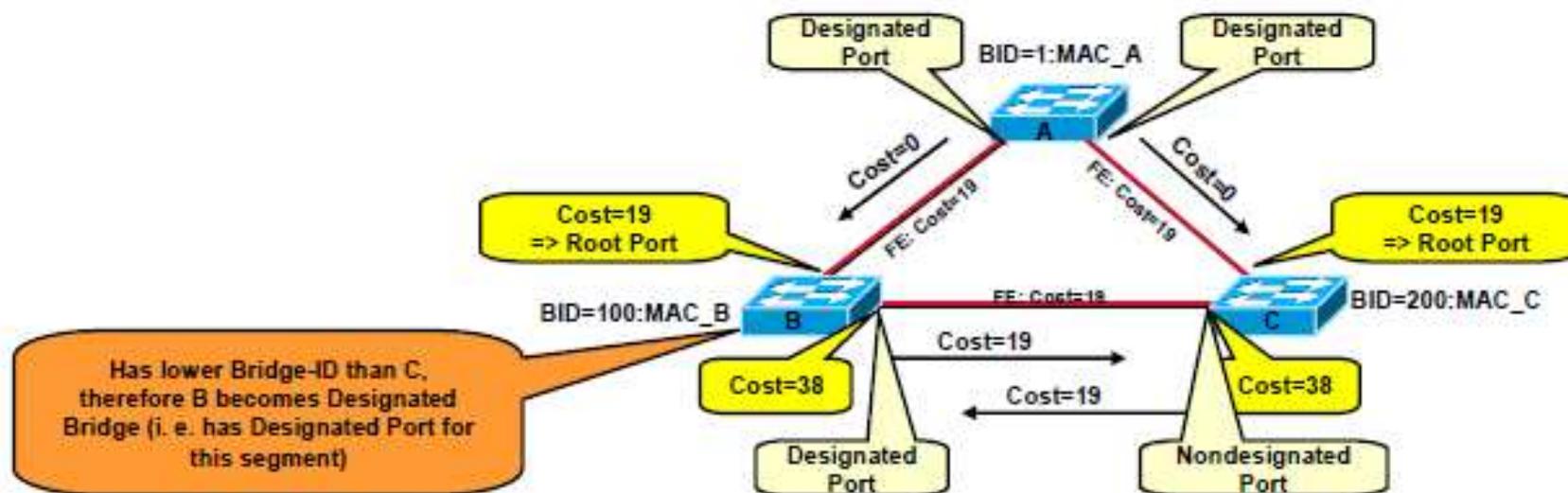
Three steps to create spanning tree:

1. Elect Root Bridge (Each L2-network has exactly one Root Bridge)
2. Elect Root Ports (Each non-root bridge has exactly one Root Port)
3. Elect Designated Ports (Each segment has exactly one Designated Port)

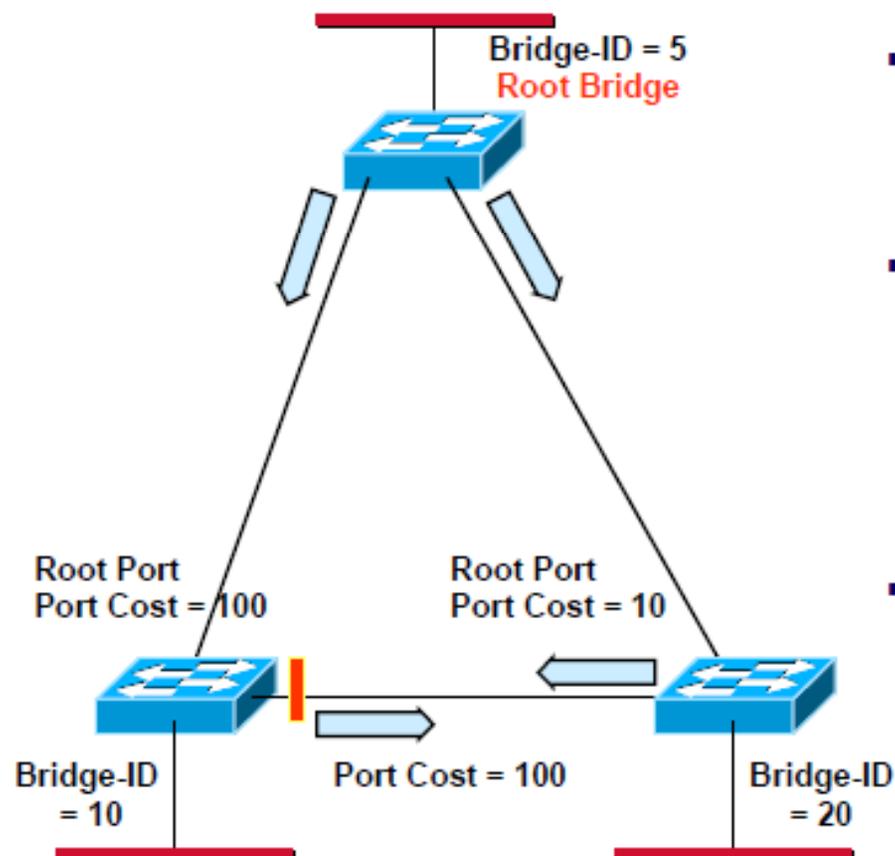
The "better" BPDU contains

1. Lowest Bridge ID
2. Lowest Path Cost to Root Bridge
3. Lowest Sender Bridge ID
4. Lowest Port ID

} Only used as tie breaker to determine Designated Port



Resumindo ...

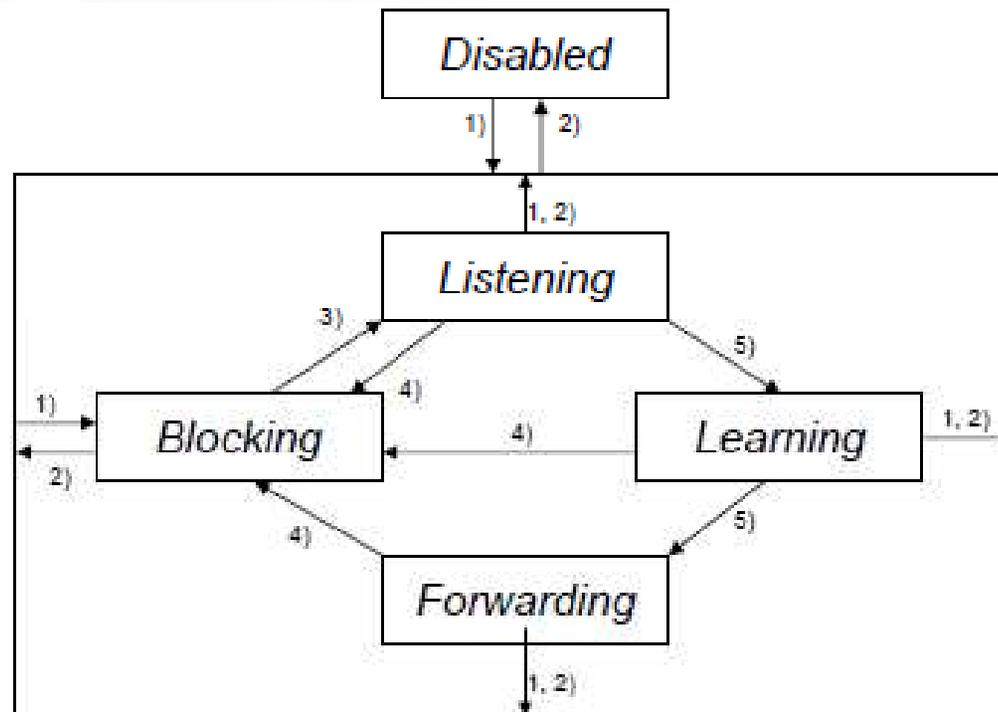


- First a **Root Bridge** is determined
 - Initially every bridge assumes itself as root
 - The bridge with lowest Bridge-ID wins
- Then the root bridge triggers BDPU sending (hello time intervals)
 - Received at "**Root Ports**" by other bridges
 - Every bridge adds its own port cost to the advertised cost and forwards the BDPU
- On each LAN segment one bridge becomes **Designated Bridge**
 - Having lowest total root path cost
 - Other bridges set redundant ports in **blocking state**

Estados do STP

- Embora Forwarding e Blocking sejam os únicos estados vistos em uma rede estável, existem na verdade 5 estados no STP:
 - Forwarding: sending/receiving user data
 - Learning: building bridging table
 - Listening: building "active" topology
 - Blocking: receives BPDUs only
 - Disable: administratively down

Estados das Portas e suas Transições



- 1) Activação por gestão ou por iniciação
- 2) Desactivação por gestão ou falha
- 3) Processo de aprendizagem do algoritmo
- 4) Bloqueio pelo algoritmo
- 5) Transição por expiração de tempo de aprendizagem

Estados das Portas e suas Transições (cont.)

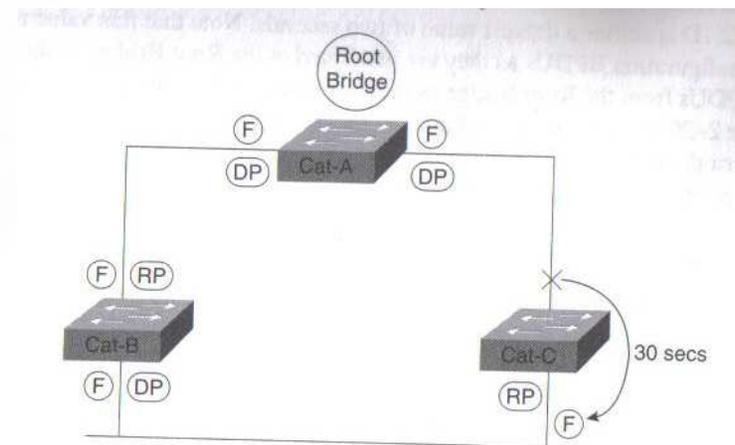
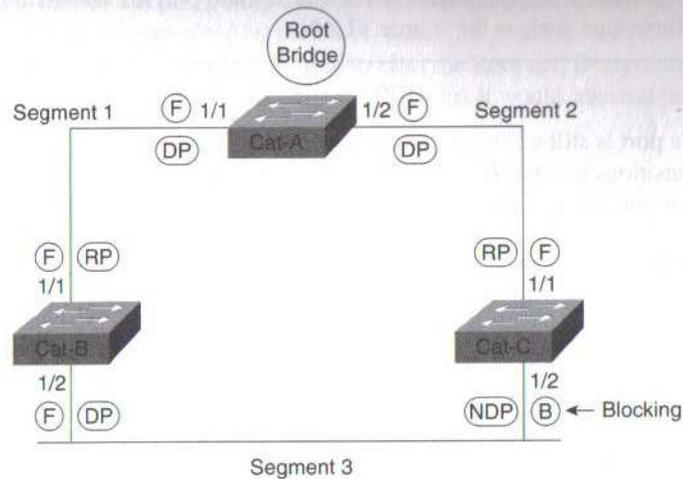


Table 6-4 documents the symbols used throughout the book to represent Spanning Tree states.

STP State and Port Symbols

State/Port	Symbol
Blocking	B
Forwarding	F
Designated Port	DP
Root Port	RP
Non-Designated Port	NDP